1

Bisimulation and Logic

Colin Stirling
School of Informatics
Edinburgh University
cps@staffmail.ed.ac.uk

1.1 Introduction

Bisimulation is a rich concept which appears in various areas of theoretical computer science as this book testifies[1]. Besides its origin by Park [Pa81] as a small refinement of the behavioural equivalence originally defined by Hennessy and Milner between basic concurrent processes [HM80, HM85], it was independently, and earlier, defined and developed in the context of the model theory of modal logic (under the names of *p-relations* and *zigzag relations*) by Van Benthem [vB84] to give an exact account of which subfamily of first-order logic is definable in modal logic. Interestingly, to make their definition of process equivalence more palatable, Hennessy and Milner introduced a modal logic to characterize it.

A labelled transition system (LTS) is a triple $(Pr, Act, \rightarrow)$, where Pr is a non-empty set of states or processes, Act is a set of labels and $\rightarrow \subseteq \wp(Pr \times Act \times Pr)$ is the transition relation. As usual, we write $P \xrightarrow{a} Q$ when $(P, a, Q) \in \rightarrow$. A transition $P \xrightarrow{a} Q$ indicates that P can perform action a and become Q. In logical presentations, there is often extra structure in a transition system, a labelling of states with atomic propositions (or colours): let *Prop* be a set of

[1] This is a draft chapter for a book on bisimulation (to be edited by Jan Rutten and Davide Sangiorgi).

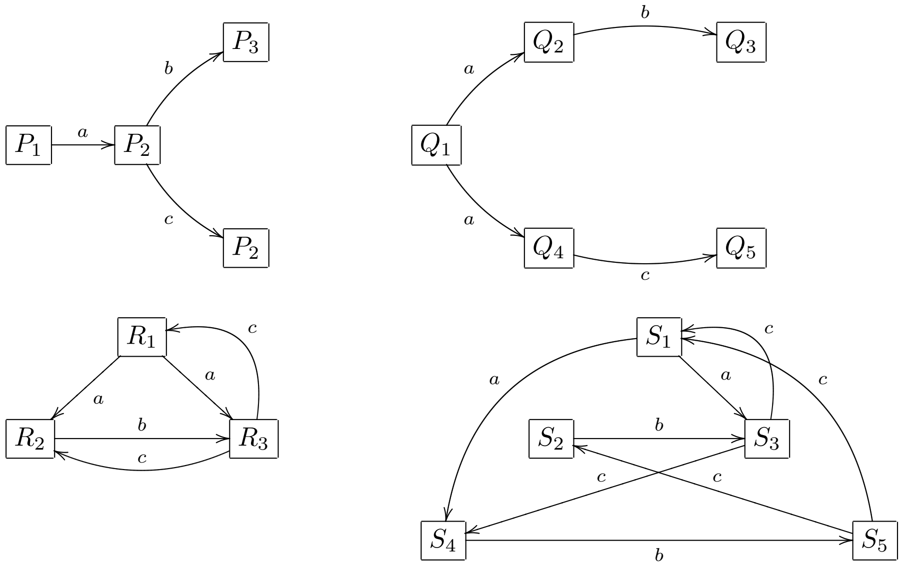

Fig. 1.1. Examples of bisimilar and non-bisimilar processes

propositions with elements p, q. Formally, this extra component is a *valuation*, a function $V : \mathit{Prop} \rightarrow \wp(\mathit{Pr})$ that maps each $p \in \mathit{Prop}$ to a set $V(p) \subseteq \mathit{Pr}$ (those states coloured p). An LTS with a valuation is often called a *Kripke model*[2].

We recall the important definition of bisimulation and bisimilarity.

Definition 1.1.1 A binary relation $\mathcal{R}$ on states of an LTS is a *bisimulation* if whenever $P_1 \mathrel{\mathcal{R}} P_2$ and $a \in A$,

(1) for all P_1' with $P_1 \xrightarrow{a} P_1'$, there is P_2' such that $P_2 \xrightarrow{a} P_2'$ and $P_1' \mathrel{\mathcal{R}} P_2'$;
(2) for all P_2' with $P_2 \xrightarrow{a} P_2'$, there is P_1' such that $P_1 \xrightarrow{a} P_1'$ and $P_1' \mathrel{\mathcal{R}} P_2'$.

P_1 is *bisimilar* to P_2, $P_1 \sim P_2$, if there is a bisimulation $\mathcal{R}$ with $P_1 \mathrel{\mathcal{R}} P_2$. $\square$

In the case of an enriched LTS with valuation V there is an extra clause in the definition of a bisimulation that it preserves colours.

(0) for all $p \in \mathit{Prop}$, $P_1 \in V(p)$ iff $P_2 \in V(p)$.

Definition 1.1.1 assumes that a bisimulation relation is between states of a single LTS. Occasionally, we also allow bisimulations between states of *different* LTSs (a minor relaxation because the disjoint union of two LTSs is an LTS).

[2] Traditionally, a Kripke model has unlabelled transitions of the form $P \rightarrow Q$ representing that state Q is *accessible* to P.

Example 1.1.2 In Figure 1.1, $R_1 \sim S_1$ because the following relation $\mathcal{R}$ is a bisimulation $\{(R_1, S_1), (R_2, S_2), (R_2, S_4), (R_3, S_3), (R_3, S_5)\}$. For instance, take the pair $(R_3, S_3) \in \mathcal{R}$; we need to show it obeys the hereditary conditions of Definition 1.1.1. $R_3 \xrightarrow{c} R_1$ and $R_3 \xrightarrow{c} R_2$; however, $S_3 \xrightarrow{c} S_1$ and $(R_1, S_1) \in \mathcal{R}$; also, $S_3 \xrightarrow{c} S_4$ and $(R_2, S_4) \in \mathcal{R}$. If this transition system were enriched with $V(p) = \{R_2, S_4\}$ then R_1 and S_1 would no longer be bisimilar. Furthermore, $P_1 \not\sim Q_1$ because P_2 can engage in both b and c transitions whereas Q_2 and Q_4 cannot. $\square$

In the remainder of this chapter, we shall describe key relationships between logics and bisimulation. In Section 1.2, we examine Henessy-Milner's modal characterisation of bisimilarity. In Section 1.3 we prove van Benthem's expressiveness result that modal logic corresponds to the fragment of first-order logic that is bisimulation invariant. These results are then extended in Sections 1.4 and 1.5 to modal mu-calculus, that is, modal logic with fixed-points, and to the bisimulation invariant fragment of monadic second-order logic.

1.2 Modal logic and bisimilarity

Let M be the following modal logic where a ranges over Act.

$$\phi \ ::= \ \texttt{tt} \ \mid \ \neg\,\phi \ \mid \ \phi_1 \vee \phi_2 \ \mid \ \langle a\rangle\phi$$

A formula is either the "true" formula $\texttt{tt}$, the negation of a formula, $\neg\phi$, a disjunction of two formulas, $\phi_1 \vee \phi_2$, or a *modal* formula, $\langle a\rangle\phi$, "diamond a ϕ". M is often called Hennessy-Milner logic as it was introduced by Hennessy and Milner to clarify process equivalence [HM80, HM85]. Unlike a standard presentation of modal logic at that time, such as [Ch80], it is *multi-modal*, involving families of modal operators, one for each element of Act, and it avoids atomic propositions. The inductive stipulation below defines when a state $P \in Pr$ of a LTS L has a modal property ϕ, written $P \models_L \phi$; however we drop the index L.

$$\begin{array}{lll}
P \models \texttt{tt} & & \\
P \models \neg\,\phi & \text{iff} & P \not\models \phi \\
P \models \phi_1 \vee \phi_2 & \text{iff} & P \models \phi_1 \text{ or } P \models \phi_2 \\
P \models \langle a\rangle\phi & \text{iff} & P' \models \phi \text{ for some } P' \text{ with } P \xrightarrow{a} P'
\end{array}$$

The critical clause here is the interpretation of $\langle a\rangle$ as "after some a-transition"; for instance, $Q_1 \models \langle a\rangle\langle b\rangle\texttt{tt}$, where Q_1 is in Figure 1.1, because $Q_1 \xrightarrow{a} Q_2$ and $Q_2 \models \langle b\rangle\texttt{tt}$. In the context of full propositional modal logic over an enriched LTS with a valuation V one adds propositions $p \in Prop$, with semantic clause

$$P \models p \ \text{ iff } \ P \in V(p).$$

Other connectives are introduced as follows: "false", $\mathtt{ff} = \neg\mathtt{tt}$, conjunction, $\phi_1 \wedge \phi_2 = \neg(\neg\phi_1 \vee \neg\phi_2)$, implication, $\phi_1 \rightarrow \phi_2 = \neg\phi_1 \vee \phi_2$ and the dual modal operator "box a", $[a]\phi = \neg\langle a\rangle\neg\phi$. Derived semantic clauses for these defined connectives are as follows.

$$\begin{array}{lll} P \not\models \mathtt{ff} & & \\ P \models \phi_1 \wedge \phi_2 & \text{iff} & P \models \phi_1 \text{ and } P \models \phi_2 \\ P \models \phi_1 \rightarrow \phi_2 & \text{iff} & \text{if } P \models \phi_1 \text{ then } P \models \phi_2 \\ P \models [a]\phi & \text{iff} & P' \models \phi \text{ for every } P' \text{ with } P \xrightarrow{a} P' \end{array}$$

So, $[a]$ means "after every a-transition"; for example $P_1 \models [a]\langle b\rangle\mathtt{tt}$ whereas $Q_1 \not\models [a]\langle b\rangle\mathtt{tt}$, where these are in Figure 1.1, because $Q_1 \xrightarrow{a} Q_4$ and $Q_4 \not\models \langle b\rangle\mathtt{tt}$.

Exercise 1.2.1 Show the following using the inductive definition of the satisfaction relation $\models$ where the processes are depicted in Figure 1.1.

(1) $S_2 \models [a](\langle b\rangle\mathtt{tt} \wedge \langle c\rangle\mathtt{tt})$
(2) $S_1 \not\models [a](\langle b\rangle\mathtt{tt} \wedge \langle c\rangle\mathtt{tt})$
(3) $S_2 \models [b][c](\langle a\rangle\mathtt{tt} \vee \langle c\rangle\mathtt{tt})$
(4) $S_1 \models [b][c](\langle a\rangle\mathtt{tt} \vee \langle c\rangle\mathtt{tt})$ □

A natural notion of equivalence between states of an LTS is induced by the modal logic (with or without atomic propositions).

Definition 1.2.2 P and P' have the same modal properties, written $P \equiv_M P'$, if $\{\phi \in M \mid P \models \phi\} = \{\phi \in M \mid P' \models \phi\}$. □

Bisimilar states have the same modal properties.

Theorem 1.2.3 If $P \sim P'$ then $P \equiv_M P'$.

Proof By structural induction on $\phi \in M$ we show for any P, P' if $P \sim P'$ then $P \models \phi$ iff $P' \models \phi$. The base case is when ϕ is $\mathtt{tt}$ which is clear (as is the case $p \in \mathit{Prop}$ when considering an enriched LTS). For the inductive step, there are three cases when $\phi = \neg\phi_1$, $\phi = \phi_1 \vee \phi_2$ and $\phi = \langle a\rangle\phi_1$, assuming the property holds for ϕ_1 and for ϕ_2. We just consider the last of these three and leave the other two as an exercise for the reader. Assume $P \models \langle a\rangle\phi_1$. So, $P \xrightarrow{a} P_1$ and $P_1 \models \phi_1$ for some P_1. However, $P \sim P'$ and so $P' \xrightarrow{a} P_1'$ for some P_1' such that $P_1 \sim P_1'$. By the induction hypothesis, if $Q \sim Q'$ then $Q \models \phi_1$ iff

$Q' \models \phi_1$. Therefore, $P'_1 \models \phi_1$ because $P_1 \models \phi_1$ and so $P' \models \langle a\rangle\phi_1$, as required. A symmetric argument applies if $P' \models \langle a\rangle\phi$. □

The converse is true in the circumstance that the LTS is *image-finite*: that is, when the set $\{P' \mid P \xrightarrow{a} P'\}$ is finite for each $P \in Pr$ and $a \in Act$.

Theorem 1.2.4 If the LTS is image-finite and $P \equiv_M P'$ then $P \sim P'$.

Proof By showing that the binary relation $\equiv_M$ is a bisimulation. Assume $P \equiv_M P'$. If the LTS is enriched then, clearly, $P \models p$ iff $P' \models p$ for any $p \in Prop$. Assume $P \xrightarrow{a} P_1$. We need to show that $P' \xrightarrow{a} P'_i$ such that $P_1 \equiv_M P'_i$. Since $P \models \langle a\rangle\mathtt{tt}$ also $P' \models \langle a\rangle\mathtt{tt}$ and, so, the set $\{P'_i \mid P' \xrightarrow{a} P'_i\}$ is non-empty. As the LTS is image-finite, this set is finite, say $\{P'_1, \ldots, P'_n\}$. If $P_1 \not\equiv_M P'_i$ for each $i : 1 \leq i \leq n$ then there are formulas $\phi_1, \ldots, \phi_n$ of M where $P_1 \not\models \phi_i$ and $P'_i \models \phi_i$ and so $P_1 \not\models \phi'$ and $P'_i \models \phi'$ for each i when $\phi' = \phi_1 \vee \ldots \vee \phi_n$. But this contradicts $P \equiv_M P'$ as $P \not\models [a]\phi'$ and $P' \models [a]\phi'$. So, for some P'_i, $1 \leq i \leq n$, $P_1 \equiv_M P'_i$. The proof for the case $P' \xrightarrow{a} P'_1$ is symmetric. □

Theorems 1.2.3 and 1.2.4 together are known as the *Hennessy-Milner Theorem*, the modal characterisation of bisimilarity. Modal formulas can, therefore, be witnesses for inequivalent (image-finite) processes; an example is that $\langle a\rangle[b]\mathtt{ff}$ distinguishes Q_1 and P_1 of Figure 1.1.

Exercise 1.2.5 Sets of formulas of M can be stratified according to their modal depth. The *modal depth* of $\phi \in M$, $\mathrm{md}(\phi)$, is defined inductively: $\mathrm{md}(\mathtt{tt}) = 0$; $\mathrm{md}(\neg\phi) = \mathrm{md}(\phi)$; $\mathrm{md}(\phi_1 \vee \phi_2) = \max\{\mathrm{md}(\phi_1), \mathrm{md}(\phi_2)\}$; $\mathrm{md}(\langle a\rangle\phi) = \mathrm{md}(\phi) + 1$. Let $\equiv^n_M$ mean having the same modal properties with modal depth at most n and recall the stratified bisimilar relations $\sim_n$. What Hennessy and Milner showed is $P \sim_n P'$ iff $P \equiv^n_M P'$.

(1) Prove by induction on n, $P \sim_n P'$ iff $P \equiv^n_M P'$.
(2) Therefore, show that the restriction to image-finite LTSs in Theorem 1.2.4 is essential.
(3) Assume an LTS where *Act* is finite and which need not be image-finite. Show that for each $P \in Pr$ and for each $n \geq 0$, there is a formula ϕ of modal depth n such that $P' \models \phi$ iff $P' \sim_n P$. (Hint: if *Act* is finite then for each $n \geq 0$ there are only finitely many inequivalent formulas of model depth n.) □

Exercise 1.2.6 Let M^∞ be modal logic M with arbitrary countable disjunction (and, therefore, conjunction because of negation). If Φ is a countable set of formulas then $\bigvee \Phi$ is a formula whose semantics is: $P \models \bigvee \Phi$ iff $P \models \phi$ for some $\phi \in \Phi$. Prove that if Pr is a countable set then $P \sim Q$ iff $P \equiv_{M^\infty} Q$. $\Box$

Next, we identify when a process has the Hennessy-Milner property [BRV01].

Definition 1.2.7 $P \in Pr$ has the *Hennessy-Milner property* iff if $P' \equiv_M P$ then $P' \sim P$. $\Box$

Exercise 1.2.8 Pr is *modally saturated* if for each $a \in Act$, $P \in Pr$ and $\Phi \subseteq M$ if for each finite set $\Phi' \subseteq \Phi$ there is a $Q \in \{Q \mid P \xrightarrow{a} Q\}$ and $Q \models \phi$ for all $\phi \in \Phi'$ then there is a $Q \in \{Q \mid P \xrightarrow{a} Q\}$ such that $Q \models \phi$ for all $\phi \in \Phi$. Show that if Pr is modally saturated then each $P \in Pr$ has the Hennessy-Milner property. (See, for instance, [BRV01] for the notion of modal saturation and how to build LTSs with this feature using ultrafilter extensions.) $\Box$

A formula ϕ is *characteristic* for process P (with respect to bisimilarity) provided that $P \models \phi$ and if $P' \models \phi$ then $P' \sim P$. An LTS is *acyclic* if its transition graph does not contain cycles; that is, if $P \in Pr$ and $P \xrightarrow{a} P'$ and $P' \xrightarrow{a_1} P_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} P_n$ for $n \geq 0$ then $P \neq P_n$.

Proposition 1.2.9 Assume an acyclic LTS $(Pr, Act, \longrightarrow)$ where Pr, Act and $Prop$ are finite. If $P \in Pr$ then there is a formula $\phi \in M$ that is characteristic for P.

Proof Assume an acyclic LTS with finite sets Pr, Act and $Prop$. For each $P \in Pr$ we define a propositional formula $\mathrm{PROP}(P)$ and for each $a \in Act$ a modal formula $\mathrm{MOD}(a, P)$. Then $\mathrm{FORM}(P) = \mathrm{PROP}(P) \wedge \bigwedge\{\mathrm{MOD}(a, P) \mid a \in Act\}$ is the characteristic formula for P.

$$\begin{array}{lcl} \mathrm{PROP}(P) & = & \bigwedge\{p \in \mathit{Prop} \mid P \models p\} \wedge \bigwedge\{\neg p \in \mathit{Prop} \mid P \not\models p\} \\ \mathrm{MOD}(a, P) & = & \bigwedge\{\langle a\rangle \mathrm{FORM}(P') \mid P \xrightarrow{a} P'\} \wedge [a] \bigvee\{\mathrm{FORM}(P') \mid P \xrightarrow{a} P'\} \end{array}$$

where as usual $\bigwedge \emptyset = \mathtt{tt}$ and $\bigvee \emptyset = \mathtt{ff}$. We need to show that $\mathrm{PROP}(P)$ is indeed well-defined and a modal formula; and that it is characteristic for P. The first depends on the fact that the LTS is acyclic and that the sets Pr, Act and $Prop$ are finite; why? The proof that $\mathrm{FORM}(P)$ is characteristic for P is also left as an exercise for the reader. $\Box$

Example 1.2.10 The LTS in Figure 1.2 is acyclic with $Pr = \{P_1, \ldots, P_4\}$,

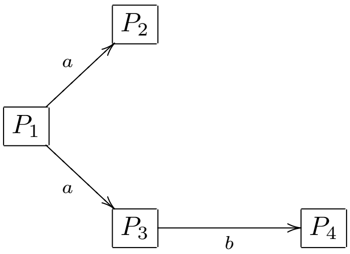

Fig. 1.2. The transition graph for Example 1.2.10

$Act = \{a, b\}$ and $Prop = \emptyset$. Now,

$$
\begin{array}{rcl}
\mathrm{FORM}(P_2) & = & \mathrm{FORM}(P_4) = [a]\mathtt{ff} \wedge [b]\mathtt{ff} \\
\mathrm{FORM}(P_3) & = & \langle a \rangle \mathrm{FORM}(P_4) \wedge [a]\mathrm{FORM}(P_4) \wedge [b[\mathtt{ff} \\
\mathrm{FORM}(P_1) & = & \langle a \rangle \mathrm{FORM}(P_2) \wedge \langle a \rangle \mathrm{FORM}(P_3) \wedge \\
 & & [a](\mathrm{FORM}(P_2) \vee \mathrm{FORM}(P_3)) \wedge [b]\mathtt{ff}
\end{array}
$$

Here, we construct the formulas starting from the nodes P_2 and P_4 that have no outgoing transitions; then we construct the formula for P_3; and then finally for P_1. □

Exercise 1.2.11 Give an example of a finite-state P such that no formula of M is characteristic for P. □

Exercise 1.2.12 Recall that trace equivalence equates two states P and Q if they can perform the same *finite sequences* of transitions.

(1) Show that Proposition 1.2.9 also holds for trace equivalence. That is, assume an acyclic LTS where Pr and Act are finite and $Prop$ is empty. Prove that if $P \in Pr$ then there is formula $\phi \in M$ that is characteristic for P with respect to trace equivalence.
(2) Construct the characteristic formula for P_1 and Q_1 of Figure 1.1.

1.3 Bisimulation invariance

An alternative semantics of modal logic emphasises *properties*. Relative to a LTS and valuation V, let $\|\phi\| = \{P \mid P \models \phi\}$: we can think of $\|\phi\|$ as the property expressed by ϕ on the LTS. In the case of the LTS in Figure 1.2, $\|\langle a \rangle \mathtt{tt} \vee \langle b \rangle \mathtt{tt}\| = \{P_1, P_3\}$.

Exercise 1.3.1 Define $\|\phi\|$ on a LTS directly by induction on ϕ (without appealing to the satisfaction relation $\models$). □

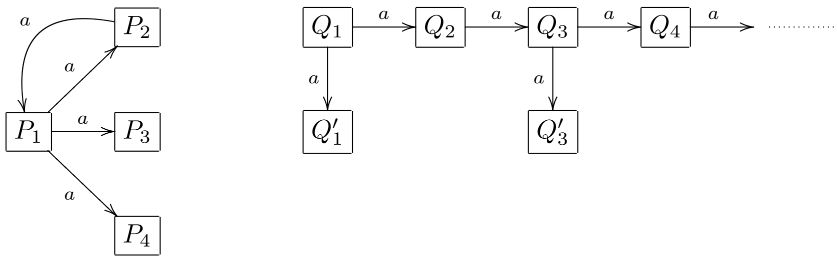

Fig. 1.3. More transition graphs

Another way of understanding Theorem 1.2.3 is that properties of states of an LTS expressed by modal formulas are *bisimulation invariant*: if $P \in \|\phi\|$ and $P \sim P'$ then $P' \in \|\phi\|$. There are many kinds of properties that are not bisimulation invariant. Examples include counting of successor transitions, "has 3 a-transitions", or invocations of finiteness such as "is finite-state" or behavioural cyclicity, "has a sequence of transitions that is eventually cyclic": each of these properties distinguishes P_1 and Q_1 in Figure 1.3 even though $P_1 \sim Q_1$. The definition of invariance is neither restricted to *monadic* properties nor to a particular logic within which properties of LTSs are expressed.

Definition 1.3.2 Assume Pr^n is $(Pr \times \ldots \times Pr)$ n-times, $n \geq 1$.

(1) An $n$ary property, $n \geq 1$, of a LTS is a set $\Gamma \subseteq Pr^n$.
(2) Property $\Gamma \subseteq Pr^n$ is *bisimulation invariant* if whenever $(P_1, \ldots, P_n) \in \Gamma$ and $P_i \sim P'_i$ for each $i : 1 \leq i \leq n$, then also $(P'_1, \ldots, P'_n) \in \Gamma$. □

Exercise 1.3.3

(1) Prove that the property $\{(P, Q) \mid P, Q \text{ are trace equivalent}\}$ is bisimulation invariant. More generally, show that if $\equiv$ is a behavioural equivalence between processes such that $P \sim Q$ implies $P \equiv Q$, then $\equiv$ is bisimulation invariant.
(2) A property $\Gamma \subseteq Pr^n$ is *safe for bisimulation* if whenever $(P_1, \ldots, P_n) \in \Gamma$ and $P_1 \sim P'_1$ then $(P'_1, \ldots, P'_n) \in \Gamma$ for some $P'_2, \ldots, P'_n$ (a notion due to van Benthem [vB98]). Show that the general transition relations $\xrightarrow{w}$, $w \in Act^*$, are safe for bisimulation.
(3) Show that if Γ is bisimulation invariant then it is safe for bisimulation.

□

Another logic within which to express properties of a LTS is first-order logic, FOL. It has a countable family of variables *Var* typically represented as x, y, z

and a binary relation E_a for each $a \in Act$ (and a monadic predicate p for each $p \in Prop$ when the LTS is enriched). Formulas of FOL have the following form.

$$\phi \ ::= \ p(x) \ \mid \ xE_a y \ \mid \ x = y \ \mid \ \neg\, \phi \ \mid \ \phi_1 \vee \phi_2 \ \mid \ \exists x.\, \phi$$

To interpret formulas with free variables we need a valuation $\sigma : Var \to Pr$ that associates a state with each variable. Also, we use a standard "updating" notation: $\sigma\{P_1/x_1, \ldots, P_n/x_n\}$ is the valuation that is the same as σ except that its value for x_i is P_i, $1 \le i \le n$ (where each x_i is distinct). We inductively define when FOL formula ϕ is true on an LTS L with respect to a valuation σ as $\sigma \models_L \phi$, where again we drop the index L.

$$\begin{array}{lll}
\sigma \models p(x) & \text{iff} & \sigma(x) \in V(p) \\
\sigma \models xE_a y & \text{iff} & \sigma(x) \xrightarrow{a} \sigma(y) \\
\sigma \models x = y & \text{iff} & \sigma(x) = \sigma(y) \\
\sigma \models \neg\phi & \text{iff} & \sigma \not\models \phi \\
\sigma \models \phi_1 \vee \phi_2 & \text{iff} & \sigma \models \phi_1 \text{ or } \sigma \models \phi_2 \\
\sigma \models \exists x.\, \phi & \text{iff} & \sigma\{P/x\} \models \phi \text{ for some } P \in Pr
\end{array}$$

The universal quantifier, the dual of $\exists x$, is introduced as $\forall x.\, \phi = \neg\exists\neg\phi$. Its derived semantic clause is: $\sigma \models \forall x.\, \phi$ iff $\sigma\{P/x\} \models \phi$ for all $P \in Pr$.

Example 1.3.4 Assume $\sigma(x_1) = P_1$ and $\sigma(x_2) = Q_1$ of Figure 1.3. Then the following pair hold.

(1) $\sigma \models \exists x.\, \exists y.\, \exists z.\, (x_1 E_a x \wedge x_1 E_a y \wedge x_1 E_a z \wedge x \neq y \wedge x \neq z \wedge y \neq z)$
(2) $\sigma \models \forall y.\, \forall z.\, (x_2 E_a y \wedge y E_a z \to z \neq x_2)$ □

There is a recognized translation of modal formulas into first-order formulas, for instance, see [BRV01].

Definition 1.3.5 The FOL translation of modal formula ϕ relative to variable x is $T_x(\phi)$ which is defined inductively.

$$\begin{array}{lll}
T_x(p) & = & p(x) \\
T_x(\mathtt{tt}) & = & x = x \\
T_x(\neg\phi) & = & \neg\, T_x(\phi) \\
T_x(\phi_1 \vee \phi_2) & = & T_x(\phi_1) \vee T_x(\phi_2) \\
T_x(\langle a \rangle \phi) & = & \exists y.\, xE_a y \wedge T_y(\phi)
\end{array}$$

□

Exercise 1.3.6

(1) For each of the following formulas ϕ, present its FOL translation $T_x(\phi)$.

(a) $[a]\langle b\rangle\mathtt{tt}$
(b) $\langle a\rangle p \to [a]\langle a\rangle p$
(c) $[a]([a]p \to p) \to [a]p$

(2) FOL^2 is first-order logic when *Var* is restricted to two variables $\{x, y\}$ which can be reused in formulas. Show that modal formulas can be translated into FOL^2. □

The translation of modal formulas into FOL, Definition 1.3.5, is clearly correct as it imitates the semantics.

Proposition 1.3.7 $P \models \phi$ iff $\sigma\{P/x\} \models T_x(\phi)$

Proof By structural induction on $\phi \in M$. For the base cases, first $P \models p$ iff $P \in V(p)$ iff $\sigma\{P/x\} \models p(x)$ iff $\sigma\{P/x\} \models T_x(p)$. Similarly, for the other base case, $P \models \mathtt{tt}$ iff $\sigma\{P/x\} \models x = x$ iff $\sigma\{P/x\} \models T_x(\mathtt{tt})$. For the inductive step we only examine the interesting case when $\phi = \langle a\rangle\phi_1$. $P \models \phi$ iff $P' \models \phi_1$ for some P' where $P \xrightarrow{a} P'$ iff $\sigma\{P'/z\} \models T_z(\phi_1)$ for some P' where $P \xrightarrow{a} P'$, by the induction hypothesis, iff $\sigma\{P/x\} \models \exists z.\, xE_a z \wedge T_z(\phi_1)$ iff $\sigma\{P/x\} \models T_x(\phi)$. □

A FOL formula with free variables is bisimulation invariant if the property it expresses is bisimulation invariant.

Definition 1.3.8 Formula $\phi \in \mathrm{FOL}$ whose free variables belong to $\{x_1, \ldots, x_n\}$ is bisimulation invariant if $\{(P_1, \ldots, P_n) \mid \sigma\{P_1/x_1, \ldots, P_n/x_n\} \models \phi\}$ is bisimulation invariant. □

Corollary 1.3.9 Any first-order formula $T_x(\phi)$ is bisimulation invariant. □

Not all first-order formulas are bisimulation invariant. The two formulas in Example 1.3.4 are cases; the first says that 'x_1 has at least three different a-transitions". Van Benthem introduced the notion of bisimulation (as a p-relation and a zig-zag relation) to identify which formulas $\phi(x) \in \mathrm{FOL}$ with one free variable are equivalent to modal formulas [vB96].

Definition 1.3.10 A FOL formula $\phi(x)$ is equivalent to modal $\phi' \in M$ provided that for any LTS and for any state P, $\sigma\{P/x\} \models \phi$ iff $P \models \phi'$. □

Van Benthem proved Proposition 1.3.12, a FOL formula $\phi(x)$ is equivalent to a modal formula iff it is bisimulation invariant. The proof utilises some model theory. Some notation first: if Φ is a set of first-order formulas then $\Phi \models \psi$ provided that for any LTS and valuation σ, if for all $\phi \in \Phi$, $\sigma \models \phi$ then $\sigma \models \psi$. The *compactness theorem* for first-order logic states that if $\Phi \models \psi$ then there is a *finite* set $\Phi' \subseteq \Phi$ such that $\Phi' \models \psi$. Next we state a further property of first-order logic that will also be used.

Fact 1.3.11 If Φ is a set of first-order formulas all of whose free variables belong to $\{x_1, \ldots, x_n\}$ and $\sigma\{P_1/x_1, \ldots, P_n/x_n\} \models \phi$ for all $\phi \in \Phi$, then there is a LTS and processes $P'_1, \ldots, P'_n \in Pr$ such that $\sigma\{P'_1/x_1, \ldots, P'_n/x_n\} \models \phi$ for all $\phi \in \Phi$ and each P'_i has the Hennessy-Milner property (Definition 1.2.7). □

Proposition 1.3.12 A FOL formula $\phi(x)$ is equivalent to a modal formula iff $\phi(x)$ is bisimulation invariant.

Proof If $\phi(x)$ is equivalent to a modal formula ϕ' then $\{P \mid \sigma\{P/x\} \models \phi\} = \|\phi'\|$ which is bisimulation invariant. For the converse property, assume that $\phi(x)$ is bisimulation invariant. Consider the following family $\Phi = \{T_x(\psi) \mid \psi \in M \text{ and } \{\phi(x)\} \models T_x(\psi)\}$. We prove $\Phi \models \phi(x)$ and, therefore, by the compactness theorem, $\phi(x)$ is equivalent to a modal formula ψ' such that $T_x(\psi') \in \Phi$. Assume $\sigma\{P/x\} \models \psi$ for all $\psi \in \Phi$. We need to show that $\sigma\{P/x\} \models \phi$. We choose a P with the Hennessy-Milner property by Fact 1.3.11. Let $\Psi = \{T_x(\psi) \mid P \models \psi\}$. First, $\Phi \subseteq \Psi$. Next we show that $\Psi \cup \{\phi\}$ is satisfiable. For suppose otherwise, $\Psi \models \neg\phi$ and so by the compactness theorem there is a finite subset $\Psi' = \{T_x(\psi_1), \ldots, T_x(\psi_k)\} \subseteq \Psi$ such that $\Psi' \models \neg\phi$. But then $\phi \models T_x(\psi')$ where ψ' is the modal formula $\neg\psi_1 \vee \ldots \vee \neg\psi_k$ and so $T_x(\psi') \in \Phi$ which contradicts that $\Phi \subseteq \Psi$. Therefore, for some Q, $\sigma\{Q/x\} \models \psi$ for all $\psi \in \Psi$ and $\sigma\{Q/x\} \models \phi$. However, $Q \sim P$ and because ϕ is bisimulation invariant, $\sigma\{P/x\} \models \phi$ as required. □

Exercise 1.3.13 Prove that a FOL formula $\phi(x_1, \ldots, x_n)$ is bisimulation invariant iff it is equivalent to a boolean combination of formulas of the following form $T_{x_1}(\psi_{11}), \ldots, T_{x_1}(\psi_{1k_1}), \ldots, T_{x_n}(\psi_{n1}), \ldots, T_{x_n}(\psi_{nk_n})$ for some $k_1, \ldots, k_n \geq 0$. □

An alternative proof of Proposition 1.3.12 appeals to *tree* (or forest) models. A LTS is a forest if it is acyclic and the "target" of each transition is unique; if $P \xrightarrow{a} Q$ and $R \xrightarrow{b} Q$ then $P = R$ and $a = b$. The transition graph that is rooted at Q_1 in Figure 1.3 is a tree (a forest with a single tree).

Given a LTS there is a way of unfolding $P \in Pr$ and all its reachable processes into a tree rooted at P which is called *unravelling*.

Definition 1.3.14 Assume a LTS $L = (Pr, Act, \rightarrow)$ with $P_0 \in Pr$. The k-unravelling of P_0, for $k \geq 0$, is the following LTS, $L_k = (Pr_k, Act, \rightarrow_k)$ where

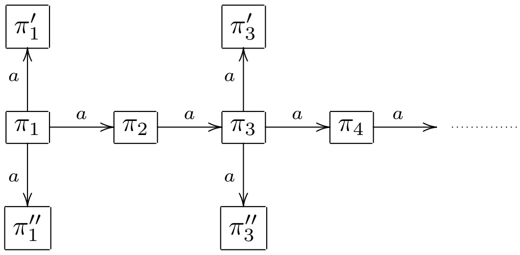

Fig. 1.4. Unravelled LTS

(1) $Pr_k = \{P_0 a_1 k_1 P_1 \dots a_n k_n P_n \mid n \geq 0, 0 \leq k_i \leq k, P_0 \xrightarrow{a_1} P_1 \dots \xrightarrow{a_n} P_n\}$;
(2) if $P \xrightarrow{a} P'$ and P is the final state in $\pi \in Pr_k$ then $\pi \xrightarrow{a}_k \pi a k' P'$ for each $0 \leq k' \leq k$;
(3) if V is the valuation for L then V_k is the valuation for L_k where $V_k(p) = \{\pi \in Pr_k \mid P$ is final in π and $P \in V(p)\}$.

The ω-unravelling of P_0, the LTS L_ω, permits all indices $k \geq 0$: so, Pr_ω includes all sequences $P_0 a_1 k_1 P_1 \dots a_n k_n P_n$ such that $P_0 \xrightarrow{a_1} P_1 \dots \xrightarrow{a_n} P_n$ and each $k_i \geq 0$. □

Example 1.3.15 The 0-unravelling of P_1 of Figure 1.3 is presented in Figure 1.4 where $\pi_1 = P_1$, $\pi_{2(i+1)} = \pi_{2i+1} a 0 P_1$, $\pi_{2i+1} = \pi_{2i} a 0 P_1$ $\pi'_{2i+1} = \pi_{2i+1} a 0 P_3$ and $\pi''_{2i+1} = \pi_{2i+1} a 0 P_4$. The reader is invited to describe the 2-unravelleing and the ω-unravelling of P_1. □

Proposition 1.3.16 For any LTS and $k : 0 \leq k \leq \omega$, if $P \in Pr$ and $\pi \in Pr_k$ and the final state in π is P, then $P \sim \pi$.

Proof Clearly, the binary relation $\mathcal{R} \subseteq \mathrm{Pr} \times \mathrm{Pr}_k$ containing all pairs (P, π) when the final state of π is P is a bisimulation because first, $P \in V(p)$ iff $\pi \in V_k(p)$ and second, $P \xrightarrow{a} P'$ iff $\pi \xrightarrow{a}_k \pi a k' P'$ for $k' \leq k$. □

Exercise 1.3.17 Let R_1 and S_1 be the processes depicted in Figure 1.1.

(1) Define the 0-unravellings of R_1 and S_1.
(2) Define the ω-unravelling of R_1 and S_1 and show that they are isomorphic.
(3) Assume L is a LTS containing P and Q and $P \sim Q$. Show that the ω-unravellings of P and Q are isomorphic.
(4) Reprove Proposition 1.3.12 using ω-unravelled LTSs. □

1.4 Modal mu-calculus

Modal logic M of Section 1.2 is not very expressive. For instance, temporal properties of states of a LTS, such as liveness, "this desirable property will eventually hold", and safety, "this defective property never holds", are not expressible in M. (Prove this; hint, use Exercise 1.2.5.) Such properties have been found to be very useful when analysing the behaviour of concurrent systems. Modal mu-calculus, μM, modal logic with fixpoints, introduced by Kozen [Ko83], has the required extra expressive power.

The setting for μM is the complete lattice generated by the *powerset* construction $\wp(Pr)$ where the ordering is $\subseteq$, join is union and meet is intersection, $\emptyset$ is the bottom element and Pr is the top element.

Exercise 1.4.1 Consider a LTS and recall the definitions of monotone and continuous function f on the powerset $\wp(Pr)$: f is monotone provided that if $S \subseteq S'$ then $f(S) \subseteq f(S')$; f is continuous just in case if $S_1, \ldots, S_n, \ldots$ is an increasing sequence of subsets of Pr, (that is, if $i \leq j$ then $S_i \subseteq S_j \subseteq Pr$), then $f(\bigcup_i S_i) = \bigcup_i f(S_i)$.

(1) Define the semantic functions $\|\langle a\rangle\|$ and $\|[a]\|$ on $\wp(Pr)$ such that for any $\phi \in M$, $\|\langle a\rangle\|\|\phi\| = \|\langle a\rangle\phi\|$ and $\|[a]\|\|\phi\| = \|[a]\phi\|$.
(2) Show that these functions $\|\langle a\rangle\|$ and $\|[a]\|$ are monotone.
(3) Prove that $\|\langle a\rangle\|$ is continuous iff the LTS is image-finite with respect to the label a; that is, if for each $P \in Pr$, the set $\{P' \mid P \xrightarrow{a} P'\}$ is finite.

$\square$

The new constructs of μM over and above those of M are

$$\phi \ ::= \ X \ \mid \ \ldots \ \mid \ \mu X.\, \phi$$

where X ranges over a family of propositional variables. The semantics for a formula ϕ of μM is the set $\|\phi\|_V \subseteq Pr$ where V is a valuation that not only maps elements of *Prop* but also propositional variables to $\wp(Pr)$. As usual we employ updating notation: $\|\phi\|_{V\{S/X\}}$ uses valuation V' like V except that $V'(X) = S$.

Exercise 1.4.2 Assume that ϕ is a formula of M when extended with propositional variables. Prove that if all free occurrences of X in ϕ are within the scope of an even number of negations and V is a valuation then the function $f : \wp(Pr) \to \wp(Pr)$ such that $f(S) = \|\phi\|_{V\{S/X\}}$ is monotone. Therefore, show the following

(1) the least fixed point $\mathbf{lfp}(f)$ exists and is the intersection of all pre-fixed points, $\bigcap\{S \mid f(S) \subseteq S\}$;

(2) the greatest fixed point $\mathbf{gfp}(f)$ exists and is the union of post-fixed points, $\bigcup\{S \mid S \subseteq f(S)\}$. $\square$

In the case of $\mu X.\, \phi$ there is, therefore, the restriction that all free occurrences of X in ϕ are within the scope of an even number of negations (to guarantee monotonicity). This formula expresses the least fixed point **lfp** of the semantic function induced by ϕ. Its dual, $\nu X.\, \phi$, expresses the greatest fixed point **gfp** and is a derived construct in μM: $\nu X.\, \phi = \neg \mu X \neg \phi\{\neg X/X\}$. Here are the semantics for μM formulas.

$$
\begin{array}{rcl}
\|p\|_V & = & V(p) \\
\|Z\|_V & = & V(Z) \\
\|\neg\phi\|_V & = & Pr - \|\phi\|_V \\
\|\phi_1 \vee \phi_2\|_V & = & \|\phi_1\|_V \cup \|\phi_2\|_V \\
\|\langle a\rangle\phi\|_V & = & \{\, P \in Pr \mid \text{for some } Q.\, P \xrightarrow{a} Q \text{ and } Q \in \|\phi\|_V \,\} \\
\|\mu Z.\, \phi\|_V & = & \bigcap\{S \subseteq Pr \mid \|\phi\|_{V\{S/Z\}} \subseteq S\}
\end{array}
$$

Exercise 1.4.3 Extend the first part of Exercise 1.4.2 by proving that if all free occurrences of X in $\phi \in \mu M$ are within the scope of an even number of negations and V is a valuation then the function f on $\wp(Pr)$ such that $f(S) = \|\phi\|_{V\{S/X\}}$ for $S \subseteq Pr$ is monotone. $\square$

Derived semantic clauses for other connectives are below.

$$
\begin{array}{rcl}
\|\phi_1 \wedge \phi_2\|_V & = & \|\phi_1\|_V \cap \|\phi_2\|_V \\
\|[a]\phi\|_V & = & \{P \in Pr \mid \text{for all } Q.\ \text{if } P \xrightarrow{a} Q \text{ then } Q \in \|\phi\|_V\} \\
\|\nu Z.\, \phi\|_V & = & \bigcup\{S \subseteq Pr \mid S \subseteq \|\phi\|_{V\{S/Z\}}\}
\end{array}
$$

P satisfies the μM formula ϕ relative to valuation V, $P \models_V \phi$, iff $P \in \|\phi\|_V$; as usual we omit V wherever possible.

The standard theory of fixpoints tells us that if f is a monotone function on a lattice, we can construct $\mathbf{lfp}(f)$ by applying f repeatedly on the least element of the lattice to form an increasing chain, whose limit is the least fixed point. Similarly, $\mathbf{gfp}(f)$ is constructed by applying f repeatedly on the largest element to form a decreasing chain, whose limit is the greatest fixed point. The stages of these iterations $\mu^\alpha X.\, \phi$ and $\nu^\alpha X.\, \phi$ can be be defined as M^∞ formulas, see Exercise 1.2.6, inductively as follows.

$$
\begin{array}{rclrcl}
\mu^0 X.\,\phi & = & \mathtt{ff} & \nu^0 X.\,\phi & = & \mathtt{tt} \\
\mu^{\beta+1} X.\,\phi & = & \phi\{\mu^\beta X.\,\phi/X\} & \nu^{\beta+1} X.\,\phi & = & \phi\{\nu^\beta X.\,\phi/X\} \\
\mu^\lambda X.\,\phi & = & \bigvee_{\beta<\lambda} \mu^\beta X.\,\phi & \nu^\lambda X.\,\phi & = & \bigwedge_{\beta<\lambda} \nu^\beta X.\,\phi
\end{array}
$$

So for a minimal fixpoint formula $\mu X.\,\phi$, if P satisfies the fixpoint, it satisfies some iterate, say the $\beta + 1$ th so that $P \models \mu^{\beta+1} X.\,\phi$. Now if we *unfold* this formula once, we get $P \models \phi\{\mu^\beta X.\,\phi/X\}$. Therefore, the fact that P satisfies the fixpoint depends, via ϕ, on the fact that other states in Pr satisfy the fixpoint *at smaller iterates than P does.* So if one follows a chain of dependencies, the chain terminates. Therefore, μ means 'finite looping', which, with a little refinement, is sufficient to understand the logic μM. On the other hand, for a maximal fixpoint $\nu X.\,\phi$, there is no such decreasing chain: $P \models \nu X.\,\phi$ iff $P \models \nu^\beta X.\,\phi$ for every iterate β iff $P \models \phi\{\nu^\beta X.\,\phi/X\}$ for every iterate β iff $P \models \phi\{\nu X.\,\phi/X\}$, and so we may loop for ever.

Example 1.4.4 Assume P_1 is the process in Figure 1.3, which can repeatedly do an a transition. P_1 fails to have the property $\mu X.\,[a]X$ (which expresses that there cannot be an infinite sequence of a transitions). Consider its iterates, $\mu^1 X.\,[a]X = [a]\mathtt{ff}$, so P_3 and P_4 have this property; $\mu^3 X.\,[a]X$ is $[a][a][a]\mathtt{ff}$ and $\mu^\omega X.\,[a]X$ is $\bigvee_{n\geq 0}[a]^n\mathtt{ff}$ where $[a]^0\mathtt{ff} = \mathtt{ff}$ and $[a]^{i+1}\mathtt{ff} = [a][a]^i\mathtt{ff}$. Consequently, $P_1 \models \nu X.\,\langle a\rangle X$. Iterates of this formula include $\nu^\omega X.\,\langle a\rangle X = \bigwedge_{n\geq 0}\langle a\rangle^n\mathtt{tt}$ where $\langle a\rangle^i$ is $\langle a\rangle$ i-times. □

Exercise 1.4.5 What properties are expressed by the following formulas?

(1) $\mu X.\, p \vee [a]X$
(2) $\mu X.\, q \vee (p \wedge \langle a\rangle X)$
(3) $\nu X.\, \neg p \wedge [a]X$
(4) $\mu X.\, \nu Y.\, (p \wedge [a]X) \vee (\neg p \wedge [a]Y)$ □

Definition 1.2.2 of $\equiv_M$, "having the same modal properties", is extended to μM; so, $P \equiv_{\mu M} P'$ means P and P' have the same μM properties, as expressed by *closed* formulas of μM (that is, formulas without free variables). Bisimilar states have the same μM properties.

Theorem 1.4.6 If $P \sim P'$ then $P \equiv_{\mu M} P'$.

Proof The proof of this uses Exercise 1.2.6 that M^∞ characterizes bisimilarity and the observation above that closed formulas of μM can be translated into M^∞. □

Theorem 1.4.7 If the LTS is image-finite and $P \equiv_{\mu M} P'$ then $P \sim P'$.

Proof Because μM contains M this follows directly from Theorem 1.2.4. □

Is image-finiteness still necessary in Theorem 1.4.7? In Exercise 1.2.5 the relationship between stratified bisimilarity, $\sim_n$, and formulas of M with modal depth n is explored. It is possible $P \not\sim Q$ but $P \sim_n Q$ for all $n \geq 0$ and so, $P \equiv_M Q$. For instance, let P be $\sum_{i\geq 0} P_i$ and $Q = P + R$ where $P_{j+1} \xrightarrow{a} P_j$, P_0 has no a transitions and $R \xrightarrow{a} R$. Unlike P, Q has an infinite sequence of a transitions: so, $P \not\equiv_{\mu M} Q$ (because $P \models \mu X.\,[a]X$). So, a more sophisticated example is needed for the presence of image-finiteness.

Example 1.4.8 The following example is from [BS07]. It uses a key property of μM, "the finite model property": if $P \models \phi$ then there is a finite LTS and a P' within it with $P' \models \phi$. Let $\phi_1, \phi_2, \ldots$ be an enumeration of all closed μM formulas over the finite label set $\{a, b\}$ that are true at some state of some LTS. Let Pr_i, with initial state P_i, be a finite LTS such that $P_i \models \phi_i$, with all Pr_i disjoint. Let Pr_0 be constructed by taking an initial state P_0 and making $P_0 \xrightarrow{a} P_i$ for all $i > 0$. Similarly, let Pr'_0 be constructed from initial state P'_0 with transitions $P'_0 \xrightarrow{a} P_i$ for all $i > 0$ and $P'_0 \xrightarrow{a} P'_0$. Clearly, $P'_0 \not\sim P_0$ because in Pr'_0 it is possible to defer indefinitely the choice of which Pr_i to enter. On the other hand, suppose that ψ is a closed μM formula, and w.l.o.g. assume the topmost operator is a modality. If the modality is $[b]$, ψ is true of both P_0 and P'_0; if it is $\langle b\rangle$, ψ is false of both; if ψ is $\langle a\rangle\psi'$, then ψ is false at both P_0 and P'_0 iff ψ' is unsatisfiable, and true at both otherwise; if ψ is $[a]\psi'$, then ψ is true at both P_0 and P'_0 iff ψ' is valid, and false at both otherwise. Consequently, $P_0 \equiv_{\mu M} P'_0$. □

Definition 1.2.7 can be extended to μM formulas: $P \in Pr$ has the extended Hennessy-Milner property provided that if $P \equiv_{\mu M} P'$ then $P' \sim P$. Little is known about this property except that, if P has the Hennessy-Milner property then it also has the extended Hennessy-Milner property.

Exercise 1.4.9 In Exercise 1.2.8 a modally saturated LTS was defined. This notion does not readily extend to μM formulas. A set of μM formulas is *unsatisfiable* if there is not a LTS and a process P belonging to it such that P satisfies every formula in the set. Show that there is an unsatisfiable set $\Phi \subseteq \mu M$ such that every finite subset $\Phi' \subseteq \Phi$ is satisfiable. Show that this is equivalent to showing that μM fails the compactness theorem. □

Another indication that μM is more expressive than M is that it contains characteristic formulas with respect to bisimilarity for finite-state processes. So, the restriction to *acyclic* LTSs in Proposition 1.2.9 can be relaxed.

Proposition 1.4.10 Assume $(Pr, Act, \rightarrow)$ where Pr, Act and $Prop$ are finite. If $P \in Pr$ then there is a formula $\phi \in \mu M$ that is characteristic for P.

Proof Let $(Pr, Act, \rightarrow)$ be a LTS with finite sets Act, $Prop$ and Pr. Assume we want to define a characteristic formula for $P \in Pr$. Let $P_1, \ldots, P_n$ be the distinct elements of Pr with $P = P_1$ and let $X_1, \ldots, X_n$ be distinct propositional variables. We define a "modal equation" $X_i = \phi_i(X_1, \ldots, X_n)$ for each i which captures the behaviour of P_i.

$$\begin{array}{lcl} X(i) & = & \mathrm{PROP}(P_i) \wedge \bigwedge\{\mathrm{MOD}'(a, P) \mid a \in Act\} \text{ where} \\ \mathrm{PROP}(P_i) & = & \bigwedge\{p \in Prop \mid P \models p\} \wedge \bigwedge\{\neg p \in Prop \mid P \not\models p\} \\ \mathrm{MOD}'(a, P_i) & = & \bigwedge\{\langle a\rangle X_j \mid P_i \xrightarrow{a} P_j\} \wedge [a] \bigvee\{X_j \mid P_i \xrightarrow{a} P_j\} \end{array}$$

where as usual $\bigwedge \emptyset = \mathtt{tt}$ and $\bigvee \emptyset = \mathtt{ff}$. We now define the characteristic formula for P_1 as ψ_1 where

$$\begin{array}{lcl} \psi_n & = & \nu X_n.\, \phi_n(X_1, \ldots, X_n) \\ \vdots & & \vdots \\ \psi_j & = & \nu X_j.\, \phi_j(X_1, \ldots, X_j, \psi_{j+1}, \ldots, \psi_n) \\ \vdots & & \vdots \\ \psi_1 & = & \nu X_1.\, \phi_1(X_1, \psi_2, \ldots, \psi_n) \end{array}$$

The proof that ψ_1 is characteristic for P is left as an exercise for the reader. □

Example 1.4.11 Let R_1, R_2 and R_3 be the processes in Figure 1.1 and assume $Prop = \emptyset$. The modal equations are as follows.

$$\begin{array}{lclcl} X_1 & = & \phi_1(X_1, X_2, X_3) & = & (\langle a\rangle X_2 \wedge \langle a\rangle X_3) \wedge [a](X_2 \vee X_3) \wedge [b]\mathtt{ff} \wedge [c]\mathtt{ff} \\ X_2 & = & \phi_2(X_1, X_2, X_3) & = & [a]\mathtt{ff} \wedge \langle b\rangle X_3 \wedge [b]X_3 \wedge [c]\mathtt{ff} \\ X_3 & = & \phi_3(X_1, X_2, X_3) & = & [a]\mathtt{ff} \wedge [b]\mathtt{ff} \wedge \langle c\rangle X_1 \wedge \langle c\rangle X_2 \wedge [c](X_1 \vee X_2) \end{array}$$

So, ψ_3 is $\nu X_3.\, \phi_3(X_1, X_2, X_3)$, and ψ_2 is $\nu X_2.\, \phi_2(X_1, X_2, \psi_3)$ and ψ_1 is the following formula

$$\nu X_1.\, (\langle a\rangle \psi_2 \wedge \langle a\rangle \psi_3) \wedge [a](\psi_2 \vee \psi_3) \wedge [b]\mathtt{ff} \wedge [c]\mathtt{ff}$$

The reader can check that $S_1 \models \psi_1$ where S_1 is also in Figure 1.1. □

Exercise 1.4.12 Provide a characteristic formula for P_1 of Figure 1.3 and show that Q_1 in the same figure satisfies it. □

The proof of Proposition 1.4.10 shows that a characteristic formula for a finite state process only uses greatest fixpoints. Furthermore, there is a more succinct representation if simultaneous fixpoints are allowed[1]. One application of characteristic formulas is the reduction of equivalence checking (whether two given processes are equivalent) to model checking (whether a given process has a given property). This is especially useful in the case when only one of the two given processes is finite state, see [KJ06] for a survey of known results which also covers weak bisimilarity and preorder checking.

A simple corollary of Theorem 1.4.6 is that μM has the tree model property. If a μM formula has a model, it has a model that is a tree. Just 0-unravel, see Definition 1.3.14, the original model, thereby preserving bisimulation. This can be strengthened to the *bounded branching degree* tree model property (just cut off all the branches that are not actually required by some diamond subformula; this leaves at most (number of diamond subformulas) branches at each node).

Clearly we cannot translate μM into FOL because of the fixpoints. (See Exercise 1.4.9.) However, it can be translated into monadic second-order logic.

1.5 Monadic second-order logic and bisimulation invariance

MSO, monadic second-order logic of LTSs, extends FOL in Section 1.3 by allowing quantification over subsets of Pr. The new constructs over and above those of FOL are

$$\phi \ ::= \ X(x) \ \mid \ \ldots \ \mid \ \exists X.\, \phi$$

where X ranges over a family of monadic predicate variables, and $\exists X.\, \phi$ quantifies over such predicates. To interpret formulas with free predicate and individual variables we extend a valuation σ to include a mapping from predicate variables to sets of states. We inductively define when MSO formula ϕ is true on an LTS L with respect to a valuation σ as $\sigma \models_L \phi$, where again we drop the index L. The new clauses are as follows.

$$\begin{array}{lll} \sigma \models X(x) & \text{iff} & \sigma(x) \in \sigma(X) \\ \sigma \models \exists X.\, \phi & \text{iff} & \sigma\{S/X\} \models \phi \text{ for some } S \subseteq Pr \end{array}$$

The universal monadic quantifier, the dual of $\exists X$, is $\forall X.\, \phi = \neg \exists X \neg \phi$. Its derived semantic clause is: $\sigma \models \forall X.\, \phi$ iff $\sigma\{S/X\} \models \phi$ for all $S \subseteq Pr$.

Example 1.5.1 Given a LTS with $Act = \{a\}$ the property that it is three

[1] Instead of defining ψ_i iteratively in the proof of Proposition 1.4.10, they are defined at the same time in a vectorial form.

colourable is expressible in MSO as follows

$$\exists X.\,\exists Y.\,\exists Z.\,\forall x.\,\phi(x,X,Y,Z) \wedge \forall y.\,\forall z.\,\psi(y,z,X,Y,Z)$$

where $\phi(x,X,Y,Z)$ expresses x has a unique colour X, Y or Z

$$(X(x) \wedge \neg Y(x) \wedge \neg Z(x)) \vee (\neg X(x) \wedge Y(x) \wedge \neg Z(x)) \vee (\neg X(x) \wedge \neg Y(x) \wedge Z(x))$$

and $\psi(y,z,X,Y,Z)$ confirms that if there is an a transition from y to z then they are not coloured the same

$$yE_a z \rightarrow \neg(X(y) \wedge X(z)) \wedge \neg(Y(y) \wedge Y(z)) \wedge \neg(Z(y) \wedge Z(z))$$

□

There is a translation of μM formulas into MSO that extends Definition 1.3.5.

Definition 1.5.2 The MSO translation of μM formulas ϕ relative to variable x is $T_x^+(\phi)$ which is defined inductively.

$$\begin{array}{lcl}
T_x^+(p) & = & p(x) \\
T_x^+(X) & = & X(x) \\
T_x^+(\texttt{tt}) & = & x = x \\
T_x^+(\neg\phi) & = & \neg\, T_x^+(\phi) \\
T_x^+(\phi_1 \vee \phi_2) & = & T_x^+(\phi_1) \vee T_x^+(\phi_2) \\
T_x^+(\langle a\rangle\phi) & = & \exists y.\, xE_a y \wedge T_y^+(\phi) \\
T_x^+(\mu X.\,\phi) & = & \forall X.\,(\forall y.\,(T_y^+(\phi) \rightarrow X(y))) \rightarrow X(x)
\end{array}$$

□

The translation of a least fixpoint formula uses quantification and implication to capture that x belongs to every pre-fixed point.

Exercise 1.5.3 For each of the following formulas ϕ, present its MSO translation $T_x^+(\phi)$.

(1) $\mu X.\,p \vee [a]X$
(2) $\mu X.\,q \vee (p \wedge \langle a\rangle X)$
(3) $\nu X.\,\neg p \wedge [a]X$
(4) $\mu X.\,\nu Y.\,(p \wedge [a]X) \vee (\neg p \wedge [a]Y)$ □

The translation of μM formulas into MSO, Definition 1.5.2, is correct.

Proposition 1.5.4 If for each variable Z, $V(Z) = \sigma(Z)$ then $P \models_V \phi$ iff $\sigma\{P/x\} \models T_x^+(\phi)$.

Proof By structural induction on $\phi \in M$. The proofs for the modal and boolean cases follow Proposition 1.3.7. There are just the two new cases. $P \models_V X$ iff $P \in V(X)$ iff $P \in \sigma(X)$ iff $\sigma\{P/x\}(x) \in \sigma\{P/x\}(X)$ iff $\sigma\{P/x\} \models T^+_x(X)$. $P \models_V \mu X.\, \phi$ iff for all S, if $\|\phi\|_{V\{S/X\}} \subseteq S$ then $P \in S$ iff for all S, if $\forall y, y \models_{V\{S/X\}} \phi$ implies $y \in S$ then $P \in S$ iff for all S, if $\forall y, \sigma\{S/X\} \models T^+_y(\phi)$ by the induction hypothesis where σ obeys that for all Z, $\sigma(Z) = V(Z)$ iff $\sigma\{P/x\} \models \forall X.\, (\forall y.\, (T^+_y(\phi) \to X(y))) \to X(x)$ iff $\sigma\{P/x\} \models T^+_x(\mu X.\, \phi)$. □

A corollary of Theorem 1.4.6 is that if ϕ is a closed μM formula then the MSO formula $\psi(x) = T^+_x(\phi)$ with one free variable is bisimulation invariant. As with FOL there are formulas of MSO which are not bisimulation invariant. Therefore, it is natural to ask the question whether van Benthem's theorem, Proposition 1.3.12, can be extended to MSO formulas. The following result was shown by Janin and Walukiewicz [JW96].

Proposition 1.5.5 A MSO formula $\phi(x)$ is equivalent to a closed μM formula iff $\phi(x)$ is bisimulation invariant.

However, its proof utilises automata (and games) which we shall provide a flavour of.

The aim is now to think of a *different* characterisation of logics on LTSs using automata or games which operate *locally* on the LTS. A particular logical formula of MSO or μM can only mention finitely many different elements of *Prop* and finitely many different elements of *Act*; therefore, we assume now that these sets are finite in any given LTS. They will constitute finite *alphabets* for automata; let $\Sigma_1 = \mathit{Act}$ and $\Sigma_2 = \wp \mathit{Prop}$.

Let us begin with the notion of an automaton familiar from introductory computer science courses.

Definition 1.5.6 An automaton $A = (S, \Sigma, \delta, s_0, F)$ consists of a finite set of states S, a finite alphabet Σ, a transition function δ, an initial state $s_0 \in S$ and an acceptance condition F.

Traditionally, A does not operate on LTSs but on words, recognizing a language, a subset of Σ^*. Assuming A is nondeterministic, its transition function $\delta : S \times \Sigma \to \wp S$. Given a word $w = a_1 \dots a_n \in \Sigma^*$, a *run* of A on w is a sequence of states $s_0 \dots s_n$ that traverses w, so $s_{i+1} \in \delta(s_i, a_{i+1})$ for each $i : 0 \le i < n$. The run is *accepting* if the sequence $s_0 \dots s_n$ obeys F: classically, $F \subseteq S$ is the subset of accepting states and $s_0 \dots s_n$ is accepting if the last state $s_n \in F$. There may be many different runs of A on w, some accepting the others rejecting, or no runs at all. The language *recognized* by A is the set of words for which there is at least one accepting run.

Example 1.5.7 Let $A = (\{s_0, s_1\}, \{a\}, \delta, s_0, \{s_0\})$ with $\delta(s_0, a) = \{s_1\}$ and $\delta(s_1, a) = \{s_0\}$. The language accepted by A is the set $\{a^{2n} \mid n \geq 0\}$ of even length words. $\square$

A simple extension is recognition of infinite length words. A run of A on $w = a_1 \ldots a_i \ldots$ is an infinite sequence of states $\pi = s_0 \ldots s_i \ldots$ that travels over w, so $s_{i+1} \in \delta(s_i, a_{i+1})$, for all $i \geq 0$; it is accepting if it obeys the condition F. Let $\mathit{inf}(\pi) \subseteq S$ contain exactly the states that occur infinitely often in π. Classically, $F \subseteq Q$ and π is accepting if $\mathit{inf}(\pi) \cap F \neq \emptyset$ which is the Büchi acceptance condition.

Büchi automata are an alternative notation for characterizing infinite paths of a LTS. There are different choices according to the alphabet Σ. If $\Sigma = \Sigma_1$ and $\pi = P_0 \xrightarrow{a_1} P_1 \xrightarrow{a_2} \ldots$ is an infinite sequence of transitions, then $\pi \models A$ if the automaton accepts the word $a_1 a_2 \ldots$; alternatively, $\Sigma = \Sigma_2$ and $\pi \models A$ if it accepts $\mathit{Prop}(P_0)\, \mathit{Prop}(P_1) \ldots$ where $\mathit{Prop}(P)$ is the subset of Prop that is true at P.

Exercise 1.5.8 Let $\mathit{Prop} = \{p\}$, $S = \{s, t\}$, $\delta(s, \{p\}) = \{t\}$, $\delta(s, \emptyset) = \{s\}$, $\delta(t, \{p\}) = \{t\}$ and $\delta(t, \emptyset) = \{t\}$, $s_0 = s$ and $F = \{t\}$. What property of an infinite run of a LTS does this Büchi automaton express? $\square$

When each formula of a logic is equivalent to an automaton, satisfiability checking reduces to the *non-emptiness* problem for those automata: whether an automaton accepts *some* word (path or whatever). This may have algorithmic benefits in reducing an apparently complex satisfiability question into simple graph-theoretic procedures: a Büchi automaton, for instance, is non-empty if there is a path $s_0 \longrightarrow^* s \in F$ and a cycle $s \longrightarrow^* s$ (equivalent to an eventually cyclic model). Indeed the introduction of Büchi and Rabin automata was for showing decidability of monadic second-order theories by reducing them to automata, see the tutorial text [GTW02] for details.

The idea of recognizing bounded branching *trees* extends the definition of A to accept n-branching infinite trees. With a word automaton, a state s' belonged to $\delta(s, a)$; now it is tuples $(s'_1, \ldots, s'_n)$ that belong to $\delta(s, a)$. A tree automaton traverses the tree, descending from a node to all n-child nodes, so the automaton splits itself into n copies, and proceeds independently. A run of the automaton is then an n-branching infinite tree labelled with states of the automaton. A run is accepting if *every* path through this tree satisfies the acceptance condition F. In the case of Rabin acceptance $F = \{(G_1, R_1), \ldots, (G_k, R_k)\}$ where each $G_i, R_i \subseteq S$ and π obeys F if there is a j such that $\mathit{inf}(\pi) \cap G_j \neq \emptyset$ and $\mathit{inf}(\pi) \cap R_j = \emptyset$. A variant definition is *parity* acceptance where F maps each state s of the automaton to a *priority* $F(s) \in \mathbb{N}$. We say that a path satisfies F if the least priority seen infinitely often is even. It is not hard to see that a parity condition is a special case of a Rabin condition; it is also true, though somewhat trickier, that a Rabin automaton can be translated to an equivalent parity automaton. Such automata can recognize bounded branching unravellings of LTSs.

Exercise 1.5.9 Tree automata characterize rooted n-branching infinite tree LTS models for μM formulas. Such a model $L \models A$ if A accepts the behaviour tree that replaces each state $P \in \mathit{Pr}$ with $\mathit{Prop}(P)$. Let $\mathit{Prop} = \{p\}$, $S = \{s, t\}$, $\delta(s, \{p\}) = \{(s, s)\}$, $\delta(s, \emptyset) = \{(t, t)\}$, $\delta(t, \{p\}) = \{(s, s)\}$ and $\delta(t, \emptyset) = \{(t, t)\}$ and $s_0 = s$. This automaton A has parity acceptance condition $F(s) = 1$ and $F(t) = 2$. What μM formula is equivalent to A over infinite binary-tree models? (Hint: what fixpoints are "coded" by states s and t?) $\Box$

There is a slight mismatch between (the unravellings of) LTSs and bounded branching trees because of the fixed branching degree and the explicit indexed successors; for instance, see the unravelled LTS of Figure 1.4. What is wanted is an automaton that can directly recognize a LTS and which preserves the virtue of a simple *local* definition of a transition function. We shall define a variant of alternating parity automata which is due to Walukiewicz (also see [KVW00]).

The range of a transition function of an automaton A will be a *local formula*. For a word automaton, if $\delta(s, a) = \{s_1, \ldots, s_m\}$ then it is the formula $s_1 \vee \ldots \vee s_m$. For a n-branching tree automaton if $\delta(s, a) = \{(s^1_1, \ldots, s^1_n), \ldots, (s^m_1, \ldots, s^m_n)\}$ then it is $((1, s^1_1) \wedge \ldots \wedge (n, s^1_n)) \vee \ldots \vee ((1, s^m_1) \wedge \ldots \wedge (n, s^m_n))$: here the element (i, s') means create an *ith*-child with label s'. A word or tree is accepted if there exists an accepting run for that word or tree; hence, the disjuncts. However, for a tree, every path through it must be accepting; hence the conjuncts. In *alternating* word automata, the transition function is given as an arbitrary boolean expression over states: for instance, $\delta(s, a) = s_1 \wedge (s_2 \vee s_3)$. In alternating tree automata it is a boolean expression over directions and states: for instance, $((1, s_1) \wedge (1, s_2)) \vee (2, s_3)$. Now the definition of a run becomes a tree in which, successor transitions obey the boolean formula. In particular, even for an alternating automaton on words, a run is a tree, and not just a word. The acceptance criterion is as before, that every path of the run must be accepting. An alternating automaton is just a two player game too where one player $\forall$ is responsible for $\wedge$ choices and the other player $\exists$ for $\vee$ choices.

The transition function for an automaton A that recognises LTSs has the form $\delta : S \times \Sigma_2 \to \Phi(\Sigma_1, S)$ where $\Phi(X, Y)$ is a set of formulas over X and Y. One idea is that this formula could be a simple modal formula. For instance, if s is the current automaton state at $P \in \mathit{Pr}$ and $\delta(s, \mathrm{Prop}(P)) = \langle a \rangle s_1 \wedge [c] s_2$ and $P \xrightarrow{a} P_1$, $P \xrightarrow{b} P_2$, $P \xrightarrow{c} Q_i$, for all $i \geq 0$ then the automaton moves to P_1

with state s_1 and to each Q_i with state s_2. As with tree automata, a run of A on a LTS is a labelled tree of arbitrary degree. Such "modal" automata when the acceptance condition for infinite branches is the parity condition have the same expressive power as μM.

However, to prove Proposition 1.5.5 Janin and Walukiewiciz use FOL formulas. The idea for atomic predicates is to replace pairs (i, s) of a tree automaton with elements of $U = \{(a, s) \mid a \in \Sigma_1 \text{ and } s \in S\}$. Now, for each $s \in S$ and $W \subseteq \mathit{Prop}$, $\delta(s, W)$ is a formula of the form

$$(*)\ \exists x_1 \ldots \exists x_k.\, (u_1(x_1) \wedge \ldots \wedge u_k(x_k)) \wedge \forall x.\, (u_1(x) \vee \ldots \vee u_k(x))$$

where each $u_i \in U$. An example, is $\phi = \exists x_1.\, \exists x_2.\, (a, s)(x_1) \wedge (b, s')(x_2) \wedge \forall x.\, (a, s)(x) \vee (b, s')(x)$. If t labels the state P of the LTS and $W = \mathit{Prop}(P)$ and $\delta(t, W) = \phi$ and $P \xrightarrow{a} P_i$, $P \xrightarrow{b} Q_j$, $i, j > 0$ then the automaton at the next step would spawn a copy at each P_i with state s and each Q_j with state s'. Notice that such a formula is quite similar to the components $\bigwedge \mathrm{MOD}'(a, P)$ of a characteristic formula described in Proposition 1.4.10. Every μM formula is equivalent to such an automaton; the different kinds of fixpoint are catered for in the parity acceptance condition.

Let $\mathit{dis}(x_1, \ldots, x_n)$ be the FOL formula $\bigwedge_{1 \leq i < j \leq n} x_i \neq x_j$. There is a very similar characterization of MSO formulas over trees. where now each $\delta(s, W)$ has the form

$$(**)\ \exists x_1 \ldots \exists x_k.\, (D \wedge u_1(x_1) \wedge \ldots \wedge u_k(x_k)) \wedge \forall x.\, D' \rightarrow (u_1(x) \vee \ldots \vee u_k(x))$$

where $D = \mathit{dis}(x_1, \ldots, x_k)$ and $D' = \mathit{dis}(x, x_1, \ldots, x_k)$.

Now the result follows: if $\phi(x)$ is an MSO formula that is bisimulation invariant then it is true on any n-unravelled model and so (*) and (**) will be equivalent for $n \geq k$.

Bibliography

[vB84] J. van Benthem. Correspondence theory. In *Handbook of Philosophical Logic*, Vol. II, ed. D. Gabbay and F. Guenthner, 167-248, Reidel, (1984).

[vB96] J. van Benthem. *Exploring Logical Dynamics.* CSLI Publications, (1996).

[vB98] J. van Benthem. Program constructions that are safe for bisimulation, *Studia Logica*, **60** 311–330 (1998).

[BRV01] P. Blackburn, M. de Rijke and Y. Venema. *Modal Logic*, Cambridge University Press, (2001).

[BS07] J. Bradfield and C. Stirling, Modal mu-calculi. In *Handbook of Modal Logic*, ed. P. Blackburn, J. van Benthem and F. Wolter, Elsevier, 721–756, (2007).

[Ch80] B. Chellas, *Modal Logic: An Introduction.* Cambridge University Press (1980).

[GTW02] E. Grädel, W. Thomas and T. Wilke (Eds.), *Automata, Logics, and Infinite Games*, *Lecture Notes in Computer Science* **2500** (2002).

[HM80] M. Hennessy and R. Milner, On observing nondeterminism and concurrency, *Lecture Notes in Computer Science* **85** 295–309 (1980).

[HM85] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of Association of Computer Machinery* **32** 137–162, 1985.

[JW96] D. Janin and I. Walukiewicz, On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic. *Lecture Notes in Computer Science* **1119** 263–277 (1996).

[Ko83] D. Kozen, Results on the propositional mu-calculus. *Theoretical Computer Science* **27** 333–354 (1983).

[KJ06] A. Kučera and P. Jančar, Equivalence-checking on infinite-state systems: techniques and results. *Theory and Practice of Logic Programming* **6**(3), 227–264 (2006).

[KVW00] O. Kupferman, M. Vardi and P. Wolper, An automata-theoretic approach to branching-time model checking, *Journal of Association of Computer Machinery* **42**(2) 312–360 (2000).

[Pa81] D. Park. Concurrency and automata on infinite sequences. *Lecture Notes in Computer Science* **154** 561–572 (1981).