Property Preservation under Bisimulations on Markov Automata

Master's thesis

Sergey Sazonov
Matriculation number 314922
RWTH Aachen University, Germany
`sergey.sazonov(at)rwth-aachen.de`

December 17, 2013

1. Reviewer: Prof. Dr. Ir. Joost-Pieter Katoen
2. Reviewer: Apl. Prof. Dr. Thomas Noll

Statutory Declaration

I hereby declare that this thesis is my own work, that I used only the sources and resources cited in the thesis, and that I have identified citations as such.

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Aachen, December 17, 2013 _____________________________________

Sergey Sazonov

Contents

1 Introduction

Markov automata were introduced in 2010 by Eisentraut, Hermanns and Zhang [8]. The importance of this model lies in its generality: it supports nondeterminism, probabilistic choice and timing with exponential rates, as well as external and internal actions. Indeed, LTSs, DTMCs, CTMCs, PAs and IMCs are all special cases of Markov automata [7]. Furthermore, in [4] Eisentraut, Hermanns, Katoen and Zhang demonstrated that MAs can express the semantics of all GSPNs (by contrast, the previous approach with CTMCs is only able to handle the so-called well-defined GSPNs).

Even though MAs are a young formalism, several significant results have already been established. In [8, 7] Eisentraut et al. defined strong, naive weak and weak bisimulations for MAs. Moreover, [7] introduced the notion of parallel composition on MAs and demonstrated that various flavours of parallel composition on LTLs, DTMCs, CTMCs, PAs and IMCs can be reduced to the new operator on MAs. In [14, 13], Timmer et al. introduced a process algebra for MAs, as well as the notions of branching bisimulation (a kind of weak bisimulation on the states of an MA) and confluence reduction (a reduction technique similar to partial order reduction). In [9] Guck et al. defined several properties of the states of an MA, of which minimum expected time and minimum long run average are of particular importance. They also presented algorithms to compute the aforementioned properties. Other notable publications include [5, 6, 2].

1.1 Contribution of the thesis

In the present work, we continue the research into the properties of the various bisimulation relations defined by [7, 8, 13]. In particular, we set out to answer the question whether minimum expected time and minimum long run average [9] are preserved under some or all of the bisimulations. In the process, we also introduce the notions of the bisimulation quotient, quotient schedulers and computations, and prove several related theorems.

1.2 Outline of the thesis

- In Chapter 2, we introduce preliminary definitions and notation, in most cases consistent with [7, 8, 9, 14]. In particular, we introduce Markov automata, related probability spaces and measures, and, most importantly, several kinds of bisimulations and state properties.
- In Chapter 3, we draw on the ideas of Martin Neuhäußer [11] and define quotient automata under strong bisimulation, as well as quotient schedulers. We then prove that quotient schedulers preserve the probability of certain sets of paths, called bisimulation-closed sets.
- In Chapters 4 and 5, we use the results of Guck et al. [9], as well as those in Chapter 3, to prove that minimum expected time and minimum long run average are preserved under strong bisimulation.

- In Chapter 6, we introduce computations on Markov automata (loosely based on the ideas of Desharnais et al. in [3]). We prove that there exists a close connection between time-abstract schedulers and computations. We also define expected time and long run average in computations, thus enabling us to use computations to prove results about these properties.
- In Chapter 7, we use the results of Chapter 6 to prove that minimum expected time and minimum long run average are preserved under branching and naive weak bisimulations.
- In Chapter 8, we again use the techniques developed in Chapter 6 to investigate the preservation of minimum expected time and minimum long run average under weak bisimulation. We first prove that weak bisimulation preserves minimum long run average. Then we present a counterexample demonstrating that weak bisimulation does *not* preserve minimum expected time in general. Finally, we show that weak bisimulation preserves minimum expected time under certain conditions.

2 Preliminaries

2.1 Distributions

The terminology and notation described in this section are taken from [7, 8, 9, 14] without significant changes.

Definition 2.1 (Subdistributions). A *subdistribution* μ over a countable set S is a function

$$\mu : S \mapsto [0, 1]$$

such that

$$\sum_{s \in S} \mu(s) \quad \leq \quad 1.$$

The *support* of μ is the set

$$\mathit{Supp}(\mu) \quad = \quad \{s \in S \mid \mu(s) > 0\}.$$

For every $S' \subseteq S$, let

$$\mu(S') \quad = \quad \sum_{s \in S'} \mu(s).$$

Let $|\mu| = \mu(S)$ denote the *size* of the subdistribution μ. We say that μ is a *full distribution*, or simply a *distribution*, if $|\mu| = 1$.

In order to specify a subdistribution explicitly, we employ the set notation. For example,

$$\mu \quad = \quad \{(s, 0.6), (t, 0.4)\}$$

denotes the distribution such that $\mu(s) = 0.6$ and $\mu(t) = 0.4$.

Finally, let $\mathit{Dist}(S)$ and $\mathit{Subdist}(S)$ be the sets of distributions and subdistributions over S, respectively.

Definition 2.2 (Operations on subdistributions). Let μ and μ' be subdistributions. We define their *sum* as

$$(\mu \oplus \mu')(s) \quad = \quad \mu(s) + \mu'(s),$$

provided that $\sum_{s \in S} (\mu(s) + \mu'(s)) \leq 1$.

Furthermore, for every $a \in \mathbb{R}_{\geq 0}$ such that $a \cdot |\mu| \leq 1$, we let $a \cdot \mu$ (or simply $a\mu$) denote the subdistribution defined by

$$(a\mu)(s) \quad = \quad a \cdot \mu(s) \, .$$

Finally, for every $s \in S$, we define the subdistribution $\mu - s$ as

$$(\mu - s)(s') \quad = \quad \begin{cases} \mu(s') & s \neq s' \\ 0 & s = s' . \end{cases}$$

Definition 2.3 (Quotient subdistribution). Let μ be a subdistribution over S, and let $\mathcal{R}$ be an equivalence relation on S. Then $\mu/\mathcal{R}$ is a subdistribution over $S/\mathcal{R}$ such that for every $C \in S/\mathcal{R}$,

$$(\mu/\mathcal{R})(C) \quad = \quad \mu(C) \, .$$

Definition 2.4 (Subdistribution equivalence). Let μ and μ' be subdistributions over S, and $\mathcal{R}$ an equivalence relation on S. Then, μ is equivalent to μ' with respect to $\mathcal{R}$, written $\mu \equiv_{\mathcal{R}} \mu'$, if $\mu/\mathcal{R} = \mu'/\mathcal{R}$.

2.2 Markov automata

In this section we follow the standard approach of [7, 8, 9, 14] to define Markov automata and closely related concepts.

Informally speaking, a Markov automaton (MA) consists of a finite set of states and a finite set of labelled transitions of two kinds: immediate (also called probabilistic) and timed (also called Markovian). An example of an MA can be seen in Figure 1 (immediate transitions are depicted with single arrowheads, timed transitions - with double arrowheads). An immediate transition is labelled with an action: α, β, etc., while a timed transition is labelled with a transition rate, which is a positive real number.

The semantics of immediate transitions is simple: upon entering a state with outgoing transitions of this kind, a transition to be executed is chosen by a scheduler (schedulers are considered in Section 2.3) and executed without delay. Note also that an immediate transition leads not to a single state, but rather to a distribution over the states, which is the principal difference between MAs and IMCs.

The semantics of timed transitions is slightly more complex. First of all, the exit rate of a state is the sum of the rates of the outgoing timed transitions. Then, upon entering a state with outgoing timed transitions, we first delay for an amount of time distributed exponentially (the parameter of the exponential distribution is given by the exit rate of the state), and then execute one of the outgoing timed transitions. In the last step, the probability of choosing the particular timed transition is proportional to the rate of this transition.

This informal explanation is formalized in this and the next sections.

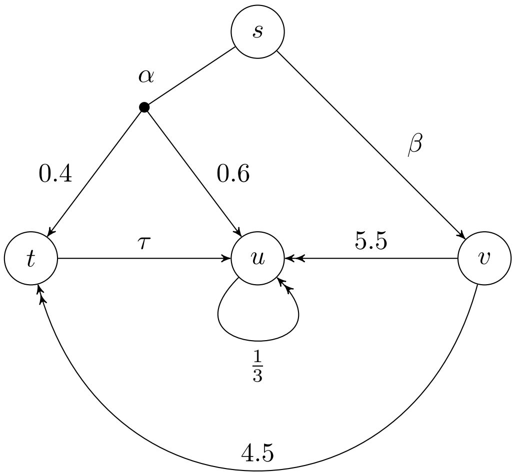


Figure 1: A Markov automaton.

Definition 2.5 (Markov automaton). A *Markov automaton* (MA) is a tuple $MA = (S, Act, \mapsto, \Rightarrow)$, where

- S is a finite nonempty set of *states*,
- Act is a finite nonempty set of *actions* containing the internal action τ,
- $\mapsto \subseteq S \times Act \times Dist\,(S)$ is a finite set of *immediate transitions*, and
- $\Rightarrow \subseteq S \times \mathbb{R}_{>0} \times S$ is a finite set of *timed transitions*.

Sometimes we also designate one of the states as the initial state, or specify an initial probability distribution over S.

For each state $s \in S$, let:

- $IT\,(s) = \{(s, \alpha, \mu) \in \{s\} \times Act \times Dist\,(S) \mid (s, \alpha, \mu) \in \mapsto\}$ denote the set of outgoing immediate transitions from s,
- $TT\,(s) = \{(s, \lambda, s') \in \{s\} \times \mathbb{R}_{>0} \times S \mid (s, \lambda, s') \in \Rightarrow\}$ denote the set of outgoing timed transitions from s, and
- $Act\,(s) = \{\alpha \in Act \mid \exists \mu \in Dist\,(S)\,.\,(s, \alpha, \mu) \in \mapsto\}$ denote the set of actions enabled in s.

Finally, we impose the following requirements on the transition relation:

1. for each state $s \in S$ such that $\tau \in Act(s)$, it must hold that $TT(s) = \emptyset$,
2. for each state $s \in S$ and action $\alpha \in Act$, there is only one outgoing transition labelled with α, i.e.

$$|\{(s, \alpha, \mu) \in \{s\} \times \{\alpha\} \times Dist(S) \mid (s, \alpha, \mu) \in \mapsto\}| \quad \leq \quad 1,$$

3. none of the states of MA are absorbing, i.e., for every state $s \in S$,

$$|IT(s) \cup TT(s)| \quad > \quad 0$$

   (note that this is no restriction, since every absorbing state can be supplied with a timed self-loop transition, allowing time to pass indefinitely without leaving the state).

A few remarks on Definition 2.5 are in order:

1. In literature, requirement 1 is referred to as the *maximal progress assumption*.
2. Requirement 2 is necessary because of the way we define schedulers (see Section 2.3): in particular, a scheduler returns a probability distribution over the actions, so that if multiple outgoing transitions with the same label were allowed, non-determinism would not be resolved completely.
3. Requirement 3 is not essential but makes working with MAs easier since all maximal paths are now infinite.
4. Finally, note that an initial state or distribution is not included into the definition of an MA. We find it convenient to specify an initial state separately when needed, because in the context of state properties it is often required to consider each state as initial.

We now formally introduce exit rates and other related concepts.

Definition 2.6 (Exit rates). Let $MA = (S, Act, \mapsto, \Rightarrow)$ be an MA. For each state $s \in S$ such that $TT(s) \neq \emptyset$, let

$$E(s) \quad = \quad \sum_{s' \in S} \sum_{(s, \lambda, s') \in TT(s)} \lambda$$

be the *exit rate* of s.

Definition 2.7 (Markovian action). We define

$$Act^{\mathsf{T}} \quad = \quad Act \, \cup \, \{\mathsf{T}\},$$

with T denoting a special action corresponding to a timed transition.

For each state $s \in S$, let

$$Act^{\mathsf{T}}(s) \quad = \quad \begin{cases} Act(s) & \text{if } TT(s) = \emptyset \\ Act(s) \cup \{\mathsf{T}\} & \text{otherwise.} \end{cases}$$

Definition 2.8 (Extended action set). The *extended action set* is defined as

$$Act^{\chi} \quad = \quad Act \cup \{\delta(r) \mid r \in \mathbb{R}_{>0}\}.$$

Furthermore, for every $s \in S$, let

$$Act^{\chi}(s) \quad = \quad \begin{cases} Act(s) & \text{if } TT(s) = \emptyset \\ Act(s) \cup \{\delta(E(s))\} & \text{otherwise.} \end{cases}$$

Definition 2.9 (Distributions). Let $MA = (S, Act, \mapsto, \Rightarrow)$ be an MA.

For each state $s \in S$, $\alpha \in Act$ and $\mu \in Dist(S)$, whenever there is a transition $s \overset{\alpha}{\longmapsto} \mu$, we set

$$\mu_{s,\alpha} \quad = \quad \mu.$$

Next, for every $s \in S$ with $TT(s) \neq \emptyset$, let $\mu_{s,\mathsf{T}}$ be the distribution over S defined by

$$\mu_{s,\mathsf{T}}(s') \quad = \quad \frac{\displaystyle\sum_{(s,\lambda,s') \in TT(s)} \lambda}{E(s)}.$$

Then, for every $s \in S$ with $TT(s) \neq \emptyset$, let

$$\mu_{s,\delta(E(s))} \quad = \quad \mu_{s,\mathsf{T}}.$$

Note that $\mu_{s,\alpha}$ is now defined for every $s \in S$ and $\alpha \in Act^{\mathsf{T}}(s) \cup Act^{\chi}(s)$.

Finally, we set $\mu_{s,\alpha}(s') = 0$ whenever $\alpha \notin Act^{\mathsf{T}}(s) \cup Act^{\chi}(s)$.

Let us now change the perspective on timed transitions. In particular, until now we have spoken about multiple timed transitions from a state, each leading to a single other state. Note, however, that conceptually those transitions represent a single transition to a distribution over the states. For instance, the delay distribution is governed not by the rate of any particular transition, but rather by the sum of the rates of all the timed transitions from the state. Similarly, the next state is chosen from a distribution defined by all the transitions together. It is, therefore, quite natural to convert the timed transitions from the given state to a single *combined* transition (see Figure 2). In the following we will label such transitions with $\delta\left(E\left(s\right)\right)$, which is consistent with Definition 2.8. Now we have two perspectives on timed transitions. In what follows, it will usually be clear which one is being used. In a situation where any confusion is possible, we will refer to individual timed transitions and combined timed transitions.

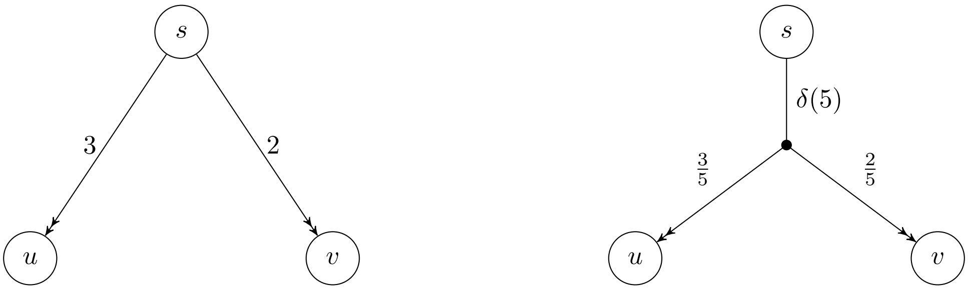


Figure 2: A state with several timed transitions (left) and the corresponding combined timed transition (right).

Note also that the two perspectives are completely interchangeable: specifying the rates of the individual timed transitions is equivalent to specifying the exit rate of the state and a target distribution over the states.

To summarize, the new perspective on the transition relation of an MA can be described as follows:

1. An immediate transition is labelled with an action (α, τ, etc.), executed without delay, and leads to a probability distribution over the states.
2. A timed transition is labelled with $\delta\left(r\right)$, where $r \in \mathbb{R}_{>0}$ is called the exit rate of the state, and leads to a distribution over the states. Such a transition is executed after a probabilistic delay distributed exponentially with the rate r.
3. There can be at most one timed transition from a state.

The new perspective is especially useful in the context of bisimulations (where the simulated transition is either an immediate transition or a combined timed transition). In fact, sometimes we don't even care about the type of the transition. We, therefore, introduce unified notation for both kinds of transitions.

Definition 2.10 (Unified transition relation). Let $s \in S$ and $\alpha \in Act^{\chi}$. We write

$$s \xrightarrow{\alpha} \mu$$

if $\alpha \in Act^{\chi}(s)$ and $\mu = \mu_{s,\alpha}$.

It is important to remember that a unified transition denotes either an immediate transition or a *combined* timed transition.

The rest of this section is devoted to additional assumptions we make about Markov automata. In particular, in many circumstances we wish to make sure that no state has both kinds of outgoing transitions. We call such an MA *semi-closed*, for the following reason: an MA is *closed* if all probabilistic transitions are labelled with τ; then, the maximal progress assumption makes sure that no state has both kinds of outgoing transitions. We would like to enforce the last requirement while keeping actions other τ. To the best of our knowledge, there is no established term for this class of Markov automata. We decided to call such MAs semi-closed.

Definition 2.11 (Semi-closed Markov automaton). A Markov automaton $MA = (S, Act, \mapsto, \Rightarrow)$ is *semi-closed* if for each state $s \in S$ it holds that $TT(s) \neq \emptyset$ implies $IT(s) = \emptyset$. In this case we define

- the set of *Markovian states* to be $MS = \{s \in S \mid TT(s) \neq \emptyset\}$, and
- the set of *probabilistic states* to be $PS = S \setminus MS$.

Note that if an MA is semi-closed, then for every $s \in S$ it holds that:

- either s has one or more outgoing immediate transitions,
- or s has *exactly one* outgoing *combined* timed transition.

At this point, it is important to note that in this paper we will sometimes work with general Markov automata and sometimes make the assumption that the MA is semi-closed. At the beginning of each chapter (or individual section) the reader will find a statement such as this: “let $MA = (S, Act, \mapsto, \Rightarrow)$ be an MA”, in which case we make *no assumptions* about Markov automata, or like this: “let $MA = (S, Act, \mapsto, \Rightarrow)$ be semi-closed”.

Furthermore, whenever an MA is assumed to be semi-closed, it is also assumed to be non-Zeno.

Definition 2.12 (Zenoness). A semi-closed Markov automaton $MA = (S, Act, \mapsto, \Rightarrow)$ is said to be *Zeno* if it contains a strongly connected component consisting of only probabilistic states.

For the rest of this paper all *semi-closed* Markov automata are assumed to be non-Zeno.

2.3 Paths and schedulers

In this section we cover the semantics of Markov automata: paths, schedulers and probability spaces. Brief descriptions of these concepts for MAs can be found in [9, 14]. Much more detailed treatments of the same topic in relation to such models as IMCs and CTMDPs can be found in [11, 17, 16].

Throughout this section, let $MA = (S, Act, \mapsto, \Rightarrow)$ be an MA.

Definition 2.13 (Paths). A *finite path* of length $n \in \mathbb{N}_{\geq 0}$ is a tuple

$$\pi = (s_0, \alpha_0, t_0, s_1, \alpha_1, ..., s_n) \quad \in \quad S \times Act^{\top} \times \mathbb{R}_{\geq 0} \times S \times Act^{\top} \times ... \times S,$$

usually written as

$$\pi \quad = \quad s_0 \xrightarrow{\alpha_0, t_0} s_1 \xrightarrow{\alpha_1, t_1} ... \xrightarrow{\alpha_{n-1}, t_{n-1}} s_n.$$

Let $|\pi| = n$ denote the length of π.

Note that the value t_i denotes the amount of time the path spends in the state s_i before moving to s_{i+1}.

We impose the following restriction on the values $t_0, ..., t_{n-1}$: for each $i \in \{0, ..., n-1\}$ it must be the case that $\alpha_i = \top$ implies $t_i > 0$. In other words, upon entering a state, a path is not allowed to execute a timed transition immediately, but must first wait for some positive amount of time.

We use the notation $\pi@t$ to refer to the sequence of states traversed by the path at time t (note that there might be more than one such state because of instantaneous probabilistic transitions).

Next, for every $i \in \{0, ..., |\pi|\}$, let $\pi[i]$ denote the ith state on π, and let $\pi[0...i]$ denote the *prefix*

$$\pi[0...i] \quad = \quad s_0 \xrightarrow{\alpha_0, t_0} s_1 \xrightarrow{\alpha_1, t_1} ... \xrightarrow{\alpha_{i-1}, t_{i-1}} s_i.$$

Furthermore, for every finite path π, let $\pi_{\downarrow}$ refer to $\pi[|\pi|]$.

The notion of finite paths is extended to infinite paths in the obvious manner. Let $Paths^n$, $Paths^*$, and $Paths^{\omega}$ denote the sets of finite paths of length n, all finite paths, and all infinite paths in MA, respectively.

Definition 2.14 (Time-abstract paths). A *finite time-abstract path* of length $n \in \mathbb{N}_{\geq 0}$ is a tuple

$$\pi = (s_0, \alpha_0, s_1, \alpha_1, ..., s_n) \quad \in \quad S \times Act^{\top} \times S \times Act^{\top} \times ... \times S,$$

usually written as

$$\pi \quad = \quad s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} ... \xrightarrow{\alpha_{n-1}} s_n.$$

We define $|\pi|$, $\pi\,[i]$, $\pi\,[0...i]$ and $\pi_{\downarrow}$ in the same way as for finite paths.

Furthermore, let

$$\Pi_\pi \quad = \quad \{s_0\} \times \{\alpha_0\} \times \mathbb{R}_{\geq 0} \times ... \times \{s_n\}$$

be the set of finite paths corresponding to π.

Let $Paths^n_{abs}$, $Paths^*_{abs}$, and $Paths^\omega_{abs}$ denote the sets of finite time-abstract paths of length n, all finite time-abstract paths, and all infinite time-abstract paths in MA, respectively.

For a path $\pi \in Paths^n$ and a finite time-abstract path $\pi_{abs} \in Paths^n_{abs}$, we say that $\pi \in \pi_{abs}$ if $\pi \in \Pi_{\pi_{abs}}$.

Definition 2.15 (Events over paths). We first define the σ-field of combined transitions as follows:

$$\mathfrak{F} \quad = \quad \sigma\left(\left\{(T \times A \times S) \mid T \in \mathfrak{B}\left(\mathbb{R}_{\geq 0}\right) \wedge A \in 2^{Act^\tau} \wedge S \in 2^S\right\}\right).$$

Then, for each $n \in \mathbb{N}_{\geq 0}$, the σ-field over $Paths^n$ is defined as

$$\mathfrak{F}_{Paths^n} \quad = \quad \sigma\left(\left\{S_0 \times M_1 \times ... \times M_n \mid S_0 \in 2^S \wedge \left(\forall i \in \{1, ..., n\}\ .\ M_i \in \mathfrak{F}\right)\right\}\right).$$

The elements of $\mathfrak{F}_{Paths^n}$ are called *measurable sets of finite paths* of length n.

Furthermore, for every $\Pi \in \mathfrak{F}_{Paths^n}$, the *cylinder* of Π is

$$Cyl\,(\Pi) \quad = \quad \{\pi \in Paths^\omega \mid \pi\,[0...n] \in \Pi\}.$$

Next, we set

$$\mathfrak{F}_{Paths^\omega} \quad = \quad \sigma\left(\bigcup_{n=0}^{\infty} \{Cyl\,(\Pi) \mid \Pi \in \mathfrak{F}_{Paths^n}\}\right).$$

Finally, $\mathfrak{F}_{Paths}$ is the standard σ-algebra generated by the disjoint union

$$\biguplus_{n=0}^{\infty} \mathfrak{F}_{Paths^n} \quad \uplus \quad \mathfrak{F}_{Paths^\omega}.$$

Definition 2.16 (Events over time-abstract paths). For each $n \in \mathbb{N}_{\geq 0}$, the σ-field over $Paths^n_{abs}$ is simply

$$\mathfrak{F}_{Paths^n_{abs}} \quad = \quad 2^{Paths^n_{abs}}.$$

Furthermore, for every $\Pi \in \mathfrak{F}_{Paths^n_{abs}}$, the *cylinder* of Π is

$$Cyl\left(\Pi\right) \quad = \quad \left\{\pi \in Paths^{\omega}_{abs} \mid \pi\left[0...n\right] \in \Pi\right\}.$$

Next, we set

$$\mathfrak{F}_{Paths^{\omega}_{abs}} \quad = \quad \sigma\left(\bigcup_{n=0}^{\infty}\left\{Cyl\left(\Pi\right) \mid \Pi \in \mathfrak{F}_{Paths^n_{abs}}\right\}\right).$$

Finally, $\mathfrak{F}_{Paths_{abs}}$ is the standard σ-algebra generated by the disjoint union

$$\biguplus_{n=0}^{\infty}\mathfrak{F}_{Paths^n_{abs}} \quad \uplus \quad \mathfrak{F}_{Paths^{\omega}_{abs}}.$$

Definition 2.17 (Measurable rectangles). A *measurable rectangle* Π of length $n \in \mathbb{N}_{\geq 0}$ is a measurable subset of $Paths^n$ of the form

$$\Pi \quad = \quad S_0 \times A_0 \times T_0 \times ... \times A_{n-1} \times T_{n-1} \times S_n,$$

where:

1. for every $i \in \{0, ..., n\}$, $S_i \in 2^S$, and
2. for every $i \in \{0, ..., n-1\}$, $A_i \in 2^{Act^{\top}}$ and $T_i \in \mathfrak{B}\left(\mathbb{R}_{\geq 0}\right)$.

For the rest of this section we assume the Markov automaton MA to be semi-closed.

We now introduce schedulers and probability measures. Note that for simplicity we require a scheduler to accept every finite path, even if it ends in a Markovian state. Of course, in the latter case, the only action any scheduler can choose is $\top$.

Definition 2.18 (Generic measurable scheduler). A *generic scheduler* on a semi-closed Markov automaton is a mapping

$$D : Paths^{*} \times Act^{\top} \mapsto [0, 1]$$

such that for all $\pi \in Paths^{*}$,

$$D\left(\pi, \cdot\right) \quad \in \quad Dist\left(Act^{\top}\left(\pi_{\downarrow}\right)\right).$$

For every $\pi \in Paths^*$ and $A \in 2^{Act^\intercal}$, we set

$$D(\pi, A) \quad = \quad \sum_{\alpha \in A} D(\pi, \alpha).$$

A generic scheduler D is *measurable* if for all $A \in 2^{Act^\intercal}$, $D^{-1}(A) : Paths^* \mapsto [0,1]$ is measurable. We assume all schedulers to be measurable. We denote the set of all generic measurable schedulers on MA by GM.

Definition 2.19 (Stationary scheduler). A scheduler $D \in GM$ is said to be *stationary* if for all $\alpha \in Act^\intercal$ and for all finite paths $\pi, \pi' \in Paths^*$, $\pi_\downarrow = \pi'_\downarrow$ implies $D(\pi, \alpha) = D(\pi', \alpha)$.

Note that in this case, the decision of the scheduler depends only on the last state on the path. We, therefore, can regard such a scheduler as a function

$$D : S \times Act^\intercal \mapsto [0,1]$$

and write $D(s, \alpha)$ instead of $D(\pi, \alpha)$.

Definition 2.20 (Deterministic scheduler). A scheduler $D \in GM$ is said to be *deterministic* if for every finite path $\pi \in Paths^*$ there exists an action $\alpha_\pi \in Act^\intercal(\pi_\downarrow)$ such that $D(\pi, \alpha_\pi) = 1$.

Definition 2.21 (Time-abstract scheduler). A scheduler $D \in GM$ is said to be *time-abstract* if for every $\pi_1, \pi_2 \in Paths^*$, $\pi_{abs} \in Paths^*_{abs}$ and $\alpha \in Act^\intercal$,

$$\pi_1 \in \pi_{abs} \wedge \pi_2 \in \pi_{abs} \quad \text{implies} \quad D(\pi_1, \alpha) = D(\pi_2, \alpha).$$

We then regard such a scheduler as a function

$$D : Paths^*_{abs} \times Act^\intercal \mapsto [0,1]$$

and write $D(\pi_{abs}, \alpha)$ instead of $D(\pi_1, \alpha)$ or $D(\pi_2, \alpha)$.

Definition 2.22 (Unified transition probability densities). For each $s \in S$, we define the density function

$$\eta_s : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}_{\geq 0}$$

as follows:

$$\eta_s(t) \quad = \quad \begin{cases} E(s)\, e^{-E(s)t} & \text{if } s \in MS \\ \delta_{Dirac}(t) & \text{if } s \in PS, \end{cases}$$

where δ_{Dirac} is the Dirac delta function. We then use the common convention that

$$\int_T f(t)\,\delta_{Dirac}(dt) \quad = \quad \begin{cases} f(0) & \text{if } 0 \in T \\ 0 & \text{otherwise,} \end{cases}$$

so that we can use unified notation for both Markovian and probabilistic states.

Definition 2.23 (Probability measures over paths). Let μ_0 be an initial distribution over S, and let $D \in GM$.

For each $n \in \mathbb{N}_{\geq 0}$, we define the probability measure

$$\underset{\mu_0,D}{\overset{n}{\Pr}} : \mathfrak{F}_{Paths^n} \mapsto [0,1]$$

on $(Paths^n, \mathfrak{F}_{Paths^n})$ as follows:

$$\underset{\mu_0,D}{\overset{n}{\Pr}}(\Pi) \quad = \quad \sum_S \sum_{Act^{\tau}} \int_{\mathbb{R}_{\geq 0}} \sum_S \ldots \sum_S \left[I_{\Pi}(\pi) \cdot \mu_0(s_0) \prod_{i=0}^{n-1} \big(D(\pi[0 \ldots i], \alpha_i) \mu_{s_i,\alpha_i}(s_{i+1}) \big) \right] \eta_{s_{n-1}}(dt_{n-1}) \ldots \eta_{s_0}(dt_0),$$

where:

1. the finite path

   $$\pi \quad = \quad s_0 \xrightarrow{\alpha_0, t_0} s_1 \xrightarrow{\alpha_1, t_1} \ldots \xrightarrow{\alpha_{n-1}, t_{n-1}} s_n$$

   is defined by the variables of summation and integration, and
2. $I_{\Pi}(\pi) = 1$ if $\pi \in \Pi$ and $I_{\Pi}(\pi) = 0$ otherwise.

Now, the probability measure

$$\underset{\mu_0,D}{\overset{\omega}{\Pr}} : \mathfrak{F}_{Paths^\omega} \mapsto [0,1]$$

on $(Paths^\omega, \mathfrak{F}_{Paths^\omega})$ is obtained by defining it on every cylinder as follows:

$$\underset{\mu_0,D}{\overset{\omega}{\Pr}}(Cyl(\Pi)) \quad = \quad \underset{\mu_0,D}{\overset{|\Pi|}{\Pr}}(\Pi),$$

and then extending it to every $\Pi \in \mathfrak{F}_{Paths^\omega}$.

In the rest of this work, we will omit the superscript n or ω and simply write $\Pr_{\mu_0,D}$ instead of $\Pr^n_{\mu_0,D}$ and $\Pr^\omega_{\mu_0,D}$. Moreover, if $\mu_0 = \{(s,1)\}$ for some $s \in S$, we denote the corresponding probability measure by $\Pr_{s,D}$.

Definition 2.24 (Probability measures over time-abstract paths). Let μ_0 be an initial distribution over S, and let $D \in GM$.

For each $n \in \mathbb{N}_{\geq 0}$, we define the probability measure

$$\Pr_{\mu_0,D}^{n} : \mathfrak{F}_{Paths_{abs}^{n}} \mapsto [0,1]$$

on $\left(Paths_{abs}^{n}, \mathfrak{F}_{Paths_{abs}^{n}}\right)$ as follows:

$$\Pr_{\mu_0,D}^{n}(\mathfrak{P}) = \sum_{\pi \in \mathfrak{P}} \Pr_{\mu_0,D}^{n}(\Pi_\pi),$$

where the probability measure $\Pr_{\mu_0,D}^{n}$ on the right side is the probability measure on $(Paths^n, \mathfrak{F}_{Paths^n})$.

Now, the probability measure

$$\Pr_{\mu_0,D}^{\omega} : \mathfrak{F}_{Paths_{abs}^{\omega}} \mapsto [0,1]$$

on $\left(Paths_{abs}^{\omega}, \mathfrak{F}_{Paths_{abs}^{\omega}}\right)$ is obtained by defining it on every cylinder as follows:

$$\Pr_{\mu_0,D}^{\omega}(Cyl(\Pi)) = \Pr_{\mu_0,D}^{|\Pi|}(\Pi),$$

and then extending it to every $\Pi \in \mathfrak{F}_{Paths_{abs}^{\omega}}$.

As for paths, we usually omit the superscript n or ω and simply write $\Pr_{\mu_0,D}$ instead of $\Pr_{\mu_0,D}^{n}$ and $\Pr_{\mu_0,D}^{\omega}$. Moreover, if $\mu_0 = \{(s,1)\}$ for some $s \in S$, we denote the corresponding probability measure by $\Pr_{s,D}$.

Lemma 2.25 (Probabilities of finite time-abstract paths under time-abstract scheduer). *Let $D \in GM$ be a time-abstract scheduler, and let $\mu_0 \in Dist(S)$ be an initial distribution over S. Then, for every time-abstract path*

$$\pi = s_0 \xrightarrow{\alpha_0} .... \xrightarrow{\alpha_{n-1}} s_n$$

it holds that

$$\Pr_{\mu_0,D}(\pi) = \mu_0(s_0) \cdot \prod_{i=0}^{n-1} \left(D(\pi[0...i], \alpha_i) \cdot \mu_{s_i,\alpha_i}(s_{i+1})\right).$$

2.4 Strong bisimulation

In this section we merely restate the definitions and results from [7, 8].

Throughout this section, let $MA = (S, Act, \mapsto, \Rightarrow)$ be an MA.

Definition 2.26 (Strong bisimulation). Let $\mathcal{R}$ be an equivalence relation on S. Then, $\mathcal{R}$ is a *strong bisimulation* on S if for all $(s, s') \in \mathcal{R}$, $\alpha \in Act^{\chi}$ and $\mu \in Dist\,(S)$, $s \xrightarrow{\alpha} \mu$ implies $s' \xrightarrow{\alpha} \mu'$ for some μ' such that $\mu \equiv_{\mathcal{R}} \mu'$.

Two states s and s' are strongly bisimilar, written $s \sim s'$, if (s, s') is contained in some strong bisimulation.

Note that the definition of strong bisimulation is nothing more than the standard definition of Segala and Lynch [12] applied to Markov automata.

Lemma 2.27 (Coarsest bisimulation relation). *The relation $\sim$ is the coarsest strong bisimulation relation on S.*

Definition 2.28 (Direct sum of Markov automata). Let $MA_1 = (S_1, Act_1, \mapsto_1, \Rightarrow_1)$ and $MA_2 = (S_2, Act_2, \mapsto_2, \Rightarrow_2)$ be Markov automata. Then their *direct sum* is defined as

$$MA_1 \oplus MA_2 \quad = \quad (S_1 \uplus S_2, Act_1 \cup Act_2, \mapsto_1 \uplus \mapsto_2, \Rightarrow_1 \uplus \Rightarrow_2)\,.$$

Definition 2.29 (Strong bisimulation on Markov automata). Let MA_1 and MA_2 be Markov automata whose initial distributions are $\mu_{0,1}$ and $\mu_{0,2}$, respectively. We say that MA_1 and MA_2 are bisimilar, denoted by $MA_1 \sim MA_2$, if $\mu_{0,1} \equiv_{\sim} \mu_{0,2}$ in $MA_1 \oplus MA_2$.

2.5 Weak bisimulations

Here, again, we closely follow [7, 8]. The notion of probabilistic weak bisimulation is considerably more complex than both ordinary weak bisimulation and probabilistic strong bisimulation. In the non-stochastic case, a transition is simulated by a sequence of transitions, all of which except one are internal (i.e., labelled with τ). In the case of MAs, however, the target of a transition is not a state but a distribution, so that we naturally come to the idea of a *tree* of transitions. By analogy with ordinary weak bisimulation, on every maximal path in this tree all but one transition must be internal.

The formalization of transition trees given in [7, 8] is fairly complex (it is presented below), but behind it lies a very simple idea: a *transition tree* a simply a tree whose nodes correspond to states in MA and whose links correspond to transitions from those states (see Figure 3). Each transition in the tree corresponds to either an immediate transition or to a *combined* timed transition in the MA. It is a rule that each node of the tree is allowed to have at most one outgoing transition.

It is quite obvious that we can associate a probability with each node of a transition tree. For example, in Figure 3, the nodes of the tree corresponding to the states s and t each has the

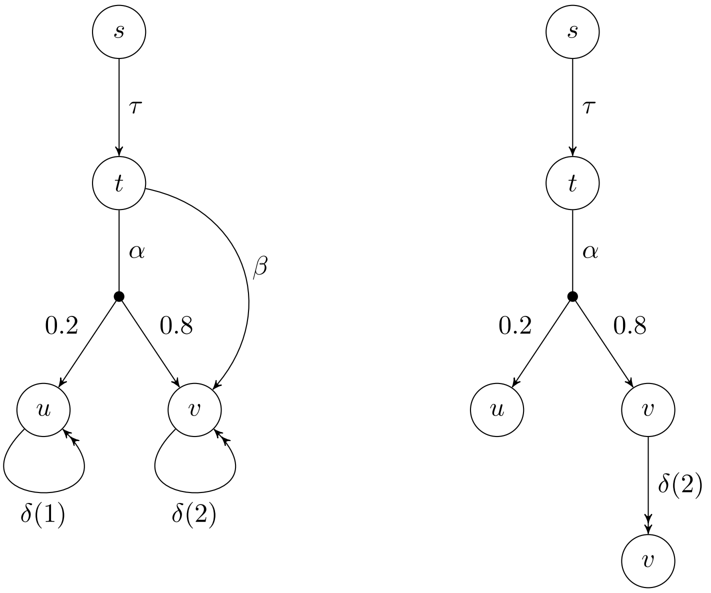


Figure 3: A Markov automaton (left) and a transition tree in it (right).

probability of 1, the node corresponding to u has the probability of 0.2, and each of the two nodes corresponding to v has the probability of 0.8. Then, it is also clear that the leaves of the tree induce a subdistribution over the states. For instance, the tree in Figure 3 induces the distribution $\{(u, 0.2), (v, 0.8)\}$.

Next, suppose α is a transition label (i.e. either an action or $\delta(r)$) such that $\alpha \neq \tau$. We call a transition tree T an α-tree if on every path from the root to a leaf there is *exactly one* non-τ transition, and this transition is labelled with α (see Figure 4 for an example). T is a τ-tree if *all* the transitions are labelled with τ (see Figure 5).

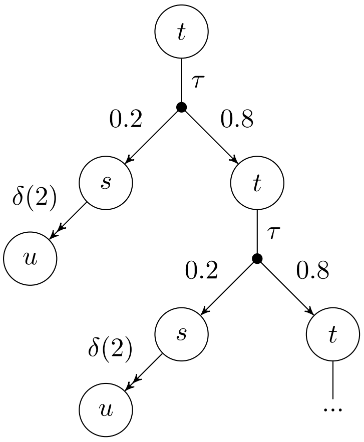


Figure 4: An infinite $\delta(2)$-tree.

Now, we introduce finite convex combinations of transition trees. A finite convex combination of α-trees (where α can be *any* label, including τ) is simply a finite weighted collection of α-trees from the same state, such that the weights sum up to unity (see Figure 6). Such a combination induces a subdistribution over the states in the obvious way. For example, the finite convex combination

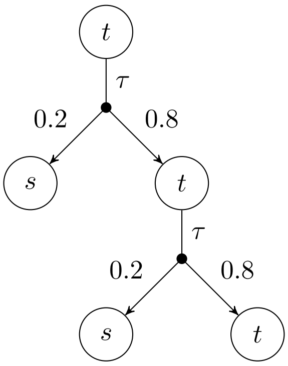


Figure 5: A τ-tree.

in Figure 6 induces the distribution $\{(u, 0.55), (v, 0.45)\}$.

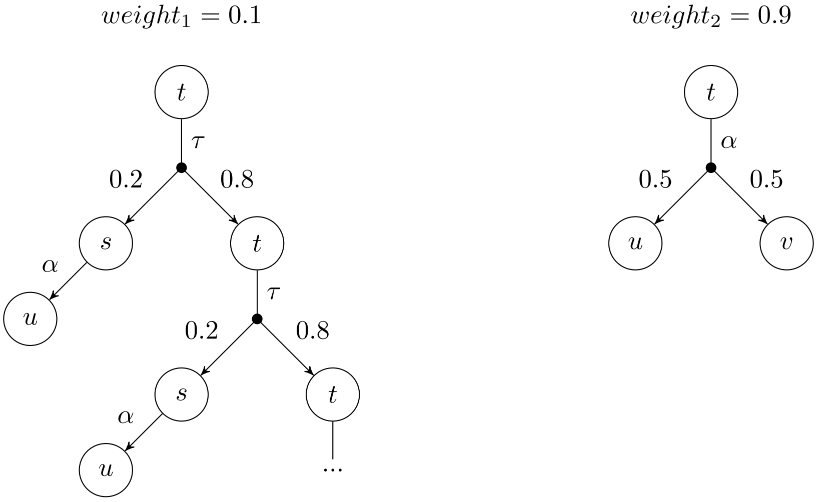


Figure 6: A finite convex combination of α-trees.

We say that a state $s \in S$ can execute an *weak α-transition* to a distribution $\mu \in \mathit{Dist}(S)$, denoted $s \overset{\alpha}{\hookrightarrow}_{\oplus} \mu$, if there exists a finite convex combination of α-trees from s inducing μ.

We can now define the first kind of weak bisimulation by adapting the definition of strong bisimulation (Definition 2.26) in the following way: when simulating a transition $s \xrightarrow{\alpha} \mu$, the state s' executes a *weak* transition $s' \overset{\alpha}{\hookrightarrow}_{\oplus} \mu'$ (as opposed to an ordinary transition in the case of strong bisimulation) such that $\mu \equiv_{\mathcal{R}} \mu'$. The bisimulation resulting from this adaptation is called *naive weak bisimulation* and is denoted $\asymp$.

Before progressing any further, let us introduce these concepts formally. We will then show that naive weak bisimulation is not always adequate, in the sense that it sometimes fails to equate states that we would intuitively consider weakly bisimilar. We will then introduce another, more complex, kind of weak bisimulation.

As before, let $MA = (S, \mathit{Act}, \mapsto, \Rightarrow)$ be an MA.

Definition 2.30 (Labelled trees). For two finite sequences of positive natural numbers $\sigma, \sigma' \in \mathbb{N}_{>0}^*$, we write $\sigma \leq \sigma'$ if σ is a prefix of σ', i.e. if there exists a possibly empty sequence ϕ such that $(\sigma \circ \phi) = \sigma'$, where $\circ$ denotes concatenation.

Let L be a (possibly uncountable) set. An *L-labelled tree* T is a partial function

$$T : \mathbb{N}_{>0}^* \mapsto L$$

such that:

1. for every $\sigma, \sigma' \in \mathbb{N}_{>0}^*$ such that $\sigma \in \mathit{dom}\left(T\right)$ and $\sigma' \leq \sigma$ it holds that $\sigma' \in \mathit{dom}\left(T\right)$,
2. for every $\sigma \in \mathbb{N}_{>0}^*$ and $i > 1$, if $(\sigma \circ i) \in \mathit{dom}\left(T\right)$, then also $\sigma \circ (i-1) \in \mathit{dom}\left(T\right)$, and
3. $\varepsilon \in \mathit{dom}\left(T\right)$.

The empty sequence ε is called the *root* of T.

For each $\sigma \in \mathit{dom}\left(T\right)$, let $\mathit{Children}\left(\sigma\right) = \{\sigma \circ i \mid (\sigma \circ i) \in \mathit{dom}\left(T\right)\}$ be the set of *children* of σ.

We then introduce the following notation:

- let $\mathit{Inner}\left(T\right) = \{\varepsilon\} \cup \{\sigma \in \mathit{dom}\left(T\right) \mid \mathit{Children}\left(\sigma\right) \neq \emptyset\}$ denote the set of *inner nodes* of T;
- let $\mathit{Leaf}\left(T\right) = \{\sigma \in \mathit{dom}\left(T\right) \mid \mathit{Children}\left(\sigma\right) = \emptyset\}$ denote the set of *leaf nodes* of T.

Note that if the tree has only one node, the root node, then this node is contained in both $\mathit{Inner}\left(T\right)$ and $\mathit{Leaf}\left(T\right)$. In any other case the two sets are disjoint.

Definition 2.31 (Transition trees, induced subdistributions). Let

$$L \quad = \quad S \times \mathbb{R}_{\geq 0} \times \left(\mathit{Act}^{\chi} \cup \{\bot\}\right).$$

A *transition tree* T is an L-labelled tree that satisfies several conditions that are specified below. Before we define the conditions, let us introduce the following notation: for a node $\sigma \in \mathit{dom}\left(T\right)$ whose label is $T\left(\sigma\right) = (s, p, \alpha) \in L$, let:

1. $\mathit{State}\left(\sigma\right) = s$ be the first element of the node's label,
2. $\mathit{Prob}\left(\sigma\right) = p$ be the second element of the node's label, and
3. $\mathit{Act}\left(\sigma\right) = \alpha$ be the third element of the node's label.

Then, T must satisfy the following conditions:

1. $\mathit{Prob}\left(\varepsilon\right) = \sum_{\sigma \in \mathit{Leaf}(T)} \mathit{Prob}\left(\sigma\right) = 1$,

2. $\forall \sigma \in \mathit{Leaf}(T) \ . \ \mathit{Act}(\sigma) = \bot$, and
3. $\forall \sigma \in \mathit{Inner}(T) \setminus \mathit{Leaf}(T) \ . \ \exists \mu \ . \ \mathit{State}(\sigma) \xrightarrow{\mathit{Act}(\sigma)} \mu$ and

$$\mathit{Prob}(\sigma) \cdot \mu \quad = \quad \{(\mathit{State}(\sigma'), \mathit{Prob}(\sigma')) \mid \sigma' \in \mathit{Children}(\sigma)\}.$$

An *internal* transition tree is a transition tree where each $\mathit{Act}(\sigma)$ is either τ or $\bot$.

Finally, we say that the distribution $\mu_T \in \mathit{Dist}(S)$ defined by

$$\mu_T \quad = \quad \bigoplus_{\sigma \in Leaf(T)} \{(\mathit{State}(\sigma), \mathit{Prob}(\sigma))\}$$

is *induced* by T.

Definition 2.32 (Weak transitions). For $s \in S$, $\alpha \in \mathit{Act}^{\chi}$ and $\mu \in \mathit{Dist}(S)$ we write

$$s \quad \overset{\alpha}{\hookrightarrow} \quad \mu$$

if

- either $\alpha = \tau$ and μ is induced by some internal transition tree T with $\mathit{State}(\varepsilon) = s$,
- or $\alpha \neq \tau$ and μ is induced by some transition tree T such that:
  - $\mathit{State}(\varepsilon) = s$,
  - on every maximal path from the root (i.e., on every path from the root ending in a leaf) exactly one node is labelled with α and all other inner nodes are labelled with τ.

Furthermore, we write

$$s \quad \overset{\alpha}{\hookrightarrow}_{\oplus} \quad \mu$$

if there is a finite indexed set $\{(c_i, \mu_i)\}_{i \in \{1,...,n\}}$ of pairs of positive real valued weights and distributions such that:

- $s \overset{\alpha}{\hookrightarrow} \mu_i$ for each $i \in \{1, ..., n\}$,
- $\sum_{i=1}^{n} c_i = 1$, and
- $\mu = \bigoplus_{i=1}^{n} (c_i \cdot \mu_i)$.

Finally, we lift the notation to subdistributions as follows. For $\mu, \mu' \in \mathit{Subdist}(S)$ and $\alpha \in \mathit{Act}^{\chi}$ we write

$$\mu \quad \overset{\alpha}{\hookrightarrow}_{\oplus} \quad \mu'$$

if for each $s \in Supp\,(\mu)$ there exists a distribution μ'_s such that $s \overset{\alpha}{\hookrightarrow}_{\oplus} \mu'_s$ and

$$\mu' \quad = \bigoplus_{s \in Supp(\mu)} \left(\mu\,(s) \cdot \mu'_s\right).$$

Definition 2.33 (Naive weak bisimulation). An equivalence relation $\mathcal{R}$ on S is called a naive weak bisimulation relation on S if for all $(s, s') \in \mathcal{R}$, $\alpha \in Act^{\chi}$ and $\mu \in Dist\,(S)$, $s \xrightarrow{\alpha} \mu$ implies $s' \overset{\alpha}{\hookrightarrow}_{\oplus} \mu'$ for some μ' such that $\mu \equiv_{\mathcal{R}} \mu'$.

We write $s \asymp s'$ if (s, s') is contained in some naive weak bisimulation relation on S.

Lemma 2.34 (Coarsest naive weak bisimulation relation). *The relation $\asymp$ is the coarsest naive weak bisimulation relation on S.*

Thus, the only difference between strong and naive weak bisimulations is that in the latter case a transition is simulated by a weak transition. Unfortunately, in [8] Eisentraut et al. observed that naive weak bisimulation fails to equate some pairs of states that should intuitively be equivalent (see the states s and t in Figure 7 for an example). They, therefore, proposed a still weaker bisimulation, which we will now introduce.

The term for the new bisimulation is simply "weak bisimulation" (denoted $\approx$). The most unusual thing about it is that it is a relation on *subdistributions* over the states. Intuitively, we can describe it as follows. Suppose μ_1, μ_2 are subdistributions over the states such that $\mu_1 \approx \mu_2$ and $Supp\,(\mu_1) = \{s_1, ..., s_n\}$. Then, μ_2 can reach a distribution μ_3 via τ-transitions, such that μ_3 can be split into n subdistributions $\mu_3 = \bigoplus_{i=1}^{n} \mu_3^i$ (one for each state in $Supp\,(\mu_1)$). For each μ_3^i it must hold that:

1. $\{(s_i, \mu_1\,(s_i))\} \approx \mu_3^i$, and
2. whenever s_i can execute a transition $s_i \xrightarrow{\alpha} \mu'$, μ_3^i can execute a weak transition $\mu_3^i \overset{\alpha}{\hookrightarrow}_{\oplus} \mu''$ with $(\mu_1\,(s_i) \cdot \mu') \approx \mu''$.

In [8], Eisentraut et al. showed that the power of weak bisimulation comes from the fact that μ_2 is allowed to execute τ-transitions before splitting into n subdistributions. In fact, if we remove this step and require that μ_2 be split directly, the resulting relation (to be precise, its restriction to subdistributions of the form $\{(s, 1)\}$) coincides with naive weak bisimulation.

We now present a formal definition of weak bisimulation. The reader is encouraged to spend some time on it, and then work through the example in Figure 7 until he or she clearly understands why $s \approx t$ even though $s \not\asymp t$.

Definition 2.35 (Weak bisimulation). A relation $\mathcal{R}$ on $\mathit{Subdist}\,(S)$ is called a weak bisimulation relation if whenever $(\mu_1, \mu_2) \in \mathcal{R}$, then:

1. $|\mu_1| = |\mu_2|$.
2. For every $\alpha \in \mathit{Act}^{\chi}$ and $s \in \mathit{Supp}\,(\mu_1)$, there exist $\mu_2^s, \mu_2^{rest} \in \mathit{Subdist}\,(S)$ such that:
   (a) $\mu_2 \overset{\tau}{\hookrightarrow}_{\oplus} (\mu_2^s \oplus \mu_2^{rest})$,
   (b) $\{(s, \mu_1\,(s))\}\ \mathcal{R}\ \mu_2^s$ and $(\mu_1 - s)\ \mathcal{R}\ \mu_2^{rest}$, and
   (c) whenever $s \xrightarrow{\alpha} \mu_1'$ for some μ_1', then $\mu_2^s \overset{\alpha}{\hookrightarrow}_{\oplus} \mu''$ with $(\mu_1\,(s) \cdot \mu_1')\ \mathcal{R}\ \mu''$.
3. For every $\alpha \in \mathit{Act}^{\chi}$ and $s \in \mathit{Supp}\,(\mu_2)$, there exist $\mu_1^s, \mu_1^{rest} \in \mathit{Subdist}\,(S)$ such that:
   (a) $\mu_1 \overset{\tau}{\hookrightarrow}_{\oplus} (\mu_1^s \oplus \mu_1^{rest})$,
   (b) $\mu_1^s\ \mathcal{R}\ \{(s, \mu_2\,(s))\}$ and $\mu_1^{rest}\ \mathcal{R}\ (\mu_2 - s)$, and
   (c) whenever $s \xrightarrow{\alpha} \mu_2'$ for some μ_2', then $\mu_1^s \overset{\alpha}{\hookrightarrow}_{\oplus} \mu''$ with $\mu''\ \mathcal{R}\ (\mu_2\,(s) \cdot \mu_2')$.

Two subdistributions μ and μ' are weakly bisimilar, written $\mu \approx \mu'$, if the pair (μ, μ') is contained in some weak bisimulation relation. We say that two states s', s' are weakly bisimilar, denoted by $s \approx s'$, if $\{(s, 1)\} \approx \{(s', 1)\}$. Similarly, $s \approx \mu$ if $\{(s, 1)\} \approx \mu$, etc. Two Markov automata are weakly bisimilar if their initial distributions are weakly bisimilar in their direct sum.

Lemma 2.36 (Coarsest weak bisimulation relation). *The relation $\approx$ is the coarsest weak bisimulation relation on Subdist (S).*

Lemma 2.37 (Weak bisimulation is an equivalence). *The relation $\approx$ is an equivalence relation.*

Lemma 2.38 (Strong, naive and weak bisimulations). *For every pair of states $s, s' \in S$,*

$$s \sim s' \quad \textit{implies} \quad s \asymp s'$$

and

$$s \asymp s' \quad \textit{implies} \quad s \approx s'.$$

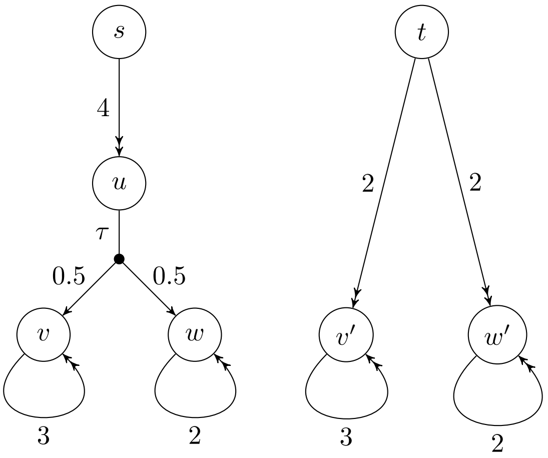


Figure 7: $s \approx t$ but $s \not\asymp t$.

2.6 Branching bisimulation

Branching bisimulation for MAs, covered in this section, was introduced by Timmer et al. in [14, 13] (the original idea for LTSs was presented in [15] by van Glabbeek and Weijland). It follows the same general approach as naive weak bisimulation: in other words, a transition $s \xrightarrow{\alpha} \mu$ is simulated by a tree of transitions from the bisimilar state s' to some distribution μ' with $\mu \equiv_{\simeq} \mu'$ (where $\simeq$ is the symbol for branching bisimulation). The corresponding tree of transitions is similar to the transition tree in the case of naive weak bisimulation, with the following differences:

1. It is *not* required that each node should have at most one outgoing transition.
2. If $\alpha \neq \tau$, then on every path from the root to a leaf the *last* transition must be labelled with α, and the other transitions must be labelled with τ (in the case of naive weak bisimulation it is not required that the α-transition come last).
3. On every path from the root to a leaf, all states except the last one must be in the same equivalence class under $\simeq$ as s and s'.

Note that if the first difference was absent, it would be completely obvious that branching bisimulation is stronger than naive weak bisimulation. With it, however, the relationship between the two bisimulations is not immediately obvious. In [13], Timmer claimed that branching bisimulation is stronger than naive weak bisimulation (although a proof was not provided). In this work we are not going to rely on this claim.

Another observation is that in the case of branching bisimulation we do not need combinations of transition trees since we allow a node to have multiple outgoing transitions.

As an example, consider Figure 8. It is a good idea for the reader to make sure he or she has a clear understanding of why $s \asymp s'$ even though $s \not\simeq s'$.

We next turn to a formal definition of branching bisimulation. In [14, 13], Timmer et al. did not use transition trees, but rather an equivalent definition involving termination-enabled schedulers. Here,

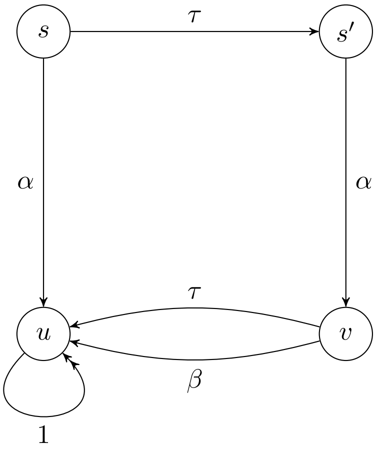


Figure 8: $s \asymp s'$ but $s \not\approx s'$.

we follow their original approach. It is, however, important to keep the alternative characterization in mind, particularly because it enables us to think about all three weak bisimulations within the same framework.

Let $MA = (S, Act, \mapsto, \Rightarrow)$ be a semi-closed MA.

Definition 2.39 (Stop action). Let

$$Act^{\perp} \quad = \quad Act^{\intercal} \cup \{\perp\},$$

where the special action $\perp$ signifies the fact that a scheduler does not pick any action, but chooses to terminate instead.

For each $s \in S$, let

$$Act^{\perp}(s) \quad = \quad Act^{\intercal}(s) \cup \{\perp\}.$$

An attentive reader might notice that we have already used the symbol $\perp$ in the definition of the transition tree (Definition 2.31). The use of the same symbol here is justified because in both cases it serves the same purpose: to specify termination (a leaf of a transition tree or the fact that a scheduler chooses not to continue).

Definition 2.40 (Measurable termination-enabled time-abstract scheduler). A *termination-enabled time-abstract scheduler* on a semi-closed Markov automaton is a mapping

$$D : Paths_{abs}^{*} \times Act^{\perp} \mapsto [0,1]$$

such that for all $\pi \in Paths_{abs}^{*}$,

$$D(\pi, \cdot) \quad \in \quad Distr\left(Act^{\perp}(\pi_{\downarrow})\right).$$

For every $\pi \in Paths^*_{abs}$ and $A \in 2^{Act^{\perp}}$, we set

$$D(\pi, A) = \sum_{\alpha \in A} D(\pi, \alpha).$$

A termination-enabled time-abstract scheduler D is *measurable* if for all $A \in 2^{Act^{\perp}}$, $D^{-1}(A) : Paths^*_{abs} \mapsto [0,1]$ is measurable.

We assume all termination-enabled time-abstract schedulers to be measurable. Let $TETAS$ be the set of all measurable termination-enabled time-abstract schedulers on MA.

Definition 2.41 (Probabilities of time-abstract paths w.r.t. termination-enabled time-abstract scheduler). Let $D \in TETAS$, and let $\mu_0 \in Dist(S)$. Then, define the function

$$\Pr_{\mu_0,D} : Paths^*_{abs} \mapsto [0,1]$$

recursively as follows:

1. $\Pr\limits_{\mu_0,D}(s_0) = \mu_0(s_0)$, and
2. $\Pr\limits_{\mu_0,D}\left(\pi \xrightarrow{\alpha_{n-1}} s_n\right) = \Pr\limits_{\mu_0,D}(\pi) \cdot D(\pi, \alpha_{n-1}) \cdot \mu_{\pi_{\downarrow},\alpha_{n-1}}(s_n)$.

Definition 2.42 (Maximal time-abstract paths induced by termination-enabled scheduler). Let $D \in TETAS$, and let $s \in S$. Then, the set of *maximal time-abstract paths from s induced by D* is defined as

$$MaxPaths^D_{abs}(s) = \left\{\pi \in Paths^*_{abs} \mid \pi[0] = s \wedge \Pr_{s,D}(\pi) > 0 \wedge D(\pi, \perp) > 0\right\}.$$

Definition 2.43 (Distribution induced by termination-enabled scheduler). Let $D \in TETAS$, and let $s \in S$ be an initial state. Then, the *induced subdistribution* $\mu^D_s \in SubDist(S)$ is defined as follows: for every $s' \in S$,

$$\mu^D_s(s') = \sum_{\substack{\pi \in MaxPaths^D_{abs}(s) \\ \pi_{\downarrow} = s'}} \Pr_{s,D}(\pi) \cdot D(\pi, \perp).$$

Definition 2.44 (Branching transitions with respect to equivalence relation). Let $s \in S$, $\alpha \in Act^{\chi}$ and $\mu \in Dist(S)$. Moreover, let $\mathcal{R}$ be an equivalence relation on S. We write

$$s \overset{\alpha}{\hookrightarrow}_{\mathcal{R}} \mu$$

if:

- either $\alpha = \tau$ and $\mu = \{(s, 1)\}$,
- or there exists some $D \in TETAS$ such that

  $$\mu_s^D \quad = \quad \mu$$

  and for every time-abstract maximal path

  $$s \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{n-1}} s_n \quad \in \quad MaxPaths^D_{abs}(s)$$

  it holds that

  - $\alpha_{n-1} = \alpha$,
  - for all $i \in \{0, ..., n-2\}$, $\alpha_i = \tau$, and
  - for all $i \in \{1, ..., n-1\}$, $(s, s_i) \in \mathcal{R}$.

Definition 2.45 (Branching bisimulation). An equivalence relation $\mathcal{R}$ on S is a *branching bisimulation relation* if for every $(s, s') \in \mathcal{R}$, $\alpha \in Act^{\chi}$ and $\mu \in Dist(S)$, $s \overset{\alpha}{\to} \mu$ implies $s' \overset{\alpha}{\hookrightarrow}_{\mathcal{R}} \mu'$ for some μ' such that $\mu \equiv_{\mathcal{R}} \mu'$.

We write $s \backsimeq s'$ if (s, s') is contained in some branching bisimulation relation on S.

Lemma 2.46 (Coarsest branching bisimulation relation). *The relation $\backsimeq$ is the coarsest branching bisimulation relation on S.*

As was noted above, in [13] Timmer claimed that branching bisimulation is stronger than naive weak bisimulation (although a proof was not provided).

Claim 2.47 (Branching bisimulation is stonger than naive weak bisimulation). For every pair of states $s, s' \in S$,

$$s \simeq s' \quad \text{implies} \quad s \asymp s'.$$

2.7 Expected time

In this section we consider the concept of expected time in Markov automata. In this, we closely follow [9].

Let us first outline the concept informally. Suppose $s \in S$, $G \subseteq S$ and $D \in GM$. Then, the symbol $eT^D(s, \Diamond G)$ denotes the expected time to reach a state in G starting from s, if non-determinism is resolved by the scheduler D. Next, $eT^{min}(s, \Diamond G)$ is the minimum of $eT^D(s, \Diamond G)$ over all $D \in GM$.

Again, we assume that the Markov automaton $MA = (S, Act, \mapsto, \Rightarrow)$ is semi-closed.

Definition 2.48 (Minimum expected time). Let $G \subseteq S$.

First, we define the function

$$V_G : Paths^\omega \mapsto [0, \infty]$$

as follows:

$$V_G(\pi) \quad = \quad \min\{t \in \mathbb{R}_{\geq 0} \mid G \cap \pi@t \neq \emptyset\},$$

with $\min(\emptyset) = +\infty$.

Then, for a scheduler $D \in GM$, the expected time to reach G is given by the function

$$eT^D(\cdot, \Diamond G) : S \mapsto [0, \infty],$$

defined by

$$eT^D(s, \Diamond G) \quad = \quad \int_{Paths^\omega} V_G(\pi) \, \Pr_{s,D}(d\pi).$$

Finally, the *minimum expected time* to reach G from an initial state $s \in S$ is given by the function

$$eT^{min}(\cdot, \Diamond G) : S \mapsto [0, \infty],$$

defined as follows:

$$eT^{min}(s, \Diamond G) \quad = \quad \inf_{D \in GM} \left\{eT^D(s, \Diamond G)\right\}.$$

Lemma 2.49 (Minimum expected time as least fixpoint). *For every $G \subseteq S$, the function eT^{min} is the unique fixpoint of the Bellman operator*

$$[L(v)](s) \quad = \quad \begin{cases} \dfrac{1}{E(s)} + \sum\limits_{s' \in S} \mu_{s,\mathsf{T}}(s') \cdot v(s') & \text{if } s \in MS \setminus G \\ \min\limits_{\alpha \in Act(s)} \left\{\sum\limits_{s' \in S} \mu_{s,\alpha}(s') \cdot v(s')\right\} & \text{if } s \in PS \setminus G \\ 0 & \text{if } s \in G. \end{cases}$$

Lemma 2.50 (Minimum expected time given by stationary deterministic scheduler). *For every $G \subseteq S$, there exists a stationary deterministic scheduler D such that for every $s \in S$ it holds that*

$$eT^{min}(s, \Diamond G) \quad = \quad eT^D(s, \Diamond G).$$

2.8 Long-run average

Once again, everything in this section is merely a restatement of the definitions and results from [9].

As in the case of expected time, suppose $s \in S$, $G \subseteq S$ and $D \in GM$. Then, the symbol $LRA^D(s, \Diamond G)$ denotes the expected *fraction* of time spent in the states in G starting from s, if non-determinism is resolved by the scheduler D. Of course, $LRA^{min}(s, \Diamond G)$ is the minimum of $LRA^D(s, \Diamond G)$ over all $D \in GM$.

Throughout this section, we assume that the Markov automaton $MA = (S, Act, \mapsto, \Rightarrow)$ is semi-closed.

Definition 2.51 (Long-run average). Let $G \subseteq S$. For a scheduler $D \in GM$, the function

$$LRA^D(\cdot, \Diamond G) : S \mapsto [0,1]$$

is defined as follows:

$$LRA^D(s, \Diamond G) \quad = \quad \int_{Paths^\omega} \left(\lim_{t \to \infty} \frac{1}{t} \int_0^t I_G(\pi@u)\ du \right) \Pr_{s,D}(d\pi),$$

where

$$I_G(\pi@u) \quad = \quad \begin{cases} 1 & \text{if } \pi@u \,\cap\, G \neq \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

The *minimum long-run average* is given by the function

$$LRA^{min}(s, \Diamond G) \quad = \quad \inf_{D \in GM} \left\{ LRA^D(s, \Diamond G) \right\}.$$

Lemma 2.52 (Minimum long-run average is given by stationary deterministic scheduler).

Let $G \subseteq S$. Then there exists a stationary deterministic scheduler D such that for every $s \in S$ it holds that

$$LRA^{min}(s, \Diamond G) \quad = \quad LRA^D(s, \Diamond G).$$

3 Bisimulation quotient and quotient schedulers

In this chapter we define the bisimulation quotient $MA/_\sim$ of a Markov automaton MA and study the connection between MA and $MA/_\sim$. In particular, we introduce the notions of bisimulation-closed sets of finite paths and quotient schedulers. Intuitively, a set of finite paths Π is bisimulation-closed if it has a natural counterpart set $\widetilde{\Pi}$ in $MA/_\sim$ (see below for a formal definition). Furthermore, we will show that *every* scheduler D has a counterpart $\widetilde{D}$ on $MA/_\sim$, imaginatively called the quotient scheduler of D, such that for every bisimulation-closed set Π, the probability of Π under D equals the probability of $\widetilde{\Pi}$ under $\widetilde{D}$.

The notion of the bisimulation quotient is well-established in the field of model checking (see, for example, [1, 11]). Furthermore, we adopted the idea of the quotient scheduler directly from [11], where it was applied to CTMDPs. Note that our proofs are somewhat different from [11] (for example, they are more detailed).

3.1 Bisimulation quotient

Let $MA = (S, Act, \hookrightarrow, \Rightarrow)$ be an MA.

Definition 3.1 (Quotient Markov automaton). Let $\mathcal{R}$ be a strong bisimulation relation on S. Then the *quotient* of MA under $\mathcal{R}$ is defined as $MA/\mathcal{R} = (S/\mathcal{R}, Act, \hookrightarrow', \Rightarrow')$, where:

1. $\dfrac{s \xhookrightarrow{\alpha} \mu}{[s]_{\mathcal{R}} \xhookrightarrow{\alpha}{}' (\mu/\mathcal{R})}$, and
2. $\dfrac{C' \in S/\mathcal{R} \qquad \lambda_{s,C'} = \sum_{s' \in C'} \sum_{(s,\lambda',s') \in TT(s)} \lambda'}{[s]_{\mathcal{R}} \overset{\lambda_{s,C'}}{\Rightarrow'} C'}$.

Furthermore, if $\mu_0 \in Dist(S)$ is the initial distribution over the states of MA, we designate $\mu_0/\mathcal{R}$ as the initial distribution over the states of $MA/\mathcal{R}$.

Theorem 3.2 (Bisimulation equivalence of MA and $MA/_\sim$).
Let MA be a Markov automaton. Then $MA \sim MA/_\sim$.

Proof. By showing that the reflexive transitive symmetric closure of $\{(s, [s]_\sim) \mid s \in S\}$ is a strong bisimulation relation on the states of $MA \oplus MA/_\sim$. □

Lemma 3.3. *For every $s \in S$, all of the following holds:*

1. *If* $TT(s) \neq \emptyset$*, then* $E(s) = \widetilde{E}([s]_\sim)$*, where* $\widetilde{E}([s]_\sim)$ *is the exit rate of the state* $[s]_\sim$ in $MA/_\sim$.
2. *If* $TT(s) \neq \emptyset$*, then for every* $C \in S/_\sim$ *it holds that* $\mu_{s,\mathsf{T}}(C) = \widetilde{\mu}_{[s]_\sim,\mathsf{T}}(C)$*, where* $\widetilde{\mu}_{[s]_\sim,\mathsf{T}}(C)$ *is the probability to move from* $[s]_\sim$ to C in $MA/_\sim$.
3. *There exists a transition* $s \overset{\alpha}{\longmapsto} \mu$ *in* MA *if and only if there exists a transition* $[s]_\sim \overset{\alpha}{\longmapsto} (\mu/_\sim)$ *in* $MA/_\sim$.

3.2 Quotient schedulers

In this section we assume that the Markov automaton $MA = (S, Act, \mapsto, \Rightarrow)$ is semi-closed.

Definition 3.4 (Bisimulation-closed measurable sets and rectangles). A measurable set of finite paths Π is *bisimulation-closed* if it has the form

$$\Pi \quad = \quad \{(s_0, \alpha_0, t_0, ..., s_n) \in S_0 \times A_0 \times \mathbb{R}_{\geq 0} \times ... \times A_{n-1} \times \mathbb{R}_{\geq 0} \times S_n \mid (t_0, ..., t_{n-1}) \in T\},$$

where:

1. for every $i \in \{0, ..., n\}$, $S_i \subseteq S$ is the union of one or more equivalence classes under $\sim$,
2. for every $i \in \{0, ..., n-1\}$, $A_i \subseteq Act^{\mathsf{T}}$, and
3. T is a measurable subset of $(\mathbb{R}_{\geq 0})^n$.

We say that Π is *simple bisimulation-closed* if for each $i \in \{0, ..., n\}$ it holds that $S_i \in S/_\sim$.

As a special case, a measurable rectangle

$$\Pi \quad = \quad S_0 \times A_0 \times T_0 \times ... \times A_{n-1} \times T_{n-1} \times S_n$$

is *bisimulation-closed* if for each $i \in \{0, ..., n\}$ it holds that S_i is the union of one or more equivalence classes under $\sim$. Π is *simple bisimulation-closed* if for each $i \in \{0, ..., n\}$ it holds that $S_i \in S/_\sim$.

For the given bisimulation-closed set of finite paths

$$\Pi \quad = \quad \{(s_0, \alpha_0, t_0, ..., s_n) \in S_0 \times A_0 \times \mathbb{R}_{\geq 0} \times ... \times A_{n-1} \times \mathbb{R}_{\geq 0} \times S_n \mid (t_0, ..., t_{n-1}) \in T\},$$

let

$$\widetilde{\Pi} \quad = \quad \{(s_0, \alpha_0, t_0, ..., s_n) \in (S_0/_\sim) \times A_0 \times \mathbb{R}_{\geq 0} \times ... \times A_{n-1} \times \mathbb{R}_{\geq 0} \times (S_n/_\sim) \mid (t_0, ..., t_{n-1}) \in T\}$$

be the corresponding set of finite paths in the quotient $MA/_\sim$.

Now, suppose $\mu_0 \in Dist(S)$ and D is a scheduler. A natural question to ask is whether we can define a scheduler $\widetilde{D}$ on $MA/_\sim$ such that

$$\Pr_{\mu_0, D}(\Pi) \quad = \quad \Pr_{\mu_0/_\sim, \widetilde{D}}\left(\widetilde{\Pi}\right)$$

for every bisimulation-closed set of finite paths Π.

The answer turns out to be yes, as we will show in the rest of this section. Before we turn to formal proofs, let us outline the idea informally. Suppose

$$\widetilde{\pi} \quad = \quad [s_0]_\sim \xrightarrow{\alpha_0, t_0} [s_1]_\sim \xrightarrow{\alpha_1, t_1} \ldots \xrightarrow{\alpha_{n-1}, t_{n-1}} [s_n]_\sim$$

is a finite path in $MA/_\sim$. Note that $\widetilde{\pi}$ corresponds to a finite number of finite paths in MA. In fact, the corresponding finite paths in MA form a simple bisimulation-closed rectangle

$$\Pi \quad = \quad [s_0]_\sim \times \{\alpha_0\} \times \{t_0\} \times [s_1]_\sim \times \ldots \times [s_n]_\sim\,.$$

Then all $\widetilde{D}$ needs to do is to compute the weighted mean of the decisions of D on all of these corresponding paths, where the weights reflect the fact that the finite paths in Π have different probabilities. Thus our first attempt might look like this:

$$\widetilde{D}(\widetilde{\pi}, \alpha) \quad = \quad \frac{\sum\limits_{\pi \in \Pi} \Pr_{\mu_0, D}(\pi) \cdot D(\pi, \alpha)}{\sum\limits_{\pi \in \Pi} \Pr_{\mu_0, D}(\pi)} \qquad\qquad \text{(wrong!)}\,.$$

Unfortunately, this is not quite correct since $\Pr_{\mu_0, D}(\pi) = 0$ whenever π contains a Markovian state (because π contains precise values of $t_0, \ldots, t_{n-1}$ instead of intervals). However, we can easily solve the problem by using the *conditional* probabilities $\Pr_{\mu_0, D}(\pi \mid t_0, \ldots, t_{n-1})$:

$$\widetilde{D}(\widetilde{\pi}, \alpha) \quad = \quad \frac{\sum\limits_{\pi \in \Pi} \Pr_{\mu_0, D}(\pi \mid t_0, \ldots, t_{n-1}) \cdot D(\pi, \alpha)}{\sum\limits_{\pi \in \Pi} \Pr_{\mu_0, D}(\pi \mid t_0, \ldots, t_{n-1})} \qquad\qquad \text{(correct!)}\,.$$

Following [11], we call these conditional probabilities history weights, which reflects their use in the definition of $\widetilde{D}$. Nevertheless, it is useful to keep their underlying nature in mind.

We now turn to formal definitions and proofs.

Definition 3.5 (History weight, quotient schedulers). First, we define the *history weight* function

$$hw : Dist(S) \times GM \times Paths^* \mapsto \mathbb{R}_{\geq 0}$$

recursively as follows:

1. $hw\left(\mu_0, D, s_0\right) = \mu_0\left(s_0\right)$ and
2. $hw\left(\mu_0, D, \pi \xrightarrow{\alpha_n, t_n} s_{n+1}\right) = hw\left(\mu_0, D, \pi\right) \cdot D\left(\pi, \alpha_n\right) \cdot \mu_{\pi_\downarrow, \alpha_n}\left(s_{n+1}\right)$.

Now, let μ_0 be an initial distribution over S and let $D \in GM$. Then, a *quotient scheduler* $\widetilde{D}_{\mu_0}$ *with respect to* D *and* μ_0 is any scheduler on $MA/_\sim$ such that, for every finite path

$$\widetilde{\pi} \quad = \quad [s_0]_\sim \xrightarrow{\alpha_0, t_0} [s_1]_\sim \xrightarrow{\alpha_1, t_1} \ldots \xrightarrow{\alpha_{n-1}, t_{n-1}} [s_n]_\sim$$

in $MA/_\sim$ whose corresponding simple bisimulation-closed rectangle in MA is

$$\Pi \quad = \quad [s_0]_\sim \times \{\alpha_0\} \times \{t_0\} \times [s_1]_\sim \times \ldots \times [s_n]_\sim\,,$$

it holds that

$$\sum_{\pi \in \Pi} hw\left(\mu_0, D, \pi\right) \quad \neq \quad 0$$

implies

$$\widetilde{D}_{\mu_0}\left(\widetilde{\pi}, \alpha_n\right) \quad = \quad \frac{\sum_{\pi \in \Pi} hw\left(\mu_0, D, \pi\right) \cdot D\left(\pi, \alpha_n\right)}{\sum_{\pi \in \Pi} hw\left(\mu_0, D, \pi\right)}$$

for every $\alpha_n \in Act^\intercal$.

A few comments on Definition 3.5:

1. There can be multiple quotient schedulers with respect to D and μ_0.
2. Whenever $\sum_{\pi \in \Pi} hw\left(\mu_0, D, \pi\right) > 0$ for some path $\widetilde{\pi}$ in $MA/_\sim$, *every* quotient scheduler with respect to D and μ_0 *must* return the same distribution defined by

$$\widetilde{D}_{\mu_0}\left(\widetilde{\pi}, \alpha_n\right) \quad = \quad \frac{\sum_{\pi \in \Pi} hw\left(\mu_0, D, \pi\right) \cdot D\left(\pi, \alpha_n\right)}{\sum_{\pi \in \Pi} hw\left(\mu_0, D, \pi\right)}.$$

3. If $\sum_{\pi \in \Pi} hw\left(\mu_0, D, \pi\right) = 0$ (which happens only when $hw\left(\mu_0, D, \pi\right) = 0$ for every $\pi \in \Pi$), the decision made by a quotient scheduler can be arbitrary.

Lemma 3.6. *For every quotient scheduler* $\widetilde{D}_{\mu_0}$ *with respect to* $D \in GM$ *and* $\mu_0 \in Dist\left(S\right)$ *it holds that, for every path* $\widetilde{\pi} \in Paths^*$ *in* $MA/_\sim$ *and* $\alpha_n \in Act^\intercal$,

$$\widetilde{D}_{\mu_0}\left(\widetilde{\pi}, \alpha_n\right) \cdot \sum_{\pi \in \Pi} hw\left(\mu_0, D, \pi\right) \quad = \quad \sum_{\pi \in \Pi} hw\left(\mu_0, D, \pi\right) \cdot D\left(\pi, \alpha_n\right),$$

independently of whether $\sum_{\pi\in\Pi} hw\left(\mu_0, D, \pi\right) = 0$ *or not.*

Lemma 3.7 (Correspondence between history weights in MA and $MA/_\sim$).

Let μ_0*,* D*,* $\widetilde{D}_{\mu_0}$*,* $\widetilde{\pi}$*, and* Π *be as in Definition 3.5. Then,*

$$\sum_{\pi\in\Pi} hw\left(\mu_0, D, \pi\right) \quad = \quad hw\left(\mu_0/_\sim, \widetilde{D}_{\mu_0}, \widetilde{\pi}\right),$$

where the history weight in the right-hand side is determined in $MA/_\sim$*.*

Proof. By induction on $|\widetilde{\pi}| = n$.

Base case: $n = 0$ (i.e. $\widetilde{\pi} = [s_0]_\sim$). Then,

$$\begin{aligned} \sum_{\pi\in\Pi} hw\left(\mu_0, D, \pi\right) &= \sum_{s\in[s_0]_\sim} \mu_0\left(s\right) \\ &= \mu_0/_\sim\left([s_0]_\sim\right) \\ &= hw\left(\mu_0/_\sim, \widetilde{D}_{\mu_0}, \widetilde{\pi}\right). \end{aligned}$$

Induction step. Assume that the claim holds for a path $\widetilde{\pi}$ of length n, whose corresponding rectangle in MA is Π. Consider its extension

$$\widetilde{\pi} \quad \xrightarrow{\alpha_n, t_n} \quad [s_{n+1}]_\sim$$

with the corresponding rectangle

$$\Pi' \quad = \quad \Pi \times \{\alpha_n\} \times \{t_n\} \times [s_{n+1}]_\sim .$$

Now,

$$\begin{aligned} \sum_{\pi'\in\Pi'} hw\left(\mu_0, D, \pi'\right) &= \sum_{\pi\in\Pi} \sum_{s\in[s_{n+1}]_\sim} hw\left(\mu_0, D, \pi \xrightarrow{\alpha_n, t_n} s\right) \\ \text{(by definition of } hw\text{)} &= \sum_{\pi\in\Pi} \sum_{s\in[s_{n+1}]_\sim} hw\left(\mu_0, D, \pi\right) \cdot D\left(\pi, \alpha_n\right) \cdot \mu_{\pi_\downarrow, \alpha_n}\left(s\right) \\ \left(\text{moving} \sum_{s\in[s_{n+1}]_\sim} \text{inside}\right) &= \sum_{\pi\in\Pi} hw\left(\mu_0, D, \pi\right) \cdot D\left(\pi, \alpha_n\right) \cdot \sum_{s\in[s_{n+1}]_\sim} \mu_{\pi_\downarrow, \alpha_n}\left(s\right). \end{aligned}$$

Next, for every $\pi \in \Pi$, $\pi_{\downarrow} \in [s_n]_{\sim}$. Then,

$$\begin{aligned}
\sum_{\pi' \in \Pi'} hw\left(\mu_0, D, \pi'\right) &= \sum_{\pi \in \Pi} hw\left(\mu_0, D, \pi\right) \cdot D\left(\pi, \alpha_n\right) \cdot \sum_{s \in [s_{n+1}]_{\sim}} \mu_{\pi_{\downarrow}, \alpha_n}\left(s\right) \\
\text{(since } \pi_{\downarrow} \in [s_n]_{\sim}\text{)} \quad &= \sum_{\pi \in \Pi} hw\left(\mu_0, D, \pi\right) \cdot D\left(\pi, \alpha_n\right) \cdot \mu_{[s_n]_{\sim}, \alpha_n}\left([s_{n+1}]_{\sim}\right) \\
\left(\text{moving } \mu_{[s_n]_{\sim}, \alpha_n}\left([s_{n+1}]_{\sim}\right) \text{ out}\right) \quad &= \mu_{[s_n]_{\sim}, \alpha_n}\left([s_{n+1}]_{\sim}\right) \cdot \sum_{\pi \in \Pi} hw\left(\mu_0, D, \pi\right) \cdot D\left(\pi, \alpha_n\right) \\
\text{(by Lemma 3.6)} \quad &= \mu_{[s_n]_{\sim}, \alpha_n}\left([s_{n+1}]_{\sim}\right) \cdot \widetilde{D}_{\mu_0}\left(\widetilde{\pi}, \alpha_n\right) \cdot \sum_{\pi \in \Pi} hw\left(\mu_0, D, \pi\right) \\
\text{(induction hypothesis)} \quad &= \mu_{[s_n]_{\sim}, \alpha_n}\left([s_{n+1}]_{\sim}\right) \cdot \widetilde{D}_{\mu_0}\left(\widetilde{\pi}, \alpha_n\right) \cdot hw\left(\mu_0/_{\sim}, \widetilde{D}_{\mu_0}, \widetilde{\pi}\right) \\
\text{(by definition of } hw\text{)} \quad &= hw\left(\mu_0/_{\sim}, \widetilde{D}_{\mu_0}, \widetilde{\pi} \xrightarrow{\alpha_n, t_n} [s_{n+1}]_{\sim}\right).
\end{aligned}$$

□

Lemma 3.8 (Probability of simple bisimulation-closed set of paths).

Let $\mu_0 \in Dist\left(S\right)$ and $D \in GM$. Then, for every simple bisimulation-closed set of paths

$$\Pi \quad = \quad \left\{\left(s_0, \alpha_0, t_0, ..., s_n\right) \in S_0 \times A_0 \times \mathbb{R}_{\geq 0} \times ... \times A_{n-1} \times \mathbb{R}_{\geq 0} \times S_n \mid \left(t_0, ..., t_{n-1}\right) \in T\right\},$$

such that $n > 0$, it holds that

$$\Pr_{\mu_0, D}\left(\Pi\right) \quad = \quad \int_{\left(\mathbb{R}_{\geq 0}\right)^n} \left[I_T\left(t_0, ..., t_{n-1}\right) \cdot \sum_{A_{0...n-1}} \sum_{S_{0...n}} hw\left(\mu_0, D, \pi\right)\right] \eta_{S_0}\left(dt_0\right) ... \eta_{S_{n-1}}\left(dt_{n-1}\right),$$

where:

1. *$S_{0...n}$ stands for $S_0 \times S_1 \times ... \times S_n$ and $A_{0...n-1}$ stands for $A_0 \times A_1 \times ... \times A_{n-1}$,*
2. *the finite path*

   $$\pi \quad = \quad s_0 \xrightarrow{\alpha_0, t_0} ... \xrightarrow{\alpha_{n-1}, t_{n-1}} s_n$$

   is defined by the variables of summation and integration,
3. *$I_T\left(t_0, ..., t_{n-1}\right) = 1$ if $\left(t_0, ..., t_{n-1}\right) \in T$ and 0 otherwise, and*
4. *for every $i \in \left(0, ..., n-1\right)$, η_{S_i} is the probability density corresponding to the state S_i of $MA/_{\sim}$.*

Proof. First of all, note that if $s \sim s'$, then $\eta_s = \eta_{s'} = \eta_{[s]_{\sim}}$.

Now, by Definition 2.23,

$$\Pr_{\mu_0, D}\left(\Pi\right) \quad = \quad \sum_{S} \sum_{Act^{\tau}} \int_{\mathbb{R}_{\geq 0}} \sum_{S} ... \sum_{S} \left[I_{\Pi}(\pi) \cdot \mu_0(s_0) \prod_{i=0}^{n-1} \left(D(\pi[0...i], \alpha_i) \mu_{s_i, \alpha_i}(s_{i+1})\right)\right] \eta_{s_{n-1}}(dt_{n-1}) ... \eta_{s_0}(dt_0).$$

Then, observe that from Definition 3.5 it follows that

$$\mu_0\left(s_0\right)\prod_{i=0}^{n-1}\left(D\left(\pi\left[0...i\right],\alpha_i\right)\mu_{s_i,\alpha_i}\left(s_{i+1}\right)\right) \quad = \quad hw\left(\mu_0,D,\pi\right).$$

Therefore,

$$\begin{aligned}
\Pr_{\mu_0,D}\left(\Pi\right) \quad &= \quad \sum_{S}\sum_{Act^{\intercal}}\int_{\mathbb{R}_{\geq 0}}\sum_{S}...\sum_{S}\left[I_{\Pi}(\pi)\cdot hw(\mu_0,D,\pi)\right]\eta_{s_{n-1}}(dt_{n-1})...\eta_{s_0}(dt_0) \\
(*) \quad &= \quad \sum_{S_0}\sum_{A_0}\int_{\mathbb{R}_{\geq 0}}\sum_{S_1}...\sum_{S_n}\left[I_T(t_0,...,t_{n-1})\cdot hw(\mu_0,D,\pi)\right]\eta_{s_{n-1}}(dt_{n-1})...\eta_{s_0}(dt_0) \\
(\text{since } \eta_{s_i} = \eta_{S_i}) \quad &= \quad \int_{\left(\mathbb{R}_{\geq 0}\right)^n}\left[\sum_{S_0\times A_0\times...\times S_n}\left(I_T(t_0,...,t_{n-1})\cdot hw(\mu_0,D,\pi)\right)\right]\eta_{S_0}(dt_0)...\eta_{S_{n-1}}(dt_{n-1}) \\
(\text{rearranging}) \quad &= \quad \int_{\left(\mathbb{R}_{\geq 0}\right)^n}\left[\sum_{A_{0...n-1}}\sum_{S_{0...n}}\left(I_T(t_0,...,t_{n-1})\cdot hw(\mu_0,D,\pi)\right)\right]\eta_{S_0}(dt_0)...\eta_{S_{n-1}}(dt_{n-1}) \\
(\text{moving } I_T\left(t_0,...,t_{n-1}\right) \text{ out}) \quad &= \quad \int_{\left(\mathbb{R}_{\geq 0}\right)^n}\left[I_T(t_0,...,t_{n-1})\cdot\sum_{A_{0...n-1}}\sum_{S_{0...n}}hw(\mu_0,D,\pi)\right]\eta_{S_0}(dt_0)...\eta_{S_{n-1}}(dt_{n-1}).
\end{aligned}$$

$(*)$ Note that this step is possible by replacing the summation of $I_{\Pi}\left(\pi\right)$ over the entire sets S and $Act^{\intercal}$ with the summation of $I_T\left(t_0,...,t_{n-1}\right)$ over the specific sets S_i and A_i. □

Theorem 3.9 (Preservation of probabilities of bisimulation-closed sets by quotient schedulers).
Let $\mu_0 \in Dist\left(S\right)$, $D \in GM$, and let $\widetilde{D}_{\mu_0}$ be a quotient scheduler with respect to D and μ_0. Then, for every bisimulation-closed set of finite paths

$$\Pi \quad = \quad \left\{\left(s_0,\alpha_0,t_0,...,s_n\right) \in S_0 \times A_0 \times \mathbb{R}_{\geq 0} \times ... \times A_{n-1} \times \mathbb{R}_{\geq 0} \times S_n \mid \left(t_0,...,t_{n-1}\right) \in T\right\},$$

it holds that

$$\Pr_{\mu_0,D}\left(\Pi\right) \quad = \quad \Pr_{\mu_0/\sim,\widetilde{D}_{\mu_0}}\left(\widetilde{\Pi}\right).$$

Proof. First of all, observe that every bisimulation-closed set of finite paths is the disjoint union of a finite number of simple bisimulation-closed sets of finite paths. Therefore, it is enough to prove that the statement holds if Π is simple bisimulation-closed, that is, when each $S_i \in S/_{\sim}$.

As in Lemma 3.8, let $S_{0...n}$ stand for $S_0 \times S_1 \times ... \times S_n$, and analogously for $A_{0...n-1}$.

Let us consider the cases $n = 0$ and $n > 0$ separately.

If $n = 0$, then

$$\begin{aligned} \Pr_{\mu_0, D}(S_0) &= \mu_0(S_0) \\ &= \mu_0/_{\sim}(S_0) \\ &= \Pr_{\mu_0/_{\sim}, \widetilde{D}_{\mu_0}}(\{S_0\}). \end{aligned}$$

If $n > 0$, then, using Lemma 3.8,

$$\Pr_{\mu_0, D}(\Pi) = \int_{\left(\mathbb{R}_{\geq 0}\right)^n} \left[I_T(t_0, \ldots, t_{n-1}) \cdot \sum_{A_{0 \ldots n-1}} \sum_{S_{0 \ldots n}} hw(\mu_0, D, \pi) \right] \eta_{S_0}(dt_0) \ldots \eta_{S_{n-1}}(dt_{n-1}),$$

where

$$\pi = s_0 \xrightarrow{\alpha_0, t_0} \ldots \xrightarrow{\alpha_{n-1}, t_{n-1}} s_n$$

is the path defined by the variables of summation and integration.

Now, let

$$\widetilde{\pi} = S_0 \xrightarrow{\alpha_0, t_0} \ldots \xrightarrow{\alpha_{n-1}, t_{n-1}} S_n,$$

where $t_0, \ldots, t_{n-1}, \alpha_0, \ldots, \alpha_{n-1}$ are again the variables of summation and integration. Furthermore, let $\mathcal{P}$ be the corresponding rectangle in MA. Then, observe that

$$\begin{aligned} \sum_{S_{0 \ldots n}} hw(\mu_0, D, \pi) &= \sum_{\pi \in \mathcal{P}} hw(\mu_0, D, \pi) \\ \text{(by Lemma 3.7)} &= hw\left(\mu_0/_{\sim}, \widetilde{D}_{\mu_0}, \widetilde{\pi}\right). \end{aligned}$$

Therefore,

$$\begin{aligned} \Pr_{\mu_0, D}(\Pi) &= \int_{\left(\mathbb{R}_{\geq 0}\right)^n} \left[I_T(t_0, \ldots, t_{n-1}) \cdot \sum_{A_{0 \ldots n-1}} \sum_{S_{0 \ldots n}} hw(\mu_0, D, \pi) \right] \eta_{S_0}(dt_0) \ldots \eta_{S_{n-1}}(dt_{n-1}). \\ \text{(as above)} &= \int_{\left(\mathbb{R}_{\geq 0}\right)^n} \left[I_T(t_0, \ldots, t_{n-1}) \cdot \sum_{A_{0 \ldots n-1}} hw\left(\mu_0/_{\sim}, \widetilde{D}_{\mu_0}, \widetilde{\pi}\right) \right] \eta_{S_0}(dt_0) \ldots \eta_{S_{n-1}}(dt_{n-1}) \\ \text{(sums over 1 element)} &= \int_{\left(\mathbb{R}_{\geq 0}\right)^n} \left[I_T(t_0, \ldots, t_{n-1}) \cdot \sum_{A_{0 \ldots n-1}} \sum_{\{S_0\} \times \ldots \times \{S_n\}} hw\left(\mu_0/_{\sim}, \widetilde{D}_{\mu_0}, \widetilde{\pi}\right) \right] \eta_{S_0}(dt_0) \ldots \eta_{S_{n-1}}(dt_{n-1}) \\ \text{(by Lemma 3.8)} &= \Pr_{\mu_0/_{\sim}, \widetilde{D}_{\mu_0}}\left(\widetilde{\Pi}\right). \end{aligned}$$

Note that in the last step we used the fact that $\widetilde{\Pi}$ is a simple bisimulation-closed set of finite paths in $MA/_{\sim}$. □

4 Preservation of minimum expected time under strong bisimulation

In this chapter we prove that minimum expected time is preserved under strong bisimulation. At this point, however, the reader might ask the following question: since in the following chapters we are planning to prove an analogous result for weaker bisimulations (naive weak bisimulation, for example), is the present proof not redundant? To see why it is not the case, suppose we have proved the following statement: if $G \subseteq S$ is closed under $\asymp$ (in other words, G is the union of zero or more equivalence classes under $\asymp$), then, for every $s, s' \in S$, $s \asymp s'$ implies $eT^{min}(s, \Diamond G) = eT^{min}(s', \Diamond G)$. Now, let us try to prove the analogous statement for $\sim$.

Proof attempt. Suppose $G \subseteq S$ is closed under $\sim$ and $s \sim s'$. Then $s \asymp s'$, and therefore $eT^{min}(s, \Diamond G) = eT^{min}(s', \Diamond G)$. QED

Unfortunately, this is wrong for the following reason: the fact that G is closed under $\sim$ does *not* imply that G is closed under $\asymp$, so that we cannot apply the result for naive weak bisimulation. Indeed, it is easy to construct a relation that is stronger than naive weak bisimulation and for which the preservation result does not hold. For example, consider Figure 9 and the equivalence $\fallingdotseq$ defined by the equivalence classes $\{s, t\}, \{u\}, \{v\}$. Clearly, $\fallingdotseq$ is even stronger than strong bisimulation, and yet $eT^{min}(s, \Diamond\{u\}) \neq eT^{min}(t, \Diamond\{u\})$.

Thus, it is indeed necessary to treat each kind of bisimulation separately. Furthermore, the proof presented in this chapter is relatively short and illustrates the usefulness of the fixpoint characterization of minimum expected time.

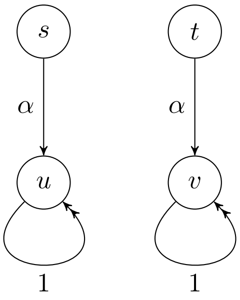


Figure 9: $eT^{min}(s, \Diamond\{u\}) \neq eT^{min}(t, \Diamond\{u\})$.

Having justified the necessity of this chapter, we turn our attention to formal proofs. Throughout the chapter, let $MA = (S, Act, \mapsto, \Rightarrow)$ be a semi-closed MA.

Lemma 4.1 (Simplified characterisation of MET as least fixpoint).

Let the function

$$MST : S \mapsto \mathbb{R}_{\geq 0}$$

be defined as follows:

$$MST(s) \quad = \quad \begin{cases} \frac{1}{E(s)} & \textit{if } s \in MS \\ 0 & \textit{otherwise.} \end{cases}$$

Now, for every $G \subseteq S$, the function

$$eT^{min}(\cdot, \Diamond G) : S \mapsto [0, \infty]$$

is the unique fixpoint of the Bellman operator

$$[L(v)](s) \quad = \quad \begin{cases} MST(s) + \min\limits_{\alpha \in Act^{\intercal}(s)} \left\{ \sum\limits_{s' \in S} \mu_{s,\alpha}(s') \cdot v(s') \right\} & \textit{if } s \in S \setminus G \\ 0 & \textit{if } s \in G. \end{cases}$$

Proof. Follows directly from Lemma 2.49 and the definition of MST. □

Theorem 4.2 (Strong bisimulation preserves minimum expected time).
Suppose $G \subseteq S$ is closed under $\sim$ (in other words, G is the union of zero or more equivalence classes under $\sim$). Then, for every $s, s' \in S$,

$$s \sim s' \quad \textit{implies} \quad eT^{min}(s, \Diamond G) = eT^{min}(s', \Diamond G).$$

Proof. First of all, in the trivial case when $G = \emptyset$ it holds that $eT^{min}(s, \Diamond G) = \infty$ for all $s \in S$, so that the claim obviously holds. Therefore, assume $G \neq \emptyset$.

Let $\widetilde{G} = G/_{\sim}$. Then, by Lemma 4.1, the function $eT^{min}\left(\cdot, \Diamond \widetilde{G}\right)$ is the unique fixpoint of the following operator:

$$\left[\widetilde{L}(v)\right](C) \quad = \quad \begin{cases} \widetilde{MST}(C) + \min\limits_{\alpha \in Act^{\intercal}(C)} \left\{ \sum\limits_{C' \in S/_{\sim}} \widetilde{\mu}_{C,\alpha}(C') \cdot v(C') \right\} & \text{if } C \in S/_{\sim} \setminus \widetilde{G} \\ 0 & \text{if } C \in \widetilde{G}, \end{cases}$$

where the tilde in $\widetilde{MST}$ and $\widetilde{\mu}$ signifies that those values refer to the quotient automaton $MA/_{\sim}$. Next, let

$$met : S \mapsto [0, \infty]$$

be defined as

$$met\left(s\right) \quad = \quad eT^{min}\left(\left[s\right]_{\sim}, \Diamond\widetilde{G}\right).$$

We now prove that *met* is the fixpoint of the following operator:

$$\left[L\left(v\right)\right]\left(s\right) \quad = \quad \begin{cases} MST\left(s\right) + \min\limits_{\alpha \in Act^{\intercal}\left(s\right)} \left\{ \sum\limits_{s' \in S} \mu_{s,\alpha}\left(s'\right) \cdot v\left(s'\right) \right\} & \text{if } s \in S \setminus G \\ 0 & \text{if } s \in G. \end{cases}$$

To this end, let $s \in S$, and consider two cases:

Case 1. If $s \in S \setminus G$, then $\left[s\right]_{\sim} \in S/_{\sim} \setminus \widetilde{G}$. Therefore,

$$\begin{aligned}
\left[L\left(met\right)\right]\left(s\right) &= MST\left(s\right) + \min_{\alpha \in Act^{\intercal}\left(s\right)} \left\{ \sum_{s' \in S} \mu_{s,\alpha}\left(s'\right) \cdot met\left(s'\right) \right\} \\
\text{(sum over eq. cl. separately)} &= MST\left(s\right) + \min_{\alpha \in Act^{\intercal}\left(s\right)} \left\{ \sum_{C' \in S/_{\sim}} \sum_{s' \in C'} \mu_{s,\alpha}\left(s'\right) \cdot met\left(s'\right) \right\} \\
\text{(by definition of } met\text{)} &= MST\left(s\right) + \min_{\alpha \in Act^{\intercal}\left(s\right)} \left\{ \sum_{C' \in S/_{\sim}} \sum_{s' \in C'} \mu_{s,\alpha}\left(s'\right) \cdot eT^{min}\left(C', \Diamond\widetilde{G}\right) \right\} \\
\left(\text{moving } eT^{min}\left(C', \Diamond\widetilde{G}\right) \text{ out}\right) &= MST\left(s\right) + \min_{\alpha \in Act^{\intercal}\left(s\right)} \left\{ \sum_{C' \in S/_{\sim}} eT^{min}\left(C', \Diamond\widetilde{G}\right) \sum_{s' \in C'} \mu_{s,\alpha}\left(s'\right) \right\} \\
\left(\mu_{s,\alpha}\left(C'\right) = \sum_{s' \in C'} \mu_{s,\alpha}\left(s'\right)\right) &= MST\left(s\right) + \min_{\alpha \in Act^{\intercal}\left(s\right)} \left\{ \sum_{C' \in S/_{\sim}} eT^{min}\left(C', \Diamond\widetilde{G}\right) \mu_{s,\alpha}\left(C'\right) \right\} \\
\text{(by Lemma 3.3 \& Definition 4.1)} &= MST\left(\left[s\right]_{\sim}\right) + \min_{\alpha \in Act^{\intercal}\left(\left[s\right]_{\sim}\right)} \left\{ \sum_{C' \in S/_{\sim}} eT^{min}\left(C', \Diamond\widetilde{G}\right) \widetilde{\mu}_{\left[s\right]_{\sim},\alpha}\left(C'\right) \right\} \\
\left(\text{by definition of } \widetilde{L}\right) &= \left[\widetilde{L}\left(eT^{min}\left(\cdot, \Diamond\widetilde{G}\right)\right)\right]\left(\left[s\right]_{\sim}\right) \\
\left(eT^{min}\left(\cdot, \Diamond\widetilde{G}\right) \text{ is fixpoint of } \widetilde{L}\right) &= eT^{min}\left(\left[s\right]_{\sim}, \Diamond\widetilde{G}\right) \\
\text{(by definition of } met\text{)} &= met\left(s\right).
\end{aligned}$$

Case 2. If $s \in G$, then $\left[s\right]_{\sim} \in \widetilde{G}$, so that

$$\begin{aligned}
\left[L\left(met\right)\right]\left(s\right) &= 0 \\
&= \left[\widetilde{L}\left(eT^{min}\left(\cdot, \Diamond\widetilde{G}\right)\right)\right]\left(\left[s\right]_{\sim}\right) \\
&= eT^{min}\left(\left[s\right]_{\sim}, \Diamond\widetilde{G}\right) \\
&= met\left(s\right).
\end{aligned}$$

Therefore, by Lemma 4.1, $eT^{min}\left(s, \Diamond G\right) = met\left(s\right) = eT^{min}\left(\left[s\right]_{\sim}, \Diamond\widetilde{G}\right)$ for all $s \in S$. But then, for every s and s' such that $s \sim s'$, it holds that

$$\begin{aligned} eT^{min}\left(s, \Diamond G\right) &= eT^{min}\left(\left[s\right]_{\sim}, \Diamond \widetilde{G}\right) \\ &= eT^{min}\left(\left[s'\right]_{\sim}, \Diamond \widetilde{G}\right) \\ &= eT^{min}\left(s', \Diamond G\right). \end{aligned}$$

This concludes the proof. □

5 Preservation of minimum long run average under strong bisimulation

In the present chapter, we shall prove that strong bisimulation preserves minimum long run average. Everything that was said at the beginning of Chapter 4 equally applies to minimum long run average, so that we cannot derive the main theorem of this chapter from the analogous result for weaker bisimulations.

As in Chapter 3, we closely follow the approach of Martin Neuhäußer [11], with only a few differences:

1. We work with MAs instead of CTMDPs.
2. Our proofs are somewhat more detailed.
3. The results in [11] are more general: in particular, our Lemmas 5.1, 5.2 and 5.3 are a special case of Theorem 7.2 in [11] (note, however, that the proof of Theorem 7.2 in [11] only considers the simplest case and states that "the other cases are similar"; by contrast, we provide a full proof for our case).

We now presents the results formally. Let $MA = (S, Act, \mapsto, \Rightarrow)$ be a semi-closed MA.

Lemma 5.1 (nth state at time t is measurable). *For each $t \in \mathbb{R}_{\geq 0}$ and $n \in \mathbb{N}_{\geq 0}$, the set*

$$\mathfrak{T}_t^n \quad = \quad \left\{ (t_0, ..., t_n) \in (\mathbb{R}_{\geq 0})^{n+1} \mid \sum_{i=0}^{n-1} t_i \leq t < \sum_{i=0}^{n} t_i \right\}$$

is Borel-measurable (in the case of $n = 0$, we set $\sum_{i=0}^{-1} t_i = 0$).

Proof. First of all, if $n = 0$, then

$$\mathfrak{T}_t^0 \quad = \quad \{ t_0 \in \mathbb{R}_{\geq 0} \mid t < t_0 \}$$

is clearly Borel-measurable.

Therefore, let us focus on the case when $n > 0$. We will show that

$$\mathfrak{T}_t^n \quad = \quad \bigcap_{\varepsilon \in \mathbb{Q}_{>0}} \quad \bigcup_{\substack{c_i, d_i \in \mathbb{Q}_{\geq 0} \\ \sum_{i=0}^{n} c_i > t \\ \sum_{i=0}^{n-1} d_i < t + \varepsilon \\ c_i < d_i}} \quad \left(\prod_{i=0}^{n-1} [c_i, d_i] \right) \times [c_n, \infty) \, .$$

$\subseteq$: If $(t_0, ..., t_n) \in \mathfrak{T}_t^n$, then we can find rational numbers $c_0, ..., c_n$ that approximate $t_0, ..., t_n$ from below as closely as we please. Consequently, $\sum_{i=0}^{n} c_i$ approximates $\sum_{i=0}^{n} t_i$ from below arbitrarily closely. Now, since $t < \sum_{i=0}^{n} t_i$, we can find $c_0, ..., c_n$ with $\sum_{i=0}^{n} c_i > t$.

Furthermore, for every $\varepsilon \in \mathbb{Q}_{>0}$, $\sum_{i=0}^{n-1} t_i < t + \varepsilon$. By a similar reasoning, we can find $d_0, ..., d_{n-1}$ such that each d_i approximates t_i from above arbitrarily closely, and, consequently, $\sum_{i=0}^{n-1} d_i$ approximates $\sum_{i=0}^{n-1} t_i$. Then we can find such $d_0, ..., d_{n-1}$ that $\sum_{i=0}^{n-1} d_i < t + \varepsilon$.

$\supseteq$: If

$$(t_0, ..., t_n) \quad \in \quad \bigcap_{\varepsilon \in \mathbb{Q}_{>0}} \quad \bigcup_{\substack{c_i, d_i \in \mathbb{Q}_{\geq 0} \\ \sum_{i=0}^{n} c_i > t \\ \sum_{i=0}^{n-1} d_i < t + \varepsilon \\ c_i < d_i}} \quad \left(\prod_{i=0}^{n-1} [c_i, d_i] \right) \times [c_n, \infty),$$

there must exist $c_0, ..., c_n$ such that $t < \sum_{i=0}^{n} c_i$ and $\sum_{i=0}^{n} c_i \leq \sum_{i=0}^{n} t_i$. But then $t < \sum_{i=0}^{n} t_i$.

Moreover, for each $\varepsilon \in \mathbb{Q}_{>0}$, there must exist $d_0, ..., d_{n-1}$ with $\sum_{i=0}^{n-1} t_i \leq \sum_{i=0}^{n-1} d_i < t + \varepsilon$. But the fact that the inequality $\sum_{i=0}^{n-1} t_i < t + \varepsilon$ is true for every $\varepsilon \in \mathbb{Q}_{>0}$ implies that $\sum_{i=0}^{n-1} t_i \leq t$. $\square$

Lemma 5.2. *Let $S_0, ..., S_{n+1} \in S/_\sim$ and let $t \in \mathbb{R}_{\geq 0}$. Then the set*

$$\mathfrak{P}_t^{S_0, ..., S_{n+1}} \quad = \quad \left\{ \pi \in S_0 \times Act^{\mathsf{T}} \times \mathbb{R}_{\geq 0} \times ... \times S_{n+1} \mid \left(\sum_{i=0}^{n-1} t_i \right) \leq t < \left(\sum_{i=0}^{n} t_i \right) \right\}$$

is measurable.

Furthermore, let

$$\widetilde{\mathfrak{P}}_t^{S_0, ..., S_{n+1}} \quad = \quad \left\{ \pi \in \{S_0\} \times Act^{\mathsf{T}} \times \mathbb{R}_{\geq 0} \times ... \times \{S_{n+1}\} \mid \left(\sum_{i=0}^{n-1} t_i \right) \leq t < \left(\sum_{i=0}^{n} t_i \right) \right\}$$

in the quotient automaton $MA/_\sim$. Then, for every scheduler $D \in GM$, every initial distribution μ_0 over S and every quotient scheduler $\widetilde{D}_{\mu_0}$, it holds that

$$\Pr_{\mu_0, D} \left(\mathfrak{P}_t^{S_0, ..., S_{n+1}} \right) \quad = \quad \Pr_{\mu_0/_\sim, \widetilde{D}_{\mu_0}} \left(\widetilde{\mathfrak{P}}_t^{S_0, ..., S_{n+1}} \right).$$

Proof. By Lemma 5.1, the set

$$\mathfrak{T}_t^n \quad = \quad \left\{ (t_0, ..., t_n) \in (\mathbb{R}_{\geq 0})^{n+1} \mid \sum_{i=0}^{n-1} t_i \leq t < \sum_{i=0}^{n} t_i \right\}$$

is Borel-measurable.

Then,

$$\mathfrak{P}_t^{S_0,\ldots,S_{n+1}} \quad = \quad \{\pi \in S_0 \times Act^{\tau} \times \mathbb{R}_{\geq 0} \times \ldots \times S_{n+1} \mid (t_0, \ldots, t_n) \in \mathfrak{T}_t^n\}$$

is a bisimulation-closed set of finite paths, and, analogously,

$$\widetilde{\mathfrak{P}}_t^{S_0,\ldots,S_{n+1}} \quad = \quad \{\pi \in \{S_0\} \times Act^{\tau} \times \mathbb{R}_{\geq 0} \times \ldots \times \{S_{n+1}\} \mid (t_0, \ldots, t_n) \in \mathfrak{T}_t^n\}\,.$$

Thus, by Theorem 3.9,

$$\Pr_{\mu_0,D}\left(\mathfrak{P}_t^{S_0,\ldots,S_{n+1}}\right) \quad = \quad \Pr_{\mu_0/_\sim,\widetilde{D}_{\mu_0}}\left(\widetilde{\mathfrak{P}}_t^{S_0,\ldots,S_{n+1}}\right),$$

as desired. $\square$

Lemma 5.3. *Suppose $G \subseteq S$ is the union of one or more equivalence classes under $\sim$ and $t \in \mathbb{R}_{\geq 0}$. Assume that $G \subseteq MS$. Then the set*

$$\mathfrak{P}_t^G \quad = \quad \{\pi \in Paths^{\omega} \mid \pi@t \,\cap\, G \neq \emptyset\}$$

is measurable.

Furthermore, let

$$\widetilde{\mathfrak{P}}_t^G \quad = \quad \{\pi \in Paths^{\omega} \mid \pi@t \,\cap\, (G/_\sim) \neq \emptyset\}$$

be the corresponding set of paths in the quotient automaton $MA/_\sim$. Then, for every scheduler $D \in GM$, every initial distribution μ_0 over S and every quotient scheduler $\widetilde{D}_{\mu_0}$, it holds that

$$\Pr_{\mu_0,D}\left(\mathfrak{P}_t^G\right) \quad = \quad \Pr_{\mu_0/_\sim,\widetilde{D}_{\mu_0}}\left(\widetilde{\mathfrak{P}}_t^G\right).$$

Proof. First, observe that

$$\mathfrak{P}_t^G \quad = \quad \biguplus_{C \in G/_\sim} \biguplus_{n=0}^{\infty} \left\{\pi \in Paths^{\omega} \mid \left(\sum_{i=0}^{n-1} t_i\right) \leq t < \left(\sum_{i=0}^{n} t_i\right) \wedge \pi[n] \in C\right\},$$

where t_i is the amount of time π spends in the state $\pi[i]$. Note that the union is disjoint since a path can traverse only one Markovian state at time t (recall that a path is not allowed to execute a timed transition immediately, but must wait for some positive amount of time).

Next, let

$$\begin{aligned}
\mathfrak{P}_t^{n,C} &= \left\{\pi \in Paths^{n+1} \mid \left(\sum_{i=0}^{n-1} t_i\right) \leq t < \left(\sum_{i=0}^{n} t_i\right) \wedge \pi\,[n] \in C\right\} \\
&= \biguplus_{\substack{(S/_\sim)^{n+2} \\ S_n = C}} \left\{\pi \in S_0 \times Act^{\mathsf{T}} \times \mathbb{R}_{\geq 0} \times ... \times S_{n+1} \mid \left(\sum_{i=0}^{n-1} t_i\right) \leq t < \left(\sum_{i=0}^{n} t_i\right)\right\} \\
&= \biguplus_{\substack{(S/_\sim)^{n+2} \\ S_n = C}} \mathfrak{P}_t^{S_0,\ldots,S_{n+1}},
\end{aligned}$$

where the sets $\mathfrak{P}_t^{S_0,\ldots,S_{n+1}}$ are as in Lemma 5.2.

Now, it is not difficult to see that

$$\begin{aligned}
\mathfrak{P}_t^{G} &= \biguplus_{C \in G/_\sim} \biguplus_{n=0}^{\infty} Cyl\left(\mathfrak{P}_t^{n,C}\right) \\
&= \biguplus_{C \in G/_\sim} \biguplus_{n=0}^{\infty} \biguplus_{\substack{(S/_\sim)^{n+2} \\ S_n = C}} Cyl\left(\mathfrak{P}_t^{S_0,\ldots,S_{n+1}}\right),
\end{aligned}$$

and, analogously,

$$\widetilde{\mathfrak{P}}_t^{G} = \biguplus_{C \in G/_\sim} \biguplus_{n=0}^{\infty} \biguplus_{\substack{(S/_\sim)^{n+2} \\ S_n = C}} Cyl\left(\widetilde{\mathfrak{P}}_t^{S_0,\ldots,S_{n+1}}\right).$$

Therefore,

$$\begin{aligned}
\Pr_{\mu_0,D}\left(\mathfrak{P}_t^{G}\right) &= \sum_{C \in G/_\sim} \sum_{n=0}^{\infty} \sum_{\substack{(S/_\sim)^{n+2} \\ S_n = C}} \Pr_{\mu_0,D}\left(\mathfrak{P}_t^{S_0,\ldots,S_{n+1}}\right) \\
\text{(by Lemma 5.2)} \quad &= \sum_{C \in G/_\sim} \sum_{n=0}^{\infty} \sum_{\substack{(S/_\sim)^{n+2} \\ S_n = C}} \Pr_{\mu_0/_\sim,\widetilde{D}_{\mu_0}}\left(\widetilde{\mathfrak{P}}_t^{S_0,\ldots,S_{n+1}}\right) \\
&= \Pr_{\mu_0/_\sim,\widetilde{D}_{\mu_0}}\left(\widetilde{\mathfrak{P}}_t^{G}\right).
\end{aligned}$$

□

Lemma 5.4 (Stationary scheduler on $MA/_{\sim}$ is quotient scheduler). *Let $\widetilde{D}$ be a stationary scheduler on $MA/_{\sim}$ and let $\mu_0 \in Dist(S)$. Define the stationary scheduler D on MA as follows: for every $s \in S$ and $\alpha \in Act^{\intercal}$,*

$$D(s, \alpha) \quad = \quad \widetilde{D}\left([s]_{\sim}, \alpha\right).$$

Then, $\widetilde{D}$ is a quotient scheduler with respect to D and μ_0.

Proof. We need to prove that if

$$\widetilde{\pi} \quad = \quad [s_0]_{\sim} \xrightarrow{\alpha_0, t_0} [s_1]_{\sim} \xrightarrow{\alpha_1, t_1} \dots \xrightarrow{\alpha_{n-1}, t_{n-1}} [s_n]_{\sim}$$

is a path in $MA/_{\sim}$ with $\sum_{\pi \in \Pi} hw(\mu_0, D, \pi) \neq 0$, then

$$\widetilde{D}(\widetilde{\pi}, \alpha) \quad = \quad \frac{\displaystyle\sum_{\pi \in \Pi} hw(\mu_0, D, \pi) \cdot D(\pi, \alpha)}{\displaystyle\sum_{\pi \in \Pi} hw(\mu_0, D, \pi)},$$

where

$$\Pi \quad = \quad [s_0]_{\sim} \times \{\alpha_0\} \times \{t_0\} \times [s_1]_{\sim} \times \dots \times [s_n]_{\sim}$$

is the rectangle in MA corresponding to $\widetilde{\pi}$.

Now:

$$\begin{aligned}
\frac{\displaystyle\sum_{\pi \in \Pi} hw(\mu_0, D, \pi) \cdot D(\pi, \alpha)}{\displaystyle\sum_{\pi \in \Pi} hw(\mu_0, D, \pi)} &= \frac{\displaystyle\sum_{\pi \in \Pi} hw(\mu_0, D, \pi) \cdot D(\pi_{\downarrow}, \alpha)}{\displaystyle\sum_{\pi \in \Pi} hw(\mu_0, D, \pi)} \\
\text{(by definition of } D\text{)} &= \frac{\displaystyle\sum_{\pi \in \Pi} hw(\mu_0, D, \pi) \cdot \widetilde{D}\left([\pi_{\downarrow}]_{\sim}, \alpha\right)}{\displaystyle\sum_{\pi \in \Pi} hw(\mu_0, D, \pi)} \\
\left([\pi_{\downarrow}]_{\sim} = \widetilde{\pi}_{\downarrow} \text{ for all } \pi \in \Pi\right) &= \frac{\displaystyle\sum_{\pi \in \Pi} hw(\mu_0, D, \pi) \cdot \widetilde{D}(\widetilde{\pi}_{\downarrow}, \alpha)}{\displaystyle\sum_{\pi \in \Pi} hw(\mu_0, D, \pi)} \\
\left(\text{moving } \widetilde{D}(\widetilde{\pi}_{\downarrow}, \alpha) \text{ out}\right) &= \widetilde{D}(\widetilde{\pi}_{\downarrow}, \alpha) \cdot \frac{\displaystyle\sum_{\pi \in \Pi} hw(\mu_0, D, \pi)}{\displaystyle\sum_{\pi \in \Pi} hw(\mu_0, D, \pi)} \\
&= \widetilde{D}(\widetilde{\pi}_{\downarrow}, \alpha) \\
\left(\widetilde{D} \text{ is stationary}\right) &= \widetilde{D}(\widetilde{\pi}, \alpha),
\end{aligned}$$

as desired. □

Theorem 5.5 (Strong bisimulation preserves minimum long run average). *Suppose $G \subseteq S$ is closed under $\sim$ (in other words, G is a union of zero or more equivalence classes under $\sim$). Then, for every $s, s' \in S$,*

$$s \sim s' \quad \textit{implies} \quad LRA^{min}(s,G) = LRA^{min}(s',G).$$

Proof. First of all, in the trivial case when $G = \emptyset$, $LRA^{min}(s,G) = 0$ for all $s \in S$. Therefore, assume $G \neq \emptyset$.

Next, note that $LRA^{min}(s,G) = LRA^{min}(s, G \cap MS)$. Therefore, without loss of generality, suppose $G \subseteq MS$, and let $\widetilde{G} = G/_\sim$.

Let $s \in S$. We will show that

$$LRA^{min}(s,G) \quad = \quad LRA^{min}\left([s]_\sim, \widetilde{G}\right),$$

where the right-hand side is determined in $MA/_\sim$.

First of all, by Lemma 2.52, there exists a scheduler $D \in GM$ on MA such that $LRA^{min}(s,G) = LRA^{D}(s,G)$. Let $\widetilde{D}_s$ be an quotient scheduler with respect to D and the initial distribution $\{(s,1)\}$ (note that such a quotient scheduler always exists). Then:

$$\begin{aligned}
LRA^{min}(s,G) &= LRA^{D}(s,G) \\
\text{(by Definition 2.51)} &= \int_{Paths^\omega} \left(\lim_{t\to\infty} \frac{1}{t}\int_0^t I_G(\pi@u)\ du\right) \Pr_{s,D}(d\pi) \\
\text{(rearranging limit and integrals *)} &= \lim_{t\to\infty} \frac{1}{t}\int_0^t \int_{Paths^\omega} I_G(\pi@u)\ \Pr_{s,D}(d\pi)\ du \\
&= \lim_{t\to\infty} \frac{1}{t}\int_0^t \Pr_{s,D}\{\pi \in Paths^\omega \mid (\pi@u\ \cap\ G) \neq \emptyset\}\ du \\
\text{(by Lemma 5.3)} &= \lim_{t\to\infty} \frac{1}{t}\int_0^t \Pr_{[s]_\sim, \widetilde{D}_s}\left\{\pi \in Paths^\omega \mid \left(\pi@u\ \cap\ \widetilde{G}\right) \neq \emptyset\right\}\ du \\
&= LRA^{\widetilde{D}_s}\left([s]_\sim, \widetilde{G}\right) \\
&\geq LRA^{min}\left([s]_\sim, \widetilde{G}\right).
\end{aligned}$$

(∗) The validity of this step is not immediately obvious: it is taken directly from [11].

On the other hand, according to Lemma 2.52, there exists a stationary scheduler $\widetilde{D}'_s$ on $MA/_\sim$ with $LRA^{min}\left([s]_\sim, \widetilde{G}\right) = LRA^{\widetilde{D}'_s}\left([s]_\sim, \widetilde{G}\right)$. Let D be the scheduler on MA defined as follows: for every $\pi \in Paths^*$ and $\alpha \in Act^{\mathsf{T}}$,

$$D(\pi,\alpha) \quad = \quad \widetilde{D}'_s\left([\pi{\downarrow}]_\sim, \alpha\right).$$

By Lemma 5.4, $\widetilde{D}'_s$ is a quotient scheduler with respect to D and $\{(s,1)\}$. Therefore:

$$\begin{aligned}
LRA^{min}\left([s]_\sim, \widetilde{G}\right) &= LRA^{\widetilde{D}'_s}\left([s]_\sim, \widetilde{G}\right) \\
\text{(by Definition 2.51 and rearranging)} &= \lim_{t\to\infty} \frac{1}{t}\int_0^t \Pr_{[s]_\sim, \widetilde{D}'_s}\left\{\pi \in Paths^\omega \mid \left(\pi@u \cap \widetilde{G}\right) \neq \emptyset\right\} du \\
\text{(by Lemma 5.3)} &= \lim_{t\to\infty} \frac{1}{t}\int_0^t \Pr_{s,D}\left\{\pi \in Paths^\omega \mid (\pi@u \cap G) \neq \emptyset\right\} du \\
&= LRA^{D}(s,G) \\
&\geq LRA^{min}(s,G).
\end{aligned}$$

Thus, for every $s, s' \in S$ such that $s \sim s'$ it holds that

$$\begin{aligned}
LRA^{min}(s,G) &= LRA^{min}\left([s]_\sim, \widetilde{G}\right) \\
&= LRA^{min}\left([s']_\sim, \widetilde{G}\right) \\
&= LRA^{min}(s',G).
\end{aligned}$$

□

6 Computations

In this chapter we introduce computations (the general idea and the term "computation" were inspired by [3]).

The computation is a generalization of the transition tree (the reader might find it helpful to refer to Figure 10 for an example). The main differences between computations and transition trees are as follows:

1. A computation node is allowed to have any finite number of outgoing transitions, while a transition tree node can have at most one.
2. A consequence of the previous point is that there is a probability distribution over the outgoing transitions of every node (unless the node has zero of them).
3. We introduce special ϵ-transitions, which become useful in the context of weak bisimulations. Recall that a state can simulate a τ-transition of a weakly bisimilar state by doing nothing: this action of doing nothing can be represented in a computation by an ϵ-transition.

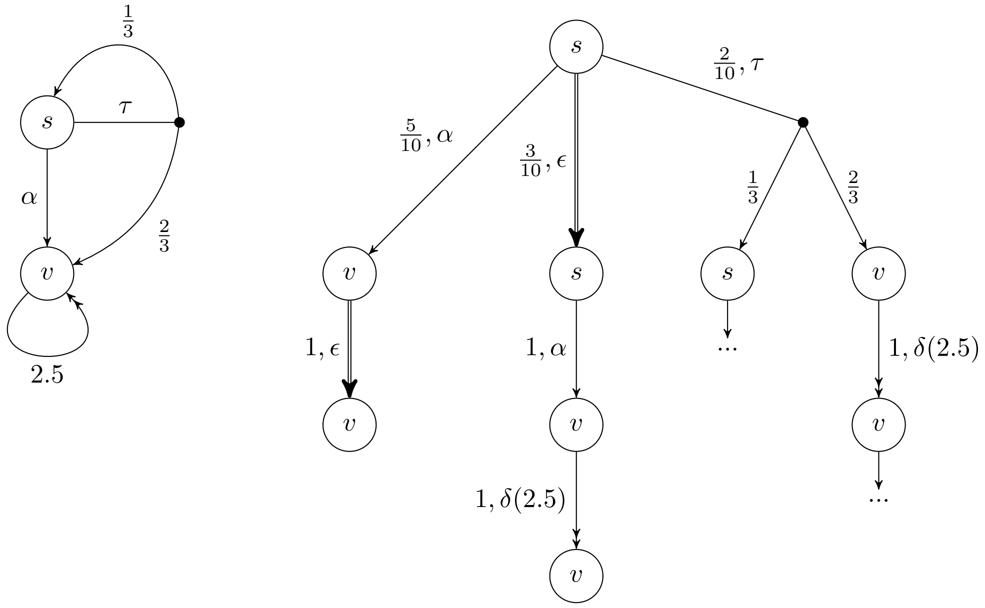


Figure 10: An MA (left) and one of the possible computations from the state s (right).

In order to understand the necessity of generalizing transition trees in this way, suppose $s \asymp s'$ and there is a transition from s, which s' can simulate by a linear combination of transition trees. But it is quite clear that this combination of transition trees is, in fact, a computation (see Figure 11)

Now, suppose the transition from s is a τ-transition and some of the corresponding transition trees from s' consist of a single node, which is possible if some of the states in the target distribution are bisimilar to s and s' (see Figure 12) We would, of course, still like to combine the trees into a computation. There are several options:

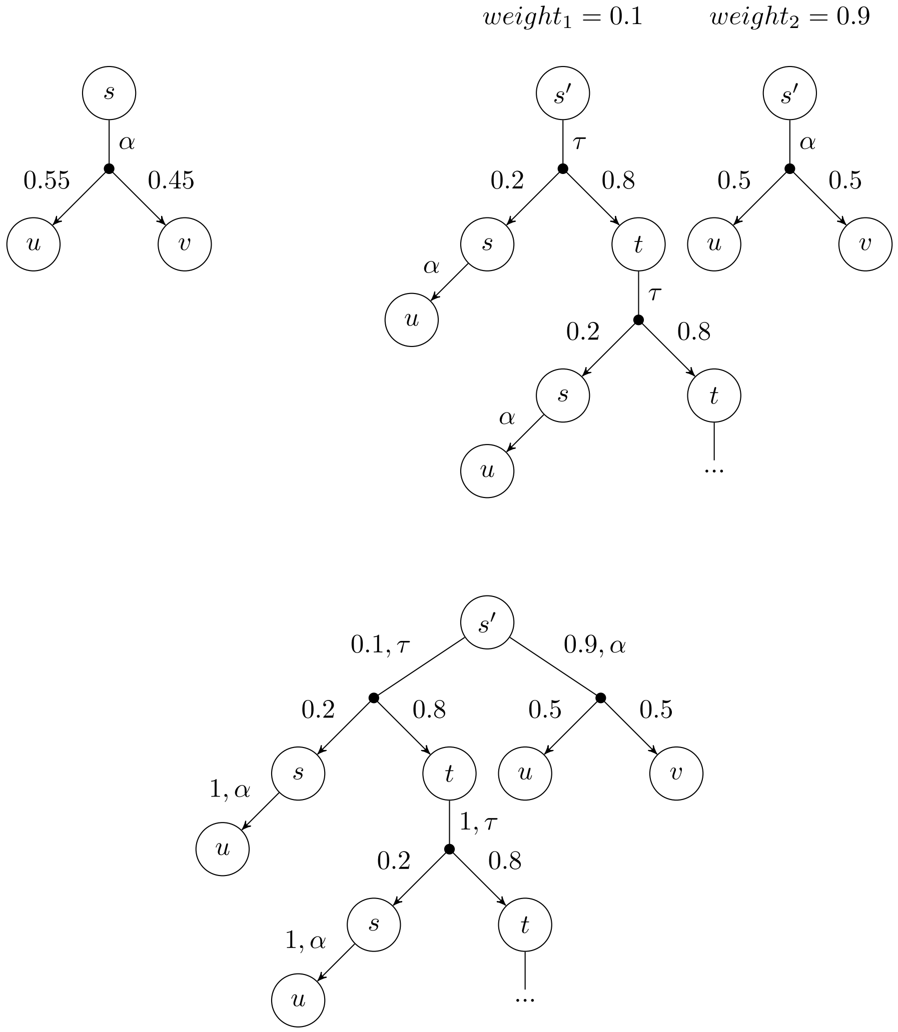


Figure 11: A transition (top left), simulated by a combination of transition trees (above right), which can be converted into a computation (bottom).

1. We could allow the probabilities of the outgoing transitions of the root node sum up to something less than unity. Then, if the probabilities sum up to some value $p < 1$, we interpret it as "stay where you are with probability $1 - p$".
2. We could still require the probabilities to sum up to unity, but introduce an ϵ-transition whose probability is $1 - p$, with the same interpretation.

For various reasons, we decided to adopt the second variant. The result of combining the transition trees into a computation is presented in Figure 12.

Throughout this chapter, let $MA = (S, Act, \mapsto, \Rightarrow)$ be a semi-closed MA.

6.1 Motivation

Recall that our ultimate goal is to prove that such state properties as minimum expected time and minimum long run average are preserved under various kinds of weak bisimulations. To be specific,

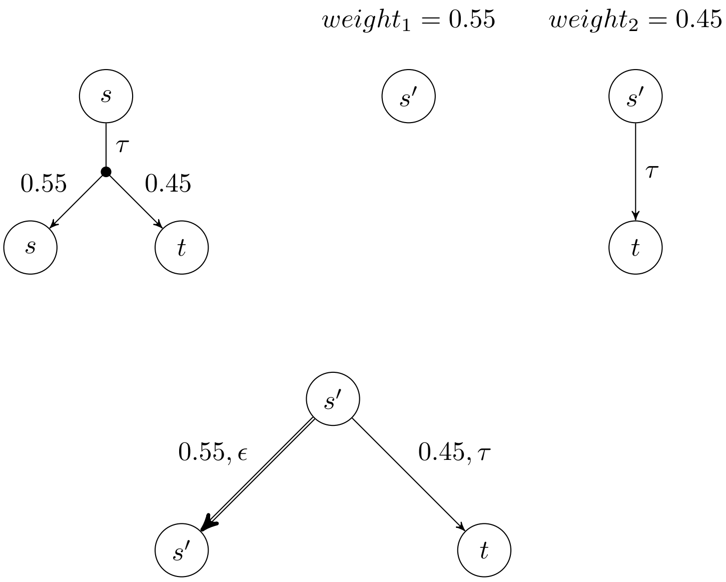


Figure 12: A τ-transition (top left), simulated by a combination of transition trees (above right), which can be converted into a computation involving ϵ-transitions (bottom).

let us consider minimum expected time and naive weak bisimulation. Suppose $s \asymp s'$ and $G \subseteq S$ is closed under $\asymp$. We need to prove that

$$eT^{min}(s, \Diamond G) \quad = \quad eT^{min}(s', \Diamond G).$$

In fact, due to the symmetry of $\asymp$ it is enough to prove that

$$eT^{min}(s, \Diamond G) \quad \geq \quad eT^{min}(s', \Diamond G).$$

Let us now think of a possible approach to the proof. First recall that by Lemma 2.50, there exists a stationary deterministic scheduler D_s with

$$eT^{min}(s, \Diamond G) \quad = \quad eT^{D_s}(s, \Diamond G).$$

Now, we could prove the result by constructing a scheduler $D_{s'}$ (not necessarily stationary or deterministic) such that

$$eT^{D_s}(s, \Diamond G) \quad \geq \quad eT^{D_{s'}}(s', \Diamond G).$$

Note that it is in this step where we have to use the assumptions that $s \asymp s'$ and G is closed under

$\asymp$. If we could do that, we would be able to write

$$\begin{aligned} eT^{min}(s, \Diamond G) &= eT^{D_s}(s, \Diamond G) \\ &\geq eT^{D_{s'}}(s', \Diamond G) \\ &\geq eT^{min}(s', \Diamond G). \end{aligned}$$

Unfortunately, it is not obvious how to construct $D_{s'}$. Here is where computations enter the picture:

1. It is intuitively clear that a time-abstract scheduler (in a combination with an initial state: in our case, s) induces a computation (which we can construct by unrolling the MA, as described in Section 6.3). For instance, the scheduler D_s induces some computation C_s from s.
2. C_s carries the same information as D_s. We can, therefore, compute $eT^{D_s}(s, \Diamond G)$ just by looking at C_s (by a procedure defined in Section 6.4). Let us denote the computed value by $eT^{C_s}(s, \Diamond G)$. Of course, $eT^{C_s}(s, \Diamond G) = eT^{D_s}(s, \Diamond G)$, but by using a different superscript symbol we emphasize that the expected time is computed by looking only at the computation C_s.
3. Furthermore, given a computation, we can reason backwards and find a time-abstract scheduler inducing this computation (the algorithm is also described in Section 6.3). For instance, if $C_{s'}$ is some computation from s', we can find a scheduler $D_{s'}$ inducing $C_{s'}$. Again, we can compute the expected time in $C_{s'}$ satisfying $eT^{C_{s'}}(s', \Diamond G) = eT^{D_{s'}}(s', \Diamond G)$.

Now, the task is reduced to the following: prove that given a computation C_s from s there exists a computation $C_{s'}$ from s' such that $eT^{C_s}(s, \Diamond G) \geq eT^{C_{s'}}(s', \Diamond G)$. Then,

$$\begin{aligned} eT^{min}(s, \Diamond G) &= eT^{D_s}(s, \Diamond G) \\ \text{(Step 1: construct } C_s \text{ from } D_s) &= eT^{C_s}(s, \Diamond G) \\ \text{(Step 2: construct } C_{s'} \text{ from } C_s) &\geq eT^{C_{s'}}(s', \Diamond G) \\ \text{(Step 3: construct } D_{s'} \text{ from } C_{s'}) &= eT^{D_{s'}}(s', \Diamond G) \\ &\geq eT^{min}(s', \Diamond G). \end{aligned}$$

At this point, the reader might doubt that constructing $C_{s'}$ with $eT^{C_s}(s, \Diamond G) \geq eT^{C_{s'}}(s', \Diamond G)$ is any easier than directly constructing $D_{s'}$ with $eT^{D_s}(s, \Diamond G) \geq eT^{D_{s'}}(s', \Diamond G)$. However, the former indeed turns out to be simpler, if only because computations can be visualized and reasoned about more easily than schedulers.

The chain of equalities and inequalities above specifies a proof plan consisting of 3 steps. In this chapter we will take care of Steps 1 and 3, and a part of Step 2. For Step 2, we will define several conditions that must be satisfied by $C_{s'}$ and prove that under those conditions it indeed holds that $eT^{C_s}(s, \Diamond G) \geq eT^{C_{s'}}(s', \Diamond G)$. In the next chapters we will complete Step 2 and the whole proof by showing that a computation $C_{s'}$ satisfying the conditions can indeed be constructed.

Finally, it should be noted that we will compose analogous proofs for minimum long run average and other weak bisimulations.

6.2 Computations

We now introduce computations formally. The section entirely consists of straightforward definitions: the results that make computations interesting and useful for our purposes will come in the next sections.

Definition 6.1 (Noop action). Let $Act^\epsilon = Act^\chi \cup \{\epsilon\}$, where ϵ is a pseudo-action signifying the fact that a state matches a τ-transition of a weakly bisimilar state by doing nothing.

Furthermore, for every $s \in S$, let $Act^\epsilon(s) = Act^\chi(s) \cup \{\epsilon\}$.

Definition 6.2 (Computation). A computation is a tuple $C = (Q, \rightarrow, L, \sigma_0)$ such that:

1. Q is a finite or countable set of nodes.
2. $L : Q \mapsto S$ is a node labeling function.
3. $\sigma_0 \in Q$ is the root node.
4. $\rightarrow \subseteq Q \times Act^\epsilon \times (0, 1] \times Dist(Q)$ is a transition relation, such that:
    (a) for each $(\sigma, \alpha, p, \mu) \in \rightarrow$, $\alpha \in Act^\epsilon(L(\sigma))$,
    (b) for every ϵ-transition $(\sigma, \epsilon, p, \mu) \in \rightarrow$ it holds that $\mu = \{(\sigma', 1)\}$ for some $\sigma' \in Q$ with $L(\sigma') = L(\sigma)$,
    (c) for each $\sigma \in Q$, the set of outgoing transitions

    $$\{(\sigma, \alpha, p, \mu) \in \{\sigma\} \times Act^\epsilon \times (0, 1] \times Dist(Q) \mid (\sigma, \alpha, p, \mu) \in \rightarrow\}$$

    is finite,
    (d) for each $\sigma \in Q$, $\sum_{(\sigma,\alpha,p,\mu) \in \rightarrow} p \in \{0, 1\}$,
    (e) for each $(\sigma, \alpha, p, \mu) \in \rightarrow$ with $\alpha \in Act^\chi$,

    $$\bigoplus_{\sigma' \in Q} \{(L(\sigma'), \mu(\sigma'))\} \quad = \quad \mu_{L(\sigma),\alpha},$$

    (f) C is a tree rooted in σ_0, i.e., there is no ingoing transition to σ_0, there is exactly one ingoing transition to every other $\sigma \in Q$, and every $\sigma \in Q$ is reachable from σ_0.

We say that C is a *computation from the state* $s \in S$ if $L(\sigma_0) = s$.

Definition 6.3 (Parents, children, leaves). Let $C = (Q, \rightarrow, L, \sigma_0)$ be a computation. For each node $\sigma' \in Q \setminus \{\sigma_0\}$, let $\mathit{Parent}(\sigma')$ be the unique $\sigma \in Q$ such that there exists a transition $(\sigma, \alpha, p, \mu) \in \rightarrow$ with $\sigma' \in \mathit{Supp}(\mu)$. Let us denote this transition by $\mathit{InTrans}(\sigma')$.

Furthermore, for each node $\sigma \in Q$, let

$$\mathit{Children}(\sigma) \quad = \quad \{\sigma' \in Q \mid \sigma = \mathit{Parent}(\sigma')\}.$$

Let $\mathit{Children}^*$ be the reflexive and transitive closure of $\mathit{Children}$.

Next, the set of *leaves* of C is defined as

$$\mathit{Leaves}(C) \quad = \quad \{\sigma \in Q \mid \mathit{Children}(\sigma) = \emptyset\}.$$

Definition 6.4 (Probabilities in computation). For a computation $C = (Q, \rightarrow, L, \sigma_0)$, let:

1. $\Pr(\sigma_0) = 1$, and
2. for every $\sigma \in Q \setminus \{\sigma_0\}$ with $\mathit{InTrans}(\sigma) = (\mathit{Parent}(\sigma), \alpha, p, \mu)$,

$$\Pr(\sigma) \quad = \quad \Pr(\mathit{Parent}(\sigma)) \cdot p \cdot \mu(\sigma).$$

Next, for each $\sigma \in Q \setminus \mathit{Leaves}(C)$ and $\alpha \in \mathit{Act}^{\epsilon}$, define

$$\Pr(\sigma, \alpha) \quad = \quad \sum_{(\sigma,\alpha,p,\mu) \in \rightarrow} p.$$

Finally, for each $\sigma \in Q \setminus \mathit{Leaves}(C)$, define

$$\Pr(\sigma, \mathsf{T}) \quad = \quad \begin{cases} \Pr(\sigma, \delta(E(L(\sigma)))) & L(\sigma) \in \mathit{MS} \\ 0 & \mathit{otherwise}. \end{cases}$$

Note that for every $\sigma \in Q \setminus \mathit{Leaves}(C)$ it holds that

$$\begin{aligned} \Pr(\sigma, \epsilon) \quad &= \quad 1 - \sum_{\alpha \in \mathit{Act}^{\mathsf{X}}(L(\sigma))} \Pr(\sigma, \alpha) \\ &= \quad 1 - \sum_{\alpha \in \mathit{Act}^{\mathsf{T}}(L(\sigma))} \Pr(\sigma, \alpha). \end{aligned}$$

Definition 6.5 (Noop-trees). Consider a computation $C = (Q, \rightarrow, L, \sigma_0)$. For every $\sigma \in Q$, let $Reach_\epsilon(\sigma)$ be the set of nodes reachable from σ via zero or more ϵ-transitions (note that $\sigma \in Reach_\epsilon(\sigma)$). Formally, $Reach_\epsilon(\sigma)$ is the smallest set such that:

1. $\sigma \in Reach_\epsilon(\sigma)$, and
2. whenever $\sigma' \in Reach_\epsilon(\sigma)$ and there exists a transition $(\sigma', \epsilon, p, \mu) \in \rightarrow$ with $\sigma'' \in Supp(\mu)$, it holds that $\sigma'' \in Reach_\epsilon(\sigma)$.

Furthermore, let

$$In_\epsilon \quad = \quad \{\sigma' \in Q \mid \exists (\sigma, \epsilon, p, \mu) \in \rightarrow \ . \ \sigma' \in Supp(\mu)\}$$

be the set of nodes of C whose incoming transitions are ϵ-transitions.

Definition 6.6 (Non-terminating computations). Let $C = (Q, \rightarrow, L, \sigma_0)$ be a computation. We call C *non-terminating* if:

1. $Leaves(C) = \emptyset$ and
2. for every $\sigma \in Q$, the probability to execute only ϵ-transitions forever is 0, that is, for every $\sigma \in Q$,

$$\sum_{\sigma' \in Reach_\epsilon(\sigma)} \Pr(\sigma') \cdot (1 - \Pr(\sigma', \epsilon)) \quad = \quad \Pr(\sigma)\,.$$

Definition 6.7 (Independent set of nodes). Consider a computation $C = (Q, \rightarrow, L, \sigma_0)$. A set of nodes $B \subseteq Q$ is said to be *independent* if for every pair of distinct $\sigma, \sigma' \in B$ it holds that

$$\sigma' \quad \notin \quad Children^*(\sigma)\,.$$

Definition 6.8 (Subdistribution induced by subset of computation nodes). Let $C = (Q, \rightarrow, L, \sigma_0)$ be a computation. Furthermore, suppose $B \subseteq Q$ is independent. Then the *subdistribution over S induced by B* is defined as

$$\mu_B \quad = \quad \bigoplus_{\sigma \in B} \{(L(\sigma), \Pr(\sigma))\}\,.$$

In the case when $B = \{\sigma\}$ for some $\sigma \in Q$, we simply write μ_σ instead of $\mu_{\{\sigma\}}$.

Definition 6.9 (Subdistribution induced by computation). Let $C = (Q, \rightarrow, L, \sigma_0)$ be a computation. The *subdistribution over S induced by C*, denoted μ_C, is defined as

$$\mu_C \quad = \quad \mu_{Leaves(C)}.$$

Definition 6.10 (Simple computation). A computation $C = (Q, \rightarrow, L, \sigma_0)$ *simple* if:

1. for every $\sigma \in Q$, there is exactly one outgoing transition from σ, and
2. there are no ϵ-transitions in $\rightarrow$.

Note that a simple computation is non-terminating.

Definition 6.11 (Traces of time-abstract paths). The *observable action trace* (or simply the *trace*) of a time-abstract path π is given by the function

$$Tr : Paths^{*}_{abs} \mapsto \left(Act^{\chi}\right)^{*},$$

defined as follows:

1. $Tr\,(s_0) = \varepsilon$ (note that ε denotes the empty sequence and is distinct from ϵ), and
2. $Tr\left(\pi \xrightarrow{\alpha} s\right) = \begin{cases} Tr\,(\pi) \circ \alpha & \text{if } \alpha \in Act \setminus \{\tau\} \\ Tr\,(\pi) \circ \delta\,(E\,(\pi_{\downarrow})) & \text{if } \alpha = \top \wedge \pi_{\downarrow} \in MS \\ Tr\,(\pi) & \text{otherwise.} \end{cases}$

Note a subtle point in the inductive part of this definition: Definition 2.14 allows a time-abstract path $\pi \xrightarrow{\top} s$ even if $\pi_{\downarrow}$ is not a Markovian state. This is why the additional condition $\pi_{\downarrow} \in MS$ is necessary. It is, however, clear that the probability of such a path is 0 due to the fact that every scheduler must choose actions belonging to $Act^{\top}\,(\pi_{\downarrow})$, so that we can ignore the issue.

Definition 6.12 (Induced paths and traces of computation nodes). Let $C = (Q, \rightarrow, L, \sigma_0)$ be a computation. For every $\sigma \in Q$, we define the *time-abstract path induced by a node*, denoted π_σ, as follows:

1. $\pi_{\sigma_0} = L\,(\sigma_0)$, and
2. for every $\sigma \in Q \setminus \{\sigma_0\}$ with $InTrans\,(\sigma) = (Parent\,(\sigma)\,, \alpha, p, \mu)$,

$$\pi_\sigma \quad = \quad \begin{cases} \pi_{Parent(\sigma)} \xrightarrow{\alpha} L\,(\sigma) & \text{if } \alpha \in Act \\ \pi_{Parent(\sigma)} \xrightarrow{\top} L\,(\sigma) & \text{if } \alpha \in Act^{\chi} \setminus Act \\ \pi_{Parent(\sigma)} & \text{if } \alpha = \epsilon. \end{cases}$$

Next, the *observable action trace* (or simply the *trace*) of a node $\sigma \in Q$, denoted $Tr(\sigma)$, is defined as

$$Tr(\sigma) \quad = \quad Tr(\pi_\sigma).$$

Definition 6.13 (α-computation). Let $\alpha \in Act^{\chi}$. A computation $C = (Q, \rightarrow, L, \sigma_0)$ is an *α-computation* if:

1. $|\mu_C| = 1$,
2. for every $\sigma \in Q$, $Reach_\epsilon(\sigma)$ is finite, and
3. for every $\sigma \in Leaves(C)$ it holds that
    (a) either $\alpha = \tau$ and $Tr(\sigma) = \varepsilon$, or
    (b) $\alpha \neq \tau$ and $Tr(\sigma) = \alpha$.

6.3 Induced computations and schedulers

In this section we will define induced computations and schedulers, which is the first half of Steps 1 and 3 of the proof plan in Section 6.1 (the second half is to specify how to calculate expected time and long run average in computations).

Lemma 6.14 (Simple computation induced by scheduler). *Let $s \in S$ be an initial state, and let $D \in GM$ be a stationary deterministic scheduler. Then there exists a simple computation $C_{s,D} = (Q, \rightarrow, L, \sigma_0)$ such that:*

1. *$L(\sigma_0) = s$, and*
2. *for every $\pi \in Paths^*_{abs}$ it holds that*

$$\Pr_{s,D}(\pi) \quad = \sum_{\{\sigma \in Q | \pi_\sigma = \pi \wedge \sigma \notin In_\epsilon\}} \Pr(\sigma).$$

Proof. The computation $C_{s,D} = (Q, \rightarrow, L, \sigma_0)$ is constructed as follows:

1. We start with a single node σ_0 labelled with s.
2. We expand the computation by adding transitions as specified by the scheduler D. In the end, there is exactly one node for every $\pi \in Paths^*_{abs}$ with $\Pr_{s,D}(\pi) > 0$.

It is quite clear that $C_{s,D}$ satisfies the requirements of this lemma. $\square$

Definition 6.15 (Induced scheduler). Let $C = (Q, \to, L, \sigma_0)$ be a non-terminating computation.

Then define the *induced time-abstract scheduler*

$$D_C : \mathit{Paths}^*_{abs} \times \mathit{Act}^{\mathsf{T}} \mapsto [0,1]$$

as follows:

$$D_C\left(\pi, \alpha\right) \quad = \quad \frac{\sum\limits_{\{\sigma \in Q \mid \pi_\sigma = \pi\}} \Pr\left(\sigma\right) \cdot \Pr\left(\sigma, \alpha\right)}{\sum\limits_{\{\sigma \in Q \mid \pi_\sigma = \pi \wedge \sigma \notin In_\epsilon\}} \Pr\left(\sigma\right)},$$

provided the denominator is non-zero. If the denominator is zero, $D_C\left(\pi, \alpha\right)$ is the uniform distribution over $\mathit{Act}^{\mathsf{T}}\left(\pi_\downarrow\right)$.

The next lemma proves that the expression above does indeed define a scheduler.

Lemma 6.16 (D_C is a valid scheduler). *For every* $\pi \in \mathit{Paths}^*_{abs}$,

$$\sum_{\alpha \in \mathit{Act}^{\mathsf{T}}} D_C\left(\pi, \alpha\right) \quad = \quad 1$$

and

$$D_C\left(\pi, \cdot\right) \quad \in \quad \mathit{Dist}\left(\mathit{Act}^{\mathsf{T}}\left(\pi_\downarrow\right)\right).$$

Proof. First of all, if

$$\sum_{\{\sigma \in Q \mid \pi_\sigma = \pi \wedge \sigma \notin In_\epsilon\}} \Pr\left(\sigma\right) \quad = \quad 0,$$

then $D_C\left(\pi, \alpha\right)$ is the uniform distribution over $\mathit{Act}^{\mathsf{T}}\left(\pi_\downarrow\right)$, so that the claim obviously holds. Therefore, suppose that

$$\sum_{\{\sigma \in Q \mid \pi_\sigma = \pi \wedge \sigma \notin In_\epsilon\}} \Pr\left(\sigma\right) \quad > \quad 0.$$

Now, it holds that

$$\begin{aligned}
\sum_{\alpha\in Act^{\top}} D_C(\pi,\alpha) &= \sum_{\alpha\in Act^{\top}} \frac{\displaystyle\sum_{\{\sigma\in Q\mid\pi_\sigma=\pi\}} \Pr(\sigma)\cdot\Pr(\sigma,\alpha)}{\displaystyle\sum_{\{\sigma\in Q\mid\pi_\sigma=\pi\wedge\sigma\notin I_\epsilon\}} \Pr(\sigma)} \\
\text{(denominator independent of } \alpha\text{)} &= \frac{\displaystyle\sum_{\alpha\in Act^{\top}}\sum_{\{\sigma\in Q\mid\pi_\sigma=\pi\}} \Pr(\sigma)\cdot\Pr(\sigma,\alpha)}{\displaystyle\sum_{\{\sigma\in Q\mid\pi_\sigma=\pi\wedge\sigma\notin In_\epsilon\}} \Pr(\sigma)} \\
\text{(moving outer sum inside)} &= \frac{\displaystyle\sum_{\{\sigma\in Q\mid\pi_\sigma=\pi\}} \Pr(\sigma)\cdot\sum_{\alpha\in Act^{\top}}\Pr(\sigma,\alpha)}{\displaystyle\sum_{\{\sigma\in Q\mid\pi_\sigma=\pi\wedge\sigma\notin In_\epsilon\}} \Pr(\sigma)}.
\end{aligned}$$

At this point, observe that

$$\{\sigma\in Q\mid\pi_\sigma=\pi\} \quad=\quad \biguplus_{\{\sigma\in Q\mid\pi_\sigma=\pi\wedge\sigma\notin In_\epsilon\}} Reach_\epsilon(\sigma).$$

Then we can continue the chain of equalities as follows:

$$\begin{aligned}
\sum_{\alpha\in Act^{\top}} D_C(\pi,\alpha) &= \frac{\displaystyle\sum_{\{\sigma\in Q\mid\pi_\sigma=\pi\}} \Pr(\sigma)\cdot\sum_{\alpha\in Act^{\top}}\Pr(\sigma,\alpha)}{\displaystyle\sum_{\{\sigma\in Q\mid\pi_\sigma=\pi\wedge\sigma\notin In_\epsilon\}} \Pr(\sigma)} \\
\text{(by considerations above)} &= \frac{\displaystyle\sum_{\{\sigma\in Q\mid\pi_\sigma=\pi\wedge\sigma\notin In_\epsilon\}}\ \sum_{\sigma'\in Reach_\epsilon(\sigma)} \Pr(\sigma')\cdot\sum_{\alpha\in Act^{\top}}\Pr(\sigma',\alpha)}{\displaystyle\sum_{\{\sigma\in Q\mid\pi_\sigma=\pi\wedge\sigma\notin In_\epsilon\}} \Pr(\sigma)} \\
\text{(by Definition 6.4)} &= \frac{\displaystyle\sum_{\{\sigma\in Q\mid\pi_\sigma=\pi\wedge\sigma\notin In_\epsilon\}}\ \sum_{\sigma'\in Reach_\epsilon(\sigma)} \Pr(\sigma')\cdot(1-\Pr(\sigma',\epsilon))}{\displaystyle\sum_{\{\sigma\in Q\mid\pi_\sigma=\pi\wedge\sigma\notin In_\epsilon\}} \Pr(\sigma)} \\
\text{(by Definition 6.6)} &= \frac{\displaystyle\sum_{\{\sigma\in Q\mid\pi_\sigma=\pi\wedge\sigma\notin In_\epsilon\}} \Pr(\sigma)}{\displaystyle\sum_{\{\sigma\in Q\mid\pi_\sigma=\pi\wedge\sigma\notin In_\epsilon\}} \Pr(\sigma)} \\
&= 1.
\end{aligned}$$

This completes the proof of the first claim. The truth of the second claim follows from the fact that for every $\alpha \in Act^{\top}$, $D_C(\pi,\alpha) > 0$ implies $\Pr(\sigma,\alpha) > 0$ for some σ with $\pi_\sigma = \pi$. Now, if $\alpha \in Act$, then Definitions 6.2 and 6.4 imply $\alpha \in Act(L(\sigma)) \subseteq Act^{\top}(L(\sigma))$, and if $\alpha = \top$, then $L(\sigma) \in MS$ by Definition 6.4 and thus, again, $\alpha \in Act^{\top}(L(\sigma))$. □

Lemma 6.17 (Correspondence between computation and induced scheduler). *Let $C = (Q, \rightarrow, L, \sigma_0)$ be a non-terminating computation from $s \in S$. Then, for every $\pi \in Paths^*_{abs}$ it holds that*

$$\Pr_{s,D_C}(\pi) \quad = \quad \sum_{\{\sigma \in Q \mid \pi_\sigma = \pi \wedge \sigma \notin In_\epsilon\}} \Pr(\sigma)\,.$$

Proof. The proof is by induction on $|\pi|$.

Base case. If $\pi = s_0$, then

$$\begin{aligned} \Pr_{s,D_C}(\pi) \quad &= \quad \begin{cases} 1 & \text{if } s_0 = s \\ 0 & \text{otherwise} \end{cases} \\ &= \quad \begin{cases} \Pr(\sigma_0) & \text{if } s_0 = s \\ 0 & \text{otherwise} \end{cases} \\ &= \quad \sum_{\{\sigma \in Q \mid \pi_\sigma = \pi \wedge \sigma \notin In_\epsilon\}} \Pr(\sigma)\,. \end{aligned}$$

Induction step. Let

$$\pi' \quad = \quad \pi \xrightarrow{\alpha_n} s_{n+1}.$$

Now, if $\Pr_{s,D_C}(\pi) = 0$, then, by the induction hypothesis,

$$\sum_{\{\sigma \in Q \mid \pi_\sigma = \pi \wedge \sigma \notin In_\epsilon\}} \Pr(\sigma) \quad = \quad 0.$$

But then it must be the case that

$$\{\sigma \in Q \mid \pi_\sigma = \pi \wedge \sigma \notin In_\epsilon\} \quad = \quad \emptyset,$$

which implies that

$$\{\sigma \in Q \mid \pi_\sigma = \pi' \wedge \sigma \notin In_\epsilon\} \quad = \quad \emptyset,$$

and thus,

$$\begin{aligned} \Pr_{s,D_C}(\pi') \quad &= \quad 0 \\ &= \quad \sum_{\{\sigma \in Q \mid \pi_\sigma = \pi' \wedge \sigma \notin In_\epsilon\}} \Pr(\sigma)\,. \end{aligned}$$

If, on the other hand, $\Pr_{s,D_C}(\pi) > 0$, then, by the induction hypothesis,

$$\sum_{\{\sigma \in Q \mid \pi_\sigma = \pi \wedge \sigma \notin In_\epsilon\}} \Pr(\sigma) \quad > \quad 0,$$

so that, by Definition 6.15,

$$D_C(\pi, \alpha) \quad = \quad \frac{\displaystyle\sum_{\{\sigma \in Q \mid \pi_\sigma = \pi\}} \Pr(\sigma) \cdot \Pr(\sigma, \alpha)}{\displaystyle\sum_{\{\sigma \in Q \mid \pi_\sigma = \pi \wedge \sigma \notin In_\epsilon\}} \Pr(\sigma)}.$$

By Lemma 2.25,

$$\Pr_{s,D_C}(\pi') \quad = \quad \Pr_{s,D_C}(\pi) \cdot D_C(\pi, \alpha_n) \cdot \mu_{\pi_\downarrow, \alpha_n}(s_{n+1}).$$

Now, if $\alpha_n \notin Act^{\mathsf{T}}(\pi_\downarrow)$, then clearly $\Pr_{s,D_C}(\pi') = 0$. But then also $\{\sigma \in Q \mid \pi_\sigma = \pi' \wedge \sigma \notin In_\epsilon\} = \emptyset$ and the result holds. Therefore, assume $\alpha_n \in Act^{\mathsf{T}}(\pi_\downarrow)$.

Then,

$$\begin{aligned}
\Pr_{s,D_C}(\pi') \quad &= \quad \Pr_{s,D_C}(\pi) \cdot D_C(\pi, \alpha_n) \cdot \mu_{\pi_\downarrow, \alpha_n}(s_{n+1}) \\
\text{(by Definition 6.15)} \quad &= \quad \Pr_{s,D_C}(\pi) \cdot \frac{\displaystyle\sum_{\{\sigma \in Q \mid \pi_\sigma = \pi\}} \Pr(\sigma) \cdot \Pr(\sigma, \alpha_n)}{\displaystyle\sum_{\{\sigma \in Q \mid \pi_\sigma = \pi \wedge \sigma \notin In_\epsilon\}} \Pr(\sigma)} \cdot \mu_{\pi_\downarrow, \alpha_n}(s_{n+1}) \\
\text{(induction hypothesis)} \quad &= \quad \left(\sum_{\{\sigma \in Q \mid \pi_\sigma = \pi\}} \Pr(\sigma) \cdot \Pr(\sigma, \alpha_n) \right) \cdot \mu_{\pi_\downarrow, \alpha_n}(s_{n+1}).
\end{aligned}$$

At this point, observe that $\pi_\sigma = \pi$ implies $L(\sigma) = \pi_\downarrow$. Therefore, we can continue the chain of equalities as follows:

$$\begin{aligned}
\Pr_{s,D_C}(\pi') \quad &= \quad \left(\sum_{\{\sigma \in Q \mid \pi_\sigma = \pi\}} \Pr(\sigma) \cdot \Pr(\sigma, \alpha_n) \right) \cdot \mu_{\pi_\downarrow, \alpha_n}(s_{n+1}) \\
(L(\sigma) = \pi_\downarrow) \quad &= \quad \sum_{\{\sigma \in Q \mid \pi_\sigma = \pi\}} \Pr(\sigma) \cdot \Pr(\sigma, \alpha_n) \cdot \mu_{L(\sigma), \alpha_n}(s_{n+1}).
\end{aligned}$$

At this point, we would like to replace $\Pr(\sigma, \alpha_n)$ with $\sum_{(\sigma, \alpha_n, p, \mu) \in \to} p$. If $\alpha_n \neq \mathsf{T}$, the validity of the substitution follows directly from Definition 6.4. If, on the other hand, $\alpha_n = \mathsf{T}$, then a minor technical problem arises: the transitions from σ in C are not labelled with T, but rather with $\delta(E(L(\sigma)))$. Therefore, in the case when $\alpha_n = \mathsf{T}$, let us change the notation and set

$\alpha_n = \delta\left(E\left(L\left(\sigma\right)\right)\right)$. Then, in both cases,

$$\begin{aligned}
\Pr_{s,D_C}\left(\pi'\right) &= \sum_{\{\sigma\in Q|\pi_\sigma=\pi\}} \Pr\left(\sigma\right)\cdot\left(\sum_{(\sigma,\alpha_n,p,\mu)\,\in\rightarrow} p\right)\cdot\mu_{L(\sigma),\alpha_n}\left(s_{n+1}\right)\\
\text{(moving everything inside)} &= \sum_{\{\sigma\in Q|\pi_\sigma=\pi\}}\ \sum_{(\sigma,\alpha_n,p,\mu)\,\in\rightarrow} \Pr\left(\sigma\right)\cdot p\cdot\mu_{L(\sigma),\alpha_n}\left(s_{n+1}\right).
\end{aligned}$$

Now, from Definition 6.2 it follows that for every transition $(\sigma,\alpha_n,p,\mu)\in\rightarrow$ it holds that

$$\mu_{L(\sigma),\alpha_n}\left(s_{n+1}\right) = \sum_{\{\sigma'\in Supp(\mu)|L(\sigma')=s_{n+1}\}} \mu\left(\sigma'\right).$$

Therefore, for every $\sigma\in Q$,

$$\begin{aligned}
\sum_{(\sigma,\alpha_n,p,\mu)\,\in\rightarrow} \Pr\left(\sigma\right)\cdot p\cdot\mu_{L(\sigma),\alpha_n}\left(s_{n+1}\right) &= \sum_{(\sigma,\alpha_n,p,\mu)\,\in\rightarrow} \Pr\left(\sigma\right)\cdot p\cdot \sum_{\{\sigma'\in Supp(\mu)|L(\sigma')=s_{n+1}\}} \mu\left(\sigma'\right)\\
\text{(moving }\Pr\left(\sigma\right)\cdot p\text{ inside)} &= \sum_{(\sigma,\alpha_n,p,\mu)\,\in\rightarrow}\ \sum_{\{\sigma'\in Supp(\mu)|L(\sigma')=s_{n+1}\}} \Pr\left(\sigma\right)\cdot p\cdot\mu\left(\sigma'\right)\\
\text{(by Definition 6.4)} &= \sum_{(\sigma,\alpha_n,p,\mu)\,\in\rightarrow}\ \sum_{\{\sigma'\in Supp(\mu)|L(\sigma')=s_{n+1}\}} \Pr\left(\sigma'\right).
\end{aligned}$$

Next, it is not difficult to see that

$$\left\{\sigma\in Q\mid \pi_\sigma=\pi'\wedge\sigma\notin In_\epsilon\right\} = \biguplus_{\{\sigma\in Q|\pi_\sigma=\pi\}}\ \biguplus_{(\sigma,\alpha_n,p,\mu)\,\in\rightarrow} \left\{\sigma'\mid\sigma'\in Supp\left(\mu\right)\ \wedge\ L\left(\sigma'\right)=s_{n+1}\right\}.$$

Putting everything together, the chain of equalities above can be continued as follows:

$$\begin{aligned}
\Pr_{s,D_C}\left(\pi'\right) &= \sum_{\{\sigma\in Q|\pi_\sigma=\pi\}}\ \sum_{(\sigma,\alpha_n,p,\mu)\,\in\rightarrow} \Pr\left(\sigma\right)\cdot p\cdot\mu_{L(\sigma),\alpha_n}\left(s_{n+1}\right)\\
&= \sum_{\{\sigma\in Q|\pi_\sigma=\pi\}}\ \sum_{(\sigma,\alpha_n,p,\mu)\,\in\rightarrow}\ \sum_{\{\sigma'\in Supp(\mu)|L(\sigma')=s_{n+1}\}} \Pr\left(\sigma'\right)\\
&= \sum_{\{\sigma\in Q|\pi_\sigma=\pi'\wedge\sigma\notin In_\epsilon\}} \Pr\left(\sigma\right),
\end{aligned}$$

as desired. □

6.4 Expected time in computations

In this section we will complete Steps 1 and 3 of the proof plan in Section 6.1 (only for the expected time case - long run average will be covered in the next section) by specifying how to calculate expected time in a computation and proving that $eT^C\left(s,\Diamond G\right) = eT^D\left(s,\Diamond G\right)$ whenever:

- the computation C from s is induced by the time-abstract scheduler D,

- or the scheduler D is induced by the computation C from s.

Lemma 6.18 (Mean of function under time-abstract scheduler). *Let* $\mu_0 \in Dist\left(S\right)$ *be an initial distribution over* S *and let* $D \in DM$ *be a time-abstract scheduler. Furthermore, suppose* $\pi \in Paths_{abs}^n$ *for some* $n \in \mathbb{N}_{\geq 0}$ *and*

$$f : Paths^n \mapsto \mathbb{R}.$$

Then,

$$\int_{Cyl(\Pi_\pi)} f\left(\pi'\left[0...n\right]\right) \Pr_{\mu_0, D}\left(d\pi'\right) \quad = \quad \Pr_{\mu_0, D}\left(\pi\right) \cdot \int_{\left(\mathbb{R}_{\geq 0}\right)^n} f\left(\pi'\right) \; \eta_{\pi[0]}\left(dt_0\right) ... \eta_{\pi[n-1]}\left(dt_{n-1}\right),$$

where:

1. Π_π *is the rectangle in* $Paths^n$ *corresponding to the time-abstract path* π *(see Definition 2.14) and*
2. π' *on the right is the finite path defined by the states and actions in* π *and the variables of integration.*

Proof. Suppose

$$\pi \quad = \quad s_0 \xrightarrow{\alpha_0} ... \xrightarrow{\alpha_{n-1}} s_n.$$

Then, for every $\pi' \in Paths^\omega$ with $\pi'\left[0...n\right] \in \pi$,

$$\begin{aligned} \mu_0\left(s_0\right) \prod_{i=0}^{n-1} \left(D\left(\pi'\left[0...i\right], \alpha_i\right) \mu_{s_i, \alpha_i}\left(s_{i+1}\right)\right) \quad &= \quad \mu_0\left(s_0\right) \prod_{i=0}^{n-1} \left(D\left(\pi\left[0...i\right], \alpha_i\right) \mu_{s_i, \alpha_i}\left(s_{i+1}\right)\right) \\ \text{(by Lemma 2.25)} \quad &= \quad \Pr_{\mu_0, D}\left(\pi\right), \end{aligned}$$

where the first step is possible because D is time-abstract.

Then, using Lemma 2.23 and the equality that we have just derived,

$$\begin{aligned} \int_{Cyl(\Pi_\pi)} f\left(\pi'\left[0...n\right]\right) \Pr_{\mu_0, D}\left(d\pi'\right) \quad &= \quad \int_{\left(\mathbb{R}_{\geq 0}\right)^n} f\left(\pi'\right) \cdot \Pr_{\mu_0, D}\left(\pi\right) \; \eta_{\pi[0]}\left(dt_0\right) ... \eta_{\pi[n-1]}\left(dt_{n-1}\right) \\ \left(\text{moving } \Pr_{\mu_0, D}\left(\pi\right) \text{ out}\right) \quad &= \quad \Pr_{\mu_0, D}\left(\pi\right) \cdot \int_{\left(\mathbb{R}_{\geq 0}\right)^n} f\left(\pi'\left[0...n\right]\right) \; \eta_{\pi[0]}\left(dt_0\right) ... \eta_{\pi[n-1]}\left(dt_{n-1}\right), \end{aligned}$$

as desired. □

Lemma 6.19 (Expected time split by time-abstract paths). *First, define the function*

$$eT : Paths_{abs}^* \mapsto \mathbb{R}_{\geq 0}$$

as follows:

1. *$eT(s_0) = 0$, and*
2. *$eT\left(\pi \xrightarrow{\alpha_n} s_{n+1}\right) = eT(\pi) + MST(\pi_{\downarrow})$, where the MST function was defined in Lemma 4.1.*

Now, suppose $s \in S$, $D \in GM$ is time-abstract, and $G \subseteq S$. Let

$$\mathfrak{G} \quad = \quad \left\{\pi \in Paths^*_{abs} \mid \pi_{\downarrow} \in G \ \wedge \ \forall i \in \{0, ..., |\pi| - 1\} \ . \ \pi[i] \notin G\right\}.$$

Then,

$$eT^D(s, \Diamond G) \quad = \quad \begin{cases} \sum\limits_{\pi \in \mathfrak{G}} eT(\pi) \cdot \Pr\limits_{s,D}(\pi) & \text{if } \sum\limits_{\pi \in \mathfrak{G}} \Pr\limits_{s,D}(\pi) = 1 \\ \infty & \text{otherwise.} \end{cases}$$

Proof. First of all, observe that

$$\sum_{\pi \in \mathfrak{G}} \Pr_{s,D}(\pi) \quad \leq \quad 1$$

due to the fact that no path in $\mathfrak{G}$ is a prefix of another path in $\mathfrak{G}$.

Now, recall that, by Definition 2.48,

$$eT^D(s, \Diamond G) \quad = \quad \int_{Paths^{\omega}} V_G(\pi) \ \Pr_{s,D}(d\pi).$$

Next, let

$$P \quad = \quad \{\pi \in Paths^{\omega} \mid V_G(\pi) < \infty\}.$$

Then, it holds that

$$P \quad = \quad \biguplus_{\pi \in \mathfrak{G}} Cyl(\Pi_{\pi})$$

(recall that by Definition 2.14, Π_{π} is the rectangle in $Paths^n$ corresponding to the time-abstract path π). Then,

$$\begin{aligned} \Pr_{s,D}(P) \quad &= \quad \sum_{\pi \in \mathfrak{G}} \Pr_{s,D}(Cyl(\Pi_{\pi})) \\ \text{(by Definition 2.23)} \quad &= \quad \sum_{\pi \in \mathfrak{G}} \Pr_{s,D}(\Pi_{\pi}) \\ \text{(by Definition 2.24)} \quad &= \quad \sum_{\pi \in \mathfrak{G}} \Pr_{s,D}(\pi). \end{aligned}$$

Thus, if $\sum_{\pi\in\mathfrak{G}} \Pr_{s,D}(\pi) < 1$, then $\Pr_{s,D}(Paths^{\omega} \setminus P) > 0$, so that

$$\begin{aligned}
eT^D(s, \Diamond G) &= \int_{Paths^{\omega}} V_G(\pi)\ \Pr_{s,D}(d\pi) \\
&= \int_{P} V_G(\pi)\ \Pr_{s,D}(d\pi) + \int_{Paths^{\omega}\setminus P} V_G(\pi)\ \Pr_{s,D}(d\pi) \\
\text{(retaining only 2nd integral)} &\geq \int_{Paths^{\omega}\setminus P} V_G(\pi)\ \Pr_{s,D}(d\pi) \\
(V_G(\pi) = \infty\ \forall \pi \in Paths^{\omega} \setminus P) &= \infty \cdot \int_{Paths^{\omega}\setminus P} \Pr_{s,D}(d\pi) \\
&= \infty \cdot \Pr_{s,D}(Paths^{\omega} \setminus P) \\
\left(\Pr_{s,D}(Paths^{\omega} \setminus P) > 0\right) &= \infty.
\end{aligned}$$

On the other hand, if $\sum_{\pi\in\mathfrak{G}} \Pr_{s,D}(\pi) = 1$, then

$$\begin{aligned}
eT^D(s, \Diamond G) &= \int_{Paths^{\omega}} V_G(\pi)\ \Pr_{s,D}(d\pi) \\
&= \int_{P} V_G(\pi)\ \Pr_{s,D}(d\pi) \\
&= \sum_{\pi\in\mathfrak{G}} \int_{Cyl(\Pi_\pi)} V_G(\pi')\ \Pr_{s,D}(d\pi').
\end{aligned}$$

Now, for every $\pi \in \mathfrak{G}$,

$$\begin{aligned}
\int_{Cyl(\Pi_\pi)} V_G(\pi')\ \Pr_{s,D}(d\pi') &= \int_{Cyl(\Pi_\pi)} \left(\sum_{i=0}^{|\pi|-1} t_i\right) \Pr_{s,D}(d\pi') \\
\text{(rearranging sum and integral)} &= \sum_{i=0}^{|\pi|-1} \int_{Cyl(\Pi_\pi)} t_i\ \Pr_{s,D}(d\pi') \\
\text{(by Lemma 6.18)} &= \sum_{i=0}^{|\pi|-1} \left[\Pr_{s,D}(\pi) \cdot \int_{(\mathbb{R}_{\geq 0})^{|\pi|}} t_i\ \eta_{\pi[0]}(dt_0) ... \eta_{\pi[|\pi|-1]}\left(dt_{|\pi|-1}\right)\right] \\
\left(\text{moving } \Pr_{s,D}(\pi) \text{ outside}\right) &= \Pr_{s,D}(\pi) \cdot \sum_{i=0}^{|\pi|-1} \int_{(\mathbb{R}_{\geq 0})^{|\pi|}} t_i\ \eta_{\pi[0]}(dt_0) ... \eta_{\pi[|\pi|-1]}\left(dt_{|\pi|-1}\right) \\
\text{(all integrals except } i\text{th one equal 1)} &= \Pr_{s,D}(\pi) \cdot \sum_{i=0}^{|\pi|-1} \int_{\mathbb{R}_{\geq 0}} t_i\ \eta_{\pi[i]}(dt_i) \\
\text{(mean time to stay in } \pi[i]\text{)} &= \Pr_{s,D}(\pi) \cdot \sum_{i=0}^{|\pi|-1} MST(\pi[i]) \\
\text{(by definition of } eT\text{)} &= \Pr_{s,D}(\pi) \cdot eT(\pi),
\end{aligned}$$

where t_i is the amount of time the path $\pi' \in Cyl(\Pi_\pi)$ stays in the ith state (which is always the same state $\pi[i]$ since all the paths belong to the same time-abstract path π).

But then, continuing the chain of equalities above,

$$\begin{aligned} eT^{D}(s, \Diamond G) &= \sum_{\pi \in \mathfrak{G}} \int_{Cyl(\Pi_\pi)} V_G(\pi') \Pr_{s,D}(d\pi') \\ &= \sum_{\pi \in \mathfrak{G}} eT(\pi) \cdot \Pr_{s,D}(\pi), \end{aligned}$$

as desired. □

Definition 6.20 (Expected time in computations). Let the function

$$eT : Q \mapsto \mathbb{R}_{\geq 0}$$

be defined as

$$eT(\sigma) = eT(\pi_\sigma).$$

Next, suppose $G \subseteq S$. Let

$$\mathcal{G} = \{\sigma \in Q \mid \pi_\sigma \in \mathfrak{G} \ \wedge \ \sigma \notin In_\epsilon\}.$$

Now, define the expected time to reach G in the computation C from s as follows:

$$eT^{C}(s, \Diamond G) = \begin{cases} \sum_{\sigma \in \mathcal{G}} eT(\sigma) \Pr(\sigma) & \text{if } \sum_{\sigma \in \mathcal{G}} \Pr(\sigma) = 1 \\ \infty & \text{otherwise.} \end{cases}$$

Lemma 6.21 ($\mathcal{G}$ is independent). *Let $C = (Q, \rightarrow, L, \sigma_0)$ and $D \in GM$. Suppose $G \subseteq S$, and let*

$$\mathfrak{G} = \{\pi \in Paths^{*}_{abs} \mid \pi_{\downarrow} \in G \ \wedge \ \forall i \in \{0, ..., |\pi| - 1\} \ . \ \pi[i] \notin G\}$$

(as in Lemma 6.19) and

$$\mathcal{G} = \{\sigma \in Q \mid \pi_\sigma \in \mathfrak{G} \ \wedge \ \sigma \notin In_\epsilon\}$$

(as in Definition 6.20). Then $\mathcal{G}$ is independent.

Proof. The proof is by contradiction. Suppose $\sigma, \sigma' \in \mathcal{G}$ are such that $\sigma \neq \sigma'$ and $\sigma' \in Children^{*}(\sigma)$. Clearly, the time-abstract path π_σ is a prefix of $\pi_{\sigma'}$, that is, $\pi_\sigma = \pi_{\sigma'}[0...i]$ for some $i \in \{0, ..., |\pi_{\sigma'}|\}$. Moreover, since $\sigma' \notin In_\epsilon$, it must be the case that $i < |\pi_{\sigma'}|$ (see Definition 6.11). But then $\pi_{\sigma'}[i] \in G$, which contradicts the definition of $\mathfrak{G}$. □

Lemma 6.22 (Trace equality implies expected time equality). *Let* $C = (Q, \rightarrow, L, \sigma_0)$*. Then, for every pair of nodes* $\sigma, \sigma' \in Q$*,*

$$Tr(\sigma) = Tr(\sigma') \quad \text{implies} \quad eT(\sigma) = eT(\sigma').$$

Proof. By a straightforward induction on the structure of C we can prove that for every $\sigma \in Q$,

$$eT(\sigma) \quad = \quad eT_{Tr}(Tr(\sigma)),$$

where the function eT_{Tr} is defined as:

1. $eT_{Tr}(\varepsilon) = 0$ and
2. For a sequence $a \in (Act^{\chi})^*$ and an action $\alpha \in Act^{\chi}$,

$$eT_{Tr}(a \circ \alpha) \quad = \quad \begin{cases} eT_{Tr}(a) & \text{if } \alpha \in Act \\ eT_{Tr}(a) + \frac{1}{r} & \text{if } \alpha = \delta(r). \end{cases}$$

The key to the proof is the fact that, by Definition 6.2, every transition from every node $\sigma \in Q$ must be labelled with an action in $Act^{\chi}(L(\sigma))$ (with a single exception when $\alpha = \epsilon$). Thus, whenever an outgoing transition of a node σ is labelled with $\delta(r)$ for some $r \in \mathbb{R}_{>0}$, we can be sure that $L(\sigma) \in MS$ and, moreover, $MST(L(\sigma)) = {}^1\!/_r$. Analogously, if the label is $\alpha \in Act$, we know that $L(\sigma) \in PS$ and $MST(L(\sigma)) = 0$.

The details of the proof are rather uninteresting, so that we leave them out for brevity. □

Lemma 6.23 (Expected time for time-abstract scheduler and in computation). *Let* $C_s = (Q, \rightarrow, L, \sigma_0)$ *be a computation from a state* s *and* D *be a time-abstract scheduler such that for every* $\pi \in Paths^*_{abs}$ *it holds that*

$$\Pr_{s,D}(\pi) \quad = \quad \sum_{\{\sigma \in Q \mid \pi_\sigma = \pi \wedge \sigma \notin In_\epsilon\}} \Pr(\sigma).$$

Let $G \subseteq S$*. Then*

$$eT^D(s, \Diamond G) \quad = \quad eT^{C_s}(s, \Diamond G).$$

Proof. Recall that

$$\mathfrak{G} \quad = \quad \{\pi \in Paths^*_{abs} \mid \pi_\downarrow \in G \ \wedge \ \forall i \in \{0, ..., |\pi| - 1\} \ . \ \pi[i] \notin G\}$$

was defined in Lemma 6.19, while

$$\mathcal{G} \quad = \quad \{\sigma \in Q \mid \pi_\sigma \in \mathfrak{G} \ \wedge \ \sigma \notin In_\epsilon\}$$

was defined in Definition 6.20.

First, note that

$$\mathcal{G} \quad = \quad \biguplus_{\pi \in \mathfrak{G}} \{\sigma \in Q \mid \pi_\sigma = \pi \ \wedge \ \sigma \notin In_\epsilon\}\,.$$

Then, using the assumption,

$$\begin{aligned}\sum_{\pi \in \mathfrak{G}} \Pr_{s,D}(\pi) \quad &= \quad \sum_{\pi \in \mathfrak{G}} \sum_{\{\sigma \in Q \mid \pi_\sigma = \pi \wedge \sigma \notin In_\epsilon\}} \Pr(\sigma) \\ &= \quad \sum_{\sigma \in \mathcal{G}} \Pr(\sigma)\,.\end{aligned}$$

Now, Lemma 6.21 implies that

$$\sum_{\sigma \in \mathcal{G}} \Pr(\sigma) \quad \leq \quad 1.$$

If $\sum_{\sigma \in \mathcal{G}} \Pr(\sigma) = \sum_{\pi \in \mathfrak{G}} \Pr_{s,D}(\pi) < 1$, then, by Lemma 6.19 and Definition 6.20,

$$\begin{aligned} eT^{D}(s, \Diamond G) \quad &= \quad \infty \\ &= \quad eT^{C_s}(s, \Diamond G)\,.\end{aligned}$$

On the other hand, suppose $\sum_{\sigma \in \mathcal{G}} \Pr(\sigma) = \sum_{\pi \in \mathfrak{G}} \Pr_{s,D}(\pi) = 1$. In this case,

$$\begin{aligned} eT^{D}(s, \Diamond G) \quad &= \quad \sum_{\pi \in \mathfrak{G}} eT(\pi) \cdot \Pr_{s,D}(\pi) \\ \text{(by assumption)} \quad &= \quad \sum_{\pi \in \mathfrak{G}} eT(\pi) \cdot \sum_{\{\sigma \in Q \mid \pi_\sigma = \pi \wedge \sigma \notin In_\epsilon\}} \Pr(\sigma) \\ \text{(by Definition 6.20)} \quad &= \quad \sum_{\pi \in \mathfrak{G}} \sum_{\{\sigma \in Q \mid \pi_\sigma = \pi \wedge \sigma \notin In_\epsilon\}} eT(\sigma) \cdot \Pr(\sigma) \\ &= \quad \sum_{\sigma \in \mathcal{G}} eT(\sigma) \cdot \Pr(\sigma) \\ \text{(by Definition 6.20)} \quad &= \quad eT^{C_s}(s, \Diamond G)\,.\end{aligned}$$

□

Lemma 6.24 (Minimum expected time by simple computation). *Let $s \in S$ and $G \subseteq S$. Then there exists a simple computation C_s from s such that:*

$$eT^{min}(s, \Diamond G) \quad = \quad eT^{C_s}(s, \Diamond G)\,.$$

Proof. By Lemma 2.50, there exists a stationary deterministic scheduler D such that $eT^{min}(s, \Diamond G) = eT^{D}(s, \Diamond G)$. Moreover, by Lemma 6.14, the induced simple computation $C_{s,D}$ and D satisfy the conditions of Lemma 6.23. Thus,

$$\begin{aligned} eT^{min}(s, \Diamond G) &= eT^{D}(s, \Diamond G) \\ &= eT^{C_{s,D}}(s, \Diamond G), \end{aligned}$$

as desired. ☐

Lemma 6.25 (Expected time by induced scheduler). *Let $C = (Q, \rightarrow, L, \sigma_0)$ be a non-terminating computation from $s \in S$. Then, for every $G \subseteq S$,*

$$eT^{D_C}(s, \Diamond G) = eT^{C}(s, \Diamond G).$$

Proof. The result follows directly from Lemmas 6.23 and 6.17. ☐

6.5 Long run average in computations

In this section we will complete Steps 1 and 3 of the proof plan in Section 6.1 by specifying how to calculate long run average in a computation and proving that $LRA^{C}(s, \Diamond G) = LRA^{D}(s, \Diamond G)$ whenever:

- the computation C from s is induced by the time-abstract scheduler D,
- or the scheduler D is induced by the computation C from s.

Definition 6.26 (Unified transition probability densities by action). For every action $\alpha \in Act^{\chi}$, let

$$\eta_\alpha : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}_{\geq 0}$$

be defined as follows:

$$\eta_\alpha(t) = \begin{cases} r \cdot e^{-r \cdot t} & \text{if } \alpha = \delta(r) \\ \delta_{Dirac}(t) & \text{if } \alpha \in Act, \end{cases}$$

where δ_{Dirac} is the Dirac delta function.

Lemma 6.27 (Unified transition probability densities by states and actions). *For every* $s \in S$ *and* $\alpha \in Act^{\chi}(s)$,

$$\eta_s = \eta_\alpha.$$

Proof. Follows immediately from Definitions 2.22 and 6.26. □

Definition 6.28 (Probability of mth state at time t given trace). Let $a \in (Act^{\chi})^*$ be a sequence of actions containing at least one action belonging to $Act^{\chi} \setminus Act$ (i.e., an action of the form $\delta(r)$). Let $\alpha_0, ..., \alpha_m$ denote the actions from $Act^{\chi} \setminus Act$ in a.

Then, define

$$P_t(a) = \int_{(\mathbb{R}_{\geq 0})^{m+1}} I_{\mathfrak{T}_t^m}(t_0, ..., t_m)\ \eta_{\alpha_0}(dt_0) ... \eta_{\alpha_m}(dt_m),$$

where $I_{\mathfrak{T}_t^m}$ is the indicator function for the set $\mathfrak{T}_t^m$ defined in Lemma 5.1.

In the case when $a \in (Act^{\chi})^*$ does not contain any actions in $Act^{\chi} \setminus Act$, we set

$$P_t(a) = 0.$$

Definition 6.29 (Set of time-abstract paths corresponding to set of states). Let $G \subseteq S$. Then, let

$$\mathfrak{L}_G = \{\pi \in Paths_{abs}^* \mid |\pi| > 0 \wedge \pi[|\pi| - 1] \in G\}.$$

Lemma 6.30 (Probability to be in G at time t). *Let* $s \in S$ *be an initial state and let* $D \in GM$ *be a time-abstract scheduler. Suppose* $G \subseteq MS$*. Then, for every* $t \in \mathbb{R}_{\geq 0}$,

$$\Pr_{s,D}\{\pi \in Paths^\omega \mid \pi@t \cap G \neq \emptyset\} = \sum_{\pi \in \mathfrak{L}_G} \Pr_{s,D}(\pi) \cdot P_t(Tr(\pi)).$$

Proof. For every $\pi \in \mathfrak{L}_G$, let

$$\mathfrak{P}_t^\pi = \left\{\pi' \in Cyl(\Pi_\pi) \mid \left(t_0, ..., t_{|\pi|-1}\right) \in \mathfrak{T}_t^{|\pi|-1}\right\}$$

(recall that the set $\mathfrak{T}_t^{|\pi|-1}$ was defined in Lemma 5.1).

Then, analogously to Lemma 5.3,

$$\{\pi' \in Paths^\omega \mid \pi'@t \cap G \neq \emptyset\} = \biguplus_{\pi \in \mathfrak{L}_G} \mathfrak{P}_t^\pi,$$

so that

$$\Pr_{s,D}\{\pi \in Paths^{\omega} \mid \pi@t \cap G \neq \emptyset\} \quad = \quad \sum_{\pi \in \mathfrak{L}_G} \mathfrak{P}_t^{\pi}.$$

Therefore, we only need to prove that for every $\pi \in \mathfrak{L}_G$ it holds that

$$\Pr_{s,D}\left(\mathfrak{P}_t^{\pi}\right) \quad = \quad \Pr_{s,D}(\pi) \cdot P_t\left(Tr(\pi)\right).$$

To this end, suppose $\pi \in \mathfrak{L}_G$. Let

$$\pi \quad = \quad s_0 \xrightarrow{\alpha_0} \dots \xrightarrow{\alpha_{n-1}} s_n.$$

Then,

$$\begin{aligned}
\Pr_{s,D}\left(\mathfrak{P}_t^{\pi}\right) \quad &= \quad \int_{Cyl(\Pi_\pi)} I_{\mathfrak{T}_t^{n-1}}\left(t_0, \dots, t_{n-1}\right)\ \Pr_{s,D}\left(d\pi'\right) \\
\text{(by Lemma 6.18)} \quad &= \quad \Pr_{s,D}(\pi) \cdot \int_{\left(\mathbb{R}_{\geq 0}\right)^n} I_{\mathfrak{T}_t^{n-1}}\left(t_0, \dots, t_{n-1}\right)\ \eta_{s_0}\left(dt_0\right) \dots \eta_{s_{n-1}}\left(dt_{n-1}\right).
\end{aligned}$$

Now, if $\Pr_{s,D}(\pi) = 0$, then clearly

$$\begin{aligned}
\Pr_{s,D}\left(\mathfrak{P}_t^{\pi}\right) \quad &= \quad 0 \\
&= \quad \Pr_{s,D}(\pi) \cdot P_t\left(Tr(\pi)\right).
\end{aligned}$$

If, on the other hand, $\Pr_{s,D}(\pi) > 0$, then it must be the case that for every $i \in \{0, \dots, n-1\}$ it holds that $\alpha_i \in Act^{\mathsf{T}}(s_i)$. In particular, $\alpha_i = \mathsf{T}$ if and only if $s_i \in MS$. Using this fact, let us set $\alpha_i = \delta\left(E(s_i)\right)$ whenever $s_i \in MS$ (in other words, let us convert the actions α_i from Act^{T} to Act^{χ}). Then, by Lemma 6.27, $\eta_{s_i} = \eta_{\alpha_i}$. Thus,

$$\Pr_{s,D}\left(\mathfrak{P}_t^{\pi}\right) \quad = \quad \Pr_{s,D}(\pi) \cdot \int_{\left(\mathbb{R}_{\geq 0}\right)^n} I_{\mathfrak{T}_t^{n-1}}\left(t_0, \dots, t_{n-1}\right)\ \eta_{\alpha_0}\left(dt_0\right) \dots \eta_{\alpha_{n-1}}\left(dt_{n-1}\right).$$

We now show that we can remove the integrals corresponding to probabilistic actions without affecting the validity of the equation. Indeed, the first thing to note that whenever $\alpha_i \in Act$, η_{α_i} permits only one value of t_i, namely $t_i = 0$. In effect, we can simply remove the integrals corresponding to probabilistic actions and replace the corresponding variables t_i with zeros. Let $i_0, \dots, i_m$ be the indices of actions belonging to $Act^{\chi} \setminus Act$ in $\alpha_0, \dots, \alpha_{n-1}$. Note that $\alpha_{n-1} =$

$\delta\left(E\left(s_{n-1}\right)\right)$, so that $i_m = n-1$. Then,

$$\begin{aligned}
\left(t_0, ..., t_{n-1}\right) \in \mathfrak{T}_t^{n-1} \quad &\Leftrightarrow \quad \sum_{i=0}^{n-2} t_i \leq t < \sum_{i=0}^{n-2} t_i + t_{n-1} \\
&\Leftrightarrow \quad \sum_{j=0}^{m-1} t_{i_j} \leq t < \sum_{j=0}^{m-1} t_{i_j} + t_{n-1} \\
&\Leftrightarrow \quad \left(t_{i_0}, ..., t_{i_m}\right) \in \mathfrak{T}_t^{m},
\end{aligned}$$

so that

$$I_{\mathfrak{T}_t^{n-1}}\left(t_0, ..., t_{n-1}\right) \quad = \quad I_{\mathfrak{T}_t^{m}}\left(t_{i_0}, ..., t_{i_m}\right).$$

Therefore,

$$\begin{aligned}
\Pr_{s,D}\left(\mathfrak{P}_t^{\pi}\right) \quad &= \quad \Pr_{s,D}(\pi) \cdot \int_{\left(\mathbb{R}_{\geq 0}\right)^n} I_{\mathfrak{T}_t^{n-1}}\left(t_0, ..., t_{n-1}\right)\ \eta_{\alpha_0}\left(dt_0\right) ... \eta_{\alpha_{n-1}}\left(dt_{n-1}\right) \\
&= \quad \Pr_{s,D}(\pi) \cdot \int_{\left(\mathbb{R}_{\geq 0}\right)^{m+1}} I_{\mathfrak{T}_t^{m}}\left(t_{i_0}, ..., t_{i_m}\right)\ \eta_{\alpha_{i_0}}\left(dt_{i_0}\right) ... \eta_{\alpha_{i_m}}\left(dt_{i_m}\right) \\
\text{(by Definition 6.28)} \quad &= \quad \Pr_{s,D}(\pi) \cdot P_t\left(Tr(\pi)\right).
\end{aligned}$$

□

Lemma 6.31 (LRA split by time-abstract paths). *Let $D \in GM$ be a time-abstract scheduler. Suppose $G \subseteq MS$. Then, for every $s \in S$,*

$$LRA^D(s, \Diamond G) \quad = \quad \lim_{t \to \infty} \frac{1}{t} \int_0^t \sum_{\pi \in \mathfrak{L}_G} \Pr_{s,D}(\pi) \cdot P_u\left(Tr(\pi)\right)\ du.$$

Proof. By Definition 2.51,

$$LRA^D(s, \Diamond G) \quad = \quad \int_{Paths^{\omega}} \left(\lim_{t \to \infty} \frac{1}{t} \int_0^t I_G(\pi @ u)\ du\right) \Pr_{s,D}(d\pi),$$

where

$$I_G(\pi @ u) \quad = \quad \begin{cases} 1 & \text{if } \pi @ u \ \cap\ G \neq \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

Now,

$$\begin{aligned}
LRA^{D}(s, \Diamond G) &= \int_{Paths^{\omega}} \left(\lim_{t\to\infty} \frac{1}{t} \int_{0}^{t} I_G(\pi @ u)\ du \right) \Pr_{s,D}(d\pi) \\
\text{(rearranging *)} \quad &= \lim_{t\to\infty} \frac{1}{t} \int_{0}^{t} \int_{Paths^{\omega}} I_G(\pi @ u) \Pr_{s,D}(d\pi)\ du \\
&= \lim_{t\to\infty} \frac{1}{t} \int_{0}^{t} \Pr_{s,D}\{\pi \in Paths^{\omega} \mid \pi @ u \cap G \neq \emptyset\}\ du \\
\text{(by Lemma 6.30)} \quad &= \lim_{t\to\infty} \frac{1}{t} \int_{0}^{t} \sum_{\pi \in \mathfrak{L}_G} \Pr_{s,D}(\pi) \cdot P_u(Tr(\pi))\ du.
\end{aligned}$$

($*$) The validity of this step is not immediately obvious: it is taken directly from [11]. □

Definition 6.32 (LRA in computation). Let $C = (Q, \twoheadrightarrow, L, \sigma_0)$ be a computation from a state s. Furthermore, suppose $G \subseteq MS$. Then, let

$$LRA^{C}(s, \Diamond G) = \lim_{t\to\infty} \frac{1}{t} \int_{0}^{t} \sum_{\{\sigma \in Q \mid \pi_\sigma \in \mathfrak{L}_G \wedge \sigma \notin In_\epsilon\}} \Pr(\sigma) \cdot P_u(Tr(\sigma))\ du.$$

Lemma 6.33 (LRA by scheduler and in computation). *Let $C_s = (Q, \twoheadrightarrow, L, \sigma_0)$ be a computation from a state s and D be a time-abstract scheduler such that for every $\pi \in Paths^{*}_{abs}$ it holds that*

$$\Pr_{s,D}(\pi) = \sum_{\{\sigma \in Q \mid \pi_\sigma = \pi \wedge \sigma \notin In_\epsilon\}} \Pr(\sigma).$$

Let $G \subseteq MS$. Then

$$LRA^{D}(s, \Diamond G) = LRA^{C_s}(s, \Diamond G).$$

Proof. Using Lemma 6.31,

$$\begin{aligned}
LRA^{D}(s, \Diamond G) &= \lim_{t\to\infty} \frac{1}{t} \int_{0}^{t} \sum_{\pi \in \mathfrak{L}_G} \Pr_{s,D}(\pi) \cdot P_u(Tr(\pi))\ du \\
\text{(by assumption)} \quad &= \lim_{t\to\infty} \frac{1}{t} \int_{0}^{t} \sum_{\pi \in \mathfrak{L}_G} \left(\sum_{\{\sigma \in Q \mid \pi_\sigma = \pi \wedge \sigma \notin In_\epsilon\}} \Pr(\sigma) \right) \cdot P_u(Tr(\pi))\ du \\
\text{(by Definition 6.12)} \quad &= \lim_{t\to\infty} \frac{1}{t} \int_{0}^{t} \sum_{\pi \in \mathfrak{L}_G} \sum_{\{\sigma \in Q \mid \pi_\sigma = \pi \wedge \sigma \notin In_\epsilon\}} \Pr(\sigma) \cdot P_u(Tr(\sigma))\ du \\
\text{(merging two sums)} \quad &= \lim_{t\to\infty} \frac{1}{t} \int_{0}^{t} \sum_{\{\sigma \in Q \mid \pi_\sigma \in \mathfrak{L}_G \wedge \sigma \notin In_\epsilon\}} \Pr(\sigma) \cdot P_u(Tr(\sigma))\ du \\
\text{(by Definition 6.32)} \quad &= LRA^{C_s}(s, \Diamond G).
\end{aligned}$$

□

Lemma 6.34 (Minimum LRA given by simple computation). *Let $G \subseteq MS$. Then, for every $s \in S$, there exists a simple computation C_s from s with*

$$LRA^{min}(s, \Diamond G) \quad = \quad LRA^{C_s}(s, \Diamond G).$$

Proof. Follows immediately from Lemmas 2.52, 6.14 and 6.33. ☐

Lemma 6.35 (Long run average by induced scheduler). *Let $C = (Q, \to, L, \sigma_0)$ be a non-terminating computation from $s \in S$. Then, for every $G \subseteq MS$,*

$$LRA^{D_C}(s, \Diamond G) \quad = \quad LRA^{C}(s, \Diamond G).$$

Proof. The result follows directly from Lemmas 6.33 and 6.17. ☐

6.6 Weakly simulating computations

At this point, we have completed Steps 1 and 3 of the proof plan in Section 6.1. We now turn our attention to Step 2, which is arguably the hardest part of the entire proof. In particular, we need to show that if s is weakly bisimilar to s' (with respect to the chosen weak bisimulation), G is closed under the chosen bisimulation and C_s is a non-terminating computation from s, then there exists a nonterminating computation $C_{s'}$ from s' with $eT^{C_s}(s, \Diamond G) \geq eT^{C_{s'}}(s', \Diamond G)$ and $LRA^{C_s}(s, \Diamond G) \geq LRA^{C_{s'}}(s', \Diamond G)$ (strictly speaking, we do not have to use the same $C_{s'}$ for both expected time and long run average, but it turns out that we can). Note that we can safely assume that C_s is non-terminating because we get a simple computation from Step 1. Furthermore, we need $C_{s'}$ to be non-terminating since it is required by the construction of the induced scheduler (see Definition 6.15).

In order to develop a sense of the problem, let us consider the construction of $C_{s'}$ for naive weak bisimulation. Consider the Markov automaton MA and the computation C_s from the state s presented in Figure 13. We want to construct $C_{s'}$ from the state s'.

The idea is simple enough. We construct $C_{s'}$ recursively: start with a single root node γ_0 corresponding to s' and expand the computation downwards by simulating the transitions in C_s one by one. To this end, consider the transition $\sigma_0 \xrightarrow{\alpha,1} \{(\sigma_1, 0.5), (\sigma_2, 0.5)\}$ in C_s. It can be simulated by the computation from s' depicted in Figure 14. Note that by the definition of $\asymp$, the distribution induced by the leaves of the computation on the right is equivalent to the distribution $\{(u, 0.5), (v, 0.5)\}$. Clearly, the nodes γ_1 and γ_2 correspond to the node σ_1, and the node γ_3 corresponds to the node σ_2.

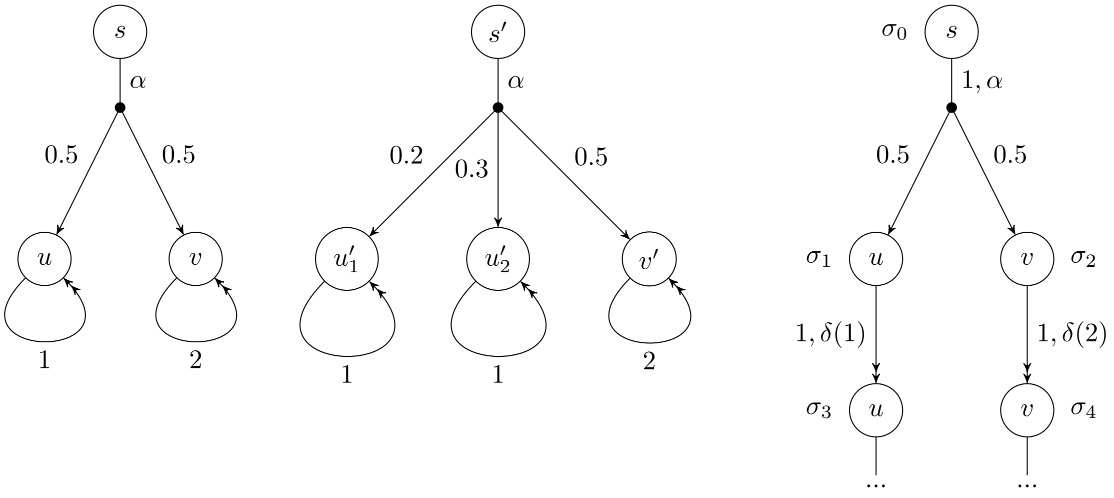


Figure 13: A Markov automaton (left) and a computation C_s from s (right).

Let us capture this correspondence by a mapping w from the nodes of C_s to *sets* of nodes of $C_{s'}$. Thus, $w(\sigma_1) = \{\gamma_1, \gamma_2\}$ and $w(\sigma_2) = \{\gamma_3\}$. Moreover, we can set $w(\sigma_0) = \{\gamma_0\}$.

Continuing the process recursively, we simulate the transition $\sigma_1 \xrightarrow{\delta(1),1} \sigma_2$ by attaching appropriate transitions *to every node in* $w(\sigma_1)$. We continue the process to infinity, simulating all the transitions in C_s one by one. The result is presented in Figure 15 (the dashed arrows denote the mapping w).

At this point, we might start noticing that the way we constructed $C_{s'}$ implies that:

1. For every node σ of C_s and every $\gamma \in w(\sigma)$, $Tr(\sigma) = Tr(\gamma)$.
2. For every node σ of C_s, $\Pr(\sigma) = \sum_{\gamma \in w(\sigma)} \Pr(\gamma)$.
3. The mapping w preserves the ancestor / descendant relationships between the nodes. For example, every node in $w(\sigma_3)$ must have a parent belonging to $w(\sigma_1)$, etc.

By exploiting these (and a couple of other) properties of $C_{s'}$, we can now prove that $eT^{C_s}(s, \Diamond G) \geq eT^{C_{s'}}(s', \Diamond G)$ and $LRA^{C_s}(s, \Diamond G) \geq LRA^{C_{s'}}(s', \Diamond G)$.

We could construct a separate proof for each kind of bisimulation. However, this would be wasteful because large chunks of the proofs would be identical. Fortunately, it turns out that we can factor the common parts out in the following way: we can define a set of conditions on $C_{s'}$ that are sufficient for $eT^{C_s}(s, \Diamond G) \geq eT^{C_{s'}}(s', \Diamond G)$ (or $LRA^{C_s}(s, \Diamond G) \geq LRA^{C_{s'}}(s', \Diamond G)$) to hold. The 3 observations above are a subset of these conditions. We then consider each bisimulation separately and prove that a computation satisfying the conditions can be constructed (very similar to how we have just done it for naive weak bisimulation).

One minor complication is that the conditions for minimum expected time and minimum long run average turn out to be slightly different, which leads us to define *three* sets of conditions on $C_{s'}$:

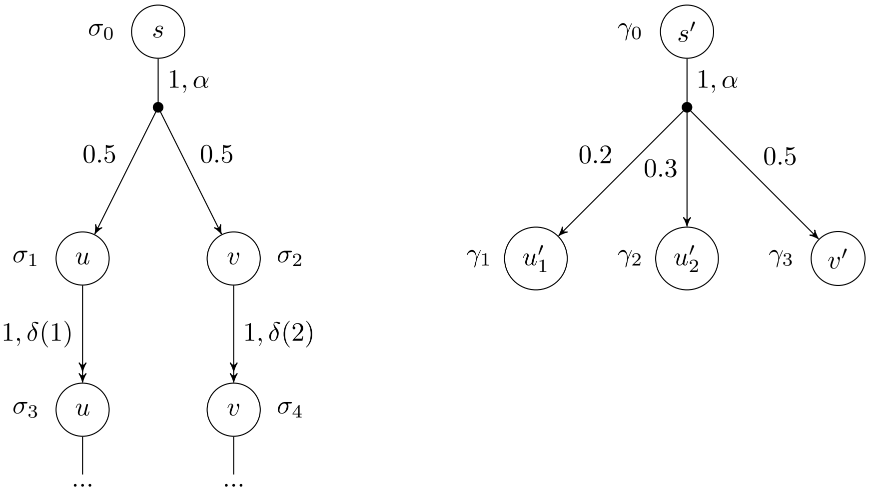


Figure 14: C_s (left) and a computation simulating the transition $\sigma_0 \xrightarrow{\alpha,1} \{(\sigma_1, 0.5), (\sigma_2, 0.5)\}$ (right).

1. Conditions A: those that are shared by minimum expected time and minimum long run average.
2. Conditions B: those that are specific to minimum expected time.
3. Conditions C: those that are specific to minimum long run average.

Thus, if $C_{s'}$ satisfies Conditions A and B, it must hold that $eT^{C_s}(s, \Diamond G) \geq eT^{C_{s'}}(s', \Diamond G)$, and if it satisfies Conditions A and C, it must hold that $LRA^{C_s}(s, \Diamond G) \geq LRA^{C_{s'}}(s', \Diamond G)$.

Here is our plan:

1. In Section 6.6 (the present one) we will define Conditions A. We call any computation satisfying Conditions A a *weakly simulating computation.*
2. In Section 6.7 we will define Conditions B and prove that Conditions A and B together imply $eT^{C_s}(s, \Diamond G) \geq eT^{C_{s'}}(s', \Diamond G)$.
3. In Section 6.8 we will define Conditions C and prove that Conditions A and C together imply $LRA^{C_s}(s, \Diamond G) \geq LRA^{C_{s'}}(s', \Diamond G)$.
4. In the following chapters we will consider each bisimulation separately and prove that given a computation C_s it is possible to construct a computation $C_{s'}$ such that Conditions A, B and C hold (for some bisimulations this goal can only be achieved partially).

We now turn to a formal definition of a weakly simulating computation. We will also prove several lemmas that will become useful in the following sections.

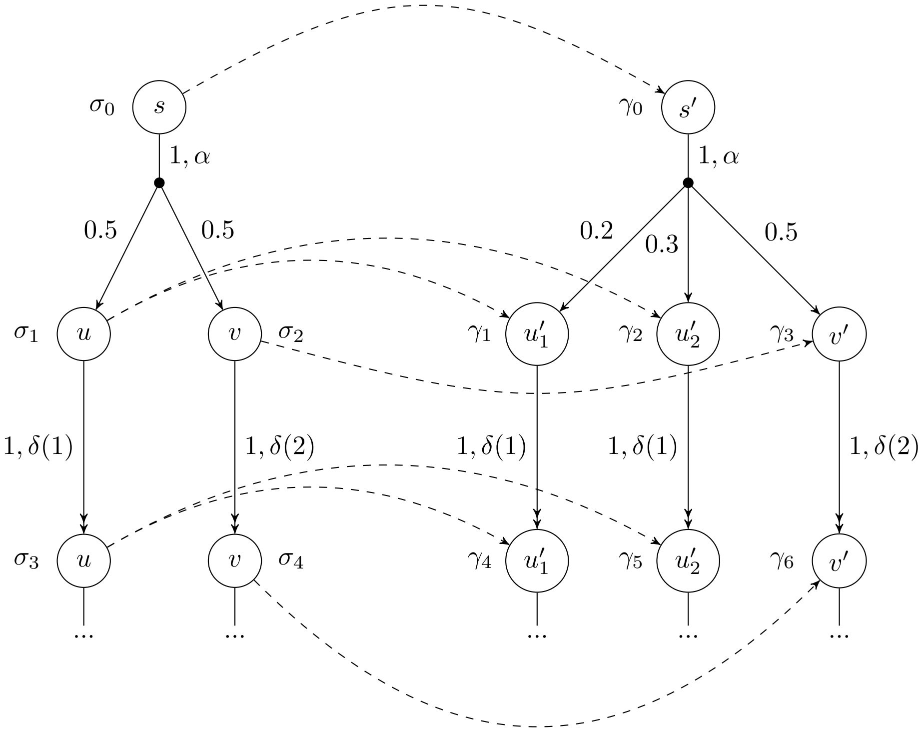


Figure 15: C_s (left) and $C_{s'}$ (right).

Definition 6.36 (Weakly simulating computation). Let $C = (Q, \to, L, \sigma_0)$ be a non-terminating computation. Then a non-terminating computation $C' = (Q', \to', L', \gamma_0)$ is said to *weakly simulate* C, denoted $C \preceq C'$, if there exists a function $w : Q \mapsto 2^{Q'}$ such that:

1. for every $\sigma \in Q$ and $\gamma \in w(\sigma)$,

$$Tr(\sigma) \quad = \quad Tr(\gamma),$$

2. for every $\sigma \in Q$,

$$\Pr(\sigma) \quad = \quad \sum_{\gamma \in w(\sigma)} \Pr(\gamma),$$

3. for every $\sigma \in Q$, $w(\sigma)$ is independent,
4. for every $\sigma \in Q$ and $\sigma' \in Children(\sigma)$,

$$w(\sigma') \quad \subseteq \quad Children^*(w(\sigma))$$

(where $Children^*(w(\sigma)) = \bigcup_{\gamma \in w(\sigma)} Children^*(\gamma)$), and

5. for every $\sigma \in Q$ and $\sigma' \notin Children^* (\sigma)$,

$$w(\sigma') \cap Children^* (w(\sigma)) \quad = \quad \emptyset.$$

Lemma 6.37 (Images of distrinct nodes are disjoint). *Let C, C' and w be as in Definition 6.36. Then, for every $\sigma, \sigma' \in Q$,*

$$\sigma \neq \sigma' \quad \textit{implies} \quad w(\sigma) \cap w(\sigma') = \emptyset.$$

Proof. Suppose $\sigma \neq \sigma'$. Then it must be the case that either $\sigma \notin Children^* (\sigma')$ or $\sigma' \notin Children^* (\sigma)$. Without loss of generality, suppose $\sigma' \notin Children^* (\sigma)$. But then, by Condition 5 of Definition 6.36,

$$w(\sigma') \cap Children^* (w(\sigma)) \quad = \quad \emptyset.$$

But $w(\sigma) \subseteq Children^* (w(\sigma))$, and the result follows. □

Lemma 6.38 (w preserves independence). *Let C, C' and w be as in Definition 6.36. Then, for every independent $B \subseteq Q$, $w(B)$ is independent.*

Proof. Suppose $B \subseteq Q$ is independent. Let $\gamma, \gamma' \in w(B)$ be distinct. We shall prove that

$$\gamma' \quad \notin \quad Children^* (\gamma).$$

Let $\sigma, \sigma' \in B$ be such that $\gamma \in w(\sigma)$ and $\gamma' \in w(\sigma')$. If $\sigma = \sigma'$, the result follows from Condition 3 of Definition 6.36. Thus, suppose $\sigma \neq \sigma'$. Then, since B is independent, it follows that $\sigma' \notin Children^* (\sigma)$, so that, by Condition 5 of Definition 6.36,

$$w(\sigma') \cap Children^* (w(\sigma)) \quad = \quad \emptyset.$$

But $\gamma' \in w(\sigma')$ and $\gamma \in w(\sigma)$, which implies that $\gamma' \notin Children^* (\gamma)$, as desired. □

To understand the next lemma, assume $\sigma \in Q$ and $\gamma \in w(\sigma)$. Then, suppose that we start in γ and traverse the computation downwards, always choosing the next transition according to the transition probabilities. Then, we will encounter some node from $w(Children(\sigma))$ (i.e., a node corresponding to some child of σ) with probability one.

Lemma 6.39. *Let C, C' and w be as in Definition 6.36. Then, for every $\sigma \in Q$ and $\gamma \in w(\sigma)$,*

$$\Pr(\gamma) \quad = \sum_{\gamma' \in Children^*(\gamma) \cap w(Children(\sigma))} \Pr(\gamma').$$

Proof. When reading this proof, the reader might find it useful to take a look at Figure 16.

Suppose $\sigma \in Q$. First of all, $Children(\sigma)$ is a independent set of nodes, so that, by Lemma 6.38, $w(Children(\sigma))$ is also independent. Then clearly $Children^*(\gamma) \cap w(Children(\sigma))$ is independent for every $\gamma \in w(\sigma)$. Then, for every $\gamma \in w(\sigma)$,

$$\Pr(\gamma) \quad \geq \sum_{\gamma' \in Children^*(\gamma) \cap w(Children(\sigma))} \Pr(\gamma').$$

This gives us the first half of the result.

Next,

$$\begin{aligned}
\sum_{\gamma'' \in w(\sigma)} \Pr(\gamma'') &= \Pr(\sigma) \\
\text{(by Definitions 6.2 and 6.4)} &= \sum_{\sigma' \in Children(\sigma)} \Pr(\sigma') \\
\text{(by Condition 2 of Definition 6.36)} &= \sum_{\sigma' \in Children(\sigma)} \sum_{\gamma' \in w(\sigma')} \Pr(\gamma') \\
\text{(by Lemma 6.37)} &= \sum_{\gamma' \in w(Children(\sigma))} \Pr(\gamma').
\end{aligned}$$

Now, for every $\gamma \in w(\sigma)$,

$$\begin{aligned}
\Pr(\gamma) &= \left(\sum_{\gamma'' \in w(\sigma)} \Pr(\gamma'')\right) - \left(\sum_{\gamma'' \in w(\sigma) \setminus \{\gamma\}} \Pr(\gamma'')\right) \\
(*) &\leq \left(\sum_{\gamma' \in w(Children(\sigma))} \Pr(\gamma')\right) - \left(\sum_{\gamma'' \in w(\sigma) \setminus \{\gamma\}} \; \sum_{\gamma' \in Children^*(\gamma'') \cap w(Children(\sigma))} \Pr(\gamma')\right) \\
&= \left(\sum_{\gamma' \in w(Children(\sigma))} \Pr(\gamma')\right) - \left(\sum_{\gamma' \in Children^*(w(\sigma) \setminus \{\gamma\}) \cap w(Children(\sigma))} \Pr(\gamma')\right) \\
(**) &= \sum_{\gamma' \in Children^*(\gamma) \cap w(Children(\sigma))} \Pr(\gamma').
\end{aligned}$$

$(*)$ In this step we use the fact that $\Pr(\gamma) \geq \sum_{\gamma' \in Children^*(\gamma) \cap w(Children(\sigma))} \Pr(\gamma')$ for every σ and $\gamma \in w(\sigma)$, as was shown at the very beginning of this proof.

$(**)$ In this step we use the fact that $w(Children(\sigma)) \subseteq Children^*(w(\sigma))$ (Condition 4 of Definition 6.36).

This completes the proof. □

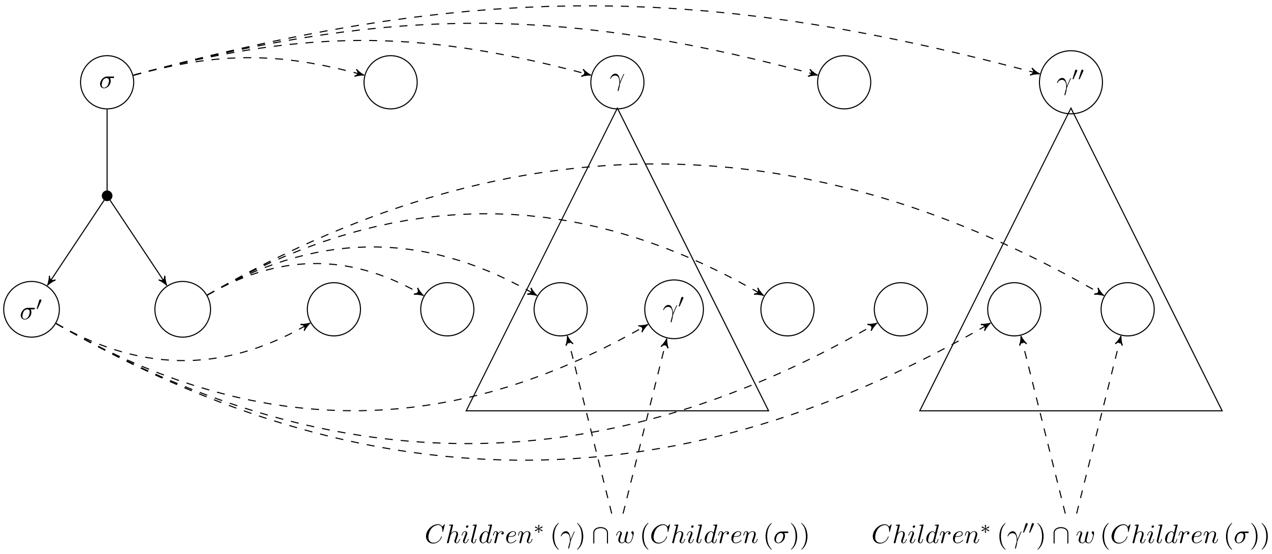


Figure 16: Illustration to Lemma 6.39.

6.7 Minimum expected time in weakly simulating computations

In this section we will prove that Conditions A (defined in Section 6.6) and B (defined in the statement of Lemma 6.40) together imply that $eT^{C}(s, \Diamond G) \geq eT^{C'}(s', \Diamond G)$.

Lemma 6.40 (Expected time in weakly simulating computation). *Let C, C' and w be as in Definition 6.36, and let $G \subseteq S$. Suppose that for every $\sigma \in Q, \gamma \in Q'$ it holds that*

$$L(\sigma) \in G \wedge \gamma \in w(\sigma) \quad \textit{implies} \quad L'(\gamma) \in G.$$

Then,

$$eT^{C}(s, \Diamond G) \quad \geq \quad eT^{C'}(s', \Diamond G).$$

Proof. First of all, the claim is trivially true if $eT^{C}(s, \Diamond G) = \infty$. Therefore, suppose $eT^{C}(s, \Diamond G) < \infty$.

To avoid confusion, let

$$\mathcal{G} \quad = \quad \{\sigma \in Q \mid \pi_\sigma \in \mathfrak{G} \wedge \sigma \notin In_\epsilon\}$$

and

$$\mathcal{G}' \quad = \quad \{\gamma \in Q' \mid \pi_\gamma \in \mathfrak{G} \wedge \gamma \notin In_\epsilon\},$$

where

$$\mathfrak{G} \quad = \quad \{\pi \in Paths^*_{abs} \mid \pi_{\downarrow} \in G \ \wedge \ \forall i \in \{0, ..., |\pi| - 1\} \ . \ \pi[i] \notin G\}$$

was defined in Lemma 6.19.

By Definition 6.20, $eT^C(s, \Diamond G) < \infty$ implies that

$$\sum_{\sigma \in \mathcal{G}} \Pr(\sigma) \quad = \quad 1.$$

Then,

$$\begin{aligned} 1 &= \sum_{\sigma \in \mathcal{G}} \Pr(\sigma) \\ \text{(by Definition 6.36)} &= \sum_{\sigma \in \mathcal{G}} \sum_{\gamma \in w(\sigma)} \Pr(\gamma) \\ \text{(by Lemma 6.37)} &= \sum_{\gamma \in w(\mathcal{G})} \Pr(\gamma). \end{aligned}$$

Now, consider a node $\gamma \in w(\mathcal{G})$. For every $\sigma \in \mathcal{G}$ it holds that $\pi_\sigma \in \mathfrak{G}$, which implies that $L(\sigma) \in G$. By assumption, it must then hold that $L'(\gamma) \in G$. Then, γ must reside in a subtree rooted at some node $\gamma_{top} \in \mathcal{G}'$ (which is the topmost node labelled with a state in G on the path from the root to γ). Then, continuing the chain of equalities above:

$$\begin{aligned} 1 &= \sum_{\gamma \in w(\mathcal{G})} \Pr(\gamma) \\ &= \sum_{\gamma_{top} \in \mathcal{G}'} \sum_{\gamma \in Children^*(\gamma_{top}) \cap w(\mathcal{G})} \Pr(\gamma). \end{aligned}$$

At this point, observe that, by Lemmas 6.21 and 6.38, $w(\mathcal{G})$ is independent. Together with the fact that $\sum_{\gamma \in w(\mathcal{G})} \Pr(\gamma) = 1$ this means that for every $\gamma_{top} \in \mathcal{G}'$,

$$\sum_{\gamma \in Children^*(\gamma_{top}) \cap w(\mathcal{G})} \Pr(\gamma) \quad = \quad \Pr(\gamma_{top}).$$

Therefore,

$$\begin{aligned} 1 &= \sum_{\gamma_{top} \in \mathcal{G}'} \sum_{\gamma \in Children^*(\gamma_{top}) \cap w(\mathcal{G})} \Pr(\gamma) \\ &= \sum_{\gamma_{top} \in \mathcal{G}'} \Pr(\gamma_{top}). \end{aligned}$$

Then, by Definition 6.20,

$$eT^{C'}\left(s', \Diamond G\right) \quad = \quad \sum_{\gamma_{top} \in \mathcal{G}'} eT\left(\gamma_{top}\right) \cdot \Pr\left(\gamma_{top}\right).$$

Now, observe that, by Condition 1 of Definition 6.36, for every $\sigma \in Q$ and $\gamma \in Q'$, $\gamma \in w(\sigma)$ implies $Tr(\sigma) = Tr(\gamma)$, which in turn implies $eT(\sigma) = eT(\gamma)$ by Lemma 6.22.

Thus,

$$\begin{aligned}
eT^{C}\left(s, \Diamond G\right) &= \sum_{\sigma \in \mathcal{G}} eT(\sigma) \cdot \Pr(\sigma) \\
\text{(by Definition 6.36)} &= \sum_{\sigma \in \mathcal{G}} eT(\sigma) \cdot \sum_{\gamma \in w(\sigma)} \Pr(\gamma) \\
(\gamma \in w(\sigma) \text{ implies } eT(\sigma) = eT(\gamma)) &= \sum_{\sigma \in \mathcal{G}} \sum_{\gamma \in w(\sigma)} eT(\gamma) \cdot \Pr(\gamma) \\
\text{(as before)} &= \sum_{\gamma \in w(\mathcal{G})} eT(\gamma) \cdot \Pr(\gamma). \\
\text{(as before)} &= \sum_{\gamma_{top} \in \mathcal{G}'} \sum_{\gamma \in Children^{*}(\gamma_{top}) \cap w(\mathcal{G})} eT(\gamma) \cdot \Pr(\gamma) \\
(\text{since } eT(\gamma_{top}) \leq eT(\gamma)) &\geq \sum_{\gamma_{top} \in \mathcal{G}'} eT(\gamma_{top}) \cdot \sum_{\gamma \in Children^{*}(\gamma_{top}) \cap w(\mathcal{G})} \Pr(\gamma) \\
\text{(as before)} &= \sum_{\gamma_{top} \in \mathcal{G}'} eT(\gamma_{top}) \cdot \Pr(\gamma_{top}) \\
&= eT^{C'}\left(s', \Diamond G\right),
\end{aligned}$$

as desired. □

Lemma 6.41 (Preservation of minimum expected time). *Suppose $G \subseteq S$. Let $s, s' \in S$ be such that for every simple computation $C = (Q, \rightarrow, L, \sigma_0)$ from s there exists a weakly simulating computation $C' = (Q', \rightarrow', L', \gamma_0)$ from s' such that the corresponding function w satisfies the following requirement: for every $\sigma \in Q, \gamma \in Q'$ it holds that*

$$L(\sigma) \in G \wedge \gamma \in w(\sigma) \quad \textit{implies} \quad L'(\gamma) \in G.$$

Then,

$$eT^{min}\left(s, \Diamond G\right) \quad \geq \quad eT^{min}\left(s', \Diamond G\right).$$

Proof. By Lemma 6.24, there exists a simple computation C from s such that

$$eT^{min}\left(s, \Diamond G\right) \quad = \quad eT^{C}\left(s, \Diamond G\right).$$

Then:

$$\begin{array}{rcl} eT^{min}\left(s,\Diamond G\right) & = & eT^{C}\left(s,\Diamond G\right) \\ \text{(by assumption and Lemma 6.40)} & \geq & eT^{C'}\left(s',\Diamond G\right) \\ \text{(by Lemma 6.25)} & = & eT^{D_{C'}}\left(s',\Diamond G\right) \\ \text{(by Definition 2.48)} & \geq & eT^{min}\left(s',\Diamond G\right), \end{array}$$

as desired. □

6.8 Long run average in weakly simulating computations

In this section we will prove that Conditions A (defined in Section 6.6) and C (defined in the statement of Lemma 6.42) together imply that $LRA^{C}\left(s,\Diamond G\right) \geq LRA^{C'}\left(s',\Diamond G\right)$ (in fact, we will even prove that the conditions are sufficient for the two values to be exactly equal).

Lemma 6.42 (Long run average in weakly simulating computation). *Let C, C' and w be as in Definition 6.36, and let $G \subseteq MS$. Suppose that for every $\sigma \in Q$ with $L\left(\sigma\right) \in MS$, every $\gamma \in w\left(\sigma\right)$ and every $\gamma' \in Children^{*}\left(\gamma\right)$ with $L'\left(\gamma'\right) \in MS$ and $Tr\left(\gamma\right) = Tr\left(\gamma'\right)$ (i.e. for every γ' with $L'\left(\gamma'\right) \in MS$ reachable from γ via zero or more τ- and ϵ-transitions) it holds that*

$$L\left(\sigma\right) \in G \quad \textit{if and only if} \quad L'\left(\gamma'\right) \in G.$$

Then,

$$LRA^{C}\left(s,\Diamond G\right) \quad = \quad LRA^{C'}\left(s',\Diamond G\right).$$

Proof. From Definition 6.32 it is clear that it is enough to prove that, for every $u \in \mathbb{R}_{\geq 0}$,

$$\sum_{\{\sigma\in Q\mid \pi_{\sigma}\in\mathfrak{L}_{G}\wedge\sigma\notin In_{\epsilon}\}} \Pr\left(\sigma\right)\cdot P_{u}\left(Tr\left(\sigma\right)\right) \quad = \quad \sum_{\{\gamma\in Q'\mid \pi_{\gamma}\in\mathfrak{L}_{G}\wedge\gamma\notin In_{\epsilon}\}} \Pr\left(\gamma\right)\cdot P_{u}\left(Tr\left(\gamma\right)\right).$$

To simplify notation, let

$$Q_{G} \quad = \quad \left\{\sigma \in Q \mid \pi_{\sigma} \in \mathfrak{L}_{G} \wedge \sigma \notin In_{\epsilon}\right\}.$$

Then,

$$\begin{aligned}\sum_{\sigma\in Q_G}\Pr(\sigma)\cdot P_u\left(Tr(\sigma)\right) &= \sum_{\sigma\in Q_G}\left(\sum_{\gamma\in w(\sigma)}\Pr(\gamma)\right)\cdot P_u\left(Tr(\sigma)\right)\\ \text{(by Condition 1 of Definition 6.36)} &= \sum_{\sigma\in Q_G}\sum_{\gamma\in w(\sigma)}\Pr(\gamma)\cdot P_u\left(Tr(\gamma)\right)\\ \text{(by Lemma 6.37)} &= \sum_{\gamma\in w(Q_G)}\Pr(\gamma)\cdot P_u\left(Tr(\gamma)\right).\end{aligned}$$

Thus, we need to prove that, for every $u\in\mathbb{R}_{\geq 0}$,

$$\sum_{\gamma\in w(Q_G)}\Pr(\gamma)\cdot P_u\left(Tr(\gamma)\right) = \sum_{\{\gamma\in Q'\mid\pi_\gamma\in\mathfrak{L}_G\wedge\gamma\notin In_\epsilon\}}\Pr(\gamma)\cdot P_u\left(Tr(\gamma)\right).$$

Let us now investigate the connection between the sets $w(Q_G)$ and $\{\gamma\in Q'\mid\pi_\gamma\in\mathfrak{L}_G\wedge\gamma\notin In_\epsilon\}$. When reading the next paragraphs, the reader is advised to take a look at Figure 17.

First, consider a node $\gamma\in w(\sigma)$ with $\sigma\in Q_G$, which means that $\pi_\sigma\in\mathfrak{L}_G$ and $\sigma\notin In_\epsilon$ (note that the latter condition is actually redundant since C is simple and, therefore, does not contain ϵ-transitions). Let $\sigma_p = Parent(\sigma)$. Because $\pi_\sigma\in\mathfrak{L}_G$, $L(\sigma_p)\in G$ and the transition from σ_p to σ must be labelled with $\alpha=\delta\left(E\left(L(\sigma_p)\right)\right)$.

Now, Condition 4 of Definition 6.36 implies that there exists a node $\gamma_{start}\in Q'$ such that $\gamma_{start}\in w(\sigma_p)$ and $\gamma\in Children^*(\gamma_{start})$. Furthermore, from Condition 1 of Definition 6.36 it follows that

$$\begin{aligned}Tr(\gamma) &= Tr(\sigma)\\ &= Tr(\sigma_p)\circ\alpha\\ &= Tr(\gamma_{start})\circ\alpha.\end{aligned}$$

In other words, the path from γ_{start} to γ in C' contains exactly one α-transition, all other transitions being labelled with τ or ϵ (note also that it implies that $\gamma\neq\gamma_{start}$). Let $Start(\gamma)=\gamma_{start}$, and let $Src(\gamma)$ and $Dst(\gamma)$ be the source and target nodes of the α-transition on the path from $Start(\gamma)$ to γ.

By the same reasoning, $Start(\gamma)$, $Src(\gamma)$ and $Dst(\gamma)$ can be defined for every $\gamma\in w(Q_G)$.

We shall now prove that

$$\{\gamma_{dst}\in Q'\mid\pi_{\gamma_{dst}}\in\mathfrak{L}_G\wedge\gamma_{dst}\notin In_\epsilon\} = Dst\left(w(Q_G)\right).$$

($\Longleftarrow$) First, consider the node $Dst(\gamma)$ for some $\gamma\in w(Q_G)$. By construction, $Dst(\gamma)$ is the target of an α-transition (for some $\alpha\in Act^{\chi}\setminus Act$), and thus $Dst(\gamma)\notin In_\epsilon$. Moreover, for the corresponding $Src(\gamma)$ it holds that $L'\left(Src(\gamma)\right)\in G$ since $L(\sigma_p)\in G\subseteq MS$, $Start(\gamma)\in w(\sigma_p)$, $L'\left(Src(\gamma)\right)\in MS$ and $Tr\left(Src(\gamma)\right)=Tr\left(Start(\gamma)\right)$. Therefore, $\pi_{Dst(\gamma)}\in\mathfrak{L}_G$.

($\Longrightarrow$) Conversely, suppose $\gamma_{dst} \in Q'$ with $\pi_{\gamma_{dst}} \in \mathfrak{L}_G$ and $\gamma_{dst} \notin In_\epsilon$. We need to find a node $\gamma \in w(Q_G)$ such that $\gamma_{dst} = Dst(\gamma)$. Let $\gamma_{src} = Parent(\gamma_{dst})$. It must be the case that γ_{dst} is the target of a transition labelled with $\alpha = \delta(E(L'(\gamma_{src})))$. Moreover, $L'(\gamma_{src}) \in G$. Now, on the path from the root node γ_0 to γ_{src}, there must be at least one node belonging to $w(Q)$. Let γ_{start} be the last such node. Let σ_p be such that $w(\sigma_p) = \gamma_{start}$. Next, from Lemma 6.39 it follows that there must exist at least one $\gamma \in Children^*(\gamma_{dst})$ with $\gamma \in w(Children(\sigma_p))$. Let $\sigma \in Children(\sigma_p)$ be such that $\gamma = w(\sigma)$. Then $Tr(\sigma) = Tr(\gamma)$, $Tr(\sigma_p) = Tr(\gamma_{start})$ and $Tr(\sigma)$ is longer than $Tr(\sigma_p)$ by at most one action. We know that there is an α-transition on the path from γ_{start} to γ. It follows that every other transition on the path is labelled with either τ or ϵ, and furthermore, the transition from σ_p to σ is labelled with α. Then, $L(\sigma_p) \in G$ since $L(\sigma_p) \in MS$, $\gamma_{start} \in w(\sigma_p)$, $L'(\gamma_{src}) \in G \subseteq MS$ and $Tr(\gamma_{start}) = Tr(\gamma_{src})$. It immediately follows that $\sigma \in Q_G$, $\gamma \in w(Q_G)$ and $\gamma_{dst} = Dst(\gamma)$.

We have now established that

$$\{\gamma \in Q' \mid \pi_\gamma \in \mathfrak{L}_G \land \gamma \notin In_\epsilon\} \quad = \quad Dst(w(Q_G)).$$

At this point, let us take a step back and observe what we have achieved so far:

$$\begin{aligned} LRA^C(s, \Diamond G) \quad &= \quad LRA^{C'}(s', \Diamond G) \\ &\Uparrow \\ \forall u \in \mathbb{R}_{\geq 0}. \sum_{\gamma \in w(Q_G)} \Pr(\gamma) \cdot P_u(Tr(\gamma)) \quad &= \quad \sum_{\{\gamma \in Q' \mid \pi_\gamma \in \mathfrak{L}_G \land \gamma \notin In_\epsilon\}} \Pr(\gamma) \cdot P_u(Tr(\gamma)) \\ &\Updownarrow \\ \forall u \in \mathbb{R}_{\geq 0}. \sum_{\gamma \in w(Q_G)} \Pr(\gamma) \cdot P_u(Tr(\gamma)) \quad &= \quad \sum_{\gamma_{dst} \in Dst(w(Q_G))} \Pr(\gamma_{dst}) \cdot P_u(Tr(\gamma_{dst})). \end{aligned}$$

Now, on the left we sum over all $\gamma \in w(Q_G)$. For every such γ, there is the corresponding $Dst(\gamma)$. Furthermore, several such nodes $\gamma \in w(Q_G)$ can have the same value of $Dst(\gamma)$ (see Figure 18), so that we can split the sum $\sum_{\gamma \in w(Q_G)}$ into the double sum $\sum_{\gamma_{dst} \in Dst(w(Q_G))} \sum_{\{\gamma \in w(Q_G) \mid Dst(\gamma) = \gamma_{dst}\}}$ to get:

$$\begin{aligned} LRA^C(s, \Diamond G) \quad &= \quad LRA^{C'}(s', \Diamond G) \\ &\Uparrow \\ \forall u. \sum_{\gamma \in w(Q_G)} \Pr(\gamma) P_u(Tr(\gamma)) \quad &= \quad \sum_{\gamma_{dst} \in Dst(w(Q_G))} \Pr(\gamma_{dst}) P_u(Tr(\gamma_{dst})) \\ &\Updownarrow \\ \forall u. \sum_{\gamma_{dst} \in Dst(w(Q_G))} \sum_{\{\gamma \in w(Q_G) \mid Dst(\gamma) = \gamma_{dst}\}} \Pr(\gamma) P_u(Tr(\gamma)) \quad &= \quad \sum_{\gamma_{dst} \in Dst(w(Q_G))} \Pr(\gamma_{dst}) P_u(Tr(\gamma_{dst})). \end{aligned}$$

Next, for every $\gamma \in w(Q_G)$ it holds that $Tr(Dst(\gamma)) = Tr(\gamma)$. Thus,

$$\begin{array}{rcl}
LRA^{C}\left(s, \Diamond G\right) & = & LRA^{C'}\left(s', \Diamond G\right) \\
 & \Uparrow & \\
\forall u. \quad \sum_{\gamma_{dst} \in Dst(w(Q_G))} \sum_{\{\gamma \in w(Q_G) \mid Dst(\gamma) = \gamma_{dst}\}} \Pr\left(\gamma\right) P_u\left(Tr\left(\gamma\right)\right) & = & \sum_{\gamma_{dst} \in Dst(w(Q_G))} \Pr\left(\gamma_{dst}\right) P_u\left(Tr\left(\gamma_{dst}\right)\right) \\
 & \Updownarrow & \\
\forall u. \quad \sum_{\gamma_{dst} \in Dst(w(Q_G))} P_u\left(Tr\left(\gamma_{dst}\right)\right) \sum_{\{\gamma \in w(Q_G) \mid Dst(\gamma) = \gamma_{dst}\}} \Pr\left(\gamma\right) & = & \sum_{\gamma_{dst} \in Dst(w(Q_G))} P_u\left(Tr\left(\gamma_{dst}\right)\right) \Pr\left(\gamma_{dst}\right) \\
 & \Uparrow & \\
\forall \gamma_{dst} \in Dst\left(w\left(Q_G\right)\right) \quad \sum_{\{\gamma \in w(Q_G) \mid Dst(\gamma) = \gamma_{dst}\}} \Pr\left(\gamma\right) & = & \Pr\left(\gamma_{dst}\right).
\end{array}$$

Therefore, in order to finish the proof, we need to show that for every $\gamma_{dst} \in Dst\left(w\left(Q_G\right)\right)$ it holds that

$$\Pr\left(\gamma_{dst}\right) = \sum_{\{\gamma \in w(Q_G) \mid Dst(\gamma) = \gamma_{dst}\}} \Pr\left(\gamma\right).$$

Indeed, suppose $\gamma_{dst} \in Dst\left(\gamma\right)$ for some $\gamma = w\left(Q_G\right)$, and let $\sigma \in Q$ be such that $\gamma = w\left(\sigma\right)$. Furthermore, let $\sigma_p = Parent\left(\sigma\right)$. By construction,

$$\left\{\gamma \in w\left(Q_G\right) \mid Dst\left(\gamma\right) = \gamma_{dst}\right\} = Children^{*}\left(\gamma_{dst}\right) \cap w\left(Children\left(\sigma_p\right)\right).$$

Now, by Lemma 6.39,

$$\Pr\left(Start\left(\gamma\right)\right) = \sum_{\gamma' \in Children^{*}(Start(\gamma)) \cap w(Children(\sigma_p))} \Pr\left(\gamma'\right)$$

and by Lemma 6.38, $Children^{*}\left(Start\left(\gamma\right)\right) \cap w\left(Children\left(\sigma_p\right)\right)$ is independent. We can interpret these two facts as follows: if we traverse C' from $Start\left(\gamma\right)$ downwards (respecting the transition probabilities), we encounter *exactly one* node belonging to $w\left(Children\left(\sigma_p\right)\right)$ with probability one. But there can be no $\gamma' \in w\left(Children\left(\sigma_p\right)\right)$ on the path from $Start\left(\gamma\right)$ to $Dst\left(\gamma\right)$ because the transition $InTrans\left(Dst\left(\gamma\right)\right)$ is the first transition on the path that is not labelled with τ or ϵ. Thus,

$$\begin{aligned}
\Pr\left(\gamma_{dst}\right) &= \sum_{\gamma \in Children^{*}(\gamma_{dst}) \cap w(Children(\sigma_p))} \Pr\left(\gamma\right) \\
&= \sum_{\{\gamma \in w(Q_G) \mid Dst(\gamma) = \gamma_{dst}\}} \Pr\left(\gamma\right).
\end{aligned}$$

This completes the proof. □

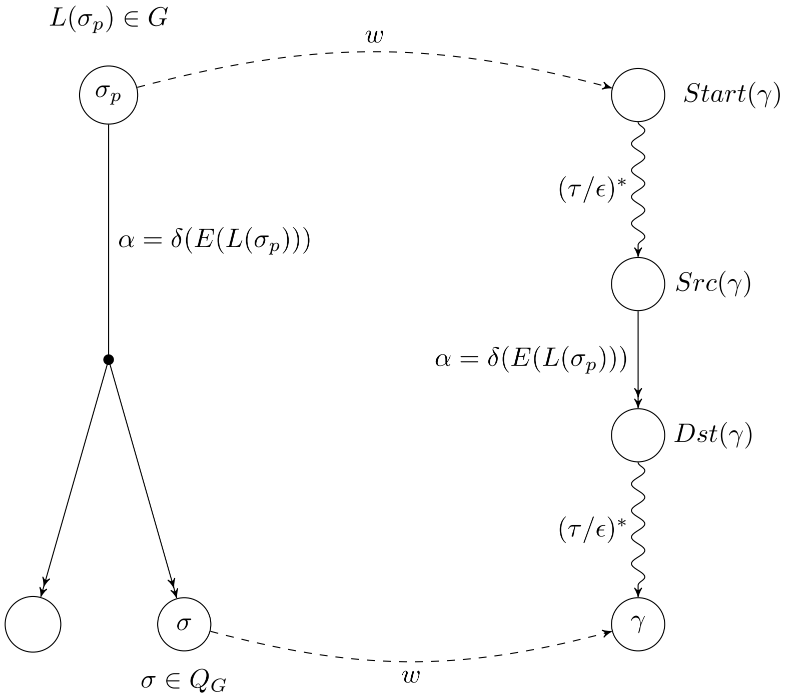


Figure 17: Illustration to Lemma 6.42 (first part).

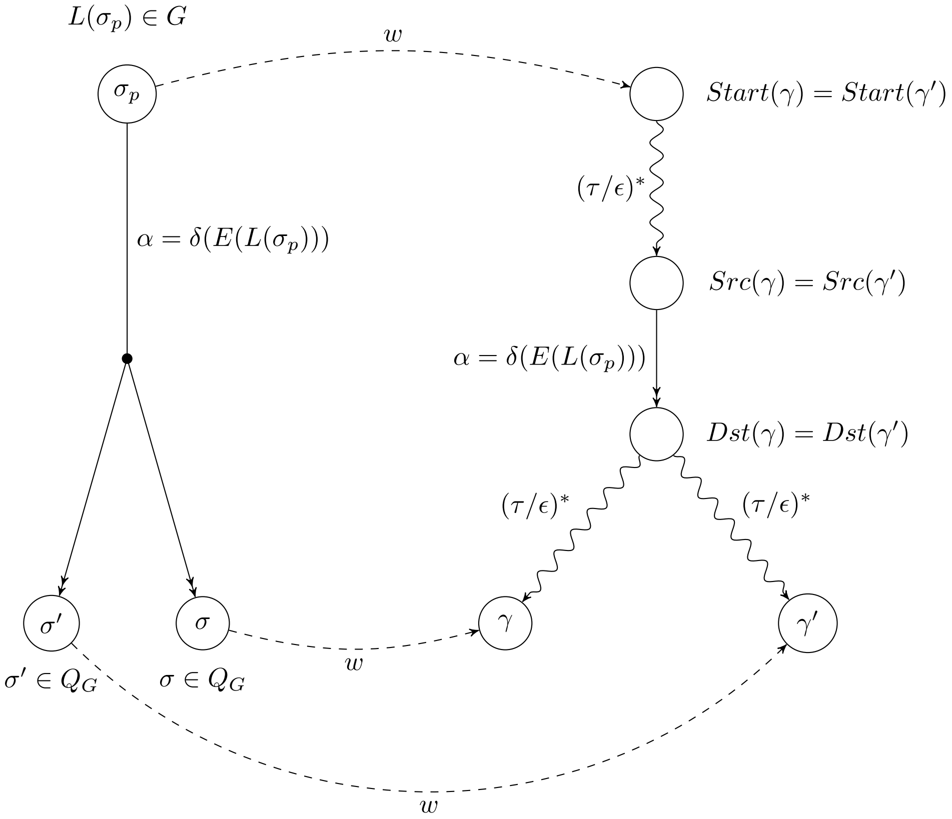


Figure 18: Illustration to Lemma 6.42 (second part).

Lemma 6.43 (Preservation of long run average). *Suppose* $G \subseteq S$. *Let* $s, s' \in S$ *be such that for every simple computation* $C = (Q, \rightarrow, L, \sigma_0)$ *from* s *there exists a weakly simulating computation* $C' = (Q', \rightarrow', L', \gamma_0)$ *from* s' *such that the corresponding function* w *satisfies the following requirement: for every* $\sigma \in Q$ *with* $L(\sigma) \in MS$, *every* $\gamma \in w(\sigma)$ *and every* $\gamma' \in Children^*(\gamma)$ *with* $L'(\gamma') \in MS$ *and* $Tr(\gamma) = Tr(\gamma')$ *(i.e. for every* γ' *with* $L'(\gamma') \in MS$ *reachable from* γ *via zero or more* τ*- and* ϵ*-transitions) it holds that*

$$L(\sigma) \in G \quad \textit{if and only if} \quad L'(\gamma') \in G.$$

Then,

$$LRA^{min}(s, \Diamond G) \quad \geq \quad LRA^{min}(s', \Diamond G).$$

Proof. First of all, observe that for every $s \in S$,

$$LRA^{min}(s, \Diamond G) \quad = \quad LRA^{min}(s, \Diamond(G \cap MS)).$$

Now, by Lemma 6.34, there exists a simple computation C from s such that

$$LRA^{min}(s, \Diamond(G \cap MS)) \quad = \quad LRA^{C}(s, \Diamond(G \cap MS)).$$

Then:

$$\begin{aligned} LRA^{min}(s, \Diamond G) &= LRA^{C}(s, \Diamond(G \cap MS)) \\ \text{(by assumption and Lemma 6.42)} &= LRA^{C'}(s', \Diamond(G \cap MS)) \\ \text{(by Lemma 6.35)} &= LRA^{D_{C'}}(s', \Diamond(G \cap MS)) \\ \text{(by Definition 2.51)} &\geq LRA^{min}(s', \Diamond(G \cap MS)) \\ &= LRA^{min}(s', \Diamond G), \end{aligned}$$

as desired. □

7 Property preservation under naive weak bisimulation and branching bisimulation

The goal of this chapter is to finish the proof that minimum expected time and minimum long run average are preserved under both naive weak and branching bisimulations. We refer the reader to the discussion at the beginning of Chapter 4 for an explanation why we need to treat each bisimulation separately. The reader should also be familiar with the contents of Chapter 6, particularly the discussions in Sections 6.1 and 6.6.

Recall that by now we have all parts of the proof in place, except one: we need to show that if $s \asymp s'$ (respectively $s \simeq s'$), G is closed under $\asymp$ (respectively $\simeq$) and C_s is a non-terminating computation from s, there exists a non-terminating computation $C_{s'}$ from s' such that Conditions A, B and C hold (the conditions are defined in Sections 6.6, 6.7 and 6.8).

Fortunately, we can save ourselves some work by noticing that naive weak bisimulation and branching bisimulation are quite similar: indeed, both can be characterized as follows: a transition $s \xrightarrow{\alpha} \mu$ is simulated by an *α-computation* C from the bisimilar state s' such that $\mu_C \equiv_{\asymp} \mu$ (respectively $\mu_C \equiv_{\simeq} \mu$). The difference between the two bisimulations lies in the restrictions they impose on C - for example, in the case of naive weak bisimulation C must correspond to a finite convex linear combination of transition trees, etc. As it turns out, we can prove that an appropriate $C_{s'}$ can be constructed for *every* bisimulation that matches a transition with a computation.

The plan is now as follows:

1. Define a class of bisimulations, called *relaxed bisimulations*, as follows: an equivalence relation $\mathcal{R}$ is a *relaxed bisimulation* if whenever $(s, s') \in \mathcal{R}$ and $s \xrightarrow{\alpha} \mu$, then there exists an α-computation C from s' such that $\mu_C \equiv_{\mathcal{R}} \mu$.
2. Prove that both $\asymp$ and $\simeq$ are relaxed bisimulations.
3. Prove that if $\mathcal{R}$ is a *relaxed bisimulation*, $(s, s') \in \mathcal{R}$, G is closed under $\mathcal{R}$ and C_s is a non-terminating computation from s, there exists a non-terminating computation $C_{s'}$ from s', for which Conditions A, B and C hold.

Let us now realize this plan. Throughout this chapter, let $MA = (S, Act, \mapsto, \Rightarrow)$ be a semi-closed MA.

Definition 7.1 (Relaxed bisimulation relations). An equivalence relation $\mathcal{R}$ on S is a *relaxed bisimulation relation* on S if for every $(s, s') \in \mathcal{R}$, $\alpha \in Act^{\chi}$ and $\mu \in Dist(S)$, $s \xrightarrow{\alpha} \mu$ implies that there exists an α-computation $C_{s'}$ from s' such that $\mu_{C_{s'}} \equiv_{\mathcal{R}} \mu$.

Lemma 7.2 (Computation induced by a weak transition). *Suppose* $s \overset{\alpha}{\hookrightarrow}_{\oplus} \mu$ *for some* $s \in S$, $\alpha \in Act^{\chi}$ *and* $\mu \in Dist(S)$. *Then there exists an* α*-computation* C_s *from* s *with* $\mu_{C_s} = \mu$.

Proof. By Definition 2.32, there exists a finite indexed set $\{(c_i, \mu_i)\}_{i\in\{1,...,n\}}$ of pairs of positive real valued weights and distributions such that:

- $s \overset{\alpha}{\hookrightarrow} \mu_i$ for each $i \in \{1, ..., n\}$,
- $\sum_{i=1}^{n} c_i = 1$, and
- $\mu = \bigoplus_{i=1}^{n} (c_i \cdot \mu_i)$.

In other words, there exists a finite convex combination of transition trees from s such that its induced distribution is equal to μ. We now construct the corresponding α-computation as follows:

1. We start with a single node σ_0 labelled with s.
2. We append the transition trees not consisting of a single state s to σ_0:
    (a) the coefficients c_i become the probabilities of the transitions going out of σ_0,
    (b) the probabilities of all other transitions are 1.
3. Each tree consisting of a single node s becomes an ϵ-transition from σ_0, with the probability given by the corresponding c_i.

It is clear that the leaves of C_s are in a one-to-one correspondence with the leaves of the transition trees. It follows immediately that

$$\mu_{C_s} \quad = \quad \mu.$$

We now make sure that the resulting computation $C_s = (Q, \rightarrow, L, \sigma_0)$ is an α-computation (see Definition 6.13). Indeed:

1. $|\mu_{C_s}| = |\mu| = 1$,
2. the only node with outgoing ϵ-transitions is σ_0, and $|Reach_\epsilon(\sigma_0)| \leq n$, and
3. every leaf of C_s has the right trace (either α if $\alpha \neq \tau$ or ε otherwise) due to the analogous requirement on the transitions trees in Definition 2.32.

□

Lemma 7.3 (Naive weak bisimulation is relaxed bismulation relation). *The relation $\asymp$ is a relaxed bisimulation relation on S.*

Proof. The result follows immediately from Definition 2.33 and Lemma 7.2. □

Lemma 7.4 (Branching bisimulation is relaxed bismulation relation). *The relation $\simeq$ is a relaxed bisimulation relation on S.*

Proof. Suppose $s \simeq s'$ and $s \xrightarrow{\alpha} \mu$ for some $\alpha \in Act^{\chi}$ and $\mu \in Dist\left(S\right)$. Then, by Definitions 2.45 and 2.44, there exists some $D \in TETAS$ such that

$$\mu_{s'}^{D} \quad \equiv_{\simeq} \quad \mu$$

and

- either $\alpha = \tau$ and $D\left(s', \bot\right) = 1$,
- or for every time-abstract maximal path

  $$s' \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} ... \xrightarrow{\alpha_{n-1}} s_n \quad \in \quad MaxPaths_{abs}^{D}\left(s'\right)$$

  it holds that

  - $\alpha_{n-1} = \alpha$,
  - for all $i \in \{0, ..., n-2\}$, $\alpha_i = \tau$, and
  - for all $i \in \{1, ..., n-1\}$, $s' \simeq s_i$.

We now construct the corresponding α-computation as follows:

1. We start with a single node σ_0 labelled with s'.
2. We expand the computation by adding transitions as specified by the scheduler D. In the end, there is exactly one node for every $\pi \in Paths_{abs}^{*}$ with $\Pr_{s',D}\left(\pi\right) > 0$.
3. Whenever $D\left(\pi_{\sigma}, \bot\right) > 0$ for some node $\sigma \in Q$, we add a new node σ_{stop} and add an ϵ-transition from σ to σ_{stop} with the probability $D\left(\pi_{\sigma}, \bot\right)$.

It is not difficult to see that the conditions of Definitions 6.13 and 7.1 are satisfied. $\square$

Lemma 7.5 (Order of transitions in computation). *Let $C = \left(Q, \to, L, \sigma_0\right)$ be a computation. Then there exists a total order $\preceq$ on the set $\to$ of transitions of C such that for every $\sigma, \sigma' \in Q$,*

$$\sigma' \in Children^{*}\left(\sigma\right) \quad \textit{implies} \quad InTrans\left(\sigma\right) \preceq InTrans\left(\sigma'\right).$$

In other words, for every transition t from a node σ it holds that every transition on the path from σ_0 to σ occurs before t in $\preceq$.

Proof. Recall that, by Definition 6.2, C is a finitely-branching tree rooted in σ_0. Then let $\preceq$ be the order in which the transitions are discovered during breadth-first search on C. Obviously, $\preceq$ satisfies the requirement above. $\square$

Lemma 7.6 (Existence of weakly simulating computation for relaxed bisimulation). *Let $\mathcal{R}$ be a relaxed bisimulation relation on S. Suppose $s, s' \in S$ and $(s, s') \in \mathcal{R}$. Let $C = (Q, \rightarrow, L, \sigma_0)$ be a simple computation from s. Then there exists a computation $C' = (Q', \rightarrow', L', \gamma_0)$ from s' such that $C \preceq C'$. Moreover, the corresponding function w satisfies the following condition: for every $\sigma \in Q$ and $\gamma \in Q'$, $\gamma \in w(\sigma)$ implies $(L(\sigma), L'(\gamma)) \in \mathcal{R}$.*

Proof. Suppose $(s, s') \in \mathcal{R}$ and $C = (Q, \rightarrow, L, \sigma_0)$ is a simple computation from s. We construct C' and w inductively. In particular, we start with a single node labelled with s', and add transitions to simulate the transitions in C in any order satisfying the requirement of Lemma 7.5.

Base case. $C'_0 = (\{\gamma_0\}, \emptyset, L'_0, \gamma_0)$ with $L'_0(\gamma_0) = s'$, and $w_0(\sigma_0) = \{\gamma_0\}$.

Induction step. Given a computation $C'_n = (Q'_n, \rightarrow'_n, L'_n, \gamma_0)$ with the first n transitions of C simulated, and the corresponding w_n, we construct C'_{n+1} and w_{n+1} as follows. Suppose the next transition to be simulated is (σ, α, p, μ) (note that $p = 1$ since C is simple). Let $v = L(\sigma)$. Note that by Definition 6.10, C does not contain ϵ-transitions, and therefore $\alpha \in Act^{\chi}(s)$.

By the induction hypothesis, for every $\gamma \in w_n(\sigma)$ with $v' = L'_n(\gamma)$ it holds that $(v, v') \in \mathcal{R}$. Then, by Definition 7.1, there exists an α-computation C_γ from v' with $\mu_{C_\gamma} \equiv_{\mathcal{R}} \mu_{v,\alpha}$.

We now extend C'_n as follows (while reading the rest of the proof, the reader might find it helpful to refer to the example located right after this lemma). For every $\gamma \in w_n(\sigma)$:

1. Append C_γ to γ.
2. Then, the subdistribution induced by the newly added leaves is equivalent to $\Pr(\gamma) \cdot \mu_{v,\alpha}$ with respect to $\mathcal{R}$. Now, for every such leaf γ', whose associated state is v'':
   (a) let $EC = \{\sigma' \in Supp(\mu) \mid (L(\sigma'), v'') \in \mathcal{R}\}$ be those children of σ whose states belong to the same equivalence class under $\mathcal{R}$ as v'', and let $m = |EC|$,
   (b) arbitrarily assign the indices $1, ..., m$ to the nodes in EC (let us refer to the nodes as $\sigma_1, ..., \sigma_m$)
   (c) split γ' into m nodes $\{\gamma'_1, ..., \gamma'_m\}$ and replace the transition

$$InTrans(\gamma') = (Parent(\gamma'), \beta, p', \mu'')$$

with the transition

$$(Parent(\gamma'), \beta, p', \mu'''),$$

where

$$\mu''' = (\mu'' - \gamma') \oplus \bigoplus_{i=1}^{m} \left\{ \left(\gamma'_i, \mu''(\gamma') \cdot \frac{\mu(\sigma_i)}{\sum_{j=1}^{m} \mu(\sigma_j)} \right) \right\}.$$

(d) add each γ'_i to $w_{n+1}(\sigma_i)$.

Having constructed an infinite sequence of computations C'_n and functions w_n, we set

$$C' = \left(\bigcup_{n=0}^{\infty} Q'_n, \bigcup_{n=0}^{\infty} \rightarrow'_n, \bigcup_{n=0}^{\infty} L'_n, \gamma_0 \right)$$

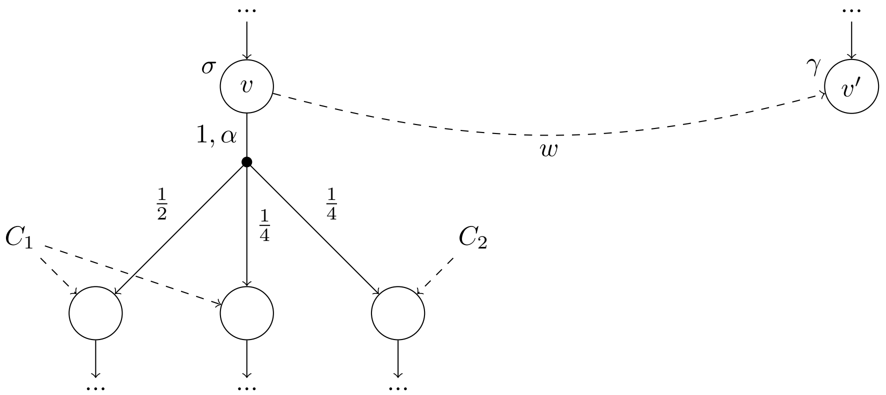
and

$$w = \bigcup_{n=0}^{\infty} w_n.$$

It is not difficult to see that the conditions in Definition 6.36 are satisfied. Furthermore, for every $\sigma \in Q$ and $\gamma \in Q'$, $\gamma \in w(\sigma)$ implies $(L(\sigma), L'(\gamma)) \in \mathcal{R}$.

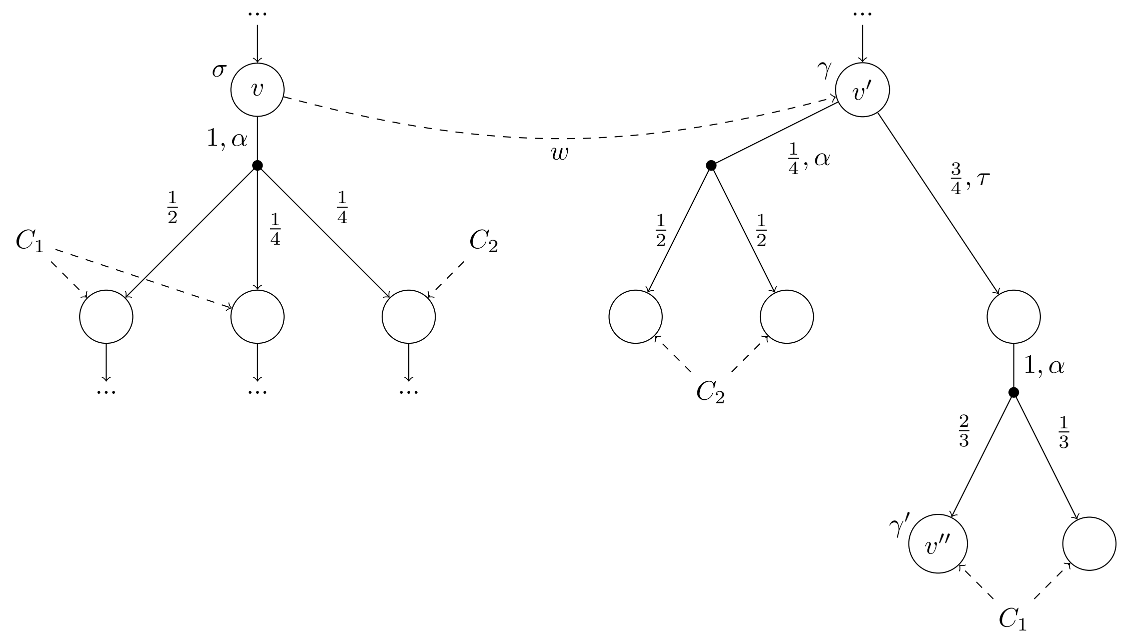
We also need to prove that C' is non-terminating (see Definition 6.6). It is clear that $\mathit{Leaves}(C') = \emptyset$. The second condition in Definition 6.6 is a bit trickier, and we will consider it separately in Lemma 7.7. ☐

Example. Suppose the next transition to be simulated is as follows (C_1 and C_2 are equivalence classes under $\mathcal{R}$, and $\gamma \in Q'_n$ is such that $\gamma \in w_n(\sigma)$).

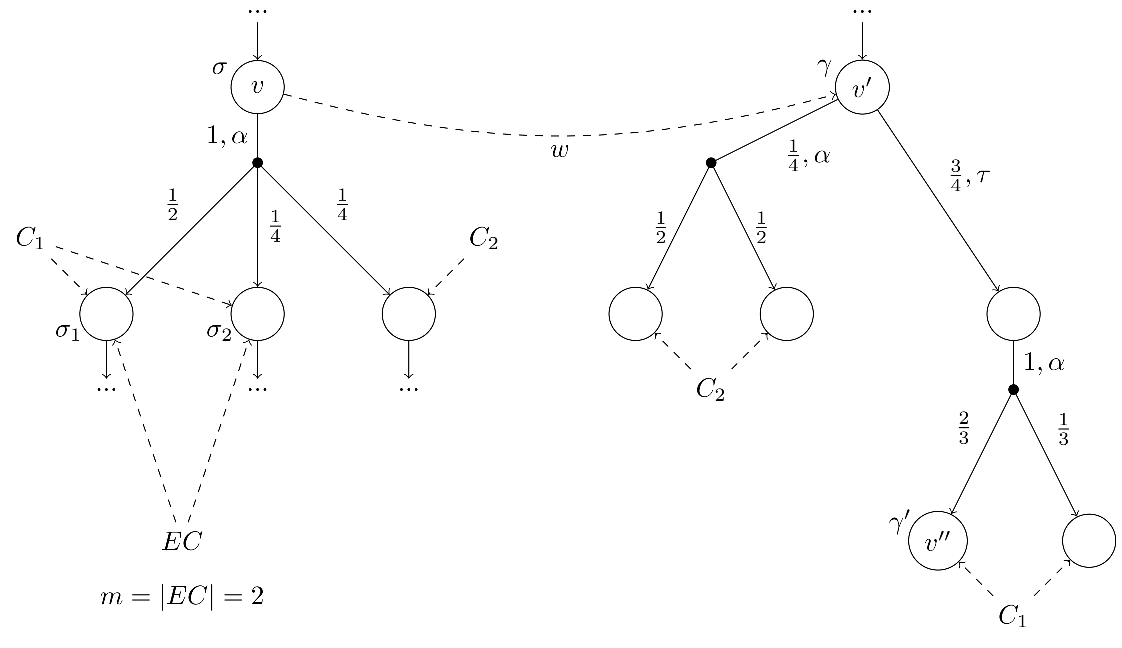


The algorithm then proceeds as follows.

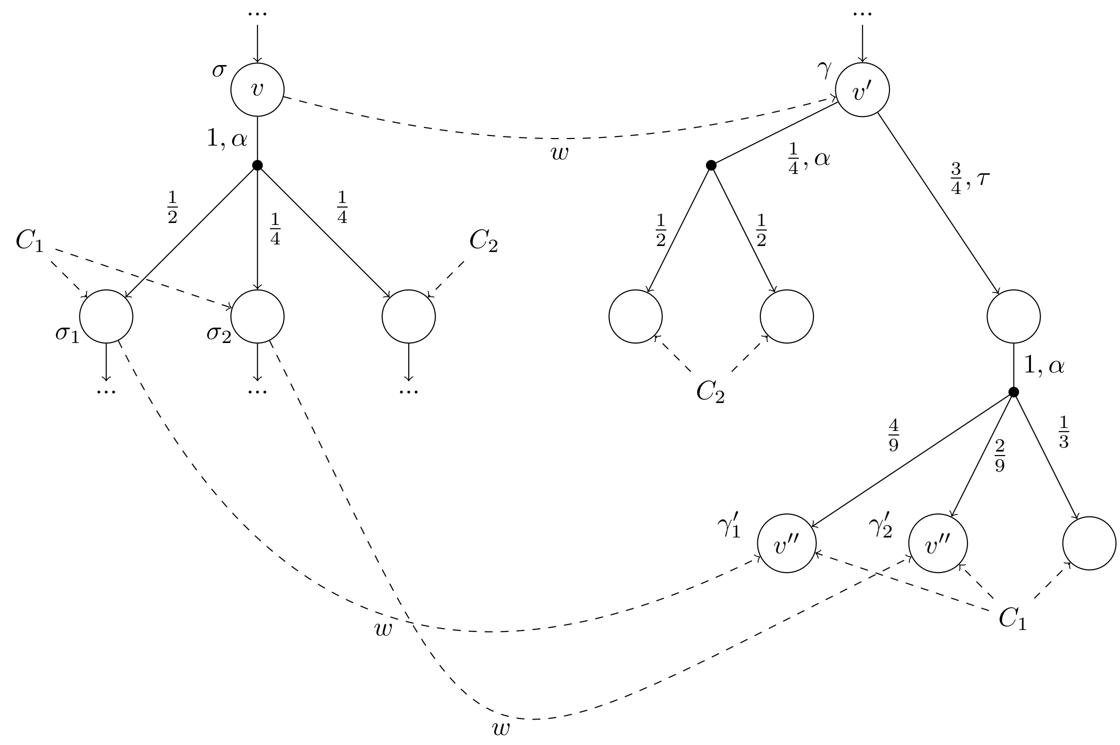
Step 1. The computation simulating the α-transition from v is appended to γ:



Step 2. For each newly added leaf, for example, the one denoted by γ' in the figure above, we determine the values of EC and m according to the equivalence class of the state associated with γ', and denote the nodes in EC by $\sigma_1, ..., \sigma_m$:



Step 3. Finally, γ' is split into $m = 2$ nodes corresponding to the nodes $\sigma_1, ..., \sigma_m$ (with the ingoing transition updated accordingly):



After every leaf has been processed in this way, the entire transition has been simulated.

Lemma 7.7 (Weakly simulating computation is non-terminating). *For every $\gamma \in Q'$ it holds that*

$$\sum_{\gamma' \in Reach_\epsilon(\gamma)} \Pr(\gamma') \cdot (1 - \Pr(\gamma, \epsilon)) \quad = \quad \Pr(\gamma).$$

Proof. First of all, for every $i \in \{0, ..., \infty\}$, let $Reach^i_\epsilon(\gamma)$ be the set of nodes reachable from γ by exactly i ϵ-transitions. Then, clearly,

$$Reach_\epsilon(\gamma) \quad = \quad \biguplus_{i=0}^{\infty} Reach^i_\epsilon(\gamma).$$

Let n be maximum i such that $Reach^i_\epsilon(\gamma)$ is non-empty, setting $n = \infty$ in the case when all the $Reach^i_\epsilon(\gamma)$ are non-empty. For each i, let

$$\Pr\left(Reach^i_\epsilon(\gamma)\right) \quad = \quad \sum_{\gamma' \in Reach^i_\epsilon(\gamma)} \Pr(\gamma').$$

Now,

$$\begin{aligned}\sum_{\gamma' \in Reach_{\epsilon}(\gamma)} \Pr(\gamma') \cdot (1 - \Pr(\gamma, \epsilon)) &= \sum_{i=0}^{n} \sum_{\gamma' \in Reach_{\epsilon}^{i}(\gamma)} \Pr(\gamma') \cdot (1 - \Pr(\gamma', \epsilon)) \\ &= \sum_{i=0}^{n} \left(\Pr\left(Reach_{\epsilon}^{i}(\gamma)\right) - \Pr\left(Reach_{\epsilon}^{i+1}(\gamma)\right)\right).\end{aligned}$$

Now, if n is finite, then $\Pr\left(Reach_{\epsilon}^{n+1}(\gamma)\right) = 0$, so that the sum is clearly $\Pr\left(Reach_{\epsilon}^{0}(\gamma)\right) = \Pr(\gamma)$, as desired.

If, on the other hand, $n = \infty$, then

$$\sum_{\gamma' \in Reach_{\epsilon}(\gamma)} \Pr(\gamma') \cdot (1 - \Pr(\gamma, \epsilon)) = \sum_{i=0}^{\infty} \left(\Pr\left(Reach_{\epsilon}^{i}(\gamma)\right) - \Pr\left(Reach_{\epsilon}^{i+1}(\gamma)\right)\right).$$

We now show that this infinite series converges to $\Pr(\gamma)$. Consider the partial sum of the first $n+1$ terms:

$$\begin{aligned}\sum_{i=0}^{n} \left(\Pr\left(Reach_{\epsilon}^{i}(\gamma)\right) - \Pr\left(Reach_{\epsilon}^{i+1}(\gamma)\right)\right) &= \Pr\left(Reach_{\epsilon}^{0}(\gamma)\right) - \Pr\left(Reach_{\epsilon}^{n+1}(\gamma)\right) \\ &= \Pr(\gamma) - \Pr\left(Reach_{\epsilon}^{n+1}(\gamma)\right).\end{aligned}$$

Thus, we only need to show that

$$\lim_{i \to \infty} \Pr\left(Reach_{\epsilon}^{i}(\gamma)\right) = 0.$$

Suppose it is not so. Then there is a non-zero probability to execute only ϵ-transitions starting from γ. But then there is a non-zero probability to execute only τ-transitions in C, which implies that the Markov automaton exhibits Zeno behaviour. But this contradicts our general assumption that the automaton is non-Zeno. The result follows. □

Lemma 7.8. *Let $\mathcal{R}$ be a relaxed bisimulation relation on S. Suppose $s \in S$ does not have outgoing τ-transitions and $s' \in S$ is such that $(s, s') \in \mathcal{R}$. Then for every state t reachable from s' via zero or more τ-transitions it holds that $(s, t) \in \mathcal{R}$.*

Proof. The proof is by induction on the number of τ-transitions from s' to t.

Base case. In the case when $s' = t$ the claim holds by assumption.

Induction step. Suppose $v \in S$ is reachable from s' via n τ-transitions and $v \overset{\tau}{\mapsto} \mu'$ with $t \in Supp(\mu')$. By the induction hypothesis, $(s, v) \in \mathcal{R}$. Then, by Definition 7.1, there exists a τ-computation C_s from s such that $\mu_{C_s} \equiv_{\mathcal{R}} \mu'$. But s has no outgoing τ-transitions by assumption. Therefore, $\mu_{C_s} = \{(s, 1)\}$. It follows immediately that $(s, t) \in \mathcal{R}$, as desired. □

Lemma 7.9 (Preservation of minimum expected time under relaxed bisimulation). *Let $\mathcal{R}$ be a relaxed bisimulation relation on S. Let $s, s' \in S$. Suppose $G \subseteq S$ is the union of zero or more equivalence classes under $\mathcal{R}$. Then*

$$(s, s') \in \mathcal{R} \quad \textit{implies} \quad eT^{min}(s, \Diamond G) = eT^{min}(s', \Diamond G).$$

Proof. By Lemmas 7.6 and 6.41,

$$eT^{min}(s, \Diamond G) \quad \geq \quad eT^{min}(s', \Diamond G)$$

(note that the requirement of Lemma 6.41 is satisfied because G is closed under $\mathcal{R}$).

By symmetry, it also holds that $eT^{min}(s, \Diamond G) \leq eT^{min}(s', \Diamond G)$, so that, in fact,

$$eT^{min}(s, \Diamond G) \quad = \quad eT^{min}(s', \Diamond G),$$

as desired. □

Lemma 7.10 (Preservation of minimum long run average under relaxed bisimulation). *Let $\mathcal{R}$ be a relaxed bisimulation relation on S. Let $s, s' \in S$. Suppose $G \subseteq S$ is the union of zero or more equivalence classes under $\mathcal{R}$. Then*

$$(s, s') \in \mathcal{R} \quad \textit{implies} \quad LRA^{min}(s, \Diamond G) = LRA^{min}(s', \Diamond G).$$

Proof. By Lemma 6.43 and the symmetry of $\mathcal{R}$, we only need to prove that for every simple computation $C_s = (Q, \rightarrow, L, \sigma_0)$ from s there exists a weakly simulating computation $C_{s'} = (Q', \rightarrow', L', \gamma_0)$ from s' such that the corresponding function w satisfies the following requirement: for every $\sigma \in Q$ with $L(\sigma) \in MS$, every $\gamma \in w(\sigma)$ and every $\gamma' \in Children^*(\gamma)$ with $L'(\gamma') \in MS$ and $Tr(\gamma) = Tr(\gamma')$ (i.e. for every γ' with $L'(\gamma') \in MS$ reachable from γ via zero or more τ- and ϵ-transitions) it holds that

$$L(\sigma) \in G \quad \text{if and only if} \quad L'(\gamma') \in G.$$

To this end, suppose $C_s = (Q, \rightarrow, L, \sigma_0)$ is a simple computation from s. Then, by Lemma 7.6, there exists a weakly simulating computation $C_{s'} = (Q', \rightarrow', L', \gamma_0)$ from s' such that for every $\sigma \in Q$ and $\gamma \in w(\sigma)$ it holds that $(L(\sigma), L'(\gamma)) \in \mathcal{R}$.

Now, let $\sigma \in Q$ with $L(\sigma) \in MS$ (note that it implies that $L(\sigma)$ has no outgoing τ-transitions), $\gamma \in w(\sigma)$, and $\gamma' \in Children^*(\gamma)$ with $L'(\gamma') \in MS$ and $Tr(\gamma) = Tr(\gamma')$.

Since $\gamma \in w(\sigma)$, $(L(\sigma), L'(\gamma)) \in \mathcal{R}$. Furthermore, $L'(\gamma')$ must be reachable from $L'(\gamma)$ via zero or more τ-transitions (because γ' is reachable from γ via zero or more τ- or ϵ-transitions). Then, by Lemma 7.8, $(L(\sigma), L'(\gamma')) \in \mathcal{R}$.

Now, since G is closed under $\mathcal{R}$, it follows that

$$L(\sigma) \in G \quad \text{if and only if} \quad L'(\gamma') \in G,$$

as desired. $\square$

Theorem 7.11 (Preservation of minimum expected time under naive weak bisimulation).
Let $s, s' \in S$. Suppose $G \subseteq S$ is the union of zero or more equivalence classes under $\asymp$. Then,

$$s \asymp s' \quad \textit{implies} \quad eT^{min}(s, \Diamond G) = eT^{min}(s', \Diamond G).$$

Theorem 7.12 (Preservation of minimum expected time under branching bisimulation).
Let $s, s' \in S$. Suppose $G \subseteq S$ is the union of zero or more equivalence classes under $\simeq$. Then,

$$s \simeq s' \quad \textit{implies} \quad eT^{min}(s, \Diamond G) = eT^{min}(s', \Diamond G).$$

Theorem 7.13 (Preservation of minimum long run average under naive weak bisimulation).
Let $s, s' \in S$. Suppose $G \subseteq S$ is the union of zero or more equivalence classes under $\asymp$. Then,

$$s \asymp s' \quad \textit{implies} \quad LRA^{min}(s, \Diamond G) = LRA^{min}(s', \Diamond G).$$

Theorem 7.14 (Preservation of minimum long run average under branching bisimulation).
Let $s, s' \in S$. Suppose $G \subseteq S$ is the union of zero or more equivalence classes under $\simeq$. Then,

$$s \simeq s' \quad \textit{implies} \quad LRA^{min}(s, \Diamond G) = LRA^{min}(s', \Diamond G).$$

8 Property preservation under weak bisimulation

Now, we would like to reproduce our proof from the previous chapter for the case of weak bisimulation. Unfortunately, it turns out that, in general, weak bisimulation does *not* preserve minimum expected time. Let us consider a counterexample first, and then try to understand what goes wrong.

Theorem 8.1 (Weak bisimulation does not preserve minimum expected time). *There exist a semi-closed Markov automaton $MA = (S, Act, \mapsto, \Rightarrow)$, a set $G \subseteq S$ closed under $\approx$, and a pair of states $s, t \in S$ such that $s \approx t$ and $eT^{min}(s, \Diamond G) \neq eT^{min}(t, \Diamond G)$.*

Proof. Consider the Markov automaton MA in Figure 19. Arrows with single arrowheads denote immediate transitions and arrows with double arrowheads denote timed transitions.

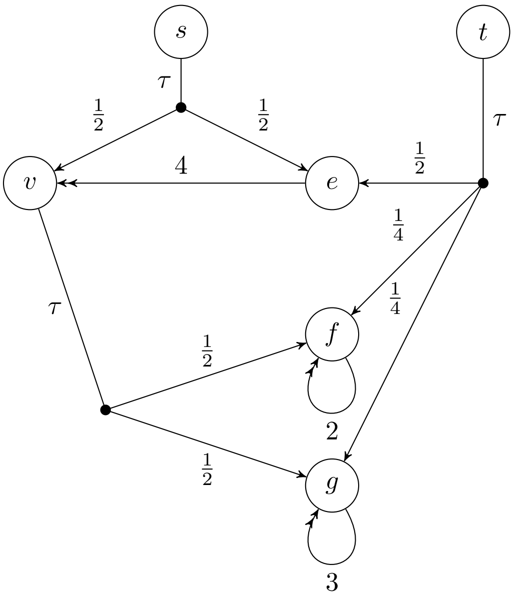


Figure 19: $s \approx t$, but $eT^{min}\left(s, \Diamond [v]_{\approx}\right) \neq eT^{min}\left(t, \Diamond [v]_{\approx}\right)$.

We first show that the states e, f and g belong to three different equivalence classes under $\approx$. Indeed, if e and f were weakly bisimilar, f would be able to execute some τ-transitions and reach a state with an outgoing timed transition with rate 4 (in order to match the corresponding transition from e), which is not the case. The same reasoning applies to the other two pairs of states.

Next, we claim that $s \approx t$. We prove it by showing that the following relation $\mathcal{R}$:

$$\begin{array}{rcl}
\left\{\left(v, \frac{1}{2}\right)\right\} & \mathcal{R} & \left\{\left(f, \frac{1}{4}\right), \left(g, \frac{1}{4}\right)\right\} \\
\left\{\left(e, \frac{1}{2}\right), \left(v, \frac{1}{2}\right)\right\} & \mathcal{R} & \left\{\left(e, \frac{1}{2}\right), \left(f, \frac{1}{4}\right), \left(g, \frac{1}{4}\right)\right\} \\
\{(s, 1)\} & \mathcal{R} & \{(t, 1)\} \\
\mu & \mathcal{R} & \mu \qquad \forall \mu \in \mathit{Subdist}\,(S)
\end{array}$$

is a weak bisimulation relation.

First, we will prove that the pair of distributions $\mu_1 = \left\{\left(v, \frac{1}{2}\right)\right\}$ and $\mu_2 = \left\{\left(f, \frac{1}{4}\right), \left(g, \frac{1}{4}\right)\right\}$ satisfies the requirements of a weak bisimulation relation:

1. $|\mu_1| = \frac{1}{2} = |\mu_2|$.
2. For $v \in \mathit{Supp}\,(\mu_1)$. Let $\mu_2^s = \mu_2$ and $\mu_2^{rest} = \emptyset$. Then:
   (a) $\mu_2 \overset{\tau}{\hookrightarrow}_{\oplus} (\mu_2^s \oplus \mu_2^{rest})$,
   (b) $\{(v, \mu_1(v))\} = \mu_1 \,\mathcal{R}\, \mu_2 = \mu_2^s$ and $(\mu_1 - v) = \emptyset \,\mathcal{R}\, \emptyset = \mu_2^{rest}$,
   (c) the transition
   $$v \xmapsto{\tau} \left\{\left(f, \frac{1}{2}\right), \left(g, \frac{1}{2}\right)\right\}$$
   can be matched by
   $$\mu_2^s \overset{\tau}{\hookrightarrow}_{\oplus} \mu_2^s$$
   (by doing nothing) since
   $$\left(\mu_1(v) \cdot \left\{\left(f, \frac{1}{2}\right), \left(g, \frac{1}{2}\right)\right\}\right) = \mu_2 \quad \mathcal{R} \quad \mu_2 = \mu_2^s.$$
3. For $f \in \mathit{Supp}\,(\mu_2)$. Let $\mu_1^s = \left\{\left(f, \frac{1}{4}\right)\right\}$ and $\mu_1^{rest} = \left\{\left(g, \frac{1}{4}\right)\right\}$. Then:
   (a) $\mu_1 \overset{\tau}{\hookrightarrow}_{\oplus} (\mu_1^s \oplus \mu_1^{rest})$,
   (b) $\mu_1^s = \left\{\left(f, \frac{1}{4}\right)\right\} \,\mathcal{R}\, \left\{\left(f, \frac{1}{4}\right)\right\} = \{(f, \mu_2(f))\}$ and $\mu_1^{rest} = \left\{\left(g, \frac{1}{4}\right)\right\} \,\mathcal{R}\, \left\{\left(g, \frac{1}{4}\right)\right\} = (\mu_2 - f)$,
   (c) Since $\mu_1^s = \left\{\left(f, \frac{1}{4}\right)\right\} = \{(f, \mu_2(f))\}$, clearly every transition from f can be matched by the same transition from μ_1^s.
4. For $g \in \mathit{Supp}\,(\mu_2)$ - completely analogous.

Now, we show that the pair of distributions $\mu_1 = \left\{\left(e, \frac{1}{2}\right), \left(v, \frac{1}{2}\right)\right\}$ and $\mu_2 = \left\{\left(e, \frac{1}{2}\right), \left(f, \frac{1}{4}\right), \left(g, \frac{1}{4}\right)\right\}$ satisfies the requirements of a weak bisimulation relation:

1. $|\mu_1| = 1 = |\mu_2|$.

2. For $e \in Supp\,(\mu_1)$. Let $\mu_2^s = \left\{\left(e, \frac{1}{2}\right)\right\}$ and $\mu_2^{rest} = \left\{\left(f, \frac{1}{4}\right), \left(g, \frac{1}{4}\right)\right\}$. Then:
    (a) $\mu_2 \overset{\tau}{\hookrightarrow}_{\oplus} (\mu_2^s \oplus \mu_2^{rest})$,
    (b) $\{(e, \mu_1(e))\} = \left\{\left(e, \frac{1}{2}\right)\right\} \ \mathcal{R}\ \left\{\left(e, \frac{1}{2}\right)\right\} = \mu_2^s$ and $(\mu_1 - e) = \left\{\left(v, \frac{1}{2}\right)\right\} \ \mathcal{R}\ \left\{\left(f, \frac{1}{4}\right), \left(g, \frac{1}{4}\right)\right\} = \mu_2^{rest}$, and
    (c) since $\{(e, \mu_1(e))\} = \left\{\left(e, \frac{1}{2}\right)\right\} = \mu_2^s$, every transition from e can be matched by the same transition from μ_2^s.

3. For $v \in Supp\,(\mu_1)$. Let $\mu_2^s = \left\{\left(f, \frac{1}{4}\right), \left(g, \frac{1}{4}\right)\right\}$ and $\mu_2^{rest} = \left\{\left(e, \frac{1}{2}\right)\right\}$. Then:
    (a) $\mu_2 \overset{\tau}{\hookrightarrow}_{\oplus} (\mu_2^s \oplus \mu_2^{rest})$,
    (b) $\{(v, \mu_1(v))\} = \left\{\left(v, \frac{1}{2}\right)\right\} \ \mathcal{R}\ \left\{\left(f, \frac{1}{4}\right), \left(g, \frac{1}{4}\right)\right\} = \mu_2^s$ and $(\mu_1 - v) = \left\{\left(e, \frac{1}{2}\right)\right\} \ \mathcal{R}\ \left\{\left(e, \frac{1}{2}\right)\right\} = \mu_2^{rest}$,
    (c) the transition

$$v \ \overset{\tau}{\longmapsto}\ \left\{\left(f, \frac{1}{2}\right), \left(g, \frac{1}{2}\right)\right\}$$

    can be matched by

$$\mu_2^s \ \overset{\tau}{\hookrightarrow}_{\oplus}\ \mu_2^s$$

    (by doing nothing) since

$$\left(\mu_1(v) \cdot \left\{\left(f, \frac{1}{2}\right), \left(g, \frac{1}{2}\right)\right\}\right) = \mu_2^s \quad \mathcal{R} \quad \mu_2^s.$$

4. For $e \in Supp\,(\mu_2)$. Let $\mu_1^s = \left\{\left(e, \frac{1}{2}\right)\right\}$ and $\mu_1^{rest} = \left\{\left(v, \frac{1}{2}\right)\right\}$. Then:
    (a) $\mu_1 \overset{\tau}{\hookrightarrow}_{\oplus} (\mu_1^s \oplus \mu_1^{rest})$,
    (b) $\mu_1^s = \left\{\left(e, \frac{1}{2}\right)\right\} \ \mathcal{R}\ \left\{\left(e, \frac{1}{2}\right)\right\} = \{(e, \mu_2(e))\}$ and $\mu_1^{rest} = \left\{\left(v, \frac{1}{2}\right)\right\} \ \mathcal{R}\ \left\{\left(f, \frac{1}{4}\right), \left(g, \frac{1}{4}\right)\right\} = (\mu_2 - e)$, and
    (c) since $\mu_1^s = \left\{\left(e, \frac{1}{2}\right)\right\} = \{(e, \mu_2(e))\}$, every transition from e can be matched by the same transition from μ_1^s.

5. For $f \in Supp\,(\mu_2)$. Let $\mu_1^s = \left\{\left(f, \frac{1}{4}\right)\right\}$ and $\mu_1^{rest} = \left\{\left(e, \frac{1}{2}\right), \left(g, \frac{1}{4}\right)\right\}$. Then:
    (a) $\mu_1 \overset{\tau}{\hookrightarrow}_{\oplus} (\mu_1^s \oplus \mu_1^{rest})$,
    (b) $\mu_1^s = \left\{\left(f, \frac{1}{4}\right)\right\} \ \mathcal{R}\ \left\{\left(f, \frac{1}{4}\right)\right\} = \{(f, \mu_2(f))\}$ and $\mu_1^{rest} = \left\{\left(e, \frac{1}{2}\right), \left(g, \frac{1}{4}\right)\right\} \ \mathcal{R}\ \left\{\left(e, \frac{1}{2}\right), \left(g, \frac{1}{4}\right)\right\} = (\mu_2 - f)$, and
    (c) since $\mu_1^s = \left\{\left(f, \frac{1}{4}\right)\right\} = \{(f, \mu_2(f))\}$, every transition from f can be matched by the same transition from μ_1^s.

6. For $g \in \mathit{Supp}\,(\mu_2)$ - completely analogous.

Next, we show that the pair of distributions $\mu_1 = \{(s,1)\}$ and $\mu_2 = \{(t,1)\}$ satisfies the requirements of a weak bisimulation relation:

1. $|\mu_1| = 1 = |\mu_2|$.
2. For $s \in \mathit{Supp}\,(\mu_1)$. Let $\mu_2^s = \mu_2$ and $\mu_2^{rest} = \emptyset$. Then:
    (a) $\mu_2 \overset{\tau}{\hookrightarrow}_{\oplus} (\mu_2^s \oplus \mu_2^{rest})$,
    (b) $\{(s, \mu_1(s))\} = \mu_1 \mathcal{R} \mu_2 = \mu_2^s$ and $(\mu_1 - v) = \emptyset \mathcal{R} \emptyset = \mu_2^{rest}$, and
    (c) the transition
    $$s \overset{\tau}{\longmapsto} \left\{\left(v, \frac{1}{2}\right), \left(e, \frac{1}{2}\right)\right\}$$
    can be matched by
    $$\mu_2^s \overset{\tau}{\hookrightarrow}_{\oplus} \left\{\left(e, \frac{1}{2}\right), \left(f, \frac{1}{4}\right), \left(g, \frac{1}{4}\right)\right\}$$
    since
    $$\left(\mu_1(s) \cdot \left\{\left(v, \frac{1}{2}\right), \left(e, \frac{1}{2}\right)\right\}\right) = \left\{\left(v, \frac{1}{2}\right), \left(e, \frac{1}{2}\right)\right\} \quad \mathcal{R} \quad \left\{\left(e, \frac{1}{2}\right), \left(f, \frac{1}{4}\right), \left(g, \frac{1}{4}\right)\right\}.$$
3. For $t \in \mathit{Supp}\,(\mu_2)$. Let $\mu_1^s = \mu_1$ and $\mu_1^{rest} = \emptyset$. Then:
    (a) $\mu_1 \overset{\tau}{\hookrightarrow}_{\oplus} (\mu_1^s \oplus \mu_1^{rest})$,
    (b) $\mu_1^s = \mu_1 \mathcal{R} \mu_2 = \{(t, \mu_2(t))\}$ and $\mu_1^{rest} = \emptyset \mathcal{R} \emptyset = (\mu_2 - t)$, and
    (c) the transition
    $$t \overset{\tau}{\longmapsto} \left\{\left(e, \frac{1}{2}\right), \left(f, \frac{1}{4}\right), \left(g, \frac{1}{4}\right)\right\}$$
    can be matched by
    $$\mu_1^s \overset{\tau}{\hookrightarrow}_{\oplus} \left\{\left(v, \frac{1}{2}\right), \left(e, \frac{1}{2}\right)\right\}$$
    since
    $$\begin{aligned} \left\{\left(v, \frac{1}{2}\right), \left(e, \frac{1}{2}\right)\right\} \quad \mathcal{R} & \quad \left\{\left(e, \frac{1}{2}\right), \left(f, \frac{1}{4}\right), \left(g, \frac{1}{4}\right)\right\} \\ = & \quad \left(\mu_2(t) \cdot \left\{\left(e, \frac{1}{2}\right), \left(f, \frac{1}{4}\right), \left(g, \frac{1}{4}\right)\right\}\right). \end{aligned}$$

Thus, $s \approx t$.

Next, we will prove that $[v]_{\approx} = \{v\}$. Indeed, $v \not\approx f$ and $v \not\approx g$, because neither f nor g can reach a distribution weakly bisimilar to $\left\{\left(f, \frac{1}{2}\right), \left(g, \frac{1}{2}\right)\right\}$. Furthermore, $v \not\approx e$ since v cannot reach a state with an outgoing timed transition with rate 4. Finally, $v \not\approx s$ and $v \not\approx t$ because both s and t have transitions leading to distributions involving e, while v cannot match such a transition.

Finally, observe that every path starting in s must pass through v, so that $eT^{min}\left(s, \Diamond [v]_{\approx}\right)$ is finite. By contrast, $eT^{min}\left(t, \Diamond [v]_{\approx}\right) = \infty$ since there is a non-zero probability of never reaching v starting from t. The result follows. □

This negative result may come as a surprise: after all, it is not difficult to see that in the case of weak bisimulation, a transition $s \xrightarrow{\alpha} \mu$ can be simulated by an α-computation C from the bisimilar state s'. So what goes wrong? The answer is that, by contrast with relaxed bisimulations, the resulting distribution μ_C *does not have to be equivalent to* μ *with respect to the probabilities they assign to the equivalence classes under* $\approx$. In other words, in general it is *not* the case that $\mu_C \equiv_{\approx} \mu$. Instead, it is only required that $\mu_C \approx \mu$. Indeed, let us consider the MA in Figure 19 again. The transition $s \xrightarrow{\tau} \{(v, 0.5), (e, 0.5)\}$ is matched by $t \xrightarrow{\tau} \{(e, 0.5), (f, 0.25), (g, 0.25)\}$. Now, it is true that $\{(v, 0.5), (e, 0.5)\} \approx \{(e, 0.5), (f, 0.25), (g, 0.25)\}$. However, these distributions do not assign the same probabilities to the equivalence classes under $\approx$, in particular to $[v]_{\approx}$. It is not difficult to see that this is indeed what causes the problem.

In spite of this negative result, we will prove that weak bisimulation *does* preserve minimum expected time under certain conditions. In particular, under the condition that every equivalence class under $\approx$ in the target set G contains at least one state with no outgoing τ-transitions. This result is by no means obvious. Let us first outline the idea informally.

First, it has already been mentioned that if $s \approx s'$, a transition $s \xrightarrow{\alpha} \mu$ can be simulated by an α-computation C from s' such that $\mu \approx \mu_C$. We now claim that we can do even slightly better. Suppose $Supp(\mu) = \{s_1, ..., s_n\}$. Now, recall that from the definition of weak bisimulation it follows that μ_C can execute some τ-transitions and reach a distribution μ' that can be split into n subdistributions μ'_i such that for each $i \in \{1, ..., n\}$, $s_i \approx \mu'_i$. We can then attach the τ-transitions to the leaves of C and obtain a computation C' whose leaves can be partitioned into n subsets Φ_i with $s_i \approx \mu_{\Phi_i}$ for each i.

In other words, given a transition $s \xrightarrow{\alpha} \mu$, there is a computation C' from s' whose leaves can be partitioned into subsets corresponding to each state in $Supp(\mu)$.

Now, a moment's thought is enough to realize that, given a non-terminating computation C_s from s, we can construct a weakly simulating computation $C_{s'}$ from s' such that for each node σ of C_s it holds that $\mu_\sigma \approx \mu_{w(\sigma)}$. Of course, in general, this is not enough to prove that $eT^{C_s}(s, \Diamond G) \geq eT^{C_{s'}}(s', \Diamond G)$. In particular, we would like to assert that for every σ corresponding to a state in G and every $\gamma \in w(\sigma)$ it holds that $L(\sigma) \approx L'(\gamma)$, but the latter is not implied by $\mu_\sigma \approx \mu_{w(\sigma)}$.

Suppose, however, that every equivalence class under $\approx$ in G contains a state with no outgoing τ-transitions. It turns out that in this case $\mu_\sigma \approx \mu_{w(\sigma)}$ does imply $L(\sigma) \approx L'(\gamma)$ whenever $L(\sigma) \in G$

and $\gamma \in w(\sigma)$! To see why, consider the following lemma.

For the rest of this chapter, let $MA = (S, Act, \mapsto, \Rightarrow)$ be a semi-closed MA.

Lemma 8.2 (Weak bisimulation between state without τ-transitions and distribution). *Suppose $s \in S$ has no outgoing τ-transitions and $\mu_2 \in Dist(S)$ such that $s \approx \mu_2$. Then $s \approx s'$ for every $s' \in Supp(\mu_2)$.*

Proof. Suppose $s' \in Supp(\mu_2)$. According to Definition 2.35, there exist μ_1^s, μ_1^{rest} such that:

1. $s \overset{\tau}{\hookrightarrow}_{\oplus} (\mu_1^s \oplus \mu_1^{rest})$, and
2. $\mu_1^s \approx \{(s', \mu_2(s'))\}$ (note that it follows that $|\mu_1^s| = \mu_2(s')$).

But since s has no outgoing τ-transitions, the only choice for μ_1^s is $\mu_1^s = \{(s, \mu_2(s'))\}$. Therefore,

$$\{(s, \mu_2(s'))\} \quad \approx \quad \{(s', \mu_2(s'))\}.$$

Then, clearly,

$$s \quad \approx \quad s',$$

as desired. □

At this point, it becomes quite clear that we have a proof. For every node σ with $L(\sigma) \in G$, $L(\sigma) \approx \frac{1}{\Pr(\sigma)} \cdot \mu_{w(\sigma)} \approx v$ for some state v with no outgoing τ-transitions. But then, for every $\gamma \in w(\sigma)$, $L'(\gamma) \approx v$, so that $L'(\gamma) \in G$, and by Lemma 6.41, $eT^{min}(s, \Diamond G) \geq eT^{min}(s', \Diamond G)$. This argument is formalized later in this chapter.

By a similar argument we can show that weak bisimulation preserves minimum long run average, even in the general case.

We now turn to formal proofs.

Lemma 8.3. *Suppose $s \in S$ has no outgoing τ-transitions and $s' \in S$ is such that $s \approx s'$. Then for every state t reachable from s' via zero or more τ-transitions it holds that $s \approx t$.*

Proof. The proof is by induction on the number of τ-transitions from s' to t.

Base case. In the case when $s' = t$ the claim holds by assumption.

Induction step. Suppose $v \in S$ is reachable from s' via n τ-transitions and $v \overset{\tau}{\mapsto} \mu'$ with $t \in Supp(\mu')$. By the induction hypothesis, $s \approx v$. Then, by the definition of weak bisimulation, there exist μ_1^s, μ_1^{rest} such that:

1. $s \overset{\tau}{\hookrightarrow}_{\oplus} (\mu_1^s \oplus \mu_1^{rest})$,

2. $\mu_1^s \approx \{(v,1)\}$ (note that it follows that $|\mu_1^s| = 1$), and

3. $\mu_1^s \overset{\tau}{\hookrightarrow}_{\oplus} \mu$ with $\mu \approx \mu'$.

But since s has no outgoing τ-transitions, the only choice for μ_1^s is $\mu_1^s = \{(s,1)\}$. Thus it must be the case that $s \overset{\tau}{\hookrightarrow}_{\oplus} \mu$ with $\mu \approx \mu'$. But then it again follows that $\mu = \{(s,1)\}$, so that $s \approx \mu'$. Then, by Lemma 8.2, $s \approx t$. ☐

When reading the statement and proof of the following lemma, the reader might find it useful to take a look at Figure 20.

Lemma 8.4. *Suppose $\mu, \mu' \in Subdist\,(S)$ and $\mu \approx \mu'$. Let $m = |Supp\,(\mu)|$ (assume $m > 0$). Then, for every $s' \in Supp\,(\mu')$, there exists a τ-computation $C_{s'}$ from s' and a partitioning of $Leaves\,(C_{s'})$ into subsets corresponding to the states in $Supp\,(\mu)$:*

$$Leaves\,(C_{s'}) \quad = \quad \biguplus_{s \in Supp(\mu)} \Phi_{s,s'},$$

such that for every $s \in Supp\,(\mu)$,

$$\{(s, \mu\,(s))\} \quad \approx \quad \mu'_s,$$

where

$$\mu'_s \quad = \quad \bigoplus_{s' \in Supp(\mu')} \left(\mu'\,(s') \cdot \mu_{\Phi_{s,s'}}\right).$$

Furthermore, for every $s \in Supp\,(\mu)$, $\alpha \in Act^{\chi}$ and $\mu'' \in Dist\,(S)$, $s \xrightarrow{\alpha} \mu''$ implies $\mu'_s \overset{\alpha}{\hookrightarrow}_{\oplus} \mu'''$ for some $\mu''' \in Subdist\,(S)$ with $\mu\,(s) \cdot \mu'' \approx \mu'''$.

Proof. Let $s \in Supp\,(\mu)$. By Definition 2.35, $\mu' \overset{\tau}{\hookrightarrow}_{\oplus} (\mu^s \oplus \mu^{rest})$ for some $\mu^s, \mu^{rest} \in Subdist\,(S)$ with $\{(s, \mu\,(s))\} \approx \mu^s$ and $(\mu - s) \approx \mu^{rest}$. Then, Lemma 7.2 says that for every $s' \in Supp\,(\mu')$ there exists a τ-computation $C_{s',s}$ and a partitioning

$$Leaves\,(C_{s',s}) \quad = \quad \Phi_{s,s'} \uplus \Phi_{rest,s'}$$

with

$$\bigoplus_{s' \in Supp(\mu')} \mu'\,(s') \cdot \mu_{\Phi_{s,s'}} \quad \approx \quad \{(s, \mu\,(s))\}$$

and

$$\bigoplus_{s' \in Supp(\mu')} \mu'(s') \cdot \mu_{\Phi_{rest,s'}} \quad \approx \quad \mu - s.$$

We can now apply the procedure again to $\bigoplus_{s' \in Supp(\mu')} \mu'(s') \cdot \mu_{\Phi_{rest,s'}}$ and $\mu - s$ and append the corresponding computations to the leaves in $\Phi_{rest,s'}$. Applying the procedure $|Supp(\mu)|$ times, we obtain the computations $C_{s'}$ and the sets $\Phi_{s,s'}$ satisfying the requirements above. $\square$

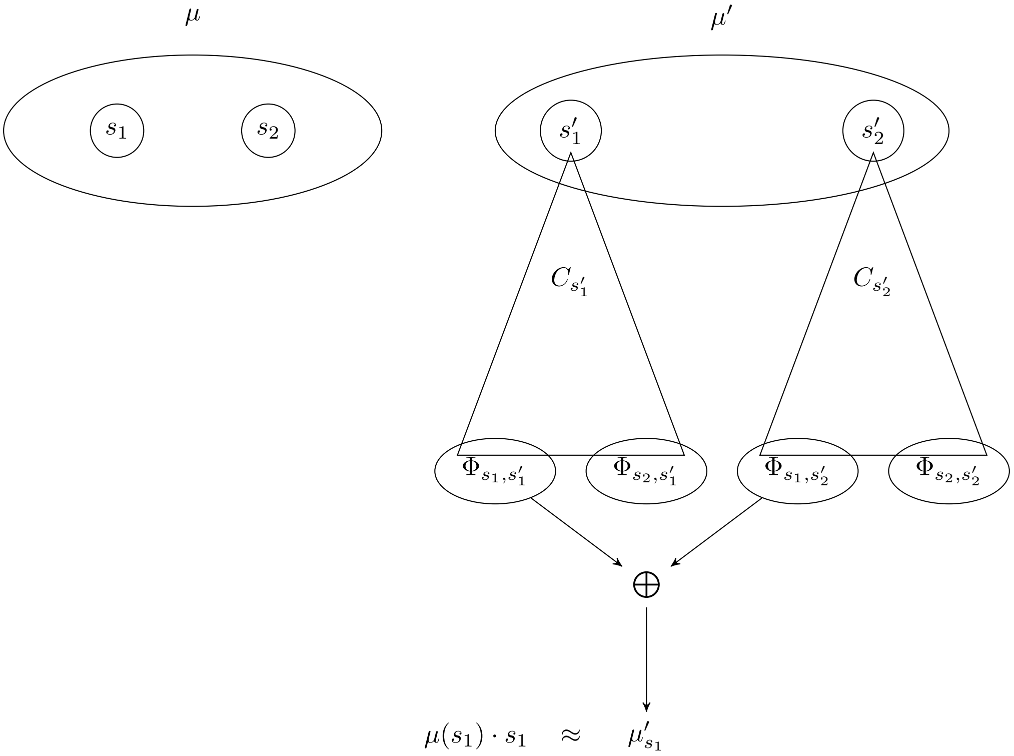


Figure 20: Illustration to Lemma 8.4.

Lemma 8.5 (Existence of weakly simulating computation). *Suppose $s, s' \in S$ and $s \approx s'$. Let $C = (Q, \rightarrow, L, \sigma_0)$ be a simple computation from s. Then there exists a computation $C' = (Q', \rightarrow', L', \gamma_0)$ from s' such that $C \preceq C'$. Moreover, the corresponding function w satisfies the following condition: for every $\sigma \in Q$, $\mu_\sigma \approx \mu_{w(\sigma)}$.*

Proof. Suppose $s \approx s'$ and $C = (Q, \rightarrow, L, \sigma_0)$ is a simple computation from s. Analogously to the proof of Lemma 7.6, we construct C' and w inductively. In particular, we start with a single node labelled with s', and add transitions to simulate the transitions in C in any order satisfying the requirement of Lemma 7.5.

Base case. $C'_0 = (\{\gamma_0\}, \emptyset, L'_0, \gamma_0)$ with $L'_0(\gamma_0) = s'$, and $w_0(\sigma_0) = \{\gamma_0\}$.

Induction step. Given a computation $C'_n = (Q'_n, \rightarrow'_n, L'_n, \gamma_0)$ with the first n transitions of C simulated, and the corresponding w_n, we construct C'_{n+1} and w_{n+1} as follows. Suppose the next transition to be simulated is (σ, α, p, μ) (note that $p = 1$ since C is simple). Let $v = L(\sigma)$. Note that by Definition 6.10, C does not contain ϵ-transitions, and therefore $\alpha \in Act^{\chi}(s)$.

By the induction hypothesis, $\mu_{\sigma} \approx \mu_{w_n(\sigma)}$. By Lemma 8.4, we can append a τ-computation to every $\gamma \in w_n(\sigma)$ so that the distribution induced by the new leaves (let us denote this set of leaves by Φ) can simulate any transition from v, including the α-transition under consideration. Then, using Lemma 7.2, append the corresponding α-computation to each of the nodes in Φ. The subdistribution induced by the newly added leaves (let us denote this set by Ψ) is weakly bisimilar to $\mu_{Children(\sigma)}$. Finally, we apply Lemma 8.4 again, appending the corresponding τ-computation to each of the nodes in Ψ. The newly added leaves (denoted Θ) can be partitioned into $|Children(\sigma)|$ subsets

$$\Theta = \biguplus_{\sigma' \in Children(\sigma)} \Theta_{\sigma'}$$

with

$$\mu_{\sigma'} \approx \mu_{\Theta_{\sigma'}}$$

for each $\sigma' \in Children(\sigma)$. We then set $w_{n+1}(\sigma') = \Theta_{\sigma'}$ for each $\sigma' \in Children(\sigma)$ (and $w_{n+1}(\sigma) = w_n(\sigma)$ for every $\sigma \in Q_n$).

Having constructed an infinite sequence of computations C'_n and functions w_n, we set

$$C' = \left(\bigcup_{n=0}^{\infty} Q'_n, \bigcup_{n=0}^{\infty} \rightarrow'_n, \bigcup_{n=0}^{\infty} L'_n, \gamma_0 \right)$$

and

$$w = \bigcup_{n=0}^{\infty} w_n.$$

It is not difficult to see that conditions in Definition 6.36 are satisfied. Furthermore, by construction, for every $\sigma \in Q$, $\mu_{\sigma} \approx \mu_{w(\sigma)}$.

The fact that C' is non-terminating (see Definition 6.6) can be established as in Lemma 7.7. $\square$

Theorem 8.6 (Preservation of minimum expected time under weak bisimulation). *Let $s, s' \in S$. Suppose $G \subseteq S$ is the union of zero or more equivalence classes under $\approx$ such that each equivalence class in G contains a state with no outgoing τ-transitions. Then,*

$$s \approx s' \quad \textit{implies} \quad eT^{min}(s, \Diamond G) = eT^{min}(s', \Diamond G).$$

Proof. By Lemma 6.41 and the symmetry of $\approx$, we only need to prove that for every simple computation $C_s = (Q, \rightarrow, L, \sigma_0)$ from s there exists a weakly simulating computation $C_{s'} = (Q', \rightarrow', L', \gamma_0)$ from s' such that the corresponding function w satisfies the following requirement: for every $\sigma \in Q, \gamma \in Q'$ it holds that

$$L(\sigma) \in G \wedge \gamma \in w(\sigma) \quad \text{implies} \quad L'(\gamma) \in G.$$

To this end, suppose $C_s = (Q, \rightarrow, L, \sigma_0)$ is a simple computation from s. Then, by Lemma 8.5, there exists a weakly simulating computation $C_{s'} = (Q', \rightarrow', L', \gamma_0)$ from s' such that for every $\sigma \in Q$, $\mu_\sigma \approx \mu_{w(\sigma)}$. Note that the latter condition implies that

$$L(\sigma) \quad \approx \quad \frac{1}{\Pr(\sigma)} \cdot \mu_{w(\sigma)}.$$

Now, suppose $\sigma \in Q$ is such that $L(\sigma) \in G$. Then there must exist a state $t \in S$ with no outgoing τ-transitions such that $L(\sigma) \approx t$. But then $t \approx \frac{1}{\Pr(\sigma)} \cdot \mu_{w(\sigma)}$, and by Lemma 8.2, for every $\gamma \in w(\sigma)$ it holds that $L(\sigma) \approx t \approx L'(\gamma)$. Therefore, $L'(\gamma) \in G$. The result follows. ☐

Theorem 8.7 (Preservation of minimum long run average under weak bisimulation). *Let $s, s' \in S$. Suppose $G \subseteq S$ is the union of zero or more equivalence classes under $\approx$. Then,*

$$s \approx s' \quad \textit{implies} \quad LRA^{min}(s, \Diamond G) = LRA^{min}(s', \Diamond G).$$

Proof. By Lemma 6.43 and the symmetry of $\approx$, we only need to prove that for every simple computation $C_s = (Q, \rightarrow, L, \sigma_0)$ from s there exists a weakly simulating computation $C_{s'} = (Q', \rightarrow', L', \gamma_0)$ from s' such that the corresponding function w satisfies the following requirement: for every $\sigma \in Q$ with $L(\sigma) \in MS$, every $\gamma \in w(\sigma)$ and every $\gamma' \in Children^*(\gamma)$ with $L'(\gamma') \in MS$ and $Tr(\gamma) = Tr(\gamma')$ (i.e. for every γ' with $L'(\gamma') \in MS$ reachable from γ via zero or more τ- and ϵ-transitions) it holds that

$$L(\sigma) \in G \quad \text{if and only if} \quad L'(\gamma') \in G.$$

To this end, suppose $C_s = (Q, \rightarrow, L, \sigma_0)$ is a simple computation from s. Then, by Lemma 8.5, there exists a weakly simulating computation $C_{s'} = (Q', \rightarrow', L', \gamma_0)$ from s' such that for every $\sigma \in Q$, $\mu_\sigma \approx \mu_{w(\sigma)}$.

Now, let $\sigma \in Q$ with $L(\sigma) \in MS$ (note that it implies that $L(\sigma)$ has no outgoing τ-transitions), $\gamma \in w(\sigma)$, and $\gamma' \in Children^*(\gamma)$ with $L'(\gamma') \in MS$ and $Tr(\gamma) = Tr(\gamma')$.

By Lemma 8.2, $L(\sigma) \approx L'(\gamma)$. But $L'(\gamma')$ must be reachable from $L'(\gamma)$ via zero or more τ-transitions (because γ' is reachable from γ via zero or more τ- or ϵ-transitions). Then, by Lemma 8.3, $L(\sigma) \approx L'(\gamma')$.

Now, since G is closed under $\approx$, it follows that

$$L(\sigma) \in G \quad \text{if and only if} \quad L'(\gamma') \in G,$$

as desired. $\square$

9 Conclusion

In this work we have investigated the preservation of certain properties of the states of a Markov automaton under four different bisimulation relations. In particular:

1. Following [7, 8, 14, 13, 9], we defined the syntax and semantics of Markov automata, four different kinds of bisimulations on the states of an MA as well as two state properties: minimum expected time and minimum long run average.
2. We defined the strong bisimulation quotient $MA/_{\sim}$ of a Markov automaton MA, quotient schedulers and bisimulation-closed sets of finite paths, and showed that the probabilities of such sets of paths are preserved by quotient schedulers.
3. We used the bisimulation quotient and the fixpoint characterization of minimum expected time to prove that this property is preserved under strong bisimulation.
4. We used the concept of quotient schedulers to show that minimum long run average is preserved under strong bisimulation.
5. We defined computations and explored the connection between time-abstract schedulers and computations. Furthermore, we defined expected time and long run average can in computations and showed that the new definition is consistent with the definition involving schedulers.
6. We used computations to prove that minimum expected time and long run average are preserved under naive weak and strong bisimulations.
7. We also showed that, in general, minimum expected time is *not* preserved under weak bisimulation, but *is* preserved if every equivalence class under $\approx$ in the target set contains a state with no outgoing τ-transitions.
8. Finally, we proved that minimum long run average is preserved under weak bisimulation.

We hope that our results will prove useful in the field of model checking of Markov automata, since they provide a way to replace an MA with a smaller bisimilar automaton that still satisfies the same properties. We also feel that the notions of quotient schedulers and computations are general enough to be useful in proving other results about Markov automata.

9.1 Future work

We would like to conclude by listing several ideas for future research.

1. In this work we restricted ourselves to *minimum* expected time and long run average. It is quite clear that in the case of long run average our approach can be easily extended to the

maximum case. In fact, here is the proof:

$$\begin{array}{rcl} LRA^{max}\left(s, \Diamond G\right) & = & LRA^{D}\left(s, \Diamond G\right) \\ & = & LRA^{D}\left(s, \Diamond\left(G \cap MS\right)\right) \\ \text{(by Lemmas 6.14 and 6.33)} & = & LRA^{C}\left(s, \Diamond\left(G \cap MS\right)\right) \\ \text{(by assumption and Lemma 6.42)} & = & LRA^{C'}\left(s', \Diamond\left(G \cap MS\right)\right) \\ \text{(by Lemma 6.35)} & = & LRA^{D_{C'}}\left(s', \Diamond\left(G \cap MS\right)\right) \\ \text{(by the definition of } LRA^{max}\text{)} & \leq & LRA^{max}\left(s', \Diamond\left(G \cap MS\right)\right) \\ & = & LRA^{max}\left(s', \Diamond G\right). \end{array}$$

Maximum expected time is more problematic, however, as can be seen from the following "proof":

$$\begin{array}{rcl} eT^{max}\left(s, \Diamond G\right) & = & eT^{D}\left(s, \Diamond G\right) \\ & = & eT^{C}\left(s, \Diamond G\right) \\ \text{(problem!)} & \geq & eT^{C'}\left(s', \Diamond G\right) \\ & = & eT^{D_{C'}}\left(s', \Diamond G\right) \\ & \leq & eT^{max}\left(s', \Diamond G\right). \end{array}$$

It is now clear that the case of maximum long run average works out because we proved strict equality in Lemma 6.42, and the same is not true for Lemma 6.40. Investigating the preservation of maximum expected time is, therefore, the first idea for future research.

2. We think that our approach involving computations can be used to prove the preservation of other properties, for example minimum time bounded reachability. In fact, we conjecture that minimum time bounded reachability is preserved under strong, branching and naive weak bisimulations (and possibly under weak bisimulation if each equivalence class in the target set contains a state without τ-transitions).

3. To the best of our knowledge, no efficient (polynomial) algorithms exist to compute the four bisimulations. The development of such algorithms seems to be a prerequisite for applying our results in practice.

References

[1] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT Press, 2008.

[2] Yuxin Deng and Matthew Hennessy. On the Semantics of Markov Automata. *Inf. Comput.*, 222:139–168, 2013.

[3] Josée Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. Weak Bisimulation is Sound and Complete for pCTL*. *Inf. Comput.*, 208(2):203–219, 2010.

[4] Christian Eisentraut, Holger Hermanns, Joost-Pieter Katoen, and Lijun Zhang. A Semantics for Every GSPN. In José Manuel Colom and Jörg Desel, editors, *Petri Nets*, volume 7927 of *Lecture Notes in Computer Science*, pages 90–109. Springer, 2013.

[5] Christian Eisentraut, Holger Hermanns, Julia Krämer, Andrea Turrini, and Lijun Zhang. Deciding Bisimilarities on Distributions. In Joshi et al. [10], pages 72–88.

[6] Christian Eisentraut, Holger Hermanns, Johann Schuster, Andrea Turrini, and Lijun Zhang. The Quest for Minimal Quotients for Probabilistic Automata. In Nir Piterman and Scott A. Smolka, editors, *TACAS*, volume 7795 of *Lecture Notes in Computer Science*, pages 16–31. Springer, 2013.

[7] Christian Eisentraut, Holger Hermanns, and Lijun Zhang. Concurrency and Composition in a Stochastic World. In Paul Gastin and François Laroussinie, editors, *CONCUR*, volume 6269 of *Lecture Notes in Computer Science*, pages 21–39. Springer, 2010.

[8] Christian Eisentraut, Holger Hermanns, and Lijun Zhang. On Probabilistic Automata in Continuous Time. In *LICS*, pages 342–351. IEEE Computer Society, 2010.

[9] Dennis Guck, Hassan Hatefi, Holger Hermanns, Joost-Pieter Katoen, and Mark Timmer. Modelling, Reduction and Analysis of Markov Automata. In Joshi et al. [10], pages 55–71.

[10] Kaustubh R. Joshi, Markus Siegle, Mariëlle Stoelinga, and Pedro R. D'Argenio, editors. *Quantitative Evaluation of Systems - 10th International Conference, QEST 2013, Buenos Aires, Argentina, August 27-30, 2013. Proceedings*, volume 8054 of *Lecture Notes in Computer Science*. Springer, 2013.

[11] Martin R. Neuhäußer. *Model Checking Nondeterministic and Randomly Timed Systems*. PhD thesis, RWTH Aachen University, 2010.

[12] Roberto Segala and Nancy A. Lynch. Probabilistic Simulations for Probabilistic Processes. In Bengt Jonsson and Joachim Parrow, editors, *CONCUR*, volume 836 of *Lecture Notes in Computer Science*, pages 481–496. Springer, 1994.

[13] M. Timmer. *Efficient Modelling, Generation and Analysis of Markov Automata*. PhD thesis, University of Twente, Enschede, September 2013.

[14] Mark Timmer, Jaco van de Pol, and Mariëlle Stoelinga. Confluence Reduction for Markov Automata. In Víctor A. Braberman and Laurent Fribourg, editors, *FORMATS*, volume 8053 of *Lecture Notes in Computer Science*, pages 243–257. Springer, 2013.

[15] Rob J. van Glabbeek and W. P. Weijland. Branching Time and Abstraction in Bisimulation Semantics. *J. ACM*, 43(3):555–600, 1996.

[16] Nicolás Wolovick and Sven Johr. A Characterization of Meaningful Schedulers for Continuous-Time Markov Decision Processes. In Eugene Asarin and Patricia Bouyer, editors, *FORMATS*, volume 4202 of *Lecture Notes in Computer Science*, pages 352–367. Springer, 2006.

[17] Lijun Zhang and Martin R. Neuhäußer. Model Checking Interactive Markov Chains. In Javier Esparza and Rupak Majumdar, editors, *TACAS*, volume 6015 of *Lecture Notes in Computer Science*, pages 53–68. Springer, 2010.