Strictness of the modal μ-Calculus Hierarchy

Luca Alberucci

Informatik und Angewandte Mathematik
Universität Bern
albe@iam.unibe.ch

1 Introduction

This chapter proves two main results. The translation from automata to modal μ-calculus and the hierarchy theorem for the modal μ-calculus.

The first main result was initially proven by Niwinski in [4]. He introduces automata on semi-algebras and shows the equivalence with certain fixpoint terms on, so-called, powerset algebras. By using the fact that on binary structures the μ-calculus corresponds to a certain powerset algebra, this result can then be applied to the modal μ-calculus to obtain the equivalence of alternating automata to the calculus, on binary transition systems.

We will give a direct translation of automata to μ-formulae, using the alternating automata model introduced by Wilke in [5], and discussed in this volume in chapter 9. The translation will be such that for every automaton there is a μ-formula which is equivalent to it on all transition systems. In this sense the new result we get is a generalization of the original one by Niwinski, since we are not restricting ourself to binary transition systems.

Combining our result with the one shown in chapter 10 in this volume we get the equivalence of automata with μ-calculus on all transition systems. The established equivalence will be such that the alternation depth of the μ-formulae corresponds to the index of the automata.

The second main result was first proved independently by Bradfield in [2] and Lenzi in [3]. Our proof of will follow the one of Arnold in [1], where the strictness of the hierarchy on binary trees is shown.

So we will first prove a hierarchy theorem for alternating automata. Then by using the equivalence result established previously, we get the hierarchy theorem for the modal μ-calculus.

This chapter is organised in five sections: Section 11.2 reviews the basic notions which will be needed later. Section 11.3 introduces some hierarchies, both for μ-formulae and for alternating automata. In Section 11.4 we will translate alternating automata to modal μ-calculus. And in Section 11.5 we will prove the hierarchy theorems for alternating automata and for the modal μ-calculus.

2 Preliminaries

This section reviews the notions which have previously been defined in this volume. In our context there are some subtle differences with definition and notation.

Starting from a set of **propositional variables** P we define as usual the class of all **μ-formulae** L_μ to be the smallest set such that $P \subseteq \mathrm{L}_\mu$, $\mathbf{0}, \mathbf{1} \subseteq \mathrm{L}_\mu$ and if φ, ψ are both contained in L_μ, then so are $\varphi \wedge \psi, \varphi \vee \psi, \neg\varphi, \Box\varphi, \Diamond\varphi$. Furthermore if X is a propositional variable, which appears only positively in a formula $\varphi \in \mathrm{L}_\mu$ (i.e. in the scope of an even number of negations) then $\nu X.\varphi \in \mathrm{L}_\mu$ and $\mu X.\varphi \in \mathrm{L}_\mu$, whereby we say that X is **bounded**, as opposed to being **free**. By $Free(\varphi)$ we denote all the free propositional variables of φ.

The formulae of L_μ are then interpretated in so called **transition systems**. A transition system $\mathcal{S}$ is of the form (S, R, λ), where S is the set of states, R is a binary relation on S and λ is a valuation, whereby for each $p \in P$ λ assigns a subset of S. A **pointed transition system**, $(\mathcal{S}, s_{\mathrm{I}})$, is a transition system $\mathcal{S}$ and a state $s_{\mathrm{I}} \in S$. If $\mathcal{S}$ is of the form (S, R, λ), $S' \subseteq S$ and $p \in P$, then the expression $\mathcal{S}[p \mapsto S']$ denotes the transition system $(S, R, \lambda[p \mapsto S'])$, where $\lambda[p \mapsto S']$ is defined as:

$$\lambda[p \mapsto S'](p') = \begin{cases} S' & \text{if } p' = p \\ \lambda(p') & \text{if } p' \neq p. \end{cases}$$

Given a μ-formula φ and a transition system $\mathcal{S}$, the set $\|\varphi\|_{\mathcal{S}} \subseteq S$ is defined as in Definition **??** of chapter 10, such that it denotes the states where φ holds. Suppose φ contains propositional variables $X_1, \ldots, X_n$ (often we will write $\varphi(X_1, \ldots, X_n)$) and that $S_1, \ldots, S_n$ are all sets of states of a given transition system $\mathcal{S}$. Then $\|\varphi(S_1, \ldots, S_n)\|_{\mathcal{S}}$ is equal $\|\varphi(X_1, \ldots, X_n)\|_{\mathcal{S}[X_1 \mapsto S_1, \ldots, X_n \mapsto S_n]}$.

Our definition of an **alternating tree automaton** is very similar to the one given in chapter 9. The three differences being that we always specify the set of propositional variables P, that the priority function may not be total and that we allow by definition what in chapter 9 is called complex transition conditions.

An alternating tree automaton $\mathcal{A}$ is a tuple $\mathcal{A} = (Q, P, q_{\mathrm{I}}, \delta, \Omega)$ where

- Q is a finite set of **states**,
- P is a set of propositional variables,
- $q_{\mathrm{I}} \in Q$ is the **initial state**,
- $\Omega : Q \to \omega$ is a (partial) **priority function** and
- $\delta : Q \to TC^{Q \cup P}$ is a **transition function**, where $TC^{Q \cup P}$ is the set of all **transition conditions** over $Q \cup P$ and is defined inductively as follows:
  - (i) $\mathbf{0}, \mathbf{1} \in TC^{Q \cup P}$,
  - (ii) $p, \neg p \in TC^{Q \cup P}$ for all $p \in P$,
  - (iii) If $q \in Q$ then $q, \Box q, \Diamond q \in TC^{Q \cup P}$,
  - (iv) If $d \in TC^{Q \cup P}$ and $d' \in TC^{Q \cup P}$ then $d \wedge d'$, $d \vee d' \in TC^{Q \cup P}$.

Notice that a transition conditon $\delta(q)$ can be interpreted as a μ-formula over propositional variables in $Q \cup P$. We will sometimes write $\delta_q(q_1, \ldots, q_n)$ if $\delta(q)$ can be interpreted as a μ-formula with variables from $q_1, \ldots, q_n$.

In chapter 9 the use of only simple transition conditions ((iv) allows only $d, d' \in Q$) is in fact no real restriction. Since there it can also be shown that an alternating automaton with complex transition conditions can be transformed into an equivalent one with only simple conditions. The **index** $\mathrm{ind}(\mathcal{A})$ of an automaton $\mathcal{A}$ is defined as follows:

$$\mathrm{ind}(\mathcal{A}) := \max\Big(\Big\{|\Omega(Q')| \,\Big|\, Q' \subseteq Q, Q' \text{ is strongly connected}\Big\} \cup \Big\{0\Big\}\Big).$$

The definition of **acceptance** of a pointed transition system $(\mathcal{S}, s_{\mathrm{I}})$ by an automaton $\mathcal{A}$ can be made in two ways. The first can be done as in chapter 9, with parity games. In this case since we have complex transition conditions we first have to transform an automaton to an equivalent simple one. The second way needs the notion of accepting **run** of an automaton over a transition system. We will first define runs only for automata with simple transition conditions. With the help of Lemma 1 and Exercise 2 the notion of run can be extended to automata with complex transition conditions.

Let $\mathcal{A}$ be an automaton with state q_0 and $\mathcal{S}$ be a transition system with state s_0. We define ϱ to be a q_0-run on s_0 of $\mathcal{A}$ on $\mathcal{S}$ if ϱ is a $(S \times Q)$-vertex-labeled tree of the form (V, E, ℓ), where V is the set of vertices, E is a binary relation on V, and ℓ is the labeling function. If v_0 is the root of V, then $\ell(v_0) = (s_0, q_0)$. Moreover for all vertices $v \in V$, with label (s, q), the following requirements must be fulfilled:

- $\delta(q) \neq \mathbf{0}$,
- If $\delta(q) = p$ then $s \in \|p\|_{\mathcal{S}}$, and if $\delta(q) = \neg p$ then $s \notin \|p\|_{\mathcal{S}}$,
- If $\delta(q) = q'$, then there is a $v' \in vE$ such that $\ell(v') = (s, q')$,
- If $\delta(q) = \Diamond q'$, then there is a $v' \in vE$ such that $\ell(v') = (s', q')$ where $s' \in sR$,
- If $\delta(q) = \Box q'$, then for all $s' \in sR$ there is a $v' \in vE$ such that $\ell(v') = (s', q')$,
- If $\delta(q) = q' \vee q''$, then there is a $v' \in vE$ such that we have $\ell(v') = (s, q')$ or $\ell(v') = (s, q'')$,
- If $\delta(q) = q' \wedge q''$, then there are $v', v'' \in vE$ such that $\ell(v') = (s, q')$ and $\ell(v'') = (s, q'')$.

where $vE = \{v' \in V \mid (v, v') \in E\}$, similarly for sR.

An infinite branch of a run is **accepting** if the highest priority which appears infinitely often is even. A run is accepting when all infinite branches are accepting. An automaton $\mathcal{A}$ accepts a pointed transiton system $\mathcal{S} = (S, s_{\mathrm{I}})$ if there is an accepting q_{I}-run on s_{I} of $\mathcal{A}$ on $\mathcal{S}$ (where q_{I} is the initial state of $\mathcal{A}$).

Let $\mathcal{A} = (Q, P, q_1, \delta, \Omega)$ be an automaton, $\mathcal{S} = (S, R, \lambda)$ a transition system and $\varrho = (V, E, \ell)$ a run. For all $v \in V$ and $q \in Q$ we have:

$$S_{vE|q} := \{s \in S \mid (s, q) \in vE\}$$

The following lemma can be proven by unwinding the definitions.

Lemma 1. *Let $\mathcal{A} = (\{q_1, \ldots, q_n\}, P, q_I, \delta, \Omega)$ be an automaton, $\mathcal{S} = (S, R, \lambda)$ a transition system and $\varrho = (V, E, \ell)$ a $(S \times Q)$-vertex-labeled tree with root v_0. The following two sentences are equivalent:*

- *$\varrho = (V, E, \ell)$ is a q_0-run on s_0 of $\mathcal{A}$ on $\mathcal{S}$.*
- *$\ell(v_0) = (s_0, q_0)$ and for all vertices v which are labeled by (s, q) we have $s \in \|\delta_q(S_{vE|q_1}, \ldots, S_{vE|q_n})\|_{\mathcal{S}}$.*

Exercise 2. Give a direct definition of run for automata with complex transition conditions, without using the notion of equivalent simple automaton. Show that your automaton accepts the same runs as the equivalent simple automaton. (Hint: Use the equivalence established in Lemma 1.)

3 Hierarchies

In this section we introduce some hierarchies, both for alternating automata and for μ-formulae.

Before we introduce a hierarchy on the formulae L_μ let us define two operators μ and ν on μ-formulae. Let Φ be a class of μ-formulae. We define $\mu(\Phi)$ to be smallest class of formulae such that the following requirements are fulfilled.

(i) $\Phi \subseteq \mu(\Phi)$ and $\neg\Phi \subseteq \mu(\Phi)$, where $\neg\Phi := \{\neg\varphi \mid \varphi \in \Phi\}$.
(ii) If $\psi \in \mu(\Phi)$ then $\mu X.\psi \in \mu(\Phi)$ (provided that each appearence of X in ψ is positive).
(iii) If $\psi, \varphi \in \mu(\Phi)$ then $\psi \wedge \varphi \in \mu(\Phi)$, $\psi \vee \varphi \in \mu(\Phi)$, $\Box\psi \in \mu(\Phi)$ and $\Diamond\psi \in \mu(\Phi)$.
(iv) If $\psi, \varphi \in \mu(\Phi)$ and $X \notin Free(\psi)$, then $\varphi[\psi/X] \in \mu(\Phi)$.

$\nu(\Phi)$ is defined analogously to $\mu(\Phi)$ with the only difference that (ii) is substituted by:

(ii') If $\psi \in \nu(\Phi)$ then $\nu X.\psi \in \nu(\Phi)$ (provided that each appearence of X in ψ is positive).

With the help of the previous definitions we introduce two modal μ-calculus hierarchies. The first one is on the syntactical side, i.e. it is a hierarchy of classes of μ-formulae, and the second one is on the semantical side, i.e. it is a hierarchy of classes of transition systems.

For all natural numbers n we define the classes of μ-formulae Σ_n^μ and Π_n^μ inductively.

- Σ_0^μ and Π_0^μ are equal and consist of all fixpointfree μ-formulae.
- $\Sigma_{n+1}^\mu := \mu(\Pi_n^\mu)$.
- $\Pi_{n+1}^\mu := \nu(\Sigma_n^\mu)$.

All Π_n^μ and Σ_n^μ form the (syntactic) modal μ-calculus hierarchy.

To define the semantical counterpart of this hierarchy, we introduce the class $\|\varphi\|$ for all μ-formulae φ, it consists of all pointed transition systems $(\mathcal{S}, s_I)$ such that $s_I \in \|\varphi\|_{\mathcal{S}}$. The semantical modal μ-calculus hierarchy consists now of all $\Sigma_n^{\mu\,\mathbf{TR}}$ and $\Pi_n^{\mu\,\mathbf{TR}}$, which are the following classes of pointed transition systems:

- ${\Sigma_n^\mu}^{\mathbf{TR}} = \{\|\varphi\| \mid \varphi \in \Sigma_n^\mu\}$.
- ${\Pi_n^\mu}^{\mathbf{TR}} = \{\|\varphi\| \mid \varphi \in \Pi_n^\mu\}$.

It is obvious that the following holds:

$$Fm(\mathrm{L}_\mu) \equiv \bigcup_{n\in\omega} \Sigma_n^\mu \equiv \bigcup_{n\in\omega} \Pi_n^\mu.$$

And moreover from the definitions above, we can easily prove that:

$$(\Sigma_n^\mu \cup \Pi_n^\mu) \subsetneq \Pi_{n+1}^\mu,$$

and that

$$(\Sigma_n^\mu \cup \Pi_n^\mu) \subsetneq \Sigma_{n+1}^\mu.$$

This clearly shows that on the syntactical side we have a strict hierarchy of formulae. Showing an equivalent result on the semantical side will be the second main result of this chapter.

Lemma 3. *The following holds for all natural numbers n:*

- ${\Sigma_n^\mu}^{\mathbf{TR}} = \{\|\neg\varphi\| \mid \varphi \in \Pi_n^\mu\}$,
- ${\Pi_n^\mu}^{\mathbf{TR}} = \{\|\neg\varphi\| \mid \varphi \in \Sigma_n^\mu\}$.

Proof. By Lemma **??**n chapter 20 we have for all transition system $\mathcal{S}$ and all formulae φ:

$$\|\neg\nu X.\varphi\|_{\mathcal{S}} = \|\mu X.\neg\varphi[\neg X/X]\|_{\mathcal{S}} \quad \text{and} \quad \|\neg\mu X.\varphi\|_{\mathcal{S}} = \|\nu X.\neg\varphi[\neg X/X]\|_{\mathcal{S}}.$$

With this fact we easily can prove the lemma.

Let us now introduce analogous notions for automata, where we will also talk of a syntactical and a semantical side. Let us first introduce a syntactical hierarchy, it consists of the following classes of alternating automata.

- $\Sigma_0 = \Pi_0$ consists of all autmata with index 0.
- $\Sigma_n (n > 0)$ is the class of all alternating automata of index $\leq n$ where the maximal priority on a strongly connected component of the transition graph of the automaton is odd.
- $\Pi_n (n > 0)$ is the class of all alternating automata of index $\leq n$ where the maximal priority on a strongly connected component of the transition graph of the automaton is even.

For the semantical part, if $\|\mathcal{A}\|$ is the class of all pointed transition systems accepted by an automaton $\mathcal{A}$, we define

- $\Sigma_n^{\mathbf{TR}} = \{\|\mathcal{A}\| \mid \mathcal{A} \in \Sigma_n\}$,
- $\Pi_n^{\mathbf{TR}} = \{\|\mathcal{A}\| \mid \mathcal{A} \in \Pi_n\}$.

We conclude this section by stating a lemma, which follows from the complementation Theorem for alternating automata in chapter 9.

Lemma 4. *For all automata $\mathcal{A} \in \Sigma_n (\in \Pi_n)$ there is an automaton $\hat{\mathcal{A}} \in \Pi_n (\in \Sigma_n)$ such that:*

$$\|\hat{\mathcal{A}}\| = \mathbf{TR} - \|\mathcal{A}\|$$

where $\mathbf{TR}$ *is the class of all pointed transition systems.*

Exercise 5. Suppose $\mathcal{A} \in \Sigma_n$. Show that there is an equivalent automaton $\mathcal{A}'$ (i.e. $\|\mathcal{A}\| = \|\mathcal{A}'\|$), where the range of the priority function is $\{0, \ldots, n-1\}$ if n is even, and $\{1, \ldots, n\}$ if n is odd. Formulate and show the analogous claim for Π_n-automata.

4 From alternating automata to μ-calculus

In this section we discuss how to translate alternating automata to μ-calculus formulae.

We begin with a reformulation of Theorem **??** in chapter 20, in terms of the μ-calculus.

Lemma 6. *Assume that we have μ-formulae $\delta_1(s_1, \ldots, s_k), \ldots, \delta_k(s_1, \ldots, s_k)$ contained in a class of functions Φ such that all s_j $(j = 1, \ldots, k)$ appear only positively. Moreover define for all transition systems $\mathcal{S} = (S, R, \lambda)$ a functional $\mathcal{F}_{\mathcal{S}} : \mathscr{P}(S^k) \to \mathscr{P}(S^k)$ as follows:*

$$\mathcal{F}_{\mathcal{S}} : (S_1, \ldots, S_k) \mapsto (\|\delta_1(S_1, \ldots, S_k)\|_{\mathcal{S}}, \ldots, \|\delta_k(S_1, \ldots, S_k)\|_{\mathcal{S}}).$$

There are μ-formulae $\tau_1, \ldots, \tau_k$ in $\mu(\Phi)$ and $\rho_1, \ldots, \rho_k$ in $\nu(\Phi)$ such that the following holds for all transition systems $\mathcal{S}$ (where $\mathrm{GFP}(\mathcal{F}_{\mathcal{S}})$ *denotes the greatest fixpoint of $\mathcal{F}_{\mathcal{S}}$ and* $\mathrm{LFP}(\mathcal{F}_{\mathcal{S}})$ *the least fixpoint):*

$$(\|\tau_1\|_{\mathcal{S}}, \ldots, \|\tau_k\|_{\mathcal{S}}) = \mathrm{GFP}(\mathcal{F}_{\mathcal{S}})$$

and:

$$(\|\rho_1\|_{\mathcal{S}}, \ldots, \|\rho_k\|_{\mathcal{S}}) = \mathrm{LFP}(\mathcal{F}_{\mathcal{S}}).$$

Example. We illustrate, how we can construct these simultaneous fixpoints in the case $k = 2$, i.e. we have $\delta_1(X, Y)$ and $\delta_2(X, Y)$.

- $\tau_1 = \nu X.\delta_1(X, Y)[\nu Y.\delta_2(X, Y)/Y]$,
- $\tau_2 = \nu Y.\delta_2(X, Y)[\nu X.\delta_1(X, Y)/X]$,
- $\rho_1 = \mu X.\delta_1(X, Y)[\mu Y.\delta_2(X, Y)/Y]$,
- $\rho_2 = \mu Y.\delta_2(X, Y)[\mu X.\delta_1(X, Y)/X]$.

Before the main theorem let us prove a usefull lemma.

Lemma 7. *Let $\mathcal{A}$ be an alternating automaton. There is a μ-formula ρ in Σ_1^{μ} such that, for any pointed transition system $(\mathcal{S}, s_{\mathrm{I}})$, we have:*

$s_{\mathrm{I}} \in \|\rho\|_{\mathcal{S}}$ *iff there exists a finite run of $\mathcal{A}$ on s_{I}.*

Proof. Suppose the automaton has states $Q = \{q_1, \ldots, q_k\}$ and transition function δ. By definition of the transition conditions we see that all δ_i are in Π_0^μ (where $\delta(q_i)$ abbreviates δ_i). Now remember that all δ_i can be interpretated as formulae which can contain propositional variables among $q_1, \ldots, q_k$ (i.e. $\delta_i(q_1, \ldots, q_k)$). So by Lemma 6 there are formulae $\rho_i \in \Sigma_1^\mu$ for all $i \in \{1, \ldots, k\}$ such that for each transition system $\mathcal{S} = (S, R, \lambda)$ the functional:

$$\mathcal{F}_\mathcal{S} : (S_1, \ldots, S_k) \mapsto (\|\delta_1(S_1, \ldots, S_k)\|_\mathcal{S}, \ldots, \|\delta_k(S_1, \ldots, S_k)\|_\mathcal{S})$$

has the least fixpoint:

$$\mathrm{LFP}(\mathcal{F}_\mathcal{S}) = (\|\rho_1\|_\mathcal{S}, \ldots, \|\rho_k\|_\mathcal{S}).$$

Now, for all $i = 1, \ldots, k$, let $\mathcal{A}_i^{\mathcal{S}*}$ denote the set of $s \in S$, for which there is a finite q_i-run on s of $\mathcal{A}$.. It is enough to prove:

$$(\mathcal{A}_1^{\mathcal{S}*}, \ldots, \mathcal{A}_k^{\mathcal{S}*}) = (\|\rho_1\|_\mathcal{S}, \ldots, \|\rho_k\|_\mathcal{S}).$$

To prove this, by Tarski-Knaster (Theorem **??** in chapter 20), it is enough to show the following two things:

(i) $\mathcal{F}_\mathcal{S}(\mathcal{A}_1^{\mathcal{S}*}, \ldots, \mathcal{A}_k^{\mathcal{S}*}) \subseteq (\mathcal{A}_1^{\mathcal{S}*}, \ldots, \mathcal{A}_k^{\mathcal{S}*})$,

(ii) For all $(S_1, \ldots, S_k) \subseteq S^k$ such that $\mathcal{F}_\mathcal{S}(S_1, \ldots, S_k) \subseteq (S_1, \ldots, S_k)$ we have:

$$(\mathcal{A}_1^{\mathcal{S}*}, \ldots, \mathcal{A}_k^{\mathcal{S}*}) \subseteq (S_1, \ldots, S_k).$$

Let us first prove (i). Suppose that $(s_1, \ldots, s_k) \in \mathcal{F}_\mathcal{S}(\mathcal{A}_1^{\mathcal{S}*}, \ldots, \mathcal{A}_k^{\mathcal{S}*})$, so all s_i are in $\|\delta_i(\mathcal{A}_1^{\mathcal{S}*}, \ldots, \mathcal{A}_k^{\mathcal{S}*})\|$. By Lemma 1 there is the "beginning of a run" with root (s_i, q_i) and leaves (s'_j, q_j) where j ranges over $\{1, \ldots, k\}$ and $s'_j \in \mathcal{A}_j^{\mathcal{S}*}$. From this we get $s_i \in \mathcal{A}_i^{\mathcal{S}*}$, which proves (i).

To prove (ii), let $(S_1, \ldots, S_k)$ satisfy the premise of (ii) and suppose that $(s_1, \ldots, s_k) \in (\mathcal{A}_1^{\mathcal{S}*}, \ldots, \mathcal{A}_k^{\mathcal{S}*})$. By induction on the maximal depth d of the accepting runs for $(s_1, \ldots, s_k)$ we prove $(s_1, \ldots, s_k) \in (S_1, \ldots, S_k)$.

- $d = 1$: In this case we have $(s_1, \ldots, s_k) \in \mathcal{F}_\mathcal{S}(\emptyset, \ldots, \emptyset)$ and so, using the monotonicity of $\mathcal{F}$, we get the desired result.
- $d > 1$: Now suppose, for $i = 1, \ldots, k$, that $\ell(v_i) = (s_i, q_i)$ is the root of the accepting q_i-run for s_i. By induction hypothesis for all the sons v'_j of v_i, which all are of the form $\ell(v'_j) = (s'_j, q_{l(v'_j)})$ $(l(v'_j) \in \{1 \ldots, k\})$, we have $s'_j \in S_{l(v'_j)}$. So, with a very similar argument to the one used in (i), we see that, for all i, s_i is in the i-component of $\mathcal{F}_\mathcal{S}(S_1, \ldots, S_k)$. With the premise of (ii) we get the desired result.

So the proof is completed.

We are now able to prove the main result of this section.

Theorem 8. *For any alternating automaton $\mathcal{A} = (Q, \Sigma, \delta, q_{\mathrm{I}}, \Omega)$ one can construct a μ-formula $\tau_{\mathcal{A}}$ (over propositional variables $\Sigma \cup Q$), such that, for all pointed transition systems $(\mathcal{S}, s_{\mathrm{I}})$, we have:*

$$\mathcal{A} \text{ accepts } (\mathcal{S}, s_{\mathrm{I}}) \quad \Leftrightarrow \quad s_{\mathrm{I}} \in \|\tau_{\mathcal{A}}\|_{\mathcal{S}}.$$

Moreover, if $\mathcal{A}$ is Σ_n, then $\tau_{\mathcal{A}}$ can be chosen in Σ^{μ}_{n+1}; if $\mathcal{A}$ is Π_n, then $\tau_{\mathcal{A}}$ can be chosen in Π^{μ}_{n+1}.

Proof. The proof goes by induction on the index n of the automaton. We assume for all alternating automata $\mathcal{A}$ that the the priority function is defined only strongly connected components of the transition graph. Moreover, we assume that the cardinality of the range of the priority function of an automaton of index n is also n. This is no real restriction, since all automata are equivalent to one fulfilling these assumtions.

There will be two cases for the induction step ($n > 0$):

Case 1: If the maximal priority m is even, we consider k auxiliary automata of index $\leq n-1$, in which the states of $\Omega^{-1}[m]$ are moved into variables. Then we apply the greatest fixpoint operator.

Case 2: If the maximal priority m is odd, we consider the complement $\hat{\mathcal{A}}$ of our automaton $\mathcal{A}$. By Lemma 4, $\hat{\mathcal{A}}$ can be chosen to have the same index as $\mathcal{A}$, but with maximal priority even. Thus, if we assume that the induction step for Case 1 has been made, we have a Π^{μ}_{n+1}-formula $\tau_{\hat{\mathcal{A}}}$ representing the complement. By Lemma 3 we know that there is a formula $\tau_{\mathcal{A}} \in \Sigma^{\mu}_{n+1}$ which is equivalent to $\neg\tau_{\hat{\mathcal{A}}}$. It is easy to check that $\tau_{\mathcal{A}}$ is the Σ^{μ}_{n+1}-formula fulfilling the requirements of the theorem.
So, only the induction step for Case 1 has to be done.

The informal description above, shows that greatest fixpoints capture the automata with even maximal priority and the least fixpoints, as negations of greatest fixpoints, the automata with an odd maximal priority.

Before we do the induction, let us explain what means "moving states into variables". We need to define two transformations for automata.

The first takes an automaton $\mathcal{A} = (Q, \Sigma, \delta, q_{\mathrm{I}}, \Omega)$ and a set $X \subsetneq Q$, such that $q_{\mathrm{I}} \notin X$ and defines a new automaton

$$\mathcal{A}_{free(X)} = (Q - X, \Sigma \cup X, \delta', q_{\mathrm{I}}, \Omega')$$

where δ' and Ω' are the restrictions of δ (resp. Ω) to $Q - X$.
This is the transformation which converts states of the automaton into variables. Notice that the runs of $\mathcal{A}_{free(X)}$ are like the "beginning" of a run of the automaton $\mathcal{A}$. If we reach a point (s, q), where $q \in X$ the run of $\mathcal{A}_{free(X)}$ stops, whereas if it was a run of the automaton $\mathcal{A}$ it would go on.

The second transformation on automata helps us to deal with the restriction $q_{\mathrm{I}} \notin X$ we had on the first transformation. It takes an automaton as above, a state $q \in Q$ and a new symbol $\hat{q} \notin (Q \cup \Sigma)$ and defines a new automaton

$$\mathcal{A}_{start(q)} = (Q \cup \{\hat{q}\}, \Sigma, \delta'', \hat{q}, \Omega)$$

Where δ'' is equal to δ on Q and $\delta''(\hat{q}) = \delta(q)$.

It is clear, that $\mathcal{A}_{start(q)}$ accepts the same pointed transition systems as $\mathcal{A}$ with initial state q. Moreover, note that for all $X \subseteq Q$, the introduction of $\hat{q}$ makes possible for all automata to do the operation $(\mathcal{A}_{start(q)})_{free(X)}$ (shorter $\mathcal{A}_{start(q)free(X)}$).

Let us do the induction on the index n of an automaton $\mathcal{A} = (Q, \Sigma, \delta, q_{\mathrm{I}}, \Omega)$.

$\underline{n = 0:}$ In this case every run of an automaton must be finite, and so the theorem follows from Lemma 7.

$\underline{n > 0:}$ As shown before it is enough to do the induction step only for Case 1. We define U to be the set of states $\Omega^{-1}[m]$, where m is the maximal priority, assuming that $q_{\mathrm{I}} \notin U$; otherwise we consider the semantically equivalent automaton $\mathcal{A}_{start(q_{\mathrm{I}})}$. Suppose $U = \{q_1, \ldots, q_k\}$. We consider the automata $\mathcal{A}_{free(U)}$ and $\mathcal{A}_{start(q_i)free(U)}$ for all $i = 1, \ldots, k$. It is easy to see, that all these automata are of index $\leq n - 1$. So by induction hypothesis, there are μ-formulae $\tau_0(\boldsymbol{q})$ and $\tau_1(\boldsymbol{q}), \ldots, \tau_k(\boldsymbol{q})$ in Σ_n^μ (where $\boldsymbol{q} \equiv (q_1, \ldots, q_k)$), such that for all pointed transition systems $(\mathcal{S}, s_{\mathrm{I}})$ we have:

$$\mathcal{A}_{free(U)} \text{ accepts } (\mathcal{S}, s_{\mathrm{I}}) \quad \Leftrightarrow \quad s_{\mathrm{I}} \in \|\tau_0(\boldsymbol{q})\|_{\mathcal{S}}$$

and

$$\mathcal{A}_{start(q_i)free(U)} \text{ accepts } (\mathcal{S}, s_{\mathrm{I}}) \quad \Leftrightarrow \quad s_{\mathrm{I}} \in \|\tau_i(\boldsymbol{q})\|_{\mathcal{S}},$$

for all $i = 1, \ldots, k$.

Now consider the functionals $\mathcal{F}_{\mathcal{S}} : \mathcal{P}(S^k) \to \mathcal{P}(S^k)$ with:

$$\mathcal{F}_{\mathcal{S}} : (S_1, \ldots, S_k) \mapsto (\|\tau_1(S_1, \ldots, S_k)\|_{\mathcal{S}}, \ldots, \|\tau_k(S_1, \ldots, S_k)\|_{\mathcal{S}}).$$

By Lemma 6 there are μ-formulae $\rho_1, \ldots, \rho_k$ in Π_{n+1}^μ such that for all transition systems $\mathcal{S}$, $(\|\rho_1\|_{\mathcal{S}}, \ldots, \|\rho_k\|_{\mathcal{S}})$ is the greatest fixpoint of $\mathcal{F}_{\mathcal{S}}$.

In order to do the induction step let us make the following claim.

Claim. For all pointed transition systems $(\mathcal{S}, s_{\mathrm{I}})$ and for all $i = 1 \ldots, k$ we have the two following facts:

(1)
$$\mathcal{A}_{start(q_i)} \text{ accepts } (\mathcal{S}, s_{\mathrm{I}}) \quad \Leftrightarrow \quad s_{\mathrm{I}} \in \|\rho_i\|_{\mathcal{S}}.$$

(2)
$$\mathcal{A} \text{ accepts } (\mathcal{S}, s_{\mathrm{I}}) \quad \Leftrightarrow \quad s_{\mathrm{I}} \in \|\tau_0[\rho_1/q_1, \ldots, \rho_k/q_k]\|_{\mathcal{S}}.$$

Since $\tau_0[\rho_1/q_1, \ldots, \rho_k/q_k] \in \Pi_{n+1}^\mu$ the claim completes the induction step for Case 1.

We will prove the claim by first showing that (1) implies (2), and then by showing the correctness if (1).

– (1) implies (2): Let us remark, that by choice of τ_0 and by (1) the following equivalence holds:

$$s_I \in \|\tau_0[\rho_1/q_1, \ldots, \rho_k/q_k]\|_{\mathcal{S}} \quad \Leftrightarrow \quad \mathcal{A}_{free(U)} \text{ accepts } (\mathcal{S}', s_{\mathrm{I}})$$

where $\mathcal{S}' \equiv \mathcal{S}[q_1 \mapsto \mathcal{A}^{\mathcal{S}}_{start(q_1)}, \ldots, q_k \mapsto \mathcal{A}^{\mathcal{S}}_{start(q_k)}]$ and $\mathcal{A}^{\mathcal{S}}_{start(q_i)}$ is the set of states s in S such that $\mathcal{A}_{start(q_i)}$ accepts $(\mathcal{S}, s)$.
So it is enough to show:

$$\mathcal{A} \text{ accepts } (\mathcal{S}, s_{\mathrm{I}}) \quad \Leftrightarrow \quad \mathcal{A}_{free(U)} \text{ accepts } (\mathcal{S}', s_{\mathrm{I}})$$

To prove the "only if" direction let us assume that ϱ is a q_{I}-run on s_{I} of the automaton $\mathcal{A}$ on $\mathcal{S}$. We want to convert it into a q_{I}-run on s_{I} of the automaton $\mathcal{A}_{free(U)}$ on $\mathcal{S}'$. Let us do the conversion for every branch of ϱ. If we have a branch where there is no state of U, then we do not change anything, otherwise, by the first $q_i \in U$ appearing in the branch, we cut it up. The new end point we get is of the form (s, q_i), where by assumption $(\mathcal{S}, s)$ is accepted by $\mathcal{A}$ with new initial state q_i. Using the fact that this automaton is equivalent to $\mathcal{A}_{start(q_i)}$ and that q_i is now a variable, which by definition is true in $s \in S$ (under the valuation λ'), we get the desired result. The proof of the "if" direction follows similar arguments, to the one brought before.
– (1): As before $\mathcal{A}^{\mathcal{S}}$ is the set of all points s in $\mathcal{S}$ such that $(\mathcal{S}, s)$ is accepted by $\mathcal{A}$. By definition of ρ_i we have to prove that $(\mathcal{A}^{\mathcal{S}}_{start(q_1)}, \ldots, \mathcal{A}^{\mathcal{S}}_{start(q_k)})$ is the greatest fixpoint of $\mathcal{F}_{\mathcal{S}}$, and so by Tarski-Knaster:

(i) $(\mathcal{A}^{\mathcal{S}}_{start(q_1)}, \ldots, \mathcal{A}^{\mathcal{S}}_{start(q_k)}) \subseteq \mathcal{F}_{\mathcal{S}}(\mathcal{A}^{\mathcal{S}}_{start(q_1)}, \ldots, \mathcal{A}^{\mathcal{S}}_{start(q_k)})$
(ii) For all $(S_1, \ldots, S_k) \subseteq S^k$ such that $(S_1, \ldots, S_k) \subseteq \mathcal{F}_{\mathcal{S}}(S_1, \ldots, S_k)$ we have:

$$(S_1, \ldots, S_k) \subseteq (\mathcal{A}^{\mathcal{S}}_{start(q_1)}, \ldots, \mathcal{A}^{\mathcal{S}}_{start(q_k)}).$$

We first prove (i). Let remind ourselves that the i-th component of the tuple $\mathcal{F}_{\mathcal{S}}(\mathcal{A}^{\mathcal{S}}_{start(q_1)}, \ldots, \mathcal{A}^{\mathcal{S}}_{start(q_k)})$ is of the form:

$$\|\tau_i(\mathcal{A}^{\mathcal{S}}_{start(q_1)}/q_1, \ldots, \mathcal{A}^{\mathcal{S}}_{start(q_k)}/q_k)\|_{\mathcal{S}}.$$

Since τ_i was the formula equivalent to the automaton $\mathcal{A}_{start(q_i)free(U)}$ it is enough to show the following, for all states s in S:

$$\mathcal{A}_{start(q_i)} \text{ accepts } (\mathcal{S}, s) \quad \Rightarrow \quad \mathcal{A}_{start(q_i)free(U)} \text{ accepts } (\mathcal{S}', s)$$

where $\mathcal{S}'$ is $\mathcal{S}[q_1 \mapsto \mathcal{A}^{\mathcal{S}}_{start(q_1)}, \ldots q_k \mapsto \mathcal{A}^{\mathcal{S}}_{start(q_k)}]$.
Since a very similar result has been proven above, we omit the details.

To prove (ii) let $(S_1, \ldots, S_k)$ satisfy the premise of (ii), and let $s_i \in S_i$. Since $s_i \in \|\tau_i(S_1, \ldots, S_k)\|_{\mathcal{S}}$, by hypothesis about τ_i we have $s_i \in \mathcal{A}^{\mathcal{S}'}_{start(q_i)}$, where $\mathcal{S}'$ is $\mathcal{S}[q_1 \mapsto S_1, \ldots, q_k \mapsto S_k]$ (remember that τ_i is of the form $\tau_i(q_1, \ldots, q_k)$). So there is an accepting run of $\mathcal{A}_{start(q_i)free(U)}$ with the property that if it has a vertex (q_j, s_j) such that $q_j \in U$, then it is a leaf and we have $s_j \in S_j$. Hence we can reapply the premise of (ii) and construct a s_j-run of $\mathcal{A}_{start(q_j)free(U)}$, such that for all leaves of the form $(q_{j'}, s_{j'})$ with $q_{j'} \in U$ the premise of (ii) can be "re-"reapplied . By the iteration of this process, in the limes case, we get an accepting run of $\mathcal{A}_{start(q_i)}$ for s_i, since the following holds for all branches. If the branch is finite, then its end point is of the form (σ, s_i), where $\sigma \notin \{q_1, \dots, q_k\}$. By assumption we have $s_i \in \lambda(\sigma) = \lambda'(\sigma)$ (where λ' is the valuation of $\mathcal{S}'$). For the infinite branches we have two cases. For the first case the infinite branch contains only finitely many states q, which are in U, thus it easily follows, that from the last appearance of a $q \in U$ on, this branch is the same as a branch of an accepting run of an automaton $\mathcal{A}_{start(q_l)free(U)}$. So the highest priority appearing infinitely often must be even, and the branch is accepted. For the other case, there are infinitely many states of U in the branch, and since U is the set where the priority function has its maximal value m and m is even, we have an accepted branch.

So the proof is completed.

We end this section by giving an example of an automaton and an equivalent μ-formula obtained with the construction described in the proof.

Example. Given an automaton $\mathcal{A} = (\{q_0, q_1, q_2\}, \{p_1, p_2\}, \delta, q_0, \Omega)$ such that $\delta(q_0) = \Box q_1$, $\delta(q_1) = q_2 \vee \Diamond q_0$ and $\delta(q_2) = p_1 \wedge \Box q_1$, and such that $\Omega(q_0) \uparrow$ and $\Omega(q_1) = \Omega(q_2) = 0$. We construct an equivalent μ-formula, following the proof of Theorem 8 (we use trivial equivalences of μ-formulae to get more compact representations).

We set $U = \{q_1, q_2\}$. By construction the formula φ equivalent to the automaton has the structure $\tau_0[\rho_1/q_1, \rho_2/q_2]$, where the formulae τ_0, ρ_1, ρ_2 are defined as follows:

- τ_0 is equivalent to $\mathcal{A}_{free(U)}$,
- ρ_1, ρ_2 are formulae such that for all $\mathcal{S}$ we have $(\|\rho_1\|, \|\rho_2\|) = \mathrm{GFP}(\mathcal{F}_1)$, where $\mathcal{F}_1$ is the functional:

$$\mathcal{F}_1 : (S_1, S_2) \mapsto \|\tau_1(S_1, S_2), \tau_2(S_1, S_2)\|_{\mathcal{S}},$$

  where $\tau_1(q_1, q_2)$ is the formula equivalent to $\mathcal{A}_{start(q_1)free(U)}$ and $\tau_2(q_1, q_2)$ is the formula equivalent to $\mathcal{A}_{start(q_2)free(U)}$.

By construction we also have for all transition systems $\mathcal{S}$:

- $\mathcal{A}_{free(U)}$ is equivalent to $\mathrm{LFP}(\mathcal{F}_2)$ with:

$$\mathcal{F}_2 : S \mapsto \|\Box q_1\|_{\mathcal{S}}.$$

- $\mathcal{A}_{start(q_1)free(U)}$ is equivalent to the second component of $\mathrm{LFP}(\mathcal{F}_3)$ with:

$$\mathcal{F}_3 : (S_0, S_1) \mapsto \|(\Box q_1, q_2 \vee \Diamond S_0)\|_{\mathcal{S}}.$$

- $\mathcal{A}_{start(q_2)free(U)}$ is equivalent to the second component of LFP($\mathcal{F}_4$) with:

$$\mathcal{F}_4 : (S_0, S_2) \mapsto \|(\Box q_1, p_1 \wedge \Box q_1)\|_{\mathcal{S}}.$$

Putting all this together we obtain (Example 4 may be useful for a better understanding):

- $\tau_0 = \Box q_1$,
- $\tau_1 = q_2 \vee \Diamond\Box q_1$,
- $\tau_2 = p_1 \wedge \Box q_1$.

So we get:

$$\mathcal{F}_1 : (S_1, S_2) \mapsto \|S_2 \vee \Diamond\Box S_1, p_1 \wedge \Box S_1\|_{\mathcal{S}}$$

which gives us:

$$(\rho_1, \rho_2) = (\nu X.((p_1 \wedge \Box X) \vee \Diamond\Box X), \nu Y.(p_1 \wedge \Box \nu X.(p_1 \wedge \Box X)))$$

and so we have:

$$\varphi = \Box(\nu X.((p_1 \wedge \Box X) \vee \Diamond\Box X)).$$

5 Strictness of the hierarchy

In this section we will give the proof that the hierarchy of modal μ-calculus on transition systems is strict. We proceed similarly to Arnold in [1]. We will first show the strictness of a hierarchy induced by automata and then use this to get the main result.

We assume that for all automata $\mathcal{A} \in \Sigma_n$ the range of the priority function is Ω_{Σ_n} and that for all automata $\mathcal{A} \in \Pi_n$ the range is Ω_{Π_n}, where Σ_n and Π_n are defined as follows:

- $n = 0$: If $\mathcal{A} \in \Sigma_n$ or $\mathcal{A} \in \Pi_n$ then $\Omega_{\Sigma_0} = \Omega_{\Pi_0} = \emptyset$.
- n even ($n \neq 0$):
  - If $\mathcal{A} \in \Sigma_n$ then $\Omega_{\Sigma_n} = \{0, \ldots, n-1\}$.
  - If $\mathcal{A} \in \Pi_n$ then $\Omega_{\Pi_n} = \{1, \ldots, n\}$.
- n odd:
  - If $\mathcal{A} \in \Sigma_n$ then $\Omega_{\Sigma_n} = \{1, \ldots, n\}$.
  - If $\mathcal{A} \in \Pi_n$ then $\Omega_{\Pi_n} = \{0, \ldots, n-1\}$.

The assumptions we have made above are no real restriction, since all automata are equivalent to one fulfilling these assumptions (see Exercise 5).

For every natural number n we now introduce the **$\boldsymbol{\Sigma_n}$-Test Automaton** T_{Σ_n} and the **$\boldsymbol{\Pi_n}$-Test Automaton** T_{Π_n}.

All T_{Σ_n} are of the form (where u is a new symbol):

$$T_{\Sigma_n} = (Q_{\Sigma_n}, P_{\Sigma_n}, q_u, \delta_{\Sigma_n}, \Omega)$$

and all T_{Π_n} are of the form:

$$T_{\Pi_n} = (Q_{\Pi_n}, P_{\Pi_n}, q_u, \delta_{\Pi_n}, \Omega)$$

where:

- $Q_{\Sigma_n} = \Omega_{\Sigma_n} \cup \{q_u\}$,
- $Q_{\Pi_n} = \Omega_{\Pi_n} \cup \{q_u\}$,
- $P_{\Sigma_n} = \{c_u\} \cup \{c_i \mid i \in \Omega_{\Sigma_n}\} \cup \{d_u\} \cup \{d_i \mid i \in \Omega_{\Sigma_n}\}$,
- $P_{\Pi_n} = \{c_u\} \cup \{c_i \mid i \in \Omega_{\Pi_n}\} \cup \{d_u\} \cup \{d_i \mid i \in \Omega_{\Pi_n}\}$,
- for all states $q_j \in Q_{\Sigma_n}$ we have:

$$\delta(q_j) = \bigvee_{i \in Q_{\Sigma_n}} (c_i \wedge \Box q_i) \vee \bigvee_{i \in Q_{\Sigma_n}} (d_i \wedge \Diamond q_i),$$

- for all states $q_j \in Q_{\Pi_n}$ we have:

$$\delta(q_j) = \bigvee_{i \in Q_{\Pi_n}} (c_i \wedge \Box q_i) \vee \bigvee_{i \in Q_{\Pi_n}} (d_i \wedge \Diamond q_i),$$

- $\Omega(q_j) = j$ if $j \neq u$ and $\Omega(q_u) \uparrow$.

We will reduce the problem of acceptance of a binary transition system $\mathcal{S}$ (i.e. a transition system whose underlying structure is a binary tree) by a given automaton $\mathcal{A} \in \Sigma_n$ (resp. Π_n) to the acceptance of another transition system $G_{\mathcal{A},q}(\mathcal{S})$ by T_{Σ_n} (resp. T_{Π_n}). Before we describe the reduction let us explain some notation for binary transition systems. A binary transition system (S, R, λ) is of the form $a(t_1, t_2)$, if S is a binary tree, where the sons of the root generate the subtrees t_1 and t_2 (if there is no i-son, we set $t_i = \epsilon$) and if the valuation λ of the root is a.

Let $\mathcal{A} = (Q, P, q_{\mathrm{I}}, \delta, \Omega) \in \Sigma_n$ (resp. Π_n) be an automaton and q a state of $\mathcal{A}$. For all binary transition systems $a(t_1, t_2)$ we describe $G_{\mathcal{A},q}(a(t_1, t_2))$ as the result of a procedure $G_{\mathcal{A},q}$ (with parameter q) starting from the root of $a(t_1, t_2)$, as follows:

We can see $\delta(q)$ as a binary tree whose internal nodes are labelled by $\vee$ or $\wedge$ and the leaves are of one of the following forms: $q', \Diamond q', \Box q', p (p \in P), \mathbf{0}, \mathbf{1}$.

$G_{\mathcal{A},q}(a(t_1, t_2))$ is obtained by substituting in $\delta(q)$

- c_i for $\wedge$, d_i for $\vee$, where $i = \Omega(s)$,
- c_u for $\wedge$, d_u for $\vee$, where $\Omega(s) \uparrow$,
- $G_{\mathcal{A},q'}(a(t_1, t_2))$ for q',
- $d_i(G_{\mathcal{A},q'}(t_1), G_{\mathcal{A},q'}(t_2))$ for $\Diamond q'$, where $i = \Omega(q')$ and $t_1 \neq \epsilon$ or $t_2 \neq \epsilon$,
- $c_i(G_{\mathcal{A},q'}(t_1), G_{\mathcal{A},q'}(t_2))$ for $\Box q'$, where $i = \Omega(q')$ and $t_1 \neq \epsilon$ or $t_2 \neq \epsilon$,
- $d_u(G_{\mathcal{A},q'}(t_1), G_{\mathcal{A},q'}(t_2))$ for $\Diamond q'$, where $\Omega(q') \uparrow$ and $t_1 \neq \epsilon$ or $t_2 \neq \epsilon$,
- $c_u(G_{\mathcal{A},q'}(t_1), G_{\mathcal{A},q'}(t_2))$ for $\Box q'$, where $\Omega(q') \uparrow$ and $t_1 \neq \epsilon$ or $t_2 \neq \epsilon$,
- Q_{Σ_n} (resp.Q_{Π_n}) for $\mathbf{1}$ and for p, if $p \in a$,
- $\emptyset$ for $\mathbf{0}$ and for p, if $p \notin a$.
- Q_{Σ_n} (resp.Q_{Π_n}) for $\Box q$, if $t_1 = t_2 = \epsilon$,
- $\emptyset$ for $\Diamond q$, if $t_1 = t_2 = \epsilon$.

It is easy to see that $G_{\mathcal{A},q}(a(t_1, t_2))$ must be a binary transition system.

The proof of the following lemma follows directly from the fact, that $G_{\mathcal{A},q}(\mathcal{S})$ is a representation of the game tree described in chapter 9 (which there is called behaviour), where the choice nodes for player 0 are the d-nodes and the choice nodes for player 1 are the c-nodes. Moreover, it follows easily from the definition that the Test Automaton accepts the game tree if and only if player 0 has a winning stategy. As a consequence we get the following lemma:

Lemma 9. *For any binary transition system $\mathcal{S}$ and any automaton $\mathcal{A}$ with initial state q, we have:*

- *If $\mathcal{A} \in \Sigma_n$:*
$$\mathcal{S} \in \|\mathcal{A}\| \quad \Leftrightarrow \quad G_{\mathcal{A},q}(\mathcal{S}) \in \|T_{\Sigma_n}\|$$
- *If $\mathcal{A} \in \Pi_n$:*
$$\mathcal{S} \in \|\mathcal{A}\| \quad \Leftrightarrow \quad G_{\mathcal{A},q}(\mathcal{S}) \in \|T_{\Pi_n}\|$$

The following example will illustrate, how this transformation on binary transition systems works.

Example. Figure 1 shows $\mathcal{S}$, $G_{\mathcal{A},q_1}(\mathcal{S})$ and $G_{\mathcal{A},q_1}(G_{\mathcal{A},q_1}(\mathcal{S})) =: G^2_{\mathcal{A},q_1}(\mathcal{S})$, where $\mathcal{S}$ is a simple binary transition system over a set of propositional variables $\{p_1, p_2\}$ and $\mathcal{A} = (\{q_1, q_2, q_3\}, \{p_1, p_2\}, \delta, q_1, \Omega)$ is an alternating Π_2 automaton with:

- $\delta(q_1) = \Box q_2$,
- $\delta(q_2) = \Diamond q_1 \vee q_3$,
- $\delta(q_3) = p_1$,
- $\Omega(q_1) = 1$ and
- $\Omega(q_2) = \Omega(q_3) = 2$.

The next lemma shows that $G_{\mathcal{A},q}$ has a fixpoint $F_{\mathcal{A},q}$ (or simply $F_{\mathcal{A}}$, if q is the initial state of $\mathcal{A}$).

Lemma 10. *For all automata $\mathcal{A}$ and states q there is a transition system $F_{\mathcal{A},q}$ such that:*
$$G_{\mathcal{A},q}(F_{\mathcal{A},q}) \equiv F_{\mathcal{A},q}$$

Proof. Let us begin with the following claim.

Claim. Given two binary transition systems $\mathcal{S}$ and $\mathcal{S}'$, which are identical upto depth m, then $G_{\mathcal{A},q}(\mathcal{S})$ and $G_{\mathcal{A},q}(\mathcal{S}')$ are identical upto depth $m+1$.

Assuming the claim as proven, it is easy to see that the limit $\lim\{G^n_{\mathcal{A},q}(\mathcal{S})\}_{n\in\omega}$ of the sequence $\{G^n_{\mathcal{A},q}(\mathcal{S})\}_{n\in\omega}$ exists and that
$$F_{\mathcal{A},q} \equiv \lim\{G^n_{\mathcal{A},q}(\mathcal{S})\}_{n\in\omega}$$
is the desired fixpoint. So only the proof for the claim stated above has to be done.

To prove the claim, we remark that the procedure $G_{\mathcal{A},q}$ works down the two transition systems beginning from the root, and since both systems are equal until depth m, the same subprocedures are executed until then. Suppose we

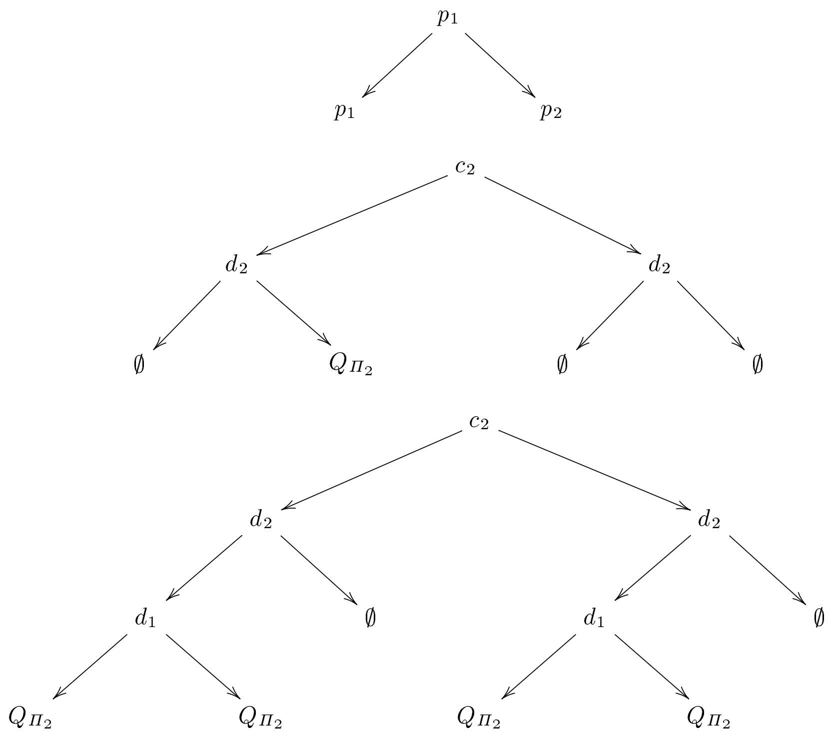

Fig. 1. $\mathcal{S}$, $G_{\mathcal{A},q_1}(\mathcal{S})$ and $G^2_{\mathcal{A},q_1}(\mathcal{S})$

have a vertex at level m (which exists in both systems at the same position), and suppose that this vertex generates the subtrees $\mathcal{S}_1$ (resp. $\mathcal{S}'_2$) of $\mathcal{S}$ (resp. $\mathcal{S}'$), Now by the observations made above $G_{\mathcal{A},q}(\mathcal{S})$ will have the subtree $G_{\mathcal{A},q'}(\mathcal{S}_1)$ and $G_{\mathcal{A},q}(\mathcal{S}')$ will have the subtree $G_{\mathcal{A},q'}(\mathcal{S}'_1)$ (if the procedure reaches this vertex, otherwise it is trivial). Since $\mathcal{S}_1$ and $\mathcal{S}'_1$ have the same root, it follows from the definition of the procedure that $G_{\mathcal{A},q'}(\mathcal{S}_1)$ and $G_{\mathcal{A},q'}(\mathcal{S}'_1)$ are equal on the root and on the sons of the root (if a son exists). This proves the claim and thus the lemma.

We now prove the hierarchy theorem:

Theorem 11. *For all n we have:*

$$\Sigma^{\mathbf{TR}}_{n+1} \neq \Sigma^{\mathbf{TR}}_{n}$$

Proof. We prove the result by contradiction. Suppose $\Sigma^{\mathbf{TR}}_{n+1} = \Sigma^{\mathbf{TR}}_{n}$, by definition it follows that $\Sigma^{\mathbf{TR}}_{n} = \Pi^{\mathbf{TR}}_{n}$. With Lemma 4 we get $\mathbf{TR} - \|T_{\Sigma_n}\| \in \Sigma_n$.

And so there exists an automaton $\mathcal{A}$ such that: $\mathbf{TR} - \|T_{\Sigma_n}\| = \|\mathcal{A}\|$. By Lemma 9 and Lemma 10 there is a transition system $F_{\mathcal{A}}$ such that:

$$F_{\mathcal{A}} \in \|T_{\Sigma_n}\| \quad \Leftrightarrow \quad F_{\mathcal{A}} \in \|\mathcal{A}\| \quad \Leftrightarrow \quad F_{\mathcal{A}} \in \mathbf{TR} - \|T_{\Sigma_n}\|$$

and hence a contradiction, which proves the theorem.

We apply the result to the modal μ-calculus, by using the fact that the index of an automaton corresponds to the alternation depth of a μ-formula, and we get the following theorem.

Theorem 12. *The alternation hierarchy of modal μ-calculus is strict.*

Proof. Suppose $\Sigma_n^{\mu\mathbf{TR}} = \Sigma_{n+1}^{\mu\mathbf{TR}}$, by the Theorems 8 in this chapter and **??** in chapter 10 we have $\Sigma_{n+1}^{\mathbf{TR}} = \Sigma_{n+2}^{\mathbf{TR}}$ which is not the case by Theorem 11.

Literature

1. A. Arnold, *The mu-calculus alternation-depth hierarchy is strict on binary trees*, Theoretical Informatics and Applications, 33 (1999), pp. 329–339.
2. J. C. Bradfield, *The modal mu-calculus alternation hierarchy is strict*, in Proceedings of the 7th International Conference on Concurrency Theory, CONCUR '96, U. Montanari and V. Sassone, eds., vol. 1119 of Lecture Notes in Computer Science, Springer-Verlag, August 1996, pp. 232–246.
3. G. Lenzi, *A hierarchy theorem for the mu-calculus*, in Proceedings of the 23rd International Colloquium on Automata, Languages and Programming, ICALP '96, F. Meyer auf der Heide and B. Monien, eds., vol. 1099 of Lecture Notes in Computer Science, Springer-Verlag, July 1996, pp. 87–97.
4. D. Niwiński, *Fixed point characterization of infinite behavior of finite-state systems*, Theoretical Computer Science, 189 (1997), pp. 1–69.
5. T. Wilke, *Alternating tree automata, parity games, and modal mu-calculus*. Manuskript, 2000.