RESULTS ON THE PROPOSITIONAL $\mu$-CALCULUS

Dexter KOZEN*
Mathematical Sciences Department, IBM Research Center, Yorktown Heights, NY 10598, U.S.A.

Abstract. In this paper we define and study a propositional $\mu$-calculus $L_\mu$, which consists essentially of propositional modal logic with a least fixpoint operator. $L_\mu$ is syntactically simpler yet strictly more expressive than Propositional Dynamic Logic (PDL). For a restricted version we give an exponential-time decision procedure, small model property, and complete deductive system, thereby subsuming the corresponding results for PDL.

1. Introduction

The propositional $\mu$-calculus refers collectively to a class of programming logics consisting of propositional model logic with a least fixpoint operator $\mu$. The $\mu$-calculus originated with Scott and De Bakker [22] and was developed by Hitchcock and Park [7], Park [17], De Bakker and De Roever [2], De Roever [20] and others. The system we consider here is very close to a system appearing in [1]. The results of this volume, however, are mostly inspired by the work of Pratt [19], who defines a propositional $\mu$-calculus $P_\mu$, shows that $P_\mu$ subsumes PDL, and extends the exponential-time decision procedure for PDL to $P_\mu$. It is not known, however, whether $P_\mu$ contains PDL strictly, and a deductive system is not given.

The usual proof rules for expressions involving least fixpoints do not readily apply to Pratt’s $P_\mu$ due to its formulation as a least root calculus rather than a least fixpoint calculus. This formulation was chosen in order to capture the reverse operator of PDL. Also, formulas of $P_\mu$ are required to satisfy a rather strong condition akin to syntactic continuity. This condition renders illegal several useful formulas: e.g., the formula $\mu X[a]X$, which is the same as $\neg a \bullet \bot$ in the notation of Streitz [21], expresses the property that the program $b$ has no infinite computations. Pratt’s syntactic restriction allows the filtration-based decision procedure of [18] to extend to $P_\mu$, whereas no filtration-based decision procedure can work in the presence of $\mu X[a]X$, as shown by Streitz [21].

Here we propose weakening the syntactic continuity requirement and returning to a least fixpoint formulation. The resulting system is called $L_\mu$. Although full $L_\mu$ is decidable, the best bound known is nonelementary [16]. However, under a natural syntactic restriction which is still somewhat weaker than full syntactic continuity, better bounds can be obtained. For the syntactically restricted version, we show:

* These results were obtained during the author’s sabbatical at the University of Aarhus, Denmark.
2. Definition of $\text{L} \mu$ and $\text{L} \mu^+$

$\text{L} \mu$ is essentially propositional modal logic with a least fixpoint operator $\mu$. $\text{L} \mu^+$ is an infinitary language containing $\text{L} \mu$, obtained by augmenting $\text{L} \mu$ with the ability to construct the $\alpha$-fold composition of a monotone operator, where $\alpha$ is any ordinal. $\text{L} \mu^+$ is useful in transfinite inductive arguments.

2.1. Syntax

The primitive nonlogical symbols of $\text{L} \mu$ and $\text{L} \mu^+$ consist of propositional constants $P, Q, \ldots$, including the constants 0, 1, propositional variables $X, Y, \ldots$, and program constants $a, b, \ldots$. Formulas $p, q, \ldots$ of $\text{L} \mu^+$ are defined inductively:

- $X$, (2.1.1) $(a)p$,
- $P$, (2.1.2) $\alpha XpX$, $\alpha$ an ordinal,
- $p \lor q$, (2.1.3) $\mu XpX$,
- $\neg p$, (2.1.4)

where in (2.1.6) and (2.1.7) $pX$ is positive in the variable $X$, i.e., every free occurrence of $X$ in $pX$ occurs in the scope of an even number of negations $\neg$. (The notions of scope, bound and free occurrences of variables, closed formulas, etc. are the same as in first-order predicate logic, where $\mu X$ and $\alpha X$ are treated as quantifiers.)

Intuitively, $\alpha XpX$ represents the $\alpha$-fold composition of the operator $\Delta XpX$ applied to 0.

2.2. Semantics

A model is a structure $M = (S, I)$, where $S$ is a set of states and $I$ an interpretation of the propositional constants and program constants as, respectively, subsets of $S$ and binary relations on $S$. We require that $I(0) = \emptyset$ and $I(1) = S$.

A valuation is a mapping assigning a subset of $S$ to each variable. Formally, a formula $p$ is interpreted as an operator $p^M$ from valuations to subsets of $S$. However, since $p^M$ will be independent of the variables not occurring free in $p$, we will view $p^M$ as a function of its free variables. We will write $p(X)$ to denote that all free variables of $p$ are among $\vec{X} = X_1, \ldots, X_n$. $p^M(\vec{A})$ to denote the value of $p^M$ on any valuation that assigns $A_i$ to $X_i$, $1 \leq i \leq n$. The operator $p^M$ is defined inductively as follows:

- (2.2.1) $X^M(\vec{A}) = A_n$,
- (2.2.2) $\neg p^M(\vec{A}) = S - p^M(\vec{A})$,
- (2.2.3) $(p \lor q)^M(\vec{A}) = p^M(\vec{A}) \cup q^M(\vec{A})$,

where, in (2.2.5),

$$(\alpha^M)[B] = \{s \in B \mid \exists \alpha \in I(\alpha)\}.$$

To define (2.1.6) and (2.1.7), let $pX$ be a formula positive in $X$, and let $\bar{X}$ denote the other free variables of $pX$. Thus $pX = p(X, \bar{X})$. We assume by induction hypothesis that the operator $p^M$ has already been defined. Because of the requirement that $pX$ be positive in $X$, the operator $p^M$ is monotone in the variable $X$ with respect to the subset relation.

- (2.2.6a) $0 Xp^M(\vec{A}) = 0^M = \emptyset$,
- (2.2.6b) $(a + 1) Xp^M(\vec{A}) = p^M(aXp^M(\vec{A}), \vec{A})$,
- (2.2.6c) $\delta Xp^M(\vec{A}) = \bigcup_{\alpha<\delta} \beta Xp^M(\vec{A})$, $\delta$ a limit ordinal,
- (2.2.7) $\mu Xp^M(\vec{A}) = \bigcup_{\alpha<\beta} \beta Xp^M(\vec{A})$,

where, in (2.2.7), the union is over all ordinals $\beta$. Taking $\mu > \alpha$ for any ordinal $\alpha$.

(2.2.6)(a-c) and (2.2.7) can be combined into the single definition:

- (2.2.8) $\alpha Xp^M(\vec{A}) = \bigcup_{\alpha<\beta} p^M(\beta Xp^M(\vec{A}), \vec{A})$,

where $\alpha$ is either an ordinal or $\mu$.

For fixed $\vec{A}$, since $p^M(X, \vec{A})$ is monotone in $X$, it follows that $\alpha Xp^M(\vec{A}) \subseteq \beta Xp^M(\vec{A})$ whenever $\alpha \leq \beta$. At some level $\kappa$, we must have

$$\alpha Xp^M(\vec{A}) = (\kappa + 1) Xp^M(\vec{A}) = \mu Xp^M(\vec{A}).$$

The least such $\kappa$ is called the closure ordinal of the operator $\mu Xp^M(X, \vec{A})$.

It follows from the Knaster-Tarski theorem that $\mu Xp^M(\vec{A})$ is the least fixpoint of the operator $\mu Xp^M(X, \vec{A})$, and that

- (2.2.9) $\mu Xp^M(\vec{A}) = \bigcap \{B \mid \mu p^M(B, \vec{A}) = B\} = \bigcap \{B \mid p^M(B, \vec{A}) \subseteq B\}$.

If $p$ is closed, then $p^M$ is constant. In this case $s$ is said to satisfy $p$ (notation: $M, s \models p$ or $s \models p$) if $s \in p^M$. 

Results on the propositional $\mu$-calculus
3. Notation and basic results

3.1. Defined operators, positive normal form

In addition to the primitive operators, we will use the usual defined Boolean operators \( \land, \rightarrow \), as well as the defined operators

\[
[a]p = \neg(a)\neg p, \quad \nu X.pX = \nu_\mu X.\neg p\neg X.
\]

The operator \( \nu \) is the greatest fixpoint operator. It follows from (2.2.9) that \( \nu X.pX \) is the greatest fixpoint of the map \( \lambda X.pX \), i.e.,

\[
(3.1.1) \quad \nu X.pX^m(\bar{A}) = \bigcup \{ B \mid B = p^m(B, \bar{A}) \} = \bigcup \{ B \mid B \in p^m(B, \bar{A}) \}
\]

and

\[
(3.1.2) \quad ((a)p)^m(\bar{A}) = [a]^m[\nu X.pX^m(\bar{A})]
\]

by (2.2.5), where

\[
[a]^m(B) = \{ s \mid \forall t, (s, t) \in (a) \rightarrow t \in B \} = S - (a^m)(S - B).
\]

It is easily proved that every \( L_\mu \) formula is equivalent to a formula over \( \land, \neg, \mu, \nu, (\cdot) \), and in which \( \neg \) is applied to primitive \( P \) only. Moreover, by renaming bound variables (Proposition 5.7(i) below), we can assume that no variable is quantified twice. Such a formula is said to be in positive normal form.

3.2. Closure

Let \( \rho_0 \) be a fixed closed formula in positive normal form. The following definitions introduce the closure \( \text{CL}(\rho_0) \) of \( \rho_0 \), the analog of the Fischer-Ladner closure of PDL [6]. For convenience, the closure is defined in terms of a mapping \( e \) on subformulas of \( \rho_0 \).

Let \( \sigma \) denote either \( \mu \) or \( \nu \). If \( X \) is a bound variable of \( \rho_0 \), there is a unique \( \mu \)- or \( \nu \)-subformula \( \sigma X.pX \) of \( \rho_0 \) in which \( X \) is quantified. We denote this subformula by \( \sigma X \) if \( \sigma X = \mu X.pX \) and a \( \nu \)-variable if \( \sigma X = \nu X.pX \).

Definition 3.2.1. Define \( p < q \) if \( q \) appears as a subformula of \( p \), and \( p < q \) if \( q \) appears as a proper subformula of \( p \). For \( p \leq q \), define \( \forall y_1, \ldots, y_m \geq 0 \), to be the sequence of all variables \( Y \) such that \( \sigma Y \leq \rho \), taken in the order

\[
\sigma Y_1 < \cdots < \sigma Y_m < \rho.
\]

For \( \bar{X} = X_1, \ldots, X_n \) a subsequence of \( \forall y \), and \( \bar{q} = q_1, \ldots, q_n \) a sequence of formulas, define

\[
p(\bar{X}/\bar{q}) = p[X_1/q_1, \ldots, X_n/q_n].
\]

where \( p[X/q] \) denotes the formula \( p \) with all free occurrences of \( X \) replaced by \( q \). Note that the order of substitution is from right to left.

Definition 3.2.2. If \( \forall y = X_1, \ldots, X_n \) let \( \forall y \) denote the sequence \( \sigma X_1, \ldots, \sigma X_n \).

Define the map \( e \) on subformulas of \( \rho_0 \) by

\[
e(p) = p[V_\rho/\sigma V_\rho].
\]

The closure of \( \rho_0 \) is the range of \( e \):

\[
\text{CL}(\rho_0) = \{ e(p) \mid \rho_0 \leq \rho \}.
\]

Note that \( e(p) \) is closed, since if \( X \) occurs free in \( p \), then \( \sigma X \leq \rho \). It is immediately clear from the definition of \( \text{CL}(\rho_0) \) that \( \text{CL}(\rho_0) \) is a finite set, and is in fact no larger than \( |\rho_0| \), the number of symbols of \( \rho_0 \). The next proposition relates \( \text{CL}(\rho_0) \) to the more usual notion of closure, as found for example in [6].

Proposition 3.2.3. \( \text{CL}(\rho_0) \) is the smallest set of closed formulas such that

(i) \( \rho_0 \in \text{CL}(\rho_0) \),

(ii) \( \neg \rho \in \text{CL}(\rho_0) \),

(iii) \( p \in \text{CL}(\rho_0) \),

(iv) \( \rho \land \sigma \in \text{CL}(\rho_0) \),

(v) \( \rho \lor \sigma \in \text{CL}(\rho_0) \),

(vi) \( \rho \rightarrow \sigma \in \text{CL}(\rho_0) \),

(vii) \( \rho \land \sigma \in \text{CL}(\rho_0) \),

(viii) \( \forall \rho \in \text{CL}(\rho_0) \).

Proof. It immediately follows from Definition 3.2.2 that

(i) \( e(p) = p \) if \( p \) is closed,

(ii) \( e(p \lor q) = e(p) \lor e(q) \),

(iii) \( e(p \land q) = e(p) \land e(q) \),

(iv) \( e(\sigma p) = (\sigma e(p)) \),

(v) \( e(\forall y p) = [\forall y e(p)] \),

(vi) \( e(\forall y p) = [\forall y e(p)] \).

where, in (xii), \( \sigma X = \forall X \). Cases (i) and (ii) are immediate from (viii). For case (iii), suppose \( p \lor q \in \text{CL}(\rho_0) \). Then \( \forall y \rho \in \text{CL}(\rho_0) \) for some subformula \( p \lor q \) of \( \rho_0 \). By (ix), \( p = e(p') \) and \( q = e(q') \), therefore \( \forall y \rho \in \text{CL}(\rho_0) \). Cases (iv)–(vii) are similar. For case (viii), suppose \( \forall X.pX \in \text{CL}(\rho_0) \). Formula \( \forall X.pX \) has exactly two pre-images under \( e \), namely \( X \) and \( \forall X = \forall X.pX \). Then \( \forall y \rho \leq p' X \), and

\[
e(p' X) = p' X[X/\sigma X][V_\rho/\sigma V_\rho] \leq p'[\forall X.pX][V_\rho/\sigma V_\rho] = p(\forall X.pX).
\]

Therefore, \( p(\forall X.pX) \in \text{CL}(\rho_0) \).

3.3. Active variables and aconjunction

Definition 3.3.1. Let \( \rho_0 \) be in positive normal form, \( \rho_0 < \rho \). A variable \( Y \) of \( \rho_0 \) is called active in \( p \) if \( \sigma Y < \rho \) and \( \forall y X \neq \bar{X} \) contains a free occurrence of \( Y \), where \( \bar{X} \) is the subsequence of \( V_\rho \) consisting of those variables \( X \) for which \( \sigma Y < \sigma X < \rho \).
The subsequence of \( V_x \) consisting of the active variables of \( p \) is denoted \( A_\nu \). The subsequence of \( A_\mu \) consisting of the active \( \mu \)-variables (resp. \( \nu \)-variables) is denoted \( \Delta_\nu \) (resp. \( \Delta_\mu \)).

If \( X \) is free in \( p \); then \( X \) is active in \( p \), but not vice versa in general; e.g., in \((3.3.2) \mu X.vY(x \mu Z((a) Yv[b]Z)), X \) is not free in \((a) Y \) but is active in \((a) Y \), since

\[
(a) Y[\sigma Z] = (a) Y[\sigma Y] = (a) vY(x \mu Z((a) Yv[b]Z))
\]

contains a free occurrence of \( X \). However, the relation ‘is active in’ is somewhat like the transitive closure of the relation ‘is free in’, in the following sense.

**Lemma 3.3.3.** If \( Y \) is active in \( p \), and \( X \) is active in \( \sigma Y \), then \( X \) is active in \( p \).

**Proof.** Let \( \tilde{X} = X_1, \ldots, X_m \tilde{Y} = Y_1, \ldots, Y_n \sigma > 0, \sigma > 0, \) be all variables such that

\[
\sigma X < \sigma X_1 < \cdots < \sigma X_m < \sigma Y_1 < \cdots < \sigma Y_n < p.
\]

Then \( Y \) is free in \( \rho[\tilde{Y}/\sigma \tilde{Y}] \) and \( X \) is free in \( \sigma Y(X/\sigma X) \); therefore \( X \) is free in \( \rho[\tilde{Y}/\sigma \tilde{Y}] \sigma Y[\tilde{X}/\sigma \tilde{X}] \sigma Y[\tilde{X}/\sigma \tilde{X}] \).

The problem of determining whether \( X \) is active in \( p \) can be reformulated as a transitive closure problem, and any standard algorithm for computing the transitive closure of a binary relation will be efficient enough for our purposes.

**Definition 3.3.4.** Let \( p_\mu \) be in positive normal form. \( p_\mu \) is aconjunctive in the \( \mu \)-variable \( X \) if, whenever \( p_\mu < p \land q \), \( X \) is active in at most one of \( p, q \); \( p_\mu \) is aconjunctive if it is aconjunctive in every \( \mu \)-variable.

Example (3.3.2) above is not aconjunctive, because \( X \mu Z((a) Yv[b]Z) \) is a subformula of (3.3.2), and the \( \mu \)-variable \( X \) is active in both \( X \) and \( \mu Z((a) Yv[b]Z) \).

Aconjunctivity is a technical restriction that is used in the proof of Theorem 6.3.1. It is related to, albeit weaker than, syntactic continuity. It is difficult to give the intuition behind the concept of aconjunctivity out of context; we therefore defer further explanation until Section 6.

4. Expressiveness results

\( L_\mu \) subsumes PDL without the reverse operator, as noted by Pratt [19]. The only least fixpoints PDL can express are of the form \((a^*)p \), which in \( L_\mu \) is expressed \( \mu X.pv(a)X \). Thus \((a^*)p \) is the least fixpoint of the monotone operator \( \lambda X.pv(a)X \).

This operator is continuous in \( X \), in the sense that

\[
p \vee (a) \left( \bigcup A \right) = \bigcup \left( p \vee (a) A \right).
\]

If any model \( M \), if \( pX \) is continuous in \( X \), then

\[
\mu X.pX^M = \omega X.pX^M,
\]

i.e., the inductive definition of \( \mu X.pX \) given in (2.2.7) above need not go beyond \( \omega \). However, there are many non-continuous operators that are potentially useful in program verification. An interesting example is provided by the operator \( \lambda X[a]X \).

Its least fixpoint in any model \( M \) is

\[
\mu X[a]X^M = \{ x \mid \text{there are no infinite } a \text{-paths out of } x \} = \neg \Delta a,
\]

where \( \Delta \) is the loop operator of Streett [21]. \( \mu X[a]X \) is a well-formed formula of \( L_\mu \), even under the restriction of aconjunctivity, but is illegal in Pratt’s system. In the model of Fig. 1, the operator \( \lambda X[a]X \) does not close at \( \omega \), since the top state satisfies \((\omega + 1)X[a]X \) but not \( \omega X[a]X \). Thus \( \lambda X[a]X \) is monotone but not continuous.

There are many useful properties that can be expressed with non-continuous operators, including liveness and fairness properties. The prototype liveness property

\[
| s |
\]

\[
\begin{array}{cccccccc}
  a & a & a & a & a & a & a & a \\
  a & a & a & a & a & a & a & a \\
  a & a & a & a & a & a & a & a \\
  a & a & a & a & a & a & a & a \\
  a & a & a & a & a & a & a & a \\
\end{array}
\]

Fig. 1.
'along every a-path, p must eventually come true' is expressed as $\mu X.p \lor [a]X$ in $L\mu$. We refer the reader to [4] for further examples.

The question raised by Pratt about the strict expressiveness of $P\mu$ over PDL is still open, but the following result of Streeter shows that $L\mu$, even without conjunctive formulas, is strictly more expressive than PDL. The proof also reveals why filtration techniques, which are used to obtain completeness and completeness results for PDL, fail for $L\mu$.

**Proposition 4.1** ([21]). $\mu X.[a]X$ is not equivalent to any PDL formula.

**Proof.** Suppose $\mu X.[a]X = p$ in all models, where $p$ is a formula of PDL. In the model $M$ of Fig. 1, $s \models \mu X.[a]X$, therefore $s \models p$. The proof of the small model property of PDL [6] allows $M$ to be collapsed to a finite model $N$ by identifying states that are indistinguishable by formulas of FL($p$), the Fitch-Ladner closure of $p$. If $[t]$ is the equivalence class of $t$ in the collapsed model, then $N, [t] \models q$ iff $M, s \models q$ for any $q \in$ FL($p$). In particular, $[\top] \models p$. But $[\top]$ cannot satisfy $\mu X.[a]X$, since the collapsing must have created a loop, therefore there is an infinite a-path out of $[\top]$. □

The above proof assumes that $\mu X.[a]X = p$ in all models and derives a contradiction. However, it is possible to show that $L\mu$ is strictly more expressive than PDL in the stronger sense that there is a model $M$ and a formula $q$ of $L\mu$ such that no PDL formula $p$ is equivalent to $q$ on $M$.

**Proposition 4.2** ([12]). In the model given by Fig. 2, the formula $\mu X.[a]X$ defines the even states, whereas all PDL formulas, even with test and reverse, define only finite and cofinite sets.

![Diagram of a model with a path labeled with 'a' and transitions labeled 'a' and 'b'.](image)

Intuitively, PDL cannot simulate an unbounded alternation of $[a]$ and $(a)$.

Full $L\mu$ encodes $\Delta P\mu$ of Streeter [21], since $[\alpha X.(a)X]$ is the restriction of a-conjunctive, $L\mu$ can be shown to encode well-structured $\Delta P\mu$, which is $\Delta P\mu$ with the $*$ and $\lor$ operators constrained to appear only in the context of the deterministic program constructors.

If $p$ then $a$ else $b = p?; a \lor \neg p?; b$ and while $p$ do $a = (p?; a)^*; \neg p?$. Primitive programs need not be deterministic (see [8]).

5. A deductive system

The deductive system is equational, as in [15], involving equations $p = q$ and implications $p \Rightarrow q$. The latter can be considered as an abbreviation for $p \lor q = q$. The logical axioms and rules are those of equational logic, including substitution of equals for equals, provided the syntactic restrictions on $\mu$ formulas are not violated. The nonlogical axioms are the following:

1. **(5.1) Axioms of Boolean algebra.**
2. **(5.2) $\alpha X \lor (\alpha Y = (\alpha X \lor \neg Y)).$**
3. **(5.3) $\alpha X \land [a]Y = (\alpha X \land Y).$**
4. **(5.4) $\alpha X = 0.$**
5. **(5.5) $p(\mu X.pX)$ is $\mu X.pX, \mu X.pX$ free for X in pX.$**
6. **(5.6) $pY \Rightarrow \mu X.pX \Rightarrow Y.$ Y does not occur in pX.$**

A formula $p$ is *consistent* if not $\vdash p = 0$. Axioms (5.1)–(5.4) are those of propositional modal logic. Axioms (5.5) and (5.6) say that $\mu X.pX$ is the least object $A$ such that $\mu X.pX = A$. Axiom (5.6) is the fixpoint induction rule of Park [17].

The following are some basic theorems of this system. The reader is referred to [1, 20] for the proofs, which are omitted here.

**Proposition 5.7.** (i) (Change of bound variable)

$$\mu X.pX = \mu Y.pY, \quad X, Y \text{ free for Z in pZ},$$

(ii) $pX \Rightarrow qX \Rightarrow \sigma X.pX \Rightarrow \sigma X.qX$,

(iii) (Monotonicity)

$$q \Rightarrow r \Rightarrow p(q) \Rightarrow p(r), \quad X \text{ positive in pX},$$

(iv) $p(\sigma X.pX) = \sigma X.pX, \quad \sigma X.pX \text{ free for X in pX},$

(v) $\mu X.q = q, \quad X \text{ not free in q},$

(vi) $p(\mu X.q \lor pX) \Rightarrow q \Rightarrow \mu X.pX \Rightarrow q, \quad q \text{ free for X in pX}.$

**Proof of (vi)**

(a) $p(\mu X.q \lor pX) \Rightarrow q$ (by assumption),

(b) $p(q \land \mu X.q(p(q \land X))) \Rightarrow q$ (by (a), (5.1) and (iii)),

(c) $p(q \land \mu X.q(p(q \land X))) \Rightarrow q \land p(q \land \mu X.q(p(q \land X)))$

(by (b) and (5.1)),

(d) $p(q \land \mu X.q(p(q \land X))) \Rightarrow \mu X.q(p(q \land X))$ (by (c) and (5.5)).
6. Complexity and deductive completeness

In this section we prove completeness of the deductive system of Section 5 and give an exponential time decision procedure and small model property for $L^u$ under the restriction of admissibility. $L^u$ is decidable without this restriction [15], but is not known to be elementary. These results are proved simultaneously, using a tableau method.

6.1. Construction of the tableau

Let $\mu$ be in positive normal form. In this section we construct a tableau $T$ for $\mu$. $T$ is a labeled tree constructed inductively downward by applying the extension rules described below. Certain edges of $T$ will be labeled with primitive programs, others will be unlabelled. Each node $s$ of $T$ will be labeled with a set $\Gamma_s$ of subformulas of $\mu$.

Initially, $T$ consists of a single node $\mu^0$ labeled $\{\mu_0\}$. The tree is extended downward by applying the following five extension rules to the leaves, in an order to be specified later.

6.1.1 $\land$-rule. If $p \land q \in \Gamma_s$, create node $s$ with $\Gamma_s = (\Gamma_s - \{p \land q\}) \cup \{p, q\}$ and unlabelled edge $s \rightarrow t$.

6.1.2 $\lor$-rule. If $p \lor q \in \Gamma_s$, create two new nodes $t, u$ with $\Gamma_t = (\Gamma_s - \{p \lor q\}) \cup \{p\}$, $\Gamma_u = (\Gamma_s - \{p \lor q\}) \cup \{q\}$ and unlabelled edges $s \rightarrow t, s \rightarrow u$.

6.1.3 $\sigma$-rule. If $\sigma X. pX \in \Gamma_s$, create $s$ labeled $\Gamma_s = (\Gamma_s - \{\sigma X. pX\}) \cup \{pX\}$ and unlabelled edge $s \rightarrow t$.

6.1.4 $X$-rule. If $X \in \Gamma_s$, and if $\sigma X. pX \in \Gamma_s$, create $t$ labeled $\Gamma_t = (\Gamma_t - \{X\}) \cup \{pX\}$ and unlabelled edge $s \rightarrow t$.

6.1.5 $\bot$-rule. For each $\bot p \in \Gamma_s$, create $t$ labeled $\Gamma_t = \{p\} \cup \{q\} \cup \{b \mid b \in \Gamma_b\}$ and edge $s \rightarrow t$ labeled $b$.

Note that the $\lor$-rule creates two new successors, the $\bot$-rule creates a new successor for each formula of the form $(b)p$, and all other rules create one new successor. In the last case, the unique successor of $s$ is denoted $s^+$.

The construction process maintains several lists $C$ of integer counters $c$, which count applications of the $X$-rule to active variables of formulas in $\Gamma_s$. There is one list $C(s, p)$ for each $p \in \Gamma_s$ and the lists are disjoint. If $A_s = X_1, \ldots, X_m$, then $C(s, p) = (c_1, \ldots, c_n)$, where $c_i$ counts applications of the $X$-rule to $X_i$. The counter $c_i$ is associated with $X_i$ throughout its lifetime. We denote this correspondence by $X(c_i) = X_i$. In general, there may be several counters at node $s$ associated with the same variable $X$, since $X$ may be active in several formulas of $\Gamma_s$, but these counters will appear on different lists.

The integer value contained in $c_i$ at node $s$ is denoted $c_i(s)$. If $X(c)$ is a $\sigma$-variable, $c$ is called a $\sigma$-counter, and $c(s)$ will always fall in the interval $0 \leq c(s) \leq 2^{lu}$. If $X(c)$ is a $\lor$-variable, $c$ is called a $\lor$-counter, and $c(s) \in \{0, 1\}$. A $\lor$-counter $c$ is used only as a one-bit flag to determine how recently the $\sigma$- or $X$-rule has been applied to $X(c)$.

If $C$ is a list, let $C_\mu$ (resp. $C_\lor$) denote the sublist of $C$ consisting of all $\sigma$-counters (resp. $\lor$-counters). The construction process also maintains a global list $G$ consisting of all existing $\sigma$-counters. $G(s)$ is a shuffle-merge of the lists $C_\mu(s, p), p \in \Gamma_s$. Thus the order of the $\sigma$-counters in $G$ is consistent with their order on the lists $C_\mu$. Whereas the order of the counters on $C$ is static and determined by the order $<_{\sigma}$ on $\sigma X(c)$, the order on the global list $G$ is dynamic and depends on the construction up to that point. $G(s)$ imparts a priority to the $\sigma$-counters existing at $s$, with the leastmost of highest priority.

The lists and counters are maintained as follows. We start with a single list $C(\rho, \mu^0)$ at the root, and $C(\rho, \mu^0) = C(\rho, \mu) = (\bot)$, since $\mu$ has no active variables. The lists and counters are updated at each application of an extension rule as follows.

6.1.6 When the $\land$-rule is applied to $\sigma X. pX$ at node $s$, recall that $\Gamma_s$, is obtained from $\Gamma_s$ by replacing $\sigma X. pX$ with $pX$. If $X$ is free in $pX$, then $pX$ has a new active variable that was not active in $\sigma X. pX$, namely $X$. A new counter $c$ is created with $X(c) = X$ and $c(s+) = 0$, and we append $c$ to the right end of $C(s, \sigma X. pX)$ to get $C(s+, pX)$. If $X$ is a $\sigma$-variable, the new counter is also appended to the right end of $G$, indicating lowest priority. If $X$ is not free in $pX$, then we take $C(s+, pX) = C(s, \sigma X. pX)$ and $G(s+) = G(s)$, but by Proposition 5.7(v) we can assume w.l.o.g. that this does not happen.

6.1.7 When the $\lor$-rule is applied to $p \lor q$ at node $s$ with successors $t, u$ as in (6.1.2), recall that the formula $p \lor q$ replaces $\bot p \lor q$ in $\Gamma_s$ and $q$ replaces $\bot q$ in $\Gamma_s$. We obtain $C(t, p)$ (resp. $C(u, q)$) from $C(s, p \lor q)$ by deleting all counters $c$ such that $X(c)$ is not active in $p$ (resp. $q$). Any deleted $\sigma$-counters also disappear from the global lists $G(t)$ and $G(u)$.
(6.1.8) When the $\wedge$-rule is applied to $p \land q$ at node $s$, then we obtain $C(s^+, p)$, 
(resp. $C(s^+, q)$) from $C(s, p \land q)$ by deleting all counters $c$ such that $X(c)$ is not 
in $p$ (resp. $q$). The global list $G$ remains unchanged. It is here that the 
condition of confluence is used: whereas a $\wedge$-counter on $C(s, p \land q)$ may appear 
on both $C(s^+, p)$ and $C(s^+, q)$, $C_{\mu}(s, p \land q)$ cleanly splits into disjoint lists 
$C_{\mu}(s^+, p)$ and $C_{\mu}(s^+, q)$, since each $\mu$-variable active in $p \land q$ is active in exactly 
one of $p, q$. If confluence was not satisfied, the $\mu$-counters on $G$ would have 
to be duplicated.

(6.1.9) When the $X$-rule is applied to a variable $X$ at $s$, and $\alpha X = \alpha X.pX$, take 
$cX = cX.pX$, and set $c(s^+) = c(s) + 1$, where $c$ is the unique counter on 
$C(s^+, pX)$ and $C(s, X)$ such that $X(c) = X$. Note that $c$ appears rightmost on these 
lists, since $\sigma Y < \alpha X$ for all variables $Y$ active in $X$. If $X$ is a $\mu$-variable, we reset 
all $\mu$-counters of lower priority than $c$ to 0 (recall that $d$ is of lower priority than 
$c$ if it appears to the right of $c$ on the global list $G$). We also reset to 0 any $\nu$-counter 
appearing on any $C(s^+, p)$ to the right of some $\mu$-counter that is incremented or 
reset to 0.

(6.1.10) When the $\cdot$-rule is applied at $s$, then for any successor $t, \Gamma_s$, of the form 
$b[i_1, \ldots, i_n]$, where $(b[i], b[i], \ldots, b[i]) \in \Gamma$. Take $C(t, p) = C(s, b[i])$ and $C(t, q) = 
C(s, b[i])$, 1 $\leq i < n$. $G(t)$ is obtained from $G(s)$ by deleting all counters not 
appearing on $C(t, p)$ or some $C(t, q)$. All $\nu$-counters are reset to 0.

(6.1.11) If $p \in \Gamma_s$ and the $\wedge$, $\cdot$, $\sigma$, or $X$-rule is applied at $s$ to some $q \neq p$, and $t$ 
is a successor of $s$, then $p \in \Gamma_t$. In case we take $C(t, p) = C(s, p)$ and leave all 
counters on $C(t, p)$ intact.

(6.1.12) After updating the lists according to (6.1.11)--(6.1.11), $C(t, p)$ may be 
temporary ill-defined. For example, if $p, p \land q \in \Gamma$, and the $\wedge$-rule is applied to $p \land q$, 
then (6.1.8) defines $C(s^+, p)$ to be a sublist of $C(s, p \land q)$, but (6.1.11) defines 
$C(s^+, p) = C(s, p)$. For another example, if $(b[i] b[i] \ldots b[i]) \in \Gamma$, and the $\cdot$-rule is applied 
then, at the successor $t$ corresponding to $(b[i] b[i] \ldots b[i])$, (6.1.10) defines $C(t, p) = C(s, b[i])$ and 
$C(t, q) = C(s, b[i])$. Whenever such a conflict occurs, the list of higher priority is 
kept and the other is discarded, where the priority of a list is determined by the 
position in $G$ of its highest priority $\mu$-counter. If the lists contain no $\mu$-counters, 
say $C' = (c_1, \ldots, c_n)$ and $C'' = (c'_1, \ldots, c'_n)$, then we discard $C''$ and 
set $c_i := \max(c_i, c'_{i})$, 1 $\leq i < n$.

Whenever two lists $C', C''$ are in conflict and $C'$ is the one that is discarded, we 
write $C' \rightarrow C$ and $C'' \leftrightarrow C$ for $c 
\in C', c \in C$ with $X(c') = X(c)$.

(6.1.13) Whenever a $\mu$-counter changes priority due to the deletion of a higher 
priority $\mu$-counter, it is reset to 0. Whenever a $\mu$-counter $c \in C$ is incremented or 
reset to 0, and $d$ is a $\nu$-counter appearing to the right of $c$ on $C$, then $d$ is also 
reset to 0.

### 6.2. The algorithm

We now describe an alternating Turing machine algorithm to construct the tableau. 
The algorithm starts with one process at the root $\tau$. It then applies the extension 
rules in a regular fashion, accepting or rejecting on certain conditions described 
below. When visiting node $s$ of $T$, a process has representation of $\Gamma$, written on its 
tape. It also maintains all the lists of counters as described above. At applications 
of the $\cdot$-rule, it makes an existential branch, spawning two subprocesses, each taking 
one of the successors. At applications of the $\cdot$-rule, it branches universally, 
spawning several processes, one for each successor.

At any node, the $\wedge$, $\cdot$, $\sigma$, and $X$-rules are applied first. The $X$-rule may only 
be applied to a $\sigma$-variable $X \in \Gamma$, if $c(s) = 0$, where $c \in C(s, X)$ and $X(c) = X$.

Whenever one of the following conditions obtains, the process takes the indicated 
action.

(6.2.1) There exist $P, \neg P \in \Gamma$, Halt and reject.

(6.2.2) Some $\mu$-counter exceeds $2^{n^3}$. Halt and reject.

(6.2.3) The only rule that applies is the $\cdot$-rule (i.e., $\Gamma$, contains only formulas of 
the form $P, \neg P, (a)p, (a)p_i$, or $\nu$-variables $X$ whose counters are nonzero), and 
neither of the previous conditions holds. Apply the $\cdot$-rule.

(6.2.4) No rule applies and none of the previous conditions hold. Halt and accept.

Let $|G|$ denote the maximum length of $G(s)$. Since $G(s)$ is a shuffle of at most 
$|p_0|$ lists $C_0(s, p)$ and each $|C_0(s, p)| \ll |p_0| |G| \ll |p_0|^3$. The above algorithm requires 
at most $|p_0|^3$ space, enough to encode $\Gamma$, and $|G| \ll |p_0|^3$ counters, each containing 
a nonnegative integer at most $2^{n^3}$. Despite the possibility of infinite computations, 
this alternating algorithm can be simulated in deterministic exponential time [3].

The next lemma is used here to show that one of the conditions (6.2.1)--(6.2.4) 
must obtain after a finite time. The lemma is used again in Section 6.3.

**Definition 6.2.5.** Let $s = s_0, s_1, s_2, s_3, \ldots, s_n = t$ be nodes along some path in $T$ 
such that $s_i$ is an immediate successor of $s_{i-1}$, 1 $\leq i < n$. Let $c = c_1, \ldots, c_n$ be counters 
such that $c_i$ exists in the interval $[s_i, s_{i+1})$, and $c_i = c_i$, at $t$, (therefore $c_i$ no longer 
exists at $s_i$). Let $a_i$ be the number of times $c_i$ is incremented in the interval $[s_i, t]$, and 
define

$$a_i = \sum_{t \in [s_i, t]} a_i.$$

**Lemma 6.2.6.** If either (i) $c$ is a $\mu$-counter, or (ii) $c$ is a $\nu$-counter and the $\cdot$-rule is 
not applied in the interval $[s, t]$, then

$$a_i \ll |p_i| |2^{n^3}|.$$
Proof. (i) Let $c$, $s$, $t$, $c_1$, $s$, $t_i$, $s_i$, $t_i$ be as in Definition 6.2.5. Note that $X(c) = X(c_i)$, $1 \leq i \leq n$, and let $p \in F_s$, such that $c \in C(p, s)$. Using Lemma 3.3.3, it can be shown that $d_i$ exists throughout the interval $[s, t_i]$ leftmost on the same list as $c_i$, and $d_i \rightarrow \cdots \rightarrow d_i$. Since the priority of $d_i$ never decreases, and $d_{i+1}$ is of higher priority than $d_i$, the sequence $d_i \rightarrow \cdots \rightarrow d_i$ does not increase to $[G]$. Let $N = 2^n + i$, the maximum value of $c_i$ in the interval $[s, t_i]$. $c_i$'s priority can increase at most $[G]$ times. Between priority changes, whenever $c_i$ is reset to 0, a counter of higher priority is incremented. Thus $c_i$ can be incremented or reset to 0 at most $N^M$ times before either $c_i$ or a higher priority counter exceeds $N$ and condition (6.2.2) obtains, causing the process to halt and reject. Thus $c_i$ can change priority, be reset, or be incremented at most $[G][N^M]$ times. This gives an upper bound on the number of Definition 6.6.2.5, thus

$$\alpha(c, s, t) \leq [G][N^M] = |\alpha|^2/[G].$$

(ii) If there exists a $\mu$-variable $Y$ active in $X(c)$, then for each $i$, there exists a $\mu$-counter $d_i$ appearing leftmost on the same list as $c_i$ throughout the interval $[s, t_i]$. As above, the length of the sequence $c_i \rightarrow \cdots \rightarrow c_i$ is at most $[G]$. Within the interval $[s, t_i]$, $c_i$ can be reset to 0 on only if the $X$-rule is applied (6.1.10) or some $\mu$-counter to the left of $c_i$ is incremented or reset to 0 (6.1.13). The former does not occur by assumption. The latter occurs only if the rightmost $\mu$-counter to the left of $c_i$ is incremented or reset to 0. By (i), this can happen at most $[G][N^M]$ times, from which the bound follows.

If there does not exist a $\mu$-variable active in $X(c)$, then $c$ cannot be reset in the interval $[s, t]$, since neither $\alpha(c, s, t)$ nor $\alpha(c, s, t)$. Thus $c$ and $c_i$ can be incremented at most once in $[s, t]$, since the $X$-rule is never applied when a counter is nonzero, and therefore $\alpha(c, s, t) \leq 1$.

Lemma 6.2.7. One of conditions (6.2.1)-(6.2.4) must obtain after a finite time.

Proof. Suppose there were an infinite path in $T$ with the $\nu$, $\lambda$, $\sigma$- and $X$-rules applied along that path without one of (6.2.1)-(6.2.4) ever obtaining. Each rule except the $X$-rule decreases the size of $\Gamma_n$ as measured by the total number of symbols in $\Gamma_n$; therefore there must exist a variable $X$ to which the $X$-rule is applied infinitely often. Moreover, $X$ can be chosen such that $\sigma X$ is $<\sigma$-minimal.

By Lemma 6.2.6, each $c$ with $X(c) = X$ that exists along the path must disappear after a finite time. This says that a new counter for $X$ is created infinitely often through application of the $\sigma$-rule. But then there must be a $Y$ with $\sigma Y < \sigma X$ such that the $X$-rule is applied to $Y$ infinitely often along the path, contradicting the $<\sigma$-minimality of $X$. □

6.3. Proof of main theorem

The following theorem asserts the correctness of the algorithm and the completeness of the deductive system of Section 5 simultaneously.

Theorem 6.3.1. The following are equivalent:
(i) $\eta_n$ is consistent,
(ii) the algorithm does not reject,
(iii) $\eta_n$ has a finite-tree-like model of depth exponential in $|\alpha|$.

Proof of (i)⇒(ii). Suppose $\eta_n$ is consistent. First we construct a formula $e'(s, p)$ for each $p \in F_s$, such that $e'(s, p) \leq e(p)$. $e'(s, p)$ is formed by joining certain closed formulas $r(s, c) \in C_p(s, p)$ (to be defined later) with certain subformulas of $e(p)$, as follows. Let $\eta = X_1, \ldots, X_n$. For each $X \in A_{\eta_n}$, define $C_p(s, p)$ with $X(c) = X_1$ and $\sigma X = \mu X_2$, let $q_1 = \mu X_2 (r(s, c) \land q X)$. For $X \in \eta$, let $q_1 = \sigma X$. Define

$$e'(s, p) = \eta'[\overline{q_1}].$$

By Proposition 5.7(iii), $e'(s, p) < e(p)$.

Each $r(s, c)$ consists of a conjunction of closed formulas, defined inductively down the tree. If neither the $\sigma$- nor the $X$-rule is applied at $s$, or if the $\eta$- or $X$-rule is applied to a $\nu$-variable, let

$$(6.3.2) r(s, c) = r(s, c)$$

for all successors $t$ of $s$ and counters $c \in G(t)$. If the $\sigma$-rule is applied to $\mu X_2 p X$ at $s$, yielding a new counter $\alpha$ on $C(s+, p X)$ with $X(c) = X$, define

$$(6.3.3) r(s+, c) = \nu s+ t,$$

$$(6.3.4) r(s+, s) = r(s, s), \quad d \in C(s+, p X), d \neq c.$$

If the $X$-rule is applied to the $\mu$-variable $X$ at $s$, and $c \in C(s+, X)$ with $X(c) = X$, define

$$(6.3.5) r(s+, d) = r(s, d), \quad d \in C(s+, p X), d \neq c.$$

(6.3.6) $r(s+, d) = r(s, d)$ if $d$ is of lower priority than $c$.

(6.3.7) $r(s+, c) = r(s, c) \land \neg \Delta_{\eta}.$

where

$$\Delta_{\eta} = \{e'(s+, p) \mid p \in F_s, p \neq X\}.$$
Note that \( r(s, c) \) consists of a conjunction of \( \phi(s) \) closed formulas (by convention, \( \wedge \emptyset = 1 \)):

\[
\begin{align*}
  r(s, c) = & \quad \neg \Delta_i^e \wedge \cdots \wedge \neg \Delta_i^{n-1},
\end{align*}
\]

where \( s_i, 0 \leq i < c(s) \), is the most recent ancestor of \( s \) such that \( c \) had value \( i \).

Let

\[
\Delta_i = \{ e'(s, p) \mid p \in \Gamma_i \}.
\]

We now construct a set \( B \) of nodes of \( T \) containing the root \( r \), such that

\[
(6.3.8) \text{ if } s \in B \text{ and the } -v\text{-rule was applied at } s, \text{ then at least one successor of } s \text{ is in } B.
\]

(6.3.9) for any other node \( s \in B \), all successors of \( s \) are in \( B \),

\[
(6.3.10) \text{ for every } s \in B, \Delta_i \text{ is consistent.}
\]

The set \( B \) is constructed inductively down the tree. First set \( B \seteq \{ r \}; \Delta = \{ r \} \) is consistent by assumption.

Suppose \( s \in B \) and the \( -v\text{-rule} \) is applied to \( p \lor q \) at \( s \), and \( u, v \) are the two successors of \( s \).

If \( p \in \Gamma_i \), already, and \( C(s, p) \) is of higher priority than the sub-list of \( C(s, p \lor q) \) corresponding to the active variables of \( p \), then the latter list is deleted in (6.1.12), so that \( \Delta_i \subseteq \Delta_i^e \). Then \( \Delta_i \) is consistent since \( \Delta_i^e \), so we can extend \( B \) by taking \( B \seteq B \cup \{ t \} \). Similarly, if \( q \in \Gamma_i \) and \( C(s, q) \) is of higher priority, then we can take \( B \seteq B \cup \{ u \} \). If either of the above cases holds, then

\[
e'(s, p \lor q) \equiv e'(s, p) \lor e'(s, q), \quad \wedge \Delta_i \equiv \wedge \Delta_i^e \lor \wedge \Delta_i^e,
\]

By Axiom (5.1), one of \( \Delta_i, \Delta_i^e \) must be consistent, say \( \Delta_i \). Set \( B \seteq B \cup \{ t \} \).

Similarly, at applications of the \( \wedge\text{-}, \langle \rangle\text{-}, \neg\text{-} \) and \( -0\text{-rules} \), the \( -0\text{-rule} \) to \( -0\text{-variables} \), \( B \) can be extended to include all successors, since if \( \Delta_i \), then \( \Delta_i^e \), is consistent for all successors \( t \).

At an application of the \( -v\text{-rule} \) to a \( -v\text{-variable} \) at \( s \), a new \( r(s^+, c) \) appears, but it is true at that point, and so does not affect the consistency of \( \Delta_i^e \), by Proposition 5.7(iii).

At applications of the \( \wedge\text{-}, \langle \rangle\text{-}, \neg\text{-} \) and \( -0\text{-rules} \), the \( -0\text{-rule} \) to \( -0\text{-formulas} \), and applications of the \( \wedge\text{-}, \langle \rangle\text{-}, \neg\text{-} \) and \( -0\text{-rules} \), the \( -0\text{-rule} \) to \( -0\text{-variables} \) at \( s \), it is here that we use Proposition 5.7(iv).

\[
\begin{align*}
  \Delta_i^e \equiv \langle e'(s, X) \rangle & \equiv \langle \mu X. (r(s, c) \land p X) \rangle
\end{align*}
\]

Thus \( \Delta_i^e \) is consistent when \( \Delta_i \) is consistent when \( X \)-rule is applied to \( X \)-variable \( X \) at \( s \). It is here that we use Proposition 5.7(iv).

\[
\begin{align*}
  \Delta_i^e \equiv \langle e'(s, X) \rangle & \equiv \langle \mu X. (r(s, c) \land p X) \rangle
\end{align*}
\]

Thus \( \Delta_i^e \) is consistent when \( \Delta_i \) is consistent when \( X \)-rule is applied to \( X \)-variable \( X \) at \( s \). It is here that we use Proposition 5.7(iv).

\[
\begin{align*}
  \Delta_i^e \equiv \langle e'(s, X) \rangle & \equiv \langle \mu X. (r(s, c) \land p X) \rangle
\end{align*}
\]

Thus \( \Delta_i^e \) is consistent when \( \Delta_i \) is consistent when \( X \)-rule is applied to \( X \)-variable \( X \) at \( s \). It is here that we use Proposition 5.7(iv).

\[
\begin{align*}
  \Delta_i^e \equiv \langle e'(s, X) \rangle & \equiv \langle \mu X. (r(s, c) \land p X) \rangle
\end{align*}
\]

Thus \( \Delta_i^e \) is consistent when \( \Delta_i \) is consistent when \( X \)-rule is applied to \( X \)-variable \( X \) at \( s \). It is here that we use Proposition 5.7(iv).

\[
\begin{align*}
  \Delta_i^e \equiv \langle e'(s, X) \rangle & \equiv \langle \mu X. (r(s, c) \land p X) \rangle
\end{align*}
\]
We now define a model $M = (S, I)$ from $T'$. Let $S$ be the set of nodes of $T'$ such that either the $(\cdot)$-rule was applied at $s$, or no rule was applicable at $s$ (thus $s$ is a leaf). Each edge out of a node in $S$ is labeled with a unique program constant, and all other edges are unlabeled. For $x \in T'$, let $U(x)$ be the set of nodes of $T'$ consisting of $s$ and all ancestors on the path back up to, but not including, the most recent ancestor in $S$; or back up to and including the root, if no ancestor of $s$ is in $S$. Note that if $s \in U$ and $s \rightarrow u$ in $T'$, then by Lemma 6.2.7 there exists a unique node $t$ in $S$ such that $u \in U(t)$. For $x, y \in S$, let $(x, y) \in E(a)$ if there is an edge from $x$ to $a$ node in $U(t)$ labeled $a$. Let $s \in I(P)$ if $P \models a_{s}$. We construct a set closed formulas $\Theta$, of $L_{\mathcal{P}}$ for each $s \in T'$, as follows. Let $p \in P$, $t \models \phi = \phi \{ X, \ldots, X \}_{a} = \phi \{ X \}_{a} \{ p \}_{a}$, and let $c \in \mathcal{C}(s, p)$ with $X = X(c)$, let

$$a(s, c) = \sup_{a(t, c)} a(t, c, t), \quad q(s) = a(s, c)X_{p},$$

where $a(s, c, t)$ is given as in Definition 6.2.5. If $X_i \in V_{q} - A_{\mu}$, let $q_i = \phi X_i$. Let $q = q_1, \ldots, q_n$, and define

$$\epsilon(s, p) = \phi \{ \phi \{ X \}_{q} \}, \quad \Theta_i = \{ \epsilon(s, p) \}_{a} \{ p \}_{a}, \quad \Theta = \{ \Theta_i \}_{a} \{ a \}_{a} \{ \Theta \}_{a} \{ U(s) \}_{a}.$$

Define $p' \equiv p$ if $p'$ is identical to $p$ except that some $\alpha$ occurring in subformulas $\alpha X_i$ of $p$ may be replaced by some $\beta < a$. We show that, for all $s \in S$,

(6.3.16) if $p \equiv p \models a_{s} \equiv a_{s}$, then $p \models a_{s}$,

(6.3.17) if $p \equiv p \models a_{s} \equiv a_{s}$, then either $p \models a_{s} \equiv a_{s}$, or $a_{s} \models a_{s}$.

(6.3.18) if $a_{s} \models a_{s} \equiv a_{s}$, then $S \models a_{s} \equiv a_{s}$, $p \models a_{s} \equiv a_{s}$, and $p \models a_{s} \equiv a_{s}$,

(6.3.19) if $a_{s} \models a_{s} \equiv a_{s}$, then $S \models a_{s} \equiv a_{s}$, $p \models a_{s} \equiv a_{s}$, and $p \models a_{s} \equiv a_{s}$,

(6.3.20) if $a_{s} \models a_{s} \equiv a_{s}$, then $S \models a_{s} \equiv a_{s}$, $p \models a_{s} \equiv a_{s}$, and $p \models a_{s} \equiv a_{s}$.

Proof of (6.3.16). If $p \models a_{s} \equiv a_{s}$, then $S \models a_{s} \equiv a_{s}$, $p \models a_{s} \equiv a_{s}$, and $p \models a_{s} \equiv a_{s}$.

Proof of (6.3.17). The proof of (6.3.17) is similar to (6.3.16).

Proof of (6.3.18). If $(a) \models \Theta$, then $(a) \models \Theta$. By the $(\cdot)$-rule, some $\alpha$-successor $u$ of $s$ has $p \models a_{s} \equiv a_{s}$, and by Lemma 6.2.6 there is a unique descendant $t$ of $u$ with $u \in T'$ and $u \in U(t)$. Then $(s, t) \models I(a)$ and $a \models \Theta$.

Proof of (6.3.19). This proof is similar to that of (6.3.18), except that $p' \models a_{s}$ appears in $p'$, instead of $p$, because if $[\alpha]p \models a_{s}$, then $[\alpha]p \models a_{s}$, and $p \models a_{s}$ appears in $I'$, for all $\alpha$-successors $s$ of $s$, and

$$a(s, c) = \sup_{a(t, c)} a(t, c) \quad \text{for any } c \in \mathcal{C}(s, p'),$$

where the supremum is taken over all $\alpha$-successors $u$ of $s$. Some of these $a(t, c)$ may be strictly less than $a(s, c)$.

Proof of (6.3.20). Either $a = \mu$ or $a = \omega$. If $a = \mu$ and $Xp \models a_{s}$, then $\exists t \in U(s)$, $\exists p' \models Xp \models a_{s}$. Let $\mu Xp = \epsilon(t, X) \models a_{s} Xp \models a_{s}$, and the $\alpha$-rule applied to $\mu Xp$ at $t$. Then $p' \models Xp$, and

$$\epsilon(t', p') = p \{ \epsilon(t, c) \} \models Xp \models a_{s},$$

where $c = \epsilon(t, c) \models Xp \models a_{s}$, and $\epsilon(t', c) \models Xp \models a_{s}$.

Now define

$$q^\alpha = \{ \epsilon \models q \} \models q \in \epsilon \models q \models q^\alpha.$$
By definition of $M$

$$P^m = \{ p \in P \mid \exists \sigma \in \Sigma, \exists n \geq 0 : p = p^m \}$$

$$- P^m = \{ p \in P \mid \forall \sigma \in \Sigma, \forall n \geq 0 : p \neq p^m \}$$

$$X(q) = X^m(q)$$

For the case $p \lor q$

$$p \lor q(q)^m \subseteq p(q)^m \lor q(q)^m$$

$$\subseteq p^m(q)^m \lor q^m(q)^m$$

by induction hypothesis

$$= (p \lor q)^m(q)^m$$

by (2.2.3).

The case $p \land q$ is similar, using (6.3.16). For the case $(a) p$

$$(a) p(q)^m \subseteq (a^m)(p(q)^m)$$

$$\subseteq (a^m)(p^m(q)^m)$$

by induction hypothesis and the monotonicity of $(a^m)$

$$= (a)p^m(q)^m$$

by (2.2.5).

For the case $[a] p$

$$[a] p(q)^m \subseteq [a^m] \left( \bigcup_{n \in \omega} p(q)^m \right)$$

$$\subseteq [a^m][p(q)^m]$$

by monotonicity of $[a^m]$

$$\subseteq [a^m](p^m(q)^m)$$

by induction hypothesis and the monotonicity of $(a^m)$

$$= [a]p^m(q)^m$$

by (3.1.2).

For the case $a X p X$, where either $a = \mu$ or $a \in \omega$

$$a X p X(q)^m \subseteq p(B X p X(q), q)^m$$

for some $B < a$, by (6.3.20)

$$\subseteq p^m(B X p X(q), q)^m$$

by induction hypothesis on $p$

$$\subseteq p^m(B X p X^m(q), q)^m$$

by induction hypothesis on $B$ and the monotonicity of $p^m$

$$= (B + 1) X p X^m(q)^m$$

by (2.2.6b)

$$\subseteq a X p X^m(q)^m$$

since $B + 1 \leq a$. Finally, for the case $v X p X$

$$v X p X(q)^m \subseteq p(v X p X(q), q)^m$$

by (6.3.21)

$$\subseteq p^m(v X p X(q), q)^m$$

by induction hypothesis.

Results on the propositional $\mu$-calculus

By (3.1.1), $v X p X^m(q)^m$ is the greatest fixpoint of the operator $\lambda X p^m(X, q)^m$, therefore

$$v X p X(q)^m \subseteq v X p X^m(q)^m.$$ 

This completes the proof of (6.3.22).

Taking $p = \rho_0$ in (6.3.22), we get $\rho_0 \in \rho_0^m \subseteq \rho_0^m$, therefore $M, \rho_0^m \rho_0$. A finite tree-like model of the appropriate size can be obtained from $M$ by the technique in [8, 13].

Proof of (III) $\Rightarrow$ (I). This asserts the soundness of the deductive system and is left to the reader.

Acknowledgment

I would like to thank Rivi Sherman and Joe Halpern for pointing out mistakes in an earlier version [14], and Steve Bloom, Allen Emerson and Joe Halpern for valuable suggestions.

References