Università degli Studi di Cagliari

Dipartimento di Matematica e Informatica
Dottorato di Ricerca in Informatica
Ciclo XXV

Ph.D. Thesis

A theory of agreements and protection

S.S.D. INF/01

Candidate
Tiziana Cimoli

Supervisor
G. Michele Pinna

PhD Coordinator
G. Michele Pinna

Final examination academic year 2011/2012
June 3, 2013

Abstract

In this thesis we propose a theory of contracts. Contracts are modelled as interacting processes with an explicit association of obligations and objectives. Obligations are specified using event structures. In this model we formalise two fundamental notions of contracts, namely *agreement* and *protection*.

These notions arise naturally by interpreting contracts as multi-player concurrent games. A participant *agrees* on a contract if she has a strategy to reach her objectives (or to make another participant sanctionable for a violation), whatever the moves of her counterparts. A participant is *protected* by a contract when she has a strategy to defend herself in all possible contexts, even in those where she has not reached an agreement.

When obligations are represented using classical event structures, we show that agreement and protection mutually exclude each other for a wide class of contracts.

To reconcile agreement with protection we propose a novel formalism for modelling contractual obligations: event structures with circular causality.

We study this model from a foundational perspective, and we relate it with classical event structures. Using this model, we show how to construct contracts which guarantee both agreement and protection.

We relate our contract model with Propositional Contract Logic, by establishing a correspondence between provability in the logic and the notions of agreement and strategies.

This is a first step towards reducing the gap between two main paradigms for modelling contracts, that is the one which interprets them as interactive systems, and the one based on logic.

Contents

List of Figures

Glossary of notation

$\#$	binary conflict relation (p. 23)
$\vdash$	enabling (p. 24)
$\Vdash$	circular enabling (p. 70)
$\rightarrow_{\mathcal{E}}$	LTS semantics of an ES $\mathcal{E}$ (p. 25)
$\rightarrow_{\mathcal{E}}$	LTS semantics of a CES $\mathcal{E}$ (p. 92)
$\rightharpoonup_{\mathcal{U}_{\mathcal{E}}}$	LTS of urgent events (p. 98)
ε	empty sequence (p. 21)
$\mid$	interleaving operator between sequences (p. 21)
$\overline{\sigma}$	set of elements of a sequence (p. 21)
$\sigma\downarrow$	sequence σ with duplicate events removed (p. 75)
$\langle e_0 e_1 \ldots \rangle$	sequence (p. 21)
$\Gamma(\sigma)$	least credit of trace σ (p. 76)
$\Gamma^+(C, X, e)$	credit when adding event e to X-trace with events C (p. 91)
$\Gamma^-(C, X, e)$	credit when removing an event e to X-trace with events C (p. 77)
$\twoheadrightarrow$	contractual implication (p. 29)
$[\mathcal{E}]_{\mathcal{F}}$	PCL encoding for configuration derivation of a CES $\mathcal{E}$ (p. 120)
$[\mathcal{E}]_{\mathcal{R}}$	PCL encoding for reachable-events derivation of a CES $\mathcal{E}$ (p. 117)
$[\mathcal{E}]_{\mathcal{U}}$	PCL encoding for urgent-events derivation of a CES $\mathcal{E}$ (p. 126)
Con	set of the finite conflict-free subsets of E (p. 23)
CES	event structure with circular enabling (p. 70)

CF(X)	predicate true iff X is conflict-free (p. 23)
$DF(\sigma)$	sequence σ is duplicate-free (p. 21)
$\mathcal{E}$	event structure (p. 24)
$\mathcal{E}(\mathcal{F})$	the event structure obtained by the set $\mathcal{F}$ (p. 27)
$\mathcal{E}(\Phi)$	conflict-free CES generated from the payoff function Φ (p. 111)
$\hat{\mathcal{E}}(\mathcal{F})$	the CES obtained by the set $\mathcal{F}$ (p. 83)
ES	event structure (p. 24)
$\mathcal{F}_{\mathcal{E}}$	set of set of events of the ES or CES $\mathcal{E}$, called configurations (p. 71)
$\mathcal{F}_{\mathcal{E}}^{P}(X)$	trace with past P and credit X (p. 95)
$\mathcal{F}_{\mathcal{E}}(X)$	set of set of events of the ES or CES $\mathcal{E}$, called configurations (p. 71)
LTS	labelled transition system (p. 22)
$\mathcal{P}^{X}$	prudent events with past X (p. 106)
$\wp_{fin}(X)$	set of all finite subsets of X (p. 21)
$\wp_{\widehat{fin}}^{\leq 1}(X)$	set of all finite subsets of X of cardinality at most 1 (p. 21)
PCL	propositional contract logic (p. 29)
$\mathcal{R}_{\mathcal{E}}^{C}(X)$	reachable events with past C and credit X (p. 96)
$\mathcal{R}_{\mathcal{E}}$	set of reachable events of $\mathcal{E}$ (p. 85)
$\hat{\mathcal{R}}(X)$	set of X-reachable event in a conflict-free CES (p. 86)
$\mathcal{T}_{\mathcal{E}}^{C}(X)$	trace with past C and credit X (p. 95)
$\mathcal{T}_{\mathcal{E}}$	set of all the traces of the CES $\mathcal{E}$ (p. 71)
$\mathcal{T}_{\mathcal{E}}(X)$	set of all the X-traces of the CES $\mathcal{E}$ (p. 71)
$\mathcal{U}_{\mathcal{E}}^{C}(X)$	set of urgent events in (C, X) (p. 97)
$\hat{\mathcal{U}}_{\mathcal{E}}^{C}(X)$	set of urgent events in (C, X) (p. 101)

Chapter 1

Introduction

1.1 Motivations

Many of today human activities, from business and financial transactions, to collaborative and social applications, run over complex interorganizational systems, based, for instance, on service-oriented computing (SOC) and cloud computing technologies. These technologies foster the implementation of complex software systems through the composition of basic building blocks, called *services*. Ensuring reliable coordination of these components is fundamental to avoid critical, possibly irreparable problems, ranging from economic losses in case of commercial activities, to risks for human life in case of safety-critical applications.

Ideally, in the SOC paradigm an application is constructed by dynamically discovering and composing services published by different organizations. Services have to *cooperate* to accomplish the overall tasks, while at the same time they have to *compete* to achieve the specific goals of their stakeholders. These goals may be conflicting, and likely they are in the case of mutually distrusted organizations. Thus, services must play a double role: while cooperating together, they have to protect themselves against other service's misbehavior (either unintentional or malicious).

The lack of precise guarantees about the reliability and security of services is a main deterrent for industries wishing to move their applications and business to the cloud. Quoting from [A+10]:

> "absent radical improvements in security technology, we expect that users will use contracts and courts, rather than clever security engineering, to guard against provider malfeasance".

A key problem is then how to drive safe and fair interactions among distributed participants which are possibly mutually distrusted, and have possibly conflicting individual goals. In addition to the well-known difficulties of distributed software systems (distribution, concurrency, heterogeneity, mobility, *etc.*), services, cloud components and infrastructures are often under the governance of different providers, possibly competing among each other.

Analysis and verification techniques cannot completely solve these issues, as they can only be applied to the software components under one's control, while no assumptions can be made about the components made available by other providers. Therefore, standard compositional techniques have to be adapted to cope with the situation where providers fail to keep the promises made, or even choose not to.

Furthermore, services and cloud applications are increasing in size, and the larger an application, the greater the probability that some of its components deviates from the expected behaviour (either because of unintentional bugs, or maliciousness of a competing organization).

To sum up, since services act in a competitive, possibly adversarial setting, the possibility that a service behaviour may diverge from the expected one is quite realistic. To protect themselves against possible misconducts, services should postpone actual collaboration until reaching an agreement on the mutually offered behaviour. This requires a preliminary step, where each service declares her promised behaviour, i.e. her *contract*.

1.2 Contract-oriented computing

We believe that contracts can play an important role in the specification and implementation of reliable and secure distributed systems.

We envision a *contract-oriented computing* paradigm [BTZ12a], where interactions are driven by contracts which formalise Service-Level Agreements. Contracts specify the behavior of a software component, from the point of view of the interactions it may participate in, and the goals it tries to reach. Differently from most of the approaches based on behavioural types [HYC08], which use contracts only in the "matchmaking" phase, a contract-oriented component is not supposed to be *honest*, in that it may not keep the promises made.

In a contract-oriented application, participants advertise their contracts to some *contract brokers*, which are the contract-oriented analogous of service repositories in the Web Service paradigm (see e.g. Fig. 1.1). Participants wait until the contract broker finds an *agreement* among the contracts in its hands. An agreement is a property of contracts which guarantees that each honest participant may reach her objectives, provided that the other participants cooperate honestly. If an honest participant does not reach her objectives, then some other participant can be blamed. When an agreement is found, a session is created among the participants involved in the contract, so that they can interact.

An external judge may inspect the contract and the status of the session. In the case a violation is found, the judge will eventually provide the prescribed compensations/punishments.

The underlying assumption of this paradigm is that participants trust the contract broker. In a context populated by attackers, it may happen that a dishonest contract broker creates a fraudulent session, making participants interact in the ab-

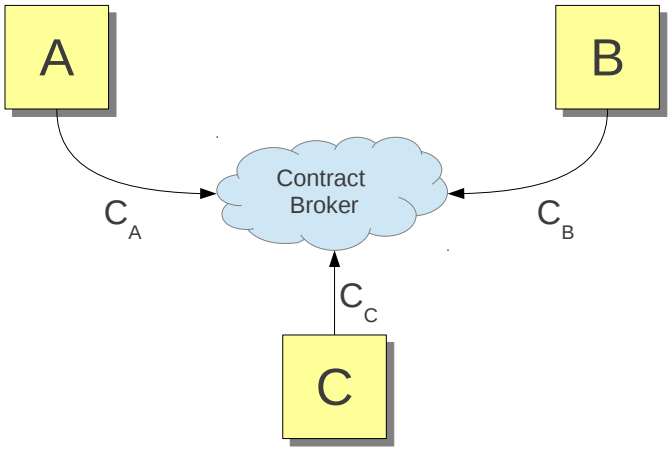


Figure 1.1: A contract broker collects the contracts advertised by participants.

sence of an agreement. In this way, the contract broker may swindle an unaware participant, possibly with the cooperation of an accomplice. Note that the broker or his accomplice may perform this scam while adhering to their contracts, and so they will not be liable for violations.

A crucial problem is then how to devise contracts which protect participants from malicious contract brokers, while at the same time allowing honest brokers to find agreements. A good contract should allow a participant to reach her goals in contexts where the other participants are cooperative, and prevent her from performing imprudent actions which could be exploited by malicious participants.

1.3 Contribution

In this thesis we propose a foundational model for contracts which uses event structures (ES [Win86]) to specify participants' obligations. By abstracting away from the concrete details of contract languages, our model is a first step towards a unifying framework for reasoning about contracts, in the same spirit that event structures can be used as an underlying semantics for a variety of concrete models of concurrency.

Event structures can provide a basic semantic model for assume/guarantee rules, by interpreting the enabling $b \vdash a$ as the contract clause: "I will do a *after* you have done b". However, event structures do not capture a typical aspect of contracts, i.e. the capability of reaching an agreement when the assumptions and the guarantees of the parties mutually match. For instance, in the event structure with enablings $b \vdash a$ and $a \vdash b$, none of the events a and b are reachable, because of the circularity of the constraints. An agreement would still be possible if one of the parties is willing to accept a weaker contract. Of course, the contract "I will do b" (modelled

as $\vdash b$) will lead to an agreement with the contract $b \vdash a$, but it offers no protection to the participant who offers it, e.g. it can be stipulated without having anything in return.

We provide a formal definition for the two key notions intuitively introduced above, i.e. *agreement* and *protection*. To do that, we borrow techniques from game theory, by interpreting contracts as multi-player concurrent games.

A participant agrees on a contract if she has a strategy to reach her objectives (or make another participant chargeable for a violation), whatever the moves of her adversaries. A participant is protected by a contract when she has a strategy to defend herself in all possible contexts, even in those where she has not reached an agreement. A first result is that agreement and protection cannot coexist for a broad class of objectives. That is, if we are given the objectives of a set of participants, it does not exits a contract which protects them all, and at the same time admits an agreement. Roughly, the problem is that, when the offers of the participants mutually depend on their requests, the participant which risks in doing the first step is not protected.

To overcome this negative result, we extend event structures with a new relation of causality $\Vdash$, called *circular causality*. The contract $a \Vdash b$ (intuitively, "I will do a if you *promise* to do b") reaches an agreement with the dual contract $b \Vdash a$, while protecting the participant who offers it. While in classical event structures an action a which causally depends on an action b can only be performed after b; in an event structure with circular causality (CES), the new relation $\Vdash$ allows a to happen before b, under the (legally binding) promise that b will be eventually performed. CES are a conservative extension of ES. We prove that the family of configurations of an ES can be generated by a CES by exploiting only the new circular enabling.

Using this refined model for contracts, we devise a technique that, stating from the participants payoffs, constructs a set of contracts which protect their participants, and still admit an agreement.

We then establish a connection between our contract model and Propositional Contract Logic (PCL [BZ10a]), an extension of intuitionistic logic which allows for circular reasoning. To do that, we first reduce the problem of deciding agreeements in (conflict-free) contracts to provability in PCL. Then we show how to construct winning strategies through a suitable encoding of contracts into PCL formulae.

1.4 Synopsis

This dissertation presents both unpublished material, and some published one. We briefly describe below its organization.

Chapter 2: Basics introduces some of the basic notions and the notations used throughout our subsequent technical development.

Chapter 3: Event structures reviews some basic definitions and results about general event structures (ES). These notions will be used in Chapter 6 to provide the foundations for our model for contracts, and later on in Chapter 7 to study an extension of ES which allows for a form of circular reasoning.

Chapter 4: Propositional Contract Logic summarizes the main definitions and results of Propositional Contract Logic. These will be used in Chapter 9 to develop some relations between our contract model and PCL.

Chapter 5: Contracts: a brief survey presents some related work in the area of contracts.

Chapter 6: Contracts presents our model for contracts, where event structures are used to express obligations. The notions of agreement and protection are formalised in a game-theoretic setting. The main result is that agreement and protection cannot coexist for a wide class of contracts.

Part of this material borrows from [BCZ13].

Chapter 7: Event structures with circular causality extends events structures with a new enabling relation which allows for dealing with circular constraints. This new model (CES) is investigated in a foundational way, by describing its main properties and discussing its relations with standard ES.

Part of this material borrows from [BCPZ12a] and [BCPZ13].

Chapter 8: Reconciling agreement and protection extends the contract model in Chapter 6, by allowing obligations to be expressed as CES, and by accordingly revisiting the notions of agreement and protection. The main result is that agreement and protection can now coexists.

Part of this material borrows from [BCZ13].

Chapter 9: A logical view of contracts show that the notions of agreement and winning strategies in the contract model of Chapter 8 are related to that of provability in PCL. This is obtained through three encodings of CES into PCL, which establish strong equivalences between provability in PCL and configurations, reachable events 9.1, and urgent events.

Part of this material borrows from [BCPZ12b] and [BCGZ13].

Chapter 10: Discussion compares our framework with other related approaches.

Chapter 11: Conclusions contains a summarized view of our work, and proposes some directions for further work.

Part I

Background

Chapter 2

Basics

In this chapter we introduce some basic notation, to be used in the later chapters of this thesis. We assume the reader to be already familiar with the basic definitions of sets, relations, functions, partial orders, and graphs, as well as with the basic theory of formal languages and their abstract syntax.

Sets For a set X and a predicate P, we denote with $\{x \in X \mid P(x)\}$ the set of elements x of X for which $P(x)$ is true. We denote with $\wp(X)$ the set of all subsets of X, i.e. $\wp(X) = \{Y \mid Y \subseteq X\}$, with $\wp_{\mathit{fin}}(X)$ the set of all *finite* subsets of X, and with $\wp_{\mathit{fin}}^{\leq 1}(X)$ the set of all *finite* subsets of X of cardinality at most 1.

Sequences We denote with $\langle e_0\, e_1 \ldots\rangle$ the (possibly infinite) sequence of elements $e_0, e_1, \ldots$. We typically use lowercase greek letters, e.g. $\sigma, \eta, \ldots$ to refer to sequences. For a sequence σ, we write $\overline{\sigma}$ for the set of events in σ. We write σ_i for the subsequence $\langle e_0 \ldots e_{i-1}\rangle$. If $\sigma = \langle e_0 \ldots e_n\rangle$ is finite, we write $\sigma\, e$ for the sequence $\langle e_0 \ldots e_n\, e\rangle$. The empty sequence is denoted by ε. The predicate $DF(\sigma)$ is true whenever $\sigma = \langle e_0 e_1 \ldots\rangle$ has no duplicates, i.e. for all $i \leq n$, $e_i \notin \overline{\sigma_i}$.

For a set S, we denote with S^* the set of finite sequences over S, and with S^ω the set of finite and infinite sequences over S.

Interleaving We denote with $\sigma_0|\sigma_1$ the set of all the sequences obtained by mixing the element of σ_0 with the element of σ_1 and preserving the internal order of the sequences. Formally: for all $\sigma, \sigma', \eta, \eta' \in E^*$, $a, b \in E$, we inductively define operator $|$ as follows:

$$\sigma \mid \eta = \begin{cases} \{\eta\} & \text{if } \sigma = \varepsilon \\ \{\sigma\} & \text{if } \eta = \varepsilon \\ \{\langle a(\sigma' \mid \eta)\rangle, \langle b(\sigma \mid \eta')\rangle\} & \text{if } \sigma = a\sigma' \text{ and } \eta = b\eta' \end{cases}$$

For instance, let $\sigma = \langle a_0 a_1\rangle$ and $\eta = \langle b_0 b_1\rangle$. Then we have:

$$\sigma \mid \eta = \{\langle a_0 a_1 b_0 b_1\rangle, \langle a_0 b_0 a_1 b_1\rangle, \langle a_0 b_0 b_1 a_1\rangle, \langle b_0 a_0 a_1 b_1\rangle, \langle b_0 a_0 b_1 a_1\rangle, \langle b_0 b_1 a_0 a_1\rangle\}$$

Partial functions We denote with $A \rightharpoonup B$ a partial function from A to B. We use the symbol $\perp$ to denote an undefined value.

Transition systems A *labelled transition system* (LTS) is a triple $\langle \Gamma, \Lambda, \rightarrow \rangle$, where Γ is a set of *states*, Λ is a set of *labels*, and $\rightarrow \subseteq \Gamma \times \Lambda \times \Gamma$ is the *transition relation*. We write $\gamma \xrightarrow{a} \gamma'$ when $\langle \gamma, a, \gamma' \rangle \in \rightarrow$, and $\gamma \xrightarrow{\lambda} \gamma'$ when $\lambda = \langle a_0 \ldots a_n \rangle$ and

$$\exists \gamma_0, \ldots, \gamma_{n-1}.\ \ \gamma \xrightarrow{a_0} \gamma_0 \xrightarrow{a_1} \gamma_1 \xrightarrow{a_2} \cdots \xrightarrow{a_{n-1}} \gamma_{n-1} \xrightarrow{a_n} \gamma'$$

Also, we write $\gamma \rightarrow^* \gamma'$ when there exists λ such that $\gamma \xrightarrow{\lambda} \gamma'$.

An LTS is *deterministic* when, for all $\gamma \in \Gamma$ and for all $a \in \lambda$,

$$\gamma \xrightarrow{a} \gamma' \ \ \wedge \ \ \gamma \xrightarrow{a} \gamma'' \ \ \Longrightarrow \ \ \gamma' = \gamma''$$

An *initial labelled transition system* is a tuple $\langle \Gamma, \Lambda, \rightarrow, I \rangle$, where $\langle \Gamma, \Lambda, \rightarrow \rangle$ is an LTS and $I \subseteq \Gamma$ is the set of *initial states*.

A state γ' of an initial LTS is *reachable* from a state γ if and only if

$$\exists \lambda \in \Lambda^*. \quad \gamma \xrightarrow{\lambda} \gamma'$$

A state γ is *reachable* if and only if γ is reachable from an initial state.

Hereafter, when clear from the context we shall just call LTS an initial LTS.

Chapter 3

Event structures

Event structures (ES) are a model for concurrency. introduced in [NPW81]. Since [Win86], ES are considered one of the classical model for concurrency (see [WN95] for an account on the relationships among various classical concurrency models). Notwithstanding the variety of ingredients appeared in the literature, event structures are at least equipped with a relation (written $\vdash$ in [Win86]) modelling *causality*, and another one modelling *non-determinism* (usually rendered in terms of conflicts # or consistency).

Extensions to ES often use other relations to model other kind of dependencies that may arise among various events. We recall here the relation $\nearrow \subseteq E \times E$ of [BCM01] modeling a weak causal dependency among events which can be used to model asymmetric conflicts, or the relation $\Vdash \subseteq \wp^{\leq 1}_{fin}(E) \times E \times \wp_{fin}(E)$ introduced in [BBCP04] and used to model both disjunctive causality and asymmetric conflicts, or the bundle relation $(\wp_{fin}(E) \times E)$ introduced in [Lan93] to take into account disjunctive causality.

Event structures have a rich theory, which we will not review here. In this chapter we only report some basic definitions and results which will be needed in our later technical development.

Assume a denumerable universe of *events*, ranged over by $a, b, e, \ldots$ etc. To represent sets of events which can occur together in a computation we use a binary *conflict* relation $\# \subseteq E \times E$, similarly to [Win88]. Namely, if $a\#b$ then there is no computation which contains both a and b. This is somehow less general than the approach of [Win86], where the *consistent* combinations of events which can occur together are defined through a predicate $Con \subseteq \wp_{fin}(E)$. However, the binary relation is adequate for our purposes in the later chapter of this thesis, where we shall mainly use it to model internal/external choices.

Definition 3.1 (Conflict-free and consistent sets)**.** *For a set of events X, the predicate $CF(X)$ (for "X is* conflict-free*") and the set Con are defined as:*

$$\begin{aligned} CF(X) &\triangleq \forall e, e' \in X : \neg(e\#e') \\ Con &= \{X \subseteq_{fin} E \mid CF(X)\} \end{aligned}$$

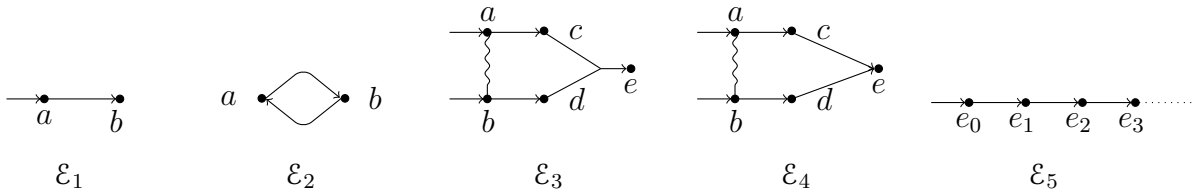


Figure 3.1: Graphical representation of ES.

Definition 3.2 (Event structure)**.** *An event structure is a triple* $\mathcal{E} = \langle E, \#, \vdash \rangle$*, where*

- E *is a set of events,*
- $\# \subseteq E \times E$ *is an irreflexive and symmetric relation, called* conflict *relation,*
- $\vdash \subseteq Con \times E$ *is the* enabling *relation. We assume* $\vdash$ saturated*, i.e.*

$$\forall X \subseteq Y \in Con.\ X \vdash e \implies Y \vdash e$$

We say that $\mathcal{E}$ *is* finite *when* E *is finite; we say that* $\mathcal{E}$ *is* conflict-free *when the conflict relation is empty.*

Notation 3.3 (Shorthands)**.** *With some abuse of notation, we will often write* $e \in \mathcal{E}$ *to mean that* e *is an event of* $\mathcal{E}$*. We adopt the following conventions:*

- $\vdash e$*. Shorthand for* $\emptyset \vdash e$*;*
- $a \vdash b$*. Shorthand for* $\{a\} \vdash b$*.*
- $X \vdash Y$*. For a finite, conflict free set* X*, this means that for* $\forall e \in Y.\ X \vdash e$*. For an infinite, conflict-free* X*,* $X \vdash Y$ *is a shorthand for* $\exists X_0 \subseteq_{fin} X.\ X_0 \vdash Y$*.*

Notation 3.4. *We adopt the following graphical notation for depicting ES: they are denoted as directed hypergraphs, where nodes stand for events. An hyperedge from a set of nodes* X *to node* e *denotes an enabling* $X \vdash e$*. A conflict* $a\#b$ *is represented by a wavy line between* a *and* b*.*

A configuration C is a "snapshot" of the behaviour of the system: a set of events C (possibly infinite) is a configuration whenever for each event $e \in C$ it is possible to find a *trace for* e *in* C, *i.e.* a finite sequence of events containing e, which is closed under the enabling relation.

Definition 3.5 (Configuration)**.** *For an* ES $\mathcal{E} = \langle E, \#, \vdash \rangle$*, we say that* $C \subseteq E$ *is a* configuration of $\mathcal{E}$ *iff* $CF(C)$*, and*

$$\forall e \in C.\ \exists \sigma = \langle e_0, \ldots, e_n \rangle.\ e \in \overline{\sigma} \subseteq C\ \wedge\ \forall i \leq n.\ \overline{\sigma_i} \vdash e_i$$

The set of all configurations of an ES $\mathcal{E}$ *is denoted by* $\mathcal{F}_{\mathcal{E}}$*.*

Example 3.6. *Consider the five ES in Fig. 3.1. We have that:*

- $\mathcal{E}_1$ *has enablings* $\vdash a$ *and* $a \vdash b$*, and we have* $\mathcal{F}_{\mathcal{E}_1} = \{\emptyset, \{a\}, \{a, b\}\}$.
- $\mathcal{E}_2$ *has enablings* $a \vdash b$ *and* $b \vdash a$*, and we have* $\mathcal{F}_{\mathcal{E}_1} = \{\emptyset\}$.
- $\mathcal{E}_3$ *has enablings* $\vdash a$, $\vdash b$, $a \vdash c$, $b \vdash d$, *and* $\{c, d\} \vdash e$*. Since* e *is enabled by both* a *and* b*, graphically we have two arrows joining in the middle, from* a,b *to* e*. The configurations of* $\mathcal{E}_3$ *are* $\emptyset$, $\{a\}$, $\{b\}$, $\{a, c\}$ *and* $\{b, d\}$*. Note that* e *never occurs in a configuration, because the conflict* $a\#b$ *prevent it to happen.*
- $\mathcal{E}_4$ *has enablings* $\vdash a$, $\vdash b$, $a \vdash c$, $b \vdash d$, $c \vdash e$ *and* $d \vdash e$*. The configurations of* $\mathcal{E}_4$ *are* $\emptyset$, $\{a\}$, $\{b\}$, $\{a, c\}$, $\{b, d\}$, $\{a, c, e\}$ *and* $\{b, d, e\}$.
- $\mathcal{E}_5$ *has configurations* $\bigcup_{i<k}\{e_i\}$*, for all* k*. Also, the infinite set* $\{e_0, e_1, \ldots\}$ *is a configuration of* $\mathcal{E}_5$.

The notion of enabling naturally induces an LTS which represents the possible computations of an ES. The states of this LTS represent the events fired so far; a transition $X \xrightarrow{e} Y$ models the event e being fired from state X, and Y is the reached state. We focus on finite configurations, hence the states of the LTS associated to an ES are finite subsets of events. The set of initial states is the set containing the empty configuration.

Definition 3.7 (LTS of an ES)**.** *For all* ES $\mathcal{E}$*, we define the labelled transition system* $\mathrm{LTS}_{\mathcal{E}} = \langle \wp_{fin}(E), E, \rightarrow_{\mathcal{E}} \rangle$ *as follows:*

$$C \xrightarrow{e}_{\mathcal{E}} C \cup \{e\} \qquad \text{iff } C \vdash e,\ e \notin C \text{ and } CF(C \cup \{e\})$$

The relations among $\mathrm{LTS}_{\mathcal{E}}$ and $\mathcal{F}_{\mathcal{E}}$ are illustrated by the two lemmata below.

Lemma 3.8 (LTS and Configurations)**.** *For all* ES $\mathcal{E}$*, and for all* $C \subseteq E$*:*

$$C \in \mathcal{F}_{\mathcal{E}} \iff \forall D \subseteq_{fin} C.\ \exists C_0.\ D \subseteq C_0 \subseteq_{fin} C.\ \emptyset \rightarrow^*_{\mathcal{E}} C_0$$

An obvious consequence of this lemma is that each finite configuration is a reachable state of the associated LTS.

Lemma 3.9. *For all* $C, C' \subseteq_{fin} E$*, and for all* $e \in E$ *such that* $CF(C' \cup \{e\})$*:*

$$C \xrightarrow{e}_{\mathcal{E}} \ \wedge\ C \subseteq C' \ \wedge\ e \notin C' \implies C' \xrightarrow{e}_{\mathcal{E}}$$

We recall now some properties of configurations, as done in [Win86] and [Win88].

Definition 3.10 (Pairwise compatibility)**.** *Let* $\mathcal{F}$ *be a family of sets. We say a subset* $\mathcal{A}$ *of* $\mathcal{F}$ *is* pairwise compatible *if and only if*

$$\forall e, e' \in \bigcup \mathcal{A}.\ \exists C \in \mathcal{F}.\ e, e' \in C$$

Note that the above definition differs from Winskel's (Def. 3.4 in [Win88]). There, a set $\mathcal{A} \subseteq \mathcal{F}$ is pairwise compatible iff for all $C, C' \in \mathcal{A}$, there exists $D \in \mathcal{F}$ such that $C \cup C' \subseteq D$. Clearly, Winskel's pairwise compatibility implies ours, while the converse is not true. For instance, consider the family of sets:

$$\mathcal{A} \;=\; \{\emptyset, \{e_0\}, \{e_1\}, \{e_0, e_1\}, \{e_0, e_2\}, \{e_1, e_2\}\} \tag{3.1}$$

Then, $\mathcal{A}$ is pairwise compatible according to Def. 3.10, while it is not according to Winskel's, because e.g. there exists no $D \in \mathcal{F}$ such that $\{e_0\} \subseteq D$ and $\{e_1, e_2\} \in D$. Clearly, both definitions imply that if $\mathcal{A}$ is pairwise compatible, then $\bigcup \mathcal{A}$ is conflict-free.

We deviate from Winskel's definition in order to make Theorem 3.14 below (Theorem 3.7 in [Win88]) holds also in the case of an event structure defined by conflicts and not by an extensional set of consistency sets.

Definition 3.11 (Families of configurations)**.** *For a set of sets $\mathcal{F}$ we define the following three properties:*

Coherence *If $\mathcal{A}$ is a pairwise compatible subset of $\mathcal{F}$, then $\bigcup \mathcal{A} \in \mathcal{F}$.*

Finiteness

$$\forall C \in \mathcal{F}.\, \forall e \in C.\, \exists C_0 \in \mathcal{F}.\, e \in C_0 \subseteq_{fin} C$$

Coincidence-freeness

$$\forall C \in \mathcal{F}.\, \forall e, e' \in C.\, \big(e \neq e' \implies (\exists C' \in \mathcal{F}.\, C' \subseteq C \;\wedge\; (e \in C' \iff e' \notin C')\big)$$

A set $\mathcal{F}$ is a family of configurations *iff it satisfies coherence, finiteness and coincidence-freeness, and in that case, $\mathcal{F}$ is called a family of configurations of E when $\bigcup \mathcal{F} = E$.*

A basic result of [Win88] is that the set of configurations of an ES forms a family of configurations.

Theorem 3.12. *For all* ES *$\mathcal{E}$, the set $\mathcal{F}_\mathcal{E}$ is a family of configurations.*

Proof. For coherence, let $\mathcal{A} \subseteq \mathcal{F}_\mathcal{E}$ be pairwise compatible in $\mathcal{F}_\mathcal{E}$. Let $e, e' \in \mathcal{A}$. By Def. 3.10, exists $C \in \mathcal{F}_\mathcal{E}$ such that $e, e' \in C$. Since $C \in \mathcal{F}_\mathcal{E}$ then $CF(C)$, hence $\neg(e \# e')$ and so $CF(\bigcup \mathcal{A})$. We now prove $\bigcup \mathcal{A} \in \mathcal{F}_\mathcal{E}$. Let $e \in \bigcup \mathcal{A}$, then there exists $C \in \mathcal{A}$ such that $e \in C$. Since $\mathcal{A} \subseteq \mathcal{F}_\mathcal{E}$, then $C \in \mathcal{F}_\mathcal{E}$ and so by Def. 3.5 there exists a sequence σ of events of C which contains e and is closed under the $\vdash$-enabling. Since $C \subseteq \bigcup \mathcal{A}$, then σ is a sequence for e in $\bigcup \mathcal{A}$ and so by Def. 3.5 we can conclude that $\bigcup A \in \mathcal{F}$.

Finiteness is straightforward by Def. 3.5, since for all $e \in C \in \mathcal{F}$, there exists a finite trace σ closed under the $\vdash$-enabling such that $e \in \overline{\sigma} \subseteq_{fin} C$ is a configuration in $\mathcal{F}$.

Coincidence-freeness is straightforward by Def. 3.5. □

Definition 3.13 ($\mathcal{E}(\mathcal{F})$)**.** *Let $\mathcal{F}$ be a family of configurations of a set E. We define the* ES $\mathcal{E}(\mathcal{F}) = (E, \#, \vdash)$ *as follows:*

(a) $e \# e' \iff \forall C \in \mathcal{F}.\ e \notin C \vee e' \notin C$

(b) $X \vdash e \iff CF(X) \wedge X$ *is finite* $\wedge\ \exists C \in \mathcal{F}.\ e \in C \subseteq X \cup \{e\}$

Our definition of $\mathcal{E}(\mathcal{F})$ differs from Winskel's one in [Win88] in two points.

First, in item (a) we say that e is *not* in conflict with e' iff there exists $C \in \mathcal{F}$ such that $e, e' \in C$, while in [Win88] the condition is that $e \in C \iff e' \in C$ for some $C \in \mathcal{F}$. We argue that the latter definition is not correct, since it implies that no events are in conflict. Indeed, by taking $C = \emptyset$, we have that $e \notin C$ and $e' \notin C$ for all e, e', and so by definition e and e' cannot be in conflict.

Second, our notion of pairwise compatibility differs from Winskel's, as remarked after Def. 3.10. Indeed, Winskel's definition seems to invalidate Theorem 3.14. A counterexample is given by the set $\mathcal{A}$ in (3.1), which is a family of configurations according to Winskel's. By construction, the ES $\mathcal{E}(\mathcal{A})$ has no conflicts and $\{e_0, e_1, e_2\} \in \mathcal{F}_{\mathcal{E}(\mathcal{A})}$ — so contradicting the theorem, because $\{e_0, e_1, e_2\} \notin \mathcal{A}$. Our definition of pairwise compatibility solves this problem, because $\mathcal{A}$ is *not* a family of configuration according to Def. 3.10. Indeed, for being a family of configurations, by coherence $\mathcal{A}$ should also comprise the set $\{e_0, e_1, e_2\}$.

We can then easily prove the following theorem:

Theorem 3.14. *For all families of configurations $\mathcal{F}$, we have $\mathcal{F}_{\mathcal{E}(\mathcal{F})} = \mathcal{F}$.*

Proof. Let $\mathcal{F}$ be a family of configurations. For ($\subseteq$), let $C \in \mathcal{F}_{\mathcal{E}(\mathcal{F})}$. By Def. 3.5 we have $CF(C)$, and that for all $e \in C$ there exists a finite sequence $\sigma^e = \langle e_0 \dots e_n \rangle$ of elements of C which contains e and such that for all $i \leq n$, $\sigma^e_i \vdash e_i$. Hence, by Def. 3.13(b),

$$\forall e_i \in \overline{\sigma^e}.\ \exists D_i \in \mathcal{F}.\ e_i \in D_i \subseteq \overline{\sigma^e_i} \cup \{e_i\}$$

Since $\bigcup\{D_i \mid e_i \in \overline{\sigma^e}\} = \overline{\sigma^e}$, the set $\{D_i \mid e_i \in \overline{\sigma^e}\}$ is pairwise compatible in $\mathcal{F}$, hence by Theorem 3.12 (coherence) we have that $\bigcup\{D_i \mid e_i \in \overline{\sigma^e}\} \in \mathcal{F}$. Again, the set $\{\overline{\sigma^e} \mid e \in C\}$ is pairwise compatible in $\mathcal{F}$, therefore by coherence $C = \bigcup\{\overline{\sigma^e} \mid e \in C\} \in \mathcal{F}$.

For ($\supseteq$), let $C \in \mathcal{F}$. By the definition of conflict in Def. 3.13(a), it must be $CF(C)$. By Theorem 3.12 (finiteness) for all $e \in C$ there exists $C^e \in \mathcal{F}$ such that $e \in C^e \subseteq_{fin} C$. By Theorem 3.12 (coincidence-freeness and coherence) we can find a finite sequence of configurations $C_0 \subset C_1 \dots \subset C_n$ such that $C_0 = \emptyset$ and $C_n k = C^e$ and

$$\forall i \in 1 \dots n.\ \exists e_i.\ C_i \setminus C_{i-1} = \{e_i\}$$

Thus, by Def. 3.13(b) it follows that $C_i \vdash e_i$. Therefore for each $e \in C$, we have found a sequence $\sigma^e = \langle e_0 \dots e_n \rangle$ closed under $\vdash$-enabling such that $e \in \overline{\sigma^e} = C^e \subseteq C$. By Def. 7.5, we conclude that $C \in \mathcal{F}_{\hat{\mathcal{E}}(\mathcal{F})}$. □

Chapter 4

Propositional Contract Logic

In this chapter we review an extension of intuitionistic logic (IPC), called Propositional Contract Logic (PCL [BZ10a]). PCL features a "contractual" form of implication, denoted by $\twoheadrightarrow$. The intuition is that a formula $p \twoheadrightarrow q$ entails q not only when p is provable, like standard intuitionistic implication, but also in the case that a "compatible" formula is assumed. This compatible formula can take different forms, but the archetypal example is the (somewhat dual) $q \twoheadrightarrow p$. While $(p \to q) \wedge (q \to p) \to p \wedge q$ is *not* a theorem of IPC, $(p \twoheadrightarrow q) \wedge (q \twoheadrightarrow p) \to p \wedge q$ is a theorem of PCL. The logic PCL is decidable. Except when stated otherwise, all the results in this chapter have been proved in [BZ10a].

The syntax of PCL extends that of IPC. It includes the standard connectives $\wedge, \vee, \to$, and contractual implication $\twoheadrightarrow$. We assume a denumerable set $a, b, \ldots$ of prime (atomic) formulae. PCL formulae are denoted with the letters $p, q, r, s, \ldots$.

Definition 4.1 (Syntax of PCL)**.** *The formulae of* PCL *are defined as:*

$$p ::= \bot \mid \top \mid a \mid p \vee p \mid p \wedge p \mid p \to p \mid p \twoheadrightarrow p$$

where $\bot$ is $false$ and $\top$ is $true$.

The proof theory of PCL has been studied through three different calculi: an Hilbert-style calculus (Def. 4.2), a Gentzen-style system (Def. 4.5), and a natural deduction system (Def. 4.3). The first two have been introduced and proved equivalent in [BZ10a], while the last has been introduced in [BCGZ13], and therein proved equivalent to the others.

Definition 4.2 (Hilbert-style axiomatisation of PCL)**.** *The Hilbert-style axiomatisation of* PCL *extends that of IPC with the following axioms:*

$$\top \twoheadrightarrow \top$$
$$(p \twoheadrightarrow p) \to p$$
$$(p' \to p) \to (p \twoheadrightarrow q) \to (q \to q') \to (p' \twoheadrightarrow q')$$

$$\frac{}{\Delta, p \vdash p}\ (\text{ID}) \qquad \frac{\Delta \vdash \bot}{\Delta \vdash p}\ (\bot\text{E})$$

$$\frac{\Delta, p \vdash q}{\Delta \vdash p \to q}\ (\to\text{I}) \qquad \frac{\Delta \vdash p \to q \qquad \Delta \vdash p}{q}\ (\to\text{E})$$

$$\frac{\Delta \vdash p \qquad \Delta \vdash q}{\Delta \vdash p \wedge q}\ (\wedge\text{I}) \qquad \frac{\Delta \vdash p \wedge q}{\Delta \vdash p}\ (\wedge\text{E1}) \qquad \frac{\Delta \vdash p \wedge q}{\Delta \vdash q}\ (\wedge\text{E2})$$

$$\frac{\Delta \vdash p}{\Delta \vdash p \vee q}\ (\vee\text{I1}) \qquad \frac{\Delta \vdash q}{\Delta \vdash p \vee q}\ (\vee\text{I2}) \qquad \frac{\Delta \vdash p \vee q \qquad \Delta, p \vdash r \qquad \Delta, q \vdash r}{\Delta \vdash r}\ (\vee\text{E})$$

Figure 4.1: Natural deduction rules for IPC.

$$\frac{\Delta \vdash q}{\Delta \vdash p \twoheadrightarrow q}\ (\twoheadrightarrow\text{I1})$$

$$\frac{\Delta \vdash p' \twoheadrightarrow q' \qquad \Delta, p \vdash p' \qquad \Delta, q' \vdash q}{\Delta \vdash p \twoheadrightarrow q}\ (\twoheadrightarrow\text{I2})$$

$$\frac{\Delta \vdash p \twoheadrightarrow q \qquad \Delta, q \vdash p}{\Delta \vdash q}\ (\twoheadrightarrow\text{E})$$

Figure 4.2: Natural deduction for PCL (rules for $\twoheadrightarrow$).

Definition 4.3 (Natural deduction system of PCL)**.** *The natural deduction system of* PCL *extends that of IPC (Fig. 4.1) with the rules in Fig. 4.2. In all the rules,* Δ *is a* set *of* PCL *formulae, and* Δ, p *is equivalent to* $\Delta \cup \{p\}$*.*

Provable formulae are contractually implied, according to rule ($\twoheadrightarrow$I1). Rule ($\twoheadrightarrow$I2) provides $\twoheadrightarrow$ with the same weakening properties of $\to$. The crucial rule is ($\twoheadrightarrow$E), which allows for the elimination of $\twoheadrightarrow$. Compared to the rule for elimination of $\to$ in IPC, the only difference is that in the context used to deduce the antecedent p, rule ($\twoheadrightarrow$E) also allows for using as hypothesis the consequence q.

Example 4.4. *Let* $\Delta = a \to b, b \twoheadrightarrow a$*. A proof of* $\Delta \vdash a$ *in the natural deduction system is:*

$$\dfrac{\Delta \vdash b \twoheadrightarrow a \quad \dfrac{\Delta \vdash a \to b \quad \Delta, a \vdash a}{\Delta, a \vdash b}(\to\text{E})}{\Delta \vdash a}(\twoheadrightarrow\text{E})$$

$$\frac{}{\Delta, p \vdash p}\ (\textsc{Id}) \qquad \frac{\Delta \vdash p \qquad \Delta,\ p \vdash q}{\Delta \vdash q}\ (\textsc{Cut})$$

$$\frac{\Delta,\ p \wedge q,\ p \vdash r}{\Delta,\ p \wedge q \vdash r}\ (\wedge\text{L1}) \qquad \frac{\Delta,\ p \wedge q,\ q \vdash r}{\Delta,\ p \wedge q \vdash r}\ (\wedge\text{L2}) \qquad \frac{\Delta \vdash p \qquad \Delta \vdash q}{\Delta \vdash p \wedge q}\ (\wedge\text{R})$$

$$\frac{\Delta, p \vee q, p \vdash r \qquad \Delta, p \vee q, q \vdash r}{\Delta, p \vee q \vdash r}\ (\vee\text{L})$$

$$\frac{\Delta \vdash p}{\Delta \vdash p \vee q}\ (\vee\text{R1}) \qquad \frac{\Delta \vdash q}{\Delta \vdash p \vee q}\ (\vee\text{R2})$$

$$\frac{\Delta,\ p \to q \vdash p \qquad \Delta,\ p \to q,\ q \vdash r}{\Delta,\ p \to q \vdash r}\ (\to\text{L}) \qquad \frac{\Delta,\ p \vdash q}{\Delta \vdash p \to q}\ (\to\text{R})$$

$$\frac{}{\Delta, \bot \vdash p}\ (\bot\text{L}) \qquad \frac{}{\Delta \vdash \top}\ (\top\text{R}) \qquad \frac{\Delta \vdash \bot}{\Delta \vdash p}\ (\textsc{WeakR})$$

Figure 4.3: Genzten-style proof system for IPC.

Definition 4.5 (Gentzen-style system of PCL)**.** *The Gentzen-style sequent calculus of* PCL*is defined by the rules in Fig. 4.3 (the IPC calculus) plus those in Fig. 4.4. In all the rules,* Δ *is a* set *of* PCL *formulae. and* Δ, p *is equivalent to* $\Delta \cup \{p\}$*.*

The style of the rules follows that in [Pfe00]. Rule (Fix) is the left rule for $\twoheadrightarrow$. It is almost the same as the left rule for $\to$, except that (in a "circular" fashion) the formula r can be used to deduce p in the first rule premise. Rule (Zero) introduces $\twoheadrightarrow$ on the right of a sequent (similarly e.g. to $\bigcirc R$ of lax logic [FM97]), while rule (PrePost) introduces $\twoheadrightarrow$ on the right, and eliminates it on the left (similarly e.g. to $\bigcirc L$ of [FM97]).

Example 4.6. *Let* $\Delta = a \to b,\ b \twoheadrightarrow a$*. A proof of* $\Delta \vdash a$ *in the Gentzen-style system is:*

$$\dfrac{\dfrac{\dfrac{}{\Delta, a \to b, a \vdash a}\ (\textsc{Id}) \qquad \dfrac{}{\Delta, a, b \vdash b}\ (\textsc{Id})}{\Delta, b \twoheadrightarrow a, a \vdash b}\ (\to\text{L}) \qquad \dfrac{}{\Delta, a \vdash a}\ (\textsc{Id})}{\Delta \vdash a}\ (\textsc{Fix})$$

Theorem 4.7. $\Delta \vdash p$ *is provable in natural deduction iff the sequent* $\Delta \vdash p$ *is provable in the Gentzen-style calculus.*

$$\frac{\Delta \vdash q}{\Delta \vdash p \twoheadrightarrow q}\ (\textsc{Zero})$$

$$\frac{\Delta,\ p \twoheadrightarrow q,\ p' \vdash p \quad \Delta,\ p \twoheadrightarrow q,\ q \vdash q'}{\Delta,\ p \twoheadrightarrow q \vdash p' \twoheadrightarrow q'}\ (\textsc{PrePost})$$

$$\frac{\Delta,\ p \twoheadrightarrow q,\ r \vdash p \quad \Delta,\ p \twoheadrightarrow q,\ q \vdash r}{\Delta,\ p \twoheadrightarrow q \vdash r}\ (\textsc{Fix})$$

Figure 4.4: Genzten-style proof system for PCL (rules for $\twoheadrightarrow$).

Proof. See [BCGZ13]. □

PCL is consistent (see proof in [BZ09a]). Also, negation-free formulae do not lead to inconsistencies.

Theorem 4.8 (Consistency of PCL)**.** PCL *is consistent, i.e.* $\nvdash \bot$*. Also, if* p *is* $\bot$*-free, then* $\nvdash p \to \bot$*.*

The Gentzen-style calculus of PCL enjoys cut elimination (see proof in [BZ09a]). A cut on a formula p is replaced by cuts on strict subformulae of p, and cuts on p having a shorter proof tree.

Theorem 4.9 (Cut Elimination)**.** *If* p *is provable in* PCL*, then there exists a proof of* p *not using the* (Cut) *rule.*

The subformula property holds in PCL (see proof in [BZ09a]). Cut-free proofs only involve subformulae of the sequent at hand.

Theorem 4.10 (Subformula Property)**.** *Let* π *be a cut-free proof of* $\Delta \vdash p$*. Then, the formulae occurring in* π *are subformulae of those occurring in* Δ *and* p*.*

Decidability of PCL it established in [BZ10a] by exploiting cut elimination and the subformula property of Gentzen-style proof system. Indeed, theorems 4.9 and 4.10 allow for exhaustively searching the proof space, so implying decidability.

Theorem 4.11 (Decidability of PCL)**.** *The logic* PCL *is decidable.*

In the later chapters of this dissertation we will mainly consider the Horn fragment of PCL, which comprises atoms, conjunctions, and non-nested intuitionistic/contractual implications.

Definition 4.12 (Horn PCL theory)**.** *A* Horn PCL theory *is a set of clauses of the form* $\alpha \to a$ *or* $\alpha \twoheadrightarrow a$*, where* α *is a possibly empty conjunction of atoms.*

The clause a is a shorthand for $\top \to a$. We denote with $\overline{\alpha}$ the set of atoms in α.

Example 4.13. *Consider the following Horn* PCL *theory* $\Delta_\star$*:*

$$\begin{aligned}\Delta_\star = \{&(e_0 \wedge e_1) \twoheadrightarrow e_6,\ e_6 \to e_3,\ e_6 \to e_4,\ e_3 \to e_0,\\ &(e_4 \wedge e_5) \twoheadrightarrow e_7,\ e_7 \to e_1,\ e_7 \to e_2,\ e_2 \to e_5\}\end{aligned}$$

It is possible to prove that $\Delta_\star \vdash e_i$ *for all* $i \in 0..7$*. Note that, were any one of the* $\twoheadrightarrow$ *in* $\Delta_\star$ *replaced with a* $\to$*, then no atoms would have been provable.*

For proving atoms (or their conjunctions) in Horn PCL theories, a strict subset of the Gentzen rules suffices.

Lemma 4.14. *Let* Δ *be a Horn* PCL *theory. If* $\Delta \vdash \alpha$ *in the Gentzen system, then a proof of* $\Delta \vdash \alpha$ *exists which uses only the rules* (Id)*,* ($\wedge$L1)*,* ($\wedge$L2)*,* ($\wedge$R)*,* ($\to$L)*,* (Fix)*.*

Proof. By induction on the depth of a Gentzen proof π of $\Delta \vdash \alpha$. Since cut elimination holds for PCL, w.l.o.g. we can choose π cut-free. The last rule used in π can be only (Id), ($\wedge$L1), ($\wedge$L2), ($\wedge$R), ($\to$L), or (Fix). The base case (Id) is trivial. For the inductive case, since Δ is a Horn PCL theory then in the premise of the last rule of π we can only have sequents of the for $\Delta' \vdash \alpha'$, where Δ' is a Horn PCL theory and α' is a conjunction of atoms. Then, the induction hypothesis concludes. □

Given a Horn PCL theory Δ, we can partition its clauses in two sets: those which are needed to deduce some atom in Δ, and those which are useless in deducing any atom. For a clause $\alpha \to a$ in Δ of the first kind, we have that both α and a are provable in Δ.

Lemma 4.15. *For a Horn* PCL *theory* Δ*, and for* $\circ \in \{\to, \twoheadrightarrow\}$*:*

$$\Delta, \alpha \circ a \ \vdash\ b \quad\Longrightarrow\quad \Delta \vdash b \ \vee\ \Delta, \alpha \circ a \ \vdash\ \alpha, a, b$$

Proof. See [BTZ12a]. □

Chapter 5

Contracts: a brief survey

The term "contract" is used to denote very different concepts. In general, a contract is considered as a property that a computational entity promises to satisfy, possibly giving evidence of this as a proof or a certificate.

Contracts have been investigated using a wide variety of models, and for a wide variety of different purposes. We identify two main paradigms for representing contracts:

- *contracts-as-processes.* In this paradigm, contracts are modelled as interacting processes. A common choice is to fix the observable events in the execution of the process (e.g. method invocations or message exchanges) and to define a contract as the set of sequences of events that respect the properties we want to model. Typical contract models in this paradigm use and extend, e.g. process algebras, Petri nets, timed automata, concurrent games, *etc.*

- *contracts-as-formulae.* In this paradigm, contracts are specified in some formal logics. The complexity of real-world scenarios, where several concepts like principals, authorizations, delegation, mandates, regulations, etc. are inextricably intermingled, have led to a steady flourishing of new logics over the years. These take inspiration and extend e.g. classical, modal, intuitionistic, deontic, default and defeasible logics. Each logic is designed to represent some particular aspect of contracts, e.g. obligations, permissions and prohibitions in deontic logics, violation of contracts in default and defeasible logics, *etc.*

We devise the following taxonomy of approaches to contracts, where we distinguish the purpose of making use of contracts (rather than focusing on how contracts are modeled). So, the main purposes contracts are used, are:

- detecting when two or more contracts are compliant;

- checking when an implementation conforms to a contract (also, when a contract refines another one);

- monitoring contracts to detect violations and assign liabilities;
- negotiating clauses before agreeing on a contract;
- disciplining the interactions among services (contract-oriented computing).

We briefly discuss below some of the main approaches to contracts, organising them according to the above-mentioned taxonomy.

5.1 Compliance

Contracts are modelled in [CGP09] in a fragment of CCS which includes prefixing (.), external choice ($+$), internal choice ($\oplus$), and recursion. The contract $\alpha.\sigma$ describes a service which will perform the action α and then will continue as σ. The contract $\sigma+\tau$ describes a service which lets the client to choose between σ or τ, but not both. The contract $\sigma \oplus \tau$ describes a service which chooses either to do σ or to do τ, as a consequence of some internal decision which is not seen and it is not predictable by the client. Actions can be of two kinds: input actions (e.g. a) and output actions (e.g. $\bar{a}$); an input synchronizes with the corresponding output (e.g. a with $\bar{a}$). A client contract complies with a service contract if they can interact (possibly forever), until the client chooses to terminates. For instance, a client contract $\bar{a}$ complies with a service contract $a + b$, (written $\bar{a} \dashv a + b$) since they can both synchronize on a. Instead it is not true that $\bar{a} \dashv a \oplus b$ since $a \oplus b$ may internally decide to do b, and so the interaction gets stuck.

The notion of compliance proposed in [CP09] is "boolean": either the client contract is compliant with the service contract, or it is not. An enhancement is proposed in [CP09], where contracts which are not compliant may become such by reordering their actions. This is done through an orchestrator, which acts as an adapter between the client and the service. Other approaches which use process algebras for defining contracts include e.g. [BZ07, CL06, BTZ12a].

On the contracts-as-formulae side, in [PS12] a decidable logic for contracts is presented, which combines deontic logic (to reason about obligations, permissions and prohibitions) with propositional dynamic logic (to reason about what happens after some actions have occurred, like for instance "after the notification he must..."). Conditional obligations, permissions and prohibitions can be of two different kinds. The first kind is represented by $[\sigma]O(\alpha)$, which may be read as "after performing σ, one is obliged to do α". The second kind uses the implication operator: $C \rightarrow O(\alpha)$ is read as "if C holds then one is obliged to perform α". If an enabled obligation is not immediately performed at its time, it is considered as a violation. To avoid most of the classical paradoxes of deontic logic, in [PS12] all the three deontic modalities are primitive, and none of them can be defined in terms of the others (differently from what happens in other deontic logics). The three modalities are applied over

actions instead of over propositions. Every action is considered deterministic according to the fact that in legal contracts, the outcome of an action is deterministic. The logic features some standard action combinators, such as + and . (for choice and sequence); a conflict relation # which represents the fact that two actions cannot be done at the same time, and a concurrency operator $\times$ to model that two actions must be done at the same time. A global clock is used to synchronize actions, and it is assumed that at each time-step all the possible actions must be performed. If an enabled obligation is not performed in time, it is considered a violation. The logic explicitly represents reparations in case of violations. For instance, the obligation modality $O_C(\alpha)$ states two facts: that α is an obligation, and that if α is not performed, then the reparation C must occur. The prohibition modality $F_C(\alpha)$ states that the action α is forbidden; in case the prohibition is violated, then the reparation C is enforced. Permissions have no reparations associated, because they cannot be violated: they can only be exercised.

PCL [BZ10a] can be viewed as a contract model, too. Compliance can be defined in terms of provability of a certain formula, which models the objective of participants. For instance, the contracts $a \twoheadrightarrow b$ and $b \twoheadrightarrow a$ are compliant with the objective $a \wedge b$, since $a \twoheadrightarrow b,\ b \twoheadrightarrow a \vdash a \wedge b$ in the proof system of PCL. Differently from [CGP09], PCL allows for a "multi-level" notion of compliance, encompassing more than two contracts. For instance, consider the contracts-as-formulae:

$$\begin{aligned} \mathsf{A} &: \ (\mathit{pay} \to \mathit{ship}) \ \wedge \ (\mathit{coupon} \to \mathit{gift}) \\ \mathsf{B}_1 &: \ \mathit{pay} \\ \mathsf{B}_2 &: \ \mathit{pay} \wedge \mathit{coupon} \end{aligned}$$

The contract of A is compliant with both the contracts of B_1 and B_2. When coupled with B_2, the contract of A entails both the obligations *ship* and *gift* for A. When coupled with B_1, we obtain a weaker agreement, since only the obligation *ship* is entailed. Although both levels of agreement are possible, in some sense the contract of B_2 yields a tighter Service-Level Agreement than that of B_1.

A finer-grained notion of compliance can be obtained by exploiting information about the past interactions among participants. Such kind of information can be obtained e.g. by reputation systems, where parties rate each other to express satisfaction about their respective behaviour. For instance, a buyer may rate the seller for the quality of the items provided, for the adherence to deadlines, *etc.* This rating guides other participants' decisions when, e.g. they wish to buy some items from the same seller. An additional side effect of this rating system is to disincentive bad behaviour, which has a positive effect on the general quality of the interactions. In [KNS08] a logical framework for reputation systems is proposed, where reputation ratings about participants are built up by analyzing past interactions. Reputation can be used by the parties to decide whether to interact or not. Causality between events is modelled through (prime) event structures. An event may happen only after some specific events have already happened; an event may be in conflict with

some others, so that its happening may exclude the occurrence of others. A judgment about the past behaviour of participants can be obtained by verifying formulae (expressed in a temporal logic with past operators) over the sequences of events observed throughout different running sessions. These formulae can be interpreted as contracts; compliance is intended as the history of past interactions respecting a given formula.

5.2 Conformance and subcontracts

Several papers address the issue of determining whether an implementation conforms to a contract. In some approaches the formalism used to specify the implementation and the contract is the same, and in such cases conformance can be seen as a "subcontract" relation.

A typical scenario for studying conformance checking is that of choreography-based specification of interorganizational processes. A choreography is a *global* specification of the behavior of a process to be projected into a set of *local* views, which specify the behavior expected from each service involved in the whole process. The local views can be interpreted as the service contracts: if the actual implementation of each service respects its contract, then the overall application is guaranteed to behave correctly.

In [vdALM$^+$10] both global and local views are modelled as open Petri nets. An open Petri net is a Petri net with some special places called *interfaces* that are used to interact with other nets. The intended scenario involves some parties agreeing upon a global contract, partitioning it, and then distributing the responsibility of implementing its parts. Each of these parts is called *public view*, while the implemented part is called *private view*. Once all the implemented parts are ready, they are put together to interact. To ensure proper interactions, every private view must conform to the corresponding public view. This guarantees the global correctness of the overall process. Correctness is rendered as *weak termination*,a property which ensures that in the composed net each non-final marking has at least one successor (there are no deadlocks), and that in the composed network each cycle of non-final markings can be left to reach a final marking (there are no livelocks). A private view conforms with its public view if it has the same interface of the public one, and if it has at least all the feasible partners as the public one. A technique is then devised to automatically check conformance. The overall result of [vdALM$^+$10] is a compositional criterion to check weak termination of applications. One starts from a choreography, projects it into a set of local views, and then refines each of them into a service implementation. These services can be verified independently (for refinement), and it is guaranteed that their composition still enjoys weak termination.

Several papers address the problem of defining subcontract relations for various calculi of contracts, e.g. [CGP09, BZ07, CL06, CP09, Pad09, CCLP06, BZ09b]. A common definition of subcontracts is that a contract σ is a subcontract of τ if every

contract which complies with σ also complies with τ. In [CGP09], the property of subcontract is expressed with the relation $\sigma \sqsubseteq \tau$. For instance, $a \oplus b \sqsubseteq a$ means that every client which is ready to do either a or b (on the service discretion) is also able to deal with a service offering only a. The relation $a \sqsubseteq a + b$ tells that every client which is ready to do a can obviously also interact with a service which leaves the client the possibility to choose between a and b. When $\sigma \sqsubseteq \tau$, it is possible to replace a service (with contract σ) with another one with contract τ, without compromising the possibility of interaction with already deployed clients.

5.3 Contract monitoring

A way to check if an interaction between participants satisfies the declared contract is that of analyzing the history of events since the interaction has started. The goal is to detect contract violations and to identify culpable participants.

In [Hen11], contracts are modelled as zero-sum games, which assign *penalties* and *rewards* to participants, with the constraint that a participant must not receive more than what the others pay. The goal of each participants is to maximise its rewards, and to minimise its penalties.

The behaviour of participants is modelled in [HKZ12] as sets of histories, recording sequences of time-stamped actions. Histories are analysed to detect contract breaches and violations. These two notions are kept distinct: a breach is a sanctionable violation. The participant which first breaches the contract is blamed. Contracts specify how to blame participants, by indicating if a trace conforms to a contract, or otherwise by indicating which parties have breached it.

In [RSE08] contracts are represented as timed automata [AD94]. The correctness (or the violation) of an execution is verified by checking its inclusion in the language accepted by the automaton. An execution over an automaton is represented as a *timed word*, i.e. a pair (σ, τ) where σ is a word of the automaton and τ is a timed sequence such that every move of the automaton $\sigma[i]$ happened at the time $\tau[i]$. Some patterns of violations are considered, e.g. latency in a response, and counting the number of acceptable errors/requests in a time window.

A technique to monitor contract executions through timed automata is also proposed in [LPSS11]. While in [RSE08] a contract violation makes the execution to abort, in [LPSS11] it is possible to recover after a violation. A predictive approach is developed to warn agents when entering in a state which could potentially lead to a violation.

[M$^+$11] studies how to assign liability to software components. This is done by analysing log files to detect contract violations. A problem is how to specify and manipulate logs in order to use them as evidence in a court of law. For instance, recording personal information of users may be considered as a privacy violation, hence it may result in invalidating the logs.

[GHM00] proposes an XML encoding for legal contracts. Contract terms and con-

ditions are modelled using *Event-Condition Actions* from active databases [UW97]. A policy is like: *when* a condition takes place, *then* the related action must happen, *unless* the deadline is not already expired, *otherwise* a compensation is provided. Events trigger actions when certain conditions are met. A contract monitor is implemented by storing actions in a SQL databases, and by composing triggers to detect contract violations.

5.4 Negotiation

In competitive settings, participants have their own goals, needs and viewpoints, and they are only concerned about their own benefits or losses. It is then unlikely that participants can stipulate a contract which exactly matches all their original requests. A preliminary phase, called *negotiation*, allows participants to reach an agreement on their respective behaviour, possibly by weakening their original requests or by strengthening their offers.

There are different kinds of negotiation [Kra01], which differs by what participants aim at obtaining. In [PE10] agents negotiate to share resources in a way called *strategic negotiation*. Agents take turn to make offers to others agents. The interaction lasts until a termination condition is met. An agent behaviour is guided by a negotiation strategy, which specifies what to offer in each turn, and what to respond to an offer. An offer may be accepted, rejected, or an agent may exit the negotiation. A way to devise better strategies is searching for resource allocations which are *Pareto optimal*, i.e. the ones where no other allocation is more profitable for an agent without damaging someone else.

Another kind of negotiation is *incentive contracting*, where an agent may try to convince another self-interested agent to do something for it, by promise of reward. A framework for automated incentive contracting is that of Contract Net [DS03]. In this framework a manager agent asks for a task. All the agents which are potentially interested evaluate it, and submit bids. Then, the manager chooses the offering agent which maximizes the manager goals.

In [BM07, BM08, BM11] a process calculus is proposed to model negotiation of Service Level Agreements. Negotiation is subject to constraints (modelled as c-semirings) that clients and services might require. The calculus manipulates constraints through primitives inspired by Concurrent Constraint Programming [Sar93]. and by calculi with name passing, like e.g. the π-calculus [MPW92]. In particular, the calculus proposed in [BM07] allows for telling, asking, checking, and retracting constraints. Only two parties may simultaneously synchronise, and this can happen only provided that they share a pre-agreed name, and that the constraints they have required are consistent.

5.5 Contract-oriented computing

Contract-oriented computing is a software design paradigm which fosters the use of contracts to discipline the interaction between clients and services. The life-cycle of a contract-oriented service is composed of three phases. In the first phase, the participants negotiate the required and offered behaviour. Upon finding an agreement, contracts are stipulated, and the terms of service they prescribe become legally binding. In the third phase, services execute their contracts (or choose not to). At run-time, services may query the contracts they have stipulated, e.g. to know their obligations and what has to be expected by their counterparts. Contract violations are handled automatically by the service infrastructure, which can provide suitable compensations and sanctions.

To the best of our knowledge, the first contract-oriented calculus has been proposed in [BZ10a]. The calculus combines features from concurrent constraints and calculi for multiparty sessions, and it uses the logic PCL as the underlying contract model. Some extensions and enhancements of this calculus have since then been proposed, e.g. [BZ10b, BTZ11, BTZ12b, BTZ12a].

In particular, [BTZ12a] develops CO_2, a contract-oriented calculus which abstracts from the actual contract model. Indeed, CO_2 relies on a few common notions about contracts, e.g. a transition system which models their evolution, and a relation which blames participants in the states where they violate contracts. Differently from the calculus in [BZ10a], CO_2 does not assume a global constraint store: to reduce the gap towards a realistic distributed implementation, it uses more concrete communication primitives, based on sessions.

A distinguished feature of CO_2 is that processes are neither supposed to respect their contracts, nor are they bound to them by any enforcing mechanism. Quite realistically, *dishonest* processes may avoid to perform some actions promised in their contracts. This may happen either intentionally, e.g. a malicious participant which tries to cheat the system, or unintentionally, e.g. an implementation bug (possibly exploited by an attacker). In both cases, the infrastructure can determine which process has caused the violation, and adequately sanction it.

A fundamental problem is how to guarantee that a process will behave honestly, in all possible contexts where it may be executed. If such guarantees can be given, then the process is protected both against bugs, and against (apparently honest) adversaries which try to make it sanctioned. A negative result in [BTZ12b] is that the problem of determining if a process is honest is undecidable. This has been proved for the class of contracts introduced in [CCLP06, CGP09].

In [BSTZ12] a type system has been proposed to safely over-approximate honesty. If a process is typeable, then it is guaranteed to respect the contract it advertises, in all possible contexts.

Part II

A theory of agreements and protection

Chapter 6

Contracts

In this chapter we introduce a model for contracts. Before formalising this notion, we give the overall intuition and fix some terminology.

The commonly accepted meaning of the word "contract" is that of some entity which has been concretised after a process of "agreement", and which has after then become "legally binding". For instance, some standard references define the word "contract" as:

> a written or spoken agreement, especially one concerning employment, sales, or tenancy, that is intended to be enforceable by law.
>
> — *Oxford English Dictionary Online*

> a legal document that states and explains a formal agreement between two different people or groups, or the agreement itself.
>
> — *Cambridge English Dictionary Online*

> an agreement between two or more parties, to perform a specific job or work order, often temporary or of fixed duration and usually governed by a written agreement.
>
> — *Wiktionary*

Our notion of contract slightly departs from this commonly accepted meaning. While we adhere to the principle that contracts are "legally binding", we do not assume that a contract may only exist after all the involved parties have reached an agreement. For instance, in our view a contract may be the (legally binding) statement made by a service through its Service Level Agreement — which indeed is a concrete entity even before any agreement is established.

To further motivate this choice, consider the terminology used in the domain of process algebras. There, both the atomic entities and their compositions are modelled as *processes*. For instance, both

$$\begin{array}{lll} P & = & \bar{a}\langle v\rangle.\ b(x) \qquad \text{(an output of } v \text{ on channel } a \text{ followed by an input on } b\text{)} \\ Q & = & a(y).\ \bar{b}\langle y+1\rangle \quad \text{(an input on } a \text{ followed by an output on } b\text{)} \end{array}$$

are processes, as well as their composition $P \mid Q$.

Now, assume that somehow there is an agreement between contracts P and Q. Then, $P \mid Q$ can be interpreted as a contract according to the standard meaning, hence a contract theory will provide tools (e.g. a model checker) to reason about it.

If we were going to accept the principle that contracts *exist* only after they have been agreed upon, then a process $P \mid Q'$, where e.g. $Q' = a(x).\ \bar{c}\langle x+1\rangle$ (the output is on the "wrong" channel) would not even exist as a contract, and so no contract theory will be able to reason about it. Such a contract theory would be like a process algebra which gives meaning to $P \mid Q$ but not to $P \mid Q'$ — which would be rather inelegant, if not useless at all. Indeed, in process algebras when P is "connected" to a system through the channel a, P must do the output, regardless of which is the component listening at the other end of the channel, and of what its behaviour will be. There is no "agreement" phase which precedes the actual interaction: e.g., P cannot decide to avoid interactions with Q'.

In our theory, we want to be able to reason about contracts before, or even in the absence of, an agreement. This will allow us to understand what happens when a service advertises its contract in an environment populated by malicious adversaries which try to exploit its weaknesses. Section 1.2 outlines our scenario, where untrusted contract brokers may establish sessions among participants, independently of the contract they agreed on. The ability of reasoning about a contract before it is composed with others is useful also when the contract broker is trusted. For instance, one can devise analysis techniques to detect when a contract can substitute for another one [CGP09].

Basic notions In our model, a contract specifies the behaviour promised and expected by a participant or set of participants. Contracts coming from different participants can be composed together. In our view, *agreement* is a property of composed contracts, which — very roughly — ensures an acceptable interaction to each participant in the composition.

Our model builds upon four principal notions:

Events are the atomic actions observed by contracts. For instance, "Alice gives an apple to Bob" is modelled as an event (say, a) in our formalism. We assume that each event is unique, i.e. it cannot occur twice in the same computation. Thus, if Alice has to give two apples to Bob, we assume two distinct events a_0, a_1.

Participants are the entities which advertise contracts, and are legally bound to perform the events they prescribe. We assume that each event is associated with a unique participant. That is, if both Alice and Carol have to give an apple to Bob, we assume two distinct events, a_{A} (for Alice) and a_{C} (for Carol).

Obligations make explicit the causal dependency between the events performed by a participant, and those to be done in return by the others. For instance, Alice's contract clause "I will give an apple to Bob after I have received a banana" induces an obligation for her to do event a after event (say) b has been performed, since she has promised to do it. Event structures are a natural model for obligations, e.g. by interpreting the above clause as the enabling $b \vdash a$.

Objectives express the degree of "satisfaction" of a participant in any play of the contract. Contracts associate each participant to an objective function, which in turn associates each play with a numerical *payoff*. We shall be quite liberal about objectives: actually, we allow for modelling them as arbitrary functions.

Chapter overview The rest of this chapter is organised as follows. In Section 6.1 we present our model for contracts, and illustrate it through some examples; in Section 6.2 we define a notion of agreement; finally, in Section 6.3 we introduce the concept of *protection*, and we show the main result of this chapter, i.e. that agreement and protection cannot coexist for a wide class of contracts.

6.1 An event-based model of contracts

A contract (Def. 6.1) specifies the obligations and the objectives of a set of participants (ranged over by $\mathsf{A}, \mathsf{B}, \ldots$). The atomic entities of a contract are the *events*, which are uniquely associated with participants through a labelling π.

Obligations are modelled as an event structure, and constrained by the enabling relation $\vdash$ of [Win88]. Intuitively, an enabling $X \vdash e$ models the fact that, if all the events in X have happened, then e is an obligation for participant $\pi(e)$. Such obligation may be discharged only by performing e, or by performing any event in conflict with e. For instance, consider an internal choice between two events a and b. This is modelled by an ES with enablings $\vdash a$, $\vdash b$ and conflict $a \# b$. After the choice (say, of a), the obligation b is discharged.

Objectives are modelled as a function Φ, which associates each participant A and each trace of events σ to a *payoff* $\Phi\mathsf{A}\sigma$. We assume a rather coarse notion of payoffs: we only have three possible outcomes which represent, respectively, success (1), failure (-1), and tie (0).

Definition 6.1 (Contract)**.** *A contract* $\mathcal{C}$ *is a 4-tuple* $(\mathcal{E}, \mathcal{A}, \pi, \Phi)$*, where:*

- $\mathcal{E} = \langle E, \#, \vdash \rangle$ *is an event structure;*
- $\mathcal{A}$ *is a set of* participants*;*
- $\pi : E \to \mathcal{A}$ *associates each event with a participant;*
- $\Phi : \mathcal{A} \rightharpoonup E^\omega \to \{-1, 0, 1\}$ *associates each participant and trace with a* payoff.

and where for all $X \vdash e$ *in* $\mathcal{E}$, $\Phi\pi(e) \neq \bot$.

The last row in Def. 6.1 imposes that contracts respect one basic requirement, namely for all $X \vdash e$ in $\mathcal{E}$, we ask that $\Phi\pi(e)$ is defined (as $\bot$ denotes undefined). Note that Φ is a partial function (denoted with the symbol $\rightharpoonup$), hence a contract does not need to define payoffs for all the participants in $\mathcal{A}$: typically, when A advertises her contract, she will not speculate about the objectives of B. This constraint asks that if a contract defines some obligations for A, then A must also declare in $\mathcal{C}$ her payoffs.

6.1.1 Contract plays

To define the semantics of a contract $\mathcal{C} = \langle \mathcal{E}, \mathcal{A}, \pi, \Phi \rangle$, we interpret it as a nonzero-sum concurrent multi-player game. The game involves the players in $\mathcal{A}$ concurrently performing events in order to reach the objectives defined by Φ.

A *play* of $\mathcal{C}$ is a (finite or infinite) sequence of events of $\mathcal{E}$. We postulate that only the obliged events are permitted. More precisely, the permitted moves after a (finite) sequence of steps σ are exactly the events enabled by $\mathcal{E}$ in $\overline{\sigma}$, i.e. e is permitted in σ iff $\overline{\sigma} \xrightarrow{e}_{\mathcal{E}}$ in $\mathrm{LTS}_{\mathcal{E}}$.

Definition 6.2 (Play)**.** *A* play *of a contract* $\mathcal{C} = \langle \mathcal{E}, \mathcal{A}, \pi, \Phi \rangle$ *is a (finite or infinite) sequence* σ *of events of* $\mathcal{E}$ *such that* $\emptyset \xrightarrow{\sigma}_{\mathcal{E}}$.

The following lemma establishes the obvious relation among plays and the configurations of the ES of the contract.

Lemma 6.3. *The events* $\overline{\sigma}$ *of a play* σ *form a configuration of* $\mathcal{E}$.

Proof. Trivial by Lemma 3.8. □

Each participant can choose a strategy to decide which of her events has to be done in each state. A strategy can prescribe to only perform the events that are enabled by the already occurred ones.

Definition 6.4 (Strategy)**.** *A* strategy Σ *for* A *is a function which associates with each finite play* $\sigma = \langle e_0 \cdots e_n \rangle$ *a set of events of* A *(possibly empty), such that if* $e \in \Sigma(\sigma)$ *then* σe *is still a play.*

When a participant acts as suggested by the strategy, the resulting play is said to be *conform* to that strategy.

Definition 6.5 (Conformance to a strategy)**.** *A play* σ conforms *to a strategy* Σ *for* A *if for all* $i \geq 0$*, if* $e_i \in \pi^{-1}(\mathsf{A})$*, then* $e_i \in \Sigma(\sigma_i)$.

6.1.2 Some examples

Example 6.6. *Suppose there are two kids who want to play together. Alice has a toy airplane, while Bob has a bike. Both kids are willing to share their toys, but they do not trust each other. Thus, before starting to play they advertise the following contracts. Alice will lend her airplane only* after *Bob has allowed her ride his bike. Bob will lend his bike unconditionally. We model the events "Alice lends her airplane" and "Bob lends his bike" as a and b, respectively. The obligations of Alice and Bob are modelled by the following ES:*

$$\mathcal{E}_{\mathsf{A}} : \ b \vdash a \qquad\qquad \mathcal{E}_{\mathsf{B}} : \ \vdash b$$

The objectives of the two kids are modelled by the functions Φ_{A} (which establishes Alice's payoff) and Φ_{B} (for Bob). Alice has a positive payoff in those traces where b has been performed, while she has a negative payoff when she performs a while not obtaining b in return. The payoffs of Bob are dual. Formally:

$$\Phi_{\mathsf{A}}\mathsf{A} \ = \ \lambda\sigma. \begin{cases} 1 & \text{if } b \in \overline{\sigma} \\ 0 & \text{if } a, b \notin \overline{\sigma} \\ -1 & \text{otherwise} \end{cases} \qquad \Phi_{\mathsf{B}}\mathsf{B} \ = \ \lambda\sigma. \begin{cases} 1 & \text{if } a \in \overline{\sigma} \\ 0 & \text{if } b, a \notin \overline{\sigma} \\ -1 & \text{otherwise} \end{cases}$$

Summing up, the contracts of Alice and Bob are $\mathcal{C}_{\mathsf{A}} = \langle \mathcal{E}_{\mathsf{A}}, \mathcal{A}, \pi, \Phi_{\mathsf{A}} \rangle$ and $\mathcal{C}_{\mathsf{B}} = \langle \mathcal{E}_{\mathsf{B}}, \mathcal{A}, \pi, \Phi_{\mathsf{B}} \rangle$, respectively, where $\mathcal{A} = \{\mathsf{A}, \mathsf{B}\}$, $\pi(a) = \mathsf{A}$, and $\pi(b) = \mathsf{B}$.

Intuitively, the contract resulting by the composition of $\mathcal{C}_{\mathsf{A}}$ and $\mathcal{C}_{\mathsf{B}}$ admits an agreement. We shall state this formally later on in Section 6.2. □

Example 6.7. *Suppose Bob lends his bike to Alice (event b), and requires Alices's toy airplane in exchange. However, Alice does not promise to lend her airplane immediately. She can either lend it in the same day (event a_0), or one day after (event a_1), or two days after (a_2),* etc. *If Alice decides not to lend the airplane at day n, she fires the event $\widetilde{a_n} \# a_n$. The obligations of Alice and Bob are modelled by the following ES:*

$$\begin{array}{l} \mathcal{E}_{\mathsf{A}} : \ b \vdash a_0, \ \ b \vdash \widetilde{a_0} \qquad \{\widetilde{a_i} \vdash a_{i+1}, \ \widetilde{a_i} \vdash \widetilde{a_{i+1}} \mid i \geq 0\} \qquad \{\widetilde{a_i} \# a_i \mid i \geq 0\} \\ \mathcal{E}_{\mathsf{B}} : \ \vdash b \end{array}$$

The overall obligations of Alice and Bob are represented in Fig. 6.1, and their payoffs are given by:

$$\Phi\mathsf{A}\sigma = \begin{cases} 1 & \text{if } b \in \overline{\sigma} \\ 0 & \text{if } b \notin \overline{\sigma} \text{ and } \overline{\sigma} \cap A = \emptyset \\ -1 & \text{otherwise} \end{cases} \qquad \Phi\mathsf{B}\sigma = \begin{cases} 1 & \text{if } \overline{\sigma} \cap A \neq \emptyset \\ 0 & \text{if } b \notin \overline{\sigma} \text{ and } \overline{\sigma} \cap A = \emptyset \\ -1 & \text{otherwise} \end{cases}$$

where $A = \{a_i \mid i \geq 0\}$. Intuitively, A agrees on the composed contract, while B does not: indeed, A can indefinitely delay the lending of her airplane. □

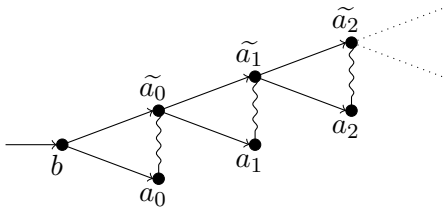


Figure 6.1: A contract with an indefinitely delayed obligation.

Example 6.8. *In [CGP09] contracts are modelled in a variant of CCS which includes prefixing, internal/external choice, and recursion. Consider e.g. a server* A *which repeatedly offers to her clients a choice between two actions* a *and* b*. The client* B *internally chooses one of his (co-)actions* $\overline{a}$ *and* $\overline{b}$*. This is modelled in [CGP09] as follows:*

$$c_{\mathsf{A}} = \mathit{rec}\ X.\ (a.X + b.X) \qquad\qquad c_{\mathsf{B}} = \mathit{rec}\ Y.\ (\overline{a}.Y \oplus \overline{b}.Y)$$

In our theory we model c_{A} *and* c_{B} *as the contracts* $\mathcal{C}_{\mathsf{A}}$ *and* $\mathcal{C}_{\mathsf{B}}$*, defined below. For all* $i \geq 0$*, let* a_i, b_i *be events of* A*, and let* $\overline{a_i}, \overline{b_i}$ *be events of* B*. The event structures of* A *and* B *are shown in Fig. 6.2 and have the following enablings and conflicts, for all* $i \geq 0$*:*

$$\mathcal{E}_{\mathsf{A}}: \ \ \overline{a_i} \vdash a_i, \ \ \overline{b_i} \vdash b_i, \ \ a_i \# b_i$$
$$\mathcal{E}_{\mathsf{B}}: \ \ \vdash \overline{a_0}, \ \ \vdash \overline{b_0}, \ \ a_i \vdash \overline{a_{i+1}}, \ \ a_i \vdash \overline{b_{i+1}}, \ \ b_i \vdash \overline{a_{i+1}}, \ \ b_i \vdash \overline{b_{i+1}}, \ \ \overline{a_i} \# \overline{b_i}$$

The payoff of a participant $P \in \{\mathsf{A}, \mathsf{B}\}$ *is positive in a* finite *play* σ *if* P *has no obligations in* σ*.*

$$\Phi^{\mathit{fin}} P\sigma \ = \ \begin{cases} 1 & \text{if } \nexists e \in \pi^{-1}(P).\ \overline{\sigma} \xrightarrow{e}_{\mathcal{E}_P} \\ -1 & \text{otherwise} \end{cases} \qquad (\sigma \text{ finite})$$

For an infinite *play, to have a positive payoff* P *require that, from any step* i *in the play, there exists a future step* $j \geq i$ *where* P *has no obligations. Otherwise, if* P *is eventually definitely "culpable" (i.e. if* $\exists i.\ \forall j \geq i.\ \Phi^{\mathit{fin}} P\sigma_j < 0$*) then her payoff is negative.*

$$\Phi P\sigma \ = \ \begin{cases} 1 & \text{if } \forall i.\ \exists j \geq i.\ \Phi^{\mathit{fin}} P\sigma_j > 0 \\ -1 & \text{otherwise} \end{cases}$$

Intuitively, the composed contract admits an agreement; we shall prove this formally in Section 6.2. □

6.1.3 Payoff functions

The definition of payoff functions in the definition of contract Def. 6.1 is quite liberal. Indeed, it also allows for uncomputable functions, which are of little use in

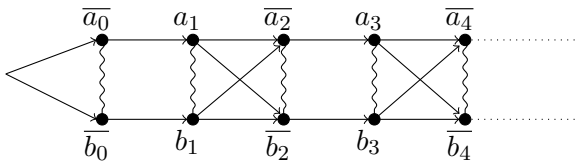


Figure 6.2: Obligations for two CCS-like contracts.

doing anything with a contract. One may then be interested in considering relevant subclasses of payoff functions, in the same spirit of the rich classification of winning conditions in game theory [CH12]. Here we shall only focus on a few subclasses of payoff functions, which are relevant in the realm of contracts.

Büchi payoffs A function Φ is a *Büchi payoff* for A if it says that A succeeds in all infinite traces where she visits a success state infinitely often, i.e. $\Phi\mathsf{A}\sigma_i = 1$ for an infinite number of finite prefixes σ_i of σ.

Definition 6.9 (Büchi payoff)**.** *We say that* Φ *is a* Büchi payoff *for* A *if for all infinite plays* σ*:*

$$\Phi\mathsf{A}\sigma \;=\; \begin{cases} 1 & \text{if } \forall i.\ \exists j \geq i.\ \Phi\mathsf{A}\sigma_j > 0 \\ -1 & \text{if } \exists i.\ \forall j \geq i.\ \Phi\mathsf{A}\sigma_j < 0 \\ 0 & \text{otherwise} \end{cases}$$

For instance, the payoffs in Example 6.8 are Büchi payoffs for A and B.

Reachability payoffs We now consider the class of payoffs which neglect the order in which events are performed.

Definition 6.10 (Reachability payoff)**.** *A function* Φ *is a* reachability payoff *for* A *if* $\Phi\mathsf{A}\sigma = \Phi\mathsf{A}\eta$ *whenever* $\overline{\sigma} = \overline{\eta}$*.*

Alternatively, Φ is a reachability payoff for A when there exist two predicates $\varphi^0, \varphi^1 \subseteq \wp(E)$ such that

$$\Phi\mathsf{A}\sigma = \begin{cases} 1 & \text{if } \overline{\sigma} \in \varphi^1 \\ 0 & \text{if } \overline{\sigma} \in \varphi^0 \\ -1 & \text{otherwise} \end{cases}$$

For instance, the payoffs in Examples 6.6 and 6.7 are reachability payoffs.

Offer-Request payoffs We now introduce another class of payoff functions, called *Offer-Request* payoffs. Intuitively, these are used by participants which want to be paid for each provided service. Each participant A has a set $\{O^0_{\mathsf{A}}, O^1_{\mathsf{A}}, \ldots\}$ of sets of events (the *offers*), and a corresponding set $\{R^0_{\mathsf{A}}, R^1_{\mathsf{A}}, \ldots\}$ (the *requests*). To be successful, whenever A performs in a play some offer O^i_{A} (in whatever order), the

play must also contain the corresponding request R^i_{A}, and at least one of the requests has to be fulfilled. Clearly, Offer-Request payoffs are also reachability payoffs.

Definition 6.11 (Offer-Request payoff)**.** *Let $\pi : E \to \mathcal{A}$. We say that Φ is an* Offer-Request payoff for A *iff there exist a (possibly infinite) sequence $(O^i, R^i)_i$ such that for all i, $O^i \subseteq \pi^{-1}(\mathsf{A})$, $\emptyset \neq R^i \subseteq E \setminus \pi^{-1}(\mathsf{A})$, and for all σ:*

$$\Phi\mathsf{A}\sigma \;=\; \begin{cases} 1 & \text{if } (\exists i.\ R^i \subseteq \overline{\sigma}) \ \wedge\ (\forall j.\ O^j \subseteq \overline{\sigma} \implies R^j \subseteq \overline{\sigma}) \\ 0 & \text{if } (\forall i.\ R^i \not\subseteq \overline{\sigma} \ \wedge\ O^i \not\subseteq \overline{\sigma}) \\ -1 & \text{otherwise} \end{cases}$$

A contract $\mathcal{C} = \langle \mathcal{E}, \mathcal{A}, \pi, \Phi \rangle$ has O-R payoffs iff Φ is an O-R payoff for all $\mathsf{A} \in \mathcal{A}$. If, additionally, all the sets O^i (resp. R^i) are finite for all $\mathsf{A} \in \mathcal{A}$, we say that $\mathcal{C}$ has finite offers *(resp.* finite requests*). If Φ has a finite number of finite offers-request, then Φ is finite.*

For instance, the payoff functions Φ_{A} and Φ_{B} in Example 6.6 are O-R payoffs for A and B. The offers and the requests of A and B are, respectively $O^0_{\mathsf{A}} = \{a\} = R^0_{\mathsf{B}}$ and, dually, $O^0_{\mathsf{B}} = \{b\} = R^0_{\mathsf{A}}$. Instead, the payoff of B in Example 6.7 is *not* an O-R payoff: indeed, the offer b must be followed by (at least) one of the events a_i.

Some remarks about O-R payoffs follow.

- A play σ has a negative payoff for a participant A if A has already done what she offered ($O^i \subseteq \overline{\sigma}$) and she has not received what she wanted ($R^i \not\subseteq \overline{\sigma}$).

- If the O-R payoff for A offers nothing for a non-empty set of requests, e.g.:

$$O^0_{\mathsf{A}} = \emptyset \qquad\qquad R^0_{\mathsf{A}} \neq \emptyset$$

  then in the play ε where no events have been performed, A has a negative payoff. Indeed, $O^0_{\mathsf{A}} = \emptyset \subseteq \overline{\varepsilon}$ but $R^0_{\mathsf{A}} \not\subseteq \overline{\varepsilon}$.

- Specifying the same offer set towards differents request sets (for instance $(\{a\}, \{b\})$,$(\{a\}, \{c\})$) is equivalent to specifying only the single clause $(\{a\}, \{b, c\})$, as the plays with positive/negative playoff are the same.

The following example shows that *not every* O-R payoff is also a Büchi payoff.

Example 6.12. *Let $\mathcal{C}_{\mathsf{A}} = \langle \mathcal{E}_{\mathsf{A}}, \mathcal{A}, \pi, \Phi_{\mathsf{A}} \rangle$ and $\mathcal{C}_{\mathsf{B}} = \langle \mathcal{E}_{\mathsf{B}}, \mathcal{A}, \pi, \Phi_{\mathsf{B}} \rangle$ be two contracts, and let Φ be the (infinite) O-R payoff for A defined by:*

$$(O^i, R^i) = (\{o^i\}, \{r^i\}) \qquad\qquad \forall i \geq 1$$

Let $\mathcal{A} = \{\mathsf{A}, \mathsf{B}\}$, and let $\pi(o^i) = \mathsf{A}$, and $\pi(r^i) = \mathsf{B}$, for all $i \geq 1$.

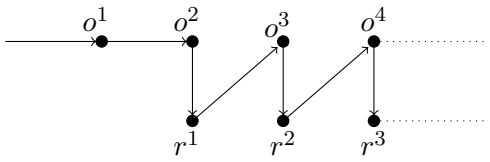


Figure 6.3: An Offer-Request payoff which is not a Büchi payoff.

Let the ES *of* A *and* B *be defined by the following enablings:*

$$\begin{array}{lll} \mathcal{E}_{\mathsf{A}}: & \vdash o^1,\ \ o^1 \vdash o^2,\ \ r^i \vdash o^{i+2} & \forall i \geq 1 \\ \mathcal{E}_{\mathsf{B}}: & o^i \vdash r^{i-1} & \forall i \geq 2 \end{array}$$

The only play σ *where* $\Phi\mathsf{A}\sigma = 1$ *is the infinite one:*

$$\sigma \ = \ \langle o^1\ o^2\ r^1\ o^3\ r^2\ o^4\ r^3 \ldots \rangle$$

Since, for all i*, the play up to* r^i *contains a non-matched* o^{i+1}*, it is not possible to find an index* i *where* $\Phi\mathsf{A}\sigma_i > 0$*. Therefore* $\Phi\mathsf{A}$ *is not a Büchi payoff.* □

Circular Offer-Request payoffs We now consider a relevant subclass of Offer-Request payoffs, where the requests of all participants mutually depend on their offers. An O-R payoff is *circular* when it is not possible to satisfy requests from all participants without each participant doing some offer (item (6.1)), and each combination of the requests is covered by a set of offers (item (6.2)).

For instance, the payoffs of Alice and Bob in Example 6.6 are circular, because their requests (e.g. a and b, respectively) match exactly their offers.

Definition 6.13 (Circular Offer-Request payoff)**.** *An O-R payoff* Φ *for participants* $\mathcal{A}$ *is* circular *when:*

$$\forall \mathcal{J} : \mathcal{A} \to \mathbb{N}.\ \exists \mathcal{L} : \mathcal{A} \to \mathbb{N}.\ \bigcup_{\mathsf{A} \in \mathcal{A}} O_{\mathsf{A}}^{\mathcal{L}\mathsf{A}} \ \subseteq \ \bigcup_{\mathsf{A} \in \mathcal{A}} R_{\mathsf{A}}^{\mathcal{J}\mathsf{A}} \tag{6.1}$$

$$\forall \mathcal{J} : \mathcal{A} \to \mathbb{N}.\ \exists \mathcal{L} : \mathcal{A} \to \mathbb{N}.\ \bigcup_{\mathsf{A} \in \mathcal{A}} O_{\mathsf{A}}^{\mathcal{L}\mathsf{A}} \ \supseteq \ \bigcup_{\mathsf{A} \in \mathcal{A}} R_{\mathsf{A}}^{\mathcal{J}\mathsf{A}} \tag{6.2}$$

We anticipate here a key feature of finite circular payoffs, which will be proved later on in Lemma 6.47. In each play where all participants "win", at some point there exists a participant A which has performed all the offers in O_{A}^i before having obtained all the requests in R_{A}^i.

Example 6.14. *Consider two participants* A, B *with the following O-R payoff:*

i	O_{A}^i	R_{A}^i
0	$\{a_0\}$	$\{b_0\}$
1	$\{a_0, a_1\}$	$\{b_0, b_1\}$
2	$\{a_0, a_1, a_2\}$	$\{b_0, b_1, b_2\}$

i	O_{B}^i	R_{B}^i
0	$\{b_0\}$	$\{a_0\}$
1	$\{b_1, b_2\}$	$\{a_0, a_1\}$
2	$\{b_0, b_1, b_2\}$	$\{a_0, a_2\}$

There are 3^2 *possible choices for the function* $\mathcal{J} : \mathcal{A} \to \{0, 1, 2\}$*. For each of these choices, we have that:*

$$\{a_0, b_0\} \;\subseteq\; \bigcup_{\mathsf{A} \in \mathcal{A}} R_{\mathsf{A}}^{\mathcal{J}\mathsf{A}} \;\subseteq\; \{a_0, a_1, a_2, b_0, b_1, b_2\}$$

Therefore, we can satisfy (6.1) *by choosing* $\mathcal{L} = \{\mathsf{A} \mapsto 0, \mathsf{B} \mapsto 0\}$*, and* (6.2) *by choosing* $\mathcal{L} = \{\mathsf{A} \mapsto 2, \mathsf{B} \mapsto 2\}$*. By Def. 6.13, we conclude that the payoffs of* A *and* B *are circular.*

Note that in any play where A *and* B *have a positive payoff, there is a prefix of the play where one of the participants has performed all her offers, but has not received the corresponding requests. For instance, for the play* $\sigma = \langle a_0\, b_0 \rangle$*,* A *has done all her offers* O_{A}^0 *in the prefix* $\langle a_0 \rangle$*, but there she has not already received* R_{A}^0*.*

If we remove the clause $(O_{\mathsf{A}}^0, R_{\mathsf{A}}^0)$*, then the payoff is no longer circular. In this case, we find a play* $\eta = \langle a_0\, b_0\, b_1 \rangle$ *where both participants have a positive payoff (because* $R_{\mathsf{A}}^1 \cup R_{\mathsf{B}}^0 \subseteq \overline{\eta}$*), but there exists no prefix of* η *where one of the participants has performed all her offers before receiving the corresponding requests.* □

Example 6.15 (Dining retailers [BZ10a])**.** *Around a table,* n *cutlery retailers are about to have dinner. At the center of the table, there is a large dish of food. Despite the food being delicious, the retailers cannot start eating right now. To do that, and follow the proper etiquette, each retailer needs a complete cutlery set, consisting of* n *pieces of different kinds. Each of the* n *retailers owns a distinct set of* n *pieces of cutlery, all of the same kind. The retailers start discussing about trading their cutlery, so that they can finally eat.*

We formalise the retailers payoffs as follows. Each retailer A_i *initially owns* n *pieces of kind* i*. For all* $j \neq i$*, the event* $e_{i,j}$ *models* A_i *giving a piece of cutlery to retailer* A_j*. Thus,* $\pi^{-1}(\mathsf{A}_i) = \{e_{i,j} \mid j \neq i\}$*. Retailer* A_i *offers* $n - 1$ *pieces of his cutlery of kind* i *in exchange for* $n - 1$ *pieces of cutlery of the other kinds.*

$$O_i = \{e_{i,j} \mid j \neq i\} \qquad\qquad R_i = \{e_{j,i} \mid j \neq i\}$$

By Def. 6.13, the payoff Φ_i *of each retailer is a finite O-R circular payoff. Indeed:*

$$\bigcup_{i \in 1..n} O_i \;=\; \{e_{i,j} \mid i \neq j\} \;=\; \bigcup_{i \in 1..n} R_i$$ □

6.1.4 Contract composition

Given two contracts $\mathcal{C}, \mathcal{C}'$, we denote with $\mathcal{C} \mid \mathcal{C}'$ their composition. We assume that whenever two contracts are composed, they both agree about events' names: same name mean same event.

If $\mathcal{C}'$ is the contract written by an adversary of $\mathcal{C}$, then a naïve composition of the two contracts could easily lead to an attack, e.g. when Mallory's contract says that Alice is obliged to give him her airplane. To prevent from such kinds of attacks,

contract composition is a partial operation: for a contract composition of $\mathcal{C}$ and $\mathcal{C}'$ to be defined, we require the following two conditions:

1. the contracts agree on the association between events and participants;
2. if one of the contracts defines a payoff for participant A, then the other contract cannot define payoffs for A.

The conditions which allow two contracts $\mathcal{C}$, $\mathcal{C}'$ to be composed are formalised in Def. 6.16 below.

Definition 6.16 (Composition of compatible contracts)**.** *Two contracts* $\mathcal{C} = \langle \mathcal{E}, \mathcal{A}, \pi, \Phi \rangle$ *and* $\mathcal{C}' = \langle \mathcal{E}', \mathcal{A}', \pi', \Phi' \rangle$ *are* compatible *whenever:*

$$\forall e, e' \in \mathcal{E} \cap \mathcal{E}'.\ e = e' \implies \pi(e) = \pi'(e) \tag{6.3}$$

$$\forall \mathsf{A} \in \mathcal{A} \cup \mathcal{A}'.\ \Phi(\mathsf{A}) = \bot \ \vee \ \Phi'(\mathsf{A}) = \bot \tag{6.4}$$

If $\mathcal{C}$, $\mathcal{C}'$ *are compatible, we define their composition as:*

$$\mathcal{C} \mid \mathcal{C}' = \langle \mathcal{E} \sqcup \mathcal{E}', \mathcal{A} \cup \mathcal{A}', \pi \cup \pi', \Phi \cup \Phi' \rangle$$

where $\mathcal{E} \sqcup \mathcal{E}' = (E \cup E', \vdash \cup \vdash', \# \cup \#')$.

Two contracts which both assign obligations to A are not compatible.

Lemma 6.17. *If* $\mathcal{C} = \langle \mathcal{E}, \mathcal{A}, \pi, \Phi \rangle$ *and* $\mathcal{C}' = \langle \mathcal{E}', \mathcal{A}', \pi', \Phi' \rangle$ *are compatible, then for all* e, e', X, X'*, we have:*

$$X \vdash e \in \mathcal{E} \ \wedge \ X' \vdash e' \in \mathcal{E}' \implies \pi(e) \neq \pi'(e') \ \wedge \ e \neq e'$$

Proof. Let $X \vdash e \in \mathcal{E}$ and $X' \vdash e' \in \mathcal{E}'$.

We first prove that e and e' belong to different participants. Let us assume by contradiction that $\pi(e) = \pi'(e') = \mathsf{A}$. By Def. 6.1 we have that $\Phi\mathsf{A} \neq \bot$ and $\Phi'\mathsf{A} \neq \bot$, which contradicts condition (6.4) in Def. 6.16.

Let us now assume by contradiction that $e = e'$. By condition (6.3), we have that $\pi(e) = \pi'(e')$, but this leads to a contradiction, as shown above. □

Example 6.18. *The contracts* $\mathcal{C}_\mathsf{A}$ *and* $\mathcal{C}_\mathsf{B}$ *in Example 6.6 are compatible, and their composition is the contract* $\mathcal{C} = \mathcal{C}_\mathsf{A} \mid \mathcal{C}_\mathsf{B} = \langle \mathcal{E}, \mathcal{A}, \pi, \Phi \rangle$ *defined as follows:*

$$\begin{aligned} \mathcal{E} &: \ \vdash b, \ \ b \vdash a \\ \mathcal{A} &: \ \{\mathsf{A}, \mathsf{B}\} \\ \pi &: \ \ \{a \mapsto \mathsf{A}, \ \ b \mapsto \mathsf{B}\} \end{aligned} \qquad \Phi P = \begin{cases} \Phi_\mathsf{A}\mathsf{A} & \text{if } P = \mathsf{A} \\ \Phi_\mathsf{B}\mathsf{B} & \text{if } P = \mathsf{B} \end{cases}$$

6.2 Agreements

A crucial notion on contracts is that of *agreement*. Intuitively, when Alice agrees on a contract $\mathcal{C}$, then she can safely initiate an interaction with the other participants, and be guaranteed that the interaction will not "go wrong" — even in the presence of attackers. This does not mean that Alice will always succeed in all interactions: in case Bob is dishonest, we do not assume that an external authority (e.g. Bob's mother) will lend the bike to Alice. We intend that Alice agrees on a contract where, in all the interactions where she does not succeed, then some other participant must be found dishonest. That is, we consider Alice satisfied if she can blame another participant. In real-world applications, a judge may provide compensations to Alice, or impose a punishment to the participant who has violated the contract. Here, we shall not explicitly model the judge, and we only focus on how to formalise the agreement property.

6.2.1 Basic definitions

Recall from Def. 6.2 that we interpret a contract as a multi-player game, where participants concurrently perform events in order to reach their objectives. The plays of this game are the conflict-free sequences of events, with the further requirement that an event e can be fired in a play σ only if e is obliged (i.e., enabled) in σ. The behaviour of each participant A is specified by a strategy Σ_{A}, defining which events of A can be done at each state of a play.

As usual in concurrency, we shall only consider those *fair* plays where an event infinitely often enabled is eventually performed. Indeed, contracts would make little sense in the presence of unfair plays, because an honest participant willing to perform a promised action could be perpetually prevented (by an unfair scheduler) from keeping her promise.

Technically, we define fairness with respect to the strategy of a participant. A play is fair for a strategy Σ (say, of A) when the other participants cannot prevent A from doing some action persistently promised by Σ.

Definition 6.19 (Fair play)**.** *We say that a play $\sigma = \langle e_0\, e_1 \cdots \rangle$ is* fair *for Σ iff:*

$$\forall i \leq |\sigma|.\ \forall e.\ \big(\forall j : i \leq j \leq |\sigma|.\ e \in \Sigma(\sigma_j)\big) \implies \exists h.\ i \leq h < |\sigma|.\ e_h = e$$

Lemma 6.20. *A play $\sigma = \langle e_0\, e_1 \cdots \rangle$ is* fair *for Σ iff:*

$$\forall i \leq |\sigma|.\ \nexists e.\ \forall j : i \leq j \leq |\sigma|.\ e \in \Sigma(\sigma_j)$$

Proof. For a play $\sigma = \langle e_0\, e_1 \cdots \rangle$ let the predicates $P(e,i)$ and $Q(e,i)$ be defined as:

$$\begin{aligned} P(e,i) &\triangleq \forall j : i \leq j \leq |\sigma|.\ e \in \Sigma(\sigma_j) \\ Q(e,i) &\triangleq \exists h \geq i.\ e_h = e \end{aligned}$$

Then, Def. 6.19 can be rewritten as: $\forall i \leq |\sigma|.\, \forall e.\, P(e,i) \implies Q(e,i)$.

When $Q(e,i)$ is true, there exists $h \geq i$ such that $e_h = e$, hence $\sigma_h\, e = \sigma_{h+1} \xrightarrow{e}\!\!\!\!\!/$. Thus, by Def. 6.4 it must be $e \notin \Sigma(\sigma_{h+1})$, which implies $P(e,i)$ to be false.

Therefore, since $\neg P(e,i)$ is true whenever $Q(e,i)$ is true, we have that in both cases $\neg P(e,i)$ is true, from which the thesis follows:

$$\sigma \text{ is fair} \iff \forall i \leq |\sigma|.\, \forall e.\, \neg P(e,i) \iff \forall i \leq |\sigma|.\, \neg\exists e.\, P(e,i) \qquad \square$$

During a play, if a participant eventually performs all the events that become enabled, then she is said to be *innocent*. The only way for not performing an enabled event, and still be considered innocent, is to perform an event which conflicts with it. If there exists even a single event of A which is enabled by the play but has not been performed (nor there is some conflict which prohibits it) then A is *culpable*.

Definition 6.21 (Innocence)**.** *We say that* A *is* innocent in σ *iff:*

$$\forall i \geq 0.\, \forall e \in \pi^{-1}(\mathsf{A}).\, \big(\overline{\sigma_i} \xrightarrow{e}_{\mathcal{E}} \implies \exists j \geq i.\, e_j \# e \;\vee\; e_j = e\big)$$

A strategy Σ *for* A *is innocent iff* A *is innocent in all fair plays which conform to* Σ*. If* A *is not innocent in* σ*, then we say she is* culpable *in* σ*.*

Not all strategies are innocent. For instance, the one which always prescribes A to do nothing is innocent only in case A really has nothing to do. There always exist strategies which guarantee A to be innocent in every (fair) play.

We say that a strategy Σ is greater than the strategy Σ', if for all plays σ, we have that $\Sigma'\sigma \subseteq \Sigma\sigma$. The greatest of such strategies is the *eager strategy*, which prescribes A to do all her enabled events.

Definition 6.22 (Eager strategy)**.** *We define the* eager strategy Σ^e_{A} *for* A *as follows:*

$$\Sigma^e_{\mathsf{A}}(\sigma) \;=\; \{e \in \pi^{-1}(\mathsf{A}) \mid \overline{\sigma} \xrightarrow{e} \}$$

Since the eager strategy Σ^e_{A} prescribes A to do all her enabled events, and since A is innocent if she performs all the events she has to, it is easy to see that A is innocent in every fair play which conforms to Σ^e_{A}.

Lemma 6.23. Σ^e_{A} *is the greatest innocent strategy for* A*.*

We now define when a participant *wins* in a play. If A is culpable, then she loses. If A is innocent, but some other participant is culpable, then A wins. Otherwise, if all participants are innocent, then A wins if she has a positive payoff in the play. This is formalised as the function $\mathcal{W}$ in Def. 6.24 below.

Definition 6.24 (Winning play)**.** *Define* $\mathcal{W} : \mathcal{A} \rightharpoonup E^\omega \to \{1, 0, -1\}$ *as:*

$$\mathcal{W}\mathsf{A}\sigma \;=\; \begin{cases} \Phi\mathsf{A}\sigma & \text{if all participants are innocent in } \sigma \\ -1 & \text{if } \mathsf{A} \text{ is culpable in } \sigma \\ +1 & \text{otherwise} \end{cases}$$

For a participant A *and a play* σ*, we say that* A wins *(resp.* loses*) in* σ *iff* $\mathcal{W}\mathsf{A}\sigma > 0$ *(resp.* $\mathcal{W}\mathsf{A}\sigma < 0$*).*

Note that in the last case, A is innocent but there exists some $\mathsf{B} \neq \mathsf{A}$ culpable in σ.

Definition 6.25 (Winning strategy)**.** *A strategy Σ is* winning *(resp.* losing*) for A iff A wins (resp. loses) in every fair play conform to Σ.*

Whenever A has a strategy Σ which allows her to win in all fair plays conform to Σ, then she *agrees* on that contract.

Definition 6.26 (Agreement)**.** *A participant* A agrees *on a contract $\mathcal{C} = \langle \mathcal{E}, \mathcal{A}, \pi, \Phi \rangle$ if and only if A has a winning strategy in $\mathcal{C}$. A contract $\mathcal{C}$ admits an agreement whenever all the participants in $\mathcal{A}$ agree on $\mathcal{C}$.*

Indeed, if A agrees on a contract, in any interaction regulated by that contract, she will win.

6.2.2 Examples

Example 6.27. *The contract $\mathcal{C}$ of Ex. 6.18 admits an agreement. The winning strategies for A and B are, respectively:*

$$\Sigma_{\mathsf{A}}(\sigma) = \begin{cases} \{a\} & \text{if } b \in \overline{\sigma} \text{ and } a \notin \overline{\sigma} \\ \emptyset & \text{otherwise} \end{cases} \qquad \Sigma_{\mathsf{B}}(\sigma) = \begin{cases} \{b\} & \text{if } b \notin \overline{\sigma} \\ \emptyset & \text{otherwise} \end{cases}$$

For A, the only plays fair and conform to Σ_{A} are ε and $\langle b\, a \rangle$. B is culpable in ε, while in $\langle b\, a \rangle$ the payoff of A is positive. For B, the only fair plays conform to Σ_{B} are $\langle b \rangle$ and $\langle b\, a \rangle$. A is culpable in $\langle b \rangle$, while in $\langle b\, a \rangle$ the payoff of B is positive. □

Example 6.28. *Let $\mathcal{C}$ be the contract resulting from the composition of the contracts of A and B in Example 6.7. We have that A agrees on $\mathcal{C}$, while B does not. A winning strategy for A is:*

$$\Sigma_{\mathsf{A}}(\sigma) \;=\; \begin{cases} \{\widetilde{a_i}\} & \text{if } \overline{\sigma} \xrightarrow{\widetilde{a_i}} \\ \emptyset & \text{otherwise} \end{cases}$$

There are only two fair plays conforming to Σ_{A}: the empty one (where B is culpable), and the infinite one $\sigma = \langle b\, \widetilde{a_0}\, \widetilde{a_1} \cdots \rangle$. In the play σ, A has a positive payoff, while B has a negative one. □

Example 6.29. *The contract resulting from the composition of the contracts of A and B in Example 6.8 admits an agreement. Indeed, the eager strategies for A and B are winning.*

$$\Sigma^e_{\mathsf{A}}(\sigma) = \begin{cases} \{a_i\} & \text{if } \overline{\sigma} \xrightarrow{a_i} \\ \{b_j\} & \text{if } \overline{\sigma} \xrightarrow{b_i} \end{cases} \qquad \Sigma^e_{\mathsf{B}}(\sigma) = \{\overline{a_i}, \overline{b_i}\} \quad \text{if } |\sigma| = i$$

To prove that Σ^e_{A} is winning for A, consider any fair play σ conform to Σ^e_{A}. If σ is empty, B is culpable in σ. If σ is finite and non-empty, it terminates with either a_i or b_i (events of A), and so B is culpable. Otherwise, if σ is infinite, then $\Phi\mathsf{A}\sigma = 1$.

To prove that Σ^e_{B} is winning for B, consider any fair play σ conform to Σ^e_{A}. If σ is finite, then it terminates with either $\overline{a_i}$ or $\overline{b_i}$, hence A is culpable in σ. Otherwise, if σ is infinite, then $\Phi\mathsf{B}\sigma = 1$. □

Example 6.30. *The eager strategy Σ^e_{A} is not always winning for A. For instance, consider the contract with $\vdash a$, $\vdash b$, $a\#b$, $\pi^{-1}(\mathsf{A}) = \{a, b\}$, and $\Phi\mathsf{A}\sigma = 1$ iff $a \in \overline{\sigma}$. We have that $\Sigma^e_{\mathsf{A}}(\varepsilon) = \{a, b\}$, but A is losing in the fair play $\sigma = \langle b \rangle$. However, A agrees on $\mathcal{C}$, because the strategy $(\lambda\sigma.\,\text{if } \overline{\sigma} \xrightarrow{a} \text{ then } \{a\} \text{ else } \emptyset)$ is winning for A.* □

Definition 6.31 (**Union of strategies**)**.** *Let Σ_a and Σ_b be two strategies, then we define their union as:*

$$\Sigma = \lambda\sigma.\Sigma_a(\sigma) \cup \Sigma_b(\sigma)$$

Example 6.32. *The union of two winning strategies is not necessarily a winning strategy. For instance, consider the contract with enablings $\vdash a$, $\vdash b$, $\{a\} \vdash a'$, $\{b\} \vdash b'$, and conflicts $a\#b'$, $a'\#b$ and where all the events belong to A. Let:*

$$\Sigma_a(\sigma) = \begin{cases} \{a\} & \text{if } \overline{\sigma} \xrightarrow{a} \\ \{a'\} & \text{if } \overline{\sigma} \xrightarrow{a'} \\ \emptyset & \text{otherwise} \end{cases} \qquad \Sigma_b(\sigma) = \begin{cases} \{b\} & \text{if } \overline{\sigma} \xrightarrow{b} \\ \{b'\} & \text{if } \overline{\sigma} \xrightarrow{b'} \\ \emptyset & \text{otherwise} \end{cases}$$

and let $\Phi\mathsf{A}\sigma$ be positive if either $a, a' \in \overline{\sigma}$, or $b, b' \in \overline{\sigma}$. Both Σ_a and Σ_b are winning strategies for A in $\mathcal{C}$, but their union $\Sigma = \lambda\sigma.\Sigma_a(\sigma) \cup \Sigma_b(\sigma)$ is not. Indeed $\Sigma(a) = \{a', b\}$, and so $\sigma = \langle a\, b \rangle$ is a fair play conform to Σ, such that $\Phi\mathsf{A}\sigma \leq 0$. Therefore, Σ is not winning for A in σ. □

6.2.3 Composition of strategies

We now define the composition $\sqcup$ of a set of strategies. Unlike for the union of winning strategies, their finite $\sqcup$-composition is guaranteed to be winning (Lemma 6.35).

Definition 6.33 (Composition of strategies)**.** *For a set of strategies $\mathcal{S}$, we define the strategy $\bigsqcup \mathcal{S}$ as:*

$$(\bigsqcup \mathcal{S})(\sigma) \;=\; \bigcup \{\Sigma(\sigma) \mid \Sigma \in \mathcal{S} \;\wedge\; \sigma \textit{ conforms to } \Sigma\}$$

Lemma 6.34. *For all sets of strategies $\mathcal{S}$, and for all σ, if σ conforms to Σ for some $\Sigma \in \mathcal{S}$, then σ conforms to $\bigsqcup \mathcal{S}$.*

Proof. Trivial by Def. 6.33 □

Lemma 6.35. *Let $\mathcal{S}$ be a finite set of strategies for participant A. Then:*

(a) If a play σ conforms to $\bigsqcup \mathcal{S}$, then there exists $\Sigma \in \mathcal{S}$ such that σ conforms to Σ.

(b) If each $\Sigma \in \mathcal{S}$ is winning for A in $\mathcal{C}$, then $\bigsqcup \mathcal{S}$ is a winning strategy for A in $\mathcal{C}$.

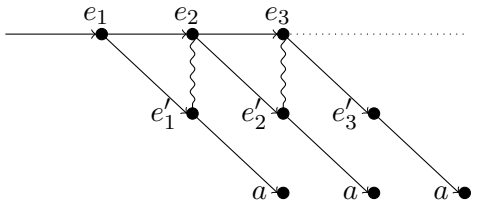


Figure 6.4: Joining an infinite set of winning strategy is not a winning strategy.

Proof. For item (a), we prove the contrapositive. Assume that σ does not conform to any $\Sigma \in \mathcal{S}$. By Def. 6.5, this means that:

$$\forall \Sigma \in \mathcal{S}.\ \exists i_\Sigma \geq 0.\ \pi(e_{i_\Sigma}) = \mathsf{A} \ \ \wedge \ \ e_{i_\Sigma} \notin \Sigma(\sigma_{i_\Sigma}) \tag{6.5}$$

Clearly, if σ_{i_Σ} does not conform to Σ, and so for all $j \geq i_\Sigma$, σ_j does not as well.

Since $\mathcal{S}$ is finite, we can take the maximum of the indices i_Σ obtained in (6.5) i.e. let:

$$k \ = \ \max\{i_\Sigma \mid \Sigma \in \mathcal{S}\}$$

By construction of k, $\pi(e_k) = \mathsf{A}$, but σ_k does not conform to any $\Sigma \in \mathcal{S}$. Then, by Def. 6.33, σ does not conform to $\bigsqcup S$.

To prove (b), let σ be a play conforming to $\bigsqcup \mathcal{S}$. By (a), there exists $\Sigma \in \mathcal{S}$ such that σ conforms to Σ. Since by hypothesis Σ is winning, then A wins in σ. So $\bigsqcup \mathcal{S}$ is a winning strategy for A. □

Unfortunately, Lemma 6.35(a) cannot be applied when the set of strategies is *infinite*. Indeed, for each event e_i of an infinite play σ fair and conforming to $\bigsqcup \mathcal{S}$, there may exists a different $\Sigma_i \in \mathcal{S}$ to whom each σ_i conforms, but not a single Σ to which the *whole* σ conforms. So, even if all the strategies in $\mathcal{S}$ are winning, $\bigsqcup \mathcal{S}$ may not be winning, as shown in the following example.

Example 6.36. *Let $\mathcal{C}_\mathsf{A} = \langle \mathcal{E}_\mathsf{A}, \mathcal{A}, \pi, \Phi_\mathsf{A} \rangle$ be a contract with the following payoff:*

$$\Phi \mathsf{A} \sigma \ = \ \begin{cases} 1 & \text{if } a \in \overline{\sigma} \\ -1 & \text{otherwise} \end{cases}$$

Let $\mathcal{E}_\mathsf{A}$ be the ES *with the following enablings and conflicts (see Fig. 6.4):*

$$\vdash\colon \ \{\ \vdash e_1\} \ \cup \ \{e_i \vdash e_{i+1} \mid i \geq 1\} \ \cup \ \{e_i \vdash e'_i \mid i \geq 1\} \ \cup \ \{e'_i \vdash a \mid i \geq 1\}$$
$$\#\colon \ \{e'_i \# e_{i+1} \mid i \geq 1\}$$

Let $\mathcal{A} = \{\mathsf{A}\}$, and let $\pi(a) = \pi(e_i) = \pi(e'_i) = \mathsf{A}$, for all $i \in \mathbb{N}$.

For all $i > 0$, let Σ^i be the strategy for A which prescribes to wait $i + 1$ events before performing a. Formally:

$$\Sigma^i(\sigma) \ = \ \begin{cases} \{e'_i\} & \text{if } |\sigma| = i \text{ and } \overline{\sigma} \xrightarrow{e'_i} \\ \{a\} & \text{if } |\sigma| = i+1 \text{ and } \overline{\sigma} \xrightarrow{a} \\ \{e_j\} & \text{if } |\sigma| < i \text{ and } \overline{\sigma} \xrightarrow{e_j} \\ \emptyset & \text{otherwise} \end{cases}$$

For all i, the strategy Σ^i is winning. Indeed, each play σ fair and conforming to Σ^i has the form $\langle e_1\ e_2\ \cdots\ e_i\ e'_i\ a\rangle$. For instance, $\langle e_1\ e'_1\ a\rangle$ is a fair play conforming to Σ^1 and $\langle e_1\ e_2\ e'_2\ a\rangle$ is a fair play conforming to Σ^2. In the play σ, the payoff of A is positive, hence Σ^i is winning for A.

Now, let $\mathcal{S} = \bigsqcup\{\Sigma^i \mid i \geq 0\}$, and let $\sigma^\infty = \langle e_1\ e_2\ e_3 \ldots\rangle$ be a play (actually, this is the only infinite play allowed in this ES*). We have that:*

- *σ^∞ is fair for $\bigsqcup\mathcal{S}$. Indeed, there does not exist an event $e \in (\bigsqcup\mathcal{S})\sigma^\infty_j$ for all i and all $j \geq i$. Thus, Lemma 6.20 states σ^∞ is fair.*
- *σ^∞ conforms to $\bigsqcup\mathcal{S}$, since for all i, σ^∞_i conforms to every Σ_j with $j > i$.*
- *A loses in σ^∞, since she will never perform the event a.*

Summing up, we have found a fair play that conforms to $\bigsqcup\mathcal{S}$ where A is not winning: therefore, $\bigsqcup\mathcal{S}$ is not a winning strategy. □

6.2.4 Agreements for Offer-Request payoffs

The following theorem gives a necessary condition for reaching an agreement on a contract with O-R payoffs. The ES must have a configuration containing at least a request set, and where all the offers are matched by the respective requests.

Lemma 6.37. *Let $\mathcal{C} = \langle\mathcal{E}, \mathcal{A}, \pi, \Phi\rangle$ be a contract with O-R payoff $\Phi\mathsf{A} = \lambda\sigma.\ \phi(\overline{\sigma})$ for A. If A agrees on $\mathcal{C}$, then there exists $C \in \mathcal{F}_\mathcal{E}$ such that $\phi(C) > 0$.*

Proof. Assume that A agrees on $\mathcal{C}$. By Def. 6.26, A has a winning strategy in $\mathcal{C}$, be it Σ_A. By Def. 6.25, A wins in every fair play which conforms to Σ_A. Among all these plays, there must exist at least one where all the participants are innocent (e.g. the play where all $\mathsf{B} \neq \mathsf{A}$ adopt the eager strategy Σ^e_B), call it σ. Since A wins in σ, by Def. 6.24 we have $\Phi\mathsf{A}\sigma > 0$. To conclude, it suffices to observe that by Lemma 6.3 for all plays σ, the set $\overline{\sigma}$ is a configuration of $\mathcal{E}$. □

The following example shows that the converse of Lemma 6.37 does not hold. Indeed, it is not enough to require that $\phi(C) > 0$ for some $C \in \mathcal{F}_\mathcal{E}$ to agree on a contract: in some cases a conflict may prevent us from reaching a positive payoff.

Example 6.38. *Let $\mathcal{C}$ be a contract with O-R payoffs, defined as follows:*

$$\begin{array}{rl}
\pi: & \{a \mapsto \mathsf{A},\ a' \mapsto \mathsf{A},\ b \mapsto \mathsf{B},\ b' \mapsto \mathsf{B}\} \\
\vdash: & \vdash a \qquad \vdash a' \qquad \vdash b \qquad \vdash b' \\
\#: & a \# a' \qquad b \# b' \\
\Phi\mathsf{A}: & O^0_\mathsf{A} = \{a\} \qquad R^0_\mathsf{A} = \{b\} \qquad O^1_\mathsf{A} = \{a'\} \qquad R^1_\mathsf{A} = \{b'\}
\end{array}$$

Even though there exist two configurations, $\{a, b\}$ and $\{a', b'\}$, where A has a positive payoff, there are also some plays, e.g. $\langle a, b'\rangle$ and $\langle a', b\rangle$, where she has a negative payoff, and hence she loses. Since A has no innocent strategy to avoid these plays, A does not agree on the contract $\mathcal{C}$. □

The following theorem establishes a sufficient condition for reaching agreements in conflict-free contracts with O-R payoffs. If there exists a configuration C in $\mathcal{C}$ which contains all the requests of A, then A agrees on $\mathcal{C}$. Since the ES of $\mathcal{C}$ is conflict-free, if the strategy of A prescribes to do all her enabled events in C, then the other participants are obliged to do their events in C. Eventually, either some participant $\mathsf{B} \neq \mathsf{A}$ is culpable, or a state is reached where the payoff of A is positive.

Theorem 6.39. *Let $\mathcal{C}$ be a contract with O-R payoff for A. If $\mathcal{E}$ is conflict-free and $\bigcup_i R^i_{\mathsf{A}} \subseteq C$ for some $C \in \mathcal{F}_{\mathcal{E}}$, then A agrees on $\mathcal{C}$.*

Proof. We will prove that the eager strategy Σ^e_{A} is winning for A in $\mathcal{C}$. Let γ be a fair play of $\mathcal{C}$ which conforms to Σ^e_{A}.

By contradiction, assume that A is *not* winning in γ. By Lemma 6.23, A is innocent in γ. Thus, by Def. 6.24 it follows that all participants are innocent and, $\Phi_{\mathsf{A}}\gamma \leq 0$. By Def. 6.11, this means that either there exist some i such that $O^i_{\mathsf{A}} \subseteq \overline{\gamma}$ and $R^i_{\mathsf{A}} \not\subseteq \overline{\gamma}$ (in case A loses), or that for all i, $O^i_{\mathsf{A}} \not\subseteq \overline{\gamma}$ and $R^i_{\mathsf{A}} \not\subseteq \overline{\gamma}$ (in case A ties). In both case, there exists at least one i such that $R^i_{\mathsf{A}} \not\subseteq \overline{\gamma}$.

Let i be such that $R^i_{\mathsf{A}} \not\subseteq \overline{\gamma}$, and let e be such that $e \in R^i_{\mathsf{A}} \setminus \overline{\gamma}$. By hypothesis, there exists $C \in \mathcal{F}_{\mathcal{E}}$ such that $\bigcup_i R^i_{\mathsf{A}} \subseteq C$; hence $e \in C$.

Since C is a configuration, and since every family of configurations enjoy finiteness, by Lemma 3.12 there exists $C' \subseteq_{fin} C$ such that $C' \in \mathcal{F}_{\mathcal{E}}$ and $e \in C'$. By Lemma 3.8, there exists a play σ such that $\emptyset \xrightarrow{\sigma}_{\mathcal{E}} \overline{\sigma}$, and $e \in \overline{\sigma} = C'$.

We will prove that $\overline{\sigma} \subseteq \overline{\gamma}$ by induction on the length of σ. The base case $\sigma_0 = \varepsilon$ is trivial. For the inductive case, we have to prove that $\overline{\sigma_{i+1}} = \overline{\sigma_i} \cup \{e_i\} \subseteq \overline{\gamma}$. By the induction hypothesis, $\overline{\sigma_i} \subseteq \overline{\gamma}$ for $i < |\sigma|$, hence it is enough to prove that $e_i \in \gamma$.

By contradiction, assume that $e_i \notin \overline{\gamma}$. Let γ_k be the shortest prefix of γ such that $\overline{\sigma_i} \subseteq \overline{\gamma_k}$. Since $\overline{\sigma_i} \vdash e_i$, by Lemma 3.9 it follows that $\overline{\gamma_h} \xrightarrow{e}_{\mathcal{E}}$ for all $h \geq k$. Since all participants are innocent in γ, and since $\mathcal{E}$ is conflict-free, by Def. 6.21 there exists $j > k$ such that the j-th event of γ is e_i — contradiction.

Summing up, we have proved that $\bigcup_i R^i_{\mathsf{A}} \subseteq \overline{\gamma}$ for all fair plays γ. Therefore, A has a winning strategy (Σ^e_{A}) in $\mathcal{C}$, from which we conclude that A agrees on $\mathcal{C}$. □

Note that the conflict-freeness requirement in Theorem 6.39 cannot be dropped. Actually, in the presence of conflicts, Example 6.38 shows that there exists a configuration containing all the requests of A, but A does not agree on the contract.

Some events in a contract may be immaterial in reaching agreements. For instance, consider an ES with the following enablings and conflicts (all events are of A):

$$\{\vdash e_i \mid i \in I\} \qquad \{e_i \# e_0 \mid i \in I\}$$

If the set I is infinite, participant A can perpetually prevent e_0 from being fired, while remaining innocent. Note instead that if the set I is finite, B could conflict the events in I, except e_0, and then A would be obliged to do it, or become culpable. Therefore, if I is infinite event e_0 cannot play any role in reaching an agreement. We call these events *internal*, and hereafter we assume that contracts are free from internal events.

Definition 6.40 (Internal events)**.** *We say an event e is* internal *in a contract $\mathcal{C}$ iff, for all $\mathcal{C}'$ compatible with $\mathcal{C}$, there exists an innocent strategy Σ for $\pi(e)$ such that, for all plays of $\mathcal{C} \mid \mathcal{C}'$ conforming to Σ, e never occurs in σ.*

6.3 Protection

In contract-oriented interactions, mutually distrusted participants advertise their contracts to a contract broker. The broker composes contracts which admit an agreement, and then establishes a session among the participants involved in them. When a participant agrees on a contract, she is guaranteed that — even in the presence of malicious participants — no interaction driven by the contract will ever go wrong. At worst, if A does not reach her objectives, then some other participant will be culpable of a contract infringement.

This model of interaction works fine under the hypothesis that contract brokers are honest, i.e. they never establish a session in absence of an agreement among all the participants.

Suppose Alice is willing to lend her airplane in exchange of Bob's bike. In her contract, she could promise to lend the airplane (unconditionally), and declare that her objective is to obtain the bike. A malicious contract broker could construct an attack by establishing a session between Alice and Mallory, whose contract just says to take the airplane and give nothing in exchange. Mallory is *not* culpable, because her contract declares no obligations, and so Alice loses.

In this scenario, the contract broker may play the role of an attacker. More precisely, our adversarial model is the following:

- the contract broker may be malicious: it can put together in a session contracts which do not admit an agreement. Note that the semantics of [BTZ12b] disallows this kind of misbehaviour, while [BTZ12a] permits it.
- a judge can only inspect the obligations declared in a contract (i.e. the event structure), and neglects the payoffs. This is reasonable since in some works, goals were not decided by participants themselves (e.g. in [BTZ12a] where the "payoffs" are decided by the contract broker).

Formally, a contract $\mathcal{C}_\mathsf{A}$ *protects* A if, whatever contract $\mathcal{C}$ is composed with $\mathcal{C}_\mathsf{A}$, A has a way to (if not win) at least non-lose in the composed contract.

Definition 6.41 (Protection)**.** *A contract* $\mathcal{C}_\mathsf{A}$ protects *participant* A *if and only if, for all contracts $\mathcal{C}$ compatible with $\mathcal{C}_\mathsf{A}$, A has a non-losing strategy in $\mathcal{C}_\mathsf{A} \mid \mathcal{C}$.*

Note that if A agrees with $\mathcal{C}$, then not necessarily $\mathcal{C}$ protects A. For instance, Mallory could join $\mathcal{C}$ with her contract $\mathcal{C}_\mathsf{M}$, and prevent Alice from borrowing Bob's bike in $\mathcal{C} \mid \mathcal{C}_\mathsf{M}$. A sufficient (yet hardly realistic) criterion for protection is to declare nonnegative payoffs for all σ. Less trivially, the following example shows a contract with possible negative payoffs which still offers protection.

Example 6.42. *The contract* $\mathcal{C}_\mathsf{B}$ *of Example 6.6 does* not *protect Bob. To prove that, consider e.g. the attacker contract* $\mathcal{C}' = \langle \mathcal{E}', \mathcal{A}, \pi, \Phi_{\mathcal{C}'} \rangle$*, where* $\mathcal{A}$ *and* π *are as in Example 6.6, while we define* $\mathcal{E}'$ *with no enablings, and* $\Phi_{\mathcal{C}'}$ *is not relevant except for being undefined on* B *(otherwise* $\mathcal{C}'$ *and* $\mathcal{C}_\mathsf{B}$ *would not be compatible). Consider then the contract* $\mathcal{C}' \mid \mathcal{C}_\mathsf{B}$*. There are only two possible strategies for* B*:*

$$\Sigma_\mathsf{B} = \lambda\sigma.\, \emptyset \qquad\qquad \Sigma'_\mathsf{B} = \lambda\sigma. \begin{cases} \{b\} & \text{if } b \notin \overline{\sigma} \\ \emptyset & \text{otherwise} \end{cases}$$

The strategy Σ_B *is losing for* B*, because* B *is not innocent under* Σ_B*. The strategy* Σ'_B *is losing as well, because in the play* $\sigma = \langle b \rangle$ *(fair and conform to* Σ_B*), no participant is culpable (according to* $\mathcal{C}' \mid \mathcal{C}_\mathsf{B}$*) and* $\Phi\mathsf{B}\sigma = -1$*. Hence by Def. 6.41,* B *is not protected by* $\mathcal{C}_\mathsf{B}$*.*

Instead, the contract $\mathcal{C}_\mathsf{A}$ *protects Alice. To show that, consider a contract* $\mathcal{C}$ *compatible with* $\mathcal{C}_\mathsf{A}$*. Let* Σ_A *be the following strategy for* A*:*

$$\Sigma_\mathsf{A} = \lambda\sigma. \begin{cases} \{a\} & \text{if } b \in \overline{\sigma} \text{ and } a \notin \overline{\sigma} \\ \emptyset & \text{otherwise} \end{cases}$$

Let σ *be a play in* $\mathcal{C} \mid \mathcal{C}_\mathsf{A}$ *fair and conform to* Σ_A*. There are two cases:*

- $b \in \overline{\sigma}$*. Since* σ *is fair for* Σ_A*, either* $a \in \overline{\sigma}$*, or there exists some* $e \in \overline{\sigma}$ *such that* $e \# a$*. In both cases,* A *is innocent in* σ*. Furthermore,* $\Phi\mathsf{A}\sigma = 1$*.*
- $b \notin \overline{\sigma}$*. By definition of* $\mathcal{C}_\mathsf{A}$*, and since* $\mathcal{C}$ *is not specifying any further obligations for* A *(otherwise it would not be compatible with* $\mathcal{C}_\mathsf{A}$*), then* A *is not culpable in* σ*. Also, since* $b \notin \overline{\sigma}$ *and* $a \notin \overline{\sigma}$*, then* $\Phi\mathsf{A}\sigma = 0$*.*

In both cases, Σ_A *is non-losing for* A*. Therefore,* $\mathcal{C}_\mathsf{A}$ *protects* A*.* □

6.3.1 Protection for Offer-Request payoffs

We now discuss protection in case the contract has Offer-Request payoffs.

A necessary condition to being protected is to specify non-empty offers sets. In fact if A were specifing an empty set of offers, she would lose in an empty play. Intuitively, A is saying that she wants something by doing nothing in exchange. This means that when *nothing is done*, she expects her requests to be satisfied. So even in the case of an empty set of obligations, A is protected only if she specifies non-empty offer sets.

Example 6.43. *Assume* $\mathcal{C}_\mathsf{A}$ *has an empty set of offers and a non-empty set of requests:*

$$O^0_\mathsf{A} = \emptyset \qquad R^0_\mathsf{A} \neq \emptyset$$

In the case where the contract of B *prescribes to* B *to do nothing, in the play where no events have been performed,* B *is innocent and* A *loses. Hence* $\mathcal{C}_\mathsf{A}$ *does not protect* A*, as correctly predicted by Lemma 6.44 below.* □

Lemma 6.44. *If the contract* $\mathcal{C}_\mathsf{A} = \langle \mathcal{E}, \mathcal{A}, \pi, \Phi \rangle$ *with O-R payoffs for* A *protects* A*, then* $\forall h.\ O^h_\mathsf{A} \neq \emptyset$

Proof. By Def. 6.41, for every contract $\mathcal{C}$ compatible with $\mathcal{C}_\mathsf{A}$, A has a non-losing strategy Σ in $\mathcal{C}_\mathsf{A} \mid \mathcal{C}$.

Let $\mathcal{C}$ have no enabling for any of the events in R^i for all i, and let σ be a fair play of $\mathcal{C}_\mathsf{A} \mid \mathcal{C}$ conform to Σ.

Since $\mathcal{C}$ has no enablings for any R^i, there exist no h such that $R^h_\mathsf{A} \subseteq \overline{\sigma}$.

According to Def. 6.11, the only way for A to lose is to have $O^i_\mathsf{A} \subseteq \overline{\sigma}$ and $R^i_\mathsf{A} \not\subseteq \overline{\sigma}$ for some i. So, since A does not lose in σ, then for all i, $O^i_\mathsf{A} \not\subseteq \overline{\sigma}$, and we conclude that $O^i_\mathsf{A} \neq \emptyset$ for all i. □

A sufficient condition for A to be protected is to promise to do what she offers in the O-R contract, only *after* the other participants have fulfilled her requests. More precisely, A is protected if, whenever she enables an offer O^i_A, the corresponding request R^i_A has been already satisfied. However in the case of circular payoffs, if every participant tries to protect himself in this way then no one will be willing to do the first move, an so no agreement will ever be reached.

Theorem 6.45. *A contract* $\mathcal{C}_\mathsf{A} = \langle \mathcal{E}, \mathcal{A}, \pi, \Phi \rangle$ *with O-R payoffs for* A *protects* A *if*

$$\forall i, Y.\ \ Y \vdash O^i_\mathsf{A} \implies R^i_\mathsf{A} \subseteq Y \tag{6.6}$$

Proof. Let $\mathcal{C}_\mathsf{A}$ be a contract with O-R payoffs for A such that (6.6) holds. Let $\mathcal{C}$ be a contract compatible with $\mathcal{C}_\mathsf{A}$. We will prove that Σ^e_A is a non-losing strategy for A in $\mathcal{C}_\mathsf{A} \mid \mathcal{C}$. Let σ be a fair play of $\mathcal{C}_\mathsf{A} \mid \mathcal{C}$ conform to Σ^e_A.

By contradiction, assume that A loses in σ, i.e. by Def. 6.24 and by Def. 6.11:

$$\exists i.\ \ O^i_\mathsf{A} \subseteq \overline{\sigma} \ \wedge\ R^i_\mathsf{A} \not\subseteq \overline{\sigma} \tag{6.7}$$

Since $O^i_\mathsf{A} \subseteq \overline{\sigma}$, then for all $e \in O^i_\mathsf{A}$ it must be $\overline{\sigma} \vdash e$, that is, $\overline{\sigma} \vdash O^i_\mathsf{A}$. By (6.6) it follows that $R^i_\mathsf{A} \subseteq \overline{\sigma}$ — which contradicts (6.7). □

Example 6.46. *The condition in Theorem 6.45 is not necessary to have protection. Indeed, in a contract for* A *with no obligations and non-empty offers,* A *would be protected, since she could do nothing and non-lose. Also, in case* A *offers an unreachable event,* A *is protected since she will never obliged to do what she offers.* □

6.3.2 Agreement and protection cannot coexist

A remarkable feature of *finite* circular payoffs is that, in each play where all participants win, at some point there exists a participant A which has performed all the offers in O^i_A before having obtained all the requests in R^i_A. Intuitively, the participant A which makes this "first step" is not protected.

Lemma 6.47. *Let* $\mathcal{C}$ *be a contract with finite circular O-R payoffs. If* σ *is a winning play for all participants in* $\mathcal{A}$*, then there exists a prefix* η *of* σ *and a participant* $\mathsf{A} \in \mathcal{A}$ *such that* $\Phi\mathsf{A}\eta < 0$.

Proof. Since σ is a winning play for $\mathcal{A}$, by Def. 6.24 it must be $\Phi \mathsf{A} \sigma > 0$ for all A. Let $\eta\, e$ be the shortest prefix of σ such that all the participants have strictly positive payoffs (of course, a winning play must have strictly positive length). We shall prove that $\Phi \mathsf{B} \eta < 0$, for some participant B.

By contradiction, assume that there exists no $\mathsf{A} \in \mathcal{A}$ such that $\Phi \mathsf{A} \eta < 0$. Since not all participants are winning in η, there should then exist some participant B such that $\Phi \mathsf{B} \eta = 0$, for all j, $R^j_{\mathsf{B}} \not\subseteq \overline{\eta}$. Since $\eta\, e$ is winning for $\mathcal{A}$ (and in particular for B), let i_{B} be such that $e \in R^{i_{\mathsf{B}}}_{\mathsf{B}} \subseteq \overline{\eta} \cup \{e\}$, and for all $\mathsf{A} \neq \mathsf{B}$ let i_{A} be such that $R^{i_{\mathsf{A}}}_{\mathsf{A}} \subseteq \overline{\eta} \cup \{e\}$. Since Φ is circular, by Def. 6.13 there exists a function $\mathcal{J} : \mathcal{A} \to \mathbb{N}$ such that $\overline{\eta} \cup \{e\} \supseteq \bigcup_{\mathsf{A} \in \mathcal{A}} R^{i_{\mathsf{A}}}_{\mathsf{A}} \supseteq \bigcup_{\mathsf{A} \in \mathcal{A}} O^{\mathcal{J}\mathsf{A}}_{\mathsf{A}}$. Therefore, $O^{\mathcal{J}\mathsf{B}}_{\mathsf{B}} \subseteq \overline{\eta} \cup \{e\}$. Since $e \in R^{i_{\mathsf{B}}}_{\mathsf{B}}$, by Def. 6.13 it must be $e \notin O^{\mathcal{J}\mathsf{B}}_{\mathsf{B}}$ (because by Def. 6.11, $O^i \cap R^i = \emptyset$ for all i), hence $O^{\mathcal{J}\mathsf{B}}_{\mathsf{B}} \subseteq \overline{\eta}$. Since by hypothesis the payoff of B is zero in η, it must be $O^{\mathcal{J}\mathsf{B}}_{\mathsf{B}} \not\subseteq \overline{\eta}$ — contradiction. □

Lemma 6.47 does not hold if the payoff is non-circular, as illustrated by the following example.

Example 6.48. *Consider the following non-circular payoff for* A, B, C*:*

$$O^1_{\mathsf{A}} = \{a, a', a''\} \qquad O^1_{\mathsf{B}} = \{b\} \qquad O^1_{\mathsf{C}} = \{c\}$$
$$R^1_{\mathsf{A}} = \{b, c\} \qquad R^1_{\mathsf{B}} = \{a, a'\} \qquad R^1_{\mathsf{C}} = \{b\}$$

In the play $\sigma = \langle a\, a'\, b\, c \rangle$ *every participant is winning, but no one is losing any prefix of* σ*. In particular:*

- *in* $\sigma_2 = \langle a\, a' \rangle$ *(and its prefixes) no participant has done all her offers.*
- *in* $\sigma_3 = \langle a\, a'\, b \rangle$, A *and* C *have not done all her offers, and* B *has obtained his requests.* □

The main result of this chapter follows. It states that if a set of contracts with finite circular O-R payoffs admits an agreement, then some of the participants is not protected, and *vice versa*.

Theorem 6.49. *Let* $\mathcal{C}_1, \ldots, \mathcal{C}_n$ *be contracts with circular finite O-R payoffs for* $\mathsf{A}_1, \ldots, \mathsf{A}_n$*, respectively. Then, at most one of the following statements is true:*

(a) $\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n$ *admits an agreement;*

(b) for all $i \in 1..n$, $\mathcal{C}_i$ *protects* A_i*.*

Proof. Assume that the statement (a) is true, i.e. all the participants agree on the contract $\mathcal{C} = \mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n$. By Def. 6.26, each $\mathsf{A}_i \in \{\mathsf{A}_1, \ldots, \mathsf{A}_k\} = \mathcal{A}$ has a winning strategy Σ_i in $\mathcal{C}$. Let σ be a fair play of $\mathcal{C}$ conform to all the Σ_i. Since all the participants win in σ, by Lemma 6.47 there exists $k \in 1..n$ and a finite prefix η of σ such that $\Phi \mathsf{A}_k \eta < 0$. By Def. 6.11, this amounts to say that there exists h such that $O^h_k \subseteq \overline{\eta}$ and $R^h_k \not\subseteq \overline{\eta}$.

We now prove that $\mathcal{C}_k$ does *not* protect A_k. To do that, we construct a contract $\mathcal{C}' = \langle \mathcal{E}', \mathcal{A}, \pi', \Phi' \rangle$ such that A_k does not have a non-losing strategy in $\mathcal{C}' \mid \mathcal{C}_k$. The functions π'

and Φ' in $\mathcal{C}'$ are almost immaterial: we just require that they make $\mathcal{C}'$ compatible with $\mathcal{C}_k$. The ES $\mathcal{E}'$ comprises a set of events $\widetilde{E}$ disjoint from the events of $\mathcal{C}$ and such that $\widetilde{e} \in \widetilde{E}$ for all event e in $\mathcal{C}$. The enablings and conflicts of $\mathcal{E}'$ are the following:

$$\{ \vdash e \mid e \in \overline{\eta} \setminus \pi^{-1}(\mathsf{A}_k)\} \ \cup \ \{ \vdash \widetilde{e} \mid e \in \pi^{-1}(\mathsf{A}_k)\}$$
$$\{e \# \widetilde{e} \mid e \in \pi^{-1}(\mathsf{A}_k) \setminus \overline{\eta}\}$$

Intuitively, $\mathcal{C}'$ enables all the events of $\mathcal{A} \setminus \{\mathsf{A}_k\}$ in η, and also all the events $\widetilde{e}$ in conflict with some event of A_k, *except* for the events in η. The goal of $\mathcal{C}'$ is to force A_k to do O_k^h, and then stop before R_k^h is reached. To implement this goal, $\mathcal{C}'$ must also be innocent in η.

By contradiction, assume that Σ is a non-losing strategy for A_k in $\mathcal{C}' \mid \mathcal{C}_k$. Assume that $\mathcal{C}'$ adopts the eager strategy $\Sigma^e_{\mathcal{C}'}$, i.e. all the enabled events are in its strategy. By fairness, there must exist a fair play ν of $\mathcal{C}' \mid \mathcal{C}_k$ which conforms to Σ and $\Sigma^e_{\mathcal{C}'}$, and where (1) all the participants are innocent (recall that the eager strategy is innocent, by Lemma 6.23) and (2) all the events $\widetilde{e}$ in conflict with the events of A_k not in η have been performed before any other event of A_k not in η. By (1), A_k is innocent in ν, hence by Def. 6.21:

$$\forall i \geq 0.\, \forall e \in \pi^{-1}(\mathsf{A}_k).\, \big(\overline{\nu_i} \xrightarrow{e} \implies \exists j \geq i.\, e_j \# e \ \vee \ e_j = e\big) \tag{6.8}$$

By (2), there cannot be events in conflict with those in $\overline{\eta} \cap \pi^{-1}(\mathsf{A}_k)$. We can then rewrite equation (6.8) as follows:

$$\forall i \geq 0.\, \forall e \in O_k^h.\, \big(\overline{\nu_i} \xrightarrow{e} \implies \exists j \geq i.\, e_j = e\big)$$

Summing up, there exists a fair play ν of $\mathcal{C}' \mid \mathcal{C}_k$ which conform to Σ, and where $O_k^h \subseteq \overline{\nu}$. Furthermore, $R_k^h \not\subseteq \overline{\nu}$, because the ES $\mathcal{E}'$ only enables the events in $\overline{\eta}$, while $\overline{\eta}$ does not contain all the events in R_k^h. By Def. 6.11, $\mathcal{W}\mathsf{A}_k\nu = \Phi\mathsf{A}_k\nu < 0$, i.e. A_k loses in ν — contradiction. □

Example 6.50. *Consider the contract $\mathcal{C}_\mathsf{A}$ with enabling $b \vdash a$ and the finite circular O-R payoff $O^0_\mathsf{A} = \{a\}$, $R^0_\mathsf{A} = \{b\}$, and the contract $\mathcal{C}_\mathsf{B}$ with enabling $a \vdash b$ and payoff $O^0_\mathsf{B} = \{b\}$, $R^0_\mathsf{B} = \{a\}$.*

Every participant is protected by her own contract, but the composed contract $\mathcal{C}_\mathsf{A} \mid \mathcal{C}_\mathsf{B}$ does not admit an agreement, as correctly predicted by Theorem 6.49. □

Agreement and protection can coexist in contracts with *infinite* circular O-R payoffs, as shown by the following example. Intuitively, when an infinite offer O_A has to match an infinite request R_B, participants A and B may take turns in doing event in $O_\mathsf{A} \cup R_\mathsf{B}$. This strategy is winning for both participants (hence they have an agreement), and protection follows because no participant completes her offer before receiving the corresponding request.

Example 6.51. *Let $\mathcal{C}_\mathsf{A} = \langle \mathcal{E}_\mathsf{A}, \mathcal{A}, \pi, \Phi_\mathsf{A}\rangle$ and $\mathcal{C}_\mathsf{B} = \langle \mathcal{E}_\mathsf{B}, \mathcal{A}, \pi, \Phi_\mathsf{B}\rangle$ be contracts with circular O-R payoffs (with infinite offers/requests) defined as follows:*

$$O_\mathsf{A} \ = \ \{e_i \mid i \in \mathbb{N}\} \ = \ R_\mathsf{B} \qquad\qquad R_\mathsf{A} \ = \ \{\overline{e_i} \mid i \in \mathbb{N}\} \ = \ O_\mathsf{B}$$

and let $\mathcal{A} = \{\mathsf{A}, \mathsf{B}\}$, $\pi(e_i) = \mathsf{A}$, $\pi(\overline{e_i}) = \mathsf{B}$ *for all* $i \in \mathbb{N}$. *Let the* ES $\mathcal{E}_\mathsf{A}$ *and* $\mathcal{E}_\mathsf{B}$ *be defined by the following enablings (and no conflicts):*

$$\mathcal{E}_\mathsf{A} : \ \{\vdash e_0\} \cup \{\overline{e_i} \vdash e_{i+1} \mid i \geq 0\} \qquad\qquad \mathcal{E}_\mathsf{B} : \ \{e_i \vdash \overline{e_i} \mid i \geq 0\}$$

The contract $\mathcal{C} = \mathcal{C}_\mathsf{A} \mid \mathcal{C}_\mathsf{B}$ *admits an agreement. We prove separately that* A *and* B *agree on* $\mathcal{C}$. *Let* Σ^e_A *be the eager strategy for* A. *Let* σ *be a fair play of* $\mathcal{C}$ *conform to* Σ^e_A. *We prove that* A *wins in* σ. *By Lemma 6.23, the strategy* Σ^e_A *makes* A *innocent in* σ. *There are two subcases. If* B *is not innocent in* σ, *then* A *wins. Otherwise, the play* σ *must be infinite, i.e.* $\overline{\sigma} = \{e_i\}_{i\in\mathbb{N}} \cup \{\overline{e_i}\}_{i\in\mathbb{N}}$. *Therefore,* $R_\mathsf{A} \subseteq \overline{\sigma}$, *and so* A *wins. To prove that* B *has a winning strategy in* $\mathcal{C}$ *we proceed similarly, by choosing the eager strategy* Σ^e_B *for* B.

We now show that $\mathcal{C}_\mathsf{A}$ *protects* A. *Let* $\mathcal{C}'$ *be compatible with* $\mathcal{C}_\mathsf{A}$. *The eager strategy* Σ^e_A *is non-losing for* A. *Indeed, in every fair play* σ *conform to* Σ^e_A, *if there exists* $\overline{e_i} \in R_\mathsf{A} \not\subseteq \overline{\sigma}$ *then* $e_{i+1} \in O_\mathsf{A} \notin \overline{\sigma}$, *and so* $\Phi\mathsf{A}\sigma \geq 0$. *To prove that* $\mathcal{C}_\mathsf{B}$ *protects* B, *we proceed similarly, by choosing the eager strategy* Σ^e_B *for* B. □

Chapter 7

Event structures with circular causality

Circular reasoning often appears in the compositional modelling and verification of concurrent systems [AL93, AP93, Mai03, VV01]. Circularity is also a frequent situation when reasoning about contracts [BZ10a]. A task may depend on others which have already been executed (dependencies in the past), but also on behalf that some other tasks will be performed in the future. Circularity arises when two or more tasks mutually rely on the guarantees provided by each other (circular dependencies).

As already noted, extensions to ES often use other relations to model other kind of dependencies. ES can provide a basic semantic model for assume/guarantee rules, by interpreting the enabling $b \vdash a$ as the promise: "I will do a *after* you have done b". However, circularity is usually prohibited in ES, either at the syntactic level, like in Winskel's prime event structures, or at the semantic level, like in Boudol's flow event structures [Bou90]. Indeed, the classical notion of causality among events only captures dependencies in the past. For instance, in the ES with enablings $b \vdash a$ and $a \vdash b$, none of the events a and b is reachable, because of the circularity of the constraints.

We propose here an extension of Winskel's event structures with a new *circular causality* relation ($\Vdash$). The ES prescribing $b \Vdash a$ (intuitively, "I will do a if you *promise* to do b") together with the other prescription $b \Vdash a$ has a configuration where both a and b have happened, despite of the circular dependencies. We stress that, to the best of our knowledge, differently from other extensions to event structures, circular dependencies are always *solved* when considering configurations, which is not our case. The configurations of these new ES do still enjoy the finiteness and finite-completeness properties of classical ES, thought they are not coincidence-free, which is correct from our point of view because of the presence of circular dependencies.

Chapter overview The rest of this chapter is organised as follows. In Section 7.1 we formally define CES and their configurations. We introduce a more general notion of configuration, called X-configuration, and we illustrate it with some examples. Then we present some basic results for traces (Section 7.2.1) and for configurations (Section 7.2.2). In Section 7.2.3 we study coherence, finiteness and coincidence-freeness in CES, and we prove that the family of configurations of an ES can be generated by a CES with circular enablings, only. In Section 7.3 we study reachable events, i.e. those events which belong to some configuration. For conflict-free CES we characterise reachability through an inductive definition, equivalent to the original one. In Section 7.4 we define an LTS for CES, and we relate it with configurations. In Section 7.5 we reconsider the above definitions, to deal with the case that credits done in the past can be discharged. In Section 7.6 we study urgent events, i.e. those events which are either $\vdash$-enabled by the past, or can be taken on credit, on the guarantee that such credit will be honoured in the future.

7.1 Basic definitions

Definition 7.1 (CES)**.** *An event structure with circular causality (*CES*) is a quadruple $\mathcal{E} = (E, \#, \vdash, \Vdash)$ where:*

- *E is a set of events,*
- *$\# \subseteq E \times E$ is an irreflexive and symmetric relation, called* conflict *relation. We say that a set $X \subseteq E$ is* conflict-free *($CF(X)$ in symbols) whenever $\forall e, e' \in X. \neg(e \# e')$. We denote with Con the set $\{X \subseteq_{fin} E \mid CF(X)\}$,*
- *$\vdash \subseteq Con \times E$ is the* enabling *relation,*
- *$\Vdash \subseteq Con \times E$ is the* circular enabling *relation,*

The relations $\vdash$ and $\Vdash$ are saturated*, i.e. for all $X, Y \in Con$ and for $\circ \in \{\vdash, \Vdash\}$:*

$$X \circ e \ \wedge \ X \subseteq Y \implies Y \circ e$$

We say that $\mathcal{E}$ is finite *when E is finite; we say that $\mathcal{E}$ is* conflict-free *when the conflict relation is empty.*

Notation shortcuts introduced for $\vdash$, also holds for $\Vdash$.

Notation 7.2. *We adopt the following conventions: $\Vdash e$ stands for $\emptyset \Vdash e$; we write $a \Vdash b$ for $\{a\} \Vdash b$. For a finite, conflict free set X, we write $X \Vdash Y$ for $\forall e \in Y.\ X \Vdash e$. For an infinite, conflict-free X, we write $X \Vdash Y$ as a shorthand for $\exists X_0 \subseteq_{fin} X.\ X_0 \Vdash Y$.*

We refine the notion of configuration in [Win88] to deal with circular causality. Intuitively, for all events e_i in the sequence $\langle e_0 \dots e_n \rangle$, e_i can either be $\vdash$-enabled by its predecessors, or $\Vdash$-enabled by the *whole* sequence, i.e:

Recalling the notation for sequences introduced in page 21, we denote with $\langle e_0\, e_1 \dots \rangle$ the (possibly infinite) sequence of elements $e_0, e_1, \dots$ and we write σ_i for the subsequence $\langle e_0 \dots e_{i-1} \rangle$.

Definition 7.3 (Configuration)**.** *For a* CES $\mathcal{E} = \langle E, \#, \vdash, \Vdash \rangle$*, we say that* $C \subseteq E$ *is a* configuration of $\mathcal{E}$ *iff* $CF(C)$*, and*

$$\forall e \in C.\ \exists \sigma = \langle e_0 \dots e_n \rangle.\ e \in \overline{\sigma} \subseteq C \ \wedge\ \forall i \leq n.\ \big(\{e_0, \dots, e_{i-1}\} \vdash e_i \ \vee\ \overline{\sigma} \Vdash e_i\big)$$

The set of all configurations of a CES $\mathcal{E}$ *is denoted by* $\mathcal{F}_\mathcal{E}$*.*

Clearly, configurations of a CES without $\Vdash$-enablings are also configurations in the sense of [Win88], hence CES are a conservative extension of Winskel's ES.

Theorem 7.4. *Let* $\mathcal{E} = \langle E, \#, \vdash \rangle$ *be an* ES*. Then* $\tilde{\mathcal{E}} = \langle E, \#, \vdash, \emptyset \rangle$ *is a* CES*.*

Consider $\mathcal{E}_7$ in Fig. 7.1, with enablings $a \vdash b$ and $b \Vdash a$. Its only non empty configuration is the set $\{a, b\}$. Indeed, the sequence $\langle a\, b \rangle$ is closed under $\Vdash$ (since a is enabled by the whole sequence) and under $\vdash$ (since b is enabled by its predecessor a). Note that there are no configurations containing only the event a. This is a peculiar difference respect to ES: if C is a finite configuration of a CES, and σe is a sequence for all the events in C closed under $\vdash$ and $\Vdash$, then not necessarily $C \setminus \{e\}$ is a configuration.

To allow for adding an event at time, and reasoning about sets of events which are not configurations, we introduce the auxiliary notion of *X-configuration* in Def. 7.5 below. In an *X-configuration* C, the set C can contain an event e even in the absence of any justification for it (either $\vdash$ or $\Vdash$), provided that e belongs to the set X. We shall say that the events in X have been taken "on credit", to remark the fact that they may have been performed in the absence of a causal justification. Configurations (i.e. $\emptyset$-configurations) represent those sets of events where all the events have found a justification, i.e. where all the credits have been "honoured".

Definition 7.5 (Traces and X-configurations)**.** *Let* $\mathcal{E} = (E, \#, \vdash, \Vdash)$ *be a* CES*, and let* $X \subseteq E$*. A conflict-free sequence* $\sigma = \langle e_0 \dots e_n \rangle \in E^*$ *without repetitions is an* X-trace *of* $\mathcal{E}$ *iff:*

$$\forall i \leq n.\ (e_i \in X \ \ \vee\ \ \overline{\sigma_i} \vdash e_i \ \ \vee\ \ \overline{\sigma} \Vdash e_i) \tag{7.1}$$

For all $C, X \subseteq E$ *we say that* C *is an* X-configuration of $\mathcal{E}$ *iff* $CF(C)$ *and:*

$$\forall e \in C.\ \exists \sigma\ X\text{-trace}.\ e \in \overline{\sigma} \subseteq C \tag{7.2}$$

The set of all X*-traces (resp.* X*-configurations) of* $\mathcal{E}$ *is denoted by* $\mathcal{T}_\mathcal{E}(X)$ *(resp.* $\mathcal{F}_\mathcal{E}(X)$*), abbreviated as* $\mathcal{T}_\mathcal{E}$ *(resp.* $\mathcal{F}_\mathcal{E}$*) when* $X = \emptyset$*.*

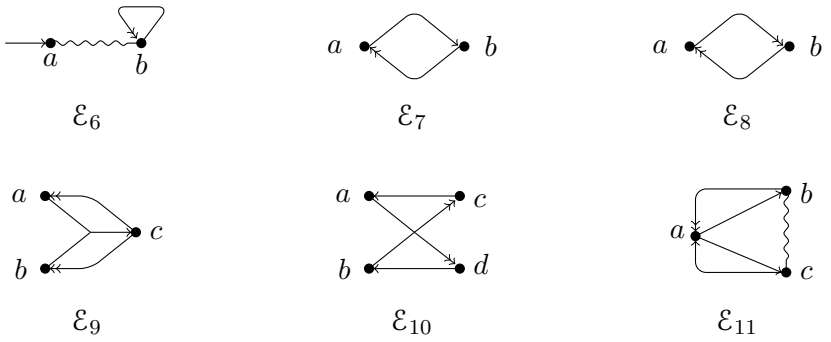


Figure 7.1: Graphical representation of CES.

Notation 7.6. *We adopt the following graphical notation for depicting* CES*: they are denoted as directed hypergraphs, where nodes stand for events. An hyperedge from a set of nodes X to node e denotes an enabling $X \circ e$, where $\circ = \vdash$ if the edge has a single arrow, and $\circ = \Vdash$ if the edge has a double arrow. A conflict $a\#b$ is represented by a waved line between a and b.*

Example 7.7. *Consider the six* CES *in Fig. 7.1.*

- $\mathcal{E}_6$ *has enablings* $\vdash a$, $b \Vdash b$, *and conflict* $a\#b$. *By Def. 7.5,* $\emptyset, \{a\}, \{b\} \in \mathcal{F}_{\mathcal{E}_6}$, *but* $\{a, b\} \notin \mathcal{F}_{\mathcal{E}_6}$.

- $\mathcal{E}_7$ *has enablings* $a \vdash b$ *and* $b \Vdash a$. *Here,* $\emptyset, \{a, b\} \in \mathcal{F}_{\mathcal{E}_7}$, *while neither* $\{a\}$ *nor* $\{b\}$ *belong to* $\mathcal{F}_{\mathcal{E}_7}$. *Furthermore, we have* $\mathcal{F}_{\mathcal{E}_7}(\{b\}) = \{\emptyset, \{b\}, \{a, b\}\}$, *and* $\mathcal{F}_{\mathcal{E}_7}(\{a\}) = \{\emptyset, \{a\}, \{a, b\}\}$.

- $\mathcal{E}_8$ *has enablings* $a \Vdash b$ *and* $b \Vdash a$. *The configurations are the same as in the previous item.*

- $\mathcal{E}_9$ *has enablings* $\{a, b\} \vdash c$, $c \Vdash a$, *and* $c \Vdash b$. *The only non-empty configuration of* $\mathcal{E}_9$ *is* $\{a, b, c\}$.

- $\mathcal{E}_{10}$ *has enablings* $\{a, b\} \Vdash c$, $\{a, b\} \Vdash d$, $c \vdash a$, *and* $d \vdash b$. *We have that* $\{a, b, c, d\} \in \mathcal{F}_{\mathcal{E}_{10}}$. *Note that, were one (or both) of the* $\Vdash$ *turned into a* $\vdash$, *then the only configuration would have been* $\emptyset$.

- $\mathcal{E}_{11}$ *has enablings* $a \vdash b$, $a \vdash c$, $b \Vdash a$ *and* $c \vdash a$, *and conflict* $b\#c$. *We have that* $\{a, b\} \in \mathcal{F}_{\mathcal{E}_{11}}$ *while* $\{a, c\}$ *is not a configuration, but it is an* $\{a\}$*-configuration.*

Example 7.8 (Dining retailers)**.** *Recall the dining retailers scenario from Example 6.15. We formalise the obligations in this scenario as follows. Each retailer* A_i *initially owns* n *pieces of kind* i. *For all* $j \neq i$, *the event* $e_{i,j}$ *models* A_i *giving a piece of cutlery to retailer* A_j. *Retailer* A_i *offers* $n - 1$ *pieces of his cutlery (of kind* i*) in exchange for* $n - 1$ *pieces of cutlery of the other kinds.*

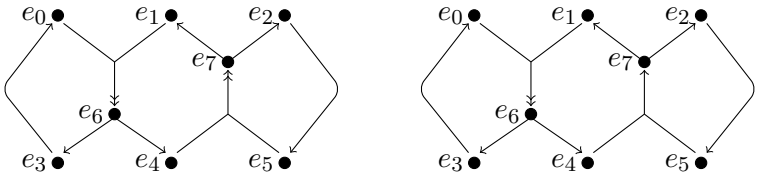


Figure 7.2: Transforming a single $\Vdash$ into a $\vdash$ makes all events unreachable.

The behaviour of retailer A_i *is modelled by the following* $n-1$ *enablings:*

$$\mathsf{A}_i : \qquad \{e_{j,i} \mid j \neq i\} \Vdash e_{i,k} \qquad \text{for all } k \neq i$$

In the CES *containing the enablings of all retailers, the set of events*

$$E \;=\; \{e_{i,j} \mid i,j \in 1..n \text{ and } i \neq j\}$$

is a configuration, hence each retailer eventually eats. Note, instead, that any strict subset of E *(except the empty one) is* not *a configuration. This models the fact that, once the retailers have started exchanging their cutlery, they are committed to continue until everyone eats.*

Example 7.9. *Consider the two* CES $\mathcal{E}_{12}$ *and* $\mathcal{E}_{13}$ *in Fig. 7.2. The only difference is that one of the* $\Vdash$*-enablings in* $\mathcal{E}_{12}$ *has been turned into a* $\vdash$*-enabling in* $\mathcal{E}_{13}$*.*

In $\mathcal{E}_{12}$*, the traces with empty credits have the form*

$$(e_6\ (e_4 \mid e_3\, e_0)) \mid (e_7\ (e_1 \mid e_2\, e_5))$$

where $\mid$ *denotes interleaving, as defined in page 21. Therefore, the set of all events* $E = \{e_i \mid i \in 0..7\}$ *is a configuration in* $\mathcal{F}_{\mathcal{E}_{12}}$*.*

Instead, in $\mathcal{E}_{13}$ *there are no traces with empty credits, except for the empty one. Therefore, the only configuration in* $\mathcal{F}_{\mathcal{E}_{13}}$ *is* $\emptyset$*.* □

Following [Win88], we assume the *axiom of finite causes*, that is, we always require an event to be enabled by a *finite* chain of events. Consider the event structure:

$$\cdots\ e_n \to\ \cdots\ e_3 \to e_2 \to e_1 \to e_0$$

for e_0 to happen, an infinite number of events must have happened *before* it. As in [Win88], we do not consider the set $\{e_i \mid i \geq 0\}$ as a configuration, because a justification of e_0 would require an infinite chain. Similarly, in the CES:

$$a_0 \twoheadleftarrow a_1 \twoheadleftarrow a_2 \twoheadleftarrow a_3\ \cdots\ \twoheadleftarrow a_n\ \cdots$$

where, for a_0 to happen, an infinity of events must happen either *before* or *after* it, the set $\{a_i \mid i \geq 0\}$ is *not* a configuration according to Def. 7.5, because a justification of a_0 would require an infinite chain. This choice is motivated by the following (less abstract) example and formalized in Lemma 7.11.

Example 7.10 (Money lender)**.** *Suppose Bob has an old debt of* €1 *with Alice, but he has no money. Hence he asks Alice to lend him* €1 *to honour his debt. Alice agrees, provided that for this* €1*, Bob will give her back* €2*. When Bob receives the money, he honours his old debt, but now he owes Alice* €2*. Since he has no money, he asks again Alice to lend him* €2*. Alice agrees, provided that Bob will give her back* €3*. Every time he asks Alice to lend him* €i*, Alice requires him to give back* €$(i+1)$*. We can model this scenario as a* CES *with events a_i and b_i (for $i \geq 1$), where a_i represents Alice lending i euros to Bob, and b_i represents Bob giving i euros to Alice. The enablings are $b_{i+1} \Vdash a_i$ and $a_i \vdash b_i$, for all $i \geq 1$. Graphically:*

$$b_1 \leftarrow a_1 \twoheadleftarrow b_2 \leftarrow a_2 \twoheadleftarrow b_3 \cdots$$

An infinite *execution σ^∞ could have the form $\langle a_1\ b_1\ a_2\ b_2\ a_3\ b_3\ a_4\ b_4 \ldots \rangle$. Note that in σ^∞, for each event a_i it is possible to find a finite prefix $\sigma = \langle a_1\, b_1 \ldots a_i\, b_i\, a_{i+1}\, b_{i+1} \rangle$ of σ^∞ which $\Vdash$-enables a_i. However σ is not a trace, because a_{i+1} is not justified: indeed, no finite subsequence σ of σ^∞ allows Bob to honour* all *the debts in σ. In the same spirit of [Win88], Def. 7.5 requires for each event in a configuration a* finite *justification, either in the past or in the future. Accordingly, $\overline{\sigma^\infty}$ is* not *a configuration.*

Lemma 7.11. *Let $C \in \mathcal{F}(X)$. For all $e \in C \setminus X$, there exists $C_0 \subseteq_{fin} C$ such that $C_0 \setminus \{e\} \vdash e$, or $C_0 \Vdash e$.*

Proof. Let $C \in \mathcal{F}(X)$, and let $e \in C \setminus X$. By Def. 7.5, there exists $\sigma = \langle e_0 \ldots e_n \rangle \in \mathcal{T}(X)$ such that $\overline{\sigma} \subseteq C$ and $e_i = e$ for some $i \leq n$. By (7.1), we have that:

$$e \in X \quad \vee \quad \overline{\sigma_i} \vdash e \quad \vee \quad \overline{\sigma} \Vdash e$$

The case $e \in X$ can be excluded by the choice of e. If e has been justified by $\overline{\sigma_i} \vdash e$, then $C_0 = \overline{\sigma_i} = C_0 \setminus \{e\}$ yields the thesis. Otherwise, if e has been justified by $\overline{\sigma} \Vdash e$, then $C_0 = \overline{\sigma}$ yields the thesis. □

7.2 Basic results

In this section we study some basic properties of event structures with circular causality. Unless stated otherwise, in all the statements below in this section we assume a CES $\mathcal{E} = (E, \#, \vdash, \Vdash)$. When clear from the context, we will omit the index $\mathcal{E}$ from $\mathcal{F}_\mathcal{E}(X)$ and $\mathcal{T}_\mathcal{E}(X)$.

7.2.1 Basic results on traces

For all sequences σ, we denote with $\sigma\!\downarrow$ the sequence obtained by eliminating from σ all the duplicate events. This is formalised in Def. 7.12 below.

Definition 7.12 (Removal of duplicates)**.** *For all $\sigma \in E^*$, $e \in E$, we inductively define the sequence $\sigma\!\downarrow$ as follows:*

$$\sigma\!\downarrow = \begin{cases} \varepsilon & \text{if } \sigma = \varepsilon \\ \sigma'\!\downarrow & \text{if } \sigma = \sigma' e \text{ and } e \in \overline{\sigma'} \\ \sigma'\!\downarrow e & \text{if } \sigma = \sigma' e \text{ and } e \notin \overline{\sigma'} \end{cases}$$

In Lemma 7.13 below, we state some basic properties of traces: every trace σ trivially belongs to $\mathcal{T}(\overline{\sigma})$, which intuitively means that we can take every event on credit without worrying about $\vdash$ and $\Vdash$; and the concatenation of two X-traces (modulo eliminating duplicated events) is an X-trace.

Lemma 7.13. *For all $X, Y \subseteq E$, and for all $\sigma, \eta \in E^*$:*

(a) $\sigma \in \mathcal{T}(\overline{\sigma})$

(b) $X \subseteq Y \implies \mathcal{T}(X) \subseteq \mathcal{T}(Y)$

(c) $\sigma \in \mathcal{T}(X) \ \wedge \ \sigma' \in \mathcal{T}(X) \ \wedge \ CF(\overline{\sigma\sigma'}) \implies (\sigma\sigma')\!\downarrow \in \mathcal{T}(X)$

(d) $\sigma \in \mathcal{T}(X) \ \wedge \ \sigma' \in \mathcal{T}(X) \ \wedge \ CF(\overline{\sigma\sigma'}) \implies (\sigma \mid \sigma')\!\downarrow \in \mathcal{T}(X)$

Proof. All the items (a), (b), (c), (d) are direct consequences of (7.1). □

Since every event in a (possibly infinite) configuration is justified by a (finite) trace, for each finite subset C_0 of a configuration we can concatenate the traces of all the events in C_0, and still obtain a trace.

Lemma 7.14. *For all $C, X \subseteq E$:*

$$C \in \mathcal{F}(X) \iff \forall C_0 \subseteq_{fin} C.\ \exists \sigma \in \mathcal{T}(X).\ C_0 \subseteq \overline{\sigma} \subseteq C$$

Proof. ($\Rightarrow$) Let $C \in \mathcal{F}(X)$, and let $C_0 \subseteq_{fin} C$. By Def. 7.5, we see that $CF(C)$, and:

$$\forall e \in C_0.\ \exists \sigma^e \in \mathcal{T}(X).\ \ e \in \overline{\sigma^e} \subseteq C$$

Since C_0 is finite, we can concatenate all the (finite, conflict-free) sequences σ^e obtained above. Let σ' be the result of such operation, and let $\sigma = \sigma'\!\downarrow$. By iterating 7.13(c) $|C_0|$ times, $\sigma \in \mathcal{T}(X)$. Also, by construction we have that $C_0 \subseteq \overline{\sigma} \subseteq C$, from which the thesis follows.

($\Leftarrow$) Assume that $\forall C_0 \subseteq_{fin} C.\ \exists \sigma \in \mathcal{T}(X).\ C_0 \subseteq \overline{\sigma} \subseteq C$, and let $e \in C$. By the hypothesis, since $\{e\} \subseteq C$, there exists $\sigma^e \in \mathcal{T}(X)$ such that $e \in \overline{\sigma^e} \subseteq C$. It remains to prove that $CF(C)$. By contradiction, assume that $\neg CF(C)$. Then, there would exist $C_0 \subseteq_{fin} C$ such that $\neg CF(C_0)$, and so by hypothesis there would also exist some $\sigma \in \mathcal{T}(X)$ with $C_0 \subseteq \overline{\sigma}$. By (7.1), it must be $CF(\overline{\sigma})$, which contradicts $\neg CF(C_0)$. □

The following result is a simple corollary of Lemma 7.14. In case the set C is finite, to test if C is an X-configuration it suffices to find an X-trace which covers all the events of C.

Corollary 7.15. *Let $C \subseteq_{fin} E$ be a finite set of events. We have that $C \in \mathcal{F}(X)$ iff there exists $\sigma \in \mathcal{T}(X)$ such that $\overline{\sigma} = C$.*

If we interpret $\mathcal{T}$ as a function from sets of events to sets of traces, we observe that $\mathcal{T}$ is monotonic, i.e. for each $X \subseteq Y$ we have $\mathcal{T}(X) \subseteq \mathcal{T}(Y)$. Informally, this means that we can arbitrarily enlarge the credit set of a trace. On the contrary, we cannot reduce the credit set while preserving the traces; for instance in the CES $\vdash a, a \vdash b$, we have that $\langle ba \rangle$ is a $\{b\}$- trace but not a $\emptyset$-trace.

For each trace σ there exists a *least* set X such that $\sigma \in \mathcal{T}(X)$. This set is constructed as shown below.

Definition 7.16 (Least credit of a trace)**.** *For all $\sigma = \langle e_0\ e_1 \cdots \rangle$, we define the set of events $\Gamma(\sigma)$ as:*

$$\Gamma(\sigma) \ = \ \{e_i \in \overline{\sigma} \mid \overline{\sigma_i} \nvdash e_i \ \wedge \ \overline{\sigma} \Vdash\!\!\!\!\!/\ e_i\}$$

Lemma 7.17. *Let $\sigma = \langle e_0\ e_1\ \ldots\ e_n \rangle \in E^*$ be a conflict-free sequence without repetitions. Then $\Gamma(\sigma)$ is the* least credit *for σ, i.e. $\sigma \in \mathcal{T}(\Gamma(\sigma))$ and for all Y such that $\sigma \in \mathcal{T}(Y)$, we have $\Gamma(\sigma) \subseteq Y$.*

Proof. Let $\sigma = \langle e_0 \ldots e_n \rangle$, and let $X = \Gamma(\sigma) = \{e_i \in \overline{\sigma} \mid \overline{\sigma_i} \nvdash e_i \ \wedge \ \overline{\sigma} \Vdash\!\!\!\!\!/\ e_i\}$. By Def. 7.5, we have that $\sigma \in \mathcal{T}(X)$. We will prove that $X \subseteq X'$ whenever $\sigma \in \mathcal{T}(X')$. Assume by contradiction that there exists $e \in X$ such that $e \notin X'$. By construction we have that $X \subseteq \overline{\sigma}$, thus there exists i such that $e_i = e$. Since $\sigma \in \mathcal{T}(X')$, by Def. 7.5 it follows that $\overline{\sigma_i} \vdash e_i$ or $\overline{\sigma} \Vdash e_i$, which contradicts the hypothesis. □

Note that $e \notin \Gamma(\sigma)$ iff either e is $\vdash$-enabled by the past events $\overline{\sigma_i}$, or it is $\Vdash$-enabled by the *whole* trace.

The credit on a trace may be redundant, and may be reduced in three ways: by eliminating from it all those events that are not present in the trace; by eliminating all those events that are $\vdash$-enabled by the previous one in the trace; and by eliminating all those events that are $\Vdash$-enabled by the whole trace. For instance let $\mathcal{E} = a \Vdash b, b \vdash a$: the trace $\sigma = \langle ba \rangle$ belongs to $\mathcal{T}(X)$ with $X = \{a, b, c, d\}$. By eliminating from X all those events that are not present in σ, we have that $\sigma \in \mathcal{T}(\{a, b\})$; furthermore we can also eliminate a since $\langle b \rangle \vdash a$ and b since $\overline{\sigma} \Vdash b$, obtaining $\sigma \in \mathcal{T}(\emptyset)$.

Lemma 7.18. *For all $X, Y \subseteq E$, and for all $\sigma = \{e_0 \ldots e_n\} \in E^*$:*

(a) $\sigma \in \mathcal{T}(X \cup Y) \ \wedge \ Y \cap \overline{\sigma} = \emptyset \implies \sigma \ \in \mathcal{T}(X)$

(b) $\sigma \in \mathcal{T}(X \cup e_i) \ \wedge \ \overline{\sigma_i} \vdash e_i \implies \sigma \in \mathcal{T}(X)$

(c) $\sigma \in \mathcal{T}(X \cup Y) \ \wedge \ \overline{\sigma} \Vdash Y \implies \sigma \in \mathcal{T}(X)$

Proof. All the items (a), (a) and (c) are direct consequences of (7.1). □

Let us observe what happens when the last event of an X-trace $\sigma = \sigma' e$ is removed. Assuming that X is the least credit for σ, in general it is not true that X is still the least credit for σ'. It may happen e.g. that e is in X but not in σ', and so e might be removed from the credit set. Furthermore, some events in σ' might require e to be $\Vdash$-enabled. For instance, consider the CES with enablings $e \Vdash e_0, \ldots, e \Vdash e_n$, and let $\sigma = \langle e_0 \ldots e_n\, e\rangle \in \mathcal{T}(\emptyset)$. By removing e from σ we obtain $\sigma' = \langle e_0 \ldots e_n\rangle \in \mathcal{T}(\{e_0, \ldots, e_n\})$, and there exists no $X \subset \{e_0, \ldots, e_n\}$ such that $\sigma' \in \mathcal{T}(X)$.

In Def. 7.19 below we define how the credit set of a trace changes when removing the last event. When the set X in Def. 7.19 is the least credit of σ, then Lemma 7.20 will guarantee that $\Gamma^-(\sigma, X, e)$ is the least credit of σe.

Definition 7.19 (Credits when removing events)**.** *For all $X \subseteq E$, for all $\sigma = \langle e_0 \ldots e_n\rangle$, and for all $e \in E$, we define:*

$$\Gamma^-(\sigma, X, e) \;=\; (X \setminus \{e\}) \;\cup\; \{e_i \in \overline{\sigma} \mid \overline{\sigma e} \Vdash e_i \;\wedge\; \overline{\sigma} \not\Vdash e_i \;\wedge\; \overline{\sigma_i} \not\vdash e_i\} \qquad (7.3)$$

Lemma 7.20. *Let $\sigma = \sigma' e = \langle e_0 \ldots e_n\rangle$, with $e_n = e$. Then,*

$$\sigma \in \mathcal{T}(X) \implies \sigma' \in \mathcal{T}(\Gamma^-(\sigma', X, e))$$

Moreover, if X is the least credit for σ', then $\Gamma^-(\sigma', X, e)$ is least credit for σ.

Proof. Let $\sigma = \sigma' e = \langle e_0 \ldots e_n\rangle \in \mathcal{T}(X)$, with $e_n = e$. Since $\sigma \in \mathcal{T}(X)$, by eq. (7.1) we have that:

$$CF(\overline{\sigma}) \;\wedge\; DF(\sigma) \;\wedge\; \forall i \leq n.\, (e_i \in X \;\vee\; \overline{\sigma_i} \vdash e_i \;\vee\; \overline{\sigma} \Vdash e_i)$$

Since σ' is a prefix of σ then $CF(\overline{\sigma'})$ and $DF(\sigma')$. Moreover, e only occurs in the last position, hence:

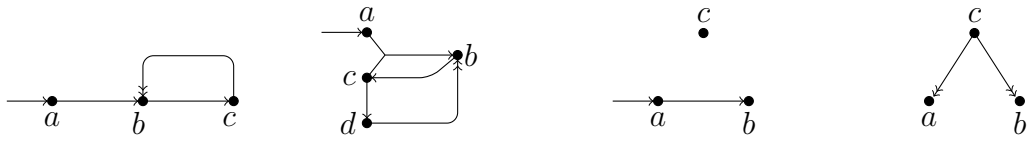
$$\forall i \leq n-1.\, (e_i \in (X \setminus \{e\}) \;\vee\; \overline{\sigma'_i} \vdash e_i \;\vee\; \overline{\sigma} \Vdash e_i) \;\wedge\; (e \in X \;\vee\; \overline{\sigma'} \vdash e \;\vee\; \overline{\sigma} \Vdash e) \quad (7.4)$$

Let us define the set D as:

$$D \;=\; \{e_i \in \overline{\sigma'} \mid \overline{\sigma} \Vdash e_i \;\wedge\; \overline{\sigma'} \not\Vdash e_i \;\wedge\; \overline{\sigma'_i} \not\vdash e_i\} \qquad (7.5)$$

We will prove that $\sigma' \in \mathcal{T}(R)$, with $R = (X \setminus \{e\}) \cup D$. Let $i \leq n-1$. By (7.4), we have three cases:

- $e_i \in X$. Since $e_i \neq e$ for all $i < n$, then $e_i \in X \setminus \{e\} \subseteq R$.
- $\overline{\sigma_i} \vdash e_i$. Since σ' is a prefix of σ, for all $i < n$ it holds that $\overline{\sigma'_i} \vdash e_i$.
- $\overline{\sigma_i} \not\vdash e_i$ and $\overline{\sigma} \Vdash e_i$. If $\overline{\sigma'} \Vdash e_i$, then e_i is justified in σ'. Otherwise, we have $e_i \in D \subseteq R$.



$\mathcal{E}_{14}$ $\mathcal{E}_{15}$ $\mathcal{E}_{16}$ $\mathcal{E}_{17}$

Figure 7.3: Minimal credit when removing events from a trace.

We have then proved that $\sigma' \in \mathcal{T}(R)$.

We now prove that R is a minimal credit for σ'. By contradiction, assume that there exists some $Y \subset R$ such that $\sigma' \in \mathcal{T}(Y)$. Pick an $e_i \in \overline{\sigma'}$ such that $e_i \in R \setminus Y$. By hypothesis, $\sigma' \in \mathcal{T}(Y)$, so it must be:

$$\overline{\sigma'_i} \vdash e_i \quad \vee \quad \overline{\sigma'} \Vdash e_i$$

We have two cases:

- $\overline{\sigma'_i} \vdash e_i$. Since $e_i \in R \setminus Y$, we have two cases: $e_i \in X \setminus \{e\}$ or $e_i \in D$. Note that by (7.5), it cannot be $e_i \in D$. If $e_i \in X \setminus \{e\}$, then we would have $\sigma \in \mathcal{T}(X \setminus \{e_i\})$, which contradicts the hypothesis that X is a minimal credit for σ.

- $\overline{\sigma'} \Vdash e_i$. As above we have two cases: $e_i \in X \setminus \{e\}$ or $e_i \in D$. Note that by (7.5), it cannot be $e_i \in D$. If $e_i \in X \setminus \{e\}$, then we would have $\sigma \in \mathcal{T}(X \setminus \{e_i\})$, which contradicts the hypothesis that X is a minimal credit for σ.

In both cases we have a contradiction; thus R is a minimal credit for σ, and by Lemma 7.17, A is a least credit. □

Example 7.21. *Consider the four* CES *in Fig. 7.3.*

- *In $\mathcal{E}_{14}$, we have $\sigma = \langle abc \rangle \in \mathcal{T}$. Note that, were the requirement $\overline{\sigma'_i} \nvdash e_i$ missing from* (7.3)*, then we would have had $R = \{b\}$, which is not the least credit for $\sigma' = \langle ab \rangle \in \mathcal{T}$.*

- *In $\mathcal{E}_{15}$, we have $\sigma = \langle abcd \rangle \in \mathcal{T}$. Note that if we substitute $\overline{\sigma'} \nvdash e_i$ for $\overline{\sigma'_i} \nvdash e_i$ in* (7.3)*, then we would incorrectly have $R = \emptyset$, while $\sigma' = \langle abc \rangle \in \mathcal{T}(\{b\}) \cap \mathcal{T}(\{c\})$.*

- *In $\mathcal{E}_{16}$, we have $\sigma = \langle abc \rangle \in \mathcal{T}(\{c\})$. Note that, were $\{e\}$ not removed from X in* (7.3)*, then Lemma 7.20 would have predicted that $\{c\}$ is least credit for $\sigma' = \langle ab \rangle$, while $\sigma' \in \mathcal{T}$.*

- *In $\mathcal{E}_{17}$, we have $\sigma = \langle abc \rangle \in \mathcal{T}(\{c\})$ Note that, were $\{e\}$ not removed from X in* (7.3)*, then Lemma 7.20 would have predicted that $\{e, a, b\}$ is least credit for $\sigma' = \langle ab \rangle$, while $\sigma' \in \mathcal{T}(\{a, b\})$.*

In Lemma 7.48 we have seen how the credit set changes when appending an event to a trace. Now we study how the credit changes when appending a whole trace. By Lemma 7.13(c), we know that the concatenation is closed under X-credit; so that the concatenated trace is still an X-trace. Concatenating a trace to another, which provides a justification for some of the events in the credit set, will make the credit set redundant.

Lemma 7.22. *Let $X, Y \subseteq E$, and let $\sigma, \eta \in E^*$ be such that $\eta \in \mathfrak{T}(X \cup Y)$ and $CF(\overline{\eta\,\sigma})$. Then:*

(a) $\sigma \in \mathfrak{T}(X) \;\wedge\; (Y \subseteq \overline{\sigma} \;\vee\; \overline{\sigma} \vdash Y \;\vee\; \overline{\sigma} \Vdash Y) \implies (\sigma\,\eta)\!\downarrow \in \mathfrak{T}(X)$

(b) $\sigma \in \mathfrak{T}(X \cup \overline{\eta}) \;\wedge\; \overline{\sigma} \Vdash Y \implies (\eta\,\sigma)\!\downarrow \in \mathfrak{T}(X)$

Proof. For item (a), let $\sigma = \langle e_0 \dots e_n \rangle$, and let

$$\chi \;=\; (\sigma\,\eta)\!\downarrow \;=\; \sigma\,\langle e_{n+1} \dots e_m \rangle$$

where $\{e_{n+1}, \dots, e_m\} = \overline{\eta} \setminus \overline{\sigma}$. By Def. 7.5 we have to prove that, for all $i \leq m$,

$$e_i \in X \;\vee\; \overline{\chi_i} \vdash e_i \;\vee\; \overline{\chi} \Vdash e_i$$

We have the following two cases:

- $0 \leq i \leq n$. Here we can justify e_i in χ as it has been justified in σ.
- $n < i \leq m$. Here the only relevant case is when e_i has been justified by $e_i \in Y \setminus X$. Indeed, in all the other cases we can justify e_i in χ as it has been justified in η, by noting that $\overline{\chi_i} \supseteq \overline{\eta_{i-n}}$ and that the operator $\downarrow$ preserves the order of events. By hypothesis we have $Y \subseteq \overline{\sigma}$ or $\overline{\sigma} \vdash Y$ or $\overline{\sigma} \Vdash Y$. In the first case, by definition of $\downarrow$ it cannot be the case that $e_i \in Y$ for any $i > n$. In the second case, e_i can be justified by $\overline{\sigma} \vdash Y$; in the third case, e_i can be justified by $\overline{\sigma} \Vdash Y$.

For the item (b), Let $\eta = \langle e_0 \dots e_n \rangle$, and let

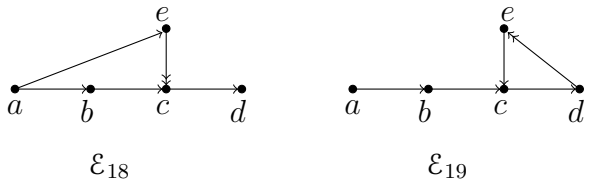
$$\chi \;=\; (\eta\,\sigma)\!\downarrow \;=\; \eta\,\langle e_{n+1} \dots e_m \rangle$$

where $\{e_{n+1}, \dots, e_m\} = \overline{\sigma} \setminus \overline{\eta}$. By Def. 7.5 we have to prove that, for all $i \leq m$,

$$e_i \in X \;\vee\; \overline{\chi_i} \vdash e_i \;\vee\; \overline{\chi} \Vdash e_i$$

We have the following two cases:

- $0 \leq i \leq n$. If $e_i \in Y$, since by hypothesis $\overline{\sigma} \Vdash Y$, then by saturation we have $\overline{\chi} \Vdash Y$. In the other cases we can justify e_i in χ as it has been justified in η.
- $n < i \leq m$. The only relevant case is when e_i has been justified by some events in $\overline{\eta} \setminus X$. However, $e_i \in \overline{\eta}$ cannot happen for any $i > n$, because $\{e_{n+1}, \dots, e_m\} = \overline{\sigma} \setminus \overline{\eta}$. In all the other cases we can justify e_i in χ as it has been justified in σ, by noting that $\overline{\chi_i} \supseteq \overline{\sigma_{i-n}}$ and that the operator $\downarrow$ preserves the order of events.



$\mathcal{E}_{18}$ $\mathcal{E}_{19}$

Figure 7.4: Concatenating traces may reduce the overall credits.

□

Example 7.23. *To illustrate Lemma 7.22, let us consider the two* CES *depicted in Figure 7.4.*

- *For Lemma 7.22*(a)*, let us consider* $\mathcal{E}_{18}$*, and let* $\eta = \langle acd \rangle \in \mathcal{T}(a, c)$*. Then if we choose* $\sigma = \langle ab \rangle \in \mathcal{T}(a)$*, we have that* $\overline{\sigma} \vdash c$ *and so* $\sigma\eta \in \mathcal{T}(a)$*. Otherwise, if we choose* $\sigma = \langle a, e \rangle \in \mathcal{T}(a)$*, we have that* $\overline{\sigma} \Vdash c$ *and so* $\sigma\eta \in \mathcal{T}(a)$*.*
- *For Lemma 7.22*(b)*, let us consider* $\mathcal{E}_{19}$*, and let* $\eta = \langle abce \rangle \in \mathcal{T}(a, e)$*. If we choose* $\sigma = \langle acd \rangle$*, we have that* $\sigma \in \mathcal{T}(\eta)$ *and* $\overline{\sigma} \Vdash e$*; hence,* $\eta\sigma \in \mathcal{T}(a)$*.* □

7.2.2 Basic results on configurations

We now study properties of configurations. Some of them derive immediately from analogous properties of traces. Every set of conflict-free events X is an X-configuration; if $\mathcal{F}$ is interpreted as a function from sets of events to sets of sets of events, then $\mathcal{F}$ is monotonic, i.e. $\mathcal{F}(X) \subseteq \mathcal{F}(Y)$ whenever $X \subseteq Y$.

Lemma 7.24. *For all* $C, C', X, Y \subseteq E$*:*

(a) $CF(X) \implies X \in \mathcal{F}(X)$

(b) $X \subseteq Y \implies \mathcal{F}(X) \subseteq \mathcal{F}(Y)$

Proof. For (a), for all e in X, let $\sigma^e = \langle e \rangle$. It is immediate to check that $\sigma^e \in \mathcal{T}(X)$ and $e \in \overline{\sigma^e} \subseteq X$, and so $X \in \mathcal{F}(X)$.

For (b), since $X \subseteq Y$, then each event e justified with $e \in X$ in equation (7.1) can also be justified with $e \in Y$, hence the thesis. □

Differently from what happens in traces, in general for configurations there exists no *least* set X such that $C \in \mathcal{F}_\mathcal{E}(X)$, as a single configuration may have many *different* minimal credit set. For instance, in a CES $\mathcal{E}$ with enablings $a \vdash b$, $b \vdash a$, we have that $\{a, b\} \in \mathcal{F}_\mathcal{E}(\{a\})$ and $\{a, b\} \in \mathcal{F}_\mathcal{E}(\{b\})$, but $\{a, b\} \notin \mathcal{F}_\mathcal{E}(\emptyset)$. The sets $\{a\}$ and $\{b\}$ are *minimal credits* for $\{a, b\}$, but there exists no least credit.

Definition 7.25 (Configuration minimal credit)**.** *For all* $C, X \subseteq E$ *we say that* X *is* a minimal credit for C *iff* $C \in \mathcal{F}(X)$ *and* $\nexists Y \subset X.\ C \in \mathcal{F}(Y)$*.*

Trivially, if X is a minimal credit for C, then $X \subseteq C$.

Lemma 7.26. *For all $C, X, Y \subseteq E$, we have that $C \in \mathcal{F}(X \cup Y) \ \wedge \ C \Vdash Y \implies C \in \mathcal{F}(X)$.*

Proof. Let $e \in C$. Since $C \in \mathcal{F}(X \cup Y)$, there exists $\eta \in \mathcal{T}(X \cup Y)$ such that $e \in \overline{\eta} \subseteq C$. Since $C \Vdash Y$, using Notation 7.2 there exists a finite subset Z of C such that $Z \Vdash Y$. Since $Z \subseteq_{fin} C \in \mathcal{F}(X \cup Y)$, by Lemma 7.14 there exists $\sigma \in \mathcal{T}(X \cup Y)$ such that $Z \subseteq \overline{\sigma} \subseteq C$. Let $\chi = (\sigma\,\eta)\downarrow$. By Lemma 7.13(c), $\chi \in \mathcal{T}(X \cup Y)$. By saturation, since $Z \Vdash Y$ and $Z \subseteq \overline{\chi}$ we also have that $\overline{\chi} \Vdash Y$. By Corollary c, it follows that $\chi \in \mathcal{T}(X)$. Since $e \in \overline{\chi} \subseteq C$, we have the thesis. □

The following lemma allows for simplifying the credit set when joining two configurations. In item (a), we have an X-configuration C and an $(X \cup C)$-configuration C'. We can then prove that $C \cup C'$ is an X-configuration: intuitively, the events in C' that were taken on credit from C can be justified with the credit set X alone, since $C \in \mathcal{F}(X)$. In item (b), we have an X-configuration C and an $(X \cup Y)$-configuration C' where $C \vdash Y$. We can then prove that $C \cup C'$ is an X-configuration: in a trace, to justify an event e taken on credit from Y, we can posticipate e after the suitable subset of C which entails it. Since $C \in \mathcal{F}(X)$, this only requires to take on credit the set X. Item (c) is similar to the previous one, except that now we deal with a circular enabling. We have an $X \cup C'$-configuration C and an $(X \cup Y)$-configuration C' where $C \Vdash Y$. We can then prove that $C \cup C'$ is an X-configuration: all the events in Y will be justified by C, and since C is justified by C', $C \cup C'$ only requires to take on credit the set X.

Lemma 7.27. *For all $C, C', X, Y \subseteq E$ such that $CF(C \cup C')$:*

(a) $C \in \mathcal{F}(X) \ \wedge \ C' \in \mathcal{F}(X \cup C) \implies C \cup C' \in \mathcal{F}(X)$

(b) $C \in \mathcal{F}(X) \ \wedge \ C' \in \mathcal{F}(X \cup Y) \ \wedge \ C \vdash Y \implies C \cup C' \in \mathcal{F}(X)$

(c) $C \in \mathcal{F}(X \cup C') \ \wedge \ C' \in \mathcal{F}(X \cup Y) \ \wedge \ C \Vdash Y \implies C \cup C' \in \mathcal{F}(X)$

Proof. Assume C, C', X, Y as in the statement.

For item (a), let $e \in C'$. Since $C' \in \mathcal{F}(X \cup C)$, Def. 7.5 prescribes that there exists $\sigma^e = \langle e_0 \dots e_n \rangle \in \mathcal{T}(X \cup C)$ such that $e \in \overline{\sigma^e} \subseteq C'$, i.e.:

$$CF(\overline{\sigma^e}) \ \wedge \ DF(\sigma^e) \ \wedge \ \forall i \leq n.\ (e_i \in X \cup C \ \vee \ \overline{\sigma^e_i} \vdash e_i \ \vee \ \overline{\sigma^e} \Vdash e_i)$$

Let Z_e be the set of e_i in σ^e for which the hypothesis $e_i \in C \setminus X$ has been used, i.e.:

$$Z_e \ = \ \{e_i \in \overline{\sigma^e} \mid e_i \in C \setminus X \ \wedge \ \overline{\sigma^e_i} \nvdash e_i \ \wedge \ \overline{\sigma^e} \nVdash e_i\}$$

Observe that $\sigma^e \in \mathcal{T}(X \cup Z_e)$. Since $Z_e \subseteq_{fin} C \in \mathcal{F}(X)$, by Lemma 7.14 there exists $\eta \in \mathcal{T}(X)$ such that $Z_e \subseteq \overline{\eta} \subseteq C$. Let $\chi = (\eta\,\sigma^e)\downarrow$. Since $\overline{\chi} \subseteq C \cup C'$

and $CF(C \cup C')$ by hypothesis, then $CF(\bar{\chi})$. Since $Z_e \subseteq \overline{\eta}$, Lemma 7.22(a) gives $\chi \in \mathcal{T}(X)$. Since $e \in \overline{\chi} \subseteq C \cup C'$, we have then proved (7.2), from which we conclude that $C \cup C' \in \mathcal{F}(X)$.

For item (b), let $e \in C \cup C'$. We have two cases. If $e \in C$, then the hypothesis $C \in \mathcal{F}(X)$ directly gives a trace which satisfies equation (7.1). So, let $e \in C'$. Since $C' \in \mathcal{F}(X \cup Y)$, Def. 7.5 prescribes that there exists $\sigma^e \in \mathcal{T}(X \cup Y)$ such that $e \in \overline{\sigma^e} \subseteq C'$. Note that $\sigma^e \in \mathcal{T}(X \cup (Y \cap \overline{\sigma^e}))$. Since $C \vdash Y$, then $C \vdash Y \cap \overline{\sigma^e}$. Then, using Notation 7.2 there exists a finite subset Z of C such that $Z \vdash Y \cap \overline{\sigma^e}$. Since $Z \subseteq_{fin} C \in \mathcal{F}(X)$, by Lemma 7.14 there exists $\eta \in \mathcal{T}(X)$ such that $Z \subseteq \overline{\eta} \subseteq C$. By saturation, since $Z \vdash Y \cap \overline{\sigma^e}$ and $Z \subseteq \overline{\eta}$ we also have that $\overline{\eta} \vdash Y \cap \overline{\sigma^e}$. Therefore, Lemma 7.22(a) gives that $(\eta\, \sigma^e)\!\downarrow \in \mathcal{T}(X)$. Since $e \in \overline{(\eta\, \sigma^e)\!\downarrow} \subseteq C \cup C'$, we conclude that $C \cup C' \in \mathcal{F}(X)$.

For item (c), Lemma 7.24(b) yields $C \in \mathcal{F}(X \cup Y \cup C')$. Since $CF(C \cup C')$, by item (a) it follows that $C \cup C' \in \mathcal{F}(X \cup Y)$. Since $C \cup C' \Vdash Y$, by Lemma 7.26 we conclude that $C \cup C' \in \mathcal{F}(X)$. □

7.2.3 Quasi-families of configurations

Analogously to what Winskel does in [Win88], we study the properties of the configurations of a CES. In [Win88], sets of configurations are called $\mathcal{F}$ and we will show that those sets are equivalent to *our* sets of configurations with empty credit $\mathcal{F}$; so we do hope that using the same symbol does not induce confusion.

Recall that coherence, finiteness and coincidence-freeness are properties of a set of sets of events (let call it $\mathcal{F}$) (3.11): coherence means that the union of every pairwise compatible subsets of $\mathcal{F}$ still belongs to $\mathcal{F}$; finiteness means that for each event in $C \in \mathcal{F}$, there always exists a finite subset of C, which belong to $\mathcal{F}$ and which contains the event; and coincidence-freeness means that for every two events in $C \in \mathcal{F}$, there always exists a subset of C, which belong to $\mathcal{F}$ and which contains only one of them.

Definition 7.28 (Quasi-families of configurations)**.** *We say that a set of sets of events $\mathcal{F}$ is a* quasi-family of configurations *iff it satisfies coherence and finiteness.*

A basic result of [Win88] is that the set of configurations of an ES forms a family of configurations. On the contrary, the set of configurations of a CES does not satisfy coincidence-freeness. A counterexample is the CES $\mathcal{E}_7$ in Fig. 7.1, where $\{a, b\} \in \mathcal{F}$, but there exists no configuration including either the single event a or b. Indeed, the absence of coincidence-freeness is a peculiar aspect of circularity: if two events are circularly dependent, each configuration that contains one of them must contain them both

Theorem 7.29. *For all* CES $\mathcal{E}$*, and for all $X \subseteq E$, the set $\mathcal{F}_{\mathcal{E}}(X)$ is a quasi-family of configurations.*

Proof. For coherence, let $\mathcal{A} \subseteq \mathcal{F}(X)$ be pairwise compatible in $\mathcal{F}(X)$. By Def. 3.10:

$$\forall e, e' \in \bigcup \mathcal{A}.\ \exists C \in \mathcal{F}(X).\ e, e' \in C$$

Since $C \in \mathcal{F}(X)$ implies $CF(C)$, it follows that $\neg(e \# e')$ for all $e, e' \in \bigcup \mathcal{A}$, and so $CF(\bigcup \mathcal{A})$. To demonstrate that $\bigcup \mathcal{A} \in \mathcal{F}(X)$, let $e \in \bigcup \mathcal{A}$. Since $\mathcal{A} \in \mathcal{F}(X)$, there exists $C \in \mathcal{A}$ such that $e \in C$. Since $C \in \mathcal{F}(X)$, by Def. 7.5 there exists $\sigma \in \mathcal{T}(X)$ such that $e \in \overline{\sigma} \subseteq_{fin} C$. Since $C \subseteq \bigcup \mathcal{A}$, by Def. 7.5 we can conclude that $\bigcup A \in \mathcal{F}(X)$.

Finiteness is straightforward by Def. 7.5, since for all $e \in C \in \mathcal{F}(X)$, the set of elements of the (finite) sequence $\sigma \in \mathcal{T}(X)$ such that $e \in \overline{\sigma} \subseteq_{fin} C$ is a configuration in $\mathcal{F}(X)$. □

Lemma 7.30 below states that, for all $C \subset C'$ in a family of configurations, one can pass from C to C' by "hops" made of exactly one event. It is essentially the same as Lemma 3.6 in [Win88], except for the different notion of pairwise compatibility used. Note that Lemma 7.30 requires coincidence-freeness, so it is not always true for the sets of configurations generated by CES.

Lemma 7.30. *Let $\mathcal{F}$ be a family of configurations. For all $C, C' \in \mathcal{F}$:*

$$C \subset C' \implies \exists e \in C' \setminus C.\ C \cup \{e\} \in \mathcal{F}$$

Proof. The proof is essentially that of Lemma 1.1.11 in [Win86], the only difference being that pairwise compatibility is used instead of finite completeness. □

Despite faithfully representing the legitimate states of a system where all the credits are honoured, sets of configurations are not as informative as we would like or need. Indeed, they are not able to discriminate among substantially different CES, e.g. like the following:

$$\mathcal{E}:\ \ a \Vdash b,\ \ b \Vdash a \qquad \mathcal{E}':\ \ a \vdash b,\ \ b \Vdash a \qquad \mathcal{E}'':\ \ a \Vdash b,\ \ b \vdash a$$

The sets of X-configurations of $\mathcal{E}, \mathcal{E}', \mathcal{E}''$ coincide, for all X. This contrasts with the different intuitive meaning of $\vdash$ and $\Vdash$, which is revealed instead by observing the traces:

$$\mathcal{T}_{\mathcal{E}} = \{\langle ab \rangle, \langle ba \rangle\} \qquad \mathcal{T}_{\mathcal{E}'} = \{\langle ab \rangle\} \qquad \mathcal{T}_{\mathcal{E}''} = \{\langle ba \rangle\}$$

To substantiate our feeling that configurations alone are not sufficiently discriminating for CES, in Theorem 7.32 we show that for all CES $\mathcal{E}$ there exists a CES $\mathcal{E}'$ without $\vdash$-enablings which has exactly the same configurations of $\mathcal{E}$. Therefore, the meaning of $\vdash$, that is the partial ordering of events, is completely lost by just observing configurations.

Definition 7.31 ($\hat{\mathcal{E}}(\mathcal{F})$)**.** *Let $\mathcal{F}$ be a quasi-family of configurations of a set E. We define the* CES $\hat{\mathcal{E}}(\mathcal{F}) = (E, \#, \emptyset, \Vdash)$ *as follows:*

(a) $e \# e' \iff \forall C \in \mathcal{F}.\ e \notin C \vee e' \notin C$

(b) $X \Vdash e \iff CF(X) \wedge X$ *is finite* $\wedge\, \exists C \in \mathcal{F}.\ e \in C \subseteq X \cup \{e\}$

Theorem 7.32. *For all quasi-families of configurations* $\mathcal{F}$*, we have* $\mathcal{F}_{\hat{\mathcal{E}}(\mathcal{F})} = \mathcal{F}$*.*

Proof. Let $\mathcal{F}$ be a quasi-family of configurations. For ($\subseteq$), let $C \in \mathcal{F}_{\hat{\mathcal{E}}(\mathcal{F})}$. By Def. 7.5 we have $CF(C)$, and for all $e \in C$ there exists C_e such that $e \in C_e \subseteq_{fin} C$, and the elements of C_e can be ordered as a trace in $\mathcal{T}_{\hat{\mathcal{E}}(\mathcal{F})}$. Since $\hat{\mathcal{E}}(\mathcal{F})$ has circular enablings only, it must be $\forall a \in C_e.\ C_e \Vdash a$. Hence, by Def. 7.31(b),

$$\forall a \in C_e.\ \exists D_a \in \mathcal{F}.\ a \in D_a \subseteq C_e \cup \{a\} = C_e$$

Since $\bigcup\{D_a \mid a \in C_e\} = C_e$, the set $\{D_a \mid a \in C_e\}$ is pairwise compatible in $\mathcal{F}$, hence by Theorem 7.29 (coherence) we have that $C_e = \bigcup\{D_a \mid a \in C_e\} \in \mathcal{F}$. Again, the set $\{C_e \mid e \in C\}$ is pairwise compatible in $\mathcal{F}$, therefore by coherence $C = \bigcup\{C_e \mid e \in C\} \in \mathcal{F}$.

For ($\supseteq$), let $C \in \mathcal{F}$. By the definition of conflict in Def. 7.31(a), it must be $CF(C)$. By Theorem 7.29 (finiteness) for all $e \in C$ there exists $C_e \in \mathcal{F}$ such that $e \in C_e \subseteq_{fin} C$. For all $a \in C_e$, we have that $a \in C_e \subseteq C_e \cup \{a\} = C_e$. Thus, by Def. 7.31(b) it follows that $C_e \Vdash a$. Since this holds for all $a \in C_e$, by Def. 7.5 any ordering σ_e of the elements of C_e is a trace in $\mathcal{T}_{\hat{\mathcal{E}}(\mathcal{F})}$. Therefore, for all $e \in C$ we have found a trace $\sigma_e \in \mathcal{T}_{\hat{\mathcal{E}}(\mathcal{F})}$ such that $e \in \overline{\sigma_e} = C_e \subseteq C$. By Def. 7.5, we conclude that $C \in \mathcal{F}_{\hat{\mathcal{E}}(\mathcal{F})}$. □

Remark 7.33. *Theorem 7.32 is the* CES *counterpart of Theorem 3.7 in [Win88], which states that for all families of configurations* $\mathcal{F}$*, there exists an ES* $\mathcal{E}(\mathcal{F})$ *such that* $\mathcal{F} = \mathcal{F}_{\mathcal{E}(\mathcal{F})}$ *(see also Th. 3.14). The definition of* $\mathcal{E}(\mathcal{F})$ *differs from our* $\hat{\mathcal{E}}(\mathcal{F})$ *in Def. 7.31 in three points. First, obviously,* $\vdash$ *is used in item b in place of* $\Vdash$*.*

Second, in item (a) *we say that e is* not *in conflict with* e' *iff there exists* $C \in \mathcal{F}$ *such that* $e, e' \in C$*, while in [Win88] the condition is that* $e \in C \iff e' \in C$ *for some* $C \in \mathcal{F}$*. We argue that the latter definition is not correct, since it implies that no events are in conflict. Indeed, by taking* $C = \emptyset$*, we have that* $e \notin C$ *and* $e' \notin C$ *for all* e, e'*, and so by definition e and* e' *cannot be in conflict.*

Third, our notion of pairwise compatibility differs from Winskel's, as remarked after Def. 3.10.

7.3 Reachable events

Knowing that an event belongs to a configuration, may be useful for contract purposes. For instance, let us consider $\mathcal{E}_7$ in Fig. 7.1, and assume that the event a belongs to Alice. Before performing a, Alice wants to know if her event will ever be honoured; i.e. if there exists a $\emptyset$-configuration which contains it. In this example, the answer is positive since $\{a, b\}$ is a configuration. Now, modify that CES eliminating the $\vdash$ enabling and only keeping the $b \Vdash a$ enabling. Now there does not exist anymore a configuration containing a: in fact a belongs only to a a-configuration and this means that the event a will never be honoured. In this case Alice should consider to not perform her event. Moreover there does not exists any X-configuration for b, which means that the event b will never happen.

We define as reachable events, all those events which belong to a configuration. More precisely, an event is X-reachable when some X-configuration contains it.

Definition 7.34 (Reachable events)**.** *For all $X \subseteq E$, we define:*

$$\mathcal{R}_{\mathcal{E}}(X) = \bigcup \mathcal{F}_{\mathcal{E}}(X)$$

*We say an event e X-*reachable *whenever $e \in \mathcal{R}_{\mathcal{E}}(X)$; we say e* reachable *when $e \in \mathcal{R}_{\mathcal{E}}(\emptyset)$. When clear from the context, we will omit the index $\mathcal{E}$ from $\mathcal{R}_{\mathcal{E}}(X)$.*

Example 7.35. *Consider the* CES *$\mathcal{E}_7$ in Fig. 7.1, with enablings $a \vdash b$ and $b \Vdash a$. Since $\{a, b\}$ is a configuration, then $\mathcal{R}(\emptyset) = \{a, b\}$. Consequently, both a and b are X-reachable for all X.*

Note that there may not exist a least X such that $e \in \mathcal{R}(X)$. For instance, in the CES with enablings $a \vdash b$, $b \vdash a$, we have that a is both $\{a\}$-reachable and $\{b\}$-reachable, but it is not $\emptyset$-reachable.

The function $\mathcal{R}$ enjoys the following basic properties:

Lemma 7.36. *For all $X, Y, C \subseteq E$:*

(a) $X \subseteq \mathcal{R}(X)$

(b) $X \subseteq Y \implies \mathcal{R}(X) \subseteq \mathcal{R}(Y)$

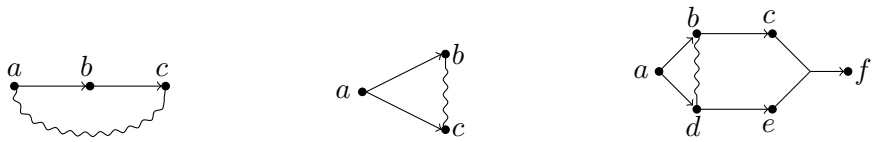
(c) $C \subseteq \mathcal{R}(X) \wedge CF(\mathcal{R}(C \cup X)) \implies \mathcal{R}(X) = \mathcal{R}(C \cup X)$.

Proof. For (a), let $e \in X$. Since $CF(\{e\})$, by Lemma 7.24(a) it follows that $\{e\} \in \mathcal{F}(\{e\})$. By Lemma 7.24(b), we also have $\{e\} \in \mathcal{F}(X)$. Thus, by Def. 7.34 it follows that $e \in \mathcal{R}(X)$.

For (b), let e in $\mathcal{R}(X)$. By Def. 7.34, there exists $C \in \mathcal{F}(X)$ such that $e \in C$. Since $X \subseteq Y$, by Lemma 7.24(b) we also have $C \in \mathcal{F}(Y)$. Hence, $e \in \mathcal{R}(Y)$.

For (c), the inclusion $\subseteq$ follows directly by item (b). For the inclusion $\supseteq$, let $e \in \mathcal{R}(C \cup X)$, and let $C = \{e_i\}_i \subseteq \mathcal{R}(X)$. By Def. 7.34, there exists $C' \in \mathcal{F}(C \cup X)$ such that $e \in C'$, and for all events $e_i \in C$, there exists $C_i \in \mathcal{F}(X)$ such that $e_i \in C_i$. Let $\mathcal{A} = \{C_i \mid e_i \in C\}$, and let $C_j, C_k \in \mathcal{A}$. By Def. 7.34, $C_j, C_k \subseteq \mathcal{R}(X)$. Since $CF(\mathcal{R}(X))$ follows by hypothesis, then $CF(C_j \cup C_k)$, and so by Theorem 7.29, $C_j \cup C_k \in \mathcal{F}(X)$. Thus, by Def. 3.10, the family of X-configurations $\mathcal{A}$ is pairwise compatible. By Theorem 7.29, $F = \bigcup \mathcal{A} \in \mathcal{F}(X)$. Since $F \subseteq \mathcal{R}(X)$ and $C' \subseteq \mathcal{R}(C \cup X)$, then $F \cup C' \subseteq \mathcal{R}(C \cup X)$, and thus $CF(F \cup C')$ follows by the premise of (c). By Lemma 7.24(b), since $C' \in \mathcal{F}(C \cup X)$ and $C \subseteq F$, then $C' \in \mathcal{F}(F \cup X)$. Therefore, by Lemma 7.27(a), $e \in F \cup C' \in \mathcal{F}(X)$, and so $e \in \mathcal{R}(X)$. □

Example 7.37. *Consider the three* CES *in Fig. 7.5.*



$\mathcal{E}_{20}$ $\mathcal{E}_{21}$ $\mathcal{E}_{22}$

Figure 7.5: Conflicts and reachable events.

(1) Note that if we weaken the conflict-freeness requirement in Lemma 7.36(c)*, and only require that $CF(C \cup X)$, then the thesis in Lemma 7.36*(c) *would not hold. Take e.g. the* CES $\mathcal{E}_{20}$*, and let $C = \{a, b\}$ and $X = \{a\}$. We have that $\mathcal{R}(X) = \{a, b\}$ and $\mathcal{R}(C \cup X) = \{a, b, c\}$, which is not conflict-free. Hence Lemma 7.36*(c) *does not apply, and in fact $\mathcal{R}(X) \neq \mathcal{R}(C \cup X)$.*

(2) In $\mathcal{E}_{21}$, we have that $\mathcal{R}(\{a\}) = \{a, b, c\}$. In this case we see that the conflict-freeness requirement in Lemma 7.36(c) *is sufficient but not necessary, since $\{a, b, c\}$ is* not *conflict-free, but nevertheless $\mathcal{R}(\{a\}) = \mathcal{R}(\mathcal{R}(\{a\})) = \{a, b, c\}$.*

(3) In $\mathcal{E}_{22}$, $\mathcal{R}(\{a\}) = \{a, b, c, d, e\}$ is not *conflict-free. Then, Lemma 7.36*(c) *does not apply and, indeed $\mathcal{R}(\{a\} \cup \{c, e\}) = \{a, b, c, d, e, f\} \supsetneq \mathcal{R}(\{a\})$.* □

7.3.1 Reachability for conflict-free CES

For conflict-free CES, we can inductively characterize the reachable events. This is done in Def. 7.38 in the form of inference rules.

Unless stated otherwise, in the rest of this section we assume every CES is conflict-free.

Definition 7.38 (Reachable events for conflict-free CES)**.** *For all $X \subseteq E$, we inductively define the set $\hat{\mathcal{R}}(X)$ as follows:*

$$\frac{e \in X}{e \in \hat{\mathcal{R}}(X)}\ (\in_{\hat{\mathcal{R}}}) \qquad \frac{\hat{\mathcal{R}}(X) \vdash e}{e \in \hat{\mathcal{R}}(X)}\ (\vdash_{\hat{\mathcal{R}}}) \qquad \frac{\hat{\mathcal{R}}(X \cup \{e\}) \Vdash e}{e \in \hat{\mathcal{R}}(X)}\ (\Vdash_{\hat{\mathcal{R}}})$$

Recall that, by saturation of $\vdash$ and by Notation 7.2, the premise $\hat{\mathcal{R}}(X) \vdash e$ in rule $\vdash_{\hat{\mathcal{R}}}$ actually means that there exists a *finite* set of events $e_1, \ldots, e_n \in \hat{\mathcal{R}}(X)$ such that $\{e_1, \ldots, e_n\} \vdash e$ (similarly for $\Vdash_{\hat{\mathcal{R}}}$). More pedantically, rule $\vdash_{\hat{\mathcal{R}}}$ actually stands for the set of rules:

$$\frac{e_1 \in \hat{\mathcal{R}}(X) \quad \cdots \quad e_n \in \hat{\mathcal{R}}(X)}{e \in \hat{\mathcal{R}}(X)} \text{ if } \{e_1, \ldots, e_n\} \vdash e$$

Example 7.39. *Consider the* CES $\mathcal{E}_7$ *of Fig.7.1. We have the following derivation:*

$$\dfrac{b \Vdash a \qquad \dfrac{a \vdash b \qquad \dfrac{\overline{a \in \{a\}}}{a \in \hat{\mathcal{R}}(\{a\})}\,(\in_{\hat{\mathcal{R}}})}{b \in \hat{\mathcal{R}}(\{a\})}\,(\Vdash_{\hat{\mathcal{R}}})}{a \in \hat{\mathcal{R}}(\emptyset)}\,(\vdash_{\hat{\mathcal{R}}})$$

then, since $a \in \hat{\mathcal{R}}(\emptyset)$*, from rule* $\vdash_{\hat{\mathcal{R}}}$ *we also obtain* $b \in \hat{\mathcal{R}}(\emptyset)$*.* □

To give a rough intuition of the characterization in Def 7.38, one can say that every event has the same *reachability degree* X of its enabling events (also said generators sets), since in absence of conflicts, the chain of generators can stay together in an X-configuration. This consideration obviously does not hold if the CES has conflicts. In general, one cannot say recursively that an event has the same degree of its generators, because somewhere in the chain of enabling there may exist a conflict which prevents the complete chain from staying in the same configuration.

Example 7.40. *Consider the* CES $\mathcal{E}_{14}$ *in Fig.7.3. By rule* $\vdash_{\hat{\mathcal{R}}}$*, the event* c *is has the same reachability degree of* b*, since* b *is its only way to be enabled. Indeed, if there exists an* X*-configuration containing* b*, it will contain also* c *without augmenting the credit set, and their reachability degree will be* X*.*

The event b *has two ways for being enabled, hence it has two sets of generators; so it has both the same degree of* a *and* c*. Since* a *is* $\emptyset$*-reachable, then also* b *is* $\emptyset$*-reachable and since* c *has the same degree of* b *then also* c *is* $\emptyset$*-reachable.*

Now consider the other way to calculate the reachability of b*. By following the* $\Vdash$*-enabling of* b *we have that,* b *has the same reachability degree of* c*, and that — again —* c *is as the degree of* b*. This seems to be stuck, but recall that the rule* $\Vdash_{\hat{\mathcal{R}}}$ *says that we may discharge a* $\Vdash$*-enabled event from its own the degree, if we encounter it in a looping derivation.*

To conclude a, b *and* c *are all* $\emptyset$*-reachable.* □

The following lemma summarizes some basic properties of the operator $\hat{\mathcal{R}}$.

Lemma 7.41. *For all* $X, Y \subseteq E$*:*

(a) $X \subseteq \hat{\mathcal{R}}(X)$

(b) $X \subseteq Y \implies \hat{\mathcal{R}}(X) \subseteq \hat{\mathcal{R}}(Y)$.

(c) $\hat{\mathcal{R}}(\hat{\mathcal{R}}(X)) = \hat{\mathcal{R}}(X)$

Proof. Item (a) is straightforward by rule $(\in_{\hat{\mathcal{R}}})$. Item (b) is by induction on the depth of the derivation of $e \in \hat{\mathcal{R}}(X)$. For item (c), the inclusion $\hat{\mathcal{R}}(X) \subseteq \hat{\mathcal{R}}(\hat{\mathcal{R}}(X))$ follows by item (a). The other inclusion can be easily proved by induction on the depth of the proof of $e \in \hat{\mathcal{R}}(\hat{\mathcal{R}}(X))$. □

Lemma 7.42. *For all* $X, Y \subseteq E$, $Y \subseteq \hat{\mathcal{R}}(X) \implies \hat{\mathcal{R}}(X \cup Y) = \hat{\mathcal{R}}(X)$.

Proof. The inclusion $\hat{\mathcal{R}}(X \cup Y) \supseteq \hat{\mathcal{R}}(X)$ follows directly by Lemma 7.41(b). For the other inclusion, by Lemma 7.41(b) and 7.41(c) we have:

$$Y \subseteq \hat{\mathcal{R}}(X) \implies \hat{\mathcal{R}}(Y) \subseteq \hat{\mathcal{R}}(\hat{\mathcal{R}}(X)) = \hat{\mathcal{R}}(X)$$

By Lemma 7.41(a), $Y \subseteq \hat{\mathcal{R}}(Y)$. Since $\hat{\mathcal{R}}(Y) \subseteq \hat{\mathcal{R}}(X)$, then by Lemma 7.41(b) we have $\hat{\mathcal{R}}(X \cup Y) \subseteq \hat{\mathcal{R}}(X \cup \hat{\mathcal{R}}(X))$. By Lemma 7.41(a), $X \subseteq \hat{\mathcal{R}}(X)$, and so $\hat{\mathcal{R}}(X \cup \hat{\mathcal{R}}(X)) = \hat{\mathcal{R}}(\hat{\mathcal{R}}(X))$. By Lemma 7.41(c), we have $\hat{\mathcal{R}}(\hat{\mathcal{R}}(X)) = \hat{\mathcal{R}}(X)$, which concludes. □

Lemma 7.43. *For all* $X \subseteq E$, $Y \subseteq_{fin} E$, $\hat{\mathcal{R}}(X \cup Y) \Vdash Y \implies Y \subseteq \hat{\mathcal{R}}(X)$.

Proof. If $Y = \emptyset$, the statement holds trivially. Otherwise, let $Y = \{e_0, \ldots, e_k\}$. For all $i \leq k$, we define $Y_i = \{e_0, \ldots, e_i\}$, and $Y^i = Y \setminus Y_i = \{e_{i+1}, \ldots, e_k\}$. We shall prove that:

$$\forall i \leq k.\ Y_i \subseteq \hat{\mathcal{R}}(X \cup Y^i) \tag{7.6}$$

To prove (7.6), we proceed by mathematical induction.

- Base case $i = 0$.

  By hypothesis, $\hat{\mathcal{R}}(X \cup Y) \Vdash e_0$. Then:

  $$\frac{\hat{\mathcal{R}}(X \cup Y) \Vdash e_0}{e_0 \in \hat{\mathcal{R}}((X \cup Y) \setminus \{e_0\})}\ (\Vdash_{\hat{\mathcal{R}}})$$

  So we have proved that $\{e_0\} = Y_0 \subseteq \hat{\mathcal{R}}((X \cup Y) \setminus \{e_0\}) \subseteq \hat{\mathcal{R}}(X \cup Y^0)$.

- Inductive case. By the induction hypothesis, we have that $Y_i \subseteq \hat{\mathcal{R}}(X \cup Y^i)$. By Lemma 7.42, $\hat{\mathcal{R}}(X \cup Y^i \cup Y_i) = \hat{\mathcal{R}}(X \cup Y^i)$. By $(\in_{\mathcal{R}})$, we have $e_{i+1} \in \hat{\mathcal{R}}(X \cup Y) = \hat{\mathcal{R}}(X \cup Y^i \cup Y_i) = \hat{\mathcal{R}}(X \cup Y^i)$. By hypothesis, $\hat{\mathcal{R}}(X \cup Y^i \cup Y_i) \Vdash e_{i+1}$, and so $\hat{\mathcal{R}}(X \cup Y^i) \Vdash e_{i+1}$. Hence we can apply rule $(\Vdash_{\hat{\mathcal{R}}})$ to obtain:

  $$\frac{\hat{\mathcal{R}}(X \cup Y^i) \Vdash e_{i+1}}{e_{i+1} \in \hat{\mathcal{R}}(X \cup Y^{i+1})}\ (\Vdash_{\hat{\mathcal{R}}}) \tag{7.7}$$

  We obtain the thesis of (7.6) as follows:

  $$\begin{aligned}
  Y_{i+1} &= Y_i \cup \{e_{i+1}\} && \text{(by Def. } Y_i\text{)} \\
  &\subseteq \hat{\mathcal{R}}(X \cup Y^i) \cup \{e_{i+1}\} && \text{(by the induction hypothesis)} \\
  &= \hat{\mathcal{R}}(X \cup Y^i) && \text{(by } e_{i+1} \in \hat{\mathcal{R}}(X \cup Y^i)\text{)} \\
  &= \hat{\mathcal{R}}(X \cup Y^{i+1} \cup \{e_{i+1}\}) && \text{(by Def. } Y^i\text{)} \\
  &= \hat{\mathcal{R}}(X \cup Y^{i+1}) && \text{(by Lemma 7.42 and (7.7))}
  \end{aligned}$$

Back to the main statement, just note that for $i = k$ in (7.6), we obtain the thesis $Y = Y_k \subseteq \hat{\mathcal{R}}(X)$. □

Lemma 7.44. *For all* $C, X \subseteq E,\ C \in \mathcal{F}(X) \implies C \subseteq \hat{\mathcal{R}}(X)$.

Proof. We will first prove that $\forall C_0 \subseteq_{fin} C.\ \forall X.\ C_0 \in \mathcal{F}(X) \implies C_0 \subseteq \hat{\mathcal{R}}(X)$. Let $C_0 \subseteq_{fin} C$, and assume that $C_0 \in \mathcal{F}(X)$, for some X. By Corollary 7.15 we have that:

$$\exists \sigma = \langle e_0 \dots e_n \rangle \in \mathcal{T}(X).\ \overline{\sigma} = C_0$$

We proceed by induction on the size of C_0. In the base case $C_0 = \emptyset$ the thesis holds trivially. For the inductive case, let us assume $C_0 \neq \emptyset$. Let $e_n = e$, let $\sigma' = \langle e_0 \dots e_{n-1} \rangle$ and let $C' = \overline{\sigma'}$. We will prove that $C' \subseteq \hat{\mathcal{R}}(X)$ and $e \subseteq \hat{\mathcal{R}}(X)$. Let:

$$D \ = \ \{e_i \in \overline{\sigma'} \mid \overline{\sigma} \Vdash e_i \ \wedge \ \overline{\sigma'} \not\Vdash e_i \ \wedge \ \overline{\sigma'_i} \not\vdash e_i\} \tag{7.8}$$

By Lemma 7.20, $\sigma' \in \mathcal{T}(X \cup D)$. By Lemma 7.15, $C' \in \mathcal{F}(X \cup D)$, and then by the induction hypothesis, $C' \subseteq \hat{\mathcal{R}}(X \cup D)$. Now, we will prove that $e \in \hat{\mathcal{R}}(X \cup D)$. Since $e = e_n \in \overline{\sigma}$, by eq. (7.1), to justify e in σ we must have:

$$e \in X \ \vee \ \overline{\sigma'} \vdash e \ \vee \ \overline{\sigma} \Vdash e$$

We have the following three cases:

- if $e \in X$, by $(\in_{\hat{\mathcal{R}}})$ we have that $e \in \hat{\mathcal{R}}(X)$ and by Lemma 7.24(b), $e \in \hat{\mathcal{R}}(X \cup D)$.
- if $\overline{\sigma'} \vdash e$, since by the induction hypothesis $C' \subseteq \hat{\mathcal{R}}(X \cup D)$ and $C' \vdash e$, then by saturation $\hat{\mathcal{R}}(X \cup D) \vdash e$. Therefore by $(\vdash_{\hat{\mathcal{R}}})$ we have:

$$\frac{\hat{\mathcal{R}}(X \cup D) \vdash e}{e \in \hat{\mathcal{R}}(X \cup D)} \ (\vdash_{\hat{\mathcal{R}}})$$

- if $\overline{\sigma} \Vdash e$, since by the induction hypothesis $C' \subseteq \hat{\mathcal{R}}(X \cup D)$, then by Lemma 7.24(b), $C_0 = C' \cup \{e\} \subseteq \hat{\mathcal{R}}(X \cup D \cup \{e\})$. By saturation, $\overline{\sigma} = C_0 \Vdash e$ implies $\hat{\mathcal{R}}(X \cup D \cup \{e\}) \Vdash e$. Therefore, by $(\Vdash_{\hat{\mathcal{R}}})$ we have:

$$\frac{\hat{\mathcal{R}}(X \cup D \cup \{e\}) \Vdash e}{e \in \hat{\mathcal{R}}(X \cup D)} \ (\Vdash_{\hat{\mathcal{R}}})$$

So we have proved that $e \in \hat{\mathcal{R}}(X \cup D)$, hence $C_0 = C' \cup \{e\} \subseteq \hat{\mathcal{R}}(X \cup D)$. Note that by (7.8), we have that $C_0 \Vdash D$, and so by saturation $\hat{\mathcal{R}}(X \cup D) \Vdash D$. Therefore by Lemma 7.43, $D \subseteq \hat{\mathcal{R}}(X)$. By Lemma 7.42, it follows that $\hat{\mathcal{R}}(X \cup D) = \hat{\mathcal{R}}(X)$, and the thesis follows because $C_0 \subseteq \hat{\mathcal{R}}(X \cup D)$.

Back to the main statement, since $C \in \mathcal{F}(X)$ we have that for all $e \in C$, there exists $\sigma^e \in \mathcal{T}(X)$ such that $e \in \overline{\sigma^e} \subseteq_{fin} C$. Since $\overline{\sigma^e} \in \mathcal{F}(X)$, we have proved above that $\overline{\sigma^e} \subseteq \hat{\mathcal{R}}(X)$. Therefore, $C = \bigcup\{\overline{\sigma^e} \mid e \in C\} \subseteq \hat{\mathcal{R}}(X)$. □

Finally, we can prove that the new characterization for reachable events is equivalent to the original one 7.34 in the case of a conflict-free CES.

Theorem 7.45. *For all $X \subseteq E$:*

$$\mathcal{R}(X) \;=\; \hat{\mathcal{R}}(X)$$

Proof. ($\subseteq$) Let $e \in \mathcal{R}(X)$. By Def. 7.34, there exists a configuration $C \in \mathcal{F}(X)$ such that $e \in C$. By Lemma 7.44, $e \in C \subseteq \hat{\mathcal{R}}(X)$.

($\supseteq$) Assume that $e \in \hat{\mathcal{R}}(X)$. We will prove that $\exists C \in \mathcal{F}(X)$ such that $e \in C$. By Def. 7.34, this will allow to conclude $e \in \mathcal{R}(X)$. We proceed by induction on the depth of the derivation of $e \in \hat{\mathcal{R}}(X)$. According to the last rule used in the derivation, we have the following three cases:

- case $(\in_{\hat{\mathcal{R}}})$. We have that
$$\frac{e \in X}{e \in \hat{\mathcal{R}}(X)}\ (\in_{\hat{\mathcal{R}}})$$
Since $e \in X$, by Lemma 7.24 we have that $e \in \{e\} \in \mathcal{F}(X)$.

- case $(\vdash_{\hat{\mathcal{R}}})$.
$$\frac{\hat{\mathcal{R}}(X) \vdash e}{e \in \hat{\mathcal{R}}(X)}\ (\vdash_{\hat{\mathcal{R}}})$$
The premise $\hat{\mathcal{R}}(X) \vdash e$ implies that there exists $D \subseteq_{fin} \hat{\mathcal{R}}(X)$ such that $D \vdash e$. By the induction hypothesis, for all $d \in D$ there exists $C_d \in \mathcal{F}(X)$ such that $d \in C_d$. Let $C = \bigcup_{d \in D} C_d$. Since $\mathcal{E}$ is conflict-free, by Theorem 7.29 it follows that $C \in \mathcal{F}(X)$. Since $D \subseteq C$ and $D \vdash e$, by saturation we have $C \vdash e$. By Lemma 7.24(a), $\{e\} \in \mathcal{F}(\{e\})$. Therefore, by Lemma 7.27(b), $C \cup \{e\} \in \mathcal{F}(X)$.

- case $[\Vdash_{\hat{\mathcal{R}}}]$
$$\frac{\hat{\mathcal{R}}(X \cup \{e\}) \Vdash e}{e \in \hat{\mathcal{R}}(X)}\ [\Vdash_{\hat{\mathcal{R}}}]$$
The premise $\hat{\mathcal{R}}(X \cup \{e\}) \Vdash e$ implies that there exists $D \subseteq_{fin} \hat{\mathcal{R}}(X \cup \{e\})$ such that $D \Vdash e$. By the induction hypothesis, for all $d \in D$ there exists $C_d \in \mathcal{F}(X \cup \{e\})$ such that $d \in C_d$. Let $C = \bigcup_{d \in D} C_d$. Since $\mathcal{E}$ is conflict-free, by Theorem 7.29 it follows that $C \in \mathcal{F}(X \cup \{e\})$. By Lemma 7.24(a), $\{e\} \in \mathcal{F}(X \cup \{e\})$. Since $D \subseteq C$ and $D \Vdash e$, by saturation we have $C \Vdash e$. Therefore, Lemma 7.27(c) gives that $C \cup \{e\} \in \mathcal{F}(X)$.

□

7.4 An LTS semantics of CES

Similarly to what has been done for event structures, we now define a Labelled Transition System (LTS) for CES.

In the LTS of events structures (Def. 3.7), an event can be fired if there are no conflicts with the previously occurred events and if it is enabled by them. With

these constraints, every state of the LTS is a configuration and every path to a state is a sequence closed under $\vdash$-enabling.

In CES, an event can always be fired: if it is not already $\vdash$ nor $\Vdash$-enabled, it may be inserted in the credit set and still obtain some X-configuration. Note that the credit set may grow up but it grows smaller by firing some event that $\Vdash$-enables some of the previously taken-on-credit occurred events.

Accordingly, every state of the LTS of a CES must remember not only the already occurred events but also the credit set accumulated so far: so every state is *a pair* (C, X) where C is an X-configuration and X is the least credit for C.

7.4.1 Adding events to a trace

We will first observe what happens when adding an event e to a trace $\sigma \in \mathcal{T}(X)$. It is always true that $\sigma e \in \mathcal{T}(X \cup \{e\})$, although $X \cup \{e\}$ may *not* be the least credit for σe.

Def. 7.46 below establishes how the credits of a trace change when adding an event. When Def. 7.46 is instantiated with $C = \overline{\sigma}$ and X is the least credit of σ, then Lemma 7.48 guarantees that $\Gamma^+(C, X, e)$ is the least credit for σe. Intuitively, in Def. 7.46 we first remove from X the set of credits which have been honoured by performing e; then, we add e unless it is justified.

Definition 7.46 (Credits when adding events)**.** *For all $C, X \subseteq E$ and for all $e \in E$, we define:*

$$\Gamma^+(C, X, e) \;=\; (X \setminus \{x \in X \mid C \cup \{e\} \Vdash x\}) \cup \begin{cases} \{e\} & \text{if } C \cup \{e\} \not\Vdash e \;\wedge\; C \not\vdash e \\ \emptyset & \text{otherwise} \end{cases}$$

Example 7.47. *Consider the* CES $\mathcal{E}_1$, $\mathcal{E}_2$, $\mathcal{E}_7$ *and* $\mathcal{E}_8$ *in Figures 3.1 and 7.1. For the maximal traces $\langle ab \rangle$, $\langle ba \rangle$, we have the following computations:*

$\mathcal{E}_1 :\ (\emptyset,\emptyset) \xrightarrow{a} (\{a\},\emptyset) \xrightarrow{b} (\{a,b\},\emptyset)$, $\qquad (\emptyset,\emptyset) \xrightarrow{b} (\{b\},\{b\}) \xrightarrow{a} (\{a,b\},\{b\})$.

$\mathcal{E}_2 :\ (\emptyset,\emptyset) \xrightarrow{a} (\{a\},\{a\}) \xrightarrow{b} (\{a,b\},\{a\})$, $\qquad (\emptyset,\emptyset) \xrightarrow{b} (\{b\},\{b\}) \xrightarrow{a} (\{a,b\},\{b\})$.

$\mathcal{E}_7 :\ (\emptyset,\emptyset) \xrightarrow{a} (\{a\},\{a\}) \xrightarrow{b} (\{a,b\},\emptyset)$, $\qquad (\emptyset,\emptyset) \xrightarrow{b} (\{b\},\{b\}) \xrightarrow{a} (\{a,b\},\{b\})$.

$\mathcal{E}_8 :\ (\emptyset,\emptyset) \xrightarrow{a} (\{a\},\{a\}) \xrightarrow{b} (\{a,b\},\emptyset)$, $\qquad (\emptyset,\emptyset) \xrightarrow{b} (\{b\},\{b\}) \xrightarrow{a} (\{a,b\},\emptyset)$.

In particular, for $\mathcal{E}_7$ the trace $\langle a \rangle$ belongs to $\mathcal{T}(\{a\})$, and $\{a\}$ is its least credit. Of course $\langle a\, b \rangle \in \mathcal{T}(\{a,b\})$, but since b is $\vdash$-enabled by $\{a\}$, it is not necessary to take b on credit. Moreover, since $b \Vdash a$, we can also remove a from the credit set: hence $\langle a\, b \rangle \in \mathcal{T}$. Indeed, the least credit for $\langle a\, b \rangle$ is computed through Def. 7.46 as $\Gamma^+(\{a\},\{a\},b) = \{a\} \setminus \{a\} \cup \emptyset = \emptyset$.

Lemma 7.48. *Let $\sigma = \sigma' e$, with $e \notin \overline{\sigma'}$ and $CF(\overline{\sigma})$. If $\sigma' \in \mathcal{T}(X)$, then $\sigma \in \mathcal{T}(\Gamma^+(\overline{\sigma'}, X, e)))$. Moreover, if X is the least credit for σ', then $\Gamma^+(\overline{\sigma'}, X, e)$ is the least credit for σ.*

Proof. Let $\sigma' = \langle e_0 \dots e_n \rangle \in \mathcal{T}(X')$, let $\sigma = \sigma' e$, and let $X = \Gamma^+(\overline{\sigma'}, X', e)$. By equation (7.1) it follows that $\sigma \in \mathcal{T}(X)$. Let X' be the least credit for σ'. We prove that X is the least credit for σ. By Def. 7.48:

$$X \;=\; \Gamma^+(\overline{\sigma'}, X', e) = (X' \setminus \{x \in X' \mid \overline{\sigma'} \cup \{e\} \Vdash x\}) \cup \begin{cases} \{e\} & \text{if } \overline{\sigma'} \cup \{e\} \not\Vdash e \;\wedge\; \overline{\sigma'} \not\vdash e \\ \emptyset & \text{otherwise} \end{cases}$$

By Lemma 7.17, $X' = \{e_i \in \overline{\sigma'} \mid \overline{\sigma'_i} \not\vdash e_i \;\wedge\; \overline{\sigma'} \not\Vdash e_i\}$. Since $\overline{\sigma'} \cup \{e\} = \overline{\sigma}$, we have:

$$X \;=\; (\{e_i \in \overline{\sigma'} \mid \overline{\sigma'_i} \not\vdash e_i \;\wedge\; \overline{\sigma'} \not\Vdash e_i\} \setminus \{x \in X' \mid \overline{\sigma} \Vdash x\}) \cup \begin{cases} \{e\} & \text{if } \overline{\sigma} \not\Vdash e \;\wedge\; \overline{\sigma'} \not\vdash e \\ \emptyset & \text{otherwise} \end{cases}$$

Since X' is the least credit for σ', we have that $X' \subseteq \overline{\sigma'}$. Hence:

$$X \;=\; (\{e_i \in \overline{\sigma'} \mid \overline{\sigma'_i} \not\vdash e_i \;\wedge\; \overline{\sigma} \not\Vdash e_i\} \;\cup\; \begin{cases} \{e\} & \text{if } \overline{\sigma} \not\Vdash e \;\wedge\; \overline{\sigma'} \not\vdash e \\ \emptyset & \text{otherwise} \end{cases}$$

By renaming e as e_{n+1}, we have $\sigma = \langle e_0 \dots e_n e_{n+1} \rangle$, hence

$$X = (\{e_i \in \overline{\sigma} \mid \overline{\sigma_i} \not\vdash e_i \;\wedge\; \overline{\sigma} \not\Vdash e_i\}$$

By Lemma 7.17, X is the least credit for σ. □

As noticed in Example 7.47, adding events to a trace may reduce the credit set. Also, observe that changing the order in which events are performed may change the credit set. In particular when an event without $\Vdash$-enablings is fired before its $\vdash$-justification, it will not be possible to remove it from the credit set by firing new events. For instance, if in Example 7.47 we fire b before a, then we cannot remove b from the credit set. This allows for correctly recording the events performed in the absence of a causal justification.

Example 7.49. *Consider the* CES $\mathcal{E}_{17}$ *in Fig. 7.3. The trace $\langle ab \rangle$ has least credit $\{a, b\}$; by adding the event c, the least credit for $\langle abc \rangle$ becomes $\{c\}$.*

7.4.2 LTS of a CES

We will now formally define our LTS.

Definition 7.50 (LTS of a CES)**.** *Given a* CES $\mathcal{E}$*, we define its labelled transition system* $\mathrm{LTS}_{\mathcal{E}} = \langle S, E, \rightarrow_{\mathcal{E}} \rangle$*, where $S = Con \times Con$, and the relation $\rightarrow_{\mathcal{E}}$ is defined as follows:*

$$\frac{e \notin C \qquad CF(C \cup \{e\})}{(C, X) \xrightarrow{e}_{\mathcal{E}} (C \cup \{e\},\, \Gamma^+(C, X, e))}$$

We say that (C, X) is a reachable state of $\mathrm{LTS}_{\mathcal{E}}$ *iff $(\emptyset, \emptyset) \rightarrow^*_{\mathcal{E}} (C, X)$. When clear from the context, we will omit the index $\mathcal{E}$ from $\rightarrow_{\mathcal{E}}$.*

We remark that if $\mathcal{E}$ has no circular enablings, then $\mathrm{LTS}_\mathcal{E}$ can be characterised in a simpler form, i.e. $(C, X) \xrightarrow{e} (C \cup \{e\}, X)$ if $C \vdash e$, and $(C, X) \xrightarrow{e} (C \cup \{e\}, X \cup \{e\})$ if $C \nvdash e$. The subrelation of $\rightarrow_\mathcal{E}$ containing only states with empty credits coincides with the transition relation defined in [Win86].

By Def. 7.50 it immediately follows that, for all CES $\mathcal{E}$, the relation $\rightarrow_\mathcal{E}$ is *deterministic*, i.e. whenever $(C, X) \xrightarrow{a} (C', X')$ and $(C, X) \xrightarrow{a} (C'', X'')$, it must be $(C', X') = (C'', X'')$. Determinism is a very desirable property, e.g. in the context of contracts, because it ensures that the events to be performed by a participant at any given time are uniquely determined by the past actions.

Two immediate consequences of Def. 7.50 are reported in Lemma 7.51 below. In item (a) we start from a state (C, X), from which we fire a sequence of events σ. Then, we reach a state (C', X') where C' exactly comprises all the events in $C \cup \overline{\sigma}$, and the events which are removed from the credits are circularly enabled by the events in $C \cup \overline{\sigma}$. Item (b) states that the set X in a state $(\overline{\sigma}, X)$ is the least credit for σ — and so $X \subseteq \overline{\sigma}$ for all reachable states $(\overline{\sigma}, X)$. Note that there may exist different reachable states (C, X) with same the same C and uncomparable X. This is because, in general, there exists no least credit for a set of events C.

Lemma 7.51. *For all $C, C', X, X' \subseteq E$, and for all $\sigma \in E^*$:*

(a) $(C, X) \xrightarrow{\sigma} (C', X') \implies \overline{\sigma} = C' \setminus C \ \wedge \ C \cup \overline{\sigma} \Vdash X \setminus X'$

(b) $\sigma \in \mathcal{T}(X) \ \wedge \ X$ *least credit for* $\sigma \iff (\emptyset, \emptyset) \xrightarrow{\sigma} (\overline{\sigma}, X)$

Proof. Item (a) is straightforward by Def. 7.50.

For item (b$\Rightarrow$), let $\sigma \in \mathcal{T}(X)$. By induction on the length of $\sigma = \langle e_0 \dots e_n \rangle$, we prove that for all $i \leq n$, if Y_i is the least credit for σ_i, then $(\emptyset, \emptyset) \xrightarrow{\sigma_i} (\overline{\sigma_i}, Y_i)$. The base case is trivial, since the minimal credit for $\sigma_0 = \varepsilon$ is $Y_0 = \emptyset$. For the inductive case, by the induction hypothesis assume that $(\emptyset, \emptyset) \xrightarrow{\sigma_i} (\overline{\sigma_i}, Y_i)$. Then, by Def. 7.50:

$$(\overline{\sigma_i}, Y_i) \xrightarrow{e_i} (\overline{\sigma_{i+1}}, \Gamma^+(\overline{\sigma_i}, Y_i, e_i))$$

By Lemma 7.48, $\Gamma^+(\overline{\sigma_i}, Y_i, e_i) = Y_{i+1}$ is the least credit for σ_{i+1}.

For (b$\Leftarrow$), by an easy inductive argument on the length of σ (using Lemma 7.48 at each step) it follows that, for all $i \leq n$, if $(\emptyset, \emptyset) \xrightarrow{\sigma_i} (\overline{\sigma_i}, Y_i)$, then Y_i is the least credit for $\overline{\sigma_i}$. This implies $\sigma \in \mathcal{T}(X)$.

□

Example 7.52. *Recall the* CES $\mathcal{E}_7$ *from Fig. 7.1, with enablings $a \vdash b$ and $b \Vdash a$. According to its* LTS *(depicted in Fig. 7.6, left), in the initial state we can fire either the event a or the event b, by taking it on credit. In the state $(\{a\}, \{a\})$ we can perform the event b, and reach the state $(\{a, b\}, \emptyset)$. Instead, when performing a in the state $(\{b\}, \{b\})$ we reach the state $(\{a, b\}, \{b\})$. The event b cannot be discharged from that credit set, since there does not exists any $\Vdash$-enabling for it.*

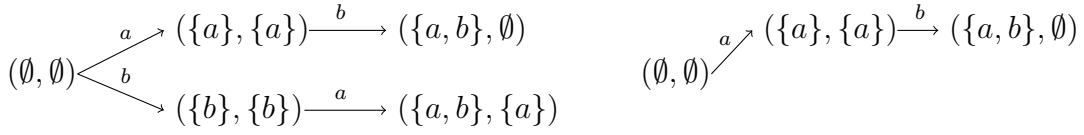


Figure 7.6: The LTS of CES $\mathcal{E}_7$ (left), and its urgent LTS (right).

The following theorem relates configurations with reachable states of the LTS. A (possibly infinite) set C is an X-configuration iff for all finite subsets D of C there exists a state with events containing D and with credits contained in X.

Theorem 7.53. *For all* CES $\mathcal{E}$*, for all* $C, X \subseteq E$*:*

$$C \in \mathcal{F}(X) \iff \forall D \subseteq_{fin} C,\, \exists X_0 \subseteq_{fin} X.\, \exists C_0.\, D \subseteq C_0 \subseteq_{fin} C.\, (\emptyset,\emptyset) \to^* (C_0, X_0)$$

Proof. ($\Rightarrow$) Let $C \in \mathcal{F}(X)$, and let $D \subseteq_{fin} C$. By Lemma 7.14, there exists $\sigma \in \mathcal{T}(X)$ such that $D \subseteq \overline{\sigma} \subseteq C$. Let $X_0 \subseteq_{fin} X$ be the least credit for σ. By Lemma 7.51, we have $(\emptyset,\emptyset) \xrightarrow{\sigma} (\overline{\sigma}, X_0)$. Therefore, the thesis follows by choosing $C_0 = \overline{\sigma}$.

($\Leftarrow$) Let $D \subseteq_{fin} C$, and assume that $(\emptyset,\emptyset) \xrightarrow{\sigma} (C_0, X_0)$, for some σ and X_0 such that $C_0 = \overline{\sigma}$, $D \subseteq C_0 \subseteq C$, and $X_0 \subseteq X$. Assume that $\sigma = \langle e_0 \dots e_n \rangle$, and that the trace has the form:

$$(\emptyset,\emptyset) \xrightarrow{e_0} (\overline{\sigma_1}, Y_1) \xrightarrow{e_1} (\overline{\sigma_2}, Y_2) \xrightarrow{e_2} \cdots \xrightarrow{e_n} (\overline{\sigma}, Y_n)$$

where $Y_n = X_0$. By Lemma 7.51(b), we have $\sigma \in \mathcal{T}(X_0)$. Since $X_0 \subseteq_{fin} X$, by Lemma 7.13(b), it is also true that $\sigma \in \mathcal{T}(X)$. Therefore, by Lemma 7.14 we conclude that $C \in \mathcal{F}(X)$.

□

This also gives an alternative way to characterise the reachable events, which uses the notion of LTS.

Lemma 7.54. *For all* CES $\mathcal{E}$*, for all* $e \in E$*, and for all* $X \subseteq E$*:*

$$e \in \mathcal{R}(X) \iff \exists C_0 \subseteq_{fin} E,\, X_0 \subseteq_{fin} X.\ (\emptyset,\emptyset) \to^* (C_0, X_0) \ \wedge\ e \in C_0$$

Proof. ($\Rightarrow$) Let $e \in \mathcal{R}(X)$. By Def. 7.34, there exists $C \in \mathcal{F}(X)$ such that $e \in C$. Since $C \in \mathcal{F}(X)$, by Theorem 7.53 there exists $X_0 \subseteq_{fin} X$ and C_0 such that $(\emptyset,\emptyset) \to^* (C_0, X_0)$ and $e \in C_0 \subseteq_{fin} C$.

($\Leftarrow$) Let σ be such that $(\emptyset,\emptyset) \xrightarrow{\sigma} (C_0, X_0)$, with $e \in C_0$ and $X_0 \subseteq_{fin} X$. By Lemma 7.51(b), $\sigma \in \mathcal{T}(X_0)$. Thus, $e \in \overline{\sigma} \in \mathcal{F}(X_0)$. By Lemma 7.24(b), $\overline{\sigma} \in \mathcal{F}(X)$. By Def. 7.34 we conclude that $e \in \mathcal{R}(X)$. □

7.5 Traces with shallow past

In order to deal with suffixes of traces, we now introduce the notion of *trace with past*. Traces with past are tightly bound with urgent events which will be the subject of the following section (7.6).

We want to characterize which sequences of events are possible from an already reached state in the LTS: given a set of already happend events and a credit set, we want to describe which sequences of actions can be done and which credit set we will have at the end.

Definition 7.55 (Trace with past)**.** *For all $C \subseteq_{fin} E$, $X \subseteq E$, we say that $\sigma = \langle e_0 \dots e_n \rangle \in E^*$ is an X-*trace with past *C iff $CF(C \cup \overline{\sigma})$, $C \cap \overline{\sigma} = \emptyset$, $DF(\sigma)$, and*

$$\forall i \leq n.\ (e_i \in X \ \ \vee \ \ C \cup \overline{\sigma_i} \vdash e_i \ \ \vee \ \ C \cup \overline{\sigma} \Vdash e_i) \tag{7.9}$$

We denote with $\mathcal{T}_{\mathcal{E}}^{C}(X)$ the set of X-traces with past C.

Note that, when $X \subseteq C$, we have $\mathcal{T}^{C}(X) = \mathcal{T}^{C}$.

The definition of trace with past is a bit more general that needed: in 7.55 it is not required that X is a credit set for C; but since it will be used mostly for those couple (C,X) which are state in the LTS, it that case we will have X as a least credit set for C.

From the definition of trace with past, we obtain the definition of configuration with past.

Definition 7.56 (Configuration with past)**.** *For all $P, X \subseteq E$ we define the set $\mathcal{F}_{\mathcal{E}}^{P}(X)$ of X-configurations with past P as follows:*

$$\mathcal{F}_{\mathcal{E}}^{P}(X) \ = \ \{C \subseteq E \setminus P \mid CF(C) \ \wedge \ \forall e \in C.\ \exists \sigma \in \mathcal{T}_{\mathcal{E}}^{P}(X).\ e \in \overline{\sigma} \subseteq C\}$$

Traces with past enjoy some basic properties, which derive directly from traces basic properties.

Lemma 7.57. *Let $C, C' \subseteq E$, and let $\sigma, \sigma', \eta, \eta' \subseteq E^*$. Then :*

(a) $C \cap \overline{\sigma} = \emptyset \implies \sigma \in \mathcal{T}^{C}(\overline{\sigma})$

(b) $X \subseteq Y \implies \mathcal{T}^{C}(X) \subseteq \mathcal{T}^{C}(Y)$

(c) $\sigma \in \mathcal{T}^{C}(X) \ \wedge \ \sigma' \in \mathcal{T}^{C}(X) \ \wedge \ CF(\overline{\sigma\sigma'}) \implies (\sigma\sigma')\downarrow \in \mathcal{T}^{C}(X)$

(d) $\sigma \in \mathcal{T}^{C}(X) \ \wedge \ \sigma' \in \mathcal{T}^{C}(X) \ \wedge \ CF(\overline{\sigma\sigma'}) \implies (\sigma \mid \sigma')\downarrow \in \mathcal{T}^{C}(X)$

Proof. Direct consequences of Def. 7.55 and of trace basic properties 7.18. □

Since every event may happen only once, it is redundant to insert events already happened into the credit set.

Lemma 7.58. *Let $X, C \subseteq E$. Then, $\mathcal{T}^{C}(X) = \mathcal{T}^{C}(X \setminus C)$*

Proof. Straightforward from Def. 7.55. □

The following lemma states that composing an X-trace σ with a trace with past σ, generates an X-trace.

Lemma 7.59. *For all $X, C \subseteq E$, and $\sigma, \eta \in E^*$:*

$$\eta \in \mathfrak{T}(X) \ \wedge \ \sigma \in \mathfrak{T}^C \ \wedge \ \overline{\eta} = C \implies \eta\sigma \in \mathfrak{T}(X)$$

Proof. Let $\sigma = \langle e_0 \dots e_n \rangle \in \mathfrak{T}^C$. By Def. 7.55 we have $CF(C \cup \overline{\sigma})$, $C \cap \overline{\sigma} = \emptyset$, $DF(\sigma)$, and

$$\forall i \leq n.\ (C \cup \overline{\sigma_i} \vdash e_i \ \vee \ C \cup \overline{\sigma} \Vdash e_i) \tag{7.10}$$

Let $\eta = \langle a_0 \dots a_k \rangle \in \mathfrak{T}(X)$ be such that $\overline{\eta} = C$. We have to prove that $\nu = \eta\sigma \in \mathfrak{T}(X)$, i.e. $CF(\overline{\nu})$, $DF(\nu)$ and

$$\forall i \leq k.\ (a_i \in X \ \vee \ \overline{\nu_i} \vdash a_i \ \vee \ \overline{\nu} \Vdash a_i) \ \wedge \tag{7.11}$$

$$\forall k < i \leq n.\ (e_i \in X \ \vee \ \overline{\nu_i} \vdash e_i \ \vee \ \overline{\nu} \Vdash e_i) \tag{7.12}$$

Since $CF(C \cup \overline{\eta})$ then $CF(\overline{\nu})$. Since $C \cup \overline{\sigma}$ and $DF(\sigma)$, we have $DF(\nu)$. Since $\eta \in \mathfrak{T}(X)$, (7.11) trivially holds. For (7.12), we have two further subcases, based on how the event e_i was justified in (7.10):

- $C \cup \overline{\sigma_i} \vdash e_i$. Since $i > k$, we have $C \cup \overline{\sigma_i} = \overline{\nu_i}$, and then $\overline{\nu_i} \vdash e_i$.
- $C \cup \overline{\sigma} \Vdash e_i$. Since $C \cup \overline{\sigma} = \overline{\nu}$, we have $\overline{\nu} \Vdash e_i$. □

The following lemma relates traces with past with the LTS of $\mathcal{E}$.

Lemma 7.60. *Let (C, X) be a reachable state of $\mathrm{LTS}_{\mathcal{E}}$. Then:*

$$\sigma \in \mathfrak{T}^C \iff \exists X_0 \subseteq X.\ (C, X) \xrightarrow{\sigma} (C \cup \overline{\sigma}, X_0)$$

Proof. ($\Rightarrow$) Let $\sigma \in \mathfrak{T}^C$. Since (C, X) is reachable, there exists η such that $(\emptyset, \emptyset) \xrightarrow{\eta} (C, X)$ and $\overline{\eta} = C$; so by Lemma 7.51, $\eta \in \mathfrak{T}(X)$. By Lemma 7.59, $\eta\sigma \in \mathfrak{T}(X)$. Let $X_0 \subseteq X$ be the least credit of $\eta\sigma$. By Lemma 7.51 we have $(\emptyset, \emptyset) \xrightarrow{\eta\sigma} (\overline{\eta\sigma}, X_0)$. Since $\mathrm{LTS}_{\mathcal{E}}$ is deterministic, therefore we conclude that $(C, X) \xrightarrow{\sigma} (C \cup \overline{\sigma}, X_0)$. The thesis follows because $\overline{\eta\sigma} = C \cup \overline{\sigma}$.

($\Leftarrow$) Since (C, X) is reachable, by Lemma 7.51 there exists $\eta \in \mathfrak{T}(X)$ such that $\overline{\eta} = C$ and $(\emptyset, \emptyset) \xrightarrow{\eta} (C, X)$. By hypothesis there exists X_0 such that $(C, X) \xrightarrow{\sigma} (C \cup \overline{\sigma}, X_0)$. Summing up, $(\emptyset, \emptyset) \xrightarrow{\eta\sigma} (C \cup \overline{\sigma}, X_0)$. By Lemma 7.51, X_0 is the least credit for $\eta\sigma$. By Lemma 7.13(b), since $\eta\sigma \in \mathfrak{T}(X_0)$ and $X_0 \subseteq X$, we have $\eta\sigma \in \mathfrak{T}(X)$. By Def. 7.55, we conclude that $\sigma \in \mathfrak{T}^C(X)$, and the thesis follows from Lemma 7.58 because $X \subseteq C$. □

Having defined traces and configurations with past, we are now ready to define reachable events with past.

Definition 7.61 (Reachable events with past)**.** *For all $C \subseteq_{fin} E$ and $X \subseteq E$, we define:*

$$\mathcal{R}^C_{\mathcal{E}}(X) = \bigcup \mathcal{F}^C_{\mathcal{E}}(X)$$

The following lemma relates reachability with past to (plain) reachability. Note that the inclusion $\mathcal{R}(X) \supseteq \mathcal{R}^C(X) \cup C$ does not hold. For instance, in the event structure with enabling $\{a\} \vdash b$ and with $C = \{a\}$, we have that $\mathcal{R}(\emptyset) = \emptyset$, but $\mathcal{R}^C(\emptyset) = \{b\}$.

Lemma 7.62. *For all $C \subseteq_{fin} E$ and $X \subseteq E$, $\mathcal{R}^C(X) \cup C = \mathcal{R}(C \cup X)$.*

Proof. For $(\supseteq)$, let $e \in \mathcal{R}(C \cup X)$. Then, there exists $\eta \in \mathcal{T}(C \cup X)$ such that $e \in \overline{\eta}$. If $e \in C$, we already have the thesis. Otherwise, assume $e \notin C$. Let η' be the sequence obtained by removing from η all the events in C, while preserving the order of the other events. Then that $\eta' \in \mathcal{T}^C(X)$ and $e \in \overline{\eta'}$, from which $e \in \mathcal{R}^C(X)$.

For $(\subseteq)$, let $e \in \mathcal{R}^C(X) \cup C$. If $e \in C$, the thesis holds trivially. Otherwise, there exists $\sigma \in \mathcal{T}^C(X)$ such that $e \in \overline{\sigma}$ and $\overline{\sigma} \cap C = \emptyset$. Let $\sigma' = \sigma_C \; \sigma$, where σ_C is an arbitrary sequentialisation of the events in C. Then $\sigma' \in \mathcal{T}(C \cup X)$, from which $e \in \mathcal{R}(C \cup X)$. □

7.6 Urgent events

Computations on $\mathrm{LTS}_{\mathcal{E}}$ are far too liberal: they allow us to fire an event either if it is ($\vdash$ or $\Vdash$)-enabled by the already fired events, or — by taking it on credit — if it will be honoured in the future, or even if it will not. Except for the conflicting events, any event can be fired, with the risk of keeping such event in the credit set forever. Intuitively, one would like to perform those events only which guarantee to eventually reach a state with empty credits. Such events will be called *urgent*.

For instance, in $\mathcal{E}_7$ the event a is urgent in the initial state $(\emptyset, \emptyset)$. In such state b is not urgent, but it will be urgent in the state $(\{a\}, \{a\})$ where a has been performed on credit.

We are assuming that non-deterministic choices are *angelic*, i.e. they are not affected by the environment. For instance, if we extend $\mathcal{E}_7$ with the enabling $a \vdash c$ and the conflict $b \# c$, we shall still say that a is urgent in the initial state: i.e., the environment cannot prevent a from being honoured by choosing the branch c. The case of *demonic* non-determinism, that is ensuring that an event performed on credit will *always* be honoured, can be better understood by setting up a suitable adversarial model. This requires a quite more complex game-theoretic treatment, and it is the subject of [BCZ13].

Definition 7.63 (Urgent events)**.** *For all $e \in E$, and for all $C, X \subseteq E$, we say that e is* urgent *in (C, X) iff*

$$\exists \sigma.\ (C, X) \xrightarrow{e\sigma}_{\mathcal{E}} (C \cup \overline{\sigma}, \emptyset)$$

We denote with $\mathcal{U}^C_{\mathcal{E}}(X)$ the set of urgent events in (C, X).

Summing up, the events that are urgent in (C, X) are those already enabled by C, or those which can be done on credit, on behalf that they will be honoured when the right choices will be made in the future.

Example 7.64. *Consider* $\mathcal{E}_7$ *and its* LTS *depicted in Fig. 7.6. Event* a *is urgent in* $(\emptyset,\emptyset)$*, because there exists a path from* $(\{a\},\{a\})$ *which leads to an empty credit set. For the same reason,* b *is urgent in* $(\{a\},\{a\})$*. On the contrary* b *is* not *urgent in* $(\emptyset,\emptyset)$*, because whatever choices are made in the future, it would not be possible to honour the credit* $\{b\}$*. Indeed, if a transition labelled* b *is taken from state* $(\emptyset,\emptyset)$ *of* $\mathrm{LTS}_{\mathcal{E}_7}$*, then all future states will contain the credit* b*.*

The following lemma provides an alternative characterisation of urgent events, in terms of traces with past. An event e is urgent in (C,X) iff there exists a trace with past C such that the first element of the trace is e, and the credit X is honoured by the events in the trace together with those in C.

Lemma 7.65. *For all* $e \in E$*, and for all reachable state* (C,X) *of* $\mathrm{LTS}_{\mathcal{E}}$*,*

$$e \in \mathcal{U}^C_{\mathcal{E}}(X) \iff \exists \sigma.\ e\sigma \in \mathcal{T}^C \ \wedge\ C \cup \overline{e\sigma} \Vdash X$$

Proof. Straightforward after Lemma 7.60 and Lemma 7.51(a). □

Pruning away from an LTS all the transitions labelled by non-urgent events, we obtain a new LTS, denoted by $\rightharpoonup_{\mathcal{U}_\mathcal{E}}$. The crucial property of the $\rightharpoonup_{\mathcal{U}_\mathcal{E}}$ is that, by following its transitions, one is always guaranteed to reach a state where all the credits have been honoured (see Lemma 7.68 below). Fig. 7.6 (right) displays the urgent LTS for $\mathcal{E}_7$.

Definition 7.66 (LTS of urgent events)**.** *We define the relation* $\rightharpoonup_{\mathcal{U}_\mathcal{E}}$ *as the largest subset of* $\rightarrow_{\mathcal{E}}$ *such that:*

$$(C,X) \ \stackrel{e}{\rightharpoonup}_{\mathcal{U}_\mathcal{E}}\ (C',X') \qquad \textit{iff}\ (C,X) \stackrel{e}{\rightarrow}_{\mathcal{E}} (C',X')\ \textit{and}\ e \in \mathcal{U}^C(X)$$

Note that in the absence of circularity the LTS $\rightharpoonup_{\mathcal{U}_\mathcal{E}}$ coincides with the LTS defined in [Win86], where the component X is always empty. For instance, let us consider an event structure without circular enablings, and let (C,X) be a reachable state in its LTS. If $X \neq \emptyset$, then the credit X will never be honoured since there are no circular enablings, hence no events will be urgent in (C,X). Otherwise, if $X = \emptyset$, then the urgent events in $(C,\emptyset)$ are exactly those events e such that $C \vdash e$. This is because, by Def. 7.50, $(C,\emptyset) \stackrel{e}{\rightarrow} (C \cup \{e\},\emptyset)$ if $C \vdash e$.

The following lemma relates the traces in $\mathcal{T}_\mathcal{E}$ with the traces in $\rightharpoonup_{\mathcal{U}_\mathcal{E}}$. The traces in $\mathcal{T}_\mathcal{E}$ are exactly those traces in $\rightharpoonup_{\mathcal{U}_\mathcal{E}}$ which lead to a state (C,X) with $X = \emptyset$.

Lemma 7.67.

$$\sigma \in \mathcal{T}_\mathcal{E} \iff (\emptyset,\emptyset) \stackrel{\sigma}{\rightharpoonup}_{\mathcal{U}_\mathcal{E}} (\overline{\sigma},\emptyset)$$

Proof. To prove ($\Rightarrow$), we distinguish between two cases. If $\sigma = \varepsilon$, the statement holds trivially. Otherwise, assume $\sigma = \nu e$, and let X be the least credit for $\nu \in \mathcal{T}(X)$. Since $\nu e \in \mathcal{T}_\mathcal{E}$, by Lemma 7.51 it must be $(\emptyset,\emptyset) \stackrel{\nu}{\rightarrow}_{\mathcal{E}} (\overline{\nu},X) \stackrel{e}{\rightarrow}_{\mathcal{E}} (\overline{\nu e},\emptyset)$. By Lemma 7.60, we have that $e \in \mathcal{T}^{\overline{\nu}}$, and clearly $\overline{\nu e} \Vdash X$. Therefore, by Lemma 7.65 we conclude that $e \in \mathcal{U}^{\overline{\nu}}(X)$, from which the thesis $(\emptyset,\emptyset) \stackrel{\nu}{\rightharpoonup}_{\mathcal{U}_\mathcal{E}} \stackrel{e}{\rightharpoonup}_{\mathcal{U}_\mathcal{E}}$.

The direction ($\Leftarrow$) follows from Lemma 7.60, since $\rightharpoonup_{\mathcal{U}_\mathcal{E}} \subseteq \rightarrow_{\mathcal{E}}$. □

The following lemma establishes a crucial property of the LTS $\rightharpoonup_{\mathcal{U}_\varepsilon}$, that is the ability to reach, starting from any state (C, X) reachable from $(\emptyset, \emptyset)$ and following only $\rightharpoonup_{\mathcal{U}_\varepsilon}$ transitions, a state where all the credits have been honoured.

Lemma 7.68 (Progress)**.** *Let (C, X) be a reachable state of $\rightharpoonup_{\mathcal{U}_\varepsilon}$. Then:*

$$\exists \eta.\ (C, X) \xrightarrow{\eta}_{\mathcal{U}_\varepsilon} (C \cup \overline{\eta}, \emptyset)$$

Proof. We first prove the following technical result. For all C, X, σ:

$$(C, X) \xrightarrow{\sigma}_{\mathcal{U}_\varepsilon} (C', \emptyset) \iff (C, X) \xrightarrow{\sigma}_{\mathcal{E}} (C', \emptyset) \tag{7.13}$$

The direction ($\Rightarrow$) of (7.13) follows because $\rightharpoonup_{\mathcal{U}_\varepsilon} \subseteq \rightarrow_{\mathcal{E}}$. The other direction can be easily proved by induction on the length of σ.

Back to the main statement, assume that $(\emptyset, \emptyset) \xrightarrow{\sigma}_{\mathcal{U}_\varepsilon} (\overline{\sigma}, X)$, for some X. If $X = \emptyset$, we conclude by choosing $\eta = \varepsilon$. Otherwise, let $\sigma = \langle e_0 \cdots e_n \rangle$, and let $C_i = \overline{\sigma_i}$. Then, $(\emptyset, \emptyset) \xrightarrow{e_0}_{\mathcal{U}} (C_1, X_1) \xrightarrow{e_1}_{\mathcal{U}} (C_2, X_2) \xrightarrow{e_2}_{\mathcal{U}} \cdots \xrightarrow{e_n}_{\mathcal{U}} (\overline{\sigma}, X)$. By Def. 7.63, we have that for all $i \leq n$, $e_i \in \mathcal{U}^{C_i}(X_i)$. In particular, for $i = n$ we have that there exists η such that $(\overline{\sigma}, X) \xrightarrow{\eta} (\overline{\sigma\eta}, \emptyset)$. The thesis follows directly by (7.13). □

The following lemma relates urgent events with reachability. Item (a) states that urgent events in (C, X) are also X-reachable with past C. Note that the converse inclusion does not hold, i.e. in the event structure with enablings $\{a\} \vdash b$ and $\{b\} \Vdash a$, both a and b are reachable, but only a is urgent in $\emptyset$. Item (b) states that reachable events in $\mathcal{E}$ are exactly those events which label some transition in the LTS $\rightharpoonup_{\mathcal{U}_\varepsilon}$.

Lemma 7.69. *For all* CES $\mathcal{E}$*, and for all $C, X \subseteq E$:*

(a) $\mathcal{U}^C_{\mathcal{E}}(X) \subseteq \mathcal{R}^C_{\mathcal{E}}(X)$

(b) $\mathcal{R}_{\mathcal{E}} = \{e \mid \exists \sigma : (\emptyset, \emptyset) \xrightarrow{\sigma}_{\mathcal{U}_\varepsilon} \xrightarrow{e}_{\mathcal{U}_\varepsilon}\}$

Proof. Item (a) is straightforward by Lemma 7.65 and Def. 7.61.

For $\subseteq$ of item (b), by Lemma 7.67 it follows that for all σ, η, if $\sigma\eta \in \mathcal{T}_{\mathcal{E}}$, then $(\emptyset, \emptyset) \xrightarrow{\sigma}_{\mathcal{U}_\varepsilon}$. It is easy to check that this implies the thesis.

For $\supseteq$, assume that $(\emptyset, \emptyset) \xrightarrow{\sigma}_{\mathcal{U}} (\overline{\sigma}, X)$. By Lemma 7.68, there exists η such that $(\overline{\sigma}, X) \xrightarrow{\eta}_{\mathcal{U}} (\overline{\sigma\eta}, \emptyset)$. By Lemma 7.67, $\sigma\eta \in \mathcal{T}_{\mathcal{E}}$. Therefore, all the events in $\overline{\sigma}$ are comprised in $\mathcal{R}_{\mathcal{E}}$. □

A relevant question is whether, for any CES $\mathcal{E}$, there exists a CES $\mathcal{E}'$ without circular enablings such that the LTS $\rightharpoonup_{\mathcal{U}_\varepsilon}$ equals to $\rightarrow_{\mathcal{E}'}$. In other words, we wonder whether the expressiveness with the urgent LTS is the same as that of the LTS of Winskel's ES. A negative answer is displayed in Fig. 7.7, which shows a CES $\mathcal{E}$ for which there exists no ES the LTS of which corresponds to $\rightharpoonup_{\mathcal{U}_\varepsilon}$. Indeed, ES cannot distinguish between two states which only differ for the credits, like e.g. $(\{a, b\}, \emptyset)$ and $(\{a, b\}, \{b\})$ in Fig. 7.7. In Winskel's ES, a transition from a state C only depends on the events in C, and not on the order in which these events have been fired. Instead, transitions in CES also depend on the credits accumulated in the history of execution.

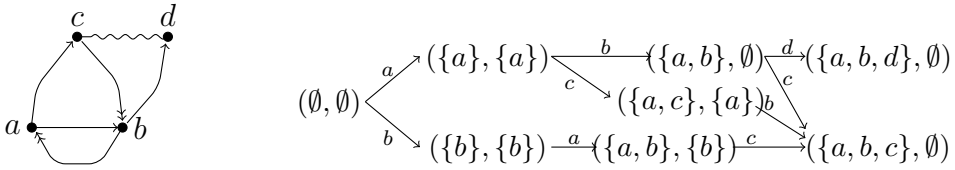


Figure 7.7: An event structure $\mathcal{E}$ (left) and the LTS $\rightharpoonup_{\mathcal{U}_\mathcal{E}}$ (right).

7.6.1 Urgency for conflict-free CES

In this section we study urgent events in the special case of conflict-free CES. The following lemma simplifies the characterisation of urgent events in terms of traces given by Lemma 7.65. This simplified characterisation only holds for those states (C, X) which are reachable in the LTS $\rightharpoonup_{\mathcal{U}_\mathcal{E}}$. Under this hypothesis, it is no longer needed to check that the credits in X are honoured: this is already guaranteed by the fact that there exists a trace $e\sigma$ in $\mathcal{T}^C$. Therefore, for conflict-free CES the LTS $\rightharpoonup_{\mathcal{U}_\mathcal{E}}$ can be simplified by eliminating the component X from the states.

Lemma 7.70. *For a a conflict-free* CES $\mathcal{E}$*, and a reachable state* (C, X) *of* $\rightharpoonup_{\mathcal{U}_\mathcal{E}}$*:*

$$e \in \mathcal{U}^C_\mathcal{E}(X) \iff \exists \sigma.\ e\sigma \in \mathcal{T}^C_\mathcal{E}$$

Proof. The ($\Rightarrow$) direction is straightforward after Lemma 7.65. For ($\Leftarrow$), let σ be such that $(\emptyset, \emptyset) \xrightarrow{\sigma}_\mathcal{U} (C, X)$ and $\overline{\sigma} = C$, and assume that $e\nu \in \mathcal{T}^C$, for some ν. By Lemma 7.68, there exists η such that $(C, X) \xrightarrow{\eta}_\mathcal{U} (\overline{\sigma\eta}, \emptyset)$. By Lemma 7.60, $\eta \in \mathcal{T}^C$, and by Lemma 7.51(a), $C \cup \overline{\eta} \Vdash X$. By Lemma 7.57(c), since $CF(e\nu\eta)$ then $e\nu\eta \in \mathcal{T}^C$. By Lemma 7.51(b), $\sigma \in \mathcal{T}(X)$. Therefore, Lemma 7.59 gives that $\sigma e\nu\eta \in \mathcal{T}(X)$, from which Lemma 7.18(c) yields $\sigma e\nu\eta \in \mathcal{T}$. Finally, Lemma 7.67 gives that $(C, X) \xrightarrow{e\nu\eta}_\mathcal{U}$, which concludes. □

Note that, if the condition that (C, X) is reachable in $\rightharpoonup_{\mathcal{U}_\mathcal{E}}$ is false, then the ($\Leftarrow$) direction of Lemma 7.70 may be false as well. For instance, in the CES with enablings $\{a\} \vdash b$ and $\{b\} \Vdash a$, consider the state $(\{b\}, \{b\})$, which is not reachable in $\rightharpoonup_{\mathcal{U}_\mathcal{E}}$. Then, $\langle a \rangle \in \mathcal{T}^{\{b\}}(\{b\})$, but a is not urgent in $(\{b\}, \{b\})$.

In the case of conflict-free CES, we provide in Def. 7.71 an alternative characterisation of urgent events. Unlike $\mathcal{U}^C(X)$, the set $\hat{\mathcal{U}}^C(X)$ also contains all the events in C.

The relation between $\mathcal{U}^C(X)$ and $\hat{\mathcal{U}}^C(X)$ is formalised in Lemma 7.72 below.

Definition 7.71 (Urgent events for conflict-free CES)**.** *For all* $C, X \subseteq E$*, we define the set* $\hat{\mathcal{U}}^C(X)$ *as follows:*

$$\frac{e \in C}{e \in \hat{\mathcal{U}}^C(X)}\ (\in_{\hat{\mathcal{U}}}) \qquad \frac{C \vdash e}{e \in \hat{\mathcal{U}}^C(X)}\ (\vdash_{\hat{\mathcal{U}}}) \qquad \frac{\mathcal{R}(C \cup X) \Vdash e}{e \in \hat{\mathcal{U}}^C(X)}\ (\Vdash_{\hat{\mathcal{U}}})$$

Lemma 7.72. *For a a conflict-free* CES $\mathcal{E}$*, and a reachable state* (C, X) *of* $\rightharpoonup_{\mathcal{U}_\mathcal{E}}$*:*

$$\hat{\mathcal{U}}^C_\mathcal{E}(X) \;=\; \mathcal{U}^C_\mathcal{E}(X) \cup C$$

Proof. For the inclusion ($\subseteq$), let $e \in \hat{\mathcal{U}}^C(X)$. If $e \in C$, we already have the thesis. Otherwise, since (C, X) is a reachable state of $\rightharpoonup_{\mathcal{U}_\mathcal{E}}$, by Lemma 7.70 it suffices to show some σ such that $e\sigma \in \mathcal{T}^C$. We now proceed by cases on the rule used to deduce $e \in \hat{\mathcal{U}}^C_\mathcal{E}(X)$.

- ($\vdash_{\hat{\mathcal{U}}}$). Let $\sigma = \varepsilon$. Then, $e\sigma \in \mathcal{T}^C$ holds, because $C \vdash e$.
- ($\Vdash_{\hat{\mathcal{U}}}$). Since $\mathcal{R}(C \cup X) \Vdash e$, by Notation 7.2 there exists $D \subseteq_{\mathit{fin}} \mathcal{R}(C \cup X)$ such that $D \Vdash e$. By Lemma 7.62, $\mathcal{R}(C \cup X) = \mathcal{R}^C(X) \cup C$. By Def. 7.61, $\mathcal{R}^C(X) = \bigcup \mathcal{F}^C(X)$. Since configurations with past enjoy coherence and $\mathcal{E}$ is conflict-free, then $D \setminus C \subseteq_{\mathit{fin}} \mathcal{R}^C(X) \in \mathcal{F}^C(X)$. Thus, by Def. 7.56, there exists $\sigma \in \mathcal{T}^C(X)$ such that $\overline{\sigma} \supseteq D \setminus C$. Since $X \subseteq C$, this implies that $\sigma \in \mathcal{T}^C$. Since $C \cup \overline{e\sigma} \supseteq C \cup D \Vdash e$, then we conclude that there exists σ' such that $e\sigma' = (e\sigma)\downarrow \in \mathcal{T}^C$.

For the inclusion ($\supseteq$), let $e \in \mathcal{U}^C(X) \cup C$. If $e \in C$, the thesis follows by rule ($\in_{\hat{\mathcal{U}}}$). Otherwise, if $e \in \mathcal{U}^C(X)$, by Lemma 7.70, there exists σ such that $e\sigma \in \mathcal{T}^C$. There are the following two cases, according to how e was justified in $e\sigma$.

- $C \vdash e$. By rule ($\vdash_{\hat{\mathcal{U}}}$), we conclude that $e \in \hat{\mathcal{U}}^C(X)$, for all X.
- $C \cup \overline{e\sigma} \Vdash e$. Since $e\sigma \in \mathcal{T}^C$, then $\overline{e\sigma} \in \mathcal{F}^C$, and so by Def. 7.61, $\overline{e\sigma} \subseteq \mathcal{R}^C$. By Lemma 7.62, $\mathcal{R}(C) = \mathcal{R}^C \cup C \supseteq \overline{e\sigma} \cup C \Vdash e$. Thus, by rule ($\Vdash_{\hat{\mathcal{U}}}$), we conclude that $e \in \hat{\mathcal{U}}^C(\emptyset)$.

□

Lemma 7.73. *For a conflict-free* CES $\mathcal{E}$*, and for all* C, C', X, Y*:*

(a) $C \subseteq C' \;\wedge\; X \subseteq Y \implies \hat{\mathcal{U}}^C(X) \subseteq \hat{\mathcal{U}}^{C'}(Y)$

(b) $Y \subseteq \mathcal{R}(X) \implies \hat{\mathcal{U}}^C(X \cup Y) \subseteq \hat{\mathcal{U}}^C(X)$

(c) $\hat{\mathcal{U}}^C(X) \subseteq \mathcal{R}(C \cup X)$

Proof. Item (a) is straightforward by Def. 7.71.

For item (b), assume that $e \in \hat{\mathcal{U}}^C(X \cup Y)$. We only have to consider rule ($\Vdash_{\hat{\mathcal{U}}}$), since in the other rules X is immaterial. By the rule premise, it must be $\mathcal{R}(C \cup X \cup Y) \Vdash e$. By Lemma 7.36 (c), $\mathcal{R}(C \cup X \cup Y) = \mathcal{R}(C \cup X)$, and by applying rule ($\Vdash_{\hat{\mathcal{U}}}$) we obtain the thesis $e \in \hat{\mathcal{U}}^C(X)$.

For item (c), by Lemma 7.72, a and 7.62, we have:

$$\hat{\mathcal{U}}^C(X) \;=\; \mathcal{U}^C(X) \cup C \;\subseteq\; \mathcal{R}^C(X) \cup C \;=\; \mathcal{R}(C \cup X)$$

□

Chapter 8

Reconciling agreement and protection

In Chapter 6 we have shown that agreement and protection cannot coexist in a relevent class of contracts, namely the contracts with circular finite Offer-Request payoffs (Theorem 6.49). As made evident by Theorem 6.45, to protect herself a participant A must obtain all her requests R^i_{A} before doing all her offers O^i_{A}. If all participants adhere to this principle, agreement is not possible.

For instance, Alice and Bob in Example 6.6 would be protected by contracts with enablings $a \vdash b$ and $b \vdash a$, but no agreement would be possible because nobody risks doing the first step.

To reconcile agreements with protection, a participant (say, Alice) could relax her contract, i.e. she could do a in change of the *promise* of Bob to do b. In this case Alice can safely do the first step, because either Bob does b, or he will be culpable of a contract violation.

To model this kind of "conditional" obligations, we shall use the theory of circular event structures introduced in Chapter 7. A circular enabling $b \Vdash a$ will be used to model the obligation for Alice to perform event a, under the guarantee that Bob will eventually be obliged to perform b (or be punishable if omitting to do that).

Chapter overview. The rest of this chapter is organised as follows. We conservatively extend the theory of contracts introduced in Chapter 6, by allowing the component $\mathcal{E}$ of a contract to be a CES. In Section 8.1 we set up the crucial notion of *prudence*. Roughly, prudent events are those which can be fired on the guarantee that the credits they create will be eventually honoured. In Section 8.2 we accordingly review the notion of agreement. Finally, in Section 8.3 we show how CES allow for reconciling agreement with protection. The main result of this chapter is that, using the refined model for contracts, it is possible to overcome the negative result in Theorem 6.49. More precisely, in Theorem 8.19 we show a synthesis technique that, starting from the participant payoffs, constructs a set of contracts which protect all the participants, and still admit an agreement.

8.1 Prudence

In this chapter we assume that the definition of a contract $\mathcal{C} = (\mathcal{E}, \mathcal{A}, \pi, \Phi)$ is the same as in Def. 6.1, except that now the component $\mathcal{E}$ is a CES.

Recall from Def. 6.2 that a *play* of a contract $\mathcal{C}$ is a (finite or infinite) sequence σ of events such that $(\emptyset, \emptyset) \xrightarrow{\sigma}_{\mathcal{E}}$. By Def. 7.50, each play $\sigma = \langle e_0 \cdots e_i \cdots \rangle$ uniquely identifies a computation in the CES $\mathcal{E}$. This computation has the form:

$$(\emptyset, \emptyset) \xrightarrow{e_0} (\overline{\sigma_1}, \Gamma(\sigma_1)) \cdots \xrightarrow{e_i} (\overline{\sigma_{i+1}}, \Gamma(\sigma_{i+1})) \cdots$$

The first element of each pair is the set of events occurred so far; the second element is the least set of events done "on credit", i.e. performed in the absence of a causal justification. After Def. 7.16, for a play η we have $\Gamma(\eta) = \{e_i \in \overline{\eta} \mid \overline{\eta_i} \not\vdash e_i \ \wedge \ \overline{\eta} \not\Vdash e_i\}$.

The notions of *strategy* and *conformance* to a strategy are as in Section 6.1.

The key difference between ES-based and CES-based contracts is the notion of innocence. In the ES-based model, a participant A is culpable in a play σ when some event e of A is enabled in σ. Here, in addition to enabled events, we consider obligations those events which can be done "on credit", under the guarantee that they will be eventually honoured, whatever events are done later on by the other participants. These events are said *prudent*.

Before setting up the crucial notion of prudent events, we provide some underlying intuitions. The definition of prudent strategies and of innocent participants is mutually coinductive. A participant A is considered *innocent* in a play σ when she has done all her prudent events in σ (otherwise A is *culpable*). Hence, if a strategy tells A to do all her prudent events, then in all *fair* plays these events must either become imprudent, or be fired.

Given a finite play σ of past events, an event e is said *prudent* in σ whenever there exists a prudent strategy Σ which prescribes to do e in σ. A strategy for A *with past* σ (namely, conform to σ) is prudent whenever, in all fair extensions of σ where all other participants are innocent, the events performed on credit by A are eventually honoured; at most, the credits coming from the past σ will be left.

Similarly to the ES-based model, we neglect those unfair plays where an action permanently enabled is not eventually performed. Indeed, in unfair plays an honest participant could be perpetually prevented from performing a prudent action.

Definition 8.1 (Prudence)**.** *A strategy Σ for A with past σ is* prudent *if, for all fair plays σ' extending σ, conform to Σ, and where all $\mathsf{B} \neq \mathsf{A}$ are innocent,*

$$\exists k > |\sigma|. \ \Gamma(\sigma'_k) \cap \pi^{-1}(\mathsf{A}) \subseteq \Gamma(\sigma)$$

An event e is prudent *in σ if there exists a prudent strategy Σ with past σ such that $e \in \Sigma(\sigma)$.*

A participant A is innocent *in $\sigma = \langle e_0\, e_1 \cdots \rangle$ iff:*

$$\forall e \in \pi^{-1}(\mathsf{A}).\ \forall i \geq 0.\ \exists j \geq i.\ e \text{ is imprudent in } \sigma_j$$

Note that the empty strategy is trivially prudent. Also, the definition of innocence for CES-based contracts conservatively extends that in Def. 6.21. More precisely, in a CES without $\Vdash$-enablings, a participants A is innocent in σ according to Def. 8.1 iff A is such according to Def. 6.21, i.e. if $\overline{\sigma} \not\Vdash e$, for all $e \in \pi^{-1}(\mathsf{A})$.

Example 8.2. *Consider the obligations modelled by the* CES $\mathcal{E}_1$, $\mathcal{E}_2$, $\mathcal{E}_7$, $\mathcal{E}_8$, $\mathcal{E}_{10}$ *in Figures 3.1 and 7.1, where* $\pi(a) = \mathsf{A}$ *and* $\pi(b) = \pi(c) = \mathsf{B}$*:*

- *in* $\mathcal{E}_1$*, the only prudent event in the empty play is* a*, which is enabled by* $\emptyset$*, and the only culpable participant is* A*. In* $\langle a \rangle$*,* b *becomes prudent, and* B *becomes culpable. In* $\langle ab \rangle$ *no event is prudent and no participant is culpable.*

- *in* $\mathcal{E}_2$*, there are no prudent events in* ε*. Instead, event* a *is prudent in* $\langle b \rangle$*, while* b *is prudent in* $\langle a \rangle$*: this is coherent with the fact that the prudence of an event does not depend on the assumption that all the events done in the past were prudent. In* $\langle ab \rangle$ *and* $\langle ba \rangle$ *no events are prudent.*

- *in* $\mathcal{E}_7$*, event* a *is prudent in* ε*: indeed, the only fair play* $a\eta$ *where* B *is innocent is* $\langle ab \rangle$*, where* $\Gamma(ab) = \emptyset$*. Instead,* b *is* not *prudent in* ε*, because* $b \in \Gamma(b\eta)$ *for all* η*. Event* b *becomes prudent in* $\langle a \rangle$*.*

- *in* $\mathcal{E}_8$*, both* a *and* b *are prudent in* ε*.*

- *in* $\mathcal{E}_{11}$*,* a *is* not *prudent in* ε*, because if* B *chooses to do* c*, then the credit* a *can no longer be honoured. Actually, no events are prudent in* ε*, while both* b *and* c *are prudent in* $\langle a \rangle$*, and* a *is prudent in both* $\langle b \rangle$ *and* $\langle c \rangle$*.*

Recall from Def. 6.22 that the strategy Σ^e_{A} which obliges A to do all her $\vdash$-enabled events is innocent for A in the ES-based contract model. Here, the eager strategy is no longer innocent, because some prudent events may exist which are not $\vdash$-enabled (e.g. event a in the empty play of $\mathcal{E}_7$). A strategy which is always guaranteed to be innocent is the "ultra-eager" one, which prescribes to do all prudent events.

Definition 8.3 (Ultra-eager strategy)**.** *We define the* ultra-eager strategy Σ^u_{A} *for* A *as follows:*

$$\Sigma^u_{\mathsf{A}} \;=\; \lambda\sigma.\,\{e \in \pi^{-1}(\mathsf{A}) \mid e \textit{ is prudent in } \sigma\}$$

Lemma 8.4. Σ^u_{A} *is an innocent strategy for* A*.*

Proof. Straightforward by Def. 8.1. □

Recall from Def. 7.61 that an event e is *reachable with past* X (in a CES $\mathcal{E}$) whenever e occurs in some trace with past X. After Lemma 7.60, this means that e occurs in some play $\sigma\eta$ where the prefix σ is a linearization of X, and the overall credits are contained in X (i.e., past debits need not be honoured).

The set $\mathcal{R}^X$ of reachable events with past X can then be alternatively characterised as follows:

$$\mathcal{R}^X \;=\; \{e \notin X \mid \exists \sigma, \eta :\; \overline{\sigma} = X,\; e \in \overline{\eta}, \text{ and } \Gamma(\sigma\eta) \subseteq X\}$$

The following lemma shows that, for conflict-free contracts, the only plays where all participants are innocent are those comprising exactly the reachable events of $\mathcal{E}$.

Lemma 8.5. *Let $\mathcal{C} = \langle \mathcal{E}, \cdots \rangle$ be a conflict-free contract, and let σ be a play of $\mathcal{C}$ where all participants are innocent. Then, $\overline{\sigma} = \mathcal{R}_{\mathcal{E}}$.*

Proof. Special case (for conflict-free CES) of Lemma 7.69(b). □

Theorem 8.6 gives an alternative characterisation of prudent events for conflict-free contracts. An event e is prudent for A in σ whenever $e \in \mathcal{P}^{\overline{\sigma}}$, which is the set of events which are $\vdash$-enabled by $\overline{\sigma}$, or $\Vdash$-enabled by $\overline{\sigma} \cup \mathcal{R}^{\overline{\sigma}}$. Indeed, this criterion is much simpler than the mutually coinductive definition of prudence in Def. 8.1.

Theorem 8.6. *For a set $X \subseteq E$, let*

$$\mathcal{P}^X \;=\; \{e \notin X \,|\, X \vdash e \text{ or } X \cup \mathcal{R}^X \Vdash e\}$$

Then, e is prudent in σ iff $e \in \mathcal{P}^{\overline{\sigma}}$.

Proof. ($\Leftarrow$) We exploit the coinduction proof principle, by defining a functor F such that $\operatorname{gfp} F$ is the coinductively defined set of prudent events. The coinduction proof principle states that:

$$x \sqsubseteq F(x) \implies x \sqsubseteq \operatorname{gfp} F$$

Intuitively, we will obtain the thesis if, taking for x the set $\mathcal{P}^{\overline{\sigma}}$ we manage to prove that $x \sqsubseteq F(x)$. More precisely, to fit the coinductive schema we consider an endofunctor F with the following type:

$$F : \wp(E \times E^*) \times \wp(\mathcal{A} \times E^*) \to \wp(E \times E^*) \times \wp(\mathcal{A} \times E^*)$$

and we define $(P, I) \sqsubseteq (P', I')$ iff $P \subseteq P'$ and $I \supseteq I'$. Define the predicate ϕ as follows:

$$\phi(\sigma, \sigma', \mathsf{A}) \;\triangleq\; \exists k > |\sigma|.\; \Gamma(\sigma'_k) \cap \pi^{-1}(\mathsf{A}) \;\subseteq\; \Gamma(\sigma)$$

Then, for $P \subseteq E \times E^*$ and $I \subseteq \mathcal{A} \times E^*$, we define $F(P, I) = (P', I')$ as follows:

$$\begin{aligned} P' = \{(e, \sigma) \mid\; & \exists \Sigma.\ \sigma e \text{ conform to } \Sigma \wedge \\ & \forall \sigma' = \sigma\, e\, \eta \text{ fair conform to } \Sigma.\ \big(\forall \mathsf{B} \neq \pi(e).\ (\mathsf{B}, \sigma') \in I\big) \\ & \implies \phi(\sigma, \sigma', \mathsf{A})\} \\ I' = \{(\mathsf{A}, \sigma) \mid\; & \forall e \in \pi^{-1}(\mathsf{A}).\ \forall i \geq 0.\ \exists j \geq i.\ (e, \sigma_j) \notin P\} \end{aligned}$$

To apply the coinduction proof principle, let:

$$\begin{aligned} P &= \{(e, \sigma) \mid e \in \mathcal{P}^{\overline{\sigma}}\} \\ I &= \{(\mathsf{A}, \sigma) \mid \mathcal{P}^{\overline{\sigma}} \cap \pi^{-1}(\mathsf{A}) = \emptyset\} \end{aligned}$$

The proof of $(P, I) \sqsubseteq F(P, I)$ proceeds then as follows.
We first show $P \subseteq P'$. Let $(e, \sigma) \in P$, let $\sigma = \langle e_0 \cdots e_n \rangle$, and let $\mathsf{A} = \pi(e)$. The choice of the strategy Σ in P' is made as follows. For all η, let:

$$\Sigma_{\mathsf{A}}(\eta) \;=\; \begin{cases} \{e_i\} & \text{if } \eta = \sigma_i \text{ and } \pi(e_i) = \mathsf{A} \\ \mathcal{P}^{\overline{\eta}} \cap \pi^{-1}(\mathsf{A}) & \text{otherwise} \end{cases}$$

Note that, since the contract is conflict-free, then Σ_{A} is well-defined; also, by construction σ conforms to Σ_{A} (in other words, Σ_{A} has past σ). Let σ' be a fair play extending σ, conform to Σ_{A}, and such that $(\mathsf{B}, \sigma') \in I$ for all $\mathsf{B} \neq \mathsf{A}$. Since $(e, \sigma) \in P$, then $e \in \mathcal{P}^{\overline{\sigma}}$. There are the following two cases:

- $\overline{\sigma} \vdash e$. In this case we have $\Gamma(\sigma e) \subseteq \Gamma(\sigma)$, hence $\phi(\sigma, \sigma', \mathsf{A})$ holds with $k = |\sigma| + 1$.
- $\overline{\sigma} \cup \mathcal{R}^{\overline{\sigma}} \Vdash e$. Since the contract is conflict-free and for all $\mathsf{B} \neq \mathsf{A}$, $(\mathsf{B}, \sigma') \in I$, then it must be $\mathcal{P}^{\overline{\sigma'}} \cap \pi^{-1}(\mathsf{B}) = \emptyset$, for all $\mathsf{B} \neq \mathsf{A}$. Furthermore, since σ' is fair for Σ_{A}, by definition of Σ_{A} it must also be $\mathcal{P}^{\overline{\sigma'}} \cap \pi^{-1}(\mathsf{A}) = \emptyset$. Summing up, $\mathcal{P}^{\overline{\sigma'}} = \emptyset$. We now prove that $\overline{\sigma} \cup \mathcal{R}^{\overline{\sigma}} \subseteq \overline{\sigma'}$. By contradiction, assume that there exists an event e_1 such that $e_1 \in \overline{\sigma} \cup \mathcal{R}^{\overline{\sigma}}$ but $e_1 \notin \overline{\sigma'}$. Clearly, σ is a prefix of σ', so it must be $e_1 \in \mathcal{R}^{\overline{\sigma}}$. By definition of $\mathcal{R}^{\overline{\sigma}}$, there exist η, η' such that $\overline{\eta} = \overline{\sigma}$, $e_1 \in \overline{\eta'}$, $\Gamma(\eta\eta') \subseteq \overline{\eta} = \overline{\sigma}$ — and indeed $\overline{\eta'} \subseteq \mathcal{R}^{\overline{\sigma}}$. Let e_0 be the first event in η' such that $e_0 \notin \overline{\sigma'}$. Then, $\eta\eta' = \eta\eta_0 e_0 \eta_1$ for some η_0, η_1 such that $\eta_0 e_0 \eta_1 = \eta'$. Clearly, $\overline{\eta\eta_0} \subseteq \overline{\sigma'}$. We have the following two sub-cases:
  - $\overline{\eta\eta_0} \vdash e_0$. If this were the case, we would have $e_0 \in \mathcal{P}^{\overline{\sigma'}}$, which contradicts the fact that $\mathcal{P}^{\overline{\sigma'}} = \emptyset$.
  - $\overline{\eta\eta'} \Vdash e_0$. Since $\overline{\eta\eta'} \subseteq \overline{\sigma} \cup \mathcal{R}^{\overline{\sigma}}$, then by saturation $\overline{\sigma} \cup \mathcal{R}^{\overline{\sigma}} \Vdash e_0$. Hence we would have $e_0 \in \mathcal{P}^{\overline{\sigma}}$, which implies $e_0 \in \mathcal{P}^{\overline{\sigma'}} = \emptyset$.

  In both cases, we have deduced a contradiction — therefore $\mathcal{R}^{\overline{\sigma}} \subseteq \overline{\sigma'}$. Summing up, by saturation we have that $\overline{\sigma'} \Vdash e$, hence $e \notin \Gamma(\sigma') \cap \pi^{-1}(\mathsf{A})$. By repeating the same argument for all the events in σ', we obtain $\Gamma(\sigma') \subseteq \Gamma(\sigma)$. This proves that $\phi(\sigma, \sigma', \mathsf{A})$, from which we obtain the thesis $P \subseteq P'$.

We now prove that $I \supseteq I'$. Actually, we shall prove the contrapositive, i.e. whenever $(\mathsf{A}, \sigma) \notin I$, it must be $(\mathsf{A}, \sigma) \notin I'$. Let $(\mathsf{A}, \sigma) \notin I$. By definition of I, there must exist some $e \in \pi^{-1}(\mathsf{A})$ such that $e \in \mathcal{P}^{\overline{\sigma}}$. Let $i = |\overline{\sigma}| - 1$. Then, for all $j \geq i$, $e \in \mathcal{P}^{\overline{\sigma_j}}$ (indeed, since $\sigma_i = \sigma$ we can only have $j = i$). By definition of P, this amounts to say that $(e, \sigma_j) \in P$. In conclusion, we have found an event $e \in \pi^{-1}(\mathsf{A})$ for which there exists some i such that, for all $j \geq i$, $(e, \sigma_j) \in P$. By definition of I', this proves that $(\mathsf{A}, \sigma) \notin I'$.
($\Rightarrow$) Assume that e is prudent for A in σ. We must prove that $e \in \mathcal{P}^{\overline{\sigma}}$. For all participants B, consider the greatest prudent strategy Σ^p_{B}. Clearly, we can pick a fair trace $\sigma' = \sigma\, e\, \nu$ such that ν conforms to *all* the strategies Σ_{B}. By fairness and by definition of innocence, all participants are innocent in σ'. We now prove that $\Gamma(\sigma') \subseteq \Gamma(\sigma)$. Let $\sigma' = \sigma\langle e_0 \cdots e_n \rangle$. By contradiction, assume that for some e_i (say, of participant B), $e_i \in \Gamma(\sigma')$ but $e_i \notin \Gamma(\sigma)$. Since all events in $e\,\nu$ are prudent, then by Def. 8.1:

$$\exists k > |\sigma_i|.\, \Gamma(\sigma'_k) \cap \pi^{-1}(\mathsf{B}) \;\subseteq\; \Gamma(\sigma_i) \subseteq \overline{\sigma_i} \not\ni e_i$$

That is, each event taken on credit is eventually removed from the credits, and thus contradicting $e_i \in \Gamma(\sigma')$. By $\Gamma(\sigma') \subseteq \Gamma(\sigma)$ and by the definition of $\mathcal{R}$, it follows that $\overline{e\nu} \subseteq \mathcal{R}^{\overline{\sigma}}$. Now there are two cases. If $\overline{\sigma} \vdash e$, then we trivially have the thesis. Otherwise, it must be the case that $\overline{\sigma'} \Vdash e$. Since $\overline{\sigma'} = \overline{\sigma e \nu} \subseteq \overline{\sigma} \cup \mathcal{R}^{\overline{\sigma}}$, by saturation we conclude that $\overline{\sigma} \cup \mathcal{R}^{\overline{\sigma}} \Vdash e$. Therefore, $e \in \mathcal{P}^{\overline{\sigma}}$. □

In case of conflict-free contracts, the following lemma states that the events in $\mathcal{P}^X_{\mathcal{E}}$ are exactly the *urgent* events with past X.

Lemma 8.7. *For all conflict-free contract $\mathcal{C} = (\mathcal{E}, \pi, \mathcal{A}, \Phi)$, and for all $X \subseteq E$,*

$$\mathcal{P}^X_{\mathcal{E}} \;=\; \mathcal{U}^X_{\mathcal{E}}$$

Proof. By Lemma 7.72, $\mathcal{U}^X_{\mathcal{E}} = \hat{\mathcal{U}}^X_{\mathcal{E}} \setminus X$. The thesis follows directly from Def. 7.71. □

The following lemma shows that the ultra-eager strategy Σ^u_{A} is prudent for conflict-free contracts. Then, Theorem 8.6 provides us with a simple criterion for constructing a prudent strategy: it suffices to perform in a state $\overline{\sigma}$ the events in $\mathcal{P}^{\overline{\sigma}}$.

Lemma 8.8. *For a conflict-free contract $\mathcal{C}$, the strategy Σ^u_{A} is prudent for A in $\mathcal{C}$.*

Proof. In a fair play $\sigma = \langle e_0 \cdots e_n \rangle$ where A does all her prudent events and all other participants are innocent, also A is innocent. By Lemma 8.5, σ contains exactly the reachable events $\mathcal{R}^{\emptyset}$. By Theorem 8.6, for all i, e_i is prudent in σ_i if either $\overline{\sigma_i} \vdash e_i$ of $\mathcal{R}^{\emptyset} \Vdash e_i$. In the first case, e_i does not augment the credits; in the second case, if e_i is taken on credit then the credit is eventually honoured, because $\overline{\sigma} = \mathcal{R}^{\emptyset}$. Therefore, $\Gamma(\sigma) = \emptyset$. □

Example 8.9. *Consider the* CES $\mathcal{E}_{11}$ *and assume that all the events belong to the same participant* A. *Event a is prudent in ε since* A *has a strategy to honour it, by performing b. Both events b and c are prudent in $\langle a \rangle$ since they are $\vdash$-enabled by a. Nevertheless Σ^u_{A} is not a prudent strategy because there exists a play $\sigma = \langle a\ c \rangle$ conform to it, where a has not been honoured — hence $\Gamma(\sigma) \neq \emptyset$.*

8.2 Agreements

We now refine the notion of winning strategy given in Def. 6.25. The items are similar to the corresponding items in Def. 6.25, except that the definitions of innocence now takes into account the events performed on credit.

Definition 8.10 (Winning play)**.** *Define the function $\widetilde{\mathcal{W}}$ as follows:*

$$\widetilde{\mathcal{W}}\mathsf{A}\sigma = \begin{cases} \Phi\mathsf{A}\sigma & \mathsf{A} \textit{ is credit-free and all participants are innocent in } \sigma \\ +1 & \textit{if } \mathsf{A} \textit{ is innocent, and some } \mathsf{B} \neq \mathsf{A} \textit{ is culpable in } \sigma \\ -1 & \textit{otherwise} \end{cases}$$

where we say that A *is* credit-free *in* σ *iff*

$$\forall e \in \pi^{-1}(\mathsf{A}).\ \forall i \geq 0.\ \exists j \geq i.\ e \notin X_j^\sigma$$

The notions of winning/losing play/strategy, agreement and protection are the same as in Sect. 6.1, except that $\widetilde{\mathcal{W}}$ is now used in place of $\mathcal{W}$.

Lemma 8.11. *Let* Σ_A *be a prudent strategy for* A*. For all fair plays* σ *conform to* Σ_A*, either* A *is credit-free in* σ*, or some* $\mathsf{B} \neq \mathsf{A}$ *is culpable in* σ*.*

Proof. Straightforward by Def. 8.1. □

Example 8.12. *Consider the contracts* $\mathcal{C}_{\mathcal{E}_i}$ *where the obligations are specified by the* CES $\mathcal{E}_1$*,* $\mathcal{E}_2$*,* $\mathcal{E}_7$*,* $\mathcal{E}_8$*,* $\mathcal{E}_{10}$ *in Figures 3.1 and 7.1. Let the goals of* A *and* B *be as follows:* A *is happy when she obtains* b*, while* B *is happy when he obtains* a*.*

$$\Phi\mathsf{A} = \{\sigma \mid b \in \overline{\sigma}\}$$
$$\Phi\mathsf{B} = \{\sigma \mid a \in \overline{\sigma}\}$$

We have that:

- $\mathcal{C}_{\mathcal{E}_1}$ *admits an agreement. The winning strategies for* A *and* B *are, respectively,*

$$\Sigma_\mathsf{A}(\sigma) = \begin{cases} \{a\} & \text{if } a \notin \overline{\sigma} \\ \emptyset & \text{otherwise} \end{cases} \qquad \Sigma_\mathsf{B}(\sigma) = \begin{cases} \{b\} & \text{if } a \in \overline{\sigma} \text{ and } b \notin \overline{\sigma} \\ \emptyset & \text{otherwise} \end{cases}$$

  Roughly, the only fair play conform to Σ_A *and* Σ_B *where both* A *and* B *are innocent is* $\sigma = \langle ab \rangle$*. We have that* A *and* B *win in* σ*, because both participants are credit-free in* σ *(see Ex. 7.47), and* $\sigma \in \Phi\mathsf{A} \cap \Phi\mathsf{B}$*.*

- $\mathcal{C}_{\mathcal{E}_2}$ *does not admit an agreement. Indeed, there are no prudent events in* ε*, hence both* A *and* B *are innocent in* ε*. If no participant takes the first step, then nobody reaches her goals. If a participant takes the first step, then the resulting trace is not credit-free. Thus, no winning strategy exists.*

- $\mathcal{C}_{\mathcal{E}_7}$ *admits an agreement. The winning strategies are as for* $\mathcal{C}_1$ *above:* A *first does* a*, then* B *does* b*. While* $\mathcal{C}_1$ *and* $\mathcal{C}_3$ *are identical from the point of view of agreements, they differ in that* $\mathcal{C}_3$ protects A*, while* $\mathcal{C}_1$ *does not. Intuitively, the enabling* $\vdash a$ *in* $\mathcal{C}_1$ *models an obligation for* A *also in those contexts where no agreement exists, while* $b \Vdash a$ *only forces* A *to do* a *when* b *is guaranteed.*

- $\mathcal{C}_{\mathcal{E}_8}$ *admits an agreement. In this case the winning strategies for* A *and* B *are:*

$$\Sigma_\mathsf{A}(\sigma) = \begin{cases} \{a\} & \text{if } a \notin \overline{\sigma} \\ \emptyset & \text{otherwise} \end{cases} \qquad \Sigma_\mathsf{B}(\sigma) = \begin{cases} \{b\} & \text{if } b \notin \overline{\sigma} \\ \emptyset & \text{otherwise} \end{cases}$$

  That is, a participant must be ready to do her action without waiting the other participant to make the first step.

- $\mathcal{C}_{\mathcal{E}_{11}}$ *does not admit an agreement. Since no events are prudent in* ε*, both participants are innocent in* ε*, but if they cannot reach their goals by doing nothing. If* A *does* a*, then* B *can* b *can choose to do* c*. This makes* B *innocent (and winning), but then* A *loses, because not credit-free in* $\langle ac \rangle$*.*

Recall from Def. 6.26 that, when a contract admits an agreement, all participants have a winning strategy. A relevant question is then how to construct a winning strategy for each participant. We answer this question in Theorem 8.13 below, where we consider the strategy obtained by following the order of events in $\mathcal{P}^{\overline{\sigma}}$. This strategy is prudent for A, and leads A to a winning play whenever A agrees on $\mathcal{C}$.

Theorem 8.13. *Let* $\mathcal{C} = (\mathcal{E}, \pi, \mathcal{A}, \Phi)$ *be a conflict-free contract, and let the strategy* Σ_{A} *be defined as:*

$$\Sigma_{\mathsf{A}}(\sigma) \;=\; \mathcal{P}^{\overline{\sigma}}_{\mathcal{E}} \,\cap\, \pi^{-1}(\mathsf{A})$$

Then, Σ_{A} *is a prudent strategy for* A *in* $\mathcal{C}$*. Moreover, if* Φ *is a reachability payoff and* $\mathcal{C}$ *admits an agreement, then* Σ_{A} *is winning for* A*.*

Proof. By Theorem 8.6, $\mathcal{P}^{\overline{\sigma}}_{\mathcal{E}}$ contains all and only the prudent events in σ. Thus, by Lemma 8.4, Σ_{A} is a prudent strategy for A. For the second part, assume that $\mathcal{C}$ admits an agreement, and that Φ is a reachability payoff induced by the predicate φ^1, φ^0. Let σ' be a play where all the participants win: then, $\overline{\sigma'} \in \varphi^1$. By Lemma 8.5, $\sigma' = \mathcal{R}^{\emptyset}$. Now, let σ be a fair play conform to Σ_{A}. If some $\mathsf{B} \neq \mathsf{A}$ is culpable, then A wins. Otherwise, by Lemma 8.5, Then, $\overline{\sigma} = \mathcal{R}^{\emptyset} = \overline{\sigma'} \in \varphi^1$, from which we conclude that A wins in σ. □

8.3 Protection

In this section we show that CES-based contracts allow for both agreements and protection in contracts with circular finite O-R payoffs. Before presenting the formal results, we give some intuition through our working example.

Example 8.14. *In Example 6.42 we have shown that the contract* $\mathcal{C}_{\mathsf{A}}$ *protects Alice, while* $\mathcal{C}_{\mathsf{B}}$ *does not protect Bob. Suppose now to change Bob's contract into a contract* $\mathcal{C}'_{\mathsf{B}}$ *where Bob relaxes his requirements. The contract* $\mathcal{C}'_{\mathsf{B}}$ *differs from* $\mathcal{C}_{\mathsf{B}}$ *only in the event structure* $\mathcal{E}'_{\mathsf{B}}$*, which contains exactly one circular enabling:* $\{a\} \Vdash b$*. Similarly to Example 6.27, the contract* $\mathcal{C}_{\mathsf{A}} \mid \mathcal{C}'_{\mathsf{B}}$ *admits an agreement. To show that, let* Σ_{A} *and* Σ_{B} *be the following strategies for* A *and* B*, respectively:*

$$\Sigma_{\mathsf{A}}(\sigma) = \begin{cases} \{a\} & \text{if } b \in \overline{\sigma} \text{ and } a \notin \overline{\sigma} \\ \emptyset & \text{otherwise} \end{cases} \qquad \Sigma_{\mathsf{B}}(\sigma) = \begin{cases} \{b\} & \text{if } b \notin \overline{\sigma} \\ \emptyset & \text{otherwise} \end{cases}$$

Roughly, the only fair play which conforms to Σ_{A} *and* Σ_{B} *where both* A *and* B *are innocent is* $\sigma = \langle ba \rangle$*, which gives rise to the following trace in* $\mathrm{LTS}_{\mathcal{E}}$*:*

$$(\emptyset, \emptyset) \;\xrightarrow{b}\; (\{b\}, \{b\}) \;\xrightarrow{a}\; (\{a, b\}, \emptyset)$$

We have that A *and* B *win in* σ*, because* $\widetilde{\mathcal{W}}\mathsf{A}\sigma = 1 = \widetilde{\mathcal{W}}\mathsf{B}\sigma$*. Thus,* Σ_{A} *and* Σ_{B} *are winning strategies for* A *and* B*, respectively, and so* $\mathcal{C}$ *admits an agreement.*

Differently from the contract $\mathcal{C}_{\mathsf{B}}$ *in Ex. 6.42, the contract* $\mathcal{C}'_{\mathsf{B}}$ *protects Bob. Let* $\mathcal{C}'$ *be a contract compatible (Def 6.16) with* $\mathcal{C}'_{\mathsf{B}}$*. Consider the ultra-eager strategy* Σ^u_{B} *for* B *(Def 8.3). Let* ν *be a fair play of* $\mathcal{C}'_{\mathsf{B}} \mid \mathcal{C}'$ *conform to* Σ^u_{B}*. By contradiction, assume that* B *loses in* ν*. By Lemma 8.4,* B *is innocent in* ν*, and so it must be* $\Phi\mathsf{B}\nu < 0$*. By definition, the payoff of* B *is negative only when* $b \in \overline{\nu}$ *and* $a \notin \overline{\nu}$*. Assume that* $\nu = \eta\, b\, \eta'$*. By definition of* Σ^u_{B}*, the event* b *was prudent in* η*, and we have the transition* $(\overline{\eta}, X^0) \xrightarrow{b} (\overline{\eta} \cup \{b\}, X^0 \cup \{b\})$*. After* B *has performed* b*, its only strategy is the empty one. By Def. 8.1, for all plays* $e_0 e_1 \cdots$ *starting from* $(\overline{\eta} \cup \{b\}, X^0 \cup \{b\})$*, there exists some* $k > 0$ *such that* $b \notin X^k$*. This means that* b *has been honoured, and the only way to do that is to perform* a*. Therefore,* $a \in \overline{\nu}$ *— contradiction.* □

We now construct a CES from an O-R payoff with finite responses. For all clauses (O, R), the CES contains the enablings $R \Vdash O$. Lemma 8.16 below reveals a key feature of circularity: the CES obtained from a circular O-R payoff has a configuration which comprises all the response sets for all the participants. Together with Theorem 8.17, this will allow for constructing a contract which admits an agreement. Theorems 8.17 and 8.18 are the CES counterpart of Theorems 6.39 and 6.45 for ES-based contracts, respectively.

Definition 8.15 (Synthesis of CES from O-R payoffs)**.** *For an O-R payoff* Φ *with clauses* $(O^i, R^i)_i$ *and finite* R^i*, define* $\mathcal{E}(\Phi)$ *as the conflict-free* CES *with (saturated) enablings* $\{R^i \Vdash O^i\}_i$*.*

Lemma 8.16. *Let* Φ *be a finite circular O-R payoff for* $\mathcal{A}$ *such that* $\Phi\mathsf{A} = \lambda\sigma.\, \phi_{\mathsf{A}}\overline{\sigma}$ *for all* $\mathsf{A} \in \mathcal{A}$*. Then,* $\exists C \in \mathcal{F}_{\mathcal{E}(\Phi)}.\ \forall \mathsf{A} \in \mathcal{A}.\ \bigcup_i R^i_{\mathsf{A}} \subseteq C$*.*

Proof. By Def. 8.15, the CES $\mathcal{E}(\Phi)$ is conflict-free, and its minimal enablings have the form $R^i_{\mathsf{A}} \Vdash O^i_{\mathsf{A}}$, for all $\mathsf{A} \in \mathcal{A}$ and for all i. Consider the set C containing all and only the events in the minimal enablings of $\mathcal{E}(\Phi)$. Since Φ has a finite number of offers-requests, and each of them is finite, then C is finite. We shall prove that $C \in \mathcal{F}_{\mathcal{E}(\Phi)}$. Since C is finite, by Lemma 7.15 it suffices to find some trace $\sigma \in \mathcal{T}_{\mathcal{E}(\Phi)}$ such that $\overline{\sigma} = C$. Consider an arbitrary ordering σ of the events of C. For all $e \in \overline{\sigma}$, we now prove that $\overline{\sigma} \Vdash e$. By Def. 7.5, this will allow to conclude that σ is a trace of $\mathcal{E}(\Phi)$. Let $e \in \overline{\sigma}$. We have the following two cases:

- $e \in O^i_{\mathsf{A}}$, for some i and $\mathsf{A} \in \mathcal{A}$. Since $R^i_{\mathsf{A}} \Vdash O^i_{\mathsf{A}} \in \mathcal{E}(\Phi)$ and all events in R^i_{A} occur in σ, then $\overline{\sigma} \Vdash e$.
- $e \in R^i_{\mathsf{A}}$, for some i and $\mathsf{A} \in \mathcal{A}$. By circularity (Def. 6.13), there exist j and B such that $e \in O^j_{\mathsf{B}}$. By Def. 8.15, $R^j_{\mathsf{B}} \Vdash O^j_{\mathsf{B}} \in \mathcal{E}(\Phi)$, and the proof proceeds as in the previous case.

Summing up, we have found that $C \in \mathcal{F}_{\mathcal{E}(\Phi)}$. By construction, C contains all the requests R^i_{A}, for all i and A. □

Theorem 8.17. *Let $\mathcal{C}$ be a contract with O-R payoff for A. If $\mathcal{E}$ is conflict-free and $\vdash$-free, and $\bigcup_i R^i_{\mathsf{A}} \subseteq C$ for some $C \in \mathcal{F}_{\mathcal{E}}$, then A agrees on $\mathcal{C}$.*

Proof. Let $\mathcal{C}$ be a contract with O-R payoffs for A. Let $C \in \mathcal{F}_{\mathcal{E}}$ be such that $\bigcup_i R^i_{\mathsf{A}} \subseteq C$. We will prove that the prudent strategy Σ^u_{A} is winning for A in $\mathcal{C}$. Let γ be a fair play of $\mathcal{C}$ which conforms to Σ^u_{A}.

By contradiction, assume that A is *not* winning in γ. By Lemma 8.4, A is innocent in γ. By Lemma 8.11, A is credit-free in γ. Thus, by Def. 8.10 it follows that all participants are innocent and $\Phi\mathsf{A}\gamma \leq 0$. By Def. 6.11, this means that either there exists some i such that $O^i_{\mathsf{A}} \subseteq \overline{\gamma}$ and $R^i_{\mathsf{A}} \not\subseteq \overline{\gamma}$ (in case A loses), or for all i, $O^i_{\mathsf{A}} \not\subseteq \overline{\gamma}$ and $R^i_{\mathsf{A}} \not\subseteq \overline{\gamma}$ (in case A non-loses). In both cases, there exists at least one i such that $R^i_{\mathsf{A}} \not\subseteq \overline{\gamma}$.

Let i be such that $R^i_{\mathsf{A}} \not\subseteq \overline{\gamma}$, and let e be such that $e \in R^i_{\mathsf{A}} \setminus \overline{\gamma}$. By hypothesis, there exists $C \in \mathcal{F}_{\mathcal{E}}$ such that $\bigcup_i R^i_{\mathsf{A}} \subseteq C$, hence $e \in C$. Since C is a configuration, it enjoys finiteness (Theorem 7.29), i.e. there exists $C_0 \subseteq_{fin} C$ such that $e \in C_0 \in \mathcal{F}_{\mathcal{E}}$. Since $\mathcal{E}$ is $\vdash$-free, it must be $C_0 \Vdash C_0$, i.e. all the events in C_0 are circularly enabled by C_0 itself.

We will prove that $C_0 \subseteq \overline{\gamma}$. Let $\gamma = \langle e_0\, e_1 \cdots \rangle$. Since all participants are innocent in γ, by Theorem 8.6 it must be:

$$\forall i \geq 0.\ \forall e.\ \left(e \in \mathcal{R}^{\overline{\sigma_i}}_{\mathcal{E}} \implies \exists j \geq i.\ e_j = e\right) \tag{8.1}$$

Now, since $e \in C_0 \in \mathcal{F}_{\mathcal{E}}$, by definition of reachable events we have $e \in \mathcal{R}^{\emptyset}_{\mathcal{E}}$, hence by (8.1) it follows that $e \in \overline{\gamma}$ — contradiction.

Summing up, we have proved that $\bigcup_i R^i_{\mathsf{A}} \subseteq \overline{\gamma}$ for all fair plays γ. Therefore, A has a winning strategy (Σ^u_{A}) in $\mathcal{C}$, and so we conclude that A agrees on $\mathcal{C}$.

□

Theorem 8.18. *For a finite conflict-free* CES *$\mathcal{E}$ and an O-R payoff Φ for A, the contract $\langle \mathcal{E}, \mathcal{A}, \pi, \Phi \rangle$ protects A if:*

$$\forall i, Y.\ \ (\forall e \in O^i_{\mathsf{A}}.\ Y \vdash e \ \vee\ Y \Vdash e) \implies R^i_{\mathsf{A}} \ \subseteq\ Y$$

Proof. Let $\mathcal{C}_{\mathsf{A}}$ be a contract with O-R payoff for A such that:

$$\forall i, Y.\ \ Y \rhd O^i_{\mathsf{A}} \implies R^i_{\mathsf{A}} \ \subseteq\ Y \tag{8.2}$$

where we write $Y \rhd O^i_{\mathsf{A}}$ as a shorthand for:

$$\forall e \in O^i_{\mathsf{A}}.\ Y \vdash e \ \vee\ Y \Vdash e$$

Let $\mathcal{E}$ be the CES of contract $\mathcal{C}_{\mathsf{A}}$, and let $\mathcal{C}$ be a contract compatible (Def 6.16) with $\mathcal{C}_{\mathsf{A}}$. Choose for A the strategy Σ^u_{A} which enables all and only the prudent events. To prove that $\mathcal{C}_{\mathsf{A}}$ protects A, we will prove that Σ^u_{A} is a non-losing strategy for A in $\mathcal{C}_{\mathsf{A}} \mid \mathcal{C}$. Let σ be a fair play of $\mathcal{C}_{\mathsf{A}} \mid \mathcal{C}$ conform to Σ^u_{A}. By Lemma 8.4, A is innocent in σ. By contradiction, assume that A loses in σ, i.e. by Def. 6.25 and by Def. 6.11:

$$\exists i.\ \ O^i_{\mathsf{A}} \subseteq \overline{\sigma} \ \wedge\ R^i_{\mathsf{A}} \not\subseteq \overline{\sigma} \tag{8.3}$$

Since $\mathcal{E}$ is finite and conflict-free, then by Lemma 8.8 the strategy Σ^u_{A} is prudent. Let $\sigma = \langle e_0\, e_1 \cdots \rangle$, and let $e_j \in O^i_{\mathsf{A}}$. Since e_j is prudent in σ_j, it must have been enabled by a $\vdash$ or by a $\Vdash$. There are the following two cases:

- $Z_j \vdash e_j$, for some $Z_j \subseteq \overline{\sigma_j}$. Then, $\overline{\sigma} \vdash e_j$.
- $Z_j \Vdash e_j$, for some Z_j. Since Σ^u_{A} is prudent, then by Lemma 8.11 A is credit-free in σ, and so $Z_j \subseteq \overline{\sigma}$.

Summing up, $\bigcup\{Z_j \mid e_j \in O^i_{\mathsf{A}}\} \subseteq \overline{\sigma} \rhd O^i_{\mathsf{A}}$ Therefore, by (8.2) it follows that $R^i_{\mathsf{A}} \subseteq \overline{\sigma}$ — which contradicts (8.3).

□

Theorem 8.19 below states that agreement and protection can coexist in CES-based contracts with circular finite O-R payoffs. Recall that Theorem 6.49 excluded this possibility for ES-based contracts. Condition (8.4) in Theorem 8.19 is technical, yet it makes the theorem applicable to a broad class of contracts with O-R payoffs (e.g. the dining retailers scenario, see Ex. 7.8). When condition (8.4) is not satisfied, Theorem 8.19 does not hold in general.

To give an example of the role played by condition (8.4), consider the following O-R payoff Φ_{A} of participant A defined by:

$$\begin{array}{lll} O^0 = \{a_0, a_1\} & O^1 = \{a_1, a_2\} & O^2 = \{a_0, a_2\} \\ R^0 = \{b_0\} & R^1 = \{b_1\} & R^2 = \{b_2\} \end{array}$$

for which condition (8.4) is *not* satisfied.

The event structure $\mathcal{E}(\Phi_{\mathsf{A}})$ obtained by this payoff contains the enablings $\{b_0\} \Vdash \{a_0, a_1\}$, $\{b_1\} \Vdash \{a_1, a_2\}$, and $\{b_2\} \Vdash \{a_0, a_2\}$.

This means that, were participant B performing b_0 (respectively b_1), then A were asked to perform a_0 and a_1 (resp. a_1, a_2) in return; which is perfectly correct from A's point of view. Instead, were participant B performing both b_0 and b_1, then A were obliged to do not only a_0, a_1 but *also* a_2; which is not what A intended to do. According to Φ_{A}, by performing a_0 and a_2 A wanted to have b_2 in return, which is not the case: so A loses. What is wrong with a payoff of this kind is the fact that an attacker can perform a particular set of actions to take advantage of the other participant: we can conclude that such a payoff function do not allow A to be protected.

Suppose now to change Φ_{A}, by requiring $R^2 = \{b_0, b_1\}$. The modified payoff now satisfies condition (8.4) and the previous attack no longer apply. Indeed, the event structure obtained by this payoff contains the enablings $\{b_0\} \Vdash \{a_0, a_1\}$, $\{b_1\} \Vdash \{a_1, a_2\}$, and $\{b_1\} \Vdash \{a_0, a_2\}$, and $\{b_0\} \Vdash \{a_0, a_2\}$. Were participant B performing b_0 and b_1, then A were asked to do a_0,a_1, and a_2; which, according to Φ_{A}, leads A to win.

Theorem 8.19. *Let $\Phi_1, \ldots, \Phi_n$ be finite circular O-R payoffs for $\mathsf{A}_1, \ldots, \mathsf{A}_n$, respectively, and such that, for all $\mathsf{A} \in \{\mathsf{A}_1, \ldots, \mathsf{A}_n\}$:*

$$\forall P \subseteq \mathbb{N}.\ \forall j.\ O^j_{\mathsf{A}} \subseteq \textstyle\bigcup_{i \in P} O^i_{\mathsf{A}} \implies R^j_{\mathsf{A}} \subseteq \bigcup_{i \in P} R^i_{\mathsf{A}} \tag{8.4}$$

Then, there exist contracts $\mathcal{C}_i = \langle \mathcal{E}_i, \mathcal{A}, \pi, \Phi_i \rangle$ for $i \in 1..n$ such that:

(a) $\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n$ admits an agreement;

(b) for all $i \in 1..n$, $\mathcal{C}_i$ protects A_i.

(c) for all plays σ of $\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n$, $\forall e \in \overline{\sigma}.\ \exists i.\ e \in O^i_{\pi(e)}$.

Proof. Let $\Phi_1, \ldots, \Phi_n$ be finite circular O-R payoffs for participants $\mathsf{A}_1, \ldots, \mathsf{A}_n$, respectively, such that, for all $\mathsf{A} \in \mathcal{A} = \{\mathsf{A}_1, \ldots, \mathsf{A}_n\}$ condition (8.4) holds, i.e.:

$$\forall P \subseteq \mathbb{N}.\ \forall j.\ O^j_{\mathsf{A}} \subseteq \bigcup_{i \in P} O^i_{\mathsf{A}} \implies R^j_{\mathsf{A}} \subseteq \bigcup_{i \in P} R^i_{\mathsf{A}}$$

For all $i \in 1..n$, let $\mathcal{E}_i = \mathcal{E}(\Phi_i)$ be the CES of contract $\mathcal{C}_i$. Let $\mathcal{C} = \mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n$, and let $\Phi = \Phi_1 \sqcup \cdots \sqcup \Phi_n$. By construction, the CES of $\mathcal{C}$ is $\mathcal{E}(\Phi) = \mathcal{E}_1 \sqcup \cdots \sqcup \mathcal{E}_n$. We prove the three items of Theorem 8.19 separately:

(a) By Lemma 8.16, we have that:

$$\exists C \in \mathcal{F}_{\mathcal{E}(\Phi)}.\ \forall \mathsf{A}_p \in \mathcal{A}.\ \bigcup_i R^i_{\mathsf{A}_p} \subseteq C$$

Since $\mathcal{E}(\Phi)$ is conflict-free and $\vdash$-free, by Theorem 8.17 it follows that each A_i agrees with $\mathcal{C}$. Therefore, $\mathcal{C}$ admits an agreement.

(b) We prove that $\mathcal{C}_i$ protects A_i by exploiting Theorem 8.18. Denote the offers-requests of A_i just as $(O^j)_j$ and $(R^j)_j$, respectively. Let Y be such that

$$Y \subseteq \bigcup \{X \mid X \Vdash e \in \mathcal{E}_i \ \wedge\ e \in O^j\} \qquad \text{and} \qquad Y \Vdash O^j$$

Let $O^j = \{e_1, \ldots, e_p\}$. Recall that $Y \Vdash O^j$ stands for $\forall h \in 1..p.\ Y \Vdash e_h$. By construction of $\mathcal{E}(\Phi_i)$:

$$\forall h \in \{1..p\}.\ \exists i_h.\ e_h \in O^{i_h} \ \wedge\ R^{i_h} \subseteq Y$$

Thus, $O^j \subseteq \bigcup_{k \in 1..p} O^{i_k}$. Since condition (8.4) holds by hypothesis, it must be:

$$R^j \ \subseteq \bigcup_{k \in 1..p} R^{i_k} \ \subseteq\ Y$$

We can then apply Theorem 8.18, and deduce that $\mathcal{C}_i$ protects A_i.

(c) Let σ be a play of $\mathcal{C}$. Since plays are traces of $\mathrm{LTS}_{\mathcal{E}(\Phi)}$, they can only contain events e such that $X \circ e \in \mathcal{E}(\Phi)$, for some X and $\circ \in \{\vdash, \Vdash\}$. By Def. 8.15, all enablings of $\mathcal{E}(\Phi)$ have the form $R^i_{\mathsf{A}} \Vdash O^i_{\mathsf{A}}$. Therefore, each event e in σ belongs to $O^i_{\pi(e)}$, for some i. □

□

Example 8.20. *Recall the dining retailers scenario from Ex. 6.15 and Ex. 7.8. The payoff Φ_i of each retailer is a finite O-R circular payoff, and condition* (8.4) *is trivially satisfied. Therefore, Theorem 8.19 allows for constructing contracts which admit an agreement and protects all retailers. The* CES *of contract C_i of retailer A_i has enablings $\{e_{j,i} \mid i \neq j\} \Vdash \{e_{i,j} \mid i \neq j\}$. The idea is simple: A_1 offers his pieces of cutlery, in exchange of the commitment of the other retailers to do the same. Since all retailers commit to the analogous contract, we have an agreement.* □

Chapter 9

A logical view of contracts

The literature proposes a wide and heterogeneous ecosystem of contract models, as discussed in Chapter 5. This makes it hard to clearly establish the differences and the analogies among different models. In particular, there is a gap between the two main paradigms for modelling contracts, i.e. the one which interprets them as interactive multi-agent systems, and the one where contracts are rendered as formulae of suitable logics.

To contribute towards reducing this gap, in this chapter we relate the theory of contracts proposed in Chapter 8 with a logic-based model of contracts. More precisely, we establish a correspondence exists between the fundamental notions in the first contract model (namely, *agreements* and *winning strategies*) and provability in the logic-based model.

The logic model we consider is Propositional Contract Logic (PCL), the main features of which have been reviewed in Chapter 4. PCL extends intuitionistic logic with a new connective ($\twoheadrightarrow$), called "contractual implication". A first observation is that, similarly to the circular enabling $\Vdash$ of CES, this new connective allows for a form of circular assume-guarantee reasoning.

Consider e.g. a participant A who promises to do a provided that she receives b in exchange, and a participant B who, dually, promises to do b in exchange of a. In our CES-based model, these obligations are represented by a CES with enablings:

$$b \Vdash a \qquad a \Vdash b$$

Given the intended payoff functions, this contract admits an agreement. The winning strategies of A and B prescribe both participants to do their events (without waiting the other to take the first step), so leading to a configuration $\{a, b\}$ of the CES. In the logical model, the scenario above is represented by the PCL formula:

$$(b \twoheadrightarrow a) \wedge (a \twoheadrightarrow b)$$

which entails both a and b in PCL. Hence, a connection seems to exist between the agreement property in our contract model and provability in PCL.

In this chapter we formalise this connection, by relating the notions of agreement and winning strategies in the game-theoretic model to that of provability in PCL.

Chapter overview. The rest of this chapter is organised as follows.

In Section 9.1 we prove that, for conflict-free CES, reachability can be reduced to provability in PCL (Theorem 9.3). The proof exploits the inductive characterization for reachable events defined in Section 7.3.1. In Section 9.2 we reduce the problem of deciding if a set of events is a configuration in a CES (possibly with conflicts) to provability in PCL (Theorem 9.9). In Section 9.3 we prove that, for conflict-free CES, urgency can be reduced to provability in PCL (Theorem 9.17). Finally, in Section 9.4 we exploit the above-mentioned reductions to relate contracts with PCL. We show in Theorem 9.18 that agreement in conflict-free contracts corresponds to provability in Horn PCL theories. Theorem 9.19 establishes that the sequences of events respecting the order imposed by proofs in PCL can be projected to winning strategies for all participants in the game-theoretic contract model.

9.1 Reachability via logic

In this section we investigate some basic relations between CES and PCL.

To do that, we shall mainly consider Horn PCL theories, which consist of sets of clauses of the form $\alpha \to a$ or $\alpha \twoheadrightarrow a$, where α is a conjunction of atoms (Def. 4.12).

A first correspondence between CES and PCL can be observed in Fig. 9.1, where we compare the three items of Lemma 7.27 with the rules (Cut), ($\to$L) and (Fix) of the Gentzen-style proof system of PCL.

Consider e.g. Lemma 7.27(a): for all sets of events C,C' and X such that the union of C and C' is conflict-free, if C is an X-configuration and C' is an $(X \cup C)$-configuration, then the union of C and C' is an X-configuration. In other words, we can discharge C from the credit set used to justify $C \cup C'$, provided that C is justified by using only the credit set X. The symmetry with (Cut) rule is quite evident: if we deduce p with hypothesis Δ and we deduce q with hypotheses Δ, p, we can deduce q with hypotheses Δ. The role of Δ is played by the set X, while the role of p is played by the set C.

Note that, under the hypotheses of Lemma 7.27, the stronger thesis $C' \in \mathcal{F}(X)$ does not hold in general. For instance, consider the CES with enablings $a \Vdash b$, $b \Vdash a$, $a \vdash c$. We have that $C = \{a, b\} \in \mathcal{F}$ and $C' = \{a, c\} \in \mathcal{F}(C)$. By Lemma 7.27(a), it then follows that $C \cup C' \in \mathcal{F}$, but C' alone is not a configuration. Similar examples hold for items (b) and (c).

In Def. 9.1 we show an encoding of conflict-free CES into Horn PCL theories, through which we show that the problem of deciding if an event is reachable can be reduced to provability in the logic. The encoding is straightforward: an enabling

$$\frac{C \in \mathcal{F}(X) \quad C' \in \mathcal{F}(X \cup C)}{C \cup C' \in \mathcal{F}(X)} \text{ 7.27(a)} \qquad \frac{\Delta \vdash p \quad \Delta,\ p \vdash q}{\Delta \vdash q} \text{ (Cut)}$$

$$\frac{C \in \mathcal{F}(X) \quad C' \in \mathcal{F}(X \cup Y) \quad C \vdash Y}{C \cup C' \in \mathcal{F}(X)} \text{ 7.27(b)} \qquad \frac{\Delta \vdash p \quad \Delta, q \vdash r \quad p \to q \in \Delta}{\Delta \vdash r} \text{ ($\to$L)}$$

$$\frac{C \in \mathcal{F}(X \cup C') \quad C' \in \mathcal{F}(X \cup Y) \quad C \Vdash Y}{C \cup C' \in \mathcal{F}(X)} \text{ 7.27(c)} \qquad \frac{\Delta, r \vdash p \quad \Delta, q \vdash r \quad p \twoheadrightarrow q \in \Delta}{\Delta \vdash r} \text{ (Fix)}$$

Figure 9.1: Basic relations between CES and PCL.

$$\begin{aligned} [(X_i \circ e_i)_{i \in I}]_{\mathcal{R}} &= \{[X_i \circ e_i]_{\mathcal{R}} \mid i \in I\} \\ [X \circ e]_{\mathcal{R}} &= \left(\bigwedge X\right) [\circ]\ e \end{aligned} \qquad \textit{where } [\circ] = \begin{cases} \to & \text{if } \circ = \vdash \\ \twoheadrightarrow & \text{if } \circ = \Vdash \end{cases}$$

Figure 9.2: Encoding reachable events in Horn PCL theories.

$X \vdash e$ is mapped to the clause $(\bigwedge X) \to e$, while a circular enabling $X \Vdash e$ is mapped to the clause $(\bigwedge X) \twoheadrightarrow e$.

Definition 9.1 (Encoding reachable events in PCL)**.** *For a conflict-free* CES $\mathcal{E} = \langle E, \#, \vdash, \Vdash \rangle$*, the Horn* PCL *theory* $[\mathcal{E}]_{\mathcal{R}}$ *is defined in Fig. 9.2, where* $X \circ e$ *means that* X *is a minimal set of events such that* $(X, e) \in\ \circ$*, for* $\circ \in \{\vdash, \Vdash\}$*.*

The main result of this section is Theorem 9.3 below. It states that an event e is reachable in a conflict-free CES $\mathcal{E}$ if and only if e is provable in the PCL theory $[\mathcal{E}]_{\mathcal{R}}$. Before stating the theorem, we introduce some notation which will be used throughout the rest of this chapter.

Notation 9.2. *For each event in* $e \in E$*, we assume an atom* e *in* PCL*. For a conjunction of atoms* $\alpha = \bigwedge_{i \in I} e_i$*, we write* $\overline{\alpha}$ *for the set* $\{e_i \mid i \in I\}$*. We extend this notation to sets* Δ *of conjunctions of atoms: we write* $\overline{\Delta}$ *for* $\bigcup\{\overline{\alpha} \mid \alpha \in \Delta\}$*, and* Δ, Δ' *for* $\Delta \cup \Delta'$*. Hereafter* α *will range over conjunctions of atoms, and* Δ *will range over sets of atoms.*

Theorem 9.3. *Let* $\mathcal{E}$ *be a finite, conflict-free* CES*. Then, for all* $e \in E$*:*

$$e \in \mathcal{R}_{\mathcal{E}} \iff [\mathcal{E}]_{\mathcal{R}} \vdash e$$

Proof. We prove the following statement. For a conflict-free CES $\mathcal{E}$, for all α and Δ:

$$\overline{\alpha} \subseteq \mathcal{R}_{\mathcal{E}}(\overline{\Delta}) \iff [\mathcal{E}]_{\mathcal{R}}, \Delta \vdash \alpha \tag{9.1}$$

The proof of the main theorem then follows directly from (9.1), with $X = \emptyset$ and $\alpha = e$.

To prove (9.1), note that by Theorem 7.45 we have $\mathcal{R}_\mathcal{E}(\overline{\Delta}) = \hat{\mathcal{R}}_\mathcal{E}(\overline{\Delta})$. Consequently, we will actually prove that $\overline{\alpha} \subseteq \hat{\mathcal{R}}_\mathcal{E}(\overline{\Delta})$ iff $[\mathcal{E}]_\mathcal{R},\ \Delta \vdash \alpha$.

($\Leftarrow$) assume that $[\mathcal{E}]_\mathcal{R},\ \Delta \vdash \alpha$.

By by Lemma 4.14, consider a proof tree Π of $[\mathcal{E}]_\mathcal{R},\ \Delta \vdash \alpha$ without occurrences of the (Cut) rule, and containing only occurrences of the rules (Id), ($\wedge$L1), ($\wedge$L2), ($\wedge$R), ($\rightarrow$L), and (Fix). We proceed by induction on the depth of Π.

The base case concerns the axiom (Id), which gives:

$$\frac{\alpha \in \Delta}{[\mathcal{E}]_\mathcal{R},\ \Delta \vdash \alpha}\ (\textsc{Id})$$

Since $\alpha \in \Delta$, then $\overline{\alpha} \subseteq \overline{\Delta}$, so by Lemmata 7.41(a) and 7.41(b) we have $\overline{\alpha} \subseteq \hat{\mathcal{R}}(\overline{\alpha}) \subseteq \hat{\mathcal{R}}(\overline{\Delta})$.

For the inductive case, we proceed by cases on the last rule used in Π. There are the following exhaustive cases:

- case ($\wedge$L1). We have that $\Delta = \Delta', q \wedge r$ for some Δ', so:

$$\frac{[\mathcal{E}]_\mathcal{R},\ \Delta',\ q \wedge r,\ q\ \vdash\ \alpha}{[\mathcal{E}]_\mathcal{R},\ \Delta',\ q \wedge r\ \vdash\ \alpha}\ (\wedge\text{L1})$$

  By the induction hypothesis, $\overline{\alpha} \subseteq \hat{\mathcal{R}}(\overline{\Delta} \cup \overline{q}) = \hat{\mathcal{R}}(\overline{\Delta})$.

- case ($\wedge$L2) similar to the previous one.

- case ($\rightarrow$L). We have $q \rightarrow a \in [\mathcal{E}]_\mathcal{R}$ for some conjunction of atoms q and atom a, and

$$\frac{[\mathcal{E}]_\mathcal{R},\ \Delta,\ q \rightarrow a\ \vdash\ q \quad [\mathcal{E}]_\mathcal{R},\ \Delta,\ a\ \vdash\ \alpha}{[\mathcal{E}]_\mathcal{R},\ \Delta,\ q \rightarrow a \vdash\ \alpha}\ (\rightarrow\text{L})$$

  By applying the induction hypothesis twice, we have $\overline{q} \subseteq \hat{\mathcal{R}}(\overline{\Delta})$, and $\overline{\alpha} \subseteq \hat{\mathcal{R}}(\overline{\Delta} \cup \{a\})$.

  Since $q \rightarrow a \in [\mathcal{E}]_\mathcal{R}$, by Def 9.1 it must be the case that $\overline{q} \vdash a\ \in \mathcal{E}$. By saturation, $\hat{\mathcal{R}}(\overline{\Delta}) \vdash a$. Thus by ($\vdash_{\hat{\mathcal{R}}}$):

$$\frac{\hat{\mathcal{R}}(\overline{\Delta}) \vdash a}{a \in \hat{\mathcal{R}}(\overline{\Delta})}\ (\vdash_{\hat{\mathcal{R}}})$$

  By Lemma 7.42, $\hat{\mathcal{R}}(\overline{\Delta} \cup \{a\}) = \hat{\mathcal{R}}(\overline{\Delta})$, from which we conclude $\overline{\alpha} \subseteq \hat{\mathcal{R}}(\overline{\Delta})$.

- case (Fix). We have $q \twoheadrightarrow a \in [\mathcal{E}]_\mathcal{R}$ for some conjunction of atoms q and atom a, and

$$\frac{[\mathcal{E}]_\mathcal{R},\ \Delta,\ q \twoheadrightarrow a,\ \alpha\ \vdash\ q \quad [\mathcal{E}]_\mathcal{R},\ \Delta,\ a\ \vdash\ \alpha}{[\mathcal{E}]_\mathcal{R},\ \Delta,\ q \twoheadrightarrow a \vdash\ \alpha}\ (\textsc{Fix})$$

  By applying the induction hypothesis twice, $\overline{q} \subseteq \hat{\mathcal{R}}(\overline{\Delta} \cup \overline{\alpha})$, and $\overline{\alpha} \subseteq \hat{\mathcal{R}}(\overline{\Delta} \cup \{a\})$.

  From the last inclusion, Lemma 7.42 yields $\hat{\mathcal{R}}(\overline{\Delta} \cup \{a\} \cup \overline{\alpha}) = \hat{\mathcal{R}}(\overline{\Delta} \cup \{a\})$. Since $\overline{q} \subseteq \hat{\mathcal{R}}(\overline{\Delta} \cup \overline{\alpha}) \subseteq \hat{\mathcal{R}}(\overline{\Delta} \cup \{a\} \cup \overline{\alpha})$, then we also have $\overline{q} \subseteq \hat{\mathcal{R}}(\overline{\Delta} \cup \{a\})$.

Since $q \twoheadrightarrow a \in [\mathcal{E}]_{\mathcal{R}}$, by Def 9.1, it must be the case that $\overline{q} \Vdash a \ \in \mathcal{E}$. Thus, by saturation $\hat{\mathcal{R}}(\overline{\Delta} \cup \{a\}) \Vdash a$. By rule $(\Vdash_{\hat{\mathcal{R}}})$, we have:

$$\frac{\hat{\mathcal{R}}(\overline{\Delta} \cup \{a\}) \Vdash a}{a \in \hat{\mathcal{R}}(\overline{\Delta})} \ (\Vdash_{\hat{\mathcal{R}}})$$

By Lemma 7.42, $\hat{\mathcal{R}}(\overline{\Delta} \cup \{a\}) = \hat{\mathcal{R}}(\overline{\Delta})$, therefore $\overline{\alpha} \subseteq \hat{\mathcal{R}}(\overline{\Delta})$.

($\Rightarrow$) let $\overline{\alpha} \subseteq \hat{\mathcal{R}}(\overline{\Delta})$, and let $e \in \overline{\alpha}$. We will prove that $[\mathcal{E}]_{\mathcal{R}}, \Delta \vdash e$, which implies the thesis. We proceed by induction on the depth of the derivation of $e \in \hat{\mathcal{R}}(\overline{\Delta})$. According to the last rule used, we have the following cases:

- case $(\in_{\hat{\mathcal{R}}})$. The premise of rule $(\in_{\hat{\mathcal{R}}})$ prescribes that $e \in \overline{\Delta}$. By suitable application of rules (ID), ($\wedge$L1) and ($\wedge$L2) we obtain the thesis $[\mathcal{E}]_{\mathcal{R}}, \Delta \vdash e$.

- case $(\vdash_{\hat{\mathcal{R}}})$. We have:

$$\frac{\hat{\mathcal{R}}(\overline{\Delta}) \vdash e}{e \in \mathcal{R}(\overline{\Delta})} \ (\vdash_{\hat{\mathcal{R}}})$$

Since $\hat{\mathcal{R}}(\overline{\Delta}) \vdash e$, there must exists a minimal $D \subseteq_{fin} \hat{\mathcal{R}}(\overline{\Delta})$ such that $D \vdash e \in \mathcal{E}$. By Def. 9.1, we have that $(\bigwedge D) \to e \in [\mathcal{E}]_{\mathcal{R}}$. By the induction hypothesis we have that for all $d \in D$, $[\mathcal{E}]_{\mathcal{R}}, \Delta \vdash d$. Then:

$$\frac{\dfrac{\overset{I.H.}{\overline{[\mathcal{E}]_{\mathcal{R}}, \Delta \vdash \bigwedge D}} \qquad \overline{[\mathcal{E}]_{\mathcal{R}}, \Delta, e \vdash e}\ (\text{ID})}{[\mathcal{E}]_{\mathcal{R}}, \Delta \vdash e}\ (\to\text{L})}$$

- case $(\Vdash_{\hat{\mathcal{R}}})$. We have:

$$\frac{\hat{\mathcal{R}}(\overline{\Delta} \cup \{e\}) \Vdash e}{e \in \hat{\mathcal{R}}(\overline{\Delta})} \ (\Vdash_{\hat{\mathcal{R}}})$$

Since $\hat{\mathcal{R}}(\overline{\Delta} \cup \{e\}) \Vdash e$, there must exist a minimal $D \subseteq_{fin} \hat{\mathcal{R}}(\overline{\Delta} \cup \{e\})$ such that $D \Vdash e \in \mathcal{E}$. Then, by Def. 9.1, we have that $(\bigwedge D) \twoheadrightarrow e \in [\mathcal{E}]_{\mathcal{R}}$. By the induction hypothesis we have that, for all $d \in D$, $[\mathcal{E}]_{\mathcal{R}}, \Delta, e \vdash d$. Then:

$$\frac{\dfrac{\overset{I.H.}{\overline{[\mathcal{E}]_{\mathcal{R}}, \Delta, e \vdash \bigwedge D}} \qquad \overline{[\mathcal{E}]_{\mathcal{R}}, \Delta, e \vdash e}\ (\text{ID})}{[\mathcal{E}]_{\mathcal{R}}, \Delta \vdash e}\ (\text{FIX})}$$ □

Example 9.4. *Suppose there are three kids who want to play together. Alice has a toy airplane, Bob has a bike, and Carl has a toy car. Each of the kids is willing to share his toy, but they have different constraints: Alice will lend her airplane only* after *Bob has allowed her ride his bike; Bob will lend his bike* after *he has played with Carl's car; Carl will lend his car if the other two kids promise to* eventually *let him play with their toys.*

We formalise this scenario as follows. Let a be the event Alice lends her airplane*; b be the event* Bob lends his bike*, and c be the event* Carl lends his car*. These constraints can be modelled in a* CES $\mathcal{E}_{toys}$ *with enablings:* $b \vdash a$, $c \vdash b$, $\{a,b\} \Vdash c$.

$$
\begin{array}{rcll}
[(X_i \circ e_i)_{i\in I}]_{\mathcal{F}} & = & \{[X_i \circ e_i]_{\mathcal{F}} \mid i \in I\} & \\
{[X \circ e]_{\mathcal{F}}} & = & \big(!e \wedge X \wedge !X\big)[\circ]\ e & \textit{where } [\circ] = \begin{cases} \to & \text{if } \circ = \vdash \\ \twoheadrightarrow & \text{if } \circ = \Vdash \end{cases} \\
{[a \# b]_{\mathcal{F}}} & = & (!a \wedge !b) \to \bot &
\end{array}
$$

Figure 9.3: Encoding configurations in Horn PCL theories.

By applying the encoding to $\mathcal{E}_{toys}$, we obtain:

$$
\begin{array}{rcl}
[\mathcal{E}_{toys}]_{\mathcal{R}} & = & \{[b \vdash a]_{\mathcal{R}}, [c \vdash b]_{\mathcal{R}}, [\{a, b\} \Vdash c]_{\mathcal{R}}\} \\
& = & \{b \to a,\ \ c \to b,\ \ (a \wedge b) \twoheadrightarrow c\}
\end{array}
$$

All the events a, b, c are reachable, since $\{a, b, c\} \in \mathcal{F}_{\mathcal{E}_{toys}}$. By Theorem 9.3, the atoms a, b, c are provable in the Horn PCL *theory $[\mathcal{E}_{toys}]_{\mathcal{R}}$. For instance, in Fig. 9.4 we show a derivation of $[\mathcal{E}_{toys}]_{\mathcal{R}} \vdash a$.* □

9.2 Configurations via logic

In Def. 9.6 we show an encoding for finite CES into PCL formulae, through which we show that the problem of deciding if a set of events is a configuration can be reduced to provability in the logic.

Intuitively, we want to test if a set of events C is an X-configuration. To do that, we first tag with a ! all the events in C, and then assume $\Delta = [\mathcal{E}]_{\mathcal{F}}, !C, X$. The encoding $[\cdot]_{\mathcal{F}}$ maps a conflict $a\#b$ to a formula $(!a \wedge !b) \to \bot$. Therefore, if C is *not* conflict-free then its encoding will deduce $\bot$. Otherwise, we check if C is entailed by Δ. The encoding of $\mathcal{E}$ will be a set of clauses of the form $\big(!e \wedge X \wedge !X\big)[\circ]\ e$. The $!e$ on the left of the clause ensures that e can be proved only if it belongs to C. Similarly, the $!X$ ensures that the justifications of e belong to C as well.

Notation 9.5. *For an event structure $\mathcal{E} = \langle E, \#, \vdash, \Vdash\rangle$ and a set $X \subseteq E$, we denote with $!X$ the set $\{!e \mid e \in X\}$. We assume $!E$ disjoint from E, i.e. $!E \cap E = \emptyset$. For a set $X \subseteq E \cup !E$, we define $X^\flat = X \cap E$, and $X^! = \{e \in E \mid !e \in X\}$.*

After Notation 9.5, for all sets of atoms $X, Y \subseteq E \cup !E$, we have (i) $X = X^\flat \cup !X^!$, and (ii) if $X \subseteq Y$, then $X^\flat \subseteq Y^\flat$ and $X^! \subseteq Y^!$.

Definition 9.6 (Encoding configurations in PCL)**.** *Let $\mathcal{E} = \langle E, \#, \vdash, \Vdash\rangle$ be a finite* CES*. The mapping $[\cdot]_{\mathcal{F}}$ from $\mathcal{E}$ into sets of* PCL *formulae is defined in Fig. 9.3.*

Example 9.7. *Recall the* CES *$\mathcal{E}_{toys}$ from Example 9.4. We have:*

$$
\begin{array}{rcl}
[\mathcal{E}_{toys}]_{\mathcal{F}} & = & \{[b \vdash a]_{\mathcal{R}},\ \ [c \vdash b]_{\mathcal{R}},\ \ [\{a, b\} \Vdash c]_{\mathcal{R}}\} \\
& = & \{(!a \wedge b \wedge !b) \to a,\ \ (!b \wedge c \wedge !c) \to b,\ \ (!c \wedge a \wedge b \wedge !a \wedge !b) \twoheadrightarrow c\}
\end{array}
$$

$$\boxed{\text{A}} = \dfrac{\dfrac{\dfrac{}{\Delta, c \vdash c}(\text{ID}) \quad \dfrac{\dfrac{}{\Delta, c, b \vdash b}(\text{ID}) \quad \dfrac{}{\Delta, c, b, a \vdash a}(\text{ID})}{\Delta, c, b \vdash a}(\to\text{L})}{\Delta, c \to b, c \vdash a}(\to \text{L}) \quad \dfrac{\dfrac{}{\Delta, c \to b, c \vdash c}(\text{ID}) \quad \dfrac{}{\Delta, c \to b, c, b \vdash b}(\text{ID})}{\Delta, c \to b, c \vdash b}(\to\text{L})}{\Delta, (a \wedge b) \twoheadrightarrow c, c \vdash (a \wedge b)}(\wedge \text{R})$$

$$\dfrac{\dfrac{\dfrac{\boxed{\text{A}} \quad \dfrac{}{\Delta, c \vdash c}(\text{Id})}{\Delta, (a \wedge b) \twoheadrightarrow c \vdash c}(\text{Fix}) \quad \dfrac{}{\Delta, b \vdash b}(\text{ID})}{\Delta, b \to a, c \to b \vdash b}(\to\text{L}) \quad \dfrac{}{\Delta, a \vdash a}(\text{ID})}{\Delta, b \to a \vdash a}(\to\text{L})$$

Figure 9.4: Atom e is reachable in $\mathcal{E}$ iff e is provable in the PCL theory $[\mathcal{E}]_{\mathcal{R}}$.

Lemma 9.8 establishes two basic properties of the encoding. Item (a) states that if a !-atom is provable, then it must already be present in Δ (i.e. it cannot be generated by the encoding). Item (b) states that, under the hypothesis $\overline{\Delta}^{\flat} \subseteq \overline{\Delta}^{!}$, if an atom e without ! is provable, then $!e$ must belong to Δ. Intuitively, this is caused by the fact that the encoding requires $!e$ on the left of all clauses which produce e.

Lemma 9.8. *For all sets of conjunctions of atoms Δ, and for all conjunctions of atoms α such that $[\mathcal{E}]_{\mathcal{F}}, \Delta \vdash \alpha$:*

(a) $\overline{\alpha}^{!} \subseteq \overline{\Delta}^{!}$

(b) $\overline{\Delta}^{\flat} \subseteq \overline{\Delta}^{!} \implies \overline{\alpha}^{\flat} \subseteq \overline{\Delta}^{!}$

Proof. Item (a) can be proved by a straightforward inductive argument on the depth of the proof of $[\mathcal{E}]_{\mathcal{F}}, \Delta \vdash \alpha$.

For item (b), by Theorem 4.9 consider a proof tree Π of $[\mathcal{E}]_{\mathcal{F}}, \Delta \vdash \alpha$ without occurrences of the (Cut) rule. The RHS of each sequent in Π is a conjunction of atoms, and so Π only contains occurrences of the rules (Id), ($\wedge$L1), ($\wedge$L2), ($\wedge$R), ($\to$L), (Fix). We proceed by induction on the depth of Π; there are the following exhaustive cases:

- (Id). The base case concerns the axiom (Id), which gives $[\mathcal{E}]_{\mathcal{F}}, \Delta \vdash \alpha$ provided that $\alpha \in \Delta$. We have that $\overline{\alpha}^{\flat} \subseteq \overline{\Delta}^{\flat} \subseteq \overline{\Delta}^{!}$.

- ($\wedge$L1), ($\wedge$L2), and ($\wedge$R). Straightforward by the induction hypothesis.

- ($\to$L). We have $p \to e \in [\mathcal{E}]_{\mathcal{F}}$ for a conjunction of atoms p and atom e, and:

$$\frac{[\mathcal{E}]_{\mathcal{F}}, \Delta, p \to e \vdash p \qquad [\mathcal{E}]_{\mathcal{F}}, \Delta, e \vdash \alpha}{[\mathcal{E}]_{\mathcal{F}}, \Delta \vdash \alpha} \ (\to\text{L})$$

  By applying item (a) on the leftmost premise of rule ($\to$L) it follows that $\overline{p}^{!} \subseteq \overline{\Delta}^{!}$. The formula $p \to e \in [\mathcal{E}]_{\mathcal{F}}$ must have been obtained as the encoding of an enabling $Z \vdash e$ in $\mathcal{E}$. Thus, by Def. 9.6 it must be the case that $p = !e \wedge Z \wedge !Z$. We then have $e \in \overline{p}^{!} \subseteq \overline{\Delta}^{!}$, and so since by hypothesis $\overline{\Delta}^{\flat} \subseteq \overline{\Delta}^{!}$:

$$\left(\overline{\Delta} \cup \{e\}\right)^{\flat} = \overline{\Delta}^{\flat} \cup \{e\} \subseteq \overline{\Delta}^{!} \cup \{e\} = \overline{\Delta}^{!}$$

  We can then apply the induction hypothesis on the rightmost premise of rule ($\to$L), and obtain the thesis $\overline{\alpha}^{\flat} \subseteq \left(\overline{\Delta} \cup \{e\}\right)^{!} = \overline{\Delta}^{!}$.

- (Fix). We have that $p \twoheadrightarrow e \in [\mathcal{E}]_{\mathcal{F}}$ for some conjunction of atoms p and atom e, and:

$$\frac{[\mathcal{E}]_{\mathcal{F}}, \Delta, p \twoheadrightarrow e, \alpha \vdash p \qquad [\mathcal{E}]_{\mathcal{F}}, \Delta, e \vdash \alpha}{[\mathcal{E}]_{\mathcal{F}}, \Delta \vdash \alpha} \ (\text{Fix})$$

  By applying item (a) to the leftmost premise of (Fix), we obtain $\overline{p}^{!} \subseteq \overline{\Delta}^{!} \cup \overline{\alpha}^{!}$. The formula $p \twoheadrightarrow e \in [\mathcal{E}]_{\mathcal{F}}$ must have been obtained as the encoding of an enabling $Z \Vdash e$

in $\mathcal{E}$. Thus, by Def. 9.6 it must be the case that $p = !e \wedge Z \wedge !Z$. We then have $e \in \overline{p}^! \subseteq \overline{\Delta}^! \cup \overline{\alpha}^!$. By applying item (a) to the rightmost premise of (Fix), we obtain $\overline{\alpha}^! \subseteq \overline{\Delta}^! \cup \{e\}^! = \overline{\Delta}^!$. Summing up,

$$(\overline{\Delta} \cup \{e\})^\flat = \overline{\Delta}^\flat \cup \{e\} \subseteq \overline{\Delta}^! \cup \{e\} \subseteq \overline{\Delta}^! \cup \overline{p}^! \subseteq \overline{\Delta}^! \cup \overline{\alpha}^! \subseteq \overline{\Delta}^!$$

We can then apply the induction hypothesis on the rightmost premise of rule (Fix), and obtain the thesis $\overline{\alpha}^\flat \subseteq (\overline{\Delta} \cup \{e\})^! = \overline{\Delta}^!$. □

The main result of this section is Theorem 9.9.

Theorem 9.9. *Let $\mathcal{E}$ be a finite* CES*. Then, for all $C \subseteq E$:*

$$C \in \mathcal{F}_\mathcal{E} \iff [\mathcal{E}]_\mathcal{F}, !C \vdash C \text{ and } [\mathcal{E}]_\mathcal{F}, !C \nvdash \bot$$

Proof. We first prove that, for all CES $\mathcal{E}$, for all $C \subseteq E$ and for all $X \subseteq E$:

$$C \in \mathcal{F}_\mathcal{E}(X) \iff [\mathcal{E}]_\mathcal{F}, !C, X \vdash C \text{ and } [\mathcal{E}]_\mathcal{F}, !C, X \nvdash \bot \tag{9.2}$$

The proof of the main statement follows directly from (9.2), with $X = \emptyset$.

For the ($\Leftarrow$) direction of (9.2), we shall first prove the following statement. For all sets of conjunctions of atoms Δ, and for all conjunctions of atoms α:

$$[\mathcal{E}]_\mathcal{F}, \Delta \vdash \alpha \ \wedge \ \overline{\Delta}^\flat \subseteq \overline{\Delta}^! \ \wedge \ CF(\overline{\Delta}^!) \implies \exists C' \in \mathcal{F}(\overline{\Delta}^\flat).\ \overline{\alpha}^\flat \subseteq C' \subseteq \overline{\Delta}^! \tag{9.3}$$

By Theorem 4.9, consider a proof tree Π of $[\mathcal{E}]_\mathcal{F}, \Delta \vdash \alpha$ without occurrences of the (Cut) rule. The RHS of each sequent in Π is a conjunction of atoms, and so Π only contains occurrences of the rules (Id), ($\wedge$L1), ($\wedge$L2), ($\wedge$R), ($\rightarrow$L), (Fix). We prove (9.3) by induction on the depth of Π.

The base case concerns the axiom (Id), which gives $[\mathcal{E}]_\mathcal{F}, \Delta \vdash \alpha$ whenever $\alpha \in \Delta$. Let $C' = \overline{\alpha}^\flat$. Then we have $C' = \overline{\alpha}^\flat \subseteq \overline{\Delta}^\flat \subseteq \overline{\Delta}^!$. Since $CF(\overline{\Delta}^!)$, then $CF(C')$. By Lemma 7.24(a) we have $C' \in \mathcal{F}(\overline{\alpha}^\flat)$, and so Lemma 7.24(b) gives the thesis $C' \in \mathcal{F}(\overline{\Delta}^\flat)$.

For the inductive case, we analyse the last rule used in Π. There are the following exhaustive cases:

- ($\wedge$L1) and ($\wedge$L2). Straightforward by the induction hypothesis.
- ($\wedge$R). For some conjunctions of atoms p and q such that $\alpha = p \wedge q$:

$$\frac{[\mathcal{E}]_\mathcal{F}, \Delta \vdash p \qquad [\mathcal{E}]_\mathcal{F}, \Delta \vdash q}{[\mathcal{E}]_\mathcal{F}, \Delta \vdash p \wedge q}\ (\wedge\text{R})$$

  By applying the induction hypothesis on the two premises, we obtain:

$$\exists C_1 \in \mathcal{F}(\overline{\Delta}^\flat).\ \overline{p}^\flat \subseteq C_1 \subseteq \overline{\Delta}^! \tag{9.4}$$

$$\exists C_2 \in \mathcal{F}(\overline{\Delta}^\flat).\ \overline{q}^\flat \subseteq C_2 \subseteq \overline{\Delta}^! \tag{9.5}$$

  Let $C' = C_1 \cup C_2$. Since $C_1, C_2 \subseteq \overline{\Delta}^!$ and $CF(\overline{\Delta}^!)$, we also have $CF(C')$. Then, by Lemma 3.11, $C' \in \mathcal{F}(\overline{\Delta}^\flat)$. Furthermore, $\overline{\alpha}^\flat = \overline{p}^\flat \cup \overline{q}^\flat \subseteq C' \subseteq \overline{\Delta}^!$.

- ($\to$L). We have $p \to e \in [\mathcal{E}]_{\mathcal{F}}$ for some conjunction of atoms p and atom e, and:

$$\frac{[\mathcal{E}]_{\mathcal{F}},\, \Delta,\, p \to e \vdash p \qquad [\mathcal{E}]_{\mathcal{F}},\, \Delta,\, e \vdash \alpha}{[\mathcal{E}]_{\mathcal{F}},\, \Delta \vdash \alpha}\ (\to\text{L})$$

  The formula $p \to e \in [\mathcal{E}]_{\mathcal{F}}$ must have been obtained as the encoding of an enabling $Z \vdash e$ in $\mathcal{E}$. Thus, by Def. 9.6 it must be the case that $p = !e \wedge Z \wedge !Z$. Since $[\mathcal{E}]_{\mathcal{F}},\, \Delta,\, p \to e \vdash p$, by Lemma 9.8 it follows that $\overline{p} \subseteq \overline{\Delta}$. Since $p = !e \wedge Z \wedge !Z$, we then have $e \in \overline{p}^{!} \subseteq \overline{\Delta}^{!}$, and so since by hypothesis $\overline{\Delta}^{\flat} \subseteq \overline{\Delta}^{!}$:

$$(\overline{\Delta} \cup \{e\})^{\flat} = \overline{\Delta}^{\flat} \cup \{e\} \subseteq \overline{\Delta}^{!} \cup \{e\} = \overline{\Delta}^{!}$$

  Note also that $CF(\overline{\Delta}^{!})$ and $CF((\overline{\Delta} \cup \{e\})^{!})$. We can then apply the induction hypothesis twice on the two premises, and obtain:

$$\exists C_1 \in \mathcal{F}(\overline{\Delta}^{\flat}).\ \overline{p}^{\flat} \subseteq C_1 \subseteq \overline{\Delta}^{!} \tag{9.6}$$

$$\exists C_2 \in \mathcal{F}(\overline{\Delta}^{\flat} \cup \{e\}).\ \overline{\alpha}^{\flat} \subseteq C_2 \subseteq \overline{\Delta}^{!} \tag{9.7}$$

  Let $C' = C_1 \cup C_2$. Since $C_1, C_2 \subseteq \overline{\Delta}^{!}$ and $CF(\overline{\Delta}^{!})$, we also have $CF(C')$. Since $Z \vdash e$ and $Z = \overline{p}^{\flat} \subseteq C_1$, then by saturation we also have that $C_1 \vdash e$. Therefore, Lemma b gives $C' \in \mathcal{F}(\overline{\Delta}^{\flat})$. The thesis follows by $\overline{\alpha}^{\flat} \subseteq \overline{\alpha}^{\flat} \cup \overline{p}^{\flat} \subseteq C' \subseteq \overline{\Delta}^{!}$.

- (Fix). We have that $p \twoheadrightarrow e \in [\mathcal{E}]_{\mathcal{F}}$ for some conjunction of atoms p and atom e, and:

$$\frac{[\mathcal{E}]_{\mathcal{F}},\, \Delta,\, p \twoheadrightarrow e,\, \alpha \vdash p \qquad [\mathcal{E}]_{\mathcal{F}},\, \Delta,\, e \vdash \alpha}{[\mathcal{E}]_{\mathcal{F}},\, \Delta \vdash \alpha}\ (\text{Fix})$$

  The formula $p \twoheadrightarrow e \in [\mathcal{E}]_{\mathcal{F}}$ must have been obtained as the encoding of a circular enabling $Z \Vdash e$ in $\mathcal{E}$. Thus, by Def. 9.6 it must be the case that $p = !e \wedge Z \wedge !Z$. By applying Lemma 9.8 on the sequent $[\mathcal{E}]_{\mathcal{F}},\, \Delta \vdash \alpha$ we have $\overline{\alpha} \subseteq \overline{\Delta}^{!}$. Therefore, $(\overline{\Delta} \cup \overline{\alpha})^{\flat} = \overline{\Delta}^{\flat} \cup \overline{\alpha}^{\flat} \subseteq \overline{\Delta}^{!} \cup \overline{\alpha} \subseteq \overline{\Delta}^{!}$, By applying Lemma 9.8 on the sequent $[\mathcal{E}]_{\mathcal{F}},\, \Delta,\, \alpha \vdash p$, we have $\overline{p}^{!} \subseteq \overline{\Delta}^{!} \cup \overline{\alpha}^{!} \subseteq \overline{\Delta}^{!}$. Thus, $(\overline{\Delta} \cup \{e\})^{\flat} \subseteq \overline{\Delta}^{!} \cup \overline{p}^{!} \subseteq \overline{\Delta}^{!} \cup \overline{p} \subseteq \overline{\Delta}^{!}$.

  Note also that, since $\overline{\alpha}^{!} \subseteq \overline{\Delta}^{!}$, then $CF(\overline{\Delta}^{!} \cup \overline{\alpha}^{!})$ holds, as well as $CF(\overline{\Delta}^{!} \cup \{e\}^{!})$. We can then apply the induction hypothesis twice on the two premises of rule (Fix):

$$\exists C_1 \in \mathcal{F}(\overline{\Delta}^{\flat} \cup \overline{\alpha}^{\flat}).\ \overline{p}^{\flat} \subseteq C_1 \subseteq \overline{\Delta}^{!} \tag{9.8}$$

$$\exists C_2 \in \mathcal{F}(\overline{\Delta}^{\flat} \cup \{e\}).\ \overline{\alpha}^{\flat} \subseteq C_2 \subseteq \overline{\Delta}^{!} \tag{9.9}$$

  Let $C' = C_1 \cup C_2$. Since $C_1, C_2 \subseteq C$ and $CF(C)$, we also have $CF(C')$. Since $Z \Vdash e$ and $Z \subseteq \overline{p}^{\flat} \subseteq C_1$, by saturation we have $C_1 \Vdash e$. Since $\overline{\alpha}^{\flat} \subseteq C_2$, by (9.8) and by Lemma 7.24(b) we have $C_1 \in \mathcal{F}(\overline{\Delta}^{\flat} \cup C_2)$. Therefore, by Lemma c, we obtain $\overline{\alpha}^{\flat} \subseteq \overline{p}^{\flat} \cup \overline{\alpha}^{\flat} \subseteq C_1 \cup C_2 = C' \in \mathcal{F}(\overline{\Delta}^{\flat})$.

We now prove that (9.3) implies the thesis. Assume that $[\mathcal{E}]_{\mathcal{F}}, !C, X \vdash C$, and $[\mathcal{E}]_{\mathcal{F}}, !C, X \nvdash \bot$. Let $Y = X \cap C$, let $\Delta = !C \cup \bigwedge Y$, and let $\alpha = \bigwedge C$. Then, $\overline{\Delta}^{!} = \overline{\alpha}^{\flat} = C$, and $\overline{\Delta}^{\flat} = Y$. By contradiction, assume that $\overline{\Delta}^{!}$ is not conflict-free. Since $\overline{\Delta}^{!} = C$, there must exist $a, b \in C$ such that $a \# b$, and so by Def. 9.6 we would have that $[\mathcal{E}]_{\mathcal{F}}, !C \vdash (!a \wedge !b) \to \bot$. This would imply that $[\mathcal{E}]_{\mathcal{F}}, !C \vdash \bot$, which would contradict the hypothesis of the lemma. Therefore, $CF(\overline{\Delta}^{!})$. Also, $\overline{\Delta}^{\flat} = Y \subseteq C = \overline{\Delta}^{\flat}$. We can then apply (9.3), from which we find some $C' \in \mathcal{F}(X)$ such that $\overline{\alpha}^{\flat} \subseteq C' \subseteq \overline{\Delta}^{!}$. Since $C = \overline{\alpha}^{\flat} \subseteq C' \subseteq \overline{\Delta}^{!} = C$, we have $C \in \mathcal{F}(Y)$. Since $Y \subseteq X$, by Lemma 7.24(b) we obtain the thesis $C \in \mathcal{F}(X)$.

For the ($\Rightarrow$) direction of (9.2), assume that $C \in \mathcal{F}_{\mathcal{E}}(X)$. Observe first that, since $CF(C)$, then for all choices of $X \subseteq E$ it cannot be the case that $[\mathcal{E}]_{\mathcal{F}}, !C, X \vdash \bot$. By Lemma 7.15, there exists $\sigma = \langle e_1 \dots e_n \rangle \in \mathcal{T}(X)$ such that $\overline{\sigma} = C$. We proceed by induction on $|C \setminus X|$. For the base case $|C \setminus X| = 0$, since $C \subseteq X$, by rule (Id) we have that $[\mathcal{E}]_{\mathcal{F}}, !C, X \vdash C$. For the inductive case, we assume that

$$C \in \mathcal{F}(Y) \implies [\mathcal{E}]_{\mathcal{F}}, !C, Y \vdash C \tag{9.10}$$

holds for all Y such that $|C \setminus Y| < |C \setminus X|$. To do that, we prove the following statement, which implies (9.10). Let $\Gamma = [\mathcal{E}]_{\mathcal{F}}, !C, Y \vdash C$. Then:

$$\forall i \leq n.\ \Gamma \vdash e_i \tag{9.11}$$

To prove (9.11) we proceed by (strong) induction on i: assuming that (9.11) holds for $1..i-1$, we prove that it holds for i. We have the following three subcases, according to the way e_i has been justified in $\sigma \in \mathcal{T}(X)$.

- $e_i \in X$. The thesis follows trivially by (Id).
- $\overline{\sigma_i} \vdash e_i$ and $e_i \notin X$. By the induction hypothesis of (9.11), $\Gamma \vdash \overline{\sigma_i}$. By Def. 7.1, there must exist a minimal $Z \subseteq \overline{\sigma_i}$ such that $Z \vdash e_i$ is an enabling in $\mathcal{E}$. By Def. 9.6, $[\mathcal{E}]_{\mathcal{F}}$ contains the formula $p \to e_i$, with $p = !e \wedge Z \wedge !Z$. The sequent $\Gamma \vdash \overline{\sigma_i}$ can be weakened as $\Gamma \vdash Z$, because $Z \subseteq \overline{\sigma_i}$. Since $e \in C$, $Z \subseteq C$, and Γ contains $!C$, we also have that $\Gamma \vdash p$. Therefore, by rule ($\to$L) we obtain the thesis:

$$\frac{\Gamma, p \to e_i \vdash p \qquad \Gamma, p \to e_i, e_i \vdash e_i}{\Gamma, p \to e_i \vdash e_i}$$

- $C \Vdash e_i$ and $e_i \notin X$. By Def. 7.1, there must exist a minimal $Z \subseteq C$ such that $Z \Vdash e_i$ is a circular enabling in $\mathcal{E}$. By Def. 9.6, $[\mathcal{E}]_{\mathcal{F}}$ contains the formula $p \twoheadrightarrow e_i$, with $p = !e_i \wedge Z \wedge !Z$. Since $e_i \in C \setminus X$, we have that $|C \setminus X| > |C \setminus (X \cup \{e_i\})|$. Since $C \in \mathcal{F}(X)$, by Lemma 7.24(b) we also have that $C \in \mathcal{F}(X \cup \{e\})$. Thus, the induction hypothesis on the statement (9.10) gives $[\mathcal{E}]_{\mathcal{F}}, !C, X, e_i \vdash C$. This sequent can be weakened as $\Gamma, e_i \vdash Z$, because $Z \subseteq C$. Since $e \in C$, $Z \subseteq C$, and Γ contains $!C$, we also deduce that $\Gamma, e_i \vdash p$. Therefore, by rule (Fix) we obtain the thesis:

$$\frac{\Gamma, p \twoheadrightarrow e_i, e_i \vdash p \qquad \Gamma, p \twoheadrightarrow e_i, e_i \vdash e_i}{\Gamma, p \twoheadrightarrow e_i \vdash e_i}$$

□

$$
\begin{aligned}
[(X_i \circ e_i)_{i\in I}]_{\mathcal{U}} &= \{[X_i \circ e_i]_{\mathcal{U}} \mid i \in I\} \cup \{!e \to Ue \mid e \in E\} \cup \{Ue \to Re \mid e \in E\} \\
[X \vdash e]_{\mathcal{U}} &= \{RX \to Re,\ \ !X \to Ue\} \\
[X \Vdash e]_{\mathcal{U}} &= \{RX \twoheadrightarrow Ue\}
\end{aligned}
$$

Figure 9.5: Encoding urgent events in Horn PCL theories.

Example 9.10. *Recall the* CES $\mathcal{E}_7$ *from Fig. 7.1. We have that:*

$$[\mathcal{E}_7] \;=\; \{(!c \wedge !a \wedge !b \wedge a \wedge b) \to c,\ \ (!a \wedge !c \wedge c) \twoheadrightarrow a,\ \ (!b \wedge !c \wedge c) \twoheadrightarrow b\}$$

Let $C = \{a, b, c\}$*. We have that* $C \in \mathcal{F}_{\mathcal{E}_7}$*, and* $[\mathcal{E}_7], !C \vdash C$*. Note that, were the* $!$*-ed atoms omitted in the premises of* $\to$ */* $\twoheadrightarrow$*, then we would have,* e.g.*,* $[\mathcal{E}_7], !a, !c \vdash a \wedge c$*, from which Theorem 9.9 would have incorrectly given* $\{a, c\} \in \mathcal{F}_{\mathcal{E}_7}$*.* □

9.3 Urgency via logic

In Def. 9.12 we show an encoding for finite, conflict-free CES into PCL formulae, through which we show that the problem of deciding if a set of events is a configuration can be reduced to provability in the logic.

Unless stated otherwise, in the rest of this section we assume a finite, conflict-free CES $\mathcal{E} = (E, \#, \vdash, \Vdash)$. For $X \vdash e$ we shall mean that X is the minimal set of events such that $(X, e) \in \vdash$ (similarly for $\Vdash$).

The main result of this section is Theorem 9.17.

Notation 9.11. *Let* $\star \in \{!, R, U\}$*. We assume three injections* $\star : E \to E$*, such that* $!E$*,* RE *and* UE *are pairwise disjoint. For a set of events* $X \subseteq E$*, we denote with* $\star X$ *the theory* $\{\star e \mid e \in X\}$*. We assume that the events occurring in* $\mathcal{E}$ *are disjoint from* $\star E$*. For a set* $X \subseteq !E \cup RE \cup UE$*, we define the projection* $X^\star = \{e \in E \mid \star e \in X\}$*. When* $\alpha = a_1 \wedge \cdots \wedge a_n$*, we write* $\star\alpha = \star a_1 \wedge \cdots \wedge \star a_n$*. When* $n = 0$*,* $\star\alpha = \top$*.*

Definition 9.12 (Encoding urgent events in PCL)**.** *Let* $\mathcal{E} = \langle E, \#, \vdash, \Vdash\rangle$ *be a finite* CES*. The mapping* $[\cdot]_{\mathcal{U}}$ *from* $\mathcal{E}$ *into* PCL *formulae is defined in Fig. 9.5.*

Intuitively, the atoms of the form $!e$ correspond to events already happened in the past, the atoms Ue correspond to the urgent events (also including the past ones), while the atoms Re are those actions which can be eventually reached by performing the urgent events. The encoding of an enabling $X \vdash e$ contains $!X \to Ue$, meaning that e becomes urgent when its preconditions X have been done, and $RX \to Re$, meaning that e is reachable whenever its preconditions are such. The encoding of a circular enabling $X \Vdash e$ contains $RX \twoheadrightarrow Ue$, meaning that e is urgent when its preconditions are guaranteed to be reachable.

Example 9.13. *Recall the* CES $\mathcal{E}_{toys}$ *from Example 9.4. We have:*

$$\begin{aligned}[\mathcal{E}_{toys}]_{\mathcal{U}} &= \{[b \vdash a]_{\mathcal{R}}, \quad [c \vdash b]_{\mathcal{R}}, \quad [\{a, b\} \Vdash c]_{\mathcal{R}}\} \\ &= \{Rb \to Ra \wedge !b \to Ua, \;\; Rc \to Rb \wedge !c \to Ub, \;\; \{Ra \wedge Rb\} \to Uc, \\ &\quad\; !a \to Ua, !b \to Ub, !c \to Uc, Ua \to Ra, Ub \to Rb, Uc \to Rc\}\end{aligned}$$

The following lemma states that the only way to derive $!e$ having the mapping $[\mathcal{E}]_{\mathcal{U}}$ and a set of hypothesis Δ is to have it in the set of hypothesis Δ, since the mapping itself does not provide any way to derive a !-event.

Lemma 9.14. *For all sets of atoms* Δ*, and for all conjunctions of atoms* α*:*

$$[\mathcal{E}]_{\mathcal{U}}, \Delta \vdash \alpha \implies \overline{\alpha}^{!} \subseteq \overline{\Delta}^{!}$$

Proof. Straightforward induction on the depth of the proof of $[\mathcal{E}]_{\mathcal{U}}, \Delta \vdash \alpha$. □

The following lemma confirms our intuition about the encoding $[\,]_{\mathcal{U}}$. In conflict-free CES, checking provability of Re suffices for determining that e is reachable (Lemma 9.15(a)). Also, checking Ue under the additional hypothesis $!C$ suffices for determining that e is urgent in C (Lemma 9.15(b)).

Lemma 9.15. *For all* $C \subseteq E$*, and for all* $e \in E$*:*

(a) $e \in \mathcal{R}_{\mathcal{E}} \iff [\mathcal{E}]_{\mathcal{U}} \vdash Re$

(b) $e \in \hat{\mathcal{U}}^{C}_{\mathcal{E}} \iff [\mathcal{E}]_{\mathcal{U}}, !C \vdash Ue$

Proof. ($\Leftarrow$), we shall first prove the following statement. For all sets of conjunctions of atoms Δ, and for all conjunctions of atoms α:

$$[\mathcal{E}]_{\mathcal{U}}, \Delta \vdash \alpha \implies \begin{cases} \overline{\alpha}^{R} \subseteq \mathcal{R}(\overline{\Delta}^{!UR}) & \text{(9.12a)} \\ \overline{\alpha}^{U} \subseteq \hat{\mathcal{U}}^{\overline{\Delta}^{!}}(\overline{\Delta}^{UR}) \cup \overline{\Delta}^{U} & \text{(9.12b)} \end{cases}$$

By Theorem 4.9, consider a proof tree Π of $[\mathcal{E}]_{\mathcal{U}}, \Delta \vdash \alpha$ without occurrences of the (Cut) rule. The RHS of each sequent in Π is a conjunction of atoms, and so Π only contains occurrences of the rules (Id), ($\wedge$L1), ($\wedge$L2), ($\wedge$R), ($\to$L), (Fix). We prove (9.12a) and (9.12b) by induction on the depth of Π.

The base case concerns the axiom (Id), which gives $[\mathcal{E}]_{\mathcal{U}}, \Delta \vdash \alpha$ whenever $\alpha \in \Delta$. For (9.12a), we have $\overline{\alpha}^{R} \subseteq \overline{\Delta}^{R} \subseteq \mathcal{R}(\overline{\Delta}^{R}) \subseteq \mathcal{R}(\overline{\Delta}^{!UR})$. For (9.12b), we have $\overline{\alpha}^{U} \subseteq \overline{\Delta}^{U}$.

For the inductive case, we analyse the last rule used in Π. There are the following exhaustive cases:

- ($\wedge$L1) and ($\wedge$L2). Straightforward by the induction hypothesis.

- ($\wedge$R). For some conjunctions of atoms p and q such that $\alpha = p \wedge q$:

$$\frac{[\mathcal{E}]_{\mathcal{U}},\, \Delta \vdash p \qquad [\mathcal{E}]_{\mathcal{U}},\, \Delta \vdash q}{[\mathcal{E}]_{\mathcal{U}},\, \Delta \vdash p \wedge q}\ (\wedge\mathrm{R})$$

  By applying the induction hypotheses of (9.12a) and (9.12b) on the two premises:

$$\begin{aligned}
\overline{\alpha}^R &= \overline{(p \wedge q)}^R = \overline{p}^R \cup \overline{q}^R \subseteq \mathcal{R}(\overline{\Delta}^{!UR}) \\
\overline{\alpha}^U &= \overline{(p \wedge q)}^U = \overline{p}^U \cup \overline{q}^U \subseteq \hat{\mathcal{U}}^{\overline{\Delta}^!}(\overline{\Delta}^{UR}) \cup \overline{\Delta}^U
\end{aligned}$$

- ($\rightarrow$L). We have $p \rightarrow q \in [\mathcal{E}]_{\mathcal{U}}$ for some conjunctions of atoms p and q, and:

$$\frac{[\mathcal{E}]_{\mathcal{U}},\, \Delta,\, p \rightarrow q \vdash p \qquad [\mathcal{E}]_{\mathcal{U}},\, \Delta,\, q \vdash \alpha}{[\mathcal{E}]_{\mathcal{U}},\, \Delta \vdash \alpha}\ (\rightarrow\mathrm{L})$$

  According to Def. 9.12, the formula $p \rightarrow q \in [\mathcal{E}]_{\mathcal{U}}$ must have one of following forms:

  - $!e \rightarrow Ue$. We have that $\overline{q}^U = \{e\}$, while $\overline{q}^{!R} = \emptyset$. By applying the induction hypothesis to the rightmost premise of the rule, we obtain:

$$\begin{aligned}
\overline{\alpha}^R &\subseteq \mathcal{R}(\overline{\Delta, q}^{!UR}) = \mathcal{R}(\overline{\Delta}^{!UR} \cup \{e\}) \\
\overline{\alpha}^U &\subseteq \hat{\mathcal{U}}^{\overline{\Delta,q}^!}(\overline{\Delta, q}^{UR}) \cup \overline{\Delta, q}^U = \hat{\mathcal{U}}^{\overline{\Delta}^!}(\overline{\Delta}^{UR} \cup \{e\}) \cup \overline{\Delta}^U \cup \{e\}
\end{aligned}$$

    By the leftmost premise of the rule we have that $[\mathcal{E}]_{\mathcal{U}},\, \Delta \vdash !e$. Then, by Lemma 9.14 it must be $e \in \overline{\Delta}^!$.
    For (9.12a), we have $\overline{\Delta}^{!UR} \cup \{e\} = \overline{\Delta}^{!UR}$, from which the thesis follows.
    For (9.12b), by Def. 7.71 we have $\hat{\mathcal{U}}^{\overline{\Delta}^!}(\overline{\Delta}^{UR} \cup \{e\}) = \hat{\mathcal{U}}^{\overline{\Delta}^!}(\overline{\Delta}^{UR})$ and $e \in \hat{\mathcal{U}}^{\overline{\Delta}^!}(X)$ for all X, from which the thesis follows.
  - $Ue \rightarrow Re$. We have that $\overline{q}^R = \{e\}$, while $\overline{q}^{!U} = \emptyset$. By applying the induction hypothesis to both premises of the rule, we obtain:

$$e \in \hat{\mathcal{U}}^{\overline{\Delta}^!}(\overline{\Delta}^{UR}) \cup \overline{\Delta}^U \tag{9.13}$$

$$\overline{\alpha}^R \subseteq \mathcal{R}(\overline{\Delta, q}^{!UR}) = \mathcal{R}(\overline{\Delta}^{!UR} \cup \{e\}) \tag{9.14}$$

$$\overline{\alpha}^U \subseteq \hat{\mathcal{U}}^{\overline{\Delta,q}^!}(\overline{\Delta, q}^{UR}) \cup \overline{\Delta, q}^U = \hat{\mathcal{U}}^{\overline{\Delta}^!}(\overline{\Delta}^{UR} \cup \{e\}) \cup \overline{\Delta}^U \tag{9.15}$$

    From (9.13), Lemma 7.36 gives that $e \in \mathcal{R}(\overline{\Delta}^{!UR}) \cup \overline{\Delta}^U = \mathcal{R}(\overline{\Delta}^{!UR})$. By applying Lemma 7.36(b) to (9.14), we have $\overline{\alpha}^R \subseteq \mathcal{R}(\overline{\Delta}^{!UR} \cup \mathcal{R}(\overline{\Delta}^{!UR}))$. Lemma 7.36(c) allows then to obtain the thesis of (9.12a), i.e. $\overline{\alpha}^R \subseteq \mathcal{R}(\overline{\Delta}^{!UR})$.
    For (9.12b), we have that:

$$\begin{aligned}
\overline{\alpha}^U &\subseteq \hat{\mathcal{U}}^{\overline{\Delta}^!}(\overline{\Delta}^{UR} \cup \{e\}) \cup \overline{\Delta}^U && \text{by (9.15)} \\
&= \hat{\mathcal{U}}^{\overline{\Delta}^!}(\overline{\Delta}^{!UR} \cup \{e\}) \cup \overline{\Delta}^U && \text{by Def. 7.71} \\
&\subseteq \hat{\mathcal{U}}^{\overline{\Delta}^!}(\overline{\Delta}^{!UR}) \cup \overline{\Delta}^U && \text{by Lemma 7.73(b)} \\
&= \hat{\mathcal{U}}^{\overline{\Delta}^!}(\overline{\Delta}^{UR}) \cup \overline{\Delta}^U && \text{by Def. 7.71}
\end{aligned}$$

– $RX \to Re$. This has been generated because the enabling $X \vdash e$ is in $\mathcal{E}$. By applying the induction hypothesis to both premises of the rule:

$$X \subseteq \mathcal{R}(\overline{\Delta}^{!UR}) \tag{9.16}$$
$$\overline{\alpha}^R \subseteq \mathcal{R}(\overline{\Delta}^{!UR} \cup \{e\}) \tag{9.17}$$
$$\overline{\alpha}^U \subseteq \hat{\mathcal{U}}^{\overline{\Delta}^!}(\overline{\Delta}^{UR} \cup \{e\}) \cup \overline{\Delta}^U \tag{9.18}$$

For (9.12a), $X \vdash e$ and (9.16) imply that $e \in \mathcal{R}(\overline{\Delta}^{!UR})$. Then, we can apply Lemma 7.36(c) to (9.17) and obtain the thesis $\overline{\alpha}^R \subseteq \mathcal{R}(\overline{\Delta}^{!UR})$.
For (9.12b), we have that:

$$\begin{aligned}
\overline{\alpha}^U &\subseteq \hat{\mathcal{U}}^{\overline{\Delta}^!}(\overline{\Delta}^{UR} \cup \{e\}) \cup \overline{\Delta}^U && \text{by (9.18)} \\
&= \hat{\mathcal{U}}^{\overline{\Delta}^!}(\overline{\Delta}^{!UR} \cup \{e\}) \cup \overline{\Delta}^U && \text{by Def. 7.71} \\
&\subseteq \hat{\mathcal{U}}^{\overline{\Delta}^!}(\overline{\Delta}^{!UR}) \cup \overline{\Delta}^U && \text{by Lemma 7.73(b)} \\
&= \hat{\mathcal{U}}^{\overline{\Delta}^!}(\overline{\Delta}^{UR}) \cup \overline{\Delta}^U && \text{by Def. 7.71}
\end{aligned}$$

– $!X \to Ue$. This has been generated because of an enabling $X \vdash e$ in $\mathcal{E}$. By applying the induction hypothesis to the rightmost premise of the rule:

$$\begin{aligned}
\overline{\alpha}^R &\subseteq \mathcal{R}(\overline{\Delta}^{!UR} \cup \{e\}) \\
\overline{\alpha}^U &\subseteq \hat{\mathcal{U}}^{\overline{\Delta}^!}(\overline{\Delta}^{UR} \cup \{e\}) \cup \overline{\Delta}^U \cup \{e\}
\end{aligned}$$

By the leftmost premise of the rule we have that $[\mathcal{E}]_{\mathcal{U}}, \Delta \vdash !X$, and so by Lemma 9.14 it must be $X \subseteq \overline{\Delta}^!$. Therefore, $\overline{\Delta}^! \vdash e$, from which we conclude that $e \in \mathcal{R}(\overline{\Delta}^!) \subseteq \mathcal{R}(\overline{\Delta}^{!UR})$. Lemma 7.36(c) gives the thesis for (9.12a).
For (9.12b), from $\overline{\Delta}^! \vdash e$ it follows that $e \in \hat{\mathcal{U}}^{\overline{\Delta}^!}(X)$, for all X. The thesis follows from Lemmata 7.73(c) and 7.73(b).

- (Fix). We have that $p \twoheadrightarrow q \in [\mathcal{E}]_{\mathcal{U}}$ for some conjunctions of atoms p and q, and:

$$\frac{[\mathcal{E}]_{\mathcal{U}}, \Delta, p \twoheadrightarrow q, \alpha \vdash p \qquad [\mathcal{E}]_{\mathcal{U}}, \Delta, q \vdash \alpha}{[\mathcal{E}]_{\mathcal{U}}, \Delta \vdash \alpha}\ \text{(Fix)}$$

By Def. 9.12, the formula $p \twoheadrightarrow q \in [\mathcal{E}]_{\mathcal{U}}$ must have been obtained as the encoding of a circular enabling $X \Vdash e$ in $\mathcal{E}$, which gives $p = RX$ and $q = Ue$.
By applying the induction hypothesis to both premises of rule (Fix):

$$X \subseteq \mathcal{R}(\overline{\Delta, \alpha}^{!UR}) = \mathcal{R}(\overline{\Delta}^{!UR} \cup \overline{\alpha}^{!UR}) \tag{9.19}$$
$$\overline{\alpha}^R \subseteq \mathcal{R}(\overline{\Delta, q}^{!UR}) \subseteq \mathcal{R}(\overline{\Delta}^{!UR} \cup \{e\}) \tag{9.20}$$
$$\overline{\alpha}^U \subseteq \hat{\mathcal{U}}^{\overline{\Delta}^!}(\overline{\Delta}^{UR} \cup \{e\}) \cup \overline{\Delta}^U \cup \{e\} \tag{9.21}$$

We have that:

$$\begin{aligned}
X &\subseteq \mathcal{R}(\overline{\Delta}^{!UR} \cup \overline{\alpha}^{!UR}) && \text{by (9.19)}\\
&\subseteq \mathcal{R}(\overline{\Delta}^{!UR} \cup \overline{\alpha}^{UR}) && \text{by Lemma 9.14}\\
&\subseteq \mathcal{R}(\overline{\Delta}^{!UR} \cup \overline{\alpha}^{U} \cup \mathcal{R}(\overline{\Delta}^{!UR} \cup \{e\})) && \text{by (9.20)}\\
&\subseteq \mathcal{R}(\overline{\Delta}^{!UR} \cup \{e\} \cup \overline{\alpha}^{U} \cup \mathcal{R}(\overline{\Delta}^{!UR} \cup \{e\})) && \text{by Lemma 7.36(b)}\\
&\subseteq \mathcal{R}(\overline{\Delta}^{!UR} \cup \overline{\alpha}^{U} \cup \{e\}) && \text{by Lemma 7.36(c)}\\
&\subseteq \mathcal{R}(\overline{\Delta}^{!UR} \cup \hat{\mathcal{U}}^{\overline{\Delta}^{!}}(\overline{\Delta}^{UR} \cup \{e\}) \cup \overline{\Delta}^{U} \cup \{e\}) && \text{by (9.21)}\\
&= \mathcal{R}(\overline{\Delta}^{!UR} \cup \hat{\mathcal{U}}^{\overline{\Delta}^{!}}(\overline{\Delta}^{UR} \cup \{e\})) && \text{by Corollary 7.43}\\
&\subseteq \mathcal{R}(\overline{\Delta}^{!UR} \cup \mathcal{R}(\overline{\Delta}^{!UR} \cup \{e\})) && \text{by Lemma 7.73(c)}\\
&= \mathcal{R}(\overline{\Delta}^{!UR} \cup \mathcal{R}(\overline{\Delta}^{!UR})) && \text{by Corollary 7.43}\\
&= \mathcal{R}(\overline{\Delta}^{!UR}) && \text{by Lemma 7.36(c)}
\end{aligned}$$

For (9.12a), since $X \subseteq \mathcal{R}(\overline{\Delta}^{!UR} \cup \{e\}) \Vdash e$, by Corollary 7.43 we obtain the thesis:

$$\overline{\alpha}^{R} \;\subseteq\; \mathcal{R}(\overline{\Delta}^{!UR} \cup \{e\}) \;=\; \mathcal{R}(\overline{\Delta}^{!UR})$$

For (9.12b), since $X \subseteq \mathcal{R}(\overline{\Delta}^{!UR}) \Vdash e$, then $e \in \mathcal{R}(\overline{\Delta}^{!UR}) = \mathcal{R}(\overline{\Delta}^{!} \cup \overline{\Delta}^{UR})$. By Def. 7.71 it follows that $e \in \hat{\mathcal{U}}^{\overline{\Delta}^{!}}(\overline{\Delta}^{UR})$. Thus:

$$\begin{aligned}
\overline{\alpha}^{U} &\subseteq \hat{\mathcal{U}}^{\overline{\Delta}^{!}}(\overline{\Delta}^{UR} \cup \{e\}) \cup \overline{\Delta}^{U} \cup \{e\} && \text{by (9.21)}\\
&= \hat{\mathcal{U}}^{\overline{\Delta}^{!}}(\overline{\Delta}^{!UR} \cup \{e\}) \cup \overline{\Delta}^{U} \cup \{e\} && \text{by Def. 7.71}\\
&\subseteq \hat{\mathcal{U}}^{\overline{\Delta}^{!}}(\overline{\Delta}^{!UR}) \cup \overline{\Delta}^{U} \cup \{e\} && \text{by Lemma 7.73(b)}\\
&= \hat{\mathcal{U}}^{\overline{\Delta}^{!}}(\overline{\Delta}^{UR}) \cup \overline{\Delta}^{U} \cup \{e\} && \text{by Def. 7.71}\\
&= \hat{\mathcal{U}}^{\overline{\Delta}^{!}}(\overline{\Delta}^{UR}) \cup \overline{\Delta}^{U} && \text{since } e \in \hat{\mathcal{U}}^{\overline{\Delta}^{!}}(\overline{\Delta}^{UR}).
\end{aligned}$$

We now prove that (9.12a) and (9.12b) imply 9.15(a) and 9.15(b), respectively.
For 9.15(a), assume that $[\mathcal{E}]_{\mathcal{U}} \vdash Re$. Let $\Delta = \emptyset$ and $\alpha = Re$. By (9.12a) we obtain:

$$\{e\} \;=\; \overline{\alpha}^{R} \;\subseteq\; \mathcal{R}(\overline{\Delta}^{!UR}) \;=\; \mathcal{R}(\emptyset) \;=\; \mathcal{R}_{\mathcal{E}}$$

For 9.15(b), assume that $[\mathcal{E}]_{\mathcal{U}}$, $!C \vdash Ue$. Let $\Delta = !C$, and let $\alpha = Ue$. Then, $\overline{\Delta}^{!} = C$, $\overline{\Delta}^{UR} = \emptyset$, and $\overline{\alpha}^{U} = \{e\}$. By (9.12b) we obtain:

$$\{e\} \;=\; \overline{\alpha}^{U} \;\subseteq\; \hat{\mathcal{U}}^{\overline{\Delta}^{!}}(\overline{\Delta}^{UR}) \cup \overline{\Delta}^{U} \;=\; \hat{\mathcal{U}}^{C}(\emptyset) \;=\; \hat{\mathcal{U}}^{C}$$

For the ($\Rightarrow$) direction of item (a), we prove the following stronger statement. For all X, if $e \in \mathcal{R}(X)$ then $[\mathcal{E}]_{\mathcal{U}}, RX \vdash Re$. Assume that $e \in \mathcal{R}(X)$. By Theorem 7.45, $e \in \hat{\mathcal{R}}(X)$. We proceed by induction on the derivation of $e \in \hat{\mathcal{R}}(X)$. According to the last rule used in the derivation, there are the following cases:

- ($\in_{\hat{\mathcal{R}}}$). We have $e \in X$, from which the thesis follows trivially.

- ($\vdash_{\hat{\mathcal{R}}}$). We have $\hat{\mathcal{R}}(X) \vdash e$. Let $C_0 \subseteq \hat{\mathcal{R}}(X)$ be a minimal set such that $C_0 \vdash e$. By the induction hypothesis, $[\mathcal{E}]_{\mathcal{U}}, RX \vdash RC_0$. Also, by Def. 9.12, $RC_0 \to Re \in [\mathcal{E}]_{\mathcal{U}}$. Therefore, by rule ($\to$L):

$$\frac{[\mathcal{E}]_{\mathcal{U}},\, RX,\, RC_0 \to Re \vdash RC_0 \qquad [\mathcal{E}]_{\mathcal{U}},\, RX,\, Re \vdash Re}{[\mathcal{E}]_{\mathcal{U}},\, RX \vdash Re}$$

- ($\Vdash_{\hat{\mathcal{R}}}$). We have $\hat{\mathcal{R}}(X \cup \{e\}) \Vdash e$. Let $C_0 \subseteq \hat{\mathcal{R}}(X \cup \{e\})$ be a minimal set such that $C_0 \Vdash e$. By the induction hypothesis, $[\mathcal{E}]_{\mathcal{U}}, RX, Re \vdash RC_0$. Also, by Def. 9.12, $RC_0 \twoheadrightarrow Ue \in [\mathcal{E}]_{\mathcal{U}}$. Therefore, by rule (Fix):

$$\frac{[\mathcal{E}]_{\mathcal{U}},\, RX,\, RC_0 \twoheadrightarrow Ue,\, Re \vdash RC_0 \qquad [\mathcal{E}]_{\mathcal{U}},\, RX,\, Ue \vdash Re}{[\mathcal{E}]_{\mathcal{U}},\, RX \vdash Re}$$

  where the second premise has been obtained because $Ue \to Re \in [\mathcal{E}]_{\mathcal{U}}$.

For the ($\Rightarrow$) direction of item (b), assume that $e \in \hat{\mathfrak{U}}^C$. We proceed by cases on the rule used in the derivation.

- ($\in_{\hat{\mathfrak{U}}}$). We have that $e \in C$. Therefore, $[\mathcal{E}]_{\mathcal{U}}, !C \vdash !e$, and since $!e \to Ue \in [\mathcal{E}]_{\mathcal{U}}$ we obtain the thesis.

- ($\vdash_{\hat{\mathfrak{U}}}$). By the rule premise, it must be $C \vdash e$. Let $C_0 \subseteq C$ be a minimal set such that $C_0 \vdash e$. Then, $\big((RC_0 \to Re) \wedge (!C_0 \to Ue)\big) \in [\mathcal{E}]_{\mathcal{U}}$. Since $C_0 \subseteq C$, then $!C_0 \to Ue$ implies that $!C \to Ue$. Therefore, $[\mathcal{E}]_{\mathcal{U}}, !C \vdash Ue$.

- ($\Vdash_{\hat{\mathfrak{U}}}$). By the rule premise, it must be $C \cup \mathcal{R}^C \Vdash e$. Let $C_0 \subseteq C \cup \mathcal{R}^C$ be a minimal set such that $C_0 \Vdash e$. By Def. 9.12, $RC_0 \twoheadrightarrow Ue \in [\mathcal{E}]_{\mathcal{U}}$. Since the encoding $[\mathcal{E}]_{\mathcal{U}}$ comprises $!e \to Ue$ and $Ue \to Re$ for all e, then $[\mathcal{E}]_{\mathcal{U}}, !C \vdash RC$. By Lemma a, $[\mathcal{E}]_{\mathcal{U}}, !C \vdash R(\mathcal{R}^C)$. Thus, $[\mathcal{E}]_{\mathcal{U}}, !C \vdash RC_0$. Then, we can weaken $RC_0 \twoheadrightarrow Ue$ to $RC_0 \to Ue$, and use rule ($\to$L) to deduce $[\mathcal{E}]_{\mathcal{U}}, !C \vdash Ue$. □

Since in this encoding ($[\mathcal{E}]_{\mathcal{U}}$) we can recognize which events are reachable, we prove that those same events are recognized reachable also in the first encoding $[\mathcal{E}]_{\mathcal{R}}$ of section 9.1.

Corollary 9.16. *Let $\mathcal{E}$ be a finite, conflict-free* CES*, and let $e \in E$. Then:*

$$[\mathcal{E}]_{\mathcal{R}} \vdash e \iff [\mathcal{E}]_{\mathcal{U}} \vdash Re$$

Proof. Straightforward by Lemma 9.15(a) and Theorem 9.3. □

In Theorem 9.17 we show that the LTS of urgent events in Def. 7.66 corresponds to the LTS of the events recognized urgent through the logic.

Theorem 9.17. *For a finite, conflict-free* CES $\mathcal{E}$*, we define the* LTS $\rightarrow_{[\mathcal{E}]_{\mathcal{U}}}$ *as:*

$$C \xrightarrow{e}_{[\mathcal{E}]_{\mathcal{U}}} C \cup \{e\} \qquad \textit{iff} \;\; [\mathcal{E}]_{\mathcal{U}}, !C \vdash Ue \;\; \wedge \;\; !C \nvdash\!\!\!\vdash !e$$

Then, $\rightharpoonup_{\mathcal{U}_\mathcal{E}} = \rightarrow_{[\mathcal{E}]_{\mathcal{U}}}$.

Proof. For the inclusion ($\subseteq$) we shall prove that, for all σ and X, if $(\emptyset,\emptyset) \xrightarrow{\sigma}_{\mathcal{U}_\mathcal{E}} (\overline{\sigma}, X)$, then $\emptyset \xrightarrow{\sigma}_{[\mathcal{E}]_{\mathcal{U}}} \overline{\sigma}$. We proceed by induction on the length of σ. The base case is trivial. For the inductive case, assume that $(\emptyset,\emptyset) \xrightarrow{\eta}_{\mathcal{U}_\mathcal{E}} \xrightarrow{e}_{\mathcal{U}_\mathcal{E}}$. By the induction hypothesis, $\emptyset \xrightarrow{\eta}_{[\mathcal{E}]_{\mathcal{U}}} \overline{\eta}$. Since $e \in \mathcal{U}^{\overline{\eta}}$, by Lemma 7.72 it follows that $e \in \hat{\mathcal{U}}^{\overline{\eta}}$. Therefore, by Lemma 9.15, $[\mathcal{E}]_{\mathcal{U}}, !\overline{\eta} \vdash Ue$. Since $e \notin \overline{\eta}$, then $!\overline{\eta} \nvdash\!\!\!\vdash !e$. Summing up, $\overline{\eta} \xrightarrow{e}_{[\mathcal{E}]_{\mathcal{U}}} \overline{\eta} \cup \{e\}$.

For the inclusion ($\supseteq$) we shall prove that, for all σ, if $\emptyset \xrightarrow{\sigma}_{[\mathcal{E}]_{\mathcal{U}}} \overline{\sigma}$ then $(\emptyset,\emptyset) \xrightarrow{\sigma}_{\mathcal{U}_\mathcal{E}} (\overline{\sigma}, X)$, for some X. We proceed by induction on the length of σ. The base case is trivial. For the inductive case, assume that $\emptyset \xrightarrow{\eta}_{[\mathcal{E}]_{\mathcal{U}}} \xrightarrow{e}_{[\mathcal{E}]_{\mathcal{U}}}$. By the induction hypothesis, $(\emptyset,\emptyset) \xrightarrow{\eta}_{\mathcal{U}_\mathcal{E}} (\overline{\eta}, X)$, where X is the least credit for η. Since $\overline{\eta} \xrightarrow{e}_{[\mathcal{E}]_{\mathcal{U}}} \overline{\eta} \cup \{e\}$, then by definition of $\rightarrow_{[\mathcal{E}]_{\mathcal{U}}}$ it must be $[\mathcal{E}]_{\mathcal{U}}, !\overline{\eta} \vdash Ue$ and $!\overline{\eta} \nvdash\!\!\!\vdash !e$. By the former and by Lemma 9.15, $e \in \hat{\mathcal{U}}^{\overline{\eta}}$; by the latter, it follows that $e \notin \overline{\eta}$. Therefore, by Lemma 7.72 we conclude that $e \in \mathcal{U}^{\overline{\eta}}$. □

9.4 Contract agreements via logic

In this sub-section we present our main results about the relation between contracts and PCL. Theorem 9.18 shows that, for conflict-free contracts and reachability payoffs, we can characterise agreement in terms of provability in PCL. Finally, Theorem 9.19 relates winning strategies with provable atoms in the encoding for urgent events.

Recall from Def. 6.10 that reachability payoffs neglect the order in which events are performed. In particular, a reachability payoff Φ for A is induced by two predicates $\varphi^0, \varphi^1 \subseteq \wp(E)$ such that

$$\Phi \mathsf{A} \sigma = \begin{cases} 1 & \text{if } \overline{\sigma} \in \varphi^1 \\ 0 & \text{if } \overline{\sigma} \in \varphi^0 \\ -1 & \text{otherwise} \end{cases}$$

The following theorem gives a logical characterisation of agreements. A conflict-free contract with CES $\mathcal{E}$ and with a reachability payoff induced by φ^1, φ^0 admits an agreement whenever the set provable atoms in $[\mathcal{E}]_{\mathcal{R}}$ satisfies the predicate φ^1.

Theorem 9.18. *Let* $\mathcal{E}$ *be a conflict-free* CES*, and let* Φ *be a reachability payoff defined by the predicates* φ^1, φ^0*. Then, the contract* $\mathcal{C} = \langle \mathcal{E}, \pi, \mathcal{A}, \Phi \rangle$ *admits an agreement iff* $\{e \mid [\mathcal{E}]_{\mathcal{R}} \vdash e\} \in \varphi^1$.

Proof. ($\Rightarrow$) assume $\mathcal{C}$ has an agreement. Since $\mathcal{E}$ is conflict-free, each play σ where all participants are innocent contains exactly the reachable events of $\mathcal{E}$. Since all participants are winning in σ, then $\Phi \mathsf{A} \sigma = 1$ for all $\mathsf{A} \in \mathcal{A}$. Hence, $\mathcal{R}_\mathcal{E} = \overline{\sigma} \in \varphi^1$. By Theorem 9.3, $\mathcal{R}_\mathcal{E} = \{e \mid [\mathcal{E}]_\mathcal{R} \vdash e\}$, from which we have the thesis.
($\Leftarrow$) the proof proceeds similarly to the above. □

Our last main result relates the winning strategies of a contract $\mathcal{C} = \langle \mathcal{E}, \pi, \mathcal{A}, \Phi \rangle$ with the encoding of urgent events in PCL. In particular, for all participants A we construct a strategy that, in a play σ, enables exactly the events e of A for which Ue is provable in the Horn PCL theory $[\mathcal{E}]_\mathcal{U}, !\overline{\sigma}$. This strategy is prudent for A, and leads A to a winning play whenever A agrees on $\mathcal{C}$.

Theorem 9.19. *Let $\mathcal{E}$ be a conflict-free* CES*, and let the strategy Σ_A be defined as:*

$$\Sigma_\mathsf{A}(\sigma) \;=\; \{e \in \pi^{-1}(\mathsf{A}) \mid [\mathcal{E}]_\mathcal{U}, !\overline{\sigma} \vdash Ue \;\wedge\; !\overline{\sigma} \nvdash\, !e\}$$

Then, Σ_A is a prudent strategy for A in $\mathcal{C} = \langle \mathcal{E}, \pi, \mathcal{A}, \Phi \rangle$. Moreover, if Φ is a reachability payoff and $\mathcal{C}$ admits an agreement, then Σ_A is winning for A.

Proof. By Theorem 9.17, $e \in \Sigma_\mathsf{A}(\sigma)$ iff $e \in \mathcal{U}_\mathcal{E}^{\overline{\sigma}}$. The thesis then follows from Theorem 8.13. □

Chapter 10

Discussion

Here we relate our approach to others appeared in literature, some of which have been briefly presented in Chapter 5.

10.1 Contracts

We have already noticed in Chapter 5 that the notion of agreement has already been addressed in different ways. For instance, our notion of agreement is similar to the notion of compliance in [CGP09] and conformance in [vdALM+10]. Instead to the best of our knowledge, ours is the first formalization of the notion of protection. Beside this, in our opinion, relevant difference, in the following we will pin point some of the other peculiar differences.

In our model we have completely neglected time and deadlines. Thus, we cannot obviously express contract constraints about events which need to be done in time. Due to this lack, our notion of culpability is quite different from the one considered in those models which monitor contracts to detect violations, e.g. [Hen11, HKZ12, RSE08, LPSS11]. There, a participant is culpable for a contract violation if something is not done before a deadline expires.

In [HKZ12], traces are analyzed to detect contract violations where only the first one represents a breach. In the case that fulfilling only a part of a contract is enough to fullfil the whole contract, then detecting which participant is culpable of a breach may be difficult. Consider for instance the contract specifing that either Alice has to fulfill an obligation by a time τ *or* that Bob has to fulfill another obligation by the same time. In case neither Alice nor Bob have fulfilled her/his obligation, [HKZ12] assigns blame non-deterministically to one of the involved parties. In our approach, a participant becomes culpable as soon as her action is enabled, and innocence can be regained as soon as that action is performed or conflicting with other actions. So, in the previous case, where a contract proposes choosing between Alice's and Bob's events, we would have modelled the choice with a conflict between the events.

Before any event is performed, both Alice and Bob are culpable, but as soon as one of them performs the event, they both become innocent.

Our notion of culpability may seem a bit draconian, in that a participant omitting to perform a single due event in a play is considered culpable, regardless of the fact that the other participants could equally be satisfied with that play. Establishing finer-grained notions of causality between a violation and the resulting failure, as done e.g. in [GMR10], seems a plausible extension of our work.

Similarly to [Hen11] we consider participants ad adversaries in a game, the plays of which represent sequences of events, possibly representing contract violations. In [Hen11] causality between events is inferred by analysing the timestamps associated to events. Instead, in our approach causality among events is defined through the (classical/circular) enabling relations of event structures. While in our approach contracts can be composed, in [Hen11] only a *global* contract is considered, with the assumption that this is the contract agreed upon by all participants, The global nature of contracts makes the notion of protection hardly definable. Another difference is that in [Hen11] the notion of strategy considers many contracts together, and the payoff must be always positive for all participants (although they may on purpose collect some penalties, if this is more profitable in terms of fulfilling goals and global rewarding). In our work, each strategy is related to a single contract, and the payoff may alternate between negative and positive values.

10.2 Circularity

In our work we deal with the issue of circularity of obligations. Circular reasoning often appears in the compositional modelling and verification of systems. Circularity issues have been investigated in assume-guarantee reasoning [AL93, AP93, Mai03, VV01], in models of workflow systems [HM10], in logic programming [SMBG06, SBMG07]. Circularity arises also when reasoning about contracts [BZ10a, BCZ13, BCP13]: circular dependencies arise when two or more tasks mutually rely on the guarantees provided by each other. Below we briefly discuss some of these approaches.

In [HM10] a generalization of prime event structures is proposed where a *response* relation (denoted with $\bullet\!\!\rightarrow$) is used to characterise the accepting traces as those where, for each $a \bullet\!\!\rightarrow b$, if a is present in the trace, then b eventually occurs after a. The response relation bears some resemblance with our $\Vdash$ relation, but there are some notable differences. First, having $a \Vdash b$ does not necessarily imply that a configuration containing a must also contain b (another enabling could have been used), whereas $a \bullet\!\!\rightarrow b$ stipulates that once one has a in a configuration, then also b must be present. Indeed, an enabling $a \Vdash b$ can be neglected, whereas $a \bullet\!\!\rightarrow b$ must be used. Also, augmenting the number of $\Vdash$-enablings increases the

number of configurations, while adding more response relations reduces the number of configurations of the event structure.

The motivations underlying the circular enabling of CES seem related to those introduced in [AP93] to compose assume-guarantee specifications [AL93]. There, the idea is that a system will give some guarantee M_1 about its behaviour, provided that the environment it operates within will behave according to some assumption M_2, and *vice versa*. In [AP93], this is rendered as the judgment $(M_1 \to M_2) \wedge (M_2 \to M_1) \vdash M_1 \wedge M_2$. However, since $\to$ is the usual intuitionistic implication, the validity of this judgment (not valid in IPC) is subject to a side condition on the interpretation of M_1, M_2 in the model. In our approach we obtain a similar goal through the circular enabling: the CES with enablings $m_1 \Vdash m_2$ and $m_2 \Vdash m_1$ has $\{m_1, m_2\}$ as a configuration.

Some preliminary work on relating event structures with the logic PCL has been reported in [BCPZ12b]. The model of [BCPZ12b] does not exploit game-theoretic notions: payoffs are just sets of events, and agreement is defined as the existence of a configuration in the CES which contains such set. In this simplified model, it is shown that an event is reachable in a CES whenever it is provable in the corresponding PCL theory. Hence, an agreement exists whenever all the events in the participant payoffs are provable. Theorem 9.18 extends this result to a more general (game-theoretic) notion of agreement and of payoff.

In this dissertation we have established a relation between the notions of winning strategy in contracts and that of urgent events in CES (Lemma 8.7). As a follow-up of this study, in [BCGZ13] we have related these notions to provability in PCL. A key observation has been that each proof in Horn PCL induces a set of atom orderings which are compatible with the proof. Each of these orderings is associated with a sequence of atoms, called *proof trace*.

To give some intuition, consider the elimination rule for $\to$:

$$\frac{\Delta \vdash \alpha \to a \qquad \Delta \vdash \alpha}{\Delta \vdash a} \ (\to\text{E})$$

The rule requires a proof of all the atoms in α in order to construct a proof of a. Accordingly, if σ is a proof trace of $\Delta \vdash \alpha$, then σa if a proof trace of $\Delta \vdash a$.

Consider now the elimination rule for $\twoheadrightarrow$:

$$\frac{\Delta \vdash \alpha \twoheadrightarrow a \qquad \Delta, a \vdash \alpha}{\Delta \vdash a} \ (\to\text{E})$$

Here, the intuition is that α needs not necessarily be proved before a: it suffices to prove α by taking a as hypothesis. Assuming that σ is a proof trace of $\Delta, a \vdash \alpha$, the proof traces of $\Delta \vdash a$ include all the interleavings between σ and a.

A main result in [BCGZ13] is that, for conflict-free contracts, proof traces correspond to sequences of prudent events.

In coinductive logic programming (CLP, [SBMG07]), both coinduction and induction can be used to give semantics to logic programs, i.e. to sets of Horn clauses. Intuitively, this can be related to CES in that $\vdash$ has an inductive flavour, while $\Vdash$ a coinductive one. However, two main differences exist between the two frameworks of CLP and CES. First, in CLP all the clauses for the same predicate have to share their inductive/coinductive nature. That is, there is no equivalent for $a_1 \vdash b$, $a_2 \Vdash b$ because b is used in both fashions. Second, CLP forbids circular dependencies between inductive and coinductive predicates, requiring stratification. For instance, CLP allows for expressing $a \Vdash b$, $b \Vdash a$, as well as $a \vdash b$, $b \Vdash c$, while it forbids $a \vdash b$, $b \Vdash a$ because b would be inductive while a would be coinductive. Other approaches mixing induction and coinduction (e.g. [LG09]) work under a similar stratification assumption.

We believe that by assuming stratification one can find good connections between CES and CLP. However, we think that (unconstrained) circularity is an essential feature of concurrent systems, and in particular of contracts. For instance, the system $a \vdash b$, $b \Vdash a$ is an archetypal scenario in contracting systems, where we are both expressing circularity between a and b, and a legitimate ordering between the events, i.e. a must occur before b. In CES, we can encompass both aspects: in the above example, $\{a, b\}$ is a configuration, and the LTS of urgent events describes the traces of events which respect the causal ordering imposed by $\vdash$-enablings (while $\Vdash$ does not prescribe any order). Therefore, requiring stratification in CES would seem to trivialize them. Note in passing that PCL requires no stratification, hence it can be meaningfully related to CES.

Chapter 11

Conclusions

Contract-oriented computing is a promising paradigm which has started to emerge in the last few years. Current approaches to the design and implementation of distributed systems typically include some aspects of contracts, e.g. WSDL for describing syntactic constraints on the usage of services, and choreography languages like WS-CDL [KBR+05], BPEL4Chor [DKLW07] and Scribble [HMB+11] for specifying the overall interaction protocol of a set of Web services.

However, none of the current proposals features a comprehensive framework for the design of contract-oriented systems.

To make a step towards this direction, we have introduced a theory of contracts, where the interactions (obligations) among the participants are represented using event structures. We have characterised two relevant notions in contract-oriented computing, namely agreement and protection.

Modelling obligations using *standard* event structures does not allow to guarantee agreement and protection at the same time. Intuitively, the kind of obligations expressible in event structures may be used to describe the willingness to do some actions provided that someone else does some others actions, but not the dependence on actions not performed yet (e.g., circular dependencies).

To solve this problem, we have extended event structures with a new enabling relation, which allows for reasoning about circular dependencies among events. To the best of our knowledge, the other event-based models appeared in the literature do not consider circular dependencies in the same way as we do.

With this extension we are able to propose a contract model where it is possible to obtain at the same time agreement and protection, which is in our opinion a relevant feature of contract-based computations.

To substantiate our approach, we have related it with a logic-based model for contracts. In particular, we have shown that the notion of agreement in our model correspond to that of provability in Propositional Contract Logic.

11.1 Main results

Here we summarize some of the main results obtained in this dissertation.

Agreement and protection We have proposed a model for contracts, through which we have formalized the intuitive notions of agreement and protection. This model builds over event structures (ES) to specify the obligations of participants. In Theorem 8.19, we have proposed a technique to synthesize, starting from the participants payoffs, a set of contracts which admit an agreement and at the same time protect their participants. Theorem 6.49 excluded this possibility for contracts based on (classical) event structures.

Circular causality We have introduced event structures with circular causality (CES), featuring a new enabling relation $\Vdash$ which allows for resolving circularity among the causes of events (e.g. as in $a \Vdash b$, $b \Vdash a$). CES are a conservative extension of Winskel's ES (Theorem 7.4). Respect to $\vdash$, the relation $\Vdash$ imposes a weaker constraint on the ordering of events in a trace. For instance, for an event e to be justified in a trace σ, it suffices to have an enabling $X \Vdash e$, with the events in X occurring somewhere in σ. We show that, starting from an arbitrary family of configurations $\mathcal{F}$, we can construct a CES $\hat{\mathcal{E}}(\mathcal{F})$ without $\vdash$ such that the configurations of $\hat{\mathcal{E}}(\mathcal{F})$ are exactly those in $\mathcal{F}$ (Theorem 7.32). This result strengthens one in [Win88], where the above is obtained through an event structure (with $\vdash$ only). Indeed, we have shown that the construction of $\hat{\mathcal{E}}(\mathcal{F})$ in [Win88] does not really exploit the notion of ordering imposed by $\vdash$, but it just needs the weaker notion given by $\Vdash$.

A logical view of contracts We have established correspondences between two fundamental notions of our model (namely, agreements and winning strategies) and provability in a logic-based model. The logic model we have considered is Propositional Contract Logic (PCL [BZ10a]). Theorem 9.18 establishes that agreement in conflict-free contracts corresponds to provability in Horn PCL theories. Theorem 9.19 establishes that the sequences of atoms respecting the order imposed by proofs in PCL can be projected to winning strategies for all participants in the game-theoretic contract model.

11.2 Future work

Our formalisation of contracts builds upon a very abstract model of concurrent computations, namely event structures and its extension with circular causality, to provide general notions of agreement and protection.

We expect that specific formalisations of agreement, e.g. the one in [CGP09], can be interpreted as instances of our general notion, in the same spirit that event

structures can provide semantics to more concrete models of concurrency, e.g. CCS, π-calculus and Petri nets [Win86]. Also, studying the notion of subcontracts and the possibility of substituting a contract with an equivalent one would be an interesting development of our approach.

Aiming at generality, we have almost neglected some relevant issues, e.g. devising efficient decision procedures for agreements. Although in the most general setting (infinite event structures, arbitrary payoff functions) we come up against the problem of undecidability, such kind of results can be obtained by considering suitable subclasses of event structures/payoff functions (e.g. model checking temporal logic on finite representations of infinite event structures, as in [Pen97]). In particular, an efficient algorithm to compute reachable events in finite, conflict-free CES could be used to decide provability in the Horn fragment of PCL(which would be useful, since provability in full PCL is PSPACE complete).

Also, extending our contract model with temporal deadlines and, more in general, with quantitative aspects (like e.g. probabilities) seems to be feasible, along the lines of analogous extensions of events structures [Kat96].

A qualitative enhancement of the notion of agreement could consider reputation information about participants: reaching an agreement may not be the only key to decide which participant is the best partner.

In our work every single participant advertises her own unmodifiable contract, and then matching contracts are searched to establish an agrement. It could be interesting however, to considering a phase of negotiation, preliminary to the agreement-phase, in order to reach a better agreement, or an agreement at all, by weakening/strengthening some constraints.

The issue of circular dependencies among events has been addressed also in the Petri nets' world. In [BCP13] a notion of *lending* Petri nets (LPNs) has been introduced. In LPNs places are partitioned into two sets: lending places and normal ones. A transition may be executed even if some of the lending places in the preset are not marked, thus *borrowing* tokens from such places. A successful computation in an LPN is a computation where all the borrowed tokens are given back. LPNs with some additional constraints, *contract nets*, have been then developed as a concrete counterpart of logical contracts specified as Horn PCL formulae. A correspondence between CES and contract nets can be established. Successful computations in a contract net correspond to configurations in the associated CES and *vice versa*. Indeed, borrowing tokens in contract nets is similar to firing events on credit. However in a CES events may be taken on credit without any restriction, whereas this is not possible in contract net, hence computations in contract nets are somehow less liberal than in CES. We also conjecture that urgent actions in LPNs, i.e. those actions which preserve the ability to reach an honoured marking, correspond to urgent events in the associated CES.

Our notion of configuration (Def. 7.5) assumes the axiom of finite causes, i.e. it requires that every event in a configuration has a *finite* justification, both in the past (through both kinds of enablings), and in the future (through $\Vdash$-enablings).

An interesting variant of our theory could be obtained by dropping the axiom of finite causes on the events taken on credit. Consider e.g. the CES:

$$e_0 \twoheadleftarrow e_1 \twoheadleftarrow e_2 \twoheadleftarrow e_3 \twoheadleftarrow e_4 \twoheadleftarrow \cdots$$

it might be arguable whether the set $C = \{e_i \mid i \geq 0\}$ has to be considered a configuration or not. For instance, if the CES models an ever-growing debit (similarly to the money-lender scenario of Example 7.10) the borrower would reasonably *not* consider C as a successful execution of the system. Indeed, Def. 7.5 rules out C as a configuration, because, for all i, there exists no *finite* trace containing e_i (hence the only configuration there is the empty one). However, in some scenarios the ability of honouring a debt "at the limit" could be acceptable. To drop the axiom of finite causes on the events taken on credit, we should also allow for *infinite* traces, e.g. $\langle e_0\, e_1 \ldots \rangle$ in the example above. This modification would make the set C above a configuration, at the cost of losing the finiteness property (Def. 7.28), and all the properties deriving from it (e.g. Lemma 7.14). Furthermore, because of the presence of infinite chains of $\Vdash$-enablings, the rules defining reachable events for conflict-free CES (Def. 7.38) must be interpreted coinductively, by allowing for infinite derivations. Note that not all infinite derivations are acceptable, e.g. an infinite path of rules $\vdash_{\hat{\mathcal{R}}}$ would violate the axiom of finite causes for $\vdash$-enablings. We conjecture that reachable events are those for which there exists a derivation where each infinite path contains an infinite number of occurrences of rule $\Vdash_{\hat{\mathcal{R}}}$.

Bibliography

[A+10] Michael Armbrust et al. A view of cloud computing. *Comm. ACM*, 53(4):50–58, 2010.

[AD94] Rajeev Alur and David L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.

[AL93] Martín Abadi and Leslie Lamport. Composing specifications. *ACM Transactions on Programming Languages and Systems*, 15(1), 1993.

[AP93] Martín Abadi and Gordon D. Plotkin. A logical view of composition. *Theoretical Computer Science*, 114(1), 1993.

[BBCP04] Paolo Baldan, Nadia Busi, Andrea Corradini, and G. Michele Pinna. Domain and event structure semantics for petri nets with read and inhibitor arcs. *Theoretical Computer Science*, 323(1-3):129–189, 2004.

[BCGZ13] Massimo Bartoletti, Tiziana Cimoli, P. Di Giamberardino, and Roberto Zunino. Contract agreements via logic. Submitted. Available online at `tcs.unica.it/papers/ces-pcl-long.pdf`., 2013.

[BCM01] Paolo Baldan, Andrea Corradini, and Ugo Montanari. Contextual Petri nets, asymmetric event structures, and processes. *Inf. Comput.*, 171(1):1–49, 2001.

[BCP13] Massimo Bartoletti, Tiziana Cimoli, and G. Michele Pinna. Lending Petri nets and contracts. In *Proc. FSEN*, 2013.

[BCPZ12a] Massimo Bartoletti, Tiziana Cimoli, G. Michele Pinna, and Roberto Zunino. Circular causality in event structures. In *ICTCS*, 2012.

[BCPZ12b] Massimo Bartoletti, Tiziana Cimoli, G. Michele Pinna, and Roberto Zunino. An event-based model for contracts. In *Proc. PLACES*, 2012.

[BCPZ13] Massimo Bartoletti, Tiziana Cimoli, G. Michele Pinna, and Roberto Zunino. Circular causality in event structures. Submitted. Available online at `tcs.unica.it/papers/ces-long.pdf`. A preliminary version of this paper has been presented at ICTCS 2012, 2013.

[BCZ13] Massimo Bartoletti, Tiziana Cimoli, and Roberto Zunino. A theory of agreements and protection. In *Proc. POST*, volume 7796 of *LNCS*. Springer, 2013.

[BM07] Maria Grazia Buscemi and Ugo Montanari. CC-Pi: A constraint-based language for specifying service level agreements. In *ESOP*, 2007.

[BM08] Maria Grazia Buscemi and Ugo Montanari. Open bisimulation for the Concurrent Constraint Pi-Calculus. In *Proc. ESOP*, 2008.

[BM11] Maria Grazia Buscemi and Ugo Montanari. Cc-pi: A constraint language for service negotiation and composition. In *Results of the SENSORIA Project*, pages 262–281. 2011.

[Bou90] Gérard Boudol. Flow event structures and flow nets. In *Semantics of Systems of Concurrent Processes*, volume 469 of *Lecture Notes in Computer Science*, pages 62–95. Springer, 1990.

[BSTZ12] Massimo Bartoletti, Alceste Scalas, Emilio Tuosto, and Roberto Zunino. Honesty by typing. *CoRR*, abs/1211.2609, 2012.

[BTZ11] Massimo Bartoletti, Emilio Tuosto, and Roberto Zunino. Contracts in distributed systems. In *ICE*, 2011.

[BTZ12a] Massimo Bartoletti, Emilio Tuosto, and Roberto Zunino. Contract-oriented computing in CO_2. *Scientific Annals in Computer Science*, 22(1):5–60, 2012.

[BTZ12b] Massimo Bartoletti, Emilio Tuosto, and Roberto Zunino. On the realizability of contracts in dishonest systems. In *Proc. COORDINATION*, volume 7274 of *Lecture Notes in Computer Science*, pages 245–260. Springer, 2012.

[BZ07] Mario Bravetti and Gianluigi Zavattaro. Towards a unifying theory for choreography conformance and contract compliance. In *Software Composition*, 2007.

[BZ09a] Massimo Bartoletti and Roberto Zunino. A logic for contracts. Technical Report DISI-09-034, University of Trento, 2009.

[BZ09b] Mario Bravetti and Gianluigi Zavattaro. A theory of contracts for strong service compliance. *Mathematical Structures in Computer Science*, 19(3):601–638, 2009.

[BZ10a] Massimo Bartoletti and Roberto Zunino. A calculus of contracting processes. In *Proc. LICS*, 2010.

[BZ10b] Massimo Bartoletti and Roberto Zunino. Primitives for contract-based synchronization. In *Proc. ICE*, 2010.

[CCLP06] Samuele Carpineti, Giuseppe Castagna, Cosimo Laneve, and Luca Padovani. A formal account of contracts for web services. In *WS-FM*, volume 4184 of *Lecture Notes in Computer Science*. Springer, 2006.

[CGP09] Giuseppe Castagna, Nils Gesbert, and Luca Padovani. A theory of contracts for web services. *ACM Transactions on Programming Languages and Systems*, 31(5), 2009.

[CH12] Krishnendu Chatterjee and Thomas A. Henzinger. A survey of stochastic ω-regular games. *J. Comput. Syst. Sci.*, 78(2):394–413, 2012.

[CL06] Samuele Carpineti and Cosimo Laneve. A basic contract language for web services. In *ESOP*, 2006.

[CP09] Giuseppe Castagna and Luca Padovani. Contracts for mobile processes. In *Proc. CONCUR*, 2009.

[DKLW07] G. Decker, O. Kopp, F. Leymann, and M Weske. BPEL4Chor: Extending BPEL for modeling choreographies. In *Proc. ICWS*, 2007.

[DS03] Randall Davis and Reid G. Smith. Negotiation distributed as a metaphor for problem solving. In *Communication in Multiagent Systems*, pages 51–97, 2003.

[FM97] Matt Fairtlough and Michael Mendler. Propositional lax logic. *Information and Computation*, 137(1), 1997.

[GHM00] Andrew Goodchild, Charles Herring, and Zoran Milosevic. Business contracts for b2b. In *ISDO*, 2000.

[GMR10] Gregor Gößler, Daniel Le Métayer, and Jean-Baptiste Raclet. Causality analysis in contract violation. In *Proc. RV*, 2010.

[Hen11] Anders Starcke Henriksen. *Adversarial Models for Cooperative Interactions*. PhD thesis, Department of Computer Science, University of Copenhagen, 2011.

[HKZ12] Tom Hvitved, Felix Klaedtke, and Eugen Zălinescu. A trace-based model for multiparty contracts. *JLAP*, 81(2):72–98, 2012.

[HM10] Thomas T. Hildebrandt and Raghava Rao Mukkamala. Declarative event-based workflow as distributed dynamic condition response graphs. In *Proc. PLACES*, volume 69 of *EPTCS*, pages 59–73, 2010.

[HMB+11] Kohei Honda, Aybek Mukhamedov, Gary Brown, Tzu-Chun Chen, and Nobuko Yoshida. Scribbling interactions with a formal foundation. In *Distributed Computing and Internet Technology*, volume 6536 of *LNCS*. Springer, 2011.

[HYC08] Kohei Honda, Nobuko Yoshida, and Marco Carbone. Multiparty asynchronous session types. In *POPL*, 2008.

[Kat96] Joost-Pieter Katoen. *Quantitative and qualitative extensions of event structures*. PhD thesis, University of Twente, 1996.

[KBR+05] N. Kavantzas, D. Burdett, G. Ritzinger, T. Fletcher, Y. Lafon, and C Barreto. Web Services Choreography Description Language v. 1.0, 2005.

[KNS08] Karl Krukow, Mogens Nielsen, and Vladimiro Sassone. A logical framework for history-based access control and reputation systems. *Journal of Computer Security*, 16(1):63–101, 2008.

[Kra01] Sarit Kraus. Automated negotiation and decision making in multiagent environments. In *EASSS*, pages 150–172, 2001.

[Lan93] Rom Langerak. Bundle event structures: a non-interleaving semantics for lotos. In *FORTE '92*, volume C-10 of *IFIP Transactions*, pages 331–346. North-Holland, 1993.

[LG09] Xavier Leroy and Hervé Grall. Coinductive big-step operational semantics. *Inf. Comput.*, 207(2):284–304, 2009.

[LPSS11] Alessio Lomuscio, Wojciech Penczek, Monika Solanki, and Maciej Szreter. Runtime monitoring of contract regulated web services. *Fundam. Inform.*, 111(3):339–355, 2011.

[M+11] Daniel Le Métayer et al. Liability issues in software engineering: the use of formal methods to reduce legal uncertainties. *Comm. ACM*, 54(4):99–106, 2011.

[Mai03] Patrick Maier. Compositional circular assume-guarantee rules cannot be sound and complete. In *FoSSaCS*, volume 2620 of *Lecture Notes in Computer Science*, pages 343–357. Springer, 2003.

[MPW92] Robin Milner, Joachim Parrow, and David Walker. A Calculus of Mobile Processes, I and II. *Information and Computation*, 100(1), 1992.

[NPW81] Mogens Nielsen, Gordon D. Plotkin, and Glynn Winskel. Petri nets, event structures and domains, part i. *Theoretical Computer Science*, 13:85–108, 1981.

[Pad09] Luca Padovani. Contract-based discovery and adaptation of web services. In *Proc. SFM*, 2009.

[PE10] Daniele Porello and Ulle Endriss. Modelling multilateral negotiation in linear logic. In *ECAI*, pages 381–386, 2010.

[Pen97] Wojciech Penczek. Model-checking for a subclass of event structures. In *Proc. TACAS*, volume 1217 of *Lecture Notes in Computer Science*, pages 145–164. Springer, 1997.

[Pfe00] Frank Pfenning. Structural cut elimination - I. intuitionistic and classical logic. *Information and Computation*, 157(1/2):84–141, 2000.

[PS12] Cristian Prisacariu and Gerardo Schneider. A dynamic deontic logic for complex contracts. *The Journal of Logic and Algebraic Programming (JLAP)*, 81(4), 2012.

[RSE08] Franco Raimondi, James Skene, and Wolfgang Emmerich. Efficient online monitoring of web-service slas. In *SIGSOFT FSE*, pages 170–180, 2008.

[Sar93] Vijay Saraswat. *Concurrent Constraint Programming*. MIT Press, 1993.

[SBMG07] Luke Simon, Ajay Bansal, Ajay Mallya, and Gopal Gupta. Co-logic programming: Extending logic programming with coinduction. In *Proc. ICALP*, 2007.

[SMBG06] Luke Simon, Ajay Mallya, Ajay Bansal, and Gopal Gupta. Coinductive logic programming. In *Logic Programming*, pages 330–345. Springer, 2006.

[UW97] Jeffrey D. Ullman and Jennifer Widom. *A First Course in Database Systems*. Prentice-Hall, 1997.

[vdALM+10] Wil M. P. van der Aalst, Niels Lohmann, Peter Massuthe, Christian Stahl, and Karsten Wolf. Multiparty contracts: Agreeing and implementing interorganizational processes. *Comput. J.*, 53(1):90–106, 2010.

[VV01] Mahesh Viswanathan and Ramesh Viswanathan. Foundations for circular compositional reasoning. In *ICALP*, 2001.

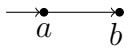
[Win86] Glynn Winskel. Event structures. In *Advances in Petri Nets*, pages 325–392, 1986.

[Win88] Glynn Winskel. An introduction to event structures. In *REX Workshop*, pages 364–397, 1988.

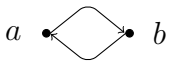
[WN95] Glynn Winskel and Mogens Nielsen. Models for concurrency. In *Handbook of Logic in Computer Science*, pages 1–148. Oxford University Press, 1995.

List of definitions

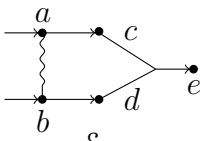
List of event structures



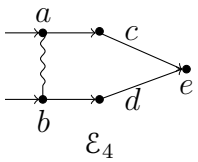
$\mathcal{E}_1$



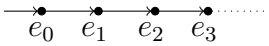
$\mathcal{E}_2$



$\mathcal{E}_3$

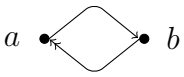


$\mathcal{E}_4$

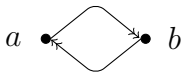


$\mathcal{E}_5$

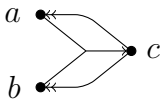
$\mathcal{E}_6$



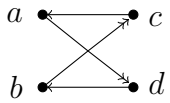
$\mathcal{E}_7$



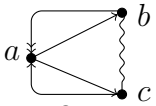
$\mathcal{E}_8$



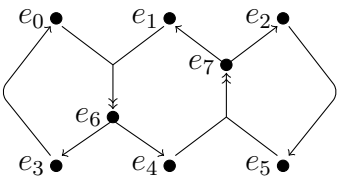
$\mathcal{E}_9$



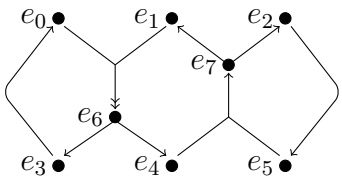
$\mathcal{E}_{10}$



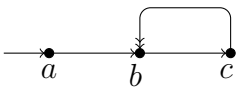
$\mathcal{E}_{11}$



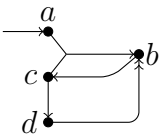
$\mathcal{E}_{12}$



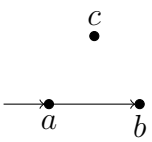
$\mathcal{E}_{13}$



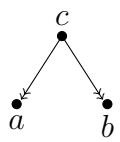
$\mathcal{E}_{14}$



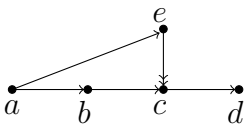
$\mathcal{E}_{15}$



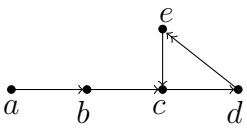
$\mathcal{E}_{16}$



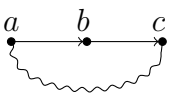
$\mathcal{E}_{17}$



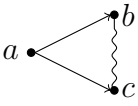
$\mathcal{E}_{18}$



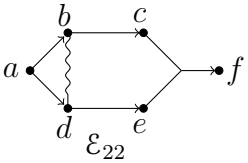
$\mathcal{E}_{19}$



$\mathcal{E}_{20}$



$\mathcal{E}_{21}$



$\mathcal{E}_{22}$