The Essence of Event-B from an Abstract Interpretation Perspective

Patrick Cousot [0000-0003-0101-9953]

Courant Institute, NYU, New York, NY, USA, pcousot@cims.nyu.edu

Abstract. We explain the basic concepts of Event-B, that is the specification of machines and refinement, in light of abstract interpretation. After formally defining the trace semantics of systems handled by Event-B, we show that machines are specifications of abstractions of this semantics while refinement is an inverse abstract interpretation reducing the degree of nondeterminism of this trace semantics. We conclude by proposing in induction principle for program verification exploiting both property abstraction and program refinement.

Keywords: Event-B · Abstraction · Abstract machine · Semantics · Refinement · Verification · Abstract interpretation · Formal methods.

1 Introduction

I have had the pleasure to have my office close to that of Jean-Raymond Abrial during my thesis in Grenoble and to share a common passion for climbing, mainly in the Vercors limestone massif and the French and Swiss granite and glacier Alps. At the time Jean-Raymond was working on *data semantics* [1] and started to be increasingly interested in formal program verification and construction. He had strong ideas on computer science, in particular program design and verification, that he explained very clearly, and was always ready for discussion on many subjects with a young ignorant trying to understand abstraction.

When Jean-Raymond left Grenoble, I was given the charge of his class on data structures, programming, and verification that gave me a strong taste for teaching, in particular formal methods.

We met later in conferences, meetings, summer schools, and seminars in particular during his stay at ETH Zurich and sometimes also by chance, like in Venice, on the Ponte dell'Accademia, where, for a long time on the bridge, we passionately discussed the genius of Venice architecture and urbanism design.

Unfortunately, Jean-Raymond left Grenoble before I had established the foundations of abstract interpretation with Radhia and therefore we have never had scientific collaborations. So I was surprised to be invited by Egon Boerger and Yamine Aït-Ameur to contribute to this Gedenkschrift with the proposed theme of "Relating program verification by abstract interpretation with proofs in Event-B". Since no one, including myself, seemed to have understood the correspondence during the "Workshop on Refinement and Abstraction" (ETL Osaka,

Japan, 15–17 November 1999), and I had made no progress since then, I thought taking on the challenge would be interesting.

Although I had read the B-book [2] decades ago, I had to learn Event-B, almost from scratch [4,7] while knowing that I could not absorb the abundant literature on Event-B (`event-b.org`) in the allotted time.

It turns out that I could slightly extend the challenge to an explanation of the semantics of Event-B in terms of abstract interpretation [16]. Of course this brings nothing new to Event-B but this formalization helps me to better understand two related forms of abstractions when respectively applied to models (as in Event-B) and to properties of these models (as in abstract interpretation). It might also be useful to conceive Event-B refinements by calculational design of the concrete from the given abstract using a formal correspondence between their semantics (such as a Galois connection or equivalent).

Event-B [4] is an evolution of the B method [2] to specify a reactive system and provide "a mathematical model of its dynamic behaviour" [3,7]. The description is reduced to contexts and machines. It proposes to observe states via events and reuses B's refinement.

The behavior of a reactive system is described by a concrete machine, more precisely its semantics defining the possible executions of machines, a name for a computer system. This semantics, described in part section 2, is a finite or infinite set of traces which is the mathematical model of the system dynamic behavior mentioned by Jean-Raymond. This trace specified by abstraction into a machine, as shown in part section 3. The objective of an Event-B development is to ultimately define this concrete semantics through several refinement steps formalized in sections 4 and 5. Finally, section 6 proposes a general induction principle to combine property abstraction and program refinement.

I assume a minimal knowledge of Event-B [4, Ch. 5. "The Event-B modeling notation and proof obligation rules" and Ch. 14. "Mathematical models for proof obligations"], [26,31] and abstract interpretation [13] (in particular the formalization of property abstraction using closure operators or Galois connections [17] as recalled in [15, Ch. 11]). For Galois connections, I write $\langle C, \sqsubseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A, \leqslant\rangle$ for $\langle C, \sqsubseteq\rangle$ and $\langle A, \leqslant\rangle$ are posets, $\alpha \in C \to A$ is the abstraction, $\gamma \in A \to C$ is the concretization, and $\forall x \in C, y \in A \,.\, \alpha(x) \leqslant y \Leftrightarrow x \sqsubseteq \gamma(y)$. One adjoint uniquely determine the other i.e. if $\langle C, \sqsubseteq\rangle \underset{\alpha}{\leftrightarrows} \langle A, \leqslant\rangle$ with unspecified concretization then there is a unique γ such that $\langle C, \sqsubseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A, \leqslant\rangle$. For upper closure operators (increasing, extensive (also expansive or inflationary), and idempotent), $\langle A, \leqslant\rangle = \langle C, \sqsubseteq\rangle$ and γ is the identity $\mathbb{1}$ on $C = A$. I use transformers $\mathsf{post}[r] \triangleq \boldsymbol{\lambda}\, P \bullet \{\mathsf{s}' \mid \exists \mathsf{s} \in P \,.\, \langle \mathsf{s}, \mathsf{s}'\rangle \in r\}$, $\mathsf{pre}[r] \triangleq \mathsf{post}[r^{-1}]$, $\widetilde{\mathsf{pre}}[r] \triangleq \neg \circ \mathsf{pre}[r] \circ \neg$, and $\widetilde{\mathsf{post}}[r] \triangleq \widetilde{\mathsf{pre}}[r^{-1}]$. In particular, $f(X) = \{f(x) \mid x \in X\} = \mathsf{post}[f]X$ is the post image of a function (graph).

2 The Semantics of Event-B Systems

An (abstract) machine "represents the mathematical model of an entire system as seen from a unique global abstract point of view." [3, p. 177] (where it was

originally called "(abstract) system"). The system is closed in that it does not allow transfer of information in or out of the system. Event-B specifies how to define the semantics of an abstract machine. I adopt the inverse approach, starting by what should be the semantics of an Event-B system and then, in section 3, how to define it, by abstraction into a machine.

2.1 States and Events. An Event-B specification usually starts by defining *contexts* that is a set of states $\mathsf{S} \in \wp(\mathbb{S})$ and operations on these states by means of carrier sets, constants, and axioms (from which theorems can be deduced). The definition of a set of events $\mathsf{E} \in \wp(\mathbb{E})$ is a component/clause **events** of a machine. Each event $\mathsf{e} \in \mathbb{E}$ is specified by an event structure [3, fig. 5.10] that describes a potential transition between a pre-state and a post-state. If there exists values of the parameters (in the clause **any**) for which the predicates of the **where** clause hold of the pre-state then actions (in the clause **then**) specify the transition. These actions [3, Sect. 5.1.8] specify the change from a prestate to a poststate which, in general, is nondeterministic.

Event-B uses set-theory and first-order logic [4, Ch. 9] to define the correct set of state definitions $\mathbb{S}$ in contexts [4, Sect. 5.1.3] and correct set of event definitions $\mathbb{E}$ in machines [4, Sect. 5.1.7]. To avoid inconsistencies, these set definitions must sometimes include proof obligations, in particular when using partially defined objects (such as division by zero). "*a failed proof reveals a bug*" [4, p. 40]. For simplicity, and since this is a classic problem in mathematics when using partially defined objects, I consider that sets, constants, and basic operations in event guards and actions of Event-B are well-defined by construction (i.e. all well-definedness proof obligations [4, Sect. 5.2.12] have been proved). For other proof obligations, I will show how they derive from the definition of the traces and transition semantics of machines.

Event-B offers the possibility of accompanying definitions in contexts with theorems requiring a proof [4, Sect. 5.2.11]. This is a practical way of organizing proofs that I ignore, as well as how they must be proved. I assume that all theorems in an Event-B definition can be proved, which amounts to saying that the Event-B semantics is well-defined.

So I am interested in the semantics of Event-B more than in its syntax and well-definedness, knowing that I neglect an important aspect of Event-B when implemented in tools like the Rodin platform [5,34].

2.2 Execution Traces. Event-B specifications ultimately define traces [4, Sect. 14.3.2]. Traditionally, traces $\mathsf{s}_0 \xrightarrow{\mathsf{e}_0} \mathsf{s}_1 \xrightarrow{\mathsf{e}_1} \dots$ are non-empty sequences of states $\mathsf{s}_i \in \mathsf{S} \in \wp(\mathbb{S})$ separated by events $\mathsf{e}_i \in \mathsf{E} \in \wp(\mathbb{E})$ meaning that in prestate s_i event e_i yields poststate s_{i+1}. I encode such traces as a sequence of pairs $\langle s_0, \mathsf{e}_0\rangle\langle s_1, \mathsf{e}_1\rangle\dots$ Traces must start with an initial state and init event [4, Sect. 2.4.16]. The finite traces are

$$\pi \in \mathbb{t}^+\langle \mathsf{S}, \mathsf{E}\rangle \triangleq \{\pi \in [0, n] \to (\mathsf{S} \times \mathsf{E}) \mid n \in \mathbb{N} \wedge \pi_0 \in \mathsf{init}\} \tag{1}$$

that is a finite sequence of a state $\mathsf{stof}(\pi_i) \triangleq (\pi_i)_1$ in S followed by an event $\mathsf{evtof}(\pi_i) \triangleq (\pi_i)_2$ in E, $i \in \mathbb{N}$ and starting from an initial event (that I replace

by a pair of initial state and an initial event in init for uniformity of the trace definition). I call such pairs $\langle \mathsf{s}, \mathsf{e} \rangle \in \mathsf{S} \times \mathsf{E}$ trace transitions. The trace transitions $\ldots \langle \mathsf{s}, \mathsf{e} \rangle \langle \mathsf{s}', \ldots \rangle \ldots$ in a trace follow from the definition of the event e which has a guard that must hold in state s and a deterministic ($:=$) or (possibly unbounded) nondeterministic ($:|$ or $:\in$) action assigning to state s to get successor state s' [4, Sect. 5.1.8]. The relation between the pre and post states can also be specified by a before-after-predicate [4, Ch. 2.4.4].

The infinite traces describe situations where the system evolves for ever.

$$\pi \in \mathbb{t}^\infty\langle \mathsf{S}, \mathsf{E} \rangle \triangleq \{\pi \in \mathbb{N} \to (\mathsf{S} \times \mathsf{E}) \mid \pi_0 \in \mathsf{init}\} \tag{2}$$

We denote concatenation of traces or sets of traces by juxtaposition, sometimes using $\cdot$ to make concatenation more explicit. If $\pi_1 \in \mathbb{t}^\infty\langle \mathsf{S}, \mathsf{E} \rangle$ then $\pi_1 \cdot \pi_2 = \pi_1$.

2.3 Trace Semantics. Given states $\mathsf{S} \in \wp(\mathbb{S})$ and events in $\mathsf{E} \in \wp(\mathbb{E})$, a trace semantics T is a finite or infinite set of finite or infinite execution traces on S and E. So T belongs to the domain $\mathbb{T}^{+\infty}\langle \mathsf{S}, \mathsf{E} \rangle$ defined as

$$\mathbb{T}^+\langle \mathsf{S}, \mathsf{E} \rangle \triangleq \wp(\mathbb{t}^+\langle \mathsf{S}, \mathsf{E} \rangle) \setminus \{\varnothing\} \qquad \mathbb{T}^\infty\langle \mathsf{S}, \mathsf{E} \rangle \triangleq \wp(\mathbb{t}^\infty\langle \mathsf{S}, \mathsf{E} \rangle) \setminus \{\varnothing\}$$
$$\mathbb{T}^{+\infty}\langle \mathsf{S}, \mathsf{E} \rangle \triangleq \mathbb{T}^+\langle \mathsf{S}, \mathsf{E} \rangle \cup \mathbb{T}^\infty\langle \mathsf{S}, \mathsf{E} \rangle$$

where $\wp$ is the powerset.

In Event-B, traces may be required to be deadlock free [4, Sect. 2.4.21] so that they must be infinite [4, Sect. 14.3.3], since 'The system always "runs" for ever' [4, p. 422]. In that case the semantics of the specified system is

$$\mathsf{T} \in \mathbb{T}^\infty\langle \mathsf{S}, \mathsf{E} \rangle \tag{3}$$

(We can consider infinite traces only since finite traces can be encoded as an infinite one with the final state and event repeated for ever.)

The trace semantics T is the strongest property of the execution traces, defined in extension. Notice also that this trace semantics covers concurrency which is handled by interleaving in Event-B, that is, assuming sequential consistency [4, Ch. 7]. However, our formalization of Event-B by abstraction of the trace semantics could be applied to weak consistency models by e.g. considering instead abstractions of the anarchic semantics of [8].

Other comparable sequential consistent specification languages like TLA+ [28] and Unity [10] have trace semantics. In Unity, there is an implicit weak fairness hypothesis (if a transition is eventually always enabled, then it is taken infinitely often). There is no such implicit fairness hypothesis in TLA+ and Event-B (see [29] for a comparison of Event-B and TLA+ on the example of distributed termination). In TLA+ the trace semantics is specified using temporal logic. Fairness requirements are specified using temporal formulas (such as WF_{vars} in [29, Fig, 1]). Similarly, Event-B makes no weak or strong process fairness hypothesis for parallel or distributed systems. Fairness has to be explicitly specified during the refinement process [7, p. 87].

2.4 Objective of Event-B. Event-B is a formal method aimed at system-level modeling and analysis which ultimate goal is to define a trace semantics

$\mathsf{T} \in \mathbb{T}^\infty\langle \mathsf{S},\, \mathsf{E}\rangle$ of the modeled discrete system by successive refinements starting from an informal, often incomplete if not incorrect, specification.

However, not all arbitrary sets of traces $\mathsf{T} \in \mathbb{T}^\infty\langle \mathsf{S},\, \mathsf{E}\rangle$ represent computations of Event-B machines. We express additional requirements on the sets T of traces using (closure) operators on $\mathbb{T}^\infty\langle \mathsf{S},\, \mathsf{E}\rangle$. Since we consider only infinite traces in Event-B, these upper or lower closures of trace semantics are much simpler than the general case considered in [22, Sect. 2.6].

2.5 History Independence. It is implicit in Even-B that the future of a computation must depend only on its current trace state, not on how that state was reached. This means that if $\pi_1 \langle \mathsf{s}, \mathsf{e}\rangle \pi_2$ and $\pi_1' \langle \mathsf{s}, \mathsf{e}\rangle \pi_2'$ are two (infinite) traces of the semantics sharing intermediate state s and (nondeterministic) event e then $\pi_1 \langle \mathsf{s}, \mathsf{e}\rangle \pi_2'$ and $\pi_1' \langle \mathsf{s}, \mathsf{e}\rangle \pi_2$ should also be infinite traces of the semantics [22, Sect. 2.6.5]. Let use define the abstraction (π_1 and π_1' are finite traces, $\langle \mathsf{s},\, \mathsf{e}\rangle$ is a trace transition, and π_2 and π_2' are (infinite) traces)

$$\mathsf{Efus}(\mathsf{T}) \triangleq \{\pi_1 \langle \mathsf{s},\, \mathsf{e}\rangle \pi_2' \mid \exists \pi_2, \pi_1' \,.\, \pi_1 \langle \mathsf{s},\, \mathsf{e}\rangle \pi_2, \pi_1' \langle \mathsf{s},\, \mathsf{e}\rangle \pi_2' \in \mathsf{T}\} \tag{4}$$

Efus is increasing and extensive for set inclusion $\subseteq$ but not necessarily idempotent. Therefore, the closure by fusion [22, Sect. 2.6.5] is defined as

$$\mathsf{Ffus}(\mathsf{T}) \triangleq \mathsf{lfp}^{\subseteq}\, \boldsymbol{\lambda}\, X \bullet \mathsf{T} \cup \mathsf{Efus}(X) \tag{5}$$

where $\mathsf{lfp}^{\sqsubseteq} F$ is the least fixpoint of the $\sqsubseteq$-increasing function F on a complete lattice for partial order $\sqsubseteq$ [33]. Ffus repeats the fusion Efus of traces infinitely many times. Ffus is an upper closure operator such that $\mathsf{Efus} \circ \mathsf{Ffus} = \mathsf{Ffus}$ and $\mathsf{Ffus} = \dot{\bigcup}_{n\in\mathbb{N}} \mathsf{Efus}^n$, pointwise [22, lem. 2.6.5 ~1]. Event-B always requires sets of traces T to be history independent, that is $\mathsf{Ffus}(\mathsf{T}) = \mathsf{T}$. This is an abstraction onto $\mathbb{T}^{\mathrm{F}}\langle \mathsf{S},\, \mathsf{E}\rangle \triangleq \{\mathsf{T} \in \mathbb{T}^\infty\langle \mathsf{S},\, \mathsf{E}\rangle \mid \mathsf{Ffus}(\mathsf{T}) = \mathsf{T}\}$,

$$\langle \mathbb{T}^\infty\langle \mathsf{S},\, \mathsf{E}\rangle, \subseteq\rangle \underset{\mathsf{Ffus}}{\overset{\mathbb{1}}{\leftrightarrows}} \langle \mathbb{T}^{\mathrm{F}}\langle \mathsf{S},\, \mathsf{E}\rangle, \subseteq\rangle \tag{6}$$

where $\mathbb{1}$ is the identity function.

2.6 Further Hypotheses on the Trace Semantics. There are several different ways to use Event-B to produce a sequential, concurrent or distributed algorithm, which may require additional restrictions on traces. For example, physical systems are generally considered to be finite rather than mathematically infinite so that the nondeterminism of computations is necessarily bounded as considered in section 2.7. In particular, determinism considered in Sect. 2.8 is ultimately necessary to get a sequential algorithm [4, Ch. 15]. Event convergence considered in Sect. 2.10 prevents starvation by requiring that a specific subset of events cannot take control of the system forever so that the other events would be postponed for ever [4, p. 64]. These restrictions are abstractions of the trace semantics and can be composed to get the appropriate trace model of the considered closed system.

2.7 Bounded Nondeterminism. I could not find in [4] the restriction that nondeterminism should be bounded. The number of events is definitely finite in [4] (since the can only be given finitely many names in machines), but infinite sets of states do not look to be prohibited.

However, the method for proving inevitability, such as event-convergence, uses variant functions in the naturals or in finite sets since the numeric variant proof obligation rule [4, Sect. 5.2.7] requires the numeric variant function to be a natural number.

A minor point is the variant function must in general depend on the current and initial states [19, p. 290]. This can always be done indirectly by adding auxiliary variables.

A more important point is that the requirement that the numeric variant function be a natural number yields a sound but incomplete proof method for unbounded nondeterminism [22, Th. 5.2.6.3~1], for which transfinite ordinals may be required [22, Sect. 5.2.6], [24].

This restriction to naturals may result from the initial decision [7] to reuse the B prover [6]. Moreover, unbounded nondeterminism as well as fairness in parallel or distributed programs can be understood as the limit of bounded nondeterminism and fairness when the unknown bound tends to infinity. This simplifies reasonings by eliminating this unknown bound so that the details of an implementation can be deliberately ignored, at least in the first steps of the Event-B development.

Yet the physical realization of unbounded nondeterminism and fairness is questionable so that a bound is not that unreasonable in practice. Finally, in a development by successive refinements where the final result must be deterministic, the nondeterminism must be bounded at some point in the development, at which point all intermediate sets of traces T in an Event-B development are subject to locally bounded nondeterminism that is $\mathsf{T} = \mathsf{BNd}(\mathsf{T})$ where BNd, defined as follows

$$\mathsf{BNd}(\mathsf{T}) \triangleq \{\pi \in \mathsf{T} \mid \forall \pi_1 \cdot \langle \mathsf{s}, \mathsf{e}\rangle \cdot \pi_2 = \pi \,.\, |\{\mathsf{s}' \mid \pi_1 \cdot \langle \mathsf{s}, \mathsf{e}\rangle \cdot \langle \mathsf{s}', \mathsf{e}'\rangle \cdot \pi_2' \in \mathsf{T}\}| \in \mathbb{N}\}, \quad (7)$$

meaning that for any state s encountered on a trace $\pi_1 \cdot \langle \mathsf{s}, \mathsf{e}\rangle \cdot \pi_2$ with next event e the number of possible successors s' on this and other possible traces $\pi_1 \cdot \langle \mathsf{s}, \mathsf{e}\rangle \cdot \langle \mathsf{s}', \mathsf{e}'\rangle \cdot \pi_2'$ with same prefix $\pi_1 \cdot \langle \mathsf{s}, \mathsf{e}\rangle$ is finite. Otherwise stated the number of possible successors s' of state s when event e occurs is finite ($|S|$ denotes the cardinality of set S).

Remark 1 (Locally versus globally bounded nondeterminism). Nondeterminism is local in (7), that is weakly bounded, since the maximum number of possible successive repetitions of an event e depends on the position of the event in the trace. This means that an event may take longer and longer to occur (e.g. after 1 then 2, 4, 8, ... computation steps) which seems compatible with the "numeric variant proof obligation rule" [4, Sect. 5.2.7]. For globally bounded nondeterminsm, the stronger definition would be $\forall \mathsf{e} \in \mathbb{E} \,.\, \exists n \in \mathbb{N} \,.\, \forall \pi_1 \langle \mathsf{s}, \mathsf{e}\rangle \pi_2 = \pi \,.\, |\{\mathsf{s}' \mid \pi_1 \langle \mathsf{s}, \mathsf{e}\rangle \langle \mathsf{s}', \mathsf{e}'\rangle \pi_2' \in \mathsf{T}\}| < n\}$ where the same bound n on the number of possible consecutive occurrences of an event e is the same wherever this event occurs in an execution trace. □

$\mathsf{BNd} \in \mathbb{T}^\infty\langle \mathsf{S}, \mathsf{E}\rangle \to \mathbb{T}^\infty\langle \mathsf{S}, \mathsf{E}\rangle$ in (7) is reductive ($T \subseteq \mathsf{BNd}(T)$) and idempotent ($\mathsf{BNd}(\mathsf{BNd}(T)) = \mathsf{BNd}(T)$) but not increasing. For example with integer states, $T_1 = \{00\}$, $T_2 = \{0n \mid n \in \mathbb{N}\}$, we have $T_1 \subseteq T_2$ but $\mathsf{BNd}(T_1) = T_1 \notin \varnothing = \mathsf{BNd}(T_2)$. It follows that we don't have a Galois connection $\langle \mathbb{T}^\infty\langle \mathsf{S}, \mathsf{E}\rangle, \subseteq\rangle \underset{\mathsf{BNd}}{\overset{\gamma_{\mathsf{BNd}}}{\leftrightarrows\!\!\!\!\twoheadrightarrow}} \langle \mathsf{BNd}(\mathbb{T}^\infty\langle \mathsf{S}, \mathsf{E}\rangle), \subseteq\rangle$ since there is no best abstraction. A counterexample is $\mathsf{T} = \{a^\infty\} \cup \{abn^\infty \mid n \in \mathbb{N}\}$. The finite trace a has infinitely many possible continuations $\{a^\infty\} \cup \{bn^\infty \mid n \in \mathbb{N}\}$ and if we eliminate this case we get $\varnothing$. But the finite trace ab has also infinitely many possible continuations $\{n^\infty \mid n \in \mathbb{N}\}$ and if we eliminate this case, it remains $\{a^\infty\}$. Taking the smallest solution $\varnothing$ is of no interest and there might be no largest solution as in $\{\underline{0}^\infty, \underline{01}^\infty, \underline{012}^\infty, \ldots\} \cup \{\{\underline{0}n \mid n \in \mathbb{N}\} \cup \{\underline{01}n \mid n \in \mathbb{N}\} \cup \{\underline{012}n \mid n \in \mathbb{N}\} \cup \ldots$ where $\underline{n}$ and $n \in \mathbb{N}$ are different states. These counterexamples are not history independent. For example $\underline{000}$ is a prefix of $\underline{0}^\infty$ and $\underline{12}^\infty$ is a suffix of $\underline{012}^\infty$ but $\underline{00012}^\infty$ is not a valid trace. Trace semantics in Event-B are always history independent, but this does not help since $\mathsf{BNd} \in \mathbb{T}^{\mathrm{F}}\langle \mathsf{S}, \mathsf{E}\rangle \to \mathbb{T}^{\mathrm{FB}}\langle \mathsf{S}, \mathsf{E}\rangle$ is not increasing either with counterexample $T_1 = \{01n^\infty \mid 0 \leqslant n \leqslant N\}$ with $N \in \mathbb{N}$ and $T_2 = \{01n^\infty \mid n \in \mathbb{N}\}$ such that $T_1 \subseteq T_2$ but $\mathsf{BNd}(T_1) = T_1 \notin \varnothing = \mathsf{BNd}(T_2)$. So we have

$$\mathsf{BNd} \in \mathbb{T}^{\mathrm{F}}\langle \mathsf{S}, \mathsf{E}\rangle \twoheadrightarrow \mathbb{T}^{\mathrm{FB}}\langle \mathsf{S}, \mathsf{E}\rangle \tag{8}$$

It is a surjective map of history independent traces in $\mathbb{T}^{\mathrm{F}}\langle \mathsf{S}, \mathsf{E}\rangle$ onto

$$\mathbb{T}^{\mathrm{FB}}\langle \mathsf{S}, \mathsf{E}\rangle \triangleq \{\mathsf{T} \in \mathbb{T}^\infty\langle \mathsf{S}, \mathsf{E}\rangle \mid \mathsf{Ffus}(\mathsf{T}) = \mathsf{T} \wedge \mathsf{BNd}(\mathsf{T}) = \mathsf{T}\} = \mathsf{BNd}(\mathsf{Ffus}(\mathbb{T}^\infty\langle \mathsf{S}, \mathsf{E}\rangle))$$

where

$$\mathsf{BNd} \circ \mathsf{Ffus} \in \mathbb{T}^\infty\langle \mathsf{S}, \mathsf{E}\rangle \twoheadrightarrow \mathbb{T}^{\mathrm{FB}}\langle \mathsf{S}, \mathsf{E}\rangle \tag{9}$$

by composing (6) and (8) in the context $\langle \mathsf{S}, \mathsf{E}\rangle \in \wp(\mathbb{S}) \times \wp(\mathbb{E})$. The bounded Event-B semantics T are defined as those in $\mathbb{T}^{\mathrm{FB}}\langle \mathsf{S}, \mathsf{E}\rangle$ that is $\mathsf{T} \in \mathbb{T}^\infty\langle \mathsf{S}, \mathsf{E}\rangle$ such that $\mathsf{BNd}(\mathsf{Ffus}(\mathsf{T})) = \mathsf{T}$. We have $\mathsf{Efus}(\mathsf{BNd}(\mathsf{Ffus}(\mathsf{T}))) = \mathsf{BNd}(\mathsf{Ffus}(\mathsf{T}))$ which implies $\mathsf{Ffus}(\mathsf{BNd}(\mathsf{Ffus}(\mathsf{T}))) = \mathsf{BNd}(\mathsf{Ffus}(\mathsf{T}))$ to that BNd preserves history independence.

For simplicity we do not consider both cases of unbounded and bounded nondeterminism, as, for unbounded nondeterminism, it is sufficient to define $\mathsf{BNd}(\mathsf{T}) \triangleq \mathsf{T}$ and to replace the naturals in variants by transfinite ordinals [19,22].

2.8 Determinism. Determinism is the limit of bounded nondeterminism where each state has at most one possible successor. In that case, the semantics of the ultimate refinement should be deterministic, that is $\mathsf{T} = \mathsf{Det}(\mathsf{T})$ where

$$\mathsf{Det}(\mathsf{T}) \triangleq \{\pi \in \mathsf{T} \mid \forall \pi_1\, \langle \mathsf{s}, \mathsf{e}\rangle\, \pi_2 = \pi \,.\, \forall \pi_1\, \langle \mathsf{s}, \mathsf{e}\rangle\, \pi_2' \in \mathsf{T} \,.\, \pi_2 = \pi_2'\} \tag{10}$$

which is reductive and idempotent but not increasing (with the same countreexample $T_1 = \{00\}$, $T_2 = \{0n \mid n \in \mathbb{N}\}$, $T_1 \subseteq T_2$ but $\mathsf{Det}(T_1) = T_1 \notin \varnothing = \mathsf{Det}(T_2)$). So we don't have a lower closure operator nor a Galois connection. We have

$$\mathsf{Det} \circ \mathsf{Ffus} \in \mathbb{T}^\infty\langle \mathsf{S}, \mathsf{E}\rangle \twoheadrightarrow \mathbb{T}^{\mathrm{FD}}\langle \mathsf{S}, \mathsf{E}\rangle \tag{11}$$

where $\mathbb{T}^{\mathrm{FD}}\langle \mathsf{S}, \mathsf{E}\rangle \triangleq \{\mathsf{T} \in \mathbb{T}^\infty\langle \mathsf{S}, \mathsf{E}\rangle \mid \mathsf{Ffus}(\mathsf{T}) = \mathsf{T} \wedge \mathsf{Det}(\mathsf{T}) = \mathsf{T}\} = \mathsf{Det}(\mathsf{Ffus}(\mathbb{T}^\infty\langle \mathsf{S}, \mathsf{E}\rangle))$ is the set of deterministic Event-B semantics.

2.9 Definition of Inevitability. Let $\mathsf{T} \in \mathbb{T}^\infty\langle \mathsf{S}, \mathsf{E}\rangle$ be a set of traces. Let $\mathsf{P}, \mathsf{Q} \in \wp(\mathsf{S} \times \mathsf{E})$ be trace state properties (defined in extension). Q *is inevitable* from P for T, written $\mathsf{Inev}(\mathsf{P}, \mathsf{T}, \mathsf{Q})$, if and only if

$$\mathsf{Inev}(\mathsf{P}, \mathsf{T}, \mathsf{Q}) \triangleq \forall \pi \in \mathsf{T} \,.\, \forall i \in \mathbb{N} \,.\, (\pi_i \in \mathsf{P}) \Rightarrow (\exists j \geqslant i \,.\, \pi_j \in \mathsf{Q}) \tag{12}$$

that is, whenever P holds during the computation, Q will always eventually hold later during this computation (i.e. $\Box(\mathsf{P} \Rightarrow \Diamond\mathsf{Q})$ in linear temporal logic [32]). A stronger requirement is that the distance $j - i$ be bounded for any trace π of T or, even stronger, the same upper bound on the distance $j - i$ may be required to hold for all traces in T.

2.10 Event Convergence. As a special case of inevitability, traces in T may be required to be *event-convergent*, meaning that no event can successively occur for ever. More precisely, no event can occur more than a bounded number of times.

$$\begin{aligned} \mathsf{evt\text{-}cvg}(\pi) \triangleq \forall \mathsf{e} \in \mathsf{E}, i \in \mathbb{N} \,.\, \exists n \in \mathbb{N} \,.\, \forall j > i \in \mathbb{N} \,. \qquad & (13)\\ (\forall k \in [i, j] \,.\, \exists \mathsf{s}' \,.\, \pi_k = \langle \mathsf{s}', \mathsf{e}\rangle) \Rightarrow (j - i < n) & \\ \mathsf{Evt\text{-}Cvg}(\mathsf{T}) \triangleq \{\pi \in \mathsf{T} \mid \mathsf{evt\text{-}cvg}(\pi)\} & \end{aligned}$$

This is globally bounded nondeterminism, a local version $\forall \mathsf{e} \in \mathsf{E} \,.\, \exists n \in \mathbb{N} \,.\, \ldots$ is also possible as observed in remark 1. $\mathsf{Evt\text{-}Cvg}$ is a lower closure operator for $\subseteq$ [22, lem. 2.6.4∼1]. Event-B may require sets of traces T to be event convergent, that is $\mathsf{Evt\text{-}Cvg}(\mathsf{T}) = \mathsf{T}$. This is an abstraction onto

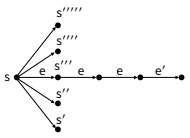
$$\mathbb{T}^{\mathrm{FBC}}\langle \mathsf{S}, \mathsf{E}\rangle \triangleq \{\mathsf{T} \in \mathbb{T}^{\mathrm{FB}}\langle \mathsf{S}, \mathsf{E}\rangle \mid \mathsf{Evt\text{-}Cvg}(\mathsf{T}) = \mathsf{T}\} = \mathsf{Evt\text{-}Cvg}(\mathbb{T}^{\mathrm{FB}}\langle \mathsf{S}, \mathsf{E}\rangle), \tag{14}$$

as follows ($\gamma_{\mathsf{Evt\text{-}Cvg}}(\mathsf{T}) \triangleq \mathsf{T} \cup \{\pi \in \mathbb{T}^{\mathrm{FB}}\langle \mathsf{S}, \mathsf{E}\rangle \mid \neg\mathsf{evt\text{-}cvg}(\pi)\}$ [15, Ex. 11.5])

$$\langle \mathbb{T}^{\mathrm{FB}}\langle \mathsf{S}, \mathsf{E}\rangle, \subseteq\rangle \underset{\mathsf{Evt\text{-}Cvg}}{\overset{\gamma_{\mathsf{Evt\text{-}Cvg}}}{\leftrightarrows}} \langle \mathbb{T}^{\mathrm{FBC}}\langle \mathsf{S}, \mathsf{E}\rangle, \subseteq\rangle \tag{15}$$

If the number $|\mathsf{E}|$ of possible events in E is finite[1], event convergence (15) implies that at any time in the computation any event will definitely occur later on.

Possible executions restricted by bounded nondeterminism (7) do limit the number of successors $\mathsf{s}', \ldots, \mathsf{s}''''$ of a state s on the right figure, while (13) limits the number of consecutive occurrences of a given even e along an execution trace.



Only some events may be required to be convergent in which case the set E of all events should be replaced in the previous definitions by the subset $E \subseteq \mathsf{E}$ of events which must be convergent. For example in the location access controller [4,

[1] In the number $|\mathsf{E}|$ of possible events in E is infinite then a new event can occur for ever. In Event-B, event must be declared as part of the machine structure and so are finite [4, Sect. 5.1.5].

Ch. 16], it is possible that no one may ever enter the location. So the pass events (corresponging to a person going from a location to another) does not belong to E. This set E may even be empty, in which case Evt-Cvg is the identity.

Finally, convergence of events in E may not be guaranteed all along the refinement process. Therefore Event-B provides the convergence attribute *status* of an event with three possible values: *convergent* (the event must strictly decrease the variant), *anticipated* (the event must not increase the variant), and *ordinary* (the event may or may not not decrease the variant) [4, p. 183]. For simplicity, I only consider *convergent* and ignore the last two possibilities. Adding a *status* component to events and a case analysis would easily take into account all three possibilities.

3 Specifying the System Semantics by Abstraction Into Machines (i.e. Discrete Transition Systems)

Assuming that a set of states S has been defined by contexts [4, Sect. 5.1.3] and a set of event identifiers E is given [4, Sect. 5.1.7], we now study how a trace semantics $\mathbb{T}^{\mathrm{EvB}}\langle \mathsf{S}, \mathsf{E}\rangle$ abstracting $\mathbb{T}^{\infty}\langle \mathsf{S}, \mathsf{E}\rangle$ can be defined in Event-B by machines [4, Sect. 5.1.5] that is transition systems (and possibly variants). Although any semantics $\mathbb{T}^{\infty}\langle \mathsf{S}, \mathsf{E}\rangle$ can be generated by a transition system (where states are a the pair of a trace and a position in that trace and transitions move forward the position in the trace), the history independent abstraction of Sect. 2.5 ensures that states of the transition system can be chosen as those states appearing on traces. So the abstraction Ffus is mandatory while the others BNd, Det, and Evt-Cvg are optional.

We consider the most demanding case by composing Ffus (6), BNd (8), and Evt-Cvg (15) in the context $\langle \mathsf{S}, \mathsf{E}\rangle \in \wp(\mathbb{S}) \times \wp(\mathbb{E})$ to get the Event-B semantics $\mathbb{T}^{\mathrm{EvB}}\langle \mathsf{S}, \mathsf{E}\rangle \triangleq \mathbb{T}^{\mathrm{FBC}}\langle \mathsf{S}, \mathsf{E}\rangle$ as an abstraction of the trace semantics $\mathbb{T}^{\infty}\langle \mathsf{S}, \mathsf{E}\rangle$, a characterization that also follows from [22, lem. 2.6.5∼3]).

$$\mathsf{Evt\text{-}Cvg} \circ \mathsf{BNd} \circ \mathsf{Ffus} \in \langle \mathbb{T}^{\infty}\langle \mathsf{S}, \mathsf{E}\rangle, \subseteq\rangle \twoheadrightarrow \langle \mathbb{T}^{\mathrm{EvB}}\langle \mathsf{S}, \mathsf{E}\rangle, \subseteq\rangle \tag{16}$$

3.1 Abstraction of a Trace Semantics to a Transition System. Given a trace semantics $\mathsf{T} \in \mathbb{T}^{\mathrm{EvB}}\langle \mathsf{S}, \mathsf{E}\rangle \subseteq \mathbb{T}^{\mathrm{F}}\langle \mathsf{S}, \mathsf{E}\rangle$ of a system in Event-B, the first abstraction is a version $\langle \mathsf{inv}, \mathsf{evts}, \mathsf{init}, \tau\rangle$ of the classic abstraction of a set of traces into a discrete transition systems [22, def. 2.3:1], [14, Sect. 6] where inv is the set of reachable states, evts is the set of reachable events encountered on traces of the semantics, $\mathsf{init} \subseteq \mathsf{inv}$ is a set of initial states (abstracting the initial event from a universal initial state as used in Event-B), and $\tau(\mathsf{e})$ is the transition relation between a state and its immediate successor after occurrence of an event e.

$$\alpha_t\langle \mathsf{S},\ \mathsf{E}\rangle(\mathsf{T}) \triangleq \langle \mathsf{inv},\ \mathsf{evts},\ \mathsf{init},\ \tau\rangle \in \mathbb{T}\mathbb{r}\langle \mathsf{S},\ \mathsf{E}\rangle \qquad \text{where} \tag{17}$$

$$\mathsf{inv} \triangleq \{\mathsf{s} \mid \exists \pi \in \mathsf{T}\ .\ \exists i \in \mathbb{N}\ .\ \exists \mathsf{e} \in \mathsf{E}\ .\ \pi_i = \langle \mathsf{s},\ \mathsf{e}\rangle\} \in \wp(\mathsf{S}) \smallsetminus \{\varnothing\} \tag{a}$$

$$\mathsf{init} \triangleq \{\mathsf{s} \mid \exists \pi \in \mathsf{T}\ .\ \exists \mathsf{e} \in \mathsf{E}\ .\ \pi_0 = \langle \mathsf{s},\ \mathsf{e}\rangle\} \subseteq \mathsf{inv} \tag{b}$$

$$\mathsf{evts} \triangleq \{\mathsf{e} \mid \exists \pi \in \mathsf{T}\ .\ \exists i \in \mathbb{N}\ .\ \exists \mathsf{s} \in \mathsf{S}\ .\ \pi_i = \langle \mathsf{s},\ \mathsf{e}\rangle\} \in \wp(\mathsf{E}) \smallsetminus \{\varnothing\} \tag{c}$$

$$\tau \triangleq \boldsymbol{\lambda}\, \mathsf{e} \bullet \{\langle \mathsf{s},\ \mathsf{s}'\rangle \mid \exists \pi \in \mathsf{T}\ .\ \exists i \in \mathbb{N}\ .\ \exists \mathsf{e}' \in \mathsf{E}\ .\ \pi_i = \langle \mathsf{s},\ \mathsf{e}\rangle \wedge \pi_{i+1} = \langle \mathsf{s}',\ \mathsf{e}'\rangle\} \in \mathsf{evts} \to \wp(\mathsf{inv} \times \mathsf{inv}) \tag{d}$$

Observe that $\mathsf{T} \in \mathbb{T}^{\mathrm{FB}}\langle \mathsf{S},\ \mathsf{E}\rangle$ implies the feasibility (since traces are infinite) and locally bounded nondeterminism requirements of Event-B (by (7))

$$\mathsf{Feas}(\langle \mathsf{inv},\ \mathsf{evts},\ \tau\rangle) \triangleq \forall \mathsf{s} \in \mathsf{inv}\ .\ \exists \mathsf{s}' \in \mathsf{inv}, \mathsf{e} \in \mathsf{evts}\ .\ \langle \mathsf{s},\ \mathsf{s}'\rangle \in \tau(\mathsf{e}) \tag{18}$$

$$\mathsf{BNd}(\langle \mathsf{inv},\ \mathsf{evts},\ \tau\rangle) \triangleq \forall \mathsf{s} \in \mathsf{inv}, \mathsf{e} \in \mathsf{evts}\ .\ |\{\mathsf{s}' \mid \langle \mathsf{s},\ \mathsf{s}'\rangle \in \tau(\mathsf{e})\}| \in \mathbb{N} \tag{19}$$

When applied to Event-B, Feas in (18) yields the feasibility proof obligation rule [4, Sect. 5.2.4] to ensure that no event can block the execution. Similarly for the nonstandard boundedness requirement, an invariant must be added to the machine and invariant preservation must be proved. For the globally bounded nondeterminism discussed in remark 1, we would have $\forall \mathsf{e} \in \mathsf{evts}\ .\ \exists n \in \mathbb{N}\ .\ \forall \mathsf{s} \in \mathsf{inv}\ .\ |\{\mathsf{s}' \mid \langle \mathsf{s},\ \mathsf{s}'\rangle \in \tau(\mathsf{e})\}| < n$.

This leads us to the definition of bounded feasible transition systems on states S and events E as

$$\mathbb{T}\mathbb{r}^{\mathrm{FB}}\langle \mathsf{S},\ \mathsf{E}\rangle \triangleq \{\langle \mathsf{inv},\ \mathsf{evts},\ \mathsf{init},\ \tau\rangle \in \mathbb{T}\mathbb{r}\langle \mathsf{S},\ \mathsf{E}\rangle(\mathsf{T}) \mid \mathsf{Feas}(\langle \mathsf{inv},\ \mathsf{evts},\ \tau\rangle) \wedge \mathsf{BNd}(\langle \mathsf{inv},\ \mathsf{evts},\ \tau\rangle)\} \tag{20}$$

The concretization, corresponding to [22, def. 2.4:1] and [4, Sect. 14.3.5, "Mathematical representation [of traces]"], is the set of traces generated by the transition system.

$$\gamma_t\langle \mathsf{S},\ \mathsf{E}\rangle(\langle \mathsf{inv},\ \mathsf{evts},\ \mathsf{init},\ \tau\rangle) \triangleq \{\pi \in \mathbb{t}^{\infty}\langle \mathsf{S},\ \mathsf{E}\rangle \mid \pi_0 \in \mathsf{init} \times \mathsf{evts} \wedge \forall i \in \mathbb{N}\ .\ \pi_i \in \mathsf{inv} \times \mathsf{evts} \wedge \forall \langle \mathsf{s},\ \mathsf{e}\rangle = \pi_i, \langle \mathsf{s}',\ \mathsf{e}'\rangle = \pi_{i+1}\ .\ \langle \mathsf{s},\ \mathsf{s}'\rangle \in \tau(\mathsf{e})\} \tag{21}$$

By $\mathsf{init} \subseteq \mathsf{inv}$ and the sound and complete induction principle for proving invariance [18, induction principle (i), p. 100], the invariance condition

$$\forall \pi \in \gamma_t\langle \mathsf{S},\ \mathsf{E}\rangle(\langle \mathsf{inv},\ \mathsf{evts},\ \mathsf{init},\ \tau\rangle)\ .\ \forall i \in \mathbb{N}\ .\ \pi_i \in \mathsf{inv} \times \mathsf{evts} \tag{22}$$

of (21) can be expressed as

$$\gamma_t\langle \mathsf{S},\ \mathsf{E}\rangle(\langle \mathsf{inv},\ \mathsf{evts},\ \mathsf{init},\ \tau\rangle) = \{\pi \in \mathbb{t}^{\infty}\langle \mathsf{S},\ \mathsf{E}\rangle \mid \pi_0 \in \mathsf{init} \times \mathsf{evts} \wedge \forall i \in \mathbb{N}, \langle \mathsf{s},\ \mathsf{e}\rangle \in \mathsf{inv} \times \mathsf{evts}, \langle \mathsf{s}',\ \mathsf{e}'\rangle \in \mathsf{S} \times \mathsf{E}\ .\ (\pi_i = \langle \mathsf{s},\ \mathsf{e}\rangle \wedge \pi_{i+1} = \langle \mathsf{s}',\ \mathsf{e}'\rangle) \Rightarrow (\langle \mathsf{s},\ \mathsf{s}'\rangle \in \tau(\mathsf{e}) \wedge \langle \mathsf{s}',\ \mathsf{e}'\rangle \in \mathsf{inv} \times \mathsf{evts})\} \tag{23}$$

This requirement in (22) gives raise, when expressed as (23), to the invariant preservation proof obligation rule [4, Sect. 5.2.2] for a transition system.

This is a Galois connection mapping sets of traces to transition systems (where transition systems are ordered by componentwise inclusion $\dot{\subseteq}$).

$$\langle \mathbb{T}^{\mathrm{FB}}\langle \mathsf{S},\, \mathsf{E}\rangle,\, \subseteq\rangle \underset{\alpha_t\langle \mathsf{S},\, \mathsf{E}\rangle}{\overset{\gamma_t\langle \mathsf{S},\, \mathsf{E}\rangle}{\leftrightarrows}} \langle \mathbb{T}\mathbb{r}^{\mathrm{FB}}\langle \mathsf{S},\, \mathsf{E}\rangle,\, \dot{\subseteq}\rangle \tag{24}$$

Remark 2. For simplicity, our abstraction of a set of traces is to one machine (transition system) only. But we can partition the set of events and abstract traces to a machine for each block of the partition. Then, the machines work in parallel under the sequential consistency hypothesis [27] so that the disjunction of the transitions relations for each machine yield the interleaved semantics. □

3.2 Abstraction of a Trace Semantics to a Variant. Unfortunately, in the event convergence case $\mathbb{T}^{\mathrm{EvB}}\langle \mathsf{S},\, \mathsf{E}\rangle = \mathbb{T}^{\mathrm{FBC}}\langle \mathsf{S},\, \mathsf{E}\rangle$ in (15), the concretization $\gamma_t\langle \mathsf{S}, \mathsf{E}\rangle(\langle \mathsf{inv}, \mathsf{evts}, \mathsf{init}, \tau\rangle)$ of a transition system $\langle \mathsf{inv}, \mathsf{evts}, \mathsf{init}, \tau\rangle$ may not belong to $\mathbb{T}^{\mathrm{EvB}}\langle \mathsf{S},\, \mathsf{E}\rangle \subsetneq \mathbb{T}^{\infty}\langle \mathsf{S},\, \mathsf{E}\rangle$ since the abstraction takes history independence and bounded nondeterminism into account, but forgets about event convergence.

To recover this missing information, we consider, in addition, the abstraction of event convergent traces into a variant function (ordered pointwise).

$$\langle \mathsf{T} \in \mathbb{T}^{\mathrm{EvB}}\langle \mathsf{S},\, \mathsf{E}\rangle,\, \subseteq\rangle \underset{\alpha_v}{\overset{\gamma_v}{\leftrightarrows}} \langle \mathbb{V}\langle \mathsf{S},\, \mathsf{E},\, \tau\rangle,\, \dot{\leqslant}\rangle \tag{25}$$

where the transition relation τ in (25) is given by

$$\langle \mathsf{inv},\, \mathsf{evts},\, \mathsf{init},\, \tau\rangle = \alpha_t\langle \mathsf{S},\, \mathsf{E}\rangle(\mathsf{T}), \tag{26}$$

the abstract domain is

$$\mathbb{V}\langle \mathsf{S},\, \mathsf{E},\, \tau\rangle \triangleq \{\mathsf{V} \in \mathsf{S} \times \mathsf{E} \to \mathbb{N} \mid \forall \mathsf{e} \in \mathsf{E}\,.\, \forall \langle \mathsf{s},\, \mathsf{s}'\rangle \in \tau(\mathsf{e})\,.\, \mathsf{V}\langle \mathsf{e},\, \mathsf{s}'\rangle < \mathsf{V}\langle \mathsf{e},\, \mathsf{s}\rangle\}, \tag{27}$$

Notice that $\mathsf{V} \in \mathsf{S}\times\mathsf{E} \to \mathbb{N}$ yields the numeric variant proof obligation rule [4, Sect. 5.2.7] (or the finite set variant proof obligation rule [4, Sect. 5.2.8] for variants in finite sets).

Finally, the abstraction is the maximal number of possible consecutive occurrences of a given event after reaching a state.

$$\alpha_v\langle \mathsf{S},\, \mathsf{E}\rangle(\mathsf{T}) \triangleq \boldsymbol{\lambda}\langle \mathsf{s},\, \mathsf{e}\rangle \bullet \max\{j - i \mid j > i \in \mathbb{N} \wedge \exists \pi \in \mathsf{T}\,. \quad \forall k \in [i, j]\,.\, \exists \mathsf{s}_k\,.\, \mathsf{s}_i = \mathsf{s} \wedge \pi_k = \langle \mathsf{s}_k,\, \mathsf{e}\rangle\} \tag{28}$$

$$\gamma_v\langle \mathsf{S},\, \mathsf{E}\rangle(\mathsf{V}) \triangleq \{\pi \in \mathbb{t}^{\infty}\langle \mathsf{S},\, \mathsf{E}\rangle \mid \mathsf{cvgt}\langle \mathsf{S},\, \mathsf{E}\rangle(\pi, \mathsf{V})\} \tag{29}$$

$$\mathsf{cvgt}\langle \mathsf{S},\, \mathsf{E}\rangle(\pi, \mathsf{V}) \triangleq \forall \mathsf{e} \in \mathsf{E}\,.\, \forall j > i \in \mathbb{N}\,. \quad (\forall k \in [i, j]\,.\, \exists \mathsf{s}_k\,.\, \pi_k = \langle \mathsf{s}_k,\, \mathsf{e}\rangle) \Rightarrow (j - i < \mathsf{V}\langle \mathsf{s}_i,\, \mathsf{e}\rangle)$$

where $\mathsf{cvgt}\langle \mathsf{S},\, \mathsf{E}\rangle(\pi, \mathsf{V})$ states that the number of consecutive occurrences of the same event e (possibly with different parameters) on trace π is bounded by the variant V. (Instead of the naturals, Event-B also allows for variant functions in finite sets which we can consider as syntactic and semantic sugar.)

The codomain of α_v is $\mathbb{V}\langle \mathsf{S}, \mathsf{E}, \tau\rangle$, that is, by definition of α_v, the variant $\mathsf{V} = \alpha_v\langle \mathsf{S}, \mathsf{E}\rangle$ strictly decreases after any transition τ of $\langle \mathsf{inv}, \mathsf{evts}, \mathsf{init}, \tau\rangle = \alpha_t\langle \mathsf{S}, \mathsf{E}\rangle(\mathsf{T})$ on the trace.

$$\forall \mathsf{e} \in \mathsf{E} \mathbin{.} \forall \langle \mathsf{s}, \mathsf{s}'\rangle \in \tau(\mathsf{e}) \mathbin{.} \mathsf{V}\langle \mathsf{e}, \mathsf{s}'\rangle < \mathsf{V}\langle \mathsf{e}, \mathsf{s}\rangle \tag{30}$$

Notice that this condition (30) on V in (31) yields the variant proof obligation rule [4, Sect. 5.2.9]. The proof that the codomain of γ_v is $\mathbb{T}^{\mathrm{EvB}}\langle \mathsf{S}, \mathsf{E}\rangle$ directly follows from the well-foundedness of $\langle \mathbb{N}, \leqslant\rangle$. Finally, the pair $\langle \alpha_v, \gamma_v\rangle$ is the Galois connection (25).

3.3 Abstraction of a Trace Semantics to a Abstract Machine (Event convergent transition system). Since we require event convergence $\mathbb{T}^{\mathrm{EvB}}\langle \mathsf{S}, \mathsf{E}\rangle = \mathbb{T}^{\mathrm{FBC}}\langle \mathsf{S}, \mathsf{E}\rangle$, an abstract machine is an event convergent transition system.

$$\mathbb{T}\mathbb{r}^{\mathrm{EvB}}\langle \mathsf{S}, \mathsf{E}\rangle \triangleq \{\langle \mathsf{inv}, \mathsf{evts}, \mathsf{init}, \tau, \mathsf{V}\rangle \mid \langle \mathsf{inv}, \mathsf{evts}, \mathsf{init}, \tau\rangle \in \mathbb{T}\mathbb{r}^{\mathrm{FB}}\langle \mathsf{S}, \mathsf{E}\rangle \wedge \mathsf{V} \in \mathbb{V}\langle \mathsf{S}, \mathsf{E}, \tau\rangle\} \tag{31}$$

where event convergence is guaranteed by the variant V.

Let us calculate the abstraction of a trace-based definition semantics $\mathsf{T} \in \mathbb{T}^{\mathrm{EvB}}\langle \mathsf{S}, \mathsf{E}\rangle$ into an abstract machine (we define $f \dot{\times} g(x) \triangleq f(x) \times g(x)$ and $\langle x_1, \ldots, x_n\rangle \times \langle y_1, \ldots, y_m\rangle \triangleq \langle x_1, \ldots, x_n, y_1, \ldots, y_m\rangle$).

$$\begin{aligned}
&\alpha_t \dot{\times} \alpha_v\langle \mathsf{S}, \mathsf{E}\rangle(\mathsf{T})\\
&= \langle \mathsf{inv}, \mathsf{init}, \mathsf{evts}, \tau, \mathsf{V}\rangle && \wr\text{def. (17) of } \alpha_t\text{, (28) of } \alpha_v\text{. and product}\int\\
&= \langle \{\mathsf{s} \mid \exists \pi \in \mathsf{T} \mathbin{.} \exists i \in \mathbb{N} \mathbin{.} \exists \mathsf{e} \in \mathsf{E} \mathbin{.} \pi_i = \langle \mathsf{s}, \mathsf{e}\rangle\}, \{\mathsf{s} \mid \exists \pi \in \mathsf{T} \mathbin{.} \exists \mathsf{e} \in \mathsf{E} \mathbin{.} \pi_0 = \langle \mathsf{s}, \mathsf{e}\rangle\}, \{\mathsf{e} \mid \exists \pi \in \mathsf{T} \mathbin{.} \exists i \in \mathbb{N} \mathbin{.} \exists \mathsf{s} \in \mathsf{S} \mathbin{.} \pi_i = \langle \mathsf{s}, \mathsf{e}\rangle\}, \boldsymbol{\lambda}\, \mathsf{e} \bullet \{\langle \mathsf{s}, \mathsf{s}'\rangle \mid \exists \pi \in \mathsf{T} \mathbin{.} \exists i \in \mathbb{N} \mathbin{.} \exists \mathsf{e}' \in \mathsf{E} \mathbin{.} \pi_i = \langle \mathsf{s}, \mathsf{e}\rangle \wedge \pi_{i+1} = \langle \mathsf{s}', \mathsf{e}'\rangle\}, \boldsymbol{\lambda}\, \langle \mathsf{s}, \mathsf{e}\rangle \bullet \max\{j - i \mid j > i \in \mathbb{N} \wedge \exists \pi \in \mathsf{T} \mathbin{.} \forall k \in [i, j] \mathbin{.} \exists \mathsf{s}_k \mathbin{.} \mathsf{s}_i = \mathsf{s} \wedge \pi_k = \langle \mathsf{s}_k, \mathsf{e}\rangle\}\rangle && \wr(17.\mathrm{a})\text{–}(17.\mathrm{d}) \text{ and } (28)\int
\end{aligned} \tag{32}$$

Let us also calculate the later needed trace-based definition of the semantics of an abstract machine $\langle \mathsf{inv}, \mathsf{evts}, \mathsf{init}, \tau, \mathsf{V}\rangle$ on $\langle \mathsf{S}, \mathsf{E}\rangle$, as follows.

$$\begin{aligned}
&\gamma_t \times \gamma_v\langle \mathsf{S}, \mathsf{E}\rangle(\langle \mathsf{inv}, \mathsf{evts}, \mathsf{init}, \tau, \mathsf{V}\rangle)\\
&= \gamma_t\langle \mathsf{S}, \mathsf{E}\rangle\langle \mathsf{inv}, \mathsf{evts}, \mathsf{init}, \tau\rangle \cap \gamma_v\langle \mathsf{S}, \mathsf{E}\rangle(\mathsf{V}) && \wr\text{def. product } \times\int\\
&= \{\pi \in \mathbb{L}^\infty\langle \mathsf{S}, \mathsf{E}\rangle \mid \pi_0 \in \mathsf{init} \times \mathsf{evts} \wedge \forall i \in \mathbb{N} \mathbin{.} \pi_i \in \mathsf{inv} \times \mathsf{evts} \wedge \forall \langle \mathsf{s}, \mathsf{e}\rangle = \pi_i, \langle \mathsf{s}', \mathsf{e}'\rangle = \pi_{i+1} \mathbin{.} \langle \mathsf{s}, \mathsf{s}'\rangle \in \tau(\mathsf{e})\} \cap \{\pi \in \mathbb{L}^\infty\langle \mathsf{S}, \mathsf{E}\rangle \mid \mathsf{cvgt}\langle \mathsf{S}, \mathsf{E}\rangle(\pi, \mathsf{V})\} && \wr\text{def. (21) of } \gamma_t \text{ and (29) of } \gamma_v\int\\
&= \{\pi \in \mathbb{L}^\infty\langle \mathsf{S}, \mathsf{E}\rangle \mid \pi_0 \in \mathsf{init} \times \mathsf{evts} \wedge \forall i \in \mathbb{N} \mathbin{.} \pi_i \in \mathsf{inv} \times \mathsf{evts} \wedge \forall \langle \mathsf{s}, \mathsf{e}\rangle = \pi_i, \langle \mathsf{s}', \mathsf{e}'\rangle = \pi_{i+1} \mathbin{.} \langle \mathsf{s}, \mathsf{s}'\rangle \in \tau(\mathsf{e}) \wedge \mathsf{cvgt}\langle \mathsf{S}, \mathsf{E}\rangle(\pi, \mathsf{V})\} && \wr\text{def. } \cap\int
\end{aligned} \tag{33}$$

In words, a trace is infinite, must start with an initial state in init and satisfy the invariant $\mathsf{inv} \times \mathsf{evts}$, successive states on the trace for a given event e must be in the transition relation $\langle \mathsf{s}, \mathsf{s}'\rangle \in \tau(\mathsf{e})$ for that event e, and the number of consecutive occurrences of the same event e is bounded by the variant V.

Thanks to history independence, we now have the Galois isomorphism (where $\langle x_1, \ldots, x_n, y_1, \ldots, y_m\rangle \dot{\subseteq} \times \dot{\leqslant} \langle x'_1, \ldots, x'_n, y'_1, \ldots, y'_m\rangle \triangleq \langle x_1, \ldots, x_n\rangle \dot{\subseteq} \langle x'_1, \ldots,$

$x'_n\rangle \wedge \langle y_1, \ldots, y_m\rangle \mathrel{\dot{\leqslant}} \langle y'_1, \ldots, y'_m\rangle = x_1 \subseteq x'_1 \wedge \ldots \wedge x_n \subseteq x'_n \wedge y_1 \leqslant y'_1 \wedge \ldots \wedge y_m \leqslant y'_m$ componentwise)

$$\langle \mathbb{T}^{\mathrm{EvB}}\langle \mathsf{S},\, \mathsf{E}\rangle,\, \subseteq\rangle \underset{\alpha_t \dot{\times} \alpha_v \langle \mathsf{S},\, \mathsf{E}\rangle}{\overset{\gamma_t \dot{\times} \gamma_v \langle \mathsf{S},\, \mathsf{E}\rangle}{\leftrightarrows}} \langle \mathbb{T}\mathbb{r}^{\mathrm{EvB}}\langle \mathsf{S},\, \mathsf{E}\rangle,\, \dot{\subseteq} \times \dot{\leqslant}\rangle \tag{34}$$

3.4 On the Definition of a Trace Semantics of Abstract Machines in Event-B. The Galois isomorphism (34) proves that the Event-B method of defining a history-independent, event convergent, bounded trace semantics (14) in $\mathbb{T}^{\mathrm{EvB}}\langle \mathsf{S},\, \mathsf{E}\rangle$ by means of a transition system and a variant $\mathbb{T}\mathbb{r}^{\mathrm{EvB}}\langle \mathsf{S},\, \mathsf{E}\rangle$ (31) is sound and complete.

Recall that the definition of machines in Event-B defines proof obligations to ensure that S and E are well defined and the specified transition system $\langle \mathsf{inv}, \mathsf{evts}, \mathsf{init}, \tau\rangle$ and variant V belong to $\mathbb{T}\mathbb{r}^{\mathrm{EvB}}\langle \mathsf{S},\, \mathsf{E}\rangle \times \mathbb{V}\langle \mathsf{S},\, \mathsf{E},\, \tau\rangle$ which ensure invariant initialization and preservation, deadlock freeness, event convergence, feasibility, and bounded nondeterminism of the trace semantics. History independence directly follows from the fact that traces are generated by the transition system, as defined by $\gamma_t\langle \mathsf{S},\, \mathsf{E}\rangle(\langle \mathsf{inv},\, \mathsf{evts},\, \mathsf{init},\, \tau\rangle)$ (formalizing e.g. context extension).

In what follows, we will need a definition of the reflexive transitive closure τ^* of the transition relation τ, as follows ($n \in \mathbb{N}$).

$$\begin{array}{l} \tau^n \triangleq \{\langle \mathsf{s},\, \mathsf{s}'\rangle \mid \exists \mathsf{s}_0, \mathsf{e}_0, \mathsf{s}_1, \mathsf{e}_1, \ldots, \mathsf{s}_n \,.\, \mathsf{s}_0 = \mathsf{s} \wedge \forall k \in [0, n[\,.\, \langle \mathsf{s}_k,\, \mathsf{s}_{k+1}\rangle \in \tau(\mathsf{e}_k) \wedge \mathsf{s}_n = \mathsf{s}'\} \\ \tau^* \triangleq \bigcup\limits_{n \in \mathbb{N}} \tau^n \end{array} \tag{35}$$

We write $\tau[\mathsf{e}]^k$ and $\tau[\mathsf{e}]^*$ for the case when all events e_k, $k \in [0, n[$ are restricted to be e.

4 Refinement of System Trace Semantics

The definition of system semantics by machines is only an aspect of Event-B. The main technique in Event-B is the development by successive refinements.

Whereas abstraction interpretation establishes a connection between different semantics (such as set of traces in operational semantics and local invariants in Hoare logic)[2], refinements establishes a connection between semantics of the same nature (i.e. traces, maybe on different states). This is not a fundamental difference since the concretization can always be used, at least in theory, to reason in the concrete (i.e. set of traces so that the abstraction becomes a closure [17, Sect. 5.2]), although this is not done in practice since set of traces are not directly implementable in a machine. The main difference is that refinement aims at providing the implementation of one trace, whereas abstract interpretation aims at providing the implementation of a set of traces, or a machine implementation of it (such as local invariants in deductive verification or

[2] Since Event-B is only concerned with machines, we only need to consider in this chapter the abstract interpretation of machines by machines (more precisely their trace semantics).

abstract domains in static analysis). Nevertheless, the reasoning in refinement and abstract interpretation are quite similar since in both cases one works by reference to a specification (which can always be represented as a set of traces).

4.1 Simple Trace Refinement. In the simplest case [4, Sect. 14.4.3/4], a refinement $\mathsf{T} \in \mathbb{T}^{\mathrm{FB}}\langle \mathsf{S}, \mathsf{E}\rangle$ of $\overline{\mathsf{T}} \in \mathbb{T}^{\mathrm{FB}}\langle \mathsf{S}, \mathsf{E}\rangle$ in Event-B eliminates some possible executions as specified by $S \in \mathbb{T}^{\mathrm{FB}}\langle \mathsf{S}, \mathsf{E}\rangle$. So T is more implementable than $\overline{\mathsf{T}}$ since some possible executions of $\overline{\mathsf{T}}$ have been eliminated in T. Define $\alpha_S(X) \triangleq X \cap S = X \smallsetminus \neg S$ and $\gamma_S(Y) \triangleq Y \cup \neg S$ so that we have $\mathsf{T} = \alpha_S(\overline{\mathsf{T}})$. This is a classic exclusion abstraction $\langle \mathbb{T}^\infty\langle \mathsf{S}, \mathsf{E}\rangle, \subseteq\rangle \underset{\alpha_S}{\overset{\gamma_S}{\leftrightarrows}} \langle \mathbb{T}^\infty\langle \mathsf{S}, \mathsf{E}\rangle, \subseteq\rangle$ [15, Ex. 11.5]. It follows that the refined system semantics is a subset of the abstract one

$$\varnothing \neq \mathsf{T} \subseteq \overline{\mathsf{T}}. \tag{36}$$

(i.e. "Any trace of a refined model must also be a trace of the abstraction" [4, Sect. 14.4.3]).

Traces can be eliminated on criteria depending on the history of computations so that history independence is not preserved (while boundedness and event convergence are). An example is the elimination of a_2bc_1 in $\{a_1bc_1, a_2bc_2, a_2bc_1, a_1bc_2\}$. The additional requirement $\mathsf{Ffus} \circ \alpha_S = \alpha_S$ is necessary to ensure that $\langle \mathbb{T}^{\mathrm{FB}}\langle \mathsf{S}, \mathsf{E}\rangle, \subseteq\rangle \underset{\alpha_S}{\overset{\gamma_S}{\leftrightarrows}} \langle \mathbb{T}^{\mathrm{FB}}\langle \mathsf{S}, \mathsf{E}\rangle, \subseteq\rangle$.

When T and $\overline{\mathsf{T}}$ are respectively defined by machines $\mathsf{t} = \langle \mathsf{inv}, \mathsf{evts}, \mathsf{init}, \tau\rangle$ and $\overline{\mathsf{t}} = \langle \overline{\mathsf{inv}}, \overline{\mathsf{evts}}, \overline{\mathsf{init}}, \overline{\tau}\rangle$, that is $\mathsf{T} = \gamma_t\langle \mathsf{S}, \mathsf{E}\rangle(\mathsf{t})$ and $\overline{\mathsf{T}} = \gamma_t\langle \mathsf{S}, \mathsf{E}\rangle(\overline{\mathsf{t}})$, simple trace refinement can be expressed in term of these machines as

$$\dot{\varnothing} \neq \mathsf{t} \mathrel{\dot{\subseteq}} \overline{\mathsf{t}} \tag{37}$$

stating that "a refined trace is also an abstract trace". Because we consider only feasible transition systems (18), the requirement "we do not want a refined trace to be unable to be extended if the corresponding abstract one is able to be extended" [4, *(I)*, p. 428] and [4, *(II)*, p. 429] is always satisfied.

Notice that the traces generates by t and $\overline{\mathsf{t}}$ are history independent so that the abstraction preserves history independence. For example eliminating a_2bc_1 in $\{a_1bc_1, a_2bc_2, a_2bc_1, a_1bc_2\}$ would require the elimination of the transition $\langle b, c_1\rangle$ hence also of the traces a_2bc_1 leaving $\{a_2bc_2, a_1bc_2\}$ only, which is history independent. This is not a practical limitation of Event-B since it is always possible to refine the transition system where states are traces with a position in the trace. In our example, the states would be pairs $\langle xby, n\rangle$ and $x \in \{a_1, a_2\}$, $y \in \{c_1, c_2\}$, and $n \in [0, 3]$, initial states $\langle xby, 0\rangle$, and transitions $\langle\langle xby, n\rangle, \langle xby, n{+}1\rangle\rangle$, $n \in [1, 3[$. Then eliminating transitions $\langle\langle a_2bc_1, n\rangle, \langle a_2bc_1, n{+}1\rangle\rangle$, $n \in [0, 3]$ produces the desired result. In Event-B it is possible to use auxiliary variables to cumulate the history of computation and obtain a similar effect.

4.2 Duality Between Refinement and Abstract Interpretation. The previous refinement α_S eliminates undesirable execution in $\neg S$. In abstract interpretation, the abstraction α_S ignores all information about the eliminated executions in $\neg S$. The abstract semantics is therefore a subset of the concrete

one. In both cases, the refinement/abstraction knows nothing about the behaviors that have been eliminated. This yields the following confusing vocabulary clash

	$\mathsf{T} = \alpha_S(\overline{\mathsf{T}})$	$\overline{\mathsf{T}} = \gamma_S(\mathsf{T})$
Event-B	T refines $\overline{\mathsf{T}}$	$\overline{\mathsf{T}}$ abstracts T
Abstract interpretation	T abstracts $\overline{\mathsf{T}}$	$\overline{\mathsf{T}}$ concretizes T

(38)

Notice that this duality also appears in the definition of the word "abstraction" in dictionaries as "the process of considering something independently of its associations, attributes, or concrete accompaniments" as in abstract interpretation or "the process of removing something" as in refinement [3].

Finally, the order dual $\langle C, \sqsupseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A, \succeq\rangle$ of a Galois connection $\langle C, \sqsubseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A, \preceq\rangle$ is a Galois connection $\langle A, \preceq\rangle \underset{\gamma}{\overset{\alpha}{\leftrightarrows}} \langle C, \sqsubseteq\rangle$ where α and γ are exchanged [15, Lem. 11.25] showing the ambiguity of the formulations "abstraction", "refinement", and "concretization" which heavily depends on the considered ordering and are exchanged for dual orderings.

4.3 Homomorphic Trace Refinement. The requirement in simple trace refinement that concrete and abstract states and events be the same can be lifted by considering a homomorphic abstraction [15, Ex. 11.6] defined by $h \in \mathsf{S} \times \mathsf{E} \to \overline{\mathsf{S}} \times \overline{\mathsf{E}}$ as

$$\alpha_h(\mathsf{T}) \triangleq \{h(\pi) \mid \pi \in \mathsf{T}\} \quad \text{where} \quad h(\pi) \triangleq \boldsymbol{\lambda}\, i \in \mathbb{N} \bullet h(\pi_i) \tag{39}$$

so that $\langle \mathbb{T}^{\mathrm{EvB}}\langle \mathsf{S}, \mathsf{E}\rangle, \subseteq\rangle \underset{\alpha_h}{\overset{\gamma_h}{\leftrightarrows}} \langle \mathbb{T}^{\mathrm{EvB}}\langle \overline{\mathsf{S}}, \overline{\mathsf{E}}\rangle, \subseteq\rangle$. Homomorphic trace refinement is then $\varnothing \neq \alpha_h(\mathsf{T}) \subseteq \overline{\mathsf{T}}$ which can be easily translated to machines. An example is [4, Sect. 2.5.2] where a variable in the state is replaced by 3 variables.

4.4 External Trace Refinement. In case trace transitions are not directly observable, we let Es and $\overline{\mathsf{Es}}$ be (external observation) sets, $f \in \mathsf{S} \times \mathsf{E} \to \mathsf{Es}$ and $\overline{f} \in \overline{\mathsf{S}} \times \overline{\mathsf{E}} \to \overline{\mathsf{Es}}$ be observation functions, and define the homomorphic abstractions $\alpha_f(\mathsf{T}) \triangleq \{f(\pi) \mid \pi \in \mathsf{T}\}$ where $f(\pi) \triangleq \boldsymbol{\lambda}\, i \in \mathbb{N} \bullet f(\pi_i)$ and $\alpha_{\overline{f}}(\overline{\mathsf{T}}) \triangleq \{\overline{f}(\overline{\pi}) \mid \overline{\pi} \in \overline{\mathsf{T}}\}$ where $\overline{f}(\overline{\pi}) \triangleq \boldsymbol{\lambda}\, i \in \mathbb{N} \bullet \overline{f}(\overline{\pi}_i)$. Given $h \in \mathsf{Es} \to \overline{\mathsf{Es}}$, external trace refinement is defined as $\varnothing \neq \alpha_h(\alpha_f(\mathsf{T})) \subseteq \alpha_{\overline{f}}(\overline{\mathsf{T}})$. This is extended to machines to get [4, *(IV)*, p. 433].

4.5 Total Simulation Refinement. Let $\rho \in (\mathsf{S} \times \mathsf{E}) \to \wp(\overline{\mathsf{S}} \times \overline{\mathsf{E}})$ be a total binary relation from concrete to abstract transition states. Define the abstraction

$$\alpha_\rho(\mathsf{T}) \triangleq \{\boldsymbol{\lambda}\, i \in \mathbb{N} \bullet \overline{\pi}_i \mid \pi \in \mathsf{T} \wedge \forall i \in \mathbb{N} \,.\, \overline{\pi}_i \in \rho(\pi_i)\} \tag{40}$$

(which is (39) when $\rho = h$) such that $\langle \mathbb{T}^{\mathrm{EvB}}\langle \mathsf{S}, \mathsf{E}\rangle, \subseteq\rangle \underset{\alpha_\rho}{\overset{\gamma_\rho}{\leftrightarrows}} \langle \mathbb{T}^{\mathrm{EvB}}\langle \overline{\mathsf{S}}, \overline{\mathsf{E}}\rangle, \subseteq\rangle$. We can define total simulation refinement as $\varnothing \neq \alpha_\rho(\mathsf{T}) \subseteq \overline{\mathsf{T}}$.

[3] Apple Dictionary, Version 2.3.0 (294).

4.6 Forward Simulation Refinement. Given $f \in \mathsf{S}\times\mathsf{E} \to \mathsf{Es}$, $\overline{f} \in \overline{\mathsf{S}}\times\overline{\mathsf{E}} \to \overline{\mathsf{Es}}$, and $h \in \mathsf{S}\times\mathsf{E} \to \overline{\mathsf{S}}\times\overline{\mathsf{E}}$, conditions [4, (C1), p. 434], [4, (C2) p. 435], and [4, (C3), p. 435] imply that the forward simulation refinement is an external trace refinement. This translates to machines to get the rules of [4, Sect. 15.5.3].

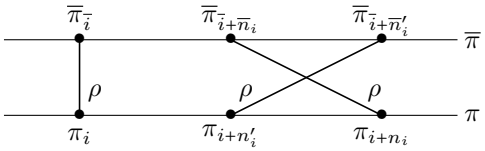
4.7 Backward Simulation Refinement. Considering [4, (C2') p. 439], we get backward simulation refinement,

4.8 Trace Refinement [4, Sect. 14.6]. Assume that $\pi \in \mathbb{t}^\infty\langle \mathsf{S},\, \mathsf{E}\rangle$, $\overline{\pi} \in \mathbb{t}^\infty\langle \overline{\mathsf{S}},\, \overline{\mathsf{E}}\rangle$, and $\rho \in \wp((\mathsf{S}\times\mathsf{E})\times(\overline{\mathsf{S}}\times\overline{\mathsf{E}}))$ is a (so-called simulation) relation (which may be restricted to be functional). ρ defines a relation between concrete and abstract states and events (as specifying in Event-B by *gluing invariants* for states [4, p. 181] and *witnesses* for variables and parameters of corresponding events [4, p. 183]).

A trace π is a *refinement* of $\overline{\pi}$ by ρ, denoted $R_\rho(\pi,\overline{\pi})$, if and only if

$$\begin{array}{l} R_\rho(\pi,\overline{\pi}) \triangleq \exists \eta > 0, i_0, \overline{i}_0 \in \mathbb{N} \,.\, \langle \pi_{i_0},\, \overline{\pi}_{\overline{i}_0}\rangle \in \rho \wedge \\ \forall i, \overline{i} \in \mathbb{N} \,.\, (\langle \pi_i,\, \overline{\pi}_{\overline{i}}\rangle \in \rho) \Rightarrow (\exists n_i, \overline{n}_{\overline{i}} \in [0,\eta] \,.\, n_i + \overline{n}_{\overline{i}} > 0 \wedge \langle \pi_{i+n_i},\, \overline{\pi}_{\overline{i}+\overline{n}_{\overline{i}}}\rangle \in \rho) \end{array} \tag{41}$$

Ultimately, some initial transitions π_{i_0} and $\overline{\pi}_{\overline{i}_0}$ must be related by ρ (one can also require the stronger $i_0 = \overline{i}_0 = 0$). Later on, if two transitions π_i and $\overline{\pi}_{\overline{i}}$ are related by ρ then farther concrete and abstract transition π_{i+n_i} and $\overline{\pi}_{\overline{i}+\overline{n}_{\overline{i}}}$ must be related by ρ, and this for ever. If $\overline{n}_{\overline{i}} = 0$ they are split [4, Sect. 14.6.1] and if $n_i = 0$ then consecutive abstract events are merged ([4, Sect. 14.6.2] is for alternative events merging also allowed by (41)). Notice that n_i and $\overline{n}_{\overline{i}}$ need not be unique as in



To avoid refinements taking larger and larger number of computation steps, they are bounded by η (depending on π and $\overline{\pi}$). It follows that an abstract computation step can be refined by bounded finitely many concrete computation steps or only bounded finitely many consecutive abstract transitions may disappear.

4.9 Trace Semantics Refinement. Let us define the refinement abstraction $\alpha_\rho(\mathsf{T})$ of $\mathsf{T} \in \mathbb{T}^\infty\langle \mathsf{S},\, \mathsf{E}\rangle$ as

$$\alpha_\rho(\mathsf{T}) \triangleq \mathsf{post}[R_\rho]\mathsf{T} = \{\overline{\pi} \in \mathbb{t}^\infty\langle \overline{\mathsf{S}},\, \overline{\mathsf{E}}\rangle \mid \exists \pi \in \mathsf{T} \,.\, R_\rho(\pi,\overline{\pi})\} \tag{42}$$

This is a Galois connection [15, (12.22)]

$$\langle \mathbb{T}^\infty\langle \mathsf{S},\, \mathsf{E}\rangle, \subseteq\rangle \underset{\alpha_\rho}{\overset{\gamma_\rho}{\leftrightarrows}} \langle \mathbb{T}^\infty\langle \overline{\mathsf{S}},\, \overline{\mathsf{E}}\rangle, \subseteq\rangle \tag{43}$$

where

$$\gamma_\rho(\overline{\mathsf{T}}) \triangleq \widetilde{\mathsf{pre}}[R_\rho]\overline{\mathsf{T}} = \{\pi \in \mathbb{L}^\infty\langle \mathsf{S}, \mathsf{E}\rangle \mid \forall \overline{\pi} \,.\, R_\rho(\pi, \overline{\pi}) \Rightarrow (\overline{\pi} \in \overline{\mathsf{T}})\} \tag{44}$$

We say that $\overline{\mathsf{T}} \in \mathbb{T}^\infty\langle \overline{\mathsf{S}}, \overline{\mathsf{E}}\rangle$ is an abstraction of $\mathsf{T} \in \mathbb{T}^\infty\langle \mathsf{S}, \mathsf{E}\rangle)$ (or T is a refinement of $\overline{\mathsf{T}}$) if and only if $R_\rho(\mathsf{T}, \overline{\mathsf{T}})$ holds, where

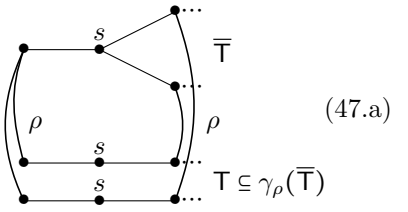
$$\overline{R}_\rho(\mathsf{T}, \overline{\mathsf{T}}) \triangleq \alpha_\rho(\mathsf{T}) \subseteq \overline{\mathsf{T}} \Leftrightarrow \mathsf{T} \subseteq \gamma_\rho(\overline{\mathsf{T}}) \tag{45}$$

which is equivalent to

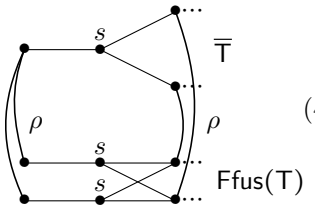
$$\begin{array}{l} \forall \pi \in \mathsf{T} \,.\, \exists \overline{\pi} \in \overline{\mathsf{T}} \,.\, \exists \eta > 0, i_0, \overline{i}_0 \in \mathbb{N} \,.\, \langle \pi_{i_0},\, \overline{\pi}_{\overline{i}_0}\rangle \in \rho \wedge \\ \qquad \forall i, \overline{i} \in \mathbb{N} \,.\, (\langle \pi_i,\, \overline{\pi}_{\overline{i}}\rangle \in \rho) \Rightarrow (\exists n_i, \overline{n}_{\overline{i}} \in [1, \eta] \,.\, \langle \pi_{i+n_i},\, \overline{\pi}_{\overline{i}+\overline{n}_{\overline{i}}}\rangle \in \rho) \end{array} \tag{46}$$

so that no concrete trace can be unrelated to an abstract trace (although the opposite is possible since some abstract behaviors may disappear in the concrete).

4.10 History Independence Is Not Necessarily Preserved by Trace Refinement. Let $\overline{\mathsf{T}} \in \mathbb{T}^{\mathrm{EvB}}\langle \overline{\mathsf{S}}, \overline{\mathsf{E}}\rangle$ be an abstract Event-B semantics (hence history independent). Let T be a refinement of $\overline{\mathsf{T}}$ so that $\mathsf{T} \subseteq \gamma_\rho(\overline{\mathsf{T}})$ by (45). The following counter-example (47.a) shows that T may not be a concrete Event-B semantics (since it is not history independent, that is $\mathsf{Ffus}(\gamma_\rho(\overline{\mathsf{T}})) \neq \gamma_\rho(\overline{\mathsf{T}})$ although $\mathsf{Ffus}(\overline{\mathsf{T}}) = \overline{\mathsf{T}}$ as seen in (47.b)).



(47.a)



(47.b)

Therefore, I define $\mathsf{T} \in \mathbb{L}^\infty\langle \mathsf{S}, \mathsf{E}\rangle)$ to be a refinement of $\overline{\mathsf{T}} \in \mathbb{T}^{\mathrm{EvB}}\langle \overline{\mathsf{S}}, \overline{\mathsf{E}}\rangle$ for $\rho \in \wp((\mathsf{S} \times \mathsf{E}) \times (\overline{\mathsf{S}} \times \overline{\mathsf{E}}))$ if and only if $\mathsf{T} \in \mathbb{T}^{\mathrm{F}}\langle \mathsf{S}, \mathsf{E}\rangle \wedge R_\rho(\mathsf{T}, \overline{\mathsf{T}})$. This condition is naturally satisfied in [4, Sect. 14.6] since Jean-Raymond defines refinement for transition systems whereas I infer it by abstraction of trace refinement.

5 Refinement of Machines

Given an abstract event convergent transition system $\overline{\mathsf{t}} = \langle \overline{\mathsf{inv}}, \overline{\mathsf{evts}}, \overline{\mathsf{init}}, \overline{\tau}, \overline{\mathsf{V}}\rangle \in \mathbb{TR}^{\mathrm{EvB}}\langle \overline{\mathsf{S}}, \overline{\mathsf{E}}\rangle$ generating the abstract semantics $\overline{\mathsf{T}} = \gamma_t \times \gamma_v(\overline{\mathsf{t}})$ and a concrete event convergent transition system $\mathsf{t} = \langle \mathsf{inv}, \mathsf{evts}, \mathsf{init}, \tau, \mathsf{V}\rangle \in \mathbb{TR}^{\mathrm{EvB}}\langle \mathsf{S}, \mathsf{E}\rangle$ generating the concrete semantics $\mathsf{T} = \gamma_t \times \gamma_v(\mathsf{t})$, we look for a sufficient (and preferably necessary) condition on $\overline{\mathsf{t}}$ and t ensuring that $\overline{R}_\rho(\mathsf{T}, \overline{\mathsf{T}})$. We proceed by calculational design. We must have

$$\begin{array}{ll} & \overline{R}_\rho(\mathsf{T}, \overline{\mathsf{T}}) \\ \Leftrightarrow & \overline{R}_\rho(\gamma_t \times \gamma_v(\mathsf{t}), \gamma_t \times \gamma_v(\overline{\mathsf{t}})) \qquad \text{⟅def. } \mathsf{T} \text{ and } \overline{\mathsf{T}}\text{⟆} \end{array}$$

$\Leftrightarrow \overline{R}_\rho(\gamma_t \times \gamma_v\langle \mathsf{inv},\ \mathsf{evts},\ \mathsf{init},\ \tau,\ \mathsf{V}\rangle, \gamma_t \times \gamma_v\langle \overline{\mathsf{inv}},\ \overline{\mathsf{evts}},\ \overline{\mathsf{init}},\ \overline{\tau},\ \overline{\mathsf{V}}\rangle)$ ⟅def. t and $\overline{\mathsf{t}}$⟆

$\Leftrightarrow \alpha_\rho(\gamma_t \times \gamma_v\langle \mathsf{inv},\ \mathsf{evts},\ \mathsf{init},\ \tau,\ \mathsf{V}\rangle) \subseteq \gamma_t \times \gamma_v\langle \overline{\mathsf{inv}},\ \overline{\mathsf{evts}},\ \overline{\mathsf{init}},\ \overline{\tau},\ \overline{\mathsf{V}}\rangle$ ⟅def. (45) of $\overline{R}_\rho$⟆

$\Leftrightarrow \{\overline{\pi} \mid \exists \pi \in \gamma_t \times \gamma_v\langle \mathsf{inv},\ \mathsf{evts},\ \mathsf{init},\ \tau,\ \mathsf{V}\rangle\ .\ \overline{R}_\rho(\pi,\overline{\pi})\} \subseteq \gamma_t \times \gamma_v\langle \overline{\mathsf{inv}},\ \overline{\mathsf{evts}},\ \overline{\mathsf{init}},\ \overline{\tau},\ \overline{\mathsf{V}}\rangle$
⟅def. (42) of α_ρ⟆

$\Leftrightarrow \forall \overline{\pi}\ .\ \forall \pi \in \gamma_t \times \gamma_v\langle \mathsf{inv},\ \mathsf{evts},\ \mathsf{init},\ \tau,\ \mathsf{V}\rangle\ .\ \overline{R}_\rho(\pi,\overline{\pi}) \Rightarrow (\overline{\pi} \in \gamma_t \times \gamma_v\langle \overline{\mathsf{inv}},\ \overline{\mathsf{evts}},\ \overline{\mathsf{init}},\ \overline{\tau},\ \overline{\mathsf{V}}\rangle)$ ⟅def. ⊆⟆

$\Leftrightarrow \forall \overline{\pi}\ .\ \forall \pi \in \{\pi \in \mathbb{t}^\infty\langle \mathsf{S},\ \mathsf{E}\rangle \mid \pi_0 \in \mathsf{init} \times \mathsf{evts} \wedge \forall i \in \mathbb{N}, \mathsf{s}, \mathsf{s}' \in \mathsf{inv}, \mathsf{e}, \mathsf{e}' \in \mathsf{E}\ .\ (\pi_i = \langle \mathsf{s},\ \mathsf{e}\rangle \wedge \pi_{i+1} = \langle \mathsf{s}',\ \mathsf{e}'\rangle) \Rightarrow (\langle \mathsf{s},\ \mathsf{s}'\rangle \in \tau(\mathsf{e})) \wedge \mathsf{cvgt}\langle \mathsf{S},\ \mathsf{E}\rangle(\pi, \mathsf{V})\}\ .$
$\overline{R}_\rho(\pi,\overline{\pi}) \Rightarrow$
$(\overline{\pi} \in \{\overline{\pi} \in \mathbb{t}^\infty\langle \overline{\mathsf{S}},\ \overline{\mathsf{E}}\rangle \mid \overline{\pi}_0 \in \overline{\mathsf{init}} \times \overline{\mathsf{evts}} \wedge \forall i \in \mathbb{N}, \overline{\mathsf{s}}, \overline{\mathsf{s}}' \in \overline{\mathsf{inv}}, \overline{\mathsf{e}}, \overline{\mathsf{e}}' \in \overline{\mathsf{E}}\ .\ (\overline{\pi}_i = \langle \overline{\mathsf{s}},\ \overline{\mathsf{e}}\rangle \wedge \overline{\pi}_{i+1} = \langle \overline{\mathsf{s}}',\ \overline{\mathsf{e}}'\rangle) \Rightarrow (\langle \overline{\mathsf{s}},\ \overline{\mathsf{s}}'\rangle \in \overline{\tau}(\overline{\mathsf{e}})) \wedge \mathsf{cvgt}\langle \overline{\mathsf{S}},\ \overline{\mathsf{E}}\rangle(\overline{\pi}, \overline{\mathsf{V}})\})$
⟅characterization (33) of $\gamma_t \times \gamma_v$⟆

$\Leftrightarrow \forall \overline{\pi}, \pi\ .\ (\pi_0 \in \mathsf{init} \times \mathsf{evts} \wedge \forall i \in \mathbb{N}, \mathsf{s}, \mathsf{s}' \in \mathsf{inv}, \mathsf{e}, \mathsf{e}' \in \mathsf{E}\ .\ (\pi_i = \langle \mathsf{s},\ \mathsf{e}\rangle \wedge \pi_{i+1} = \langle \mathsf{s}',\ \mathsf{e}'\rangle) \Rightarrow (\langle \mathsf{s},\ \mathsf{s}'\rangle \in \tau(\mathsf{e})) \wedge \mathsf{cvgt}\langle \mathsf{S},\ \mathsf{E}\rangle(\pi, \mathsf{V}) \wedge \overline{R}_\rho(\pi,\overline{\pi})) \Rightarrow$
$(\overline{\pi} \in \mathbb{t}^\infty\langle \overline{\mathsf{S}},\ \overline{\mathsf{E}}\rangle \mid \overline{\pi}_0 \in \overline{\mathsf{init}} \times \overline{\mathsf{evts}} \wedge \forall i \in \mathbb{N}, \overline{\mathsf{s}}, \overline{\mathsf{s}}' \in \overline{\mathsf{inv}}, \overline{\mathsf{e}}, \overline{\mathsf{e}}' \in \overline{\mathsf{E}}\ .\ (\overline{\pi}_i = \langle \overline{\mathsf{s}},\ \overline{\mathsf{e}}\rangle \wedge \overline{\pi}_{i+1} = \langle \overline{\mathsf{s}}',\ \overline{\mathsf{e}}'\rangle) \Rightarrow (\langle \overline{\mathsf{s}},\ \overline{\mathsf{s}}'\rangle \in \overline{\tau}(\overline{\mathsf{e}})) \wedge \mathsf{cvgt}\langle \overline{\mathsf{S}},\ \overline{\mathsf{E}}\rangle(\overline{\pi}, \overline{\mathsf{V}}))$ ⟅def. ∈⟆

$\Leftrightarrow \forall \overline{\pi}, \pi\ .\ (\pi_0 \in \mathsf{init} \times \mathsf{evts} \wedge \forall i \in \mathbb{N}, \mathsf{s}, \mathsf{s}' \in \mathsf{inv}, \mathsf{e}, \mathsf{e}' \in \mathsf{E}\ .\ (\pi_i = \langle \mathsf{s},\ \mathsf{e}\rangle \wedge \pi_{i+1} = \langle \mathsf{s}',\ \mathsf{e}'\rangle) \Rightarrow (\langle \mathsf{s}, \mathsf{s}'\rangle \in \tau(\mathsf{e})) \wedge \mathsf{cvgt}\langle \mathsf{S}, \mathsf{E}\rangle(\pi, \mathsf{V}) \wedge (\exists \eta > 0, i_0, \overline{i}_0 \in \mathbb{N}\ .\ \langle \pi_{i_0}, \overline{\pi}_{\overline{i}_0}\rangle \in \rho \wedge \forall i, \overline{i} \in \mathbb{N}\ .\ (\langle \pi_i,\ \overline{\pi}_{\overline{i}}\rangle \in \rho) \Rightarrow (\exists n_i, \overline{n}_{\overline{i}} \in [0, \eta]\ .\ n_i + \overline{n}_{\overline{i}} > 0 \wedge \langle \pi_{i+n_i},\ \overline{\pi}_{\overline{i}+\overline{n}_{\overline{i}}}\rangle \in \rho))) \Rightarrow$
$(\overline{\pi} \in \mathbb{t}^\infty\langle \overline{\mathsf{S}},\ \overline{\mathsf{E}}\rangle \mid \overline{\pi}_0 \in \overline{\mathsf{init}} \times \overline{\mathsf{evts}} \wedge \forall i \in \mathbb{N}, \overline{\mathsf{s}}, \overline{\mathsf{s}}' \in \overline{\mathsf{inv}}, \overline{\mathsf{e}}, \overline{\mathsf{e}}' \in \overline{\mathsf{E}}\ .\ (\overline{\pi}_i = \langle \overline{\mathsf{s}},\ \overline{\mathsf{e}}\rangle \wedge \overline{\pi}_{i+1} = \langle \overline{\mathsf{s}}',\ \overline{\mathsf{e}}'\rangle) \Rightarrow (\langle \overline{\mathsf{s}},\ \overline{\mathsf{s}}'\rangle \in \overline{\tau}(\overline{\mathsf{e}})) \wedge \mathsf{cvgt}\langle \overline{\mathsf{S}},\ \overline{\mathsf{E}}\rangle(\overline{\pi}, \overline{\mathsf{V}}))$ ⟅def. (41) of $\overline{R}_\rho(\pi,\overline{\pi})$⟆

$\Leftrightarrow \forall \pi, \overline{\pi}\ .\ \big((\pi_0 \in \mathsf{init} \times \mathsf{evts} \wedge \exists i_0 \in \mathbb{N}\ .\ \forall i \in [0, i_0[, \mathsf{s}, \mathsf{s}' \in \mathsf{inv}, \mathsf{e}, \mathsf{e}' \in \mathsf{E}\ .\ (\pi_i = \langle \mathsf{s},\ \mathsf{e}\rangle \wedge \pi_{i+1} = \langle \mathsf{s}',\ \mathsf{e}'\rangle) \Rightarrow (\langle \mathsf{s},\ \mathsf{s}'\rangle \in \tau(\mathsf{e})) \wedge \langle \pi_{i_0}, \overline{\pi}_0\rangle \in \rho) \Rightarrow (\overline{\pi}_0 \in \overline{\mathsf{init}})\big) \wedge$ (47)

$\big((\forall i \in \mathbb{N}, \mathsf{s}, \mathsf{s}' \in \mathsf{inv}, \mathsf{e}, \mathsf{e}' \in \mathsf{E}\ .\ (\pi_i = \langle \mathsf{s},\ \mathsf{e}\rangle \wedge \pi_{i+1} = \langle \mathsf{s}',\ \mathsf{e}'\rangle) \Rightarrow (\langle \mathsf{s},\ \mathsf{s}'\rangle \in \tau(\mathsf{e})) \wedge \mathsf{cvgt}\langle \mathsf{S}, \mathsf{E}\rangle(\pi, \mathsf{V}) \wedge \exists \eta > 0\ .\ \forall i, \overline{i} \in \mathbb{N}\ .\ (\langle \pi_i,\ \overline{\pi}_{\overline{i}}\rangle \in \rho) \Rightarrow (\exists j \in\ ]i, i+\eta]\ .\ \langle \pi_j,\ \overline{\pi}_{\overline{i}+1}\rangle \in \rho)) \Rightarrow$
$(\forall i \in \mathbb{N}, \overline{\mathsf{s}}, \overline{\mathsf{s}}' \in \overline{\mathsf{inv}}, \overline{\mathsf{e}}, \overline{\mathsf{e}}' \in \overline{\mathsf{E}}\ .\ (\overline{\pi}_i = \langle \overline{\mathsf{s}},\ \overline{\mathsf{e}}\rangle \wedge \overline{\pi}_{i+1} = \langle \overline{\mathsf{s}}',\ \overline{\mathsf{e}}'\rangle) \Rightarrow (\langle \overline{\mathsf{s}},\ \overline{\mathsf{s}}'\rangle \in \overline{\tau}(\overline{\mathsf{e}})) \wedge \mathsf{cvgt}\langle \overline{\mathsf{S}},\ \overline{\mathsf{E}}\rangle(\overline{\pi}, \overline{\mathsf{V}}))\big)$ (48)

⟅(A) separates the initial states from the later trace transitions considered in (B). This is correct, since:
(⇐) follows from $P \Rightarrow Q$ and $R \Rightarrow S$ implies $P \wedge R \Rightarrow Q \wedge S$.
(⇒) the converse might not hold e.g. in case there is no infinite trace starting from the initial state π_0 or $\overline{\pi}_0$. However, this case is prevented by $\mathsf{init} \subseteq \mathsf{inv}$ and $\overline{\mathsf{init}} \subseteq \overline{\mathsf{inv}}$ in (20) as well as feasibility (18). If (A) holds they imply that there is at least one concrete and abstract traces starting from these initial states and then (B) concludes the implication for all of these traces.⟆

(A)

$\Leftrightarrow \forall \mathsf{s}_0 \in \mathsf{init}\ .\ \exists \mathsf{s}\ .\ \langle \mathsf{s}_0,\ \mathsf{s}\rangle \in \tau^* \wedge \exists \overline{\mathsf{s}}_0 \in \overline{\mathsf{init}}\ .\ \langle \mathsf{s},\ \overline{\mathsf{s}}_0\rangle \in \rho$

This is a slight generalization of the base condition in the original definition of simulation [30] where $\mathsf{s}_0 = \mathsf{s}$.

$$
\begin{aligned}
&(B)\\
\Leftrightarrow\ &\forall \langle \mathsf{s},\, \mathsf{e}\rangle \in \mathsf{post}[\tau^*](\mathsf{init} \times \mathsf{evts}), \forall \langle \overline{\mathsf{s}},\, \overline{\mathsf{e}}\rangle \in \mathsf{post}[\overline{\tau}^*](\overline{\mathsf{init}} \times \overline{\mathsf{evts}}), \langle \mathsf{s}',\, \mathsf{e}'\rangle \,.\, \big(\langle\langle \mathsf{s},\, \mathsf{e}\rangle,\, \langle \overline{\mathsf{s}},\, \overline{\mathsf{e}}\rangle\rangle \in \rho \wedge \langle \mathsf{s},\, \mathsf{s}'\rangle \in \tau(\mathsf{e}) \wedge \mathsf{cvgt}\langle \mathsf{S},\, \mathsf{E}\rangle\langle \mathsf{inv},\, \mathsf{init},\, \mathsf{evts},\, \tau,\, \mathsf{V}\rangle\big) \Rightarrow \big(\exists \langle \overline{\mathsf{s}}',\, \overline{\mathsf{e}}'\rangle \,.\, \langle\langle \mathsf{s}',\, \mathsf{e}'\rangle,\, \langle \overline{\mathsf{s}}',\, \overline{\mathsf{e}}'\rangle\rangle \in \rho \wedge \langle \overline{\mathsf{s}},\, \overline{\mathsf{s}}'\rangle \in \overline{\tau}^*(\overline{\mathsf{e}}) \wedge \mathsf{cvgt}\langle \overline{\mathsf{S}},\, \overline{\mathsf{E}}\rangle\langle \overline{\mathsf{inv}},\, \overline{\mathsf{init}},\, \overline{\mathsf{evts}},\, \overline{\tau},\, \overline{\mathsf{V}}\rangle\big)
\end{aligned}
$$

where

$$
\begin{aligned}
&\mathsf{cvgt}\langle \mathsf{S},\, \mathsf{E}\rangle\langle \mathsf{inv},\, \mathsf{init},\, \mathsf{evts},\, \tau,\, \mathsf{V}\rangle \triangleq \forall \mathsf{e} \in \mathsf{E}, \langle \mathsf{s},\, \mathsf{e}\rangle \in \mathsf{inv} \cap \mathsf{post}[\tau^*](\mathsf{init} \times \mathsf{evts}),\\
&\qquad k \in \mathbb{N}, \langle \mathsf{s}',\, \mathsf{e}\rangle \in \mathsf{inv} \cap \mathsf{post}[\tau[\mathsf{e}]^k]\{\langle \mathsf{s},\, \mathsf{e}\rangle\} \,.\, k \leqslant \mathsf{V}\langle \mathsf{s},\, \mathsf{e}\rangle
\end{aligned}
\tag{49}
$$

expressing that any reachable state cannot be followed by more e-transitions than allowed by the variant V.

⟅The equivalence follows from the fact that π and $\overline{\pi}$ satisfy (B) implies that π_i is a descendant of the initial states and there is a transition between π_i and π_{i+1} by $\pi \in \mathbb{T}^{\mathrm{EvB}}\langle \mathsf{S},\, \mathsf{E}\rangle$ and def. post and same for $\overline{\pi}$ and the simulation relation ρ holds. Conversely, $\langle \overline{\mathsf{s}}',\, \overline{\mathsf{e}}'\rangle$ and $\langle \mathsf{s}',\, \mathsf{e}\rangle \in \mathsf{inv} \cap \mathsf{post}[\tau[\mathsf{e}]^k]\{\langle \mathsf{s},\, \mathsf{e}\rangle$ imply the existence of π by def. post, τ^*, and $\tau[\mathsf{e}]^k]$. Same for $\overline{\pi}$.

The reasoning is the same for convergence since $\mathsf{post}[\tau^*]$ is equivalent to the existence of an infinite trace by def. transition system satisfying feasibility (18).⟆

$$
\begin{aligned}
\Leftrightarrow\ &\exists I, \overline{I}, \forall \langle \mathsf{s},\, \mathsf{e}\rangle, \langle \overline{\mathsf{s}},\, \overline{\mathsf{e}}\rangle, \langle \mathsf{s}',\, \mathsf{e}'\rangle \,.\, \big((\mathsf{init} \times \mathsf{evts}) \subseteq I \wedge \mathsf{post}[\tau](I) \subseteq I \wedge (\overline{\mathsf{init}} \times \overline{\mathsf{evts}}) \subseteq \overline{I} \wedge \mathsf{post}[\overline{\tau}]\overline{I} \subseteq I \wedge \langle \mathsf{s},\, \mathsf{e}\rangle \in I, \langle \overline{\mathsf{s}},\, \overline{\mathsf{e}}\rangle \in \overline{I} \wedge \langle\langle \mathsf{s},\, \mathsf{e}\rangle,\, \langle \overline{\mathsf{s}},\, \overline{\mathsf{e}}\rangle\rangle \in \rho \wedge \langle \mathsf{s},\, \mathsf{s}'\rangle \in \tau(\mathsf{e}) \wedge \mathsf{cvgt}\langle \mathsf{S},\, \mathsf{E}\rangle\langle \mathsf{inv},\, \mathsf{init},\, \mathsf{evts},\, \tau,\, \mathsf{V}\rangle\big) \Rightarrow \big(\exists \langle \overline{\mathsf{s}}',\, \overline{\mathsf{e}}'\rangle \,.\, \langle\langle \mathsf{s}',\, \mathsf{e}'\rangle,\, \langle \overline{\mathsf{s}}',\, \overline{\mathsf{e}}'\rangle\rangle \in \rho \wedge \langle \overline{\mathsf{s}},\, \overline{\mathsf{s}}'\rangle \in \overline{\tau}^*(\overline{\mathsf{e}}) \wedge \mathsf{cvgt}\langle \overline{\mathsf{S}},\, \overline{\mathsf{E}}\rangle\langle \overline{\mathsf{inv}},\, \overline{\mathsf{init}},\, \overline{\mathsf{evts}},\, \overline{\tau},\, \overline{\mathsf{V}}\rangle\big)
\end{aligned}
\tag{50}
$$

⟅By the Turing/Naur/Floyd induction principle, we have $\mathsf{post}[\tau^*](\mathsf{init} \times \mathsf{evts}) \subseteq I$ and $\mathsf{post}[\overline{\tau}^*](\overline{\mathsf{init}} \times \overline{\mathsf{evts}}) \subseteq \overline{I}$;

($\Rightarrow$) Take $I = \mathsf{post}[\tau^*](\mathsf{init} \times \mathsf{evts})$ and $\overline{I} = \mathsf{post}[\overline{\tau}^*](\overline{\mathsf{init}} \times \overline{\mathsf{evts}})$;

($\Leftarrow$) If the property holds for all elements of I and $\overline{I}$, it holds for all elements of any of their subsets⟆

Verification condition (50) is split into several proof obligations in Event-B where I and $\overline{I}$ are machine invariants (according to the invariant preservation proof obligation rule INV [4, Sect. 5.2.2]).

(50) "ensures that when a concrete event is enabled, so is the corresponding abstract one" [4, p. 192], which is proof obligation GRD of [4, Sect. 5.2.4].

Assume that in (50), we have $\langle\langle \mathsf{s},\, \mathsf{e}\rangle,\, \langle \overline{\mathsf{s}},\, \overline{\mathsf{e}}\rangle\rangle \in \rho \wedge \langle\langle \mathsf{s},\, \mathsf{e}\rangle,\, \langle \overline{\mathsf{s}},\, \overline{\mathsf{e}}'\rangle\rangle \in \rho \wedge \langle \mathsf{s},\, \mathsf{s}'\rangle \in \tau(\mathsf{e})$ so that the concrete event merges two different abstract events, a case considered in [4, Sect. 5.2.5]. Then (50) states that for each of these two abstract events a transition must be possible. This is precisely the guard merging proof obligation rule MRG in [4, Sect. 5.2.5]. Notice that MRG is restricted to finitely possible abstract events while (50) doesn't. However, the events are finite in Event-B, so this is equivalent.

In case $\langle \bar{\mathsf{s}}, \bar{\mathsf{s}}' \rangle \in \overline{\tau}^*(\bar{\mathsf{e}})$ in (50) is restricted to $\langle \bar{\mathsf{s}}, \bar{\mathsf{s}}' \rangle \in \overline{\tau}(\bar{\mathsf{e}})$, we get the simulation proof obligation rule SIM of [4, Sect. 5.2.6]. However, (50) is more general, in that $\overline{\tau}^*$ might be the identity $\overline{\tau}^0$, which is handled in [4, Sect. 14.6.1] on splitting an abstract event. $\overline{\tau}^*$ might also correspond to multiple abstract transitions $\overline{\tau}^n$, $n > 1$, as handled in MRG [4, Sect. 14.6.2] for merging several abstract events.

The transitions $\tau(\mathsf{e})$ and $\overline{\tau}(\bar{\mathsf{e}})$ considered in (50) are defined by events in Event-B [4, Sect. 5.1.7]. The "Clause "with" in an event contains the *witness* of the corresponding abstract event. A witness has to be provided in a refining event *for each disappearing parameter* of the abstract event and *for each disappearing variable* assigned in a "non-deterministic way" in the abstract event. The non-deterministic witness proof obligation rule WFIS in [4, Sect. 5.2.10] "ensures that each witness proposed in the witness predicate of a concrete event indeed exists" [4, p. 201]. This corresponds to the condition $\exists \langle \bar{\mathsf{s}}', \bar{\mathsf{e}}' \rangle \,.\, \langle \langle \mathsf{s}', \mathsf{e}' \rangle, \langle \bar{\mathsf{s}}', \bar{\mathsf{e}}' \rangle \rangle \in \rho$ in (50).

Finally, (50) requires the event convergence (49) which is ensured by the variant proof obligation rule VAR [4, Sect. 5.2.9] which applies to both the concrete and abstract machines.

The same way that, by abstraction, the specification of a trace semantics induces proof obligations on events and machines generating this semantics, the refinement of semantics induces, by abstraction, proof obligations on the refinements of events and machines. This is a general idea in science, a global behavior is described locally, by abstraction.

6 A General Induction Principle

The ideas of Jean-Raymond on abstraction that governed the design of Event-B are of general interest, both in refinement and verification. Let us consider
- $\underline{\mathbb{S}}$ and $\overline{\mathbb{S}}$ be sets of states;
- $\underline{\mathbb{t}}$ and $\overline{\mathbb{t}}$ be the finite or infinite traces over the respective states $\underline{\mathbb{S}}$ and $\overline{\mathbb{S}}$;
- $\underline{\mathsf{T}} \in \wp(\underline{\mathbb{t}})$ and $\overline{\mathsf{T}} \in \wp(\overline{\mathbb{t}})$ be trace semantics;
- $R_s \in \wp(\wp(\underline{\mathbb{t}}) \times \wp(\overline{\mathbb{t}}))$ be a relation between trace semantics;
- $\underline{P} \in \wp(\wp(\underline{\mathbb{t}}))$ and $\overline{P} \in \wp(\wp(\overline{\mathbb{t}}))$ be semantic hyperproperties;
- $R_p \in \wp(\wp(\wp(\underline{\mathbb{t}})) \times \wp(\wp(\overline{\mathbb{t}})))$ be a relation between semantic hyperproperties.

A correctness proof can be formalized by the following induction principle

$$\frac{\overline{T} \in \overline{P}}{\underline{T} \in \underline{P}}, \quad \langle \underline{T}, \overline{T} \rangle \in R_s, \quad \langle \underline{P}, \overline{P} \rangle \in R_p \tag{51}$$

Notice that the side conditions of (51) cannot be integrated in the premise (as often done in computer science, for example in Hoare logic where side conditions appear in the premise [15, Sect. 26.7]). If $R_s = \varnothing$ or $R_p = \varnothing$ then (51) is not applicable whereas if incorporated in the premise, (51) becomes the obviously wrong axiom $\frac{\varnothing}{\underline{T} \in \underline{P}}$.

Soundness of (51) states that R_s and R_p should be chosen so that the hyperproperty $\overline{P}$ of $\overline{T}$ should imply the hyperproperty $\underline{P}$ of $\underline{T}$. Conversely, completeness of the induction principle (51) states that R_s and R_p should be chosen so

that if the hyperproperty $\underline{P}$ of $\underline{T}$ holds then this should be provable by showing that $\overline{T}$ has hyperproperty $\overline{P}$. Notice that R_p may incorporate the abstraction of hyperproperties to traces properties in $\wp(\overline{\mathbb{t}})$ and $\wp(\underline{\mathbb{t}})$.

An early example is Milner's introduction of the notion of simulation (and bisimulation, originally called "mutual simulation") [30] with it's "application to flowchart programs" in Sect. 4 to "demonstrate a simulation between two programs in a manner which bears a close relation to Floyd's method for proving correctness of a single program". The idea is further developed by Burstall [9]. Another example based on (51) is transformation in Continuation Passing Style (CPS) of higher-order functional programs before strictness analysis [25] which improves precision for the same abstract domain. In many cases however, (51) has one of R_s or R_p which is equality.

6.1 Verification by Property Abstraction. In verification by abstract interpretation one reasons on a given program with semantics $T = \underline{T} = \overline{T}$ so that $R_s = (=)$ is the identity in (51).

The condition for (51) to be sound when $R_s = (=)$ is that the premise of (51) should imply its conclusion when the side-condition $\langle \underline{P}, \overline{P} \rangle \in R_p$ holds, that is

$$\begin{aligned}
& \forall P, P' \,.\, (\langle P,\, P' \rangle \in R_p) \Rightarrow (\forall T \,.\, (T \in P') \Rightarrow T \in P) \\
\Leftrightarrow\ & R_p \subseteq \{\langle P,\, P' \rangle \mid \forall T \,.\, T \in P' \Rightarrow T \in P\} && \wr \text{def. } \subseteq \wr \\
\Leftrightarrow\ & R_p \subseteq \{\langle P,\, P' \rangle \mid P' \subseteq P\} && \wr \text{def. } \subseteq \wr \\
\Leftrightarrow\ & R_p \subseteq (\supseteq) && \wr \text{def. } \supseteq \wr
\end{aligned}$$

Completeness of (51) when $R_s = (=)$ means that when the side condition $\langle \underline{P}, \overline{P} \rangle \in R_p$ holds (so that the rule is applicable), if the conclusion is true, this should be provable by (51), so that the premise should hold. So we should have

$$\begin{aligned}
& \forall P, P' \,.\, (\langle P,\, P' \rangle \in R_p) \Rightarrow (\forall T \,.\, (T \in P) \Rightarrow T \in P') \\
\Leftrightarrow\ & R_p \subseteq (\subseteq) && \wr \text{def. } \subseteq \wr
\end{aligned}$$

The method is sound and complete if and only iff $R_p = (=)$ i.e. when no concrete information is lost in the abstract, which is the case only for isomorphic abstractions.

6.2 Verification by Program Refinement. As suggested by the explanation of the Event-B method in section 4, program verification can be done by refinement [23]. In verification by refinement, one reasons with respect to a fixed specification $P = \underline{P} = \overline{P}$ so that $R_p = (=)$ is the identity in (51).

The condition for (51) to be sound when $R_p = (=)$ is that the premise of (51) should imply its conclusion when the side-condition $\langle \underline{T}, \overline{T} \rangle \in R_s$ holds, that is, for all P and $\overline{P}$,

$$\begin{aligned}
& \forall T, \overline{T} \,.\, \langle T,\, \overline{T} \rangle \in R_s \Rightarrow (\overline{T} \in \overline{P} \Rightarrow T \in P) \\
\Leftrightarrow\ & \forall T \,.\, (\exists \overline{T} \,.\, \langle T,\, \overline{T} \rangle \in R_s \wedge \overline{T} \in \overline{P}) \Rightarrow T \in P && \wr \text{def. } \Rightarrow \wr \\
\Leftrightarrow\ & \{T \mid \exists \overline{T} \,.\, \langle T,\, \overline{T} \rangle \in R_s \wedge \overline{T} \in \overline{P}\} \subseteq P && \wr \text{def. } \subseteq \wr
\end{aligned}$$

$\Leftrightarrow \mathsf{pre}[R_s]\overline{P} \subseteq P$ ⦇def. pre⦈

$\Leftrightarrow \overline{P} \subseteq \widetilde{\mathsf{post}}[R_s]P$ ⦇[15, (12.23)]⦈

The condition for (51) to be complete when $R_p = (=)$ is that if the conclusion of (51) holds, then this must be provable by the proof principle, so that the premise should hold, that is, for all P and $\overline{P}$,

$\forall T, \overline{T} \,.\, (\langle T, \overline{T}\rangle \in R_s \wedge T \in P \Rightarrow \overline{T} \in \overline{P}$

$\Leftrightarrow \forall \overline{T} \,.\, (\exists T \,.\, \langle T, \overline{T}\rangle \in R_s \wedge T \in P) \Rightarrow \overline{T} \in \overline{P})$ ⦇def. ⇒⦈

$\Leftrightarrow \{\overline{T} \mid \exists T \,.\, T \in P \wedge \langle T, \overline{T}\rangle \in R_s\} \subseteq \overline{P}$ ⦇def. ⊆ and commutativity⦈

$\Leftrightarrow \mathsf{post}[R_s]P \subseteq \overline{P}$ ⦇def. pre⦈

$\Leftrightarrow P \subseteq \widetilde{\mathsf{pre}}[R_s]\overline{P}$ ⦇[15, (12.22)]⦈

It follows that (51) is sound and complete when $R_p = (=)$ if and only if $\mathsf{pre}[R_s]\overline{P} \subseteq P \subseteq \widetilde{\mathsf{pre}}[R_s]\overline{P}$ (or equivalently $\mathsf{post}[R_s]P \subseteq \overline{P} \subseteq \widetilde{\mathsf{post}}[R_s]P$).

6.3 Program Refinement. In machine/program refinement, the abstract specification $\overline{P}$ is the abstract machine/program semantics $\overline{T}$ itself so $\overline{P} = \{\overline{T}\}$. The refined machine/program semantics $\underline{T}$ must be a subset of the abstract machine/program semantics, so $R_s = \{\langle \underline{T}, \overline{T}\rangle \mid \varnothing \neq \underline{T} \subseteq \overline{T}\}$ and $R_p = \{\langle \underline{P}, \{\overline{T}\}\rangle \mid \underline{P} \in \wp(\overline{T}) \smallsetminus \{\varnothing\}\}$ in (51). In this case, (51) becomes "design the refined machine/program semantics $\underline{T}$ such that $\varnothing \neq \underline{T} \subseteq \overline{T}$" which is the simple trace refinement in section 4. Abstraction or concretization functions can be use to include the other refinements defined in this section.

6.4 Static Analysis. As noted in (38), the rôles are inverted in static analysis where the concrete program semantics $\overline{T}$ is the specification, $\overline{P} = \{\overline{T}\}$ is the collecting semantics, and the abstract semantics $\underline{T}$ should be an over-approximation $\underline{T} \supseteq \overline{T}$, up to either a Galois connection, a concretization, or an abstraction function [20], which can be formalized by appropriate choices of R_s and R_p in (51).

7 Conclusion

We have sketched the semantics of Events-B in terms of abstract interpretation and defined the semantics of machines and the semantics of refinement as the abstraction by Galois connections. We have derived by abstract interpretation the proof obligations for Event-B contexts and machines to satisfy the requirements of Event-B trace properties. The abstract interpretation approach [21] might also be useful to extend Event-B to consider traces for weak memory models [8] and hyperproperties [11].

Moreover, in abstract interpretation knowing the concrete semantics and the abstraction yields the abstract semantics by calculational design (e.g. [12,21]). This means that making abstractions explicit, as we did, could yield refinements by calculation design (preferred to manual refinement followed by verification).

Finally, we have considered the abstraction/refinement of both semantics and properties and shown in this context that verification and refinement are dual. By this duality, the study of only one of verification or refinement is necessary since the results immediately apply, by duality, to the other. However, the communities, techniques, and literatures are, in part, but mostly disconnected.

Acknowledgements. I thank Dominique Méry for pointing out my misconceptions of Event-B in a first version of this paper[4].

References

1. Abrial, J.: Data semantics. In: Klimbie, J.W., Koffeman, K.L. (eds.) IFIP Working Conference Data Base Management. pp. 1–60. North-Holland (1974)
2. Abrial, J.: The B-book - assigning programs to meanings. Cambridge University Press (1996)
3. Abrial, J.: Extending B without changing it (for developing distributed systems). In: Habrias, H. (ed.) Proceedings of the 1st Conference on the B method. pp. 169–191. Springer (Nov 1996)
4. Abrial, J.: Modeling in Event-B: System and Software Engineering. Cambridge University Press (2010)
5. Abrial, J., Butler, M.J., Hallerstede, S., Hoang, T.S., Mehta, F., Voisin, L.: Rodin: an open toolset for modelling and reasoning in Event-B. Int. J. Softw. Tools Technol. Transf. **12**(6), 447–466 (Nov 2010). https://doi.org/10.5555/1895404.1895406
6. Abrial, J., Cansell, D.: Click'n prove: Interactive proofs within set theory. In: TPHOLs. Lecture Notes in Computer Science, vol. 2758, pp. 1–24. Springer (2003)
7. Abrial, J., Mussat, L.: Introducing dynamic constraints in B. In: B. Lecture Notes in Computer Science, vol. 1393, pp. 83–128. Springer (1998)
8. Alglave, J., Cousot, P.: Syntax and analytic semantics of LISA. CoRR **abs/1608.06583** (Aug 2016), http://arxiv.org/abs/1608.06583
9. Burstall, R.M.: An algebraic description of programs with assertions, verification and simulation. In: Proving Assertions About Programs. pp. 7–14. ACM (1972). https://doi.org/10.1145/942578.807068
10. Chandy, K.M., Misra, J.: Parallel Program Design: A Foundation. Addison-Wesley (Jan 1988)
11. Clarkson, M.R., Schneider, F.B.: Hyperproperties. J. Comput. Secur. **18**(6), 1157–1210 (2010). https://doi.org/10.3233/JCS-2009-0393
12. Cousot, P.: The calculational design of a generic abstract interpreter. In: Broy, M., Steinbrüggen, R. (eds.) Calculational System Design. NATO ASI Series F. IOS Press, Amsterdam (1999)
13. Cousot, P.: A formal introduction to abstract interpretation. In: Pretschner, A., Müller, P., Stöckle, P. (eds.) Calculational System Design. NATO SPS, Series D, Vol. 53. IOS Press, Amsterdam (1999)

[4] Among others, I misinterpreted the parameters in the any clause of events [3, Fig. 5.10] as input or output parameters to communicate with the environment, whereas the system is closed in Event-B and these parameters are existentially quantified. Curiously, the paper is still coherent with this misconception and enables Event-B to design open systems, provided event convergence is defined not to depend on these parameters.

14. Cousot, P.: Constructive design of a hierarchy of semantics of a transition system by abstract interpretation. Theor. Comput. Sci. **277**(1–2), 47–103 (2002). https://doi.org/10.1016/S0304-3975(00)00313-3
15. Cousot, P.: Principles of Abstract Interpretation. MIT Press, 1 edn. (2021)
16. Cousot, P., Cousot, R.: Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In: POPL. pp. 238–252. ACM (1977). https://doi.org/10.1145/512950.512973
17. Cousot, P., Cousot, R.: Systematic design of program analysis frameworks. In: POPL. pp. 269–282. ACM Press (1979). https://doi.org/https://dl.acm.org/doi/10.1145/3728920
18. Cousot, P., Cousot, R.: Induction principles for proving invariance properties of programs. In: Néel, D. (ed.) Tools & Notions for Program Construction: an Advanced Course. pp. 75–119. Cambridge University Press, Cambridge, UK (Aug 1982)
19. Cousot, P., Cousot, R.: 'a la Floyd' induction principles for proving inevitability properties of programs. In: Nivat, M., Reynolds, J. (eds.) Algebraic methods in semantics. pp. 277–312. Cambridge University Press, Cambridge, UK (Dec 1985)
20. Cousot, P., Cousot, R.: Abstract interpretation frameworks. J. Log. Comput. **2**(4), 511–547 (1992). https://doi.org/10.1093/LOGCOM/2.4.511
21. Cousot, P., Wang, J.: Calculational design of hyperlogics by abstract interpretation. Proc. ACM Program. Lang. **9**(POPL), 446–478 (2025). https://doi.org/10.48550/arXiv.2411.11113
22. Cousot, R.: Fondements des méthodes de preuve d'invariance et de fatalité de programmes parallèles (in French). Thèse d'État ès sciences mathématiques, Nancy, Institut national polytechnique de Lorraine (15 November 1985), https://pcousot.github.io/publications/RadhiaCousotTheseEsSciences.PDF
23. Dijkstra, E.W.: A constructive approach to the problem of program correctness. BIT Numerical Mathematics **8(3)**, 174–186 (Sep 1968). https://doi.org/10.1007/BF01933419
24. Dijkstra, E.W.: On weak and strong termination. In: Selected Writings on Computing: A Personal Perspective. pp. 355—357. Texts and Monographs in Computer Science, Springer (1982), originally published as EWD673 (1978)
25. Filho, J.M., Burn, G.L.: Continuation passing transformation and abstract interpretation. In: Burn, G.L., Gay, S.J., Ryan, M. (eds.) Theory and Formal Methods 1993, Proceedings of the First Imperial College Department of Computing Workshop on Theory and Formal Methods, Isle of Thorns Conference Centre, Chelwood Gate, Sussex, UK, 29-31 March 1993. pp. 247–259. Workshops in Computing, Springer (1993)
26. Hoang, T.S.: An introduction to the Event-B modelling method. In: Industrial Deployment of System Engineering Methods, pp. 211–236. Springer (2013)
27. Lamport, L.: How to make a multiprocessor computer that correctly executes multiprocess programs. IEEE Trans. Computers **28**(9), 690–691 (1979). https://doi.org/10.1109/TC.1979.1675439
28. Lamport, L.: Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers. Addison-Wesley Professional (Jul 2002)
29. Méry, D., Merz, S.: Analysis of a termination detection algorithm using Event-B and TLA+. This volume (2026)
30. Milner, R.: An algebraic definition of simulation between programs. In: IJCAI. pp. 481–489. William Kaufmann (1971). https://doi.org/10.5555/1622876.1622926

31. Méry, D., Singh, N.K.: Event B, chap. 10, pp. 253–298. Formal Methods Applied to Complex Systems: Implementation of the B Method, J.-L. Boulanger (Ed.), Wiley, 1st edn. (2014)
32. Pnueli, A.: The temporal logic of programs. In: FOCS. pp. 46–57. IEEE Computer Society (1977). https://doi.org/10.1109/SFCS.1977.32
33. Tarski, A.: A lattice theoretical fixpoint theorem and its applications. Pacific J. of Math. **5**, 285–310 (1955). https://doi.org/10.2140/pjm.1955.5.285
34. Voisin, L., Abrial, J.: The Rodin platform has turned ten. In: ABZ. Lecture Notes in Computer Science, vol. 8477, pp. 1–8. Springer (2014)

A Proofs for Section 2 (The Semantics of Event-B Systems)

Proof (that BNd preserves history independence). We must prove that $\mathsf{Ffus} \circ \mathsf{BNd} \circ \mathsf{Ffus} = \mathsf{BNd} \circ \mathsf{Ffus}$. By fixpoint definition of Ffus, it is sufficient to prove that $\forall \mathsf{T} \,.\, \mathsf{Efus} \circ \mathsf{BNd} \circ \mathsf{Ffus}(T) = \mathsf{BNd} \circ \mathsf{Ffus}(T)$. Assume the contrary. By def. of Efus, we have $\pi_1 \langle \mathsf{s}, \mathsf{e} \rangle \pi_2, \pi_1' \langle \mathsf{s}, \mathsf{e} \rangle \pi_2' \in \mathsf{BNd} \circ \mathsf{Ffus}(T)$ such that $\pi_1 \langle \mathsf{s}, \mathsf{e} \rangle \pi_2' \notin \mathsf{BNd} \circ \mathsf{Ffus}(T)$. Since $\pi_1 \langle \mathsf{s}, \mathsf{e} \rangle \pi_2'$ does not belong to $\mathsf{BNd} \circ \mathsf{Ffus}(T)$ there must be a prefix of $\pi_1 \langle \mathsf{s}, \mathsf{e} \rangle \pi_2'$ with infinitely many successors. There are 3 possible cases.

1. This prefix cannot be $\pi_1 \langle \mathsf{s}, \mathsf{e} \rangle$ since then $\pi_1 \langle \mathsf{s}, \mathsf{e} \rangle \pi_2$ would have been eliminated from $\mathsf{BNd} \circ \mathsf{Ffus}(T)$ by BNd;
2. This prefix cannot be $\pi_1^0 \langle \mathsf{s}', \mathsf{e}' \rangle$ in $\pi_1 = \pi_1^0 \langle \mathsf{s}', \mathsf{e}' \rangle \pi_1^1$ since then $\pi_1 \langle \mathsf{s}, \mathsf{e} \rangle \pi_2 = \pi_1^0 \langle \mathsf{s}', \mathsf{e}' \rangle \pi_1^1 \langle \mathsf{s}, \mathsf{e} \rangle \pi_2'$ would have been eliminated from $\mathsf{BNd} \circ \mathsf{Ffus}(T)$ by BNd;
3. This prefix cannot be $\pi_1 \langle \mathsf{s}, \mathsf{e} \rangle \pi_2^0 \langle \mathsf{s}', \mathsf{e}' \rangle$ where $\pi_2 = \pi_2^0 \langle \mathsf{s}', \mathsf{e}' \rangle \pi_2^1$ since then $\pi_1 \langle \mathsf{s}, \mathsf{e} \rangle \pi_2 = \pi_1 \langle \mathsf{s}, \mathsf{e} \rangle \pi_2^0 \langle \mathsf{s}', \mathsf{e}' \rangle \pi_2^1$ would have been eliminated from $\mathsf{BNd} \circ \mathsf{Ffus}(T)$ by BNd.

By reductio ad absurdum, we conclude that $\pi_1 \langle \mathsf{s}, \mathsf{e} \rangle \pi_2' \in \mathsf{BNd} \circ \mathsf{Ffus}(T)$ proving history preservation.

B Proofs for Section 3 (Specifying the System Semantics by Abstraction Into Machines (i.e. Discrete Transition Systems))

Proof. of (24) — Assume that $\mathsf{T} \in \mathbb{T}^{\mathrm{FB}}\langle \mathsf{S}, \mathsf{E} \rangle$. We must prove that $\alpha_t\langle \mathsf{S}, \mathsf{E} \rangle(\mathsf{T}) \in \mathbb{T}\mathbb{r}^{\mathrm{FB}}\langle \mathsf{S}, \mathsf{E} \rangle$. $\mathsf{Feas}(\langle \mathsf{inv}, \mathsf{evts}, \tau \rangle)$ for (17) directly follows from feasibility for traces which itself follows from (3) and (2) requiring traces to be infinite, so no state can be blocking. $\mathsf{BNd}(\langle \mathsf{inv}, \mathsf{evts}, \tau \rangle)$ for (17.d)

— Conversely, given $\langle \mathsf{inv}, \mathsf{evts}, \mathsf{init}, \tau \rangle \in \mathbb{T}\mathbb{r}^{\mathrm{FB}}\langle \mathsf{S}, \mathsf{E} \rangle$, we must prove that $\gamma_t\langle \mathsf{S}, \mathsf{E} \rangle(\langle \mathsf{inv}, \mathsf{evts}, \mathsf{init}, \tau \rangle) \in \mathbb{T}^{\mathrm{EvB}}\langle \mathsf{S}, \mathsf{E} \rangle$. History independence 6 directly follows from the definition (23) of γ_t . If $\pi_1 \langle \mathsf{s}, \mathsf{e} \rangle \langle \mathsf{s}', \mathsf{e}' \rangle \pi_2$ and $\pi_1' \langle \mathsf{s}, \mathsf{e} \rangle \langle \mathsf{s}'', \mathsf{e}'' \rangle \pi_2'$ belong to $\gamma_t\langle \mathsf{S}, \mathsf{E} \rangle(\langle \mathsf{inv}, \mathsf{evts}, \mathsf{init}, \tau \rangle)$ then $\pi_1 \langle \mathsf{s}, \mathsf{e} \rangle$ is a finite succession of transitions starting with an initial trace event, $\langle \langle \mathsf{s}, \mathsf{e} \rangle, \langle \mathsf{s}'', \mathsf{e}'' \rangle \rangle \in \tau$ is a transition, and $\langle \mathsf{s}'', \mathsf{e}'' \rangle \pi_2'$ is an infinite succession of transitions so $\pi_1 \langle \mathsf{s}, \mathsf{e} \rangle \langle \mathsf{s}'', \mathsf{e}'' \rangle \pi_2'$ is also an infinite succession of transitions in $\mathbb{t}^{\infty}\langle \mathsf{S}, \mathsf{E} \rangle$ and therefore belongs to $\gamma_t\langle \mathsf{S}, \mathsf{E} \rangle(\langle \mathsf{inv}, \mathsf{evts}, \mathsf{init}, \tau \rangle)$ starting with an initial trace event. Locally bounded nondeterminism (7) follows from (19) and would also hold for globally bounded nondeterminism. By (23), we conclude that all traces in $\gamma_t\langle \mathsf{S}, \mathsf{E} \rangle(\langle \mathsf{inv}, \mathsf{evts}, \mathsf{init}, \tau \rangle)$ are history independent and locally bounded, proving that $\gamma_t\langle \mathsf{S}, \mathsf{E} \rangle(\langle \mathsf{inv}, \mathsf{evts}, \mathsf{init}, \tau \rangle \in \mathbb{T}^{\mathrm{FB}}\langle \mathsf{S}, \mathsf{E} \rangle$.

— Finally, we must check the definition of a Galois connection.

$$
\begin{array}{ll}
 & \alpha_t(\mathsf{T}) \mathrel{\dot{\subseteq}} \langle \mathsf{inv}, \mathsf{evts}, \mathsf{init}, \tau \rangle \\
\Leftrightarrow & \{\mathsf{s} \mid \exists \pi \in \mathsf{T} \,.\, \exists i \in \mathbb{N} \,.\, \exists \mathsf{e} \in \mathsf{E} \,.\, \pi_i = \langle \mathsf{s}, \mathsf{e} \rangle\} \subseteq \mathsf{inv} \wedge \\
 & \{\mathsf{e} \mid \exists \pi \in \mathsf{T} \,.\, \exists i \in \mathbb{N} \,.\, \exists \mathsf{s} \in \mathsf{S} \,.\, \pi_i = \langle \mathsf{s}, \mathsf{e} \rangle\} \subseteq \mathsf{evts} \wedge \\
 & \{\mathsf{s}_0 \mid \exists \pi \in \mathsf{T} \,.\, \exists \mathsf{e}_0 \in \mathsf{E} \,.\, \pi_0 = \langle \mathsf{s}_0, \mathsf{e}_0 \rangle\} \subseteq \mathsf{init} \wedge \\
 & \boldsymbol{\lambda}\, \mathsf{e} \bullet \{\langle \mathsf{s}, \mathsf{s}' \rangle \mid \exists \pi \in \mathsf{T} \,.\, \exists i \in \mathbb{N} \,.\, \exists \mathsf{e}' \in \mathsf{E} \,.\, \pi_i = \langle \mathsf{s}, \mathsf{e} \rangle \wedge \pi_{i+1} = \langle \mathsf{s}', \mathsf{e}' \rangle\} \mathrel{\dot{\subseteq}} \tau
\end{array}
$$

$$\text{(def. (17) of } \alpha_t\text{, componentwise def. of } \dot{\subseteq}\text{, and renaming)}$$

$$\Leftrightarrow\ \forall \pi \in \mathsf{T}\ .\ \forall i \in \mathbb{N}\ .\ \pi_i \in \mathsf{inv} \times \mathsf{evts} \wedge \pi_0 \in \mathsf{init} \times \mathsf{evts} \wedge \forall \mathsf{e}\ .\ \{\langle \mathsf{s}, \mathsf{s}' \rangle \mid \exists \pi \in \mathsf{T}\ .\ \exists i \in \mathbb{N}\ .\ \exists \mathsf{e}' \in \mathsf{E}\ .\ \pi_i = \langle \mathsf{s}, \mathsf{e} \rangle \wedge \pi_{i+1} = \langle \mathsf{s}', \mathsf{e}' \rangle\} \subseteq \tau(\mathsf{e})$$

$$\text{(def. } \subseteq \text{ and } \times\text{, pointwise def. } \dot{\subseteq})$$

$$\Leftrightarrow\ \forall \pi \in \mathsf{T}\ .\ \forall i \in \mathbb{N}\ .\ \pi_i \in \mathsf{inv} \times \mathsf{evts} \wedge \pi_0 \in \mathsf{init} \times \mathsf{evts} \wedge \forall \mathsf{s}, \mathsf{s}' \in \mathsf{S}, \mathsf{e}, \mathsf{e}' \in \mathsf{E}\ .\ (\pi_i = \langle \mathsf{s}, \mathsf{e} \rangle \wedge \pi_{i+1} = \langle \mathsf{s}', \mathsf{e}' \rangle) \Rightarrow (\langle \mathsf{s}, \mathsf{s}' \rangle \in \tau(\mathsf{e})$$

$$\text{(def. } \subseteq)$$

$$\Leftrightarrow\ \mathsf{T} \subseteq \{\pi \mid \pi_0 \in \mathsf{init} \times \mathsf{evts} \wedge \forall i \in \mathbb{N}\ .\ \pi_i \in \mathsf{inv} \times \mathsf{evts} \wedge \forall \mathsf{s}, \mathsf{s}' \in \mathsf{S}, \mathsf{e}, \mathsf{e}' \in \mathsf{E}\ .\ (\pi_i = \langle \mathsf{s}, \mathsf{e} \rangle \wedge \pi_{i+1} = \langle \mathsf{s}', \mathsf{e}' \rangle) \Rightarrow (\langle \mathsf{s}, \mathsf{s}' \rangle \in \tau(\mathsf{e})\}$$

$$\text{($\wedge$ commutative and def. } \subseteq)$$

$$\Leftrightarrow\ \mathsf{T} \subseteq \gamma_t\langle \mathsf{S}, \mathsf{E} \rangle(\langle \mathsf{inv}, \mathsf{evts}, \mathsf{init}, \tau \rangle) \qquad \text{(def. (21) of } \gamma_t\text{)} \qquad \square$$

Proof (of (30)). By definition of τ in (17.d), $\langle \mathsf{s}, \mathsf{s}' \rangle \in \tau(\mathsf{e})$ implies $\exists \pi \in \mathsf{T}\ .\ \exists i \in \mathbb{N}\ .\ \exists \mathsf{e}' \in \mathsf{E}\ .\ \pi_i = \langle \mathsf{s}_i, \mathsf{e} \rangle$ with $\mathsf{s}_i = \mathsf{s}$ and $\pi_{i+1} = \langle \mathsf{s}_{i+1}, \mathsf{e}_{i+1} \rangle$ with $\mathsf{s}_{i+1} = \mathsf{s}'$.

By (13), there exists a bound n such that for any such π and index i, we have $\pi_{i+2} = \langle \mathsf{s}_{i+2}, \mathsf{e} \rangle$, ..., $\pi_{i+k} = \langle \mathsf{s}_{i+k}, \mathsf{e} \rangle$, and $\pi_{i+k+1} = \langle \mathsf{s}_{i+k}, \mathsf{e}' \rangle$ with $\mathsf{e}' \neq \mathsf{e}$ and $k \leqslant n$ (including the case $k = n = 1$).

If $\ldots \pi_{i+1} \pi'_{i+2} \ldots, \pi'_{i+k} \ldots$ is the longest possible sequence of events e on any trace going through the state π_{i+1}, so that the event of π'_{i+k+1} is not e, then, by (28), the maximal number of consecutive occurrences of event e after π_{i+1} is $\mathsf{V}(\pi_{i+1}) = \alpha_v\langle \mathsf{S}, \mathsf{E} \rangle(\pi_{i+1}) = k - (i+1)$.

Let us now consider the state π_i. There are two cases.

- Either there is a trace $\ldots \pi_i \pi''_{i+2} \ldots, \pi''_{i+\ell} \ldots$ with $\ell > k$ consecutive events e, in which case $\mathsf{V}(\pi_i) = \alpha_v\langle \mathsf{S}, \mathsf{E} \rangle(\pi_i) = \ell - i > k - (i+1)$ so that (30) holds;
- Or else all such traces have less than k consecutive events e, in which case, by history independence $\mathsf{Ffus}(\mathsf{T}) = \mathsf{T}$, $\ldots \pi_i \pi_{i+1} \pi'_{i+2} \ldots, \pi'_{i+k} \ldots$ is also a trace through π_i, and therefore $\mathsf{V}(\pi_i) = \alpha_v\langle \mathsf{S}, \mathsf{E} \rangle(\pi_i) = k - i > k - (i+1)$ so that (30) holds again. $\square$

Proof (of (25)). Because of event convergence (13), $\alpha_v\langle \mathsf{S}, \mathsf{E} \rangle(\mathsf{T})$ is always well-defined since the number of successive computation steps for a given event e on any trace of T is bounded [5].

$$\alpha_v\langle \mathsf{S}, \mathsf{E} \rangle(\mathsf{T}) \mathrel{\dot{\leqslant}} \mathsf{V}$$

$$\Leftrightarrow \forall \langle \mathsf{s}, \mathsf{e} \rangle\ .\ \alpha_v\langle \mathsf{S}, \mathsf{E} \rangle(\mathsf{T})\langle \mathsf{s}, \mathsf{e} \rangle \leqslant \mathsf{V}\langle \mathsf{s}, \mathsf{e} \rangle \qquad \text{(pointwise def. } \dot{\leqslant}\text{)}$$

$$\Leftrightarrow \forall \langle \mathsf{s}, \mathsf{e} \rangle\ .\ \max\{j - i \mid j > i \in \mathbb{N} \wedge \exists \pi \in \mathsf{T}\ .\ \forall k \in [i, j]\ .\ \exists \mathsf{s}_k\ .\ \mathsf{s}_i = \mathsf{s} \wedge \pi_k = \langle \mathsf{s}_k, \mathsf{e} \rangle\} \leqslant \mathsf{V}\langle \mathsf{s}, \mathsf{e} \rangle \qquad \text{(def. (28) of } \alpha_v\text{)}$$

$$\Leftrightarrow \forall \langle \mathsf{s}, \mathsf{e} \rangle\ .\ \forall n \in \{j - i \mid j > i \in \mathbb{N} \wedge \exists \pi \in \mathsf{T}\ .\ \forall k \in [i, j]\ .\ \exists \mathsf{s}_k\ .\ \mathsf{s}_i = \mathsf{s} \wedge \pi_k = \langle \mathsf{s}_k, \mathsf{e} \rangle\}\ .\ n \leqslant \mathsf{V}\langle \mathsf{s}, \mathsf{e} \rangle \qquad \text{(def. max)}$$

$$\Leftrightarrow \forall \langle \mathsf{s}, \mathsf{e} \rangle\ .\ \forall j > i \in \mathbb{N}\ .\ (\exists \pi \in \mathsf{T}\ .\ \forall k \in [i, j]\ .\ \exists \mathsf{s}_k\ .\ \mathsf{s}_i = \mathsf{s} \wedge \pi_k = \langle \mathsf{s}_k, \mathsf{e} \rangle) \Rightarrow (j - i \leqslant \mathsf{V}\langle \mathsf{s}, \mathsf{e} \rangle) \qquad \text{(def. } \in\text{)}$$

$$\Leftrightarrow \forall \langle \mathsf{s}, \mathsf{e} \rangle\ .\ \forall j > i \in \mathbb{N}\ .\ \forall \pi \in \mathsf{T}\ .\ (\forall k \in [i, j]\ .\ \exists \mathsf{s}_k\ .\ \mathsf{s}_i = \mathsf{s} \wedge \pi_k = \langle \mathsf{s}_k, \mathsf{e} \rangle) \Rightarrow (j - i \leqslant \mathsf{V}\langle \mathsf{s}, \mathsf{e} \rangle) \qquad \text{(def. } \Rightarrow\text{)}$$

[5] Observe that the alternative definition of (13) in remark 1, which is unbounded, would require transfinite ordinals.

$\Leftrightarrow \forall \pi \in \mathsf{T} \,.\, \forall \langle \mathsf{s}, \mathsf{e}\rangle \,.\, \forall j > i \in \mathbb{N} \,.\, (\forall k \in [i,j] \,.\, \exists \mathsf{s}_k \,.\, \mathsf{s}_i = \mathsf{s} \wedge \pi_k = \langle \mathsf{s}_k, \mathsf{e}\rangle) \Rightarrow (j - i \leqslant \mathsf{V}\langle \mathsf{s}, \mathsf{e}\rangle)$ ⟅def. ∀⟆

$\Leftrightarrow \mathsf{T} \subseteq \{\pi \mid \forall \langle \mathsf{s}, \mathsf{e}\rangle \,.\, \forall j > i \in \mathbb{N} \,.\, (\forall k \in [i,j] \,.\, \exists \mathsf{s}_k \,.\, \mathsf{s}_i = \mathsf{s} \wedge \pi_k = \langle \mathsf{s}_k, \mathsf{e}\rangle) \Rightarrow (j - i \leqslant \mathsf{V}\langle \mathsf{s}, \mathsf{e}\rangle)\}$ ⟅def. ⊆⟆

$\Leftrightarrow \mathsf{T} \subseteq \{\pi \mid \mathsf{cvgt}\langle \mathsf{S}, \mathsf{E}\rangle(\pi, \mathsf{V})\}$ ⟅∀s . ∀i . ∃s_i . s_i = s always holds⟆

$\Leftrightarrow \mathsf{T} \subseteq \gamma_v\langle \mathsf{S}, \mathsf{E}\rangle(\mathsf{V})$ ⟅def. (29) of γ_v⟆ □

Proof (of (34)). The Cartesian product of Galois connections (24) and (25) is a Galois connection (34). For the isomorphism, we have

— $\alpha_t \dot{\times} \alpha_v\langle \mathsf{S}, \mathsf{E}\rangle \circ \gamma_t \dot{\times} \gamma_v\langle \mathsf{S}, \mathsf{E}\rangle(\langle \mathsf{inv}, \mathsf{evts}, \mathsf{init}, \tau, \mathsf{V}\rangle)$

$= \mathtt{let}\ \mathsf{T} = \{\pi \in \mathbb{t}^\infty\langle \mathsf{S}, \mathsf{E}\rangle \mid \pi_0 \in \mathsf{init} \times \mathsf{evts} \wedge \forall i \in \mathbb{N} \,.\, \pi_i \in \mathsf{inv} \times \mathsf{evts} \wedge \forall \langle \mathsf{s}, \mathsf{e}\rangle = \pi_i, \langle \mathsf{s}', \mathsf{e}'\rangle = \pi_{i+1} \,.\, \langle \mathsf{s}, \mathsf{s}'\rangle \in \tau(\mathsf{e}) \wedge \mathsf{cvgt}\langle \mathsf{S}, \mathsf{E}\rangle(\pi, \mathsf{V})\}\ \mathtt{in}$
$\langle \{\mathsf{s} \mid \exists \pi \in \mathsf{T} \,.\, \exists i \in \mathbb{N} \,.\, \exists \mathsf{e} \in \mathsf{E} \,.\, \pi_i = \langle \mathsf{s}, \mathsf{e}\rangle\},$
$\{\mathsf{s} \mid \exists \pi \in \mathsf{T} \,.\, \exists \mathsf{e} \in \mathsf{E} \,.\, \pi_0 = \langle \mathsf{s}, \mathsf{e}\rangle\},$
$\{\mathsf{e} \mid \exists \pi \in \mathsf{T} \,.\, \exists i \in \mathbb{N} \,.\, \exists \mathsf{s} \in \mathsf{S} \,.\, \pi_i = \langle \mathsf{s}, \mathsf{e}\rangle\},$
$\boldsymbol{\lambda}\, \mathsf{e} \bullet \{\langle \mathsf{s}, \mathsf{s}'\rangle \mid \exists \pi \in \mathsf{T} \,.\, \exists i \in \mathbb{N} \,.\, \exists \mathsf{e}' \in \mathsf{E} \,.\, \pi_i = \langle \mathsf{s}, \mathsf{e}\rangle \wedge \pi_{i+1} = \langle \mathsf{s}', \mathsf{e}'\rangle\},$
$\boldsymbol{\lambda}\, \langle \mathsf{s}, \mathsf{e}\rangle \bullet \max\{j - i \mid j > i \in \mathbb{N} \wedge \exists \pi \in \mathsf{T} \,.\, \forall k \in [i,j] \,.\, \exists \mathsf{s}_k \,.\, \mathsf{s}_i = \mathsf{s} \wedge \pi_k = \langle \mathsf{s}_k, \mathsf{e}\rangle\}\rangle$
⟅(33), (32), and def. function composition ∘⟆

$= \langle \mathsf{inv}, \mathsf{evts}, \mathsf{init}, \tau, \mathsf{V}\rangle$

⟅($\dot{\subseteq}$) follows from the Galois connection (34). By antisymmetry, it remains to prove the converse.
($\dot{\supseteq}$) is proved componentwise, by showing that

- If $\pi \in \mathsf{T}$ and $i \in \mathbb{N}$ are such that $\pi_i = \langle \mathsf{s}, \mathsf{e}\rangle$ then by def. of T and (22) all states and events are in the invariant inv × evts proving that $\{\mathsf{s} \mid \exists \pi \in \mathsf{T} \,.\, \exists i \in \mathbb{N} \,.\, \exists \mathsf{e} \in \mathsf{E} \,.\, \pi_i = \langle \mathsf{s}, \mathsf{e}\rangle\} \subseteq \mathsf{inv}$ and $\{\mathsf{e} \mid \exists \pi \in \mathsf{T} \,.\, \exists i \in \mathbb{N} \,.\, \exists \mathsf{s} \in \mathsf{S} \,.\, \pi_i = \langle \mathsf{s}, \mathsf{e}\rangle\} \subseteq \mathsf{evts}$.
- If $\pi \in \mathsf{T}$ then by definition of T, $\pi_0 \in \mathsf{init} \times \mathsf{evts}$ so if $\exists \mathsf{e} \in \mathsf{E} \,.\, \pi_0 = \langle \mathsf{s}, \mathsf{e}\rangle$ then $\mathsf{s} \in \mathsf{init}$ proving that $\{\mathsf{s} \mid \exists \pi \in \mathsf{T} \,.\, \exists \mathsf{e} \in \mathsf{E} \,.\, \pi_0 = \langle \mathsf{s}, \mathsf{e}\rangle\} \subseteq \mathsf{init}$
- If $\pi \in \mathsf{T}$, $\pi_i = \langle \mathsf{s}, \mathsf{e}\rangle$, and $\pi_{i+1} = \langle \mathsf{s}', \mathsf{e}'\rangle$ then $\langle \mathsf{s}, \mathsf{s}'\rangle \in \tau(\mathsf{e})$, proving that $\boldsymbol{\lambda}\, \mathsf{e} \bullet \{\langle \mathsf{s}, \mathsf{s}'\rangle \mid \exists \pi \in \mathsf{T} \,.\, \exists i \in \mathbb{N} \,.\, \exists \mathsf{e}' \in \mathsf{E} \,.\, \pi_i = \langle \mathsf{s}, \mathsf{e}\rangle \wedge \pi_{i+1} = \langle \mathsf{s}', \mathsf{e}'\rangle\} \mathrel{\dot{\subseteq}} \tau$
- Given any $\pi \in \mathsf{T}$, $\langle \mathsf{s}, \mathsf{e}\rangle \in \langle \mathsf{inv}, \mathsf{evts}\rangle$, and $i \in \mathbb{N}$ such that $\pi_i = \langle \mathsf{s}, \mathsf{e}\rangle$, the definition of T implies that $\forall j > i \,.\, \forall k \in [i,j] \,.\, \exists \mathsf{s}_k \,.\, \mathsf{s}_i = \mathsf{s} \wedge \pi_k = \langle \mathsf{s}_k, \mathsf{e}\rangle$. $\max\{j - i \mid j > i \in \mathbb{N} \wedge \exists \pi \in \mathsf{T} \,.\, \forall k \in [i,j] \,.\, \exists \mathsf{s}_k \,.\, \mathsf{s}_i = \mathsf{s} \wedge \pi_k = \langle \mathsf{s}_k, \mathsf{e}\rangle\} \mathrel{\dot{\leqslant}} \mathsf{V}\langle \mathsf{s}, \mathsf{e}\rangle$. Q.E.D.

— By the Galois connection (34), we know that $\forall \mathsf{T} \in \mathbb{T}^{\mathrm{EvB}}\langle \mathsf{S}, \mathsf{E}\rangle \,.\, \mathsf{T} \subseteq \gamma_t \dot{\times} \gamma_v\langle \mathsf{S}, \mathsf{E}\rangle \circ \alpha_t \dot{\times} \alpha_v\langle \mathsf{S}, \mathsf{E}\rangle(\mathsf{T})$ where $\mathsf{T} \in \mathbb{T}\mathbb{r}^{\mathrm{EvB}}\langle \mathsf{S}, \mathsf{E}\rangle$.

To prove the converse, assume that $\pi \in \gamma_t \dot{\times} \gamma_v\langle \mathsf{S}, \mathsf{E}\rangle \circ \alpha_t \dot{\times} \alpha_v\langle \mathsf{S}, \mathsf{E}\rangle(\mathsf{T})$ where $\gamma_t \dot{\times} \gamma_v\langle \mathsf{S}, \mathsf{E}\rangle \circ \alpha_t \dot{\times} \alpha_v\langle \mathsf{S}, \mathsf{E}\rangle(\mathsf{T})$ is given by (32) and (33). We must prove that $\pi \in \mathsf{T}$ where $\mathsf{T} \in \mathbb{T}^{\mathrm{EvB}}\langle \mathsf{S}, \mathsf{E}\rangle$.

By (33), this means that we can assume that $\pi_0 \in \mathsf{init} \times \mathsf{evts} \wedge \forall i \in \mathbb{N} \,.\, \pi_i \in \mathsf{inv} \times \mathsf{evts} \wedge \forall \langle \mathsf{s}, \mathsf{e}\rangle = \pi_i, \langle \mathsf{s}', \mathsf{e}'\rangle = \pi_{i+1} \,.\, \langle \mathsf{s}, \mathsf{s}'\rangle \in \tau(\mathsf{e}) \wedge \mathsf{cvgt}\langle \mathsf{S}, \mathsf{E}\rangle(\pi, \mathsf{V})\}$ where $\langle \mathsf{inv}, \mathsf{evts}, \mathsf{init}, \tau, \mathsf{V}\rangle$ is defined by (32), that is (17). Assume, by reductio ad absurdum,

that $\pi \notin \mathsf{T}$. Then there is an earliest trace transition $\langle \mathsf{s},\, \mathsf{e}\rangle$ such that $\pi = \pi_1\langle \mathsf{s},\, \mathsf{e}\rangle\langle \mathsf{s}',\, \mathsf{e}'\rangle\pi_2 \notin \mathsf{T}$ with some $\pi_1\langle \mathsf{s},\, \mathsf{e}\rangle\langle \mathsf{s}'',\, \mathsf{e}''\rangle\pi_2' \in \mathsf{T}$.

We cannot have π_1 empty since then $\langle \mathsf{s},\, \mathsf{e}\rangle\langle \mathsf{s}'',\, \mathsf{e}''\rangle\pi_2' \in \mathsf{T}$ implies that s is an initial state so that, by definition of the initial states in (32), this does not prevent $\pi = \pi_1\langle \mathsf{s},\, \mathsf{e}\rangle\langle \mathsf{s}',\, \mathsf{e}'\rangle\pi_2 = \langle \mathsf{s},\, \mathsf{e}\rangle\langle \mathsf{s}',\, \mathsf{e}'\rangle\pi_2$ to belong to T.

It follows that π_1 must be not empty. In that case $\langle \mathsf{s},\, \mathsf{e}\rangle\langle \mathsf{s}',\, \mathsf{e}'\rangle$ is a valid transition while $\langle \mathsf{s},\, \mathsf{e}\rangle\langle \mathsf{s}',\, \mathsf{e}'\rangle$ is not in (32), meaning either that $\forall \pi \in \mathsf{T} \,.\, \forall i \in \mathbb{N} \,.\, \forall \mathsf{e}'' \in \mathsf{E} \,.\, \pi_i = \langle \mathsf{s},\, \mathsf{e}\rangle \Rightarrow \pi_{i+1} \neq \langle \mathsf{s}',\, \mathsf{e}''\rangle$, a contradiction or else $\neg \mathsf{cvgt}\langle \mathsf{S},\, \mathsf{E}\rangle(\pi, \boldsymbol{\lambda}\, \langle \mathsf{s},\, \mathsf{e}\rangle \bullet \max\{j - i \mid j > i \in \mathbb{N} \wedge \exists \pi \in \mathsf{T} \,.\, \forall k \in [i, j] \,.\, \exists \mathsf{s}_k \,.\, \mathsf{s}_i = \mathsf{s} \wedge \pi_k = \langle \mathsf{s}_k,\, \mathsf{e}\rangle\})$, another contradiction. By contradiction, we conclude that $\pi \in \mathsf{T}$. □