The Propositional μ-Calculus and its Use in Model Checking

Jeffery Zucker[1]

Department of Computer Science and Systems, McMaster University, Hamilton, Ontario L8S 4K1, Canada

Abstract. This paper, and the preceding one, survey two types of formalism which has proved to be particularly successful for efficient automatic verification, or "model checking", of concurrent systems. In the previous paper we considered *branching time propositional temporal logics*, and now we turn to a more general formalism, the *propositional μ-calculus*. The emphasis, as before, is on a broad understanding rather than on technical details.

0 Introduction

We saw in the previous paper (Zucker 1992, §2) that many of the temporal operators of $\boldsymbol{CTL}$ turned out to be *fixed points* of certain functionals. In this paper we adopt this approach systematically. We will define a language $\mathcal{L}_\mu$ of temporal propositional logic which incorporates least and greatest fixed points.

Thus, although this paper is self-contained, the previous paper may form a useful introduction to it.

We repeat the *caveat* given in the introduction to the last paper. These two papers are intended mainly as introductions to the field for non-specialists, and the references should be consulted for further reading. The many worked examples in the following sections may be useful.

The present paper should also provide useful background material for the paper (Cleaveland 1992) on the Concurrency Workbench in this volume.

This paper has benefited particularly from Emerson and Lei (1986).

1 Least and greatest fixed points

We first give a lightning introduction to the mathematical theory of fixed points of functionals.

Let A be any set, $\mathcal{P}A$ the set of subsets of A, and $\Phi : \mathcal{P}A \to \mathcal{P}A$ a functional which is

- *monotonic*, *i.e.*, $B_1 \subseteq B_2 \implies \Phi(B_1) \subseteq \Phi(B_2)$;
- *continuous*, *i.e.*,
  if $B_0 \subseteq B_1 \subseteq B_2 \subseteq \dots$ then $\Phi(\bigcup_n B_n) = \bigcup_n \Phi(B_n)$, and
  if $C_0 \supseteq C_1 \supseteq C_2 \supseteq \dots$ then $\Phi(\bigcap_n C_n) = \bigcap_n \Phi(C_n)$.

[1]Research supported by a grant from the Natural Sciences and Engineering Research Council of Canada.

Then by the *Tarski-Knaster Theorem*, Φ has a *least fixed point*, *i.e.*, a least $B \subseteq A$ such that $\Phi(B) = B$. This l.f.p. can be obtained by *approximations from below*, *i.e.*,

$$\mu\Phi = \bigcup_{n=0}^{\infty} B_n$$

where $\emptyset = B_0 \subseteq B_1 \subseteq B_2 \subseteq \ldots$, and for all n, $B_{n+1} = \Phi(B_n)$.

Similarly, Φ has a *greatest fixed point*, which can be obtained by *approximations from above*:

$$\nu\Phi = \bigcap_{n=0}^{\infty} C_n$$

where $A = C_0 \supseteq C_1 \supseteq C_2 \supseteq \ldots$, and for all n, $C_{n+1} = \Phi(C_n)$.

2 Syntax

We present the syntax of our propositional μ-calculus $\mathcal{L}_\mu$. The propositional atoms may now be either constants or variables. We therefore have the following syntactic classes:

PropCon, the class of *propositional constants* $P, Q, \ldots$,

PropVar, the class of *propositional variables* $X, Y, \ldots$,

ProgAt, the class of *program atoms* or *basic actions* $a, b, \ldots$,

Form, the class of *formulae* $p, q, \ldots$, defined by

$$p \ ::= \ P \mid X \mid p \wedge q \mid \neg p \mid \mu X.p \mid \langle a \rangle p$$

where (in $\mu X.p$) X occurs *positively* in p *i.e.*, within the scope of an even number of negations. In other words, p is "syntactically monotonic in X". This guarantees the existence of a least (and) greatest fixed point.

We can then define $p \vee q$, etc., in the usual way, and also

$$\begin{aligned} \nu X.p \ &\equiv \ \neg \mu X.\neg p[X/\neg X] \\ [a]p \ &\equiv \ \neg\langle a \rangle \neg p \end{aligned}$$

where, in the first equation, X occurs positively in p, and hence also in $\neg p[X/\neg X]$. (Here '$p[X/\neg X]$' is the result of substituting $\neg X$ for all free occurrences of X in p.)

These operators have the informal meanings:

$\mu X.p(X)$ is the *least fixed point* of '$\lambda X.p(X)$',
$\nu X.p(X)$ is the *greatest fixed point* of '$\lambda X.p(X)$',
$\langle a \rangle p$ says that p holds at *some* next state reachable by a,
$[a]p$ says that p holds at *all* next states reachable by a.

We will give their formal semantics in the next section.

3 Semantics

A *model* is a triple $\mathcal{M} = (S, \mathcal{R}, L)$ (compare §2 in previous paper) where (as before) S is a finite set of states and $L : S \to \mathcal{P}(\boldsymbol{PropCon})$ is a labelling of states with propositional atoms, but now $\mathcal{R} : \boldsymbol{ProgAt} \to \mathcal{P}(S^2)$ is a mapping from program atoms a to sets of state transitions involving a.

Finiteness Assumption. We assume again that $\mathcal{M}$ is finite. (This means that S and $\boldsymbol{ProgAt}$ are finite.)

The *size* of $\mathcal{M}$ is now defined (*cf.* previous paper, §7) as

$$\begin{aligned} |\mathcal{M}| &= \|S\| + \sum_{a \in \boldsymbol{ProgAt}} \|\mathcal{R}(a)\| \\ &= \text{no. of states} + \text{total no. of transitions.} \end{aligned}$$

Remark on König's Lemma. König's Lemma states that any finitely branching tree which is infinite has an infinite path. A consequence of the Finiteness Assumption is that König's Lemma applies to any subtree of the computation tree of $\mathcal{M}$. This has an effect on the semantics, and will be used with some of the semantic equivalences given in the examples in the following sections.

We will define the meanings of formulae relative to *valuations*, *i.e.*, functions

$$\rho : \boldsymbol{PropVar} \longrightarrow \mathcal{P}(S).$$

In fact, we define $[\![p]\!]\rho \in \mathcal{P}(S)$ for all formulae p and valuations ρ, by structural induction on p. The various cases follow.

$$\begin{aligned} [\![P]\!]\rho &= \{s \mid P \in L(s)\} \\ [\![X]\!]\rho &= \rho(X) \\ [\![p \wedge q]\!]\rho &= [\![p]\!]\rho \wedge [\![q]\!]\rho \\ [\![\neg p]\!]\rho &= S \backslash [\![p]\!]\rho \\ [\![\langle a \rangle p]\!]\rho &= \{s \mid \exists t : (s,t) \in \mathcal{R}(a) : t \in [\![p]\!]\rho\} \end{aligned}$$

Note that, according to these definitions,

$$[\![[a]p]\!]\rho = \{s \mid \forall t : (s,t) \in \mathcal{R}(a) : t \in [\![p]\!]\rho\}$$

Finally, for the case $[\![\mu X.p]\!]$, we define the functional

$$\Phi_p^\rho : \mathcal{P}(S) \longrightarrow \mathcal{P}(S)$$

by

$$\Phi_p^\rho(T) = [\![p]\!]\rho\{X/T\}$$

where the *variant valuation* $\rho\{X/T\}$ is defined, for all $Y \in \boldsymbol{PropVar}$, by

$$\rho\{X/T\}(Y) = \begin{cases} T & \text{if } Y \equiv X \\ \rho(Y) & \text{otherwise.} \end{cases}$$

Note that Φ^ρ_p is *monotonic* (since X occurs positively in p), and *continuous.* Hence, by the Tarski-Knaster Theorem (§2), Φ^ρ_p has *least* and *greatest fixed points*, $\boldsymbol{\mu}\Phi^\rho_p$ and $\boldsymbol{\nu}\Phi^\rho_p$ respectively. So define

$$[\![\mu X.p]\!] \;=\; \boldsymbol{\mu}\Phi^\rho_p.$$

Note that, according to these definitions,

$$[\![\nu X.p]\!] \;=\; \boldsymbol{\nu}\Phi^\rho_p.$$

Remarks. (1) To correlate this semantic definition with the one in the previous paper, we can use the notation

$$s,\rho \models p \iff s \in [\![p]\!]\rho.$$

(Compare Remark 2 in §2, *loc. cit.*) Note that the valuation ρ is a new feature of our definition (not present in the previous paper) because of the presence of propositional variables.

(2) Let $\boldsymbol{PropVar}(p)$ be the set of propositional variables contained in p. If $\rho_1(X) = \rho_2(X)$ for all $X \in \boldsymbol{PropVar}(p)$, then $[\![p]\!]\rho_1 = [\![p]\!]\rho_2$. In particular, if p is a *sentence* (*i.e.*, $\boldsymbol{PropVar}(p) = \emptyset$), then $[\![p]\!]\rho$ is *independent* of ρ.

(3) We have

$$\begin{aligned}[\![\mu X.p]\!]\rho \;&=\; \boldsymbol{\mu}\Phi^\rho_p \\ &=\; \bigcup_{n=0}^{\infty} T_n\end{aligned}$$

where $\emptyset = T_0 \subseteq T_1 \subseteq T_2 \subseteq \ldots,$ and for all n, $T_{n+1} = \Phi^\rho_p(T_n)$. But since S (and hence $\mathcal{P}(S)$) is *finite*, for some n $T_n = T_{n+1} = \cdots = \boldsymbol{\mu}\Phi$.

Similarly,

$$\begin{aligned}[\![\nu X.p]\!]\rho \;&=\; \boldsymbol{\nu}\Phi^\rho_p \\ &=\; \bigcup_{n=0}^{\infty} T_n\end{aligned}$$

where $S = T_0 \supseteq T_1 \supseteq T_2 \supseteq \ldots,$ and for all n, $T_{n+1} = \Phi^\rho_p(T_n)$, and, again, for some n $T_n = T_{n+1} = \cdots = \boldsymbol{\nu}\Phi$.

4 Worked examples

The 'μ' and 'ν' operators can be represented by infinite disjunctions and conjunctions respectively — informally, since these infinite operators are not part of the syntax of $\mathcal{L}_\mu$:

$$\mu X.p \;=\; \bigvee_{n=0}^{\infty} p_n, \qquad \text{where } \begin{cases} p_0 = \mathsf{false} \\ p_{n+1} = p[X/p_n] \end{cases}$$

$$\nu X.q \;=\; \bigwedge_{n=0}^{\infty} q_n, \qquad \text{where } \begin{cases} q_0 = \mathsf{true} \\ q_{n+1} = q[X/q_n]. \end{cases}$$

It is instructive to work through the (informal) meanings of some $\mathcal{L}_\mu$-expressions, via these infinite disjunctions and conjunctions. We will consider $\mu X.p$ and $\nu X.p$ for a number of formulae p.

Example 1. $p \equiv \langle a\rangle X$.

(a) $\mu X.p = \bigvee_n p_n$, where

$$\begin{aligned} p_0 &= \mathsf{false} \\ p_1 &= \langle a\rangle p_0 = \langle a\rangle\mathsf{false} = \mathsf{false} \\ p_2 &= \langle a\rangle p_1 = \langle a\rangle\mathsf{false} = \mathsf{false} \\ &\vdots \end{aligned}$$

So $\mu X.\langle a\rangle X = \mathsf{false}$.

(b) $\nu X.p = \bigwedge_n q_n$, where

$$\begin{aligned} q_0 &= \mathsf{true} \\ q_1 &= \langle a\rangle q_0 = \langle a\rangle\mathsf{true} \\ q_2 &= \langle a\rangle q_1 = \langle a\rangle\langle a\rangle\mathsf{true} \\ &\vdots \\ q_n &= \langle a\rangle^n\mathsf{true} = \text{“there is a run of } a\text{'s of length } n\text{”} \\ &\vdots \end{aligned}$$

So
$$\begin{aligned} \nu X.\langle a\rangle X &= \text{“there are arbitrarily long runs of } a\text{'s”} \\ &= \text{“there is an infinite run of } a\text{'s”}. \end{aligned}$$

This last equivalence is by König's Lemma, which holds in the computation tree by our Finiteness Assumption.

Example 2. $p \equiv [a]X$.

(a) $\mu X.p = \bigvee_n p_n$, where

$$\begin{aligned} p_0 &= \mathsf{false} \\ p_1 &= [a]p_0 = [a]\mathsf{false} \\ p_2 &= [a]p_1 = [a][a]\mathsf{false} \\ &\vdots \\ p_n &= [a]^n\mathsf{false} = \text{“no run of } a\text{'s of length } n\text{”} \\ &\vdots \end{aligned}$$

So $\mu X.[a]X = \text{“no infinite run of } a\text{'s”}$.

(Alternatively: $\mu X.[a]X = \neg\nu X.\neg[a]\neg X = \neg\nu X.\langle a\rangle X$, and use Example 1$(b)$.)

(*b*) $\nu X.p \;=\; \bigwedge_n q_n$, where

$$\begin{aligned} q_0 &= \mathsf{true} \\ q_1 &= [a]q_0 \;=\; [a]\mathsf{true} \;=\; \mathsf{true} \\ &\;\vdots \\ q_n &= \mathsf{true} \\ &\;\vdots \end{aligned}$$

So $\quad \nu X.[a]X \;=\; \mathsf{true}.$

(Alternatively: $\nu X.[a]X \;=\; \neg\mu X.\neg[a]\neg X \;=\; \neg\mu X.\langle a\rangle X$, and use Example 1(*a*).)

Example 3. $p \equiv P \wedge \langle a\rangle X$

(*a*) $\mu X.p \;=\; \mathsf{false}$ (as in Example 1(*a*)).

(*b*) $\nu X.p \;=\; \bigwedge_n q_n$, where

$$\begin{aligned} q_0 &= \mathsf{true} \\ q_1 &= P \wedge \langle a\rangle q_0 \;=\; P \wedge \langle a\rangle\mathsf{true} \\ q_2 &= P \wedge \langle a\rangle q_1 \;=\; P \wedge \langle a\rangle\big(P \wedge \langle a\rangle\mathsf{true}\big) \\ &\;\vdots \\ q_n &= \text{“there is a run of } a\text{'s of length } n\text{, with } P \text{ holding throughout”} \\ &\;\vdots \end{aligned}$$

So $\quad \nu X.\big[P \wedge \langle a\rangle X\big] \;=\;$ “can perform a forever, with P holding throughout”.

Example 4. $p \equiv P \vee \langle a\rangle X$

(*a*) $\mu X.p \;=\; \bigvee_n p_n$, where

$$\begin{aligned} p_0 &= \mathsf{false} \\ p_1 &= P \vee \langle a\rangle p_0 \;=\; P \vee \langle a\rangle\mathsf{false} \;=\; P \\ p_2 &= P \vee \langle a\rangle p_1 \;=\; P \vee \langle a\rangle P \\ p_3 &= P \vee \langle a\rangle p_2 \;=\; P \vee \langle a\rangle\big(P \vee \langle a\rangle P\big) \;=\; P \vee \langle a\rangle P \vee \langle a\rangle^2 P \\ &\;\vdots \\ p_n &= P \vee \langle a\rangle P \vee \langle a\rangle^2 P \vee \ldots \vee \langle a\rangle^{n-1} P \\ &\;\vdots \end{aligned}$$

So $\quad \mu X.\big[P \vee \langle a\rangle X\big] \;=\;$ “can perform (finite) run of a's until P holds”.

(b) $\nu X.p \;=\; \bigwedge_n q_n$, where

$$
\begin{aligned}
q_0 &= \mathsf{true}\\
q_1 &= P \vee \langle a\rangle q_0 \;=\; P \vee \langle a\rangle\,\mathsf{true}\\
q_2 &= P \vee \langle a\rangle q_1 \;=\; P \vee \langle a\rangle\big(P \vee \langle a\rangle\,\mathsf{true}\big) \;=\; P \vee \langle a\rangle P \vee \langle a\rangle^2\,\mathsf{true}\\
&\;\vdots\\
q_n &= P \vee \langle a\rangle P \vee \langle a\rangle^2 P \vee \ldots \vee \langle a\rangle^{n-1} P \vee \langle a\rangle^n\,\mathsf{true}\\
&\;\vdots
\end{aligned}
$$

So $\quad \nu X.\big[P \vee \langle a\rangle X\big] \;=\;$ "can perform run of a's until P holds, or forever!".

5 Interpreting $\boldsymbol{CTL}$ in the μ-Calculus: More examples

Recall (§5 of the previous paper) the definition of the "state formulae" of $\boldsymbol{CTL}$:

$$p \;::=\; P \mid \mathsf{true} \mid p \wedge q \mid \neg p \mid \mathsf{EX}p \mid \mathsf{AX}p \mid \mathsf{E}(p\mathsf{U}q) \mid \mathsf{A}(p\mathsf{U}q).$$

We will see how to interpret these as $\mathcal{L}_\mu$-formulae. First, we can define

$$
\begin{aligned}
\mathsf{EX}p &\equiv \bigvee_a \langle a\rangle p, \quad \text{which we write as} \quad \langle -\rangle p\\
\mathsf{AX}p &\equiv \bigwedge_a \langle a\rangle p, \quad \text{which we write as} \quad [-]p.
\end{aligned}
$$

Note that the disjunction and conjunction over program atoms a shown above are *finite*, by our Finiteness Assumption, and hence '$\langle -\rangle p$' and '$[-]p$' are $\mathcal{L}_\mu$-formulae.

What about $\mathsf{E}(p\mathsf{U}q)$ and $A(p\mathsf{U}q)$? To help us, let us first try some more examples.

Example 5. $\mu X.\big[p \vee \langle -\rangle X\big] \;=\; \bigvee_n p_n$, where (as in Example 4(a))

$$
\begin{aligned}
p_n &= p \vee \langle -\rangle p \vee \langle -\rangle^2 p \vee \ldots \vee \langle -\rangle^{n-1} p\\
&= \text{"along } \textit{some} \text{ path, } p \text{ holds in at most } n-1 \text{ steps"}.
\end{aligned}
$$

So

$$
\begin{aligned}
\mu X.\big[p \vee \langle -\rangle X\big] &= \text{"along } \textit{some} \text{ path, } p \text{ eventually holds"}\\
&= \mathsf{EF}p\\
&= \mathsf{E}(\mathsf{true}\,\mathsf{U}\,p) \quad \text{(as in §5 of previous paper)}
\end{aligned}
$$

Example 6. $\mu X.\big[p \vee [-]X\big] \;=\; \bigvee_n p_n,$ where

$$\begin{aligned}
p_0 &= \mathsf{false}\\
p_1 &= p \vee [-]p_0 \;=\; p \vee [-]\mathsf{false} \;=\; p\\
p_2 &= p \vee [-]p_1 \;=\; p \vee [-]p\\
p_3 &= p \vee [-]p_2 \;=\; p \vee [-]\big(p \vee [-]p\big)\\
&\;\vdots\\
p_n &= \text{“along \textit{every} path, p holds in at most $n-1$ steps”}\\
&\;\vdots
\end{aligned}$$

So

$$\begin{aligned}
\mu X.\big[p \vee [-]X\big] &= \text{“along \textit{every} path, p eventually holds”} \quad \text{(by König's Lemma)}\\
&= \mathsf{AF}p\\
&= \mathsf{A}(\mathsf{true}\,\mathsf{U}\,p) \quad \text{(as in §5 of previous paper)}
\end{aligned}$$

Example 7.

$$\begin{aligned}
\nu X.\big[p \wedge \langle - \rangle X\big] &= \neg\mu X.\neg\big[p \wedge \langle - \rangle \neg X\big]\\
&= \neg\mu X.\big[\neg p \vee \neg\langle - \rangle \neg X\big]\\
&= \neg\mu X.\big[\neg p \vee [-]X\big]\\
&= \neg\mathsf{AF}\neg p \quad \text{(by Example 6)}\\
&= \mathsf{EG}p
\end{aligned}$$

Example 8. $\nu X.\big[p \wedge [-]X\big] \;=\; \mathsf{AG}p$
(similarly, by working with duals from Example 5).

Example 9. $\mu X.\big[q \vee (p \wedge \langle - \rangle X)\big] \;=\; \bigvee_n p_n,$ where (*cf.* Example 5)

$$\begin{aligned}
p_0 &= \mathsf{false}\\
p_1 &= q \vee (p \wedge \langle - \rangle p_0) \;=\; q \vee (p \wedge \langle - \rangle \mathsf{false}) \;=\; q\\
p_2 &= q \vee (p \wedge \langle - \rangle p_1) \;=\; q \vee (p \wedge \langle - \rangle q)\\
p_3 &= q \vee (p \wedge \langle - \rangle p_2) \;=\; q \vee \big(p \wedge \langle - \rangle (q \vee (p \wedge \langle - \rangle q))\big)\\
&\;\vdots\\
p_n &= \text{“along \textit{some} path, q holds in at most $n-1$ steps, and p holds until then”}\\
&\;\vdots
\end{aligned}$$

So $\mu X.\big[q \vee (p \wedge \langle - \rangle X)\big]$

$$\begin{aligned}
&= \text{“along \textit{some} path, q eventually holds, and p holds until then”}\\
&= \mathsf{E}(p\mathsf{U}q).
\end{aligned}$$

Example 10. $\mu X.\big[q \vee (p \wedge [-]X)\big] \;=\; \bigvee_n p_n,$ where
(working as in Example 9, but with '$\langle - \rangle$' replaced by '$[-]$' throughout)

p_n = "along *every* path, q holds in at most $n-1$ steps, and p holds until then"

So $\mu X.\big[q \vee (p \wedge [-]X)\big]$

$$\begin{aligned} &= \text{"along } \textit{every} \text{ path, } q \text{ eventually holds, and } p \text{ holds until then"} \\ &\quad \text{(by König's Lemma)} \\ &= \mathsf{A}(p\mathsf{U}q). \end{aligned}$$

Hence we have, from Examples 9 and 10, a translation of $\boldsymbol{CTL}$ into $\mathcal{L}_\mu$.

6 Expressing safety, liveness and fairness properties

Remember (see previous paper, §4) that
(1) a *safety property* means that something bad never happens; it is typically associated with the prefix AG or with ν (see Example 8);
(2) a *liveness property* means that something good eventually happens; it is typically associated with the prefix AF or with μ (see Example 6).

We give two final examples, involving an alternation of μ and ν, which are associated with fairness.

Example 11. $\nu X.\mu Y.\langle - \rangle\big[(p \wedge X) \vee Y\big]$

To work out the meaning of this, first consider the subformula

$$p(X) \;\equiv\; \mu Y.\langle - \rangle\big[p \wedge X) \vee Y\big].$$

By the technique of Example 5, it is easy to see that

$$p(X) \;=\; \text{"along some path, } p \wedge X \text{ eventually holds".}$$

Then the given formula is $\mu X.p(X) \;=\; \bigwedge_n q_n,$ where

$$\begin{aligned} q_0 &= \mathsf{true} \\ q_1 &= \text{"along some path, } p \text{ eventually holds (at least) once"} \\ q_2 &= \text{"along some path, } p \text{ eventually holds (at least) twice"} \\ &\;\;\vdots \\ q_n &= \text{"along some path, } p \text{ eventually holds (at least) } n \text{ times"} \\ &\;\;\vdots \end{aligned}$$

Then by König's Lemma, $\mu X.p(X) \;=$

$$\text{"along } \textit{some} \text{ path, } p \text{ holds } \textit{infinitely often}\text{"} \;=\; \mathsf{EF}^\infty p.$$

(See previous paper, §1, for the notation.)

Example 12. $\nu X.\mu Y.[-]\big[(p \vee X) \wedge Y\big]$

This is the dual of Example 11, so its meaning is

$$\text{“along } \textit{every} \text{ path, } p \text{ holds } \textit{almost always}\text{”} \;=\; \mathsf{AG}^{\infty} p.$$

7 The model checking algorithm for the μ-Calculus

We adapt the algorithm of the previous paper (§6) to $\mathcal{L}_\mu$, following the method of Emerson and Lei (1986).

First, for convenience, we extend the syntax of $\mathcal{L}_\mu$ given in §2, to include '$\vee$', 'ν' and '$[\cdot]$'. Each sentence of $\mathcal{L}_\mu$ is then equivalent to one in *pnf* (positive normal form), *i.e.*, having at most single negations immediately before the propositional constants.

A sentence can be put into pnf by:

- giving all the bound propositional variables distinct names;
- driving all negations inward, by the rules:

$$\begin{aligned}
\neg\neg p &= p \\
\neg(p \wedge q) &= \neg p \vee \neg q) \\
\neg(p \vee q) &= \neg p \wedge \neg q \\
\neg\langle a\rangle p &= [a]\neg p \\
\neg[a]p &= \langle a\rangle \neg p \\
\neg\mu X.p &= \nu X.\neg p[X/\neg X] \\
\neg\nu X.p &= \mu X.\neg p[X/\neg X]
\end{aligned}$$

Notation. Given a finite model $\mathcal{M} = (S, \mathcal{R}, L)$, let us write $\boldsymbol{state}(p) = \boldsymbol{state}_{\mathcal{M}}(p)$ for the set of states in $\mathcal{M}$ which satisfy a sentence p.

Note that for a sentence p, $\boldsymbol{state}(p) = [\![p]\!]\rho$ (in the notation of §3), for *any* valuation ρ (see Remark 2 in §3).

Suppose now that we are given a sentence p_0 in pnf. We want to find $\boldsymbol{state}(p_0)$.

Suppose that the propositional variables contained in p_0 are among $\{X_1, \ldots, X_n\}$. Note that these are all bound in p_0, since p_0 is a sentence, but they may be free in subformulae of p_0! Hence for subformulae p of p_0, $[\![p]\!]\rho$ is, in general, *not* independent of ρ.

We will compute $\boldsymbol{state}(p_0)$ by computing $\boldsymbol{state}(p)$ for subformulae p of p_0, where $\boldsymbol{state}(p) = [\![p]\!]\rho$, for successive "approximate" valuations ρ, defined by

$$X_1 \mapsto T_1, \ldots, X_n \mapsto T_n \tag{1}$$

for suitable $T_1, \ldots, T_n \subseteq S$.

To begin with, *initialize* $T_1, \ldots, T_n$ to $\emptyset$.

Now compute $\boldsymbol{state}(p)$ for subformulae p of p_0, by structural induction on p:

- $p \equiv P$. Then $\boldsymbol{state}(p) = \{s \in S \mid P \in L(s)\}$.
- $p \equiv \neg P$. Then $\boldsymbol{state}(p) = \{s \in S \mid P \notin L(s)\}$.
- $p \equiv X_j$ $(j = 1, \ldots, n)$. Then $\boldsymbol{state}(p) = T_j$.
- $p \equiv q_1 \wedge q_2$. Then $\boldsymbol{state}(p) = \boldsymbol{state}(q_1) \cap \boldsymbol{state}(q_2)$.
- $p \equiv q_1 \vee q_2$. Then $\boldsymbol{state}(p) = \boldsymbol{state}(q_1) \cup \boldsymbol{state}(q_2)$.
- $p \equiv \langle a \rangle q$. Then $\boldsymbol{state}(p) = \{s \in S \mid \exists t : (s,t) \in \mathcal{R}(a) : t \in \boldsymbol{state}(q)\}$.
- $p \equiv [a]q$. Then $\boldsymbol{state}(p) = \{s \in S \mid \forall t : (s,t) \in \mathcal{R}(a) : t \in \boldsymbol{state}(q)\}$.
- $p \equiv \mu X_j q$. This case and the next are the most interesting. We construct $\boldsymbol{state}(p)$ $(= T_j)$, by approximating from below (*cf.* Case 6 in §6 of the previous paper), starting with $T_j = \emptyset$. We execute the following loop, which uses the auxiliary variable T for a subset of S:

$$
\begin{aligned}
& T_j \leftarrow \emptyset; \\
& \mathsf{repeat}\ T \leftarrow T_j; \\
& \qquad T_j \leftarrow \boldsymbol{state}(q) \\
& \mathsf{until}\ T_j = T; \\
& \boldsymbol{state}(p) \leftarrow T_j.
\end{aligned}
$$

  Note that this loop terminates, *i.e.*, T_j attains a constant size after finitely many iterations, since S is finite by the Finiteness Assumption.
- $p \equiv \mu X_j q$. Exactly as the previous case, except that at the start of the loop, T_j is initialized to S instead of $\emptyset$:

$$
\begin{aligned}
& T_j \leftarrow S; \\
& \quad \vdots
\end{aligned}
$$

8 Time complexity of the algorithm

The *alternation depth* of a formula p means the maximum depth of alternating nestings of 'μ' and 'ν' in p. The idea is clear, but the exact definition is complicated. It can be found in Emerson and Lei (1986).

We then define $\mathcal{L}_{\mu_k}$ to be the fragment of $\mathcal{L}_\mu$ restricted to formulae of alternation depth $\leq k$.

Note that $\boldsymbol{CTL}$ (and many other interesting formalisms) can be interpreted in $\mathcal{L}_{\mu_1}$(see §5). Also fairness properties such as '$\mathsf{EF}^\infty p$' and '$\mathsf{AG}^\infty p$' can be interpreted in $\mathcal{L}_{\mu_2}$.

Now it is not hard to see (compare §7 of previous paper) that the complexity for $\mathcal{L}_{\mu_k}$ is a *polynomial of degree* k in $|\mathcal{M}|$ and $|p_0|$ (where $|\mathcal{M}|$ was defined in §3 and $|p_0|$ is the length of the $\mathcal{L}_{\mu_k}$ sentence p_0).

9 Further developments

9.1 "Local" algorithms

A drawback with the algorithm outline above (§7) is that one has to compute the *whole fixed point set* $[\![\mu X.q]\!]$ (or $[\![\nu X.q]\!]$) to determine whether $s, \rho \models \mu X.q$ (or $s, \rho \models \nu X.q$), for a given s. More "local" (and hence efficient) approaches which avoid this problem have been given in Stirling and Walker (1991), Winskel (1989) and Cleaveland (1990), all of which use *proof systems* in *tableau form* to derive inferences of the form (something like) $s \models q$. These systems use "syntax-directed" proof rules. For example, for 'ν', there is a *fixed point induction rule* of the form

from $\quad s \models \nu X.q \implies s \models q[X/\nu X.q]$

infer $\quad s \models \nu X.q.$

It is shown in these papers that the corresponding systems are *sound* and *complete.* For details, see the papers.

9.2 Dropping the Finiteness Assumption

Finally we point out that Bradfield and Stirling (1990) have succeeded in dropping the Finiteness Assumption. They extend the syntax to formulae of the form '$\langle K\rangle p$' and '$[K]P$' for arbitrary subsets K of a (possibly infinite) set S of states. Again, we refer the reader to the paper for details.

References

Bradfield, J. Stirling, C. (1990): Local model checking for infinite state spaces. Report no. ECS-LFCS-90-115, Laboratory for Foundations of Computer Science, University of Edinburgh.

Cleaveland, R. (1990): Tableau-based model checking in the propositional mu-calculus. Acta Informatica **27**, 725–747

Cleaveland, R. (1992): The Concurrency Workbench. In this volume.

Emerson, E.A., Lei, C.-L. (1986): Efficient model checking in fragments of the propositional mu-calculus (Extended abstract). In: Proc. IEEE Symp. on Logic in Computer Science, June 1986, Cambridge, MA, pp. 267–278

Stirling, C., Walker, D. (1991): Local model checking in the modal mu-calculus. Theoretical Computer Science **89**, 161–177.

Winskel, G. (1989): A note on model checking the modal ν-calculus. In: 16th International Colloq. on Automata, Languages and Programming, Stresa, Italy, July 1989 (ed. G. Ausiello, M. Dezani-Ciancaglini, S. Ronchi della Rocca). Lecture Notes in Conputer Science, vol. 372, Springer-Verlag, pp. 761–772

J.I. Zucker (1992): Propositional temporal logics and their use in model checking. In this volume.