12 Expressive completeness

In this chapter we compare the expressive power of the modal μ-calculus to that of monadic second-order logic. The key result that we will prove is that the modal μ-calculus has the same expressive power as the bisimulation invariant fragment of monadic second-order logic, in brief:

$$\mu\mathtt{ML} \equiv \mathtt{MSO}/\leftrightarroweq. \tag{129}$$

In fact, Theorem 12.21, the actual result that we are going to prove is a bit stronger than (129).

Our proof will be automata-theoretic in nature: after discussing two different (but equivalent) versions of monadic second-order logic in section 12.1, we show in section 12.2 that on tree models, *MSO* has the same expressive power as the class $Aut(\mathtt{1FOE})$ of automata over the one-step logic 1FOE. Since the modal μ-calculus corresponds to the class $Aut(\mathtt{1FO})$, we will prove (129) in section 12.3 via a comparison of the one-step languages 1FOE and 1FO.

12.1 Monadic second-order logic

Second-order logic is the extension of first-order logic where quantification is allowed, not only over individuals, but also over relations on the domain. In *monadic* second-order logic, this second-order quantification is restricted to unary relations, that is, subsets of the domain. The syntax of monadic second-order logic is usually defined as the extension of that of first-order logic by second-order quantifiers of the form $\exists p/\forall p$, where p is a monadic predicate symbol.

Definition 12.1 Given a set D of atomic actions, a set IVar of individual variables and a set Prop of set variables, we define the language $\mathtt{MSO}^2_{\mathsf{D}}$ as follows:

$$\varphi ::= x \doteq y \mid R_d xy \mid p(x) \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x.\varphi \mid \exists p.\varphi$$

Here x and y are variables from IVar, p is a variable from P, and $d \in \mathsf{D}$ is an atomic action.

We let $\mathtt{MSO}^2_{\mathsf{D}}(\mathsf{X}, \mathsf{P})$ denote the set of $\mathtt{MSO}^2_{\mathsf{D}}$-formulas φ of which all individual free variables are from X and all free set variables are from P. In case X is a singleton $\{x\}$, we write $\mathtt{MSO}^2_{\mathsf{D}}(x, \mathsf{P})$ rather than $\mathtt{MSO}^2_{\mathsf{D}}(\{x\}, \mathsf{P})$ $\triangleleft$

This semantics of this language is completely standard, with $\exists x$ denoting first-order quantification (that is, quantification over individual states), and $\exists p$ denoting monadic second-order quantification (that is, quantification over sets of states).

It turns out, however, that for a nice inductive translation of *MSO* to automata, it is more convenient to use a slightly nonstandard version of *MSO* that is *single-sorted* in that it only admits second-order variables, not first-order ones. Quantification over individuals can then be simulated by quantification over singleton sets. In addition, to facilitate the comparison with modal languages, which are interpreted in *pointed* Kripke models, we need to install a feature in the language that allows access to the designated or actual world of the Kripke model.

Definition 12.2 Given a set D of atomic actions, we define the language of *monadic second-order logic* $\mathtt{MSO}_\mathsf{D}$ as follows:

$$\varphi \ ::= \ p \sqsubseteq q \ \mid \ R_d pq \ \mid \ \Downarrow p \ \mid \ \neg\varphi \ \mid \ \varphi \vee \varphi \ \mid \ \exists p.\varphi,$$

where p and q are propositional variables from P. We let $\mathtt{MSO}_\mathsf{D}(\mathsf{P})$ denote the set of $\mathtt{MSO}_\mathsf{D}$-formulas of which the free variables are from P. ◁

Definition 12.3 Given a Kripke model $\mathbb{S} = \langle S, V, R\rangle$, and a designated point $s \in S$, we define the semantics of $\mathtt{MSO}$ as follows:

$$\begin{array}{lll}
\mathbb{S}, s \models p \sqsubseteq q & \text{if} & V(p) \subseteq V(q) \\
\mathbb{S}, s \models R_d pq & \text{if} & \text{for all } t \in V(p) \text{ there is a } u \in V(q) \text{ with } R_d tu \\
\mathbb{S}, s \models \Downarrow p & \text{if} & V(p) = \{s\} \\
\mathbb{S}, s \models \neg\varphi & \text{if} & \mathbb{S}, s \not\models \varphi \\
\mathbb{S}, s \models \varphi \vee \psi & \text{if} & \mathbb{S}, s \models \varphi \text{ or } \mathbb{S}, s \models \psi \\
\mathbb{S}, s \models \exists p.\varphi & \text{if} & \mathbb{S}[p \mapsto X], s \models \varphi \text{ for some } X \subseteq S.
\end{array}$$

An $\mathtt{MSO}$-formula φ is *bisimulation invariant* if $\mathbb{S}, s \leftrightarrows \mathbb{S}', s'$ implies that $\mathbb{S}, s \models \varphi \Leftrightarrow \mathbb{S}', s \models \varphi$. ◁

Remark 12.4 In fact, one may think of the formalism as a *first-order* logic of which the intended models are *power structures* of the form $\langle \wp(S), \subseteq, \vec{R}, \{s\}\rangle$, where $R_d(Y, Z)$ iff for all $y \in Y$ there is a $z \in Z$ such that $R_d yz$.) ◁

It is not too hard to see that the two languages are in fact equivalent.

Theorem 12.5 *There are effective procedures transforming a formula in* $\mathtt{MSO}^2(x, \mathsf{P})$ *into an equivalent* $\mathtt{MSO}(\mathsf{P})$*-formula, and vice versa:*

$$\mathtt{MSO}^2 \equiv \mathtt{MSO}.$$

To start with, there is a straightforward, inductively defined translation $(\cdot)' : \mathtt{MSO}_\mathsf{D}(\mathsf{P}) \to \mathtt{MSO}^2_\mathsf{D}(x, \mathsf{P})$ such that

$$\mathbb{S}, s \models \varphi \text{ iff } \mathbb{S} \models \varphi'[s],$$

for all formulas $\varphi \in \mathtt{MSO}_\mathsf{D}(\mathsf{P})$ and all pointed Kripke models $\mathbb{S}$. The only interesting clause in the inductive definition of this translation concerns the $\Downarrow$-connective, for which we set

$$(\Downarrow p)' := \forall y (p(y) \leftrightarrow y \doteq x).$$

For the opposite direction, the key observation is that $\mathtt{MSO}$ can interpret $\mathtt{MSO}^2$ by encoding individual variables as set variables denoting *singletons*. To understand how this works, we need to have a closer look at the semantics. Formulas of the language $\mathtt{MSO}^2$ are interpreted over Kripke models $\mathbb{S}$ with an *assignment*, that is, a map $\alpha : \mathsf{IVar} \to S$ interpreting the individual variables as elements of S. But then we can encode such an $\mathtt{MSO}^2$-model $\mathbb{S} = (S, R, V)$ with assignment α, as the $\mathtt{MSO}$-model $\mathbb{S}^\alpha := (M, R, V^\alpha)$ over $\mathsf{Prop} \cup \mathsf{IVar}$, where $V^\alpha(p) := V(p)$ if p is a set variabe, and $V^\alpha(x) := \{\alpha(x)\}$ if x is an individual variable.

Proposition 12.6 *There is a translation* $(\cdot)^t : \mathtt{MSO}^2_{\mathsf{D}}(\mathsf{X},\mathsf{P}) \to \mathtt{MSO}_{\mathsf{D}}(\mathsf{P} \uplus \mathsf{X})$ *such that*

$$\mathbb{S} \models \varphi[\alpha] \textit{ iff } \mathbb{S}^{\alpha} \models \varphi^t \tag{130}$$

for all $\varphi \in \mathtt{MSO}^2_{\mathsf{D}}(\mathsf{X},\mathsf{P})$*, all Kripke models* $\mathbb{S} = (S, R, V)$ *and all assignments* $\alpha : \mathsf{X} \to S$*.*

As a corollary, for all $\varphi \in \mathtt{MSO}^2_{\mathsf{D}}(x,\mathsf{P})$ *and all pointed Kripke models* $(\mathbb{S}, s)$ *we obtain*

$$\mathbb{S} \models \varphi[s] \textit{ iff } \mathbb{S}, s \models \forall x.(\Downarrow x \to \varphi^t). \tag{131}$$

Proof. The translation crucially involves the MSO-formulas $\mathtt{empty}(p)$ and $\mathtt{sing}(p)$ given by

$$\begin{array}{lcl} \mathtt{empty}(p) & := & \forall q\,(p \sqsubseteq q) \\ \mathtt{sing}(p) & := & \forall q\,\big(q \sqsubseteq p \to (\mathtt{empty}(q) \lor p \sqsubseteq q)\big). \end{array}$$

It is not hard to prove that these formulas hold in $\mathbb{S}$ iff, respectively, $V(p)$ is empty and $V(p)$ is a singleton.

With these formulas defined, we can now inductively fix the translation as follows:

$$\begin{array}{lcl} (p(x))^t & := & x \sqsubseteq p \\ (R_d xy)^t & := & R_d xy \\ (x \doteq y)^t & := & x \sqsubseteq y \land y \sqsubseteq x \\ (\neg\varphi^t & := & \neg\varphi^t \\ (\varphi_0 \lor \varphi_1)^t & := & \varphi_0^t \lor \varphi_1^t \\ (\exists x.\varphi)^t & := & \exists x.(\mathtt{sing}(x) \land \varphi^t) \\ (\exists p.\varphi)^t & := & \exists p.\varphi^t \end{array}$$

It is a routine exercise to verify (130), so we leave the details for the reader. Similarly, the proof of (131) is immediate by (130) and the definitions of the semantics of $\Downarrow$. QED

Note that the translation $(\cdot)^t$ given in the proof of Proposition 12.6 does not involve the connective $\Downarrow$. The only use of $\Downarrow$ in this setting is to mark the designated node of a *pointed* Kripke model.

12.2 Automata for monadic second-order logic

The aim of this section is to provide an automata-theoretic perspective on monadic second-order logic. That is, we will provide a construction transforming an arbitrary MSO-formula φ into an automaton $\mathbb{B}_\varphi$ that is equivalent to φ, at least, if we confine attention to tree models. In fact, we will encounter various kinds of automata, all corresponding to MSO-formulas, and all taking some fragment of *monadic first-order logic* as the co-domain of their transition map, as in Definition 10.26 and Definition 10.27.

Recall that the set $\mathtt{MFOE}(A)$ of *monadic first-order formulas over A* is given by the following grammar:

$$\alpha \;::=\; \top \mid \bot \mid a(x) \mid \neg a(x) \mid x \doteq y \mid x \not\doteq y \mid \alpha \lor \alpha \mid \alpha \land \alpha \mid \exists x.\alpha \mid \forall x.\alpha$$

where $a \in A$ and x, y are first-order (individual) variables, and that $\mathtt{MFO}(A)$ is the set of $\mathtt{MFOE}(A)$-formulas without occurrences of identity formulas (or their negations). Recall as

Position	Player	Admissible moves
$(a,s) \in A \times S$	$\exists$	$\{U : A \to \wp(R(s)) \mid (R(s),U) \models \Theta(a,\sigma_V(s))\}$
$U : A \to \wp(S)$	$\forall$	$\{(b,t) \mid t \in U(b)\}$

Table 26: Acceptance game for *MSO*-automata

well that $\mathtt{1FOE}(A)$ and $\mathtt{1FO}(A)$ are the one-step languages consisting of the sentences of, respectively, $\mathtt{MFOE}(A)$ and $\mathtt{MFO}(A)$, where each monadic predicate $a \in A$ occurs only positively. It will be convenient in this section to present one-step models using valuations rather than markings; that is, a *one-step model* will be denoted as a pair (Y,V) consisting of some set Y and an A-valuation $V : A \to \wp(Y)$.

Definition 12.7 An *MSO-automaton* over a set P of proposition letters is nothing but a $\mathtt{1FOE}$-automaton over P, that is, a quadruple $\mathbb{A} = \langle A, \Theta, \Omega, a_I\rangle$, where A, a_I and Ω are as usual, and Θ is a map $\Theta : A \times \wp(\mathsf{P}) \to \mathtt{1FOE}(A)$.

The acceptance game of such an automaton with respect to a Kripke model $\mathbb{S}$ is given in Table 12.2. The winning conditions for both finite and infinite matches are as usual. $\lhd$

In words, the acceptance game proceeds as follows. At a basic position (a,s), $\exists$ chooses a valuation U interpreting each 'predicate' $a \in A$ as a subset $U(a)$ of the set $R(s)$ of successors of s. In this choice, she is bound by the condition that the sentence $\Theta(a,\sigma_V(s))$ must be *true* in the resulting A-structure $(R(s),U)$. Once chosen, this map U itself determines the next position of the match. As a position, U belongs to player $\forall$, and all he has to do is to choose a pair (b,t) such that $t \in U(b)$. This pair (b,t) is then the next basic position of the match.

The link with modal automata is given by Proposition 10.31, stating that, seen as one-step languages, $\mathtt{1FO}$ is equivalent to $\mathtt{1ML}$. From this we obtain the equivalence in expressive power of the automata classes $Aut(\mathtt{1FO})$ and $Aut(\mathtt{1ML})$, which in its turn entails the following.

Theorem 12.8 *There are effective procedures transforming a μ-calculus formula into an equivalent MSO-automaton in $Aut(\mathtt{1FO})$, and vice versa:*

$$\mu\mathtt{ML} \equiv Aut(\mathtt{1FO}).$$

The main result of this section states a very similar result for $\mathtt{MSO}$ and arbitrary *MSO*-automata, if we confine our attention to *tree models*:

Theorem 12.9 *There are effective procedures transforming an $\mathtt{MSO}$-formula ξ into an MSO-automaton $\mathbb{A}$, and vice versa, such that the corresponding formula ξ and automaton $\mathbb{A}$ are equivalent on the class of tree models:*

$$\mathtt{MSO} \equiv Aut(\mathtt{1FOE}) \text{ (on tree models).}$$

Note that on *arbitrary* models, monadic second-order logic can express properties that cannot be captured by *MSO*-automata. For instance, it is easy to write an $\mathtt{MSO}$-formula stating that the designated point of a Kripke model lies on a cycle, but there is no *MSO*-automaton that recognizes exactly the class of pointed Kripke models with this property.

We will prove the two directions in the statement of Theorem 12.9 separately. Leaving the transformation of automata to monadic second-order formulas to the end of the section, we first concentrate on the opposite direction.

Proposition 12.10 *There is an effective procedure transforming a formula* $\varphi \in \mathtt{MSO}(\mathsf{P})$ *into an MSO-automaton* $\mathbb{B}_\varphi$ *over* P *that is equivalent to* φ *over the class of tree models. That is:*

$$\mathbb{S}, r \models \varphi \textit{ iff } \mathbb{B}_\varphi \textit{ accepts } (\mathbb{S}, r). \tag{132}$$

for any tree model $\mathbb{S}$ *with root* r*.*

We will prove Proposition 12.10 by induction on the complexity of *MSO*-formulas. The proposition below takes care of the atomic case.

Proposition 12.11 *Let* φ *be one of the atomic MSO-formulas:* Rpq*,* $p \sqsubseteq q$*, or* $\Downarrow p$*. Then there is an MSO-automaton* $\mathbb{B}_\varphi$ *that is equivalent to* φ *on tree models.*

Proof. We restrict attention to the formula *Rpq*, leaving the other cases as an exercise for the reader. The automaton $\mathbb{B}_{Rpq}$ is defined as the structure $(\{a_0, a_1\}, \Theta, \Omega, a_0)$, where Θ is given by putting:

$$\begin{array}{lcl} \Theta(a_0, c) & := & \begin{cases} \exists y \left(a_1(y) \land \forall z \left(z \neq y \to a_0(z)\right)\right) & \text{if } p \in c \\ \forall z \, a_0(z) & \text{otherwise} \end{cases} \\ \Theta(a_1, c) & := & \begin{cases} \bot & \text{if } q \notin c \\ \exists y \left(a_1(y) \land \forall z \left(z \neq y \to a_0(z)\right)\right) & \text{if } q \in c \text{ and } p \in c \\ \forall z \, a_0(z) & \text{otherwise} \end{cases} \end{array}$$

Furthermore, Ω is defined via $\Omega(a_i) := 0$ for each a_i — as a consequence, $\exists$ wins all infinite games. We leave it for the reader to verify that this automaton is of the right shape, and that it is indeed equivalent to the formula *Rpq* on tree models. QED

For the inductive step of the argument, there are three cases to consider, corresponding to, respectively, the connectives $\lor$ and $\neg$, and the (second-order) existential quantification. It turns out that the first two cases are relatively easy to handle, cf. Proposition 10.38. To take care of the existential quantification however, we need to work with *nondeterministic* automata, in which every formula $\Theta(a, c)$ has been brought into a certain normal form. Fortunately, we can prove a simulation theorem for *MSO*-automata, implying that we may transform any *MSO*-automaton into an equivalent nondeterministic one. We need some definitions on these normal forms of `1FOE`-formulas.

Definition 12.12 Fix a set A of propositional variables. We introduce some abbreviations for `MFOE`-formulas:

$$\mathtt{diff}(y_1, \ldots, y_n) := \bigwedge \{y_i \neq y_j \mid 1 \leq i < j \leq n\},$$

and, for a set $B \subseteq A$:

$$\tau_B(x) := \bigwedge_{a \in B} a(x).$$

Now define the following MFOE-sentences:

$$\begin{array}{lll}\chi^{=}_{\overline{B},\overline{C}} & := & \exists y_1 \cdots y_n \Big(\mathtt{diff}(\overline{y}) \wedge \bigwedge_i \tau_{B_i}(y_i) \wedge \forall z\, (\mathtt{diff}(\overline{y}, z) \to \bigvee_j \tau_{C_j}(z))\Big) \\ \chi_{\overline{B},\overline{C}} & := & \exists y_1 \cdots y_n \Big(\bigwedge_i \tau_{B_i}(y_i) \wedge \forall z \bigvee_j \tau_{C_j}(z)\Big)\end{array}$$

where $\overline{B} = B_1, \ldots, B_n$ and $\overline{C} = C_1, \ldots, C_m$ are two sequences of subsets of A.

Sentences of the form $\chi^{=}(\overline{B}, \overline{C})$ are said to be in *basic form*, and in *special basic form* in case each B_i and C_j is a singleton. The sets of these formulas are denoted as $\mathtt{BF}(A)$ and $\mathtt{SBF}(A)$, respectively. ◁

In words, the formula $\mathtt{diff}(y_1, \ldots, y_n)$ expresses that the variables $y_1, \ldots, y_n$ refer to n *distinct* objects of the domain. The formula $\tau_B(x)$ can be seen to state that x *realises* the type B, that is: it satisfies all predicates a in B. The formula $\chi^{=}_{\overline{B},\overline{C}}$ expresses the existence of n distinct objects realising the B-types, with all other objects realising one of the C-types. This formula is (equivalent to) the formula $\forall z \bigvee_j \tau_{C_j}(z)$ in the special case where $n = 0$, and, in case $m = 0$ as well, to the formula $\forall z \perp$ (which holds in the empty model only). As a simplified version of $\chi^{=}_{\overline{B},\overline{C}}$, the sentence $\chi_{\overline{B},\overline{C}}$ states that all types in $\overline{B}$ are witnessed by some object, while every object satisfies some C-type. Note that for the latter reason, $\chi_{\overline{B},\overline{C}}$ is generally not a semantics consequence of $\chi^{=}_{\overline{B},\overline{C}}$. Finally, observe that $\chi^{=}_{\overline{B},\overline{C}}$ and $\chi_{\overline{B},\overline{C}}$ are positive sentences, and hence, one-step formulas in 1FOE and 1FO, respectively.

Using these normal forms, we can now define the notion of a nondeterministic *MSO*-automaton.

Definition 12.13 An *MSO*-automaton $\mathbb{A} = \langle A, \Theta, \Omega, a_I \rangle$ is called *nondeterministic* if $\mathsf{Ran}(\Theta) \subseteq \mathtt{Dis}(\mathtt{SBF}(A))$, that is, every formula $\Theta(a, c)$ is a disjunction of special basic formulas. ◁

Nondeterministic automata are of interest because they admit *functional strategies* — in tree models, that is. As in Definition 10.16, we call a strategy f for $\exists$ in the acceptance game $\mathcal{A}(\mathbb{A}, \mathbb{S})@(a_I, r)$ *functional* if for every $s \in S$ there is at most one $a \in A$ such that the position (a, s) is reachable in an f-guided match of $\mathcal{A}(\mathbb{A}, \mathbb{S})@(a_I, r)$. In case $\exists$ has a functional strategy which is in addition winning, we write $\mathbb{S}, r \Vdash_s \mathbb{A}$. The following proposition states that on tree models, we may *always* assume that winning strategies are functional.

Proposition 12.14 *Let $\mathbb{A}$ be a nondeterministic MSO-automaton, and let $\mathbb{S}$ be a tree-based Kripke model with root r. Then $\mathbb{S}, r \Vdash \mathbb{A}$ iff $\mathbb{S}, r \Vdash_s \mathbb{A}$.*

As a corollary, nondeterministic *MSO*-automata are closed under existential second-order quantification.

Corollary 12.15 *Let $\mathbb{D} = \langle D, \Delta, \Omega, d_I \rangle$ be a nondeterministic MSO-automaton over the set $\mathsf{P} \cup \{p\}$. Then there is a nondeterministic automaton $\mathbb{D}^{\exists p}$ over P, such that for all tree models $(\mathbb{S}, r)$:*

$$\mathbb{D}^{\exists p} \text{ accepts } (\mathbb{S}, r) \text{ iff } \mathbb{D} \text{ accepts } (\mathbb{S}[p \mapsto T], r) \text{ for some } T \subseteq S. \quad (133)$$

Proof. Define the automaton $\mathbb{D}^{\exists p} := \langle D, \Delta^{\exists p}, \Omega, d_I \rangle$, with alphabet $C = \wp(\mathsf{P})$, by putting

$$\Delta^{\exists p}(a, c) := \Delta(a, c) \vee \Delta(a, c \cup \{p\}).$$

Clearly then $\mathbb{D}^{\exists p}$ is a nondeterministic *MSO*-automaton, so it remains to prove that $\mathbb{D}^{\exists p}$ satisfies (133). But since we may assume winning strategies to be functional, this proof is a variation on a proof given earlier, viz., that of Proposition 11.23. QED

But if *nondeterministic MSO*-automata admit existential second-order quantification, in order to transfer this closure property to the class of arbitrary automata, all we need is the following Simulation Theorem which states in particular that every *MSO*-automaton has a nondeterministic equivalent.

Theorem 12.16 (Simulation Theorem) *There are effective constructions transforming an automaton of any of the kinds below to an equivalent automaton of any other kind:*
1) $Aut(\texttt{1FOE})$*,*
2) $Aut(\texttt{Dis}(\texttt{BF}(A)))$*,*
3) $Aut(\texttt{Dis}(\texttt{SBF}(A)))$*.*

To prove the implication from 1) to 2) of this result, we need a model-theoretic result on monadic first-order logic, that will be of use later on as well.

Proposition 12.17 *There is an effective procedure transforming an arbitrary positive sentence in* $\texttt{MFOE}(A)$ *to an equivalent disjunction of sentences in basic form.*

The proof of this result, which we omit for the time being, is a fairly straightforward exercise in the theory of Ehrenfeucht-Fraïssé games.

Proof of Theorem 12.16. The implications from 3) to 2) and from 2) to 1) are trivial consequences of the definitions. The implication from 1) to 2) is immediate by Proposition 12.17.

The hardest part of the proof concerns the remaining implication, from 2) to 3). This, however, is an instance of the general simulation theorem that we proved in section 10.6. We only need to verify that the language $\texttt{Dis}(\texttt{SBF})$, seen as a one-step language, is $\bigwedge$-distributive over $\texttt{Dis}(\texttt{BF})$, and therefore, over $\texttt{1FOE}$, but we leave this as an exercise for the reader. QED

With this Simulation Theorem we have all the results that are needed for the inductive translation of second-order formulas to $\texttt{MSO}$-automata.

Proof of Proposition 12.10. As mentioned, the proposition is proved by induction on the complexity of $\varphi \in \texttt{MSO}$. The atomic case of the induction is covered by Proposition 12.11. For the induction step, the case where $\varphi = \exists p.\psi$ is taken care of by Theorem 12.16 and Corollary 12.15. The remaining cases, where respectively $\varphi = \neg\psi$ and $\varphi = \varphi_0 \vee \varphi_1$, are left as exercises for the reader. QED

Proposition 12.10 takes care of one direction of Theorem 12.9; for the opposite direction we need to find an equivalent formula $\xi_{\mathbb{A}} \in \texttt{MSO}$ for each *MSO*-automaton $\mathbb{A}$.

Proposition 12.18 *There is an effective procedure transforming an MSO-automaton* $\mathbb{A}$ *into a formula* $\xi_{\mathbb{A}} \in \mathtt{MSO}^2(\mathsf{P})$ *that is equivalent to* φ *over the class of tree models. That is:*

$$\mathbb{A} \textit{ accepts } (\mathbb{S}, r) \textit{ iff } \mathbb{S}, r \models \xi_{\mathbb{A}}. \tag{134}$$

for any tree model $\mathbb{S}$ *with root* r.

Proof. For the time being we confine ourselves to a proof sketch. The basic idea is to encode the operational semantics of an *MSO*-automaton in monadic second-order logic; this works for nondeterministic automata over tree models, since we can express the working of a functional strategy.

To give a bit more detail, fix an *MSO*-automaton $\mathbb{A}$. We first transform $\mathbb{A}$ into an equivalent nondeterministic automaton $\mathbb{D} = (D, \Theta, \Omega, d_I)$; this is possible by Theorem 12.16. It then suffices to write down a monadic second-order formula $\xi(x)$ in $\mathtt{MSO}^2(x)$ such that, for an arbitrary tree model $\mathbb{S}$ with root r:

$$\mathbb{S} \models \xi[r] \text{ iff } \exists \text{ has a functional positional winning strategy in } \mathcal{A}(\mathbb{A}, \mathbb{S})@(a_I, r).$$

Let $\mathbb{S} = (S, R, V)$ be a an arbitrary tree model with root r and let $D = \{a_1, \ldots, a_n\}$. Here we think of the a_i as second-order variables that will be quantified over existentially, in order to express the *existence* of a functional positional strategy. Take an arbitrary valuation $U : D \to \wp(S)$. It is easy to write down an $\mathtt{MSO}^2(x)$-formula $\varphi(\overline{a}, x)$ which holds of the resulting model $\mathbb{S} \oplus U$ iff $|U(a_i)| \leq 1$ for each i, so that we may think of the associated marking m_U as a potential functional strategy of $\exists$ in the acceptance game $\mathcal{A}(\mathbb{A}, \mathbb{S})$. Writing a_s for the unique state such that $a_s \in m_U$, we may then use the one-step formula $\Theta(a, \sigma_V(s))$ as a basis for a first-order formula which expresses that this potential strategy induced by U actually provides a legitimate move for $\exists$ at position (a_s, s). Finally, note that any infinite match of $\mathcal{A}(\mathbb{A}, \mathbb{S})@(a_I, r)$ corresponds to a branch of $\mathbb{S}$ (that is, an infinite path starting at r); using a second-order variable b to range over such branches, it is then fairly straightforward to write down a formula stating that the highest parity occurring infinitely often on any match of an m_U-guided match is even. QED

12.3 Expressive completeness modulo bisimilarity

A central result in the theory of basic modal logic states that modal logic corresponds to the bisimulation invariant fragment of first-order logic. In this section we will prove an extension of this result stating that the modal μ-calculus is the bisimulation invariant fragment of monadic *second-order* logic. While it is not difficult to show that every $\mu\mathtt{ML}$-formula is equivalent to a bisimulation-invariant formula in $\mathtt{MSO}$, it is the converse correspondence where the true importance of the result lies. We may see it as an expressive completeness result, stating that the modal μ-calculus is sufficiently strong to express *every* bisimulation-invariant formula in monadic second-order logic. Note that in a context such as process theory, where we consider bisimilar pointed Kripke models as different representations of the *same* process, bisimulation-invariant properties are in fact the *only* relevant ones. In such a situation, we may read the bisimulation-invariance result as saying that modal fixpoint logic has the same expressive power as monadic second-order logic, when it comes to expressing relevant properties.

► `Add examples of what can be expressed in MSO, and not in μML:`
`- every point has exactly two d-successors`
`- the actual state does not lie on a cycle`

We first show that there is truth-preserving translation mapping every formula of the modal μ-calculus to an equivalent monadic second-order formula. Recall from Remark 12.4 that $\mathtt{MSO}^2_{\mathsf{D}}(x, \mathsf{P})$ is the standard (two-sorted) version of monadic second-order logic.

Definition 12.19 For any individual variable x we define, by induction on the complexity of a formula $\varphi \in \mu\mathtt{ML}_{\mathsf{D}}$, a translation $\mathtt{ST}_x : \mu\mathtt{ML}_{\mathsf{D}}(\mathsf{P}) \to \mathtt{MSO}^2_{\mathsf{D}}(x, \mathsf{P})$.

$$\begin{array}{lcl}
\mathtt{ST}_x(p) & := & p(x) \\
\mathtt{ST}_x(\neg\varphi) & := & \neg\mathtt{ST}_x(\varphi) \\
STx(\Diamond_d\varphi) & := & \exists y(R_d xy \wedge \mathtt{ST}_y(\varphi)) \\
STx(\Diamond\varphi) & := & \exists y(Rxy \wedge \mathtt{ST}_y(\varphi)) \\
STx(\mu p.\varphi) & := & \exists p.\Big(p(x) \wedge \forall y.\big(p(y) \leftrightarrow \forall q.(\mathtt{PRE}(\varphi, q) \to q(y))\big)\Big),
\end{array}$$

where $\mathtt{PRE}(\varphi, q)$ abbreviates the formula $\forall y.(\mathtt{ST}_y(\varphi)[q/p] \to q(y))$. ◁

Theorem 12.20 *For any formula $\varphi \in \mu\mathtt{ML}$ we have $\varphi \equiv \mathtt{ST}_x(\varphi)$, in the sense that*

$$\mathbb{S}, s \Vdash \varphi \textit{ iff } \mathbb{S} \models \mathtt{ST}_x(\varphi)[s]$$

for every pointed Kripke model $(\mathbb{S}, s)$.

Proof. The proof of this theorem can be proved by a straightforward induction on the complexity of μML-formulas.

For the inductive clause of the least fixpoint operator μ, consider the formula $\mu x.\varphi$. We leave it for the reader to verify (using the inductive hypothesis) that the formula $\mathtt{PRE}(\varphi, q)$ expresses that q is a pre-fixpoint of φ, and that the formula $\forall y.\big(p(y) \leftrightarrow \forall q.(\mathtt{PRE}(\varphi, q) \to q(y))\big)$ expresses that p is the intersection of all pre-fixpoints of φ. QED

In the other direction, the actual result that we will prove is somewhat stronger than mere expressive completeness.

Theorem 12.21 *There is an effectively defined translation $(\cdot)^* : \mathtt{MSO} \to \mu\mathtt{ML}$ such that a formula $\varphi \in \mathtt{MSO}$ is invariant under bisimulations iff it is equivalent to φ^*.*

We will prove this result by automata-theoretic means. Recall that in the previous section we obtained the following characterisations of the languages MSO and μML:

$$\begin{array}{lcl}
\mathtt{MSO} & \sim & Aut(\mathtt{1FOE}) \quad \text{(on trees)} \\
\mu\mathtt{ML} & \sim & Aut(\mathtt{1FO}).
\end{array}$$

The translation $(\cdot)^* : \mathtt{MSO} \to \mu\mathtt{ML}$ mentioned in Theorem 12.21 will be based on a construction transforming 1FOE-automata into 1FO-automata, whereas this construction in its turn is based on a translation $(\cdot)^*$ at the one-step level. For the details, we need to develop some rudimentary model theory at the level of monadic first-order logic, in this case linking the one-step languages MFOE and MFO.

Recall from Definition 10.26 that $\mathtt{1FOE}(A)$ and $\mathtt{1FO}(A)$ denote the sets of A-positive sentence in the languages $\mathtt{MFOE}(A)$ and $\mathtt{MFO}(A)$ of monadic first-order logic with and without identity, respectively. Our translation $(\cdot)^*$ involves the *basic forms* of Definition 12.13. Based on Proposition 12.17, we can provide the required translation from 1FOE to 1FO.

Definition 12.22 Fix a set A of propositional variables. For an arbitrary sentence $\chi^=(\overline{B},\overline{C}) \in \mathtt{BF}(A)$ we define

$$\big(\chi^=(\overline{B},\overline{C})\big)^* := \chi(\overline{B},\overline{C}),$$

and we extend this translation to the set $\mathtt{Dis}(\mathtt{BF}(A))$, simply by putting

$$\big(\bigvee_i \alpha_i\big)^* := \bigvee_i \alpha_i^*.$$

By Proposition 12.17 we may extend this definition to a map $(\cdot)^* : \mathtt{1FOE}(A) \to \mathtt{1FO}(A)$. ◁

Observe that the translation is in fact very simple: we obtain $\big(\chi^=(\overline{B},\overline{C})\big)^*$ from $\chi^=(\overline{B},\overline{C})$ simply by forgetting about the identity formulas occurring in the latter formula.

To exhibit the model-theoretic relation between the formulas α and α^*, we need one further definition.

Definition 12.23 Let $f : D' \twoheadrightarrow D$ be a surjective map from one set D' to another set D, and let A be some set of variables. Given a valuation $V : A \to \wp D$, we define the valuation $V_f : A \to \wp D'$, by putting, for $a \in A$:

$$V_f(a) := \{s' \in D' \mid f(s') \in V(a)\},$$

and, conversely, given a valuation $U : A \to \wp D'$, we let

$$U^f(a) := \{fs' \in D \mid s' \in U(a)\}$$

define a valuation on D. ◁

The only fact that we need about these translations and valuations is the following Proposition. We will use this result to transform the winning strategy of $\exists$ in one acceptance game to a winning strategy for her in a related acceptance game.

Proposition 12.24 *Let $\alpha \in \mathtt{1FOE}(A)$ be some one-step formula, and let D be some set. We let π denote the left projection map $\pi : D \times \omega \to D$.*
1) For any A-valuation V on D we have

$$D, V \models \alpha^* \text{ iff } D \times \omega, V_\pi \models \alpha. \tag{135}$$

2) As a corollary, for any A-valuation U on $D \times \omega$ we have

$$D \times \omega, U \models \alpha \text{ only if } D, U^\pi \models \alpha^*.$$

Proof. We leave the case where D is the empty set as an exercise for the reader, and focus on the case where $D \neq \varnothing$.

For part 1) of the Proposition, let α, D and π be as in its formulation. We will prove the equivalence (135).

For the left-to-right direction of (135), assume that $\langle D, V\rangle \models \chi(\overline{B}, \overline{C})$. Let $d_1, \ldots, d_n$ be elements in D satisfying the existential part of $\chi(\overline{B}, \overline{C})$, that is, for each i we find $d_i \in \bigcap_{b\in B_i} V(b)$. From the universal part of the formula it follows that for each $d \in D$ there is a subset $C_d \subseteq A$ such that $d \in \bigcap_{c\in C_d} V(c)$. Now we move to $D \times \omega$; it is easy to see that its elements $(d_1, 1), \ldots, (d_n, n)$ provide a sequence of n *distinct* elements that satisfy $(d_i, i) \in \bigcap_{b\in B_i} V_\pi(b)$ for each i. In addition, every element (d, n) distinct from the ones in the mentioned tuple will satisfy $(d, n) \in \bigcap_{c\in C_d} V_\pi(c)$. From these observations it is immediate that $\langle D \times \omega, V_\pi\rangle \models \chi^{=}(\overline{B}, \overline{C})$.

For the opposite direction of (135), assume that $\langle D\times\omega, V_\pi\rangle \models \chi^{=}(\overline{B}, \overline{C})$. Let $(d_1, k_1), \ldots, (d_n, k_n)$ be the sequence of distinct elements of $D \times \omega$ witnessing the existential part of $\chi^{=}(\overline{B}, \overline{C})$ in $\mathbb{D}'$. Then clearly, $d_1, \ldots, d_n$ witness the existential part of $\chi(\overline{B}, \overline{C})$ in $\langle D, V\rangle$. In order to show that $\langle D, V\rangle$ also satisfies the universal part $\forall z \bigvee_j \tau_{C_j}(z)$ of χ, consider an arbitrary element $d \in D$. Take any $m \in \omega \setminus \{k_1, \ldots, k_n\}$, then (d, m) is distinct from each (d_i, k_i). It follows that for some j we have $(d, m) \in \bigcap_{c\in C_j} V_\pi(c)$, and so we obtain $d \in \bigcap_{c\in C_j} V(c)$. Since d was arbitrary this shows that indeed $\langle D, V\rangle \models \forall z \bigvee_j \tau_{C_j}(z)$. So we have proved that $\langle D, V\rangle \models \chi(\overline{B}, \overline{C})$.

For part 2), assume that $D \times \omega, U \models \alpha^*$. It is straightforward to verify that $U(a) \subseteq (U^\pi)_\pi(a)$, for all $a \in A$. Hence by monotonicity of α with respect to the proposition letters in A, it follows that $D \times \omega, (U^\pi)_\pi \models \alpha^*$. But then we find $D, U^\pi \models \alpha^*$ by part 1) of the proposition. QED

Automata

Any translation between one-step formulas naturally induces a transformation of automata. In the current setting we obtain the following.

Definition 12.25 Given an automaton $\mathbb{A} = \langle A, \Theta, \Omega, a_I\rangle$ in $Aut(\mathtt{1FOE})$, we define the map $\Theta^* : (A \times \wp(\mathsf{P})) \to \mathtt{1FO}(A)$ by putting

$$\Theta^*(a) := (\Theta(a))^*,$$

and we let $\mathbb{A}^*$ denote the automaton $\mathbb{A}^* := \langle A, \Theta^*, \Omega, a_I\rangle$. ◁

We have now arrived at the main technical result of this section. It involves the notion of the ω-unravelling $\mathbb{E}_\omega(\mathbb{S}, s)$ of a model $\mathbb{S}$ around a point s. This construction[11] generalizes that of the unravelling of a model (Definition 1.23).

Definition 12.26 Let κ be a countable cardinal with $1 \leq \kappa \leq \omega$, and let $(\mathbb{S}, s)$ be a pointed Kripke model of type (P, D). A *κ-path through* $\mathbb{S}$ is a finite (non-empty) sequence of the form

[11] In a later version of the notes, this construction will be defined in Chapter 1.

$s_0d_1k_1s_1 \cdots s_{n-1}d_nk_ns_n$, where $s_i \in S$, $d_i \in \mathsf{D}$ and $k_i < \kappa$ for each i, and such that $R_{d_{i+1}}s_is_{i+1}$ for each $i < n$. The set of such paths is denoted as $Paths^{\kappa}(\mathbb{S})$; we use the notation $Paths^{\kappa}_s(\mathbb{S})$ for the set of paths starting at s. Given such a sequence ρ, we let $\mathsf{last}(\rho) \in S$ denote its last item.

The κ-*expansion* of $\mathbb{S}$ *around* s is the transition system $\mathbb{E}_{\kappa}(\mathbb{S}, s) = \langle Paths^{\kappa}_s(\mathbb{S}), \sigma^{\kappa}\rangle$, where

$$\begin{array}{rcl}\sigma^{\kappa}_V(s_0 \cdots d_nk_ns_n) & := & \sigma_V(s_n),\\ \sigma^{\kappa}_d(s_0 \cdots d_nk_ns_n) & := & \{(s_0 \cdots d_nk_ns_ndkt) \in Paths_s(\mathbb{S}) \mid R_ds_nt, 0 < k < \kappa\}.\end{array}$$

defines the coalgebra map $\sigma^{\kappa} = (\sigma_V, (\sigma_d \mid d \in \mathsf{D}))$. ◁

It is not hard to check that the *unravelling* of a model (Definition 1.23) can be identified with its 1-expansion. It is straightforward to verify the following proposition.

Proposition 12.27 *For any countable cardinal κ with $1 \leq \kappa \leq \omega$, the function* last*, mapping a sequence to its last item, is a surjective bounded morphism from $\mathbb{E}_{\kappa}(\mathbb{S}, s)$ to $\mathbb{S}$ mapping the single-item sequence s to its single state s.*

Proposition 12.28 *Let $\mathbb{A}$ be an automaton in $Aut(\mathtt{1FOE})$, then for any pointed Kripke model $(\mathbb{S}, s)$ we have that*

$$\mathbb{S}, s \Vdash \mathbb{A}^* \text{ iff } \mathbb{E}_{\omega}(\mathbb{S}, s), s \Vdash \mathbb{A}. \tag{136}$$

Proof. Let $\mathbb{A} = \langle A, \Theta, \Omega, a_I\rangle$ and $(\mathbb{S}, s)$ be as in the formulation of the Theorem. Let f denote the (surjective) bounded morphism from $\mathbb{E}_{\omega}(\mathbb{S}, s)$ to $\mathbb{S}$, and recall that by definition f is the function last mapping an ω-path to its final element. We will only prove the right-to-left direction of (136), leaving the (slightly easier) opposite direction as an exercise to the reader.

So assume that $\mathbb{E}_{\omega}(\mathbb{S}, s), s \Vdash \mathbb{A}$. Then $\exists$ has a (positional) winning strategy h in the acceptance game $\mathcal{A}^{\omega} := \mathcal{A}(\mathbb{A}, \mathbb{E}_{\omega}(\mathbb{S}, s))@(a_0, s_0)$, where we write $a_0 := a_I$ and $s_0 := s$. We need to provide her with a winning strategy h' in the acceptance game $\mathcal{A} := \mathcal{A}(\mathbb{A}^*, \mathbb{S})@(a_0, s_0)$, and we will define h' by induction on the length of a partial $\mathcal{A}$-match $\Sigma = (a_i, s_i)_{0\leq i\leq n}$. Via a simultaneous induction we define a partial $\mathcal{A}^{\omega}$-match $\Sigma' = (a_i, s'_i)_{0\leq i\leq n}$ which will be guided by $\exists$'s winning strategy h and satisfies $f(s'_i) = s_i$, for all i.

For the inductive step of these definitions, consider a partial $\mathcal{A}$-match $\Sigma = (a_i, s_i)_{0\leq i\leq n}$. Without loss of generality we may assume that Σ itself is guided by h', and inductively we may assume the existence of an h-guided shadow match $\Sigma' = (a_i, s'_i)_{0\leq i\leq n}$ of $\mathcal{A}^{\omega}$ such that $f(s'_i) = s_i$, for all i. In order to extend the definition of h', so that it defines a move for $\exists$ in the partial match Σ, obviously we consider this partial shadow match. Let $U : A \to \wp\sigma^{\omega}_R(s')$ be the A-valuation picked by $\exists$'s winning strategy h in the match Σ'. If we compare the collections $\sigma_R(s)$ and $\sigma^{\omega}_R(s')$ of successors of s and s' respectively, it is obvious that f restricts to a surjection from $\sigma^{\omega}_R(s')$ to $\sigma_R(s)$. Hence we may take the valuation

$$U^f : A \to \wp\sigma_R(s),$$

induced by U as in Definition 12.23, as the move given by the strategy h in the partial match Σ.

To see that this move is legitimate, we need to show that

$$\sigma_R, U^f \models \Theta^*(a_n, \sigma_V(s_n)), \tag{137}$$

that is, the one-step formula $\Theta^*(a_n, \sigma_V(s_n)$ holds in the A-structure (σ_R, U^f). It will be convenient to think of $\sigma_R^\omega(s')$ as the set $\sigma_R(s) \times \omega$, and of f as the projection map $\pi : \sigma_R(s) \times \omega \to \sigma_R(s)$. Then (137) is immediate by Proposition 12.242) and the fact that

$$\sigma_R^\omega, U \models \Theta(a_n, \sigma_V(s_n)), \tag{138}$$

simply because the valuation U is the legitimate move provided by $\exists$'s winning strategy h. Clearly then, the valuation U^f is a legitimate move for $\exists$.

In order to finish the inductive definition, we need to show how to extend, for any response (b, t) of $\forall$ to $\exists$'s move U^f, the shadow match Σ' with a position (b, t') such that $ft' = t$. But this is straightforward: if (b, t) is a legitimate move for $\forall$ in $\mathcal{A}$ at position U, then we have $t \in U^f(b)$, and so by definition there is a state $t' \in \sigma_R^\omega(s')$ such that $ft' = t$ and $t' \in U(b)$. Clearly then the continuation $\Sigma' \cdot (b, t')$ of Σ' satisfies the requirements.

We will now show that the just defined strategy h' is in fact *winning* for $\exists$ in $\mathcal{A}$. For this purpose, consider a full $\mathcal{A}$-match Σ which is guided by h'.

First consider the case where Σ is finite. It is not hard to prove, using the existence of the h-guided shadow match Σ', that the player who got stuck in Σ is $\forall$.

Having taken care of the finite matches, we now consider the case where $\Sigma = (a_i, s_i)_{0 \leq i < \omega}$ is infinite. It is not difficult to see that in this case there is an h-guided infinite shadow match $\Sigma' = (a_i, s_i')_{0 \leq i < \omega}$ of $\mathcal{A}^\omega$, such that $fs_i' = s_i$ for all $i < \omega$. But since h was assumed to be a winning strategy for $\exists$ in $\mathcal{A}^\omega$, Σ' is actually won by her. But since the priority maps of $\mathbb{A}$ and $\mathbb{A}^*$ are exactly the same, from this it is immediate that $\exists$ is also the winner the $\mathcal{A}$-match Σ.
QED

Proof of main result

As we shall see now, the expressive completeness of the modal μ-calculus is an almost immediate corollary of Proposition 12.28, given our earlier automata-theoretic characterizations of MSO and the modal μ-calculus.

Proof of Theorem 12.21. Let $\varphi \in \mathtt{MSO}$ be a monadic second-order formula, and let $\mathbb{B}_\varphi \in Aut(\mathtt{1FOE})$ be the automaton as given in Theorem 12.9. Then by Theorem 12.8 there is a formula $\varphi^* \in \mu\mathtt{ML}$ that is equivalent to the translation $(\mathbb{B}_\varphi)^*$ of $\mathbb{B}_\varphi$. Clearly then φ^* has been effectively obtained from φ.

We will show that φ is invariant under bisimulations iff it is equivalent to the formula φ^*. The direction from right to left is immediate since formulas of the modal μ-calculus are bisimulation invariant.

For the opposite direction, observe that by Proposition 12.28 and the definition on φ^*, for an arbitrary pointed Kripke model $(\mathbb{S}, s)$ we have

$$\mathbb{S}, s \Vdash \varphi^* \text{ iff } \mathbb{E}_\omega(\mathbb{S}, s), s \Vdash \varphi. \tag{139}$$

Now assume that φ is bisimulation invariant, then we have that

$$\mathbb{S}, s \Vdash \varphi \text{ iff } \mathbb{E}_\omega(\mathbb{S}, s), s \Vdash \varphi. \tag{140}$$

Combining these two observations, we see that $\mathbb{S}, s \Vdash \varphi^*$ iff $\mathbb{S}, s \Vdash \varphi$. But since $(\mathbb{S}, s)$ was arbitrary, this means that φ and φ^* are equivalent, as required. QED

Notes

The result that the modal μ-calculus is the bisimulation-invariant fragment of monadic second-order logic is due to Janin & Walukiewicz [8].

Exercises

Exercise 12.1 Let (D, V) and (D', V') be two one-step models over the same set A of monadic predicates. Then (D, V) is a *quotient* of (D', V') if there is a surjection $f : D' \to D$ such that $V' = V_f$. An MFOE-sentence α is *invariant under taking quotients* if we we have that $(D, V) \models \alpha$ iff $(D', V') \models \alpha$, whenever (D, V) is a quotient of (D', V').

Let α be an MFOE-sentence. Prove that α is invariant under taking quotients iff $\alpha \equiv \alpha^*$. Conclude that 1FO is the 'quotient-invariant fragment' of 1FOE.