On modal mu-calculus and Büchi tree automata

Roope Kaivola
University of Edinburgh
Laboratory for Foundations of Computer Science

Abstract

Modal mu-calculus and automata on infinite trees are complementary ways of describing infinite tree languages. We examine a fragment of modal mu-calculus, the alternation-depth class Π_2 and its relation to tree languages recognised by Büchi-automata. It is shown that there is an exact correspondence between languages characterised by formulae in Π_2 and languages recognised by Büchi-automata which do not distinguish subtrees. This involves an inductive mapping from formulae to automata, in which the core point is a construction of powerset automata recognising maximal and minimal fixpoints of a Büchi-automaton. The construction avoids altogether the usual complex process of complementing automata when mapping formulae to automata, leading to a more transparent translation. As a corollary of the result we see that the alternation-depth hierarchy of modal mu-calculus does not collapse at Π_2.

1 Introduction

Modal mu-calculus is a language obtained by extending standard modal logic with minimal and maximal fixpoint operators. It has attracted considerable interest as a method for specifying properties of concurrent programs [EC80] [Koz83] [Sti92]. On the one hand this formalism is syntactically simple enough to make it appealing, on the other hand it is expressive enough so that most of the known temporal and dynamic logics can be translated into it.

Automata on infinite objects are of interest in this context for two reasons. First, they can be used to specify properties of concurrent systems directly. Secondly, for many temporal and modal logics of programs, including modal mu-calculus, the standard decision procedures involve translating a formula of the logic into an automaton recognising the models of the formula.

It is known that the full modal mu-calculus is equiexpressive with the Rabin-automata on infinite trees [Niw88] [EJ91]. In this paper we examine a fragment of the modal mu-calculus, in which certain restrictions apply to the allowed nesting of alternating fixpoints, and the subclass of Rabin-automata called *special* or *Büchi-automata*. More exactly, the fragment of modal mu-calculus examined is the alternation depth class Π_2.

It is established that there is a precise correspondence between the expressive power of these two formalisms. The result is shown by mapping every Π_2 formula to a Büchi-automaton recognising exactly the models of the formula, and vice versa. However, in the mapping from Büchi-automata to formulae the mapping is faithful only to the extent that the automata do not take advantage of their ability to distinguish different subtrees of a node, something no modal mu-calculus formula is able to do. In a sense the result can be seen as relating to each other the concepts of Büchi-automata as an 'easy' or more tractable subclass of the more general Rabin-automata, and the alternation depth class Π_2 as an 'easy' fragment of modal mu-calculus.

This result in itself is not entirely unexpected, as Arnold and Niwinski [AN90], and independently Takahashi [Tak86] have previously shown the equivalence of regular tree expressions in alternation depth class Π_2 and Büchi-automata for binary trees.

However, the proof provided here, especially the mapping from Π_2 to Büchi-automata, is of some interest in its own right. This mapping is defined inductively on the structure of the formula. Intuitively, for each basic formula we describe the corresponding automaton directly, and for each boolean, modal and fixpoint operator, we describe a corresponding construction mapping a Büchi-automaton to another Büchi-automaton. The novelty here is the fixpoint operators, which are taken care of using a powerset construction based on ideas of [Dam92] for Büchi-automata on strings. What this inductive construction avoids altogether is the need to complement automata on infinite objects, a notoriously complex process that is usually required in mapping formulae to automata. Also when contrasted with the techniques of [EJ91] used for mapping the full modal mu-calculus to Rabin-automata without the need for complementation, the construction used here is much less complex, thanks to the relatively simple structure of Büchi-automata.

As a somewhat surprising observation, it turns out that the mapping from Büchi-automata to mu-calculus formulae in Π_2 is very unlikely to be possible, if we use the definition of alternation depth most commonly used in modal mu-calculus literature [EL86] [Bra91]. This normally adopted definition is based on removing closed subformulae from a formula and then counting syntactically the nesting of alternating fixpoints. In this paper, as e.g. in [Niw86], an alternation depth class is defined simply as the minimal set that contains all the classes lower in the hierarchy and is closed under substitution and prefixing with the relevant fixpoint. Our results point out a subtle difference between these two approaches. The approach to alternation depth taken here turns out to lead to strictly larger alternation depth classes than the one of [EL86]. However, it seems that most of the results concerning modal mu-calculus and alternation depth that were obtained using the more restrictive definition, e.g. the time bound for model-checking modal mu-calculus formulae of [EL86], will still apply using the more liberal definition. Consequently, we would advocate the application of the more liberal definition of alternation depth classes in modal mu-calculus in general.

2 Notation and definitions

Let us first list some standard notation that will be used later.

Definition 2.1 Let Σ be an arbitrary set. We use Σ^*, Σ^+ and Σ^ω to denote the sets of finite, finite nonempty and infinite strings of elements of Σ, respectively, and define $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$. If $s = a_1 \ldots a_n \in \Sigma^*$, $|s| = n$ is the length of s. ϵ denotes the unique empty string. If $s = a_1 \ldots a_n \ldots \in \Sigma^\infty$, $s[i \ldots j]$ is the string $a_i \ldots a_j$, s^i the string $a_i a_{i+1} \ldots$, and s_i the element a_i. If t and s are strings, both $t \cdot s$ and ts denote the concatenation of t and s, and $s \preceq t$ ($s \prec t$) means that s is a prefix (a proper prefix) of t. If $s \preceq t$, $t - s$ denotes the unique string s' such that $s \cdot s' = t$. For all $k \in N$, let us define $[k] = \{i \in N \mid 1 \leq i \leq k\}$. If A is a set, $P(A)$ denotes the powerset of A. If A and A' are sets, $A \uplus A'$ denotes the disjoint union of them, and $A \rightarrow A'$ and $A \rightharpoonup A'$ the sets of functions and partial functions from A to A', respectively. We use notation $\bar{q}$ to denote vectors $\bar{q} = (q_1, \ldots, q_n)$, and $\bar{q}_i = q_i$ to denote the individual elements of a vector. $\square$

In order to relate modal mu-calculus and tree automata to each other, we take a common set of structures acting both as models of modal mu-calculus formulae and possible inputs of tree automata. Usually modal mu-calculus is interpreted over arbitrary transition systems with transitions carrying labels from some set $\mathcal{L}$, and states carrying labels expressing the truth of propositional variables from some set $\mathcal{Z}$ [Koz83] [EL86] [Sti92]. The operation of tree-automata is normally defined for infinite trees with a fixed degree k of branching, and with nodes labelled by symbols from some alphabet. As far as the branching structure is concerned, any such tree can be seen as a transition system and a mu-calculus formula meaningfully interpreted over it. Therefore, let us fix some arbitrary degree of branching $k > 0$ and take the set of k-branching infinite trees as the set of possible models.

This still leaves us the difference between modal mu-calculus models having transition labels, and input trees for Büchi-automata lacking them. As any node in a tree is the target of at most one transition, this difference can be overcome by encoding the label of a transition in the node label of the target, in essence treating transition labels as particular kinds of propositions.

However, removing transition labels from models and treating them on a par with other propositions makes it natural to dispose of transition labels altogether and replace the relativised modalities $<l>$ and $[\,l\,]$ usually present in the modal mu-calculus, by their unrelativised counterparts $\Diamond$ and $\Box$. In the following we shall use the resulting language as the point of comparison. The relativised modalities can easily be expressed by the unrelativised ones by the correspondences $<l>\phi \Leftrightarrow \Diamond(l \wedge \phi)$ and $[\,l\,]\phi \Leftrightarrow \Box(\phi \vee \bigvee_{l' \in \mathcal{L} \setminus \{l\}} l')$ where the l:s on the left side refer to transition labels, as usual, and on the right side to similarly named propositions encoding transition labels.

Definition 2.2 Let Σ be an arbitrary set. An *infinite k-branching tree labelled with* Σ is a mapping $\Pi : [k]^* \rightarrow \Sigma$. Let Π be such a tree. If $i \in [k]^*$, then Π^i is the subtree of Π rooted at node i, i.e. the mapping defined by $\Pi^i(j) = \Pi(i \cdot j)$, for all $j \in [k]^*$. If $p \in [k]^\omega$, we call *p a path of* Π. The *string corresponding to path p in* Π, denoted $\Pi[p]$, is $a_1 a_2 a_3 \ldots \in \Sigma^\omega$ where $a_j = \Pi(p[1 \ldots (j-1)])$ for all $j \geq 1$.

An *infinite tree labelled with* Σ is a partial mapping $\Pi : N^* \rightharpoonup \Sigma$ such that for all $i \in N^*$ and $j \in N$: if $\Pi(i \cdot j)$ is defined then so are $\Pi(i)$ and $\Pi(i \cdot j')$ for all $j' < j$. The notation and concepts related to infinite trees are the same as those for infinite k-branching trees. $\square$

Definition 2.3 Fix a finite set $\mathcal{Z}$ of propositions, and define the set Σ of node labels as $P(\mathcal{Z})$. Furthermore, define $\mathcal{T}$ as the set of infinite k-branching Σ-labelled trees. $\square$

Let us recall definitions of modal mu-calculus syntax and semantics, and introduce some related notation. We suppose that the formulae discussed are already in the positive normal form as defined e.g. in [Sti92], consequently omitting the negation operator from the language, and that each variable is bound at most once. These restrictions are immaterial as any formula can be mechanically transformed into an equivalent one fulfilling them.

A simplification that has taken place in the language defined below is the lack of negated propositions. This does not affect the generality of the results for two reasons. Firstly, the correspondences between the relativised and unrelativised modalities outlined above do not introduce any instances of negated propositions. Secondly, the simplification could be dropped at the small extra cost of including negative requirements on the transitions of the Büchi-automata, as in [Dam92]. The same method would also cater for an infinite set of propositions $\mathcal{Z}$. However, the current approach leads to less complicated notation without affecting the results.

Definition 2.4 The formulae of the modal mu-calculus μK are defined by the following abstract syntax: $\phi ::= \top \mid \bot \mid z \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \Box\phi \mid \Diamond\phi \mid \mu z.\phi \mid \nu z.\phi$, where z varies over the set $\mathcal{Z}$. In any μK-formula ϕ any variable z is bound by at most one fixpoint operator. $\Box$

Definition 2.5 Let $T \in \mathcal{T}$ be an infinite k-branching Σ-labelled tree and ϕ a μK-formula. The set of nodes of T satisfying ϕ, denoted $\|\phi\|_T$, is defined inductively as follows: $\|\top\|_T = [k]^*$, $\|\bot\|_T = \emptyset$, $\|z\|_T = \{i \in [k]^* \mid z \in T(i)\}$, $\|\phi_1 \wedge \phi_2\|_T = \|\phi_1\|_T \cap \|\phi_2\|_T$, $\|\phi_1 \vee \phi_2\|_T = \|\phi_1\|_T \cup \|\phi_2\|_T$, $\|\Box\phi\|_T = \{i \in [k]^* \mid \forall j \in [k] : i \cdot j \in \|\phi\|_T\}$, $\|\Diamond\phi\|_T = \{i \in [k]^* \mid \exists j \in [k] : i \cdot j \in \|\phi\|_T\}$, $\|\mu z.\phi\|_T = \bigcap\{W \subseteq [k]^* \mid \|\phi\|_{T[W/z]} \subseteq W\}$ and $\|\nu z.\phi\|_T = \bigcup\{W \subseteq [k]^* \mid W \subseteq \|\phi\|_{T[W/z]}\}$, where $T[W/z]$ is defined by: $T[W/z](i) = T(i) \cup \{z\}$ if $i \in W$, $T[W/z](i) = T(i) \setminus \{z\}$ if $i \notin W$.

The *language characterised by* a μK-formula ϕ is $L(\phi) = \{T \in \mathcal{T} \mid \epsilon \in \|\phi\|_T\}$. $\Box$

Definition 2.6 Let ϕ_0 be a μK-formula. We write $\phi_0 \trianglelefteq \phi$ to mean that ϕ is a subformula of ϕ_0. If a variable z is bound in ϕ_0 by a ν (μ) fixpoint, we call it a ν (μ) variable, respectively, and write νz (μz) for the unique subformula $\nu z.\phi$ ($\mu z.\phi$) of ϕ_0 bound by the variable z. Furthermore, we use σ to refer to both fixpoint operators ν and μ. If z is a variable and $\phi_0 \trianglelefteq \phi$, an occurrence of z in ϕ is *bound* iff it is within the subformula σz and *free* otherwise. The set of variables occurring free in ϕ is denoted by free(ϕ). If ϕ and $\phi_1, \ldots, \phi_n$ are formulae and $z_1, \ldots, z_n$ variables, $\phi[\phi_1/z_1, \ldots, \phi_n/z_n]$ is the result of simultaneously substituting ϕ_i for all free occurrences of z_i for all $i \in \{1, \ldots, n\}$ in ϕ. If ϕ, ϕ' are formulae and z a variable, we say that ϕ' *is free for* z *in* ϕ iff there is no variable z' occurring free in ϕ' such that $\phi \trianglelefteq \sigma z' \triangleleft z$. $\Box$

In the following we often omit explicit reference to the ϕ_0 on which the notation of the previous definition depends, as it can be deduced from the context.

Definition 2.7 A *Büchi-automaton* A for k-branching trees is a 4-tuple $A = (Q, q_0, \Delta, F)$, where Q is a set of states, $q_0 \in Q$ is the initial state, $\Delta \subseteq Q \times \Sigma \times (Q \setminus \{q_0\})^k$ is the transition relation, and $F \subseteq Q$ is the set of accepting states of the automaton.

If $(q, Z, \bar{q}) \in \Delta$, we write $q \xrightarrow{Z} \bar{q}$. If there exists $\bar{q}$ such that $q \xrightarrow{Z} \bar{q}$, we write $q \xrightarrow{Z}$. For $q, q' \in Q$ and $Z \in \Sigma$, we write $q \xrightarrow{Z} q'$ iff there is a $\bar{q} \in Q^k$ such that $(q, Z, \bar{q}) \in \Delta$ and $q' = \bar{q}_j$ for some $j \in [k]$. We write $q \longrightarrow q'$ to mean that there is some $Z \in \Sigma$ such that $q \xrightarrow{Z} q'$, and use $\longrightarrow^*$ to denote the reflexive and transitive closure of $\longrightarrow$.

We say that A *makes reference to* $z \in \mathcal{Z}$ iff there is a $q \in Q$ and $Z \in \Sigma$ such that $q \xrightarrow{Z}$ and $z \in Z$. The set of variables A makes reference to is denoted by $\mathrm{ref}(A)$. $\Box$

The definition of a run of a Büchi-automaton used here does not require the transition labels of the automaton and the labels of the input tree to match exactly. Rather, a transition label Z specifies that all the propositions in Z need to hold if the transition is to be taken, but some other propositions may hold as well.

Definition 2.8 Let $A = (Q, q_0, \Delta, F)$ be a Büchi-automaton and $T \in \mathcal{T}$ a tree. A *run of A on T* is an infinite k-branching tree π labelled with Δ fulfilling:

- $\pi(\epsilon) = (q_0, Z, \bar{q})$ for some Z, $\bar{q}$.
- for all $i \in [k]^*$, if $\pi(i) = (q, Z, \bar{q})$, then $Z \subseteq T(i)$
- for all $i \in [k]^*$ and all $j \in [k]$, if $\pi(i) = (q, Z, \bar{q})$ and $\pi(i \cdot j) = (q', Z', \bar{q}')$, then $q' = \bar{q}_j$.

If π is a run of A on T, we define mappings π^{fr}, π^{lab} and π^{to} by: for all $i \in [k]^*$, if $\pi(i) = (q, Z, \bar{q})$, then $\pi^{\mathrm{fr}}(i) = q$, $\pi^{\mathrm{lab}}(i) = Z$ and $\pi^{\mathrm{to}}(i) = \bar{q}$.

A run π of A is *accepting* iff for all paths $p \in [k]^\omega$ there exists a $q \in F$ such that $\pi^{\mathrm{fr}}[p]_j = q$ for infinitely many $j \in N$.

The *language recognised by* A is $L(A) = \{T \in \mathcal{T} \mid A \text{ has an accepting run } \pi \text{ on } T\}$.
$\Box$

3 Alternation depth in modal mu-calculus

In recursion theory the alternation of existential and universal quantification results in the arithmetical and analytical hierarchies. A related phenomenon takes place in mu-calculi in the form of alternation of minimal and maximal fixpoints. Besides the intriguing open theoretical questions related to the alternation-depth classes in modal mu-calculus, such as whether the hierarchy is proper or not, the issue has also practical significance: all known model-checking algorithms for modal mu-calculus are at least exponential in the alternation depth of the formula to be verified.

In the simplest form, the alternation depth classes can be defined just by looking at the syntactic nesting of fixpoint formulae.

Definition 3.1 $\Sigma^S_0 = \Pi^S_0$ is the set of μK-formulae having no fixpoint operators.

For all $n \in N$, Σ^S_{n+1} is the least set of μK-formulae fulfilling:

1 $\Sigma^S_n \cup \Pi^S_n \subseteq \Sigma^S_{n+1}$

2 if $\phi \in \Sigma^S_{n+1}$, then $\mu z.\phi \in \Sigma^S_{n+1}$

3 if $\phi_1, \phi_2 \in \Sigma^S_{n+1}$, then $\{\phi_1 \wedge \phi_2, \phi_1 \vee \phi_2, \Box\phi_1, \Diamond\phi_1\} \subseteq \Sigma^S_{n+1}$.

Π^S_{n+1} is defined symmetrically, by changing μ to ν in condition 2. $\Box$

In some respects this is a very rough way of measuring the alternation present in a formula. In particular, no difference is made between formulae of the types $\nu z.\mu x.\phi$ and $\nu z.\mu x.\phi'$, where z occurs free in ϕ but not in ϕ'. However, in the former case the subformula bound by μx is dependent on variable z whereas in the latter case it is not, although in both cases the subformulae are embedded in the scope of the νz-fixpoint. These considerations lead to a more liberal approach to defining alternation-depth classes. Intuitively, in looking at the nesting of fixpoints in a formula, we no longer pay attention to nesting in its closed subformulae. Introduced in [EL86], this approach is perhaps the most widely used in the modal mu-calculus literature. In the present framework, this idea leads to the following definition.

Definition 3.2 Let us divide the set $\mathcal{Z}$ of propositions into two disjoint parts: $\mathcal{Z}_c$, the set of *propositional constants*, and $\mathcal{Z}_v$, the set of *propositional variables*, and let us decree that only propositions of the latter kind can be bound by fixpoint operators. Furthermore, let us call a formula ϕ *closed* iff it has no free occurrences of any variable $z \in \mathcal{Z}_v$.

Let us define now $\Sigma_0^{EL} = \Pi_0^{EL}$ as the set of μK-formulae having no fixpoint operators.

For all $n \in N$, Σ_{n+1}^{EL} is the least set of μK-formulae fulfilling conditions 1–3 of 3.1 and:

4 if $\phi, \phi_1, \ldots, \phi_m \in \Sigma_{n+1}^{EL}$, $z_1, \ldots, z_m$ are distinct variables, and for all $1 \leq i \leq m$, ϕ_i are closed formulae, then $\phi[\phi_1/z_1, \ldots, \phi_m/z_m] \in \Sigma_{n+1}^{EL}$.

Π_{n+1}^{EL} is defined symmetrically, by changing μ to ν in condition 2. □

However, a still more liberal definition can be obtained by allowing arbitrary substitutions in the condition 4 above. This is the approach taken in [Niw86], and also the one we are going to adopt.

Definition 3.3 $\Sigma_0 = \Pi_0$ is the set of μK-formulae having no fixpoint operators.

For all $n \in N$, Σ_{n+1} is the least set of μK-formulae fulfilling conditions 1–3 of 3.1 and:

4' if $\phi, \phi_1, \ldots, \phi_m \in \Sigma_{n+1}$, $z_1, \ldots, z_m$ are distinct variables, and for all $1 \leq i \leq m$, ϕ_i is free for z_i in ϕ, then $\phi[\phi_1/z_1, \ldots, \phi_m/z_m] \in \Sigma_{n+1}$.

Π_{n+1} is defined symmetrically, by changing μ to ν in condition 2. □

Notice that here condition 3 is actually superfluous. It has been retained only to show the relation to the other formalisations. Intuitively, this definition simply captures the idea that an alternation depth class contains all the lower classes in the hierarchy and is closed under normal substitutions and prefixing with the relevant fixpoint.

To see the difference between the classes Π_i and Π_i^{EL}, consider the following formula:

$$\phi = \nu w.\mu x.(\Diamond(a \wedge x) \vee \nu y.\mu z(\Diamond(b \wedge (w \vee z)) \vee \Diamond(c \wedge y)))$$

where we assume that a, b, c are propositional constants. Notice that $\phi \in \Pi_2$, but $\phi \notin \Pi_2^{EL}$, since the subformula bound by νy is not closed. We shall return to contrasting the respective merits of the various definitions in section 5.

For the purposes of the results here, we shall need one fact concerning Π_2-formulae.

Proposition 3.4 Let ϕ_0 be a μK-formula. Then $\phi_0 \in \Pi_2$ iff for all variables z and z' such that $\phi_0 \trianglelefteq \mu z' \triangleleft \nu z$, z' does not occur free in νz.

Proof: Straightforward. □

4 From formulae to automata

4.1 Intuition

In this section we describe a construction mapping μK-formulae to Büchi-automata and show that for a given formula ϕ and the corresponding automaton A_ϕ, $L(\phi) = L(A_\phi)$.

The method of constructing A_ϕ is inductive: for each basic μK-formula we define a corresponding automaton directly, and for each boolean, modal and fixpoint operator we describe a method for obtaining the required automaton from the automata corresponding to the formulae the operator is applied to. Here the fixpoint operators are naturally the essential problem. The constructions relevant to them are based on the powerset construction of Dam [Dam92] and the methods of Niwinski [Niw86].

However, it is well known that the class of Büchi-recognisable tree languages is not closed under complementation [Tho89]. This means that there is no hope of devising such a general construction for all the operators, as then we would have a correspondence between the full mu-calculus and the Büchi-automata, which is clearly a contradiction. It turns out that the operator for which no such general construction can be given is the minimal fixpoint operator μ.

A crucial observation to the proof here is that the automata corresponding to Π_2-formulae have a certain structural property that makes it possible to devise a construction corresponding to the minimal fixpoint operator in this particular case.

Definition 4.1 Let $A = (Q, q_0, \Delta, F)$ be a Büchi-automaton and $z \in \mathcal{Z}$. We say that *A makes reference to z only before accepting states* iff there are no $q \in F$, $q' \in Q$, $Z \in \Sigma$ such that $q_0 \longrightarrow^* q \longrightarrow^* q' \xrightarrow{Z}$ in A and $z \in Z$. □

It turns out that all the automata corresponding to subformulae of a Π_2-formula ϕ_0 make reference to any μ-variable of it only before accepting states.

Given a formula $\phi_0 \in \Pi_2$, we show by induction that for all ϕ such that $\phi_0 \trianglelefteq \phi$, we can construct the required automaton A_ϕ. In each step of the induction proof we show that for the current ϕ we can construct the A_ϕ so that both $L(\phi) = L(A_\phi)$ and A_ϕ meets the relevant structural property, provided that we know how to do this for the immediate subformulae of ϕ. For technical reasons we need to show simultaneously that the constructions guarantee that $\mathrm{free}(\phi) = \mathrm{ref}(A_\phi)$.

4.2 Basic constructions

In this section we briefly describe the automata corresponding to the basic μK-formulae and the constructions corresponding to the boolean and modal operators.

Definition 4.2

$$\begin{array}{rcll}
A_\top & = & (\{q_0, q_1\}, q_0, \Delta, \{q_1\}), & \text{where } \Delta = \{q_0 \xrightarrow{\emptyset} (q_1, \ldots, q_1), q_1 \xrightarrow{\emptyset} (q_1, \ldots, q_1)\} \\
A_\perp & = & (\{q_0\}, q_0, \emptyset, \emptyset) & \\
A_z & = & (\{q_0, q_1\}, q_0, \Delta, \{q_1\}), & \text{where } \Delta = \{q_0 \xrightarrow{\{z\}} (q_1, \ldots, q_1), q_1 \xrightarrow{\emptyset} (q_1, \ldots, q_1)\}
\end{array}$$

□

Lemma 4.3 For all $\phi \in \{\top, \bot\} \cup \mathcal{Z}$, there exists an automaton A_ϕ such that $L(A_\phi) = L(\phi)$, $\mathrm{ref}(A) = \mathrm{free}(\phi)$ and A_ϕ makes reference to any z' only before accepting states,

Proof: Trivial. $\Box$

The constructions for the boolean operators are standard [Tho89].

Definition 4.4 Let $A_1 = (Q_1, q_{01}, \Delta_1, F_1)$ and $A_2 = (Q_2, q_{02}, \Delta_2, F_2)$ be Büchi-automata. Then $A_1 \vee A_2$ is the Büchi-automaton $(Q_1 \uplus Q_2 \uplus \{q_0\}, q_0, \Delta, F_1 \cup F_2)$, where q_0 is an arbitrary element not in $Q_1 \uplus Q_2$ and

$$\Delta = \Delta_1 \cup \Delta_2 \cup \{q_0 \xrightarrow{Z} \bar{q} \mid q_{01} \xrightarrow{Z} \bar{q} \in \Delta_1\} \cup \{q_0 \xrightarrow{Z} \bar{q} \mid q_{02} \xrightarrow{Z} \bar{q} \in \Delta_2\}$$

$A_1 \wedge A_2$ is the Büchi-automaton $(Q_1 \times Q_2 \times \{0,1,2\}, (q_{01}, q_{02}, 0), \Delta, Q_1 \times Q_2 \times \{2\})$, where $(q_1, q_2, i) \xrightarrow{Z_1 \cup Z_2} ((q_{11}, q_{21}, i_1), \ldots, (q_{1k}, q_{2k}, i_k)) \in \Delta$ iff

- $q_1 \xrightarrow{Z_1} (q_{11}, \ldots, q_{1k}) \in \Delta_1$ and $q_2 \xrightarrow{Z_2} (q_{21}, \ldots, q_{2k}) \in \Delta_2$, and
- for all $j \in [k]$: if $i \in \{0,2\}$ and $q_{1j} \notin F_1$ then $i_j = 0$, else if $q_{2j} \notin F_2$ then $i_j = 1$, else $i_j = 2$. $\Box$

Definition 4.5 Let $A = (Q, q_0, \Delta, F)$ be a Büchi-automaton. Then $\Box A$ is the Büchi-automaton $(Q \cup \{q_0'\}, q_0', \Delta', F)$, where q_0' is an element not in Q, and $\Delta' = \Delta \cup \{q_0' \xrightarrow{\emptyset} (q_0, \ldots, q_0)\}$

$\Diamond A$ is the Büchi-automaton $(Q \cup \{q_0', q_a\}, q_0', \Delta', F \cup \{q_a\})$, where q_0', q_a are not in Q, and

$$\Delta' \;=\; \Delta \cup \{q_0' \xrightarrow{\emptyset} \bar{q} \mid \exists i : \bar{q}_i = q_0 \;\wedge\; \forall j \neq i : \bar{q}_j = q_a\} \cup \{q_a \xrightarrow{\emptyset} (q_a, \ldots, q_a)\}$$

$\Box$

Lemma 4.6 Let A_1 and A_2 be Büchi-automata, ϕ_1 and ϕ_2 μK-formulae, and $z \in \mathcal{Z}$.

If $L(A_1) = L(\phi_1)$ and $L(A_2) = L(\phi_2)$, then $L(A_1 \vee A_2) = L(\phi_1 \vee \phi_2)$, $L(A_1 \wedge A_2) = L(\phi_1 \wedge \phi_2)$, $L(\Box A_1) = L(\Box \phi_1)$ and $L(\Diamond A_1) = L(\Diamond \phi_1)$.

If $\mathrm{ref}(A_1) = \mathrm{free}(\phi_1)$ and $\mathrm{ref}(A_2) = \mathrm{free}(\phi_2)$, then $\mathrm{ref}(A_1 \vee A_2) = \mathrm{free}(\phi_1 \vee \phi_2)$, $\mathrm{ref}(A_1 \wedge A_2) = \mathrm{free}(\phi_1 \wedge \phi_2)$, $\mathrm{ref}(\Box A_1) = \mathrm{free}(\Box \phi_1)$ and $\mathrm{ref}(\Diamond A_1) = \mathrm{free}(\Diamond \phi_1)$.

If A_1 and A_2 make reference to z only before accepting states, then so do $L(A_1 \vee A_2)$, $L(A_1 \wedge A_2)$, $L(\Box A_1)$ and $L(\Diamond A_1)$ as well.

Proof: Clear from the constructions. $\Box$

Example 4.7 Figure 1 contains some Büchi-automata for 2-branching trees corresponding to simple formulae. In this picture and the later ones we have used the following conventions.

Nodes are represented by circles and accepting nodes by filled circles. The initial node is the topmost one. Transitions are represented by branching arrows, which point to the two target nodes of the transition. For any transition $q \xrightarrow{Z} \bar{q}$ in our automata, the automaton has also the transitions $q \xrightarrow{Z} \bar{q}'$ for all permutations $\bar{q}'$ of $\bar{q}$. To keep the pictures readable, only one transition from any such group is shown. Furthermore,

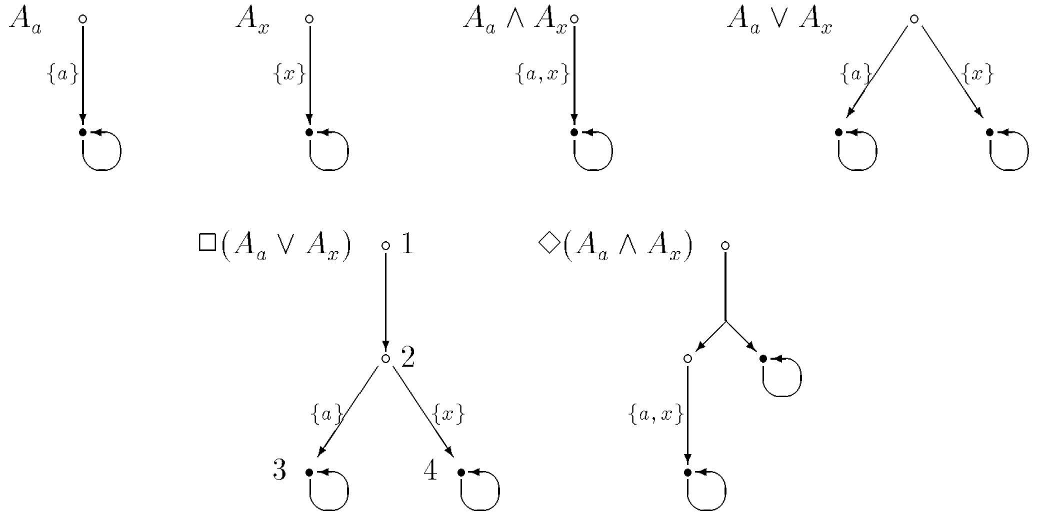


Figure 1: Büchi-automata corresponding to simple formulae

if both target nodes of a transition are the same, it is drawn as a simple, non-branching arrow. Transition labels are written next to the arrow, omitting $\emptyset$ labels.

To illustrate the relation of the pictures to the mathematical structure of the automata, notice e.g. that $A_a = (\{0,1\}, 0, \{0 \xrightarrow{\{a\}} (1,1), 1 \xrightarrow{\emptyset} (1,1)\}, \{1\})$, and that $\Diamond(A_a \wedge A_x) = (\{0,1,2,3\}, 0, \{0 \xrightarrow{\emptyset} (1,2), 0 \xrightarrow{\emptyset} (2,1), 1 \xrightarrow{\{a,x\}} (3,3), 2 \xrightarrow{\emptyset} (2,2), 3 \xrightarrow{\emptyset} (3,3)\}, \{2,3\})$. $\Box$

4.3 Fixpoints

The constructions corresponding to both fixpoint operators are based on a powerset construction introduced in [Dam92]. The intuition behind this approach can be best described as follows. Suppose that we have an automaton A corresponding to the formula ϕ and that we would like to construct an automaton corresponding to the maximal fixpoint $\nu z.\phi$. In order to decide whether a given tree $T \in \mathcal{T}$ is a model of $\nu z.\phi$, we start by running nondeterministically A down T. Each time A requires z to be true, we start a new copy of A running down that particular subtree. If all these runs are accepting, T is a model of $\nu z.\phi$. In general, the further we are from the start, the more copies of the automaton A there will be running down any particular subtree: in fact, their number is not bounded. However, if we have two copies of A running down the same subtree and they are both in the same state, then certainly if one of them has an accepting run down that subtree, then so does the other. Consequently, these two copies can be amalgamated into a single one. But this means that at any node of T we need at most $|Q|$ different copies of A running down the subtree rooted on that node. If we know that for all the runs of these copies of A every path of the run is an accepting one, we know again that T is a model of $\nu z.\phi$, and vice versa.

The only difference between the maximal and the minimal fixpoints is that in the minimal case, in addition to the above, we need to ascertain that we do not have an infinite sequence $A_1, A_2, \ldots$ of copies of A so that each A_j would require the starting of the new copy A_{j+1}.

In the following definition of an intermediate automaton, the states are sets of states of A, intuitively possible sets of copies of A. Each transition of the intermediate automaton corresponds to a set of simultaneous single transitions of A, one for each copy of A in the initial state of the transition. To be able to check that all the runs of the copies of A are accepting ones, we attach to each transition of the intermediate automaton the set of transitions of A it corresponds to.

Definition 4.8 Let $A = (Q, q_0, \Delta, F)$ be a Büchi-automaton and z a μK-variable. The *intermediate automaton* $\mathrm{fix}_z A$ *based on* A is a triple (Q', q_0', Δ'), where

- $Q' = P(Q)$ is the set of states,
- $q_0' = \{q_0\}$ is the initial state,
- $\Delta' \subseteq Q' \times \Sigma \times Q'^k \times (Q \rightharpoonup \Delta)$ is the transition relation,

and where $(P, Z, \overline{P}, \delta) \in \Delta'$ iff

1 The domain of δ is $P \subseteq Q$, i.e. δ can be viewed as a function $\delta : P \rightarrow \Delta$. For all $q \in P$ we define the functions δ^{fr}, δ^{lab} and δ^{to} by: if $\delta(q) = (q', Z', \vec{q}')$, then $\delta^{\mathrm{fr}}(q) = q'$, $\delta^{\mathrm{lab}}(q) = Z'$ and $\delta^{\mathrm{to}}(q) = \vec{q}'$.

2 For all $q \in P$, $\delta^{\mathrm{fr}}(q) = q$.

3 $q_0 \in P$ iff $P = \{q_0\}$ or there is a $q \in P \setminus \{q_0\}$ such that $z \in \delta^{\mathrm{lab}}(q)$.

4 $Z = \bigcup_{q \in P} \delta^{\mathrm{lab}}(q) \setminus \{z\}$.

5 For all $j \in [k]$, $\overline{P}_j \setminus \{q_0\} = \{\delta^{\mathrm{to}}(q)_j \mid q \in P\}$.

If $(P, Z, \overline{P}, \delta) \in \Delta'$, we write $P \xrightarrow{Z,\delta} \overline{P}$.

The concept of a *run* Π *of* $\mathrm{fix}_z A$ *on a tree* $T \in \mathcal{T}$, and the related notation Π^{fr}, Π^{lab} and Π^{to} are defined as in 2.8. $\square$

Intuitively, the conditions 1 and 2 above say that δ gives for each $q \in P$ a transition of Δ leaving from q. Condition 3 says that a new copy of A is started in the beginning and whenever a running copy of it requires z to be true, and that these are the only cases when a new copy is started. Condition 4 expresses the fact that a transition of the intermediate automaton is possible iff all the corresponding transitions of A are possible, ignoring the current fixpoint variable. Finally, condition 5 says that the state of the intermediate automaton after the transition is specified by the target states of the transitions $\delta(q)$. However, a new copy of A may be started, to be available if in the next transition some running copy of it requires z to hold.

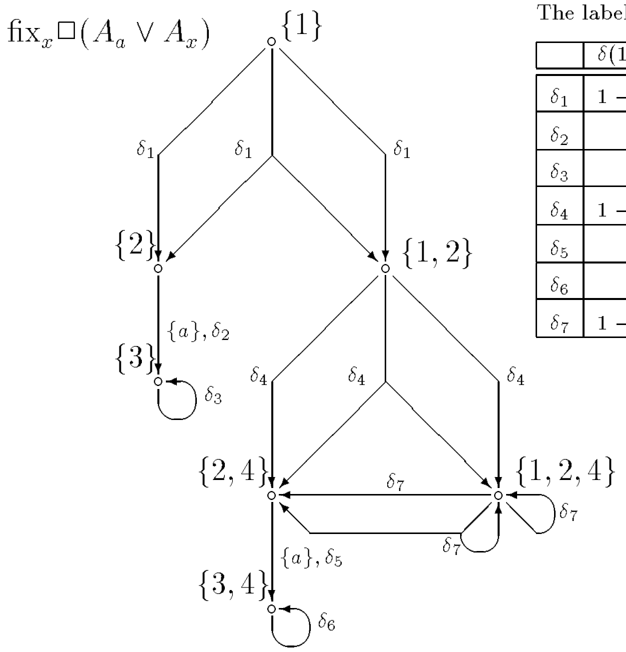
In order to decide whether a given a run Π of $\mathrm{fix}_z A$ corresponds to accepting runs of A in the sense outlined in the beginning of the section, we need to be able to extract structures corresponding to these individual runs of A from Π. This is captured by the following definition.

Definition 4.9 Let $A = (Q, q_0, \Delta, F)$ be a Büchi-automaton, $T \in \mathcal{T}$, Π a run of $\mathrm{fix}_z A$ on T, $t \in [k]^*$, $\Pi(t) = (P, Z, \overline{P}, \delta)$, and $q \in P$. The *trail of* Π *starting from point* t *and node* q, denoted $\pi(t, q)$, is the unique mapping $\pi(t, q) : [k]^* \rightarrow \Delta$, such that

- $\pi(t, q)(\epsilon) = \delta(q)$, and
- for all $i \in [k]^*$ and all $j \in [k]$, $\pi(t, q)(i \cdot j) = \delta'(\pi(t, q)^{\mathrm{to}}(i)_j)$, where $\Pi(t \cdot i \cdot j) = (P', Z', \overline{P}', \delta')$

The notations $\pi(t, q)^{\mathrm{fr}}$, $\pi(t, q)^{\mathrm{lab}}$ and $\pi(t, q)^{\mathrm{to}}$ are as in 2.8. $\square$

Notice that if Π is a run of $\mathrm{fix}_z A$ on T, $\Pi(t) = (P, Z, \overline{P}, \delta)$ and $q_0 \in P$, the only difference between $\pi(t, q_0)$ and a run of A on T^t is with respect to z:s on the transition labels.



The labelling functions δ are:

	$\delta(1)$	$\delta(2)$	$\delta(3)$	$\delta(4)$
δ_1	$1 \xrightarrow{\emptyset} (2,2)$			
δ_2		$2 \xrightarrow{\{a\}} (3,3)$		
δ_3			$3 \xrightarrow{\emptyset} (3,3)$	
δ_4	$1 \xrightarrow{\emptyset} (2,2)$	$2 \xrightarrow{\{x\}} (4,4)$		
δ_5		$2 \xrightarrow{\{a\}} (3,3)$		$4 \xrightarrow{\emptyset} (4,4)$
δ_6			$3 \xrightarrow{\emptyset} (3,3)$	$4 \xrightarrow{\emptyset} (4,4)$
δ_7	$1 \xrightarrow{\emptyset} (2,2)$	$2 \xrightarrow{\{x\}} (4,4)$		$4 \xrightarrow{\emptyset} (4,4)$

Figure 2: An intermediate automaton

Example 4.10 Figure 2 depicts the intermediate automaton fix$_x\Box(A_a \vee A_x)$ based on the Büchi-automaton $\Box(A_a \vee A_x)$ constructed earlier in example 4.7. The labels adjacent to the states of the intermediate automaton in the picture specify the set of states of the original automaton they correspond to. The numbering of states refers to figure 1. The transitions are labelled with the transition condition Z and the function δ specifying the corresponding transitions of the original automaton, again omitting $\emptyset$ transition conditions. $\Box$

4.4 Maximal fixpoints

According to the intuitions explained in the previous section, in order to decide whether a tree $T \in \mathcal{T}$ is a model of $\nu z.\phi$, we have to check that all the runs of the individual copies of A within Π are accepting ones. As these runs correspond to the trails of Π, this leads to the following definition.

Definition 4.11 Let $A = (Q, q_0, \Delta, F)$ be a Büchi-automaton, $T \in \mathcal{T}$ and Π a run of fix$_z A$ on T. We say that Π is *ν-accepting* iff for all $t \in [k]^*$, all trails $\pi(t,q)$ of Π starting from point t and all paths $p \in [k]^\omega$ of $\pi(t,q)$, there is a $q' \in F$ such that $\pi(t,q)^{\mathrm{fr}}[p]_n = q'$ for infinitely many $n \in N$. $\Box$

The following result states that the formulation of an intermediate automaton and its run being ν-accepting capture the intended meaning. The intuition behind the proof is not very involved in itself. However, the proof has been spelled out in some technical detail, as the construction of the intermediate automaton is the core point in the inductive mapping from formulae to automata.

Lemma 4.12 Let ϕ be a μK-formula and A a Büchi-automaton such that $L(\phi) = L(A)$. Then for all $T \in \mathcal{T}$, $T \in L(\nu z.\phi)$ iff fix$_z A$ has a ν-accepting run on T.

Proof: Let $T \in \mathcal{T}$ be an arbitrary tree. By definition, $T \in L(\nu z.\phi)$ iff there is an $S \subseteq [k]^*$ such that $\epsilon \in S$ and $T[S/z]^t \in L(\phi)$ for all $t \in S$. In the proof we shall show that the latter holds iff $\mathrm{fix}_z A$ has a ν-accepting run Π on T.
" => "

Suppose that there is a set S having the above properties, and write $T' = T[S/z]$. Take any $t \in S$. By definition of S, $T'^t \in L(\phi) = L(A)$, which means by assumption that A has an accepting run π_t on T'^t. On the basis of these π_t:s we are now going to construct a ν-accepting run Π of $\mathrm{fix}_z A$.

The essential problem here is that although two runs π_t and $\pi_{t'}$ would coincide in some node i of the input, i.e. $\pi_t^{\mathrm{fr}}(i-t) = \pi_{t'}^{\mathrm{fr}}(i-t')$, there is no reason why they would coincide in all the subsequent nodes. This means that we cannot simply amalgamate π_t and $\pi_{t'}$ into a single run from i onwards. Instead, we choose either π_t or $\pi_{t'}$, say π_t, and pretend that from i onwards $\pi_{t'}$ coincides with π_t. Clearly, since both π_t and $\pi_{t'}$ are accepting runs of A, this replacing of a part of $\pi_{t'}$ by a part of π_t still yields an accepting run of A. The same holds for any finite sequence of successive replacements of parts of $\pi_{t'}$. However, for an infinite sequence of such replacements, the resulting run of A is not necessarily an accepting one any more. Consequently, we have to be careful with respect to which of π_t or $\pi_{t'}$ we choose to follow from i onwards: we always take the smaller one of t and t' in the prefix-ordering. This and the well-foundedness of the prefix-ordering guarantee that there will be no t such that an infinite number of successive replacements would be made to parts of π_t.

The run Π of $\mathrm{fix}_z A$ is built inductively starting from the root. At each node i the runs π_t that could possibly be relevant to $\Pi(i)$ are those for which $t \preceq i$. Whenever two runs π_t and $\pi_{t'}$ are in the same state at node i, we follow the one of them with the smaller index t and ignore the other at node i and all its successors. More formally, the set of t:s such that π_t is relevant to $\Pi(i)$ is expressed by a mapping $V : [k]^* \to P([k]^*)$ defined inductively by: $V(\epsilon) = \{\epsilon\}$ and for all $i \in [k]^*$ and all $j \in [k]$,

$$V(i \cdot j) = \begin{cases} V'(i \cdot j) \cup \{i \cdot j\} & \text{if there is a } t \in V'(i \cdot j) \text{ such that } z \in \pi_t^{lab}(i \cdot j - t) \\ V'(i \cdot j) & \text{otherwise} \end{cases}$$

where

$$V'(i{\cdot}j) = \{t \in V(i) \mid \text{there is no } t' \in V(i) \text{ such that } t' \prec t \text{ and } \pi_{t'}^{\mathrm{fr}}(i{\cdot}j{-}t') = \pi_t^{\mathrm{fr}}(i{\cdot}j{-}t)\}$$

Notice that if there is a $t \in V'(i \cdot j)$ such that $z \in \pi_t^{\mathrm{lab}}(i \cdot j - t)$, then $z \in T'^t(i \cdot j)$, as π_t is a run of A on T'^t. By the definition of T', this implies $i \cdot j \in S$. Consequently, $V(i) \subseteq S$ for all $i \in [k]^*$.

Having now defined the runs π_t that are relevant to a given node i, the set of states these runs are in at i is easy to define. This is expressed by a mapping $P : [k]^* \to P(Q)$ defined by $P(i) = \{\pi_t^{\mathrm{fr}}(i-t) \mid t \in V(i)\}$. Thanks to the definition of V and the fact that no transition of A can lead back to q_0, for each $i \in [k]$ and each $q \in P(i)$ there is a unique $t \in V(i)$ such that $\pi_t^{\mathrm{fr}}(i-t) = q$. Let us denote this unique t by $t(i,q)$, for any given i and q.

The run Π of $\mathrm{fix}_z A$ is now defined by: for all $i \in [k]^*$, $\Pi(i) = (P, Z, \overline{P}, \delta)$, where

- $P = P(i)$,
- $Z = \bigcup_{q \in P} \pi_{t(i,q)}^{\mathrm{lab}}(i - t(i,q)) \setminus \{z\}$,
- $\overline{P}_j = P(i \cdot j)$ for all $j \in [k]$, and

- $\delta(q) = \pi_{t(i,q)}(i - t(i,q))$ for all $q \in P$.

To see that Π indeed is a run of $\mathrm{fix}_z A$ on T, we need to ensure that for all i, $\Pi(i) \in \Delta'$. Once this is done, it is easy to see that the other requirements of Π being a run are fulfilled.

It should be easy to see that $\Pi(i)$ meets the conditions 1, 2 and 4 of 4.8. To see that condition 3 is met, notice first that for all $t \in S$ and all i, $\pi_t(i) = q_0$ iff $i = \epsilon$. Take any $i \in [k]^*$ and write $\Pi(i) = (P, Z, \overline{P}, \delta)$, where $P = P(i)$. By the previous observation $q_0 \in P(i)$ iff there is a $t \in V(i)$ such that $\pi_t(i-t) = q_0$ iff $i \in V(i)$ iff $i = \epsilon$ or there is a $t \in V(i) \setminus \{i\}$ such that $z \in \pi_t^{\mathrm{lab}}(i-t)$ iff $P(i) = \{q_0\}$ or there is a $p \in P(i) \setminus \{q_0\}$ such that $z \in \delta^{\mathrm{lab}}(q)$.

To see that also condition 5 is met, notice first that for all $i \in [k]^*$ and all $j \in [k]$, $\{\pi_t^{\mathrm{fr}}(i \cdot j - t) \mid t \in V(i)\} = \{\pi_t^{\mathrm{fr}}(i \cdot j - t) \mid t \in V(i \cdot j) \setminus \{i \cdot j\}\}$. Consequently, for any $i \in [k]^*$ and any $j \in [k]$, if $\Pi(i) = (P, Z, \overline{P}, \delta)$, $\overline{P}_j = P(i \cdot j) \setminus \{q_0\} =$
$\{\pi_t^{\mathrm{fr}}(i \cdot j - t) \mid t \in V(i \cdot j)\} \setminus \{q_0\} = \{\pi_t^{\mathrm{fr}}(i \cdot j - t) \mid t \in V(i \cdot j) \setminus \{i \cdot j\}\} =$
$\{\pi_t^{\mathrm{fr}}(i \cdot j - t) \mid t \in V(i)\} = \{\pi_t^{\mathrm{to}}(i - t)_j \mid t \in V(i)\} = \{\pi_{t(i,q)}^{\mathrm{to}}(i - t(i,q))_j \mid q \in P(i)\} =$
$\{\delta^{\mathrm{to}}(q)_j \mid q \in P(i)\} = \{\delta^{\mathrm{to}}(q)_j \mid q \in P\}$.

To see that Π is ν-accepting, take any $t \in [k]^*$, any trail $\pi = \pi(t,q)$ of Π starting from node t, and a path $p \in [k]^\omega$ of π. We will show that eventually the path $\pi[p]$ coincides with a path of one of the accepting runs $\pi_{t'}$, or more formally that there is some $t' \preceq t$ and $n \in N$ such that for all $m \geq n$, $\pi^{\mathrm{fr}}[p]_m = \pi_{t'}^{\mathrm{fr}}[(t - t') \cdot p]_{|t-t'|+m}$. From this the required result follows immediately.

Notice first that by definition of $t(,)$ and V, for all $i \in [k]^*$, all $q \in P(i)$ and all $j \in [k]$, $t(i \cdot j, \pi_{t(i,q)}^{\mathrm{to}}(i - t(i,q))_j) \preceq t(i,q)$. Notice also that for this to hold it is crucial that whenever two runs coincide at a given node, we always discard the one started more recently: otherwise the claim would not hold.

If now $\Pi(i) = (P, Z, \overline{P}, \delta)$ and $q \in P$, the previous observation implies that $t(i \cdot j, \delta^{\mathrm{to}}(q)_j) \preceq t(i,q)$. Consequently, for all $i \in [k]$, $j \in [k]$, $t(t \cdot i \cdot j, \pi^{\mathrm{fr}}(i \cdot j - t)) \preceq t(t \cdot i, \pi^{\mathrm{fr}}(i - t))$. In particular, for all $m \in N$,

$$t(t \cdot p[1 \ldots (m+1)], \pi^{\mathrm{fr}}(p[1 \ldots (m+1)])) \preceq t(t \cdot p[1 \ldots m], \pi^{\mathrm{fr}}(p[1 \ldots m]))$$

As $\preceq$ is a well-founded ordering, this implies the existence of an $n \in N$ such that for all $m \geq n$

$$t(t \cdot p[1 \ldots n], \pi^{\mathrm{fr}}(p[1 \ldots n])) = t(t \cdot p[1 \ldots m], \pi^{\mathrm{fr}}(p[1 \ldots m]))$$

Defining now $t' = t(t \cdot p[1 \ldots n], \pi^{\mathrm{fr}}(p[1 \ldots n]))$, we have for all $m \geq n$

$$\begin{aligned}
\pi^{\mathrm{fr}}[p]_m &= \pi^{\mathrm{fr}}(p[1 \ldots m]) \\
&= \pi^{\mathrm{fr}}_{t(t \cdot p[1 \ldots m], \pi^{\mathrm{fr}}(p[1 \ldots m]))}(t \cdot p[1 \ldots m] - t(t \cdot p[1 \ldots m], \pi^{\mathrm{fr}}(p[1 \ldots m]))) \\
&= \pi_{t'}^{\mathrm{fr}}(t \cdot p[1 \ldots m] - t') \\
&= \pi_{t'}^{\mathrm{fr}}((t - t') \cdot p[1 \ldots m]) \\
&= \pi_{t'}^{\mathrm{fr}}[(t - t') \cdot p]_{|t-t'|+m}
\end{aligned}$$

just as required.
" <= "

Suppose then that we have a ν-accepting run Π of $\mathrm{fix}_z A$. We have to show the existence of a set $S \subseteq [k]^*$ having the required properties. Let us define

$$S = \{i \in [k]^* \mid q_0 \in \Pi^{\mathrm{fr}}(i)\}$$

Writing $T' = T[S/z]$, we need to show that for each $t \in S$, $T'^t \in L(\phi) = L(A)$, i.e. that for each $t \in S$ there is an accepting run of A on T'^t. This is easy, as we just have to take the trail $\pi(t, q_0)$ as π_t. It is clear that each π_t is a run of A on T'^t. As Π is ν-accepting, each π_t is clearly accepting, as well. □

In order to construct a Büchi-automaton for $\nu z.\phi$ on the basis of the intermediate automaton $\mathrm{fix}_z A$, we need to express the property of being ν-accepting in a manner that is more 'path-oriented', closer to the form of Büchi acceptance conditions.

Definition 4.13 Let $A = (Q, q_0, \Delta, F)$ be a Büchi-automaton, $T \in \mathcal{T}$ and Π a run of $\mathrm{fix}_z A$ on T. We say that Π is ν'*-accepting* iff for every path $p \in [k]^\omega$ of Π there is an infinite strictly increasing sequence $t_1, t_2, \ldots$ of finite prefixes of p such that for every $n \in N$ and for every $q \in \Pi^{\mathrm{fr}}(t_n)$, there is a t' such that $t_n \prec t' \prec t_{n+1}$ and $\pi^{\mathrm{fr}}(t_n, q)(t' - t_n) \in F$. □

Lemma 4.14 Let Π be a run of $\mathrm{fix}_z A$. Then Π is ν-accepting iff Π is ν'-accepting.

Proof: Straightforward. □

Intuitively, the maximal fixpoint automaton works like the intermediate automaton equipped with a mechanism for checking that for each path p of the run there is a strictly increasing sequence $t_1 \prec t_2 \prec \ldots \prec p$ such that the requirements of ν'-acceptability are fulfilled. The states of the maximal fixpoint automaton are pairs (P, P'), where P is a state of the intermediate automaton, i.e. a set of states of the original automaton, and P' a subset of P. Intuitively, P expresses the different copies of the original automaton currently running and P' the copies we are currently waiting to reach an accepting state. The transitions correspond to those of the intermediate automaton, and the second component of a state is updated by removing from it all the elements in the acceptance set of the original automaton. When the second element P' becomes empty, i.e. we have seen an accepting state for all the copies of the original automaton for which we were looking for such a state, we start again waiting for accepting states for all the currently running copies. If a path of a run leads from a state $(P_1, \emptyset)$ to a state $(P_2, \emptyset)$, we know that all the copies of the original automaton corresponding to P_1 have reached an accepting state at the latest at the state $(P_2, \emptyset)$ along that path. Consequently, if for every path of a run there are infinitely many visits to states of the type $(P, \emptyset)$, we know that there is such an infinite strictly increasing sequence as required by ν'-acceptability.

Definition 4.15 Let $A = (Q, q_0, \Delta, F)$ be a Büchi-automaton and $\mathrm{fix}_z A = (Q', q_0', \Delta')$ the intermediate automaton based on A. The Büchi-automaton $\nu z.A$ is defined as $(Q_\nu, q_{\nu 0}, \Delta_\nu, F_\nu)$, where

- $Q_\nu = \{(P, P') \mid P' \subseteq P \subseteq Q\}$,
- $q_{\nu 0} = (\{q_0\}, \{q_0\})$,
- $(P, P') \xrightarrow{Z} ((P_1, P_1'), \ldots, (P_k, P_k')) \in \Delta_\nu$ iff there is a δ such that $P \xrightarrow{Z,\delta} (P_1, \ldots, P_k) \in \Delta'$, and either

- $P' \neq \emptyset$ and for all $i \in [k]$, $P'_i = \{\delta^{\mathrm{to}}(q)_i \mid q \in P'\} \setminus F$, or
- $P' = \emptyset$ and for all $i \in [k]$, $P'_i = P_i \setminus F$

- $F_\nu = \{(P, \emptyset) \mid P \subseteq Q\}$ □

Proposition 4.16 Let ϕ be a μK-formula and A a Büchi-automaton such that $L(\phi) = L(A)$. Then $L(\nu z.\phi) = L(\nu z.A)$. Furthermore, if $\mathrm{free}(\phi) = \mathrm{ref}(A)$, then $\mathrm{free}(\nu z.\phi) = \mathrm{ref}(\nu z.A)$.

Proof: Take any $T \in \mathcal{T}$. By 4.12, $T \in L(\nu z.\phi)$ iff $\mathrm{fix}_z A$ has a ν-accepting run Π on T. By 4.14 this holds iff $\mathrm{fix}_z A$ has a ν'-accepting run Π on T. But it is easy to see that this holds iff $\nu z.A$ has an accepting run on T. Finally, $\mathrm{free}(\nu z.\phi) = \mathrm{free}(\phi) \setminus \{z\} = \mathrm{ref}(A) \setminus \{z\} = \mathrm{ref}(\nu z.A)$. □

Notice that the property of making reference to a variable only before accepting states is *not* generally preserved in the construction of $\nu z.A$. However, for our purposes the following result is enough.

Lemma 4.17 Let ϕ_0, ϕ be μK-formulae such that $\phi_0 \in \Pi_2$ and $\phi_0 \trianglelefteq \nu z.\phi$, let z' be a μ-variable of ϕ_0, and let A be a Büchi-automaton such that $\mathrm{free}(\phi) = \mathrm{ref}(A)$. Then $\nu z.A$ makes reference to z' only before accepting states.

Proof: Clearly either $\mu z' \triangleleft \nu z$ or not. If $\mu z' \triangleleft \nu z$, $z' \notin \mathrm{free}(\phi)$ by 3.4. If then $\mu z' \not\triangleleft \nu z$, clearly $z' \notin \mathrm{free}(\phi)$, as well. As by assumption $\mathrm{free}(\phi) = \mathrm{ref}(A)$, A makes no reference to z', which means that $\nu z.A$ makes no reference to z' at all. □

4.5 Minimal fixpoints

The only difference between the constructions corresponding to the minimal and the maximal fixpoint operators is that in the minimal case we have to disallow infinite regeneration of the fixpoint formula. Intuitively, such regeneration occurs whenever an individual copy of A running as a part of the intermediate automaton $\mathrm{fix}_z A$ takes a transition the label of which refers to the fixpoint variable z. The following definition expresses this intuition of one copy of A requiring the starting of another copy, this another one requiring the starting of a third copy etc.

Definition 4.18 Let $\mathrm{fix}_z A$ be an intermediate automaton based on $A = (Q, q_0, \Delta, F)$, Π a run of $\mathrm{fix}_z A$ on T, and s a string of strings, $s \in ([k]^*)^\infty$. We call s a *dependency sequence* of Π iff $s_1 = \epsilon$ and for all $n < |s|$: $s_n \preceq s_{n+1}$, $q_0 \in \Pi^{\mathrm{fr}}(s_n)$ and $z \in \pi(s_n, q_0)^{\mathrm{lab}}(s_{n+1} - s_n)$. □

Definition 4.19 Let $\mathrm{fix}_z A$ be an intermediate automaton based on A and Π a run of $\mathrm{fix}_z A$ on T. Then Π is *μ-accepting* iff Π is ν-accepting and has no infinite dependency sequences. □

Lemma 4.20 Let ϕ be a μK-formula and A a Büchi-automaton such that $L(\phi) = L(A)$. Then for all $T \in \mathcal{T}$, $T \in L(\mu z.\phi)$ iff $\mathrm{fix}_z A$ has a μ-accepting run on T.

Proof: The proof is fairly similar to that of the maximal fixpoint case. Let $T \in \mathcal{T}$ be an arbitrary tree. By definition and the Knaster-Tarski fixpoint theorem, $T \in L(\mu z.\phi)$ iff there is an ordinal $\alpha \in \mathrm{Ord}$ and a collection of sets $\{S_\beta \subseteq [k]^* \mid \beta \in \mathrm{Ord} \wedge \beta \preceq \alpha\}$ such that

- $\epsilon \in S_\alpha$
- $S_0 = \emptyset$
- for all $\beta \prec \alpha$ and all $t \in S_{\beta+1}$, $T[S_\beta/z]^t \in L(\phi)$
- for all limit ordinals $\lambda \preceq \alpha$, $S_\lambda = \bigcup_{\beta \prec \lambda} S_\beta$.

In the proof we shall show that the latter holds iff $\mathrm{fix}_z A$ has a μ-accepting run Π on T. Without loss of generality, we may assume that the α above is not a limit ordinal.

" => "

Suppose that there exist an $\alpha \in \mathrm{Ord}$ and the sets S_β having the above properties. By definition, for every $\beta \prec \alpha$ and every $t \in S_{\beta+1}$, $T[S_\beta/z]^t \in L(\phi) = L(A)$, which means that A has an accepting run $\pi_{\beta+1,t}$ on $T[S_\beta/z]^t$. As in the proof of 4.12, we are going to construct a μ-accepting run Π on the basis of these runs $\pi_{\beta,t}$. The construction is very similar to that of 4.12, so only the key points are outlined here.

As runs $\pi_{\beta,t}$ are now indexed with an ordinal as well as with the starting point of the run, we need to modify the strategy for building Π to accommodate this feature. This time, if runs $\pi_{\beta,t}$ and $\pi_{\beta',t'}$ coincide in node i of the input, we follow the one indexed with the pair that is the smaller one in the lexicographic ordering of pairs. Remember that this ordering is well-founded.

Analogously to the proof of 4.12 the set of pairs (β, t) such that $\pi_{\beta,t}$ is relevant to $\Pi(i)$ is expressed by a mapping $V : [k]^* \to P(\mathrm{Ord} \times [k]^*)$ defined inductively by: $V(\epsilon) = \{(\alpha, \epsilon)\}$ and for all $i \in [k]^*$ and all $j \in [k]$,

$$V(i \cdot j) = \begin{cases} V'(i \cdot j) \cup \{(\min\{\beta \mid i \cdot j \in S_\beta\}, i \cdot j)\} & \text{if } \exists(\beta,t) \in V'(i \cdot j) : z \in \pi^{lab}_{\beta,t}(i \cdot j - t) \\ V'(i \cdot j) & \text{otherwise} \end{cases}$$

where

$$V'(i \cdot j) = \{(\beta, t) \in V(i) \mid \neg\exists(\beta', t') \in V(i) : (\beta', t') < (\beta, t) \ \wedge \ \pi^{\mathrm{fr}}_{\beta',t'}(i \cdot j - t') = \pi^{\mathrm{fr}}_{\beta,t}(i \cdot j - t)\}$$

To see that $V()$ is well-defined, notice the following two facts. First, it can be inductively shown that if $(\beta, t) \in V(i)$, then β is not a limit ordinal. Secondly, if there is a pair $(\beta, t) \in V'(i \cdot j)$ such that $z \in \pi^{\mathrm{lab}}_{\beta,t}(i \cdot j - t)$, then $i \cdot j \in S_{\beta-1}$, which means that $\{\beta \mid i \cdot j \in S_\beta\}$ is non-empty. What is more, as $\beta - 1 \in \{\beta \mid i \cdot j \in S_\beta\}$, $\min\{\beta \mid i \cdot j \in S_\beta\} < \beta$. This last fact will be needed later in showing the nonexistence of infinite dependency sequences in the run to be constructed.

The sets $P(i)$ are defined as in 4.12, and they possess the same property as there: for each $i \in [k]$ and each $q \in P(i)$ there is a unique pair $(\beta, t) \in V(i)$ such that $\pi^{\mathrm{fr}}_{\beta,t}(i - t) = q$. Let us denote this unique pair (β, t) by $\bar{t}(i, q)$ and its components by $\beta(i, q)$ and $t(i, q)$, for any given i and q.

Defining Π and seeing that it is a ν-succesful run of $\mathrm{fix}_z A$ on T is done as in 4.12, with obvious modifications. What still has to be done is verifying that Π has no infinite dependency sequences.

Take any $i \in [k]^*$ such that $q_0 \in P(i)$, and write $\Pi(i) = (P, Z, \overline{P}, \delta)$. Notice that by the remark above, $\beta(i, q_0) \prec \beta(i, q)$ for all $q \in P(i)$ such that $z \in \delta^{\mathrm{lab}}(q)$. Consequently, $\bar{t}(i, q_0) \prec \bar{t}(i, q)$ for all such q, as well. Remember then that, as in 4.12, by definition of $\bar{t}(,)$ and V, for all $i \in [k]^*$, all $q \in P(i)$ and all $j \in [k]$, $\bar{t}(i \cdot j, \pi^{\mathrm{to}}_{\bar{t}(i,q)}(i - t(i, q))_j) \preceq \bar{t}(i, q)$.

Combining these two facts with the definition of a dependency sequence shows that if s is a dependency sequence of Π, then for all $n < |s|$, $\bar{t}(s_{n+1}, q_0) \prec \bar{t}(s_n, q_0)$. But then the existence of an infinite dependency sequence would contradict the well-foundedness of the lexicographic ordering $\prec$ of pairs of ordinals and strings.

" <= "

Suppose then that we have a μ-accepting run Π of $\mathrm{fix}_z A$. We have to show the existence of the required $\alpha \in \mathrm{Ord}$ and the collection of sets S_β. Define sets S_β for all $\beta \in \mathrm{Ord}$ inductively by

- $S_0 = \emptyset$
- $S_{\beta+1} = \{t \in [k]^* \mid q_0 \in \Pi^{\mathrm{fr}}(t) \text{ and } \forall i \in [k]^* : z \in \pi(t, q_0)^{\mathrm{lab}}(i) \Rightarrow t \cdot i \in S_\beta\}$
- $S_\lambda = \bigcup_{\beta<\lambda} S_\beta$, if λ is a limit ordinal

It is clear that S_β fulfil the required properties. It has still to be shown that there is an $\alpha \in \mathrm{Ord}$ such that $\epsilon \in S_\alpha$.

Take any $t \in [k]^*$ such that $q_0 \in \Pi^{\mathrm{fr}}(t)$. Notice that if for every $i \in [k]^*$ such that $z \in \pi(t, q_0)^{\mathrm{lab}}(i)$ there exists a β_i such that $t \cdot i \in S_{\beta_i}$, then $t \in S_{\beta+1}$, where $\beta = \mathrm{lub}\{\beta_i\}$. Consequently, if $t \notin S_\alpha$ for all $\alpha \in \mathrm{Ord}$, then there is an $i \in [k]^*$ such that $z \in \pi(t, q_0)^{\mathrm{lab}}(i)$ and $t \cdot i \notin S_\alpha$ for all $\alpha \in \mathrm{Ord}$. Denote this i by $i(t)$.

Suppose now, contrary to the claim, that there was no $\alpha \in \mathrm{Ord}$ such that $\epsilon \in S_\alpha$. Define a string s inductively by $s_1 = \epsilon$ and $s_{n+1} = s_n \cdot i(s_n)$ for all $n > 1$, where $i(s_n)$ is as above. By the observations above, s is well-defined and infinite. What is more, it is also a dependency sequence, which contradicts the assumption that Π is μ-accepting. Consequently, there is some $\alpha \in \mathrm{Ord}$ such that $\epsilon \in S_\alpha$. $\square$

In order to construct a Büchi-automaton for $\mu z.\phi$ on the basis of $\mathrm{fix}_z A$, we need a more 'Büchi-like' way of deciding whether a run Π is a μ-accepting one or not. This is where we will take advantage of the fact that for all $\mu z.\phi$ subformulae of any Π_2-formula, the automaton A corresponding to ϕ makes reference to the μ-variable z only before accepting states. Intuitively, the following concept of a μ'-accepting run disallows infinite dependency sequences by requiring first that the run does not contain any transitions, where the initial state q_0 of the automaton would require unwinding the fixpoint z, and secondly, that on each path p of the run one eventually reaches a point $t \prec p$ after which the fixpoint z is not unwound anywhere in the subtree rooted at t.

Definition 4.21 Let $A = (Q, q_0, \Delta, F)$ be a Büchi-automaton, $T \in \mathcal{T}$ and Π a run of $\mathrm{fix}_z A$ on T. We say that Π is *μ'-accepting* iff Π is ν'-accepting and

1 for all $i \in [k]^*$, if $\Pi(i) = (P, Z, \overline{P}, \delta)$ and $q_0 \in P$ then $z \notin \delta^{\mathrm{lab}}(q_0)$

2 for every path $p \in [k]^\omega$ of Π there is a $t \in [k]^*$ such that $t \prec p$ and for all $t' \in [k]^*$ such that $t \preceq t'$, $q_0 \notin \Pi^{\mathrm{fr}}(t')$. $\square$

Lemma 4.22 Let $A = (Q, q_0, \Delta, F)$ be a Büchi-automaton, z a μK-variable such that A makes reference to z only before accepting states, and let Π be a run of $\mathrm{fix}_z A$. Then Π is μ-accepting iff Π is μ'-accepting.

Proof: It is easy to see that any μ'-accepting run Π is also μ-accepting, irrespective of when A makes reference to z.

Suppose then that Π is μ-accepting. We have to show that Π fulfils the requirements of 4.21. As a μ-accepting Π is also ν-accepting, we know by 4.14 that it is ν'-accepting. It is also easy to see that condition 1 of 4.21 is fulfilled: were there an i violating the condition, we could construct an infinite dependancy sequence $s = \ldots i^\omega$ contradicting the fact that Π is μ-accepting.

To see that condition 2 of 4.21 is fulfilled, take any path $p \in [k]^\omega$. Let us describe inductively a tree B, the nodes of which are labelled with strings from $[k]^*$:

- the root of B is labelled with ϵ
- if a node is labelled with i, it has one successor for each distinct $i' \in [k]^*$ such that $i \preceq i' \prec p$ and $z \in \pi^{\mathrm{lab}}(i, q_0)(i' - i)$

Notice that each path through B is a dependency sequence of Π. As Π is μ-accepting, all such dependency sequences are finite, which means that the tree B is of finite height.

In general B can be infinitely branching. However, crucially to the whole proof here, if A makes reference to z only before accepting states, B is finitely branching. To see this, notice first that as Π is ν-accepting, for any $i \prec p$ such that $q_0 \in \Pi^{\mathrm{fr}}(i)$, there is an i'' such that $i \prec i'' \prec p$ and $\pi^{\mathrm{fr}}(i, q_0)(i'' - i)$ is an accepting state. But as A makes reference to z only before accepting states, this implies that there are at most $|i'' - i|$ distinct i' such that $i \preceq i' \prec p$ and $z \in \pi^{\mathrm{lab}}(i, q_0)(i' - i)$, which in turn means that B is finitely branching.

Since B is a finitely branching tree of finite height, and all the labels are finite strings i such that $i \prec p$, there exists a maximal element $t \prec p$ such that t labels some node of B and $i \preceq t$ for all node labels i in B. Since Π is ν'-accepting, there are t_n and t_{n+1} such that $t \preceq t_n \prec t_{n+1} \prec p$, and for every $q \in \Pi^{\mathrm{fr}}(t_n)$, there is a t' such that $t_n \prec t' \prec t_{n+1}$ and $\pi^{\mathrm{fr}}(t_n, q)(t' - t_n) \in F$. This implies also that for every $q \in \Pi^{\mathrm{fr}}(t)$, there is a t' such that $t \prec t' \prec t_{n+1}$ and $\pi^{\mathrm{fr}}(t, q)(t' - t) \in F$. But then for every $q \in \Pi^{\mathrm{fr}}(t_{n+1})$ there is a $q' \in F$ such that $q_0 \longrightarrow^* q' \longrightarrow^* q$. Remembering that A makes reference to z only before accepting states, this implies that there is no transition reachable from any $q \in \Pi^{\mathrm{fr}}(t_{n+1})$ such that it would make reference to z. But then, by the definition of Π, $q_0 \notin \Pi^{\mathrm{fr}}(t')$ for all $t_{n+1} \preceq t'$. Consequently, condition 2 of 4.21 is fulfilled by selecting $t = t_{n+1}$. □

Intuitively the Büchi-automaton $\mu z.A$ consists of two parts. Initially it behaves exactly like the intermediate automaton $\mathrm{fix}_z A$, without any accepting states. At any transition it can make a non-deterministic choice between staying in the initial part or moving on to the second part, which behaves exactly like the $\nu z.A$ automaton, with the exception of disallowing the starting of any new copies of A.

Definition 4.23 Let $A = (Q, q_0, \Delta, F)$ be a Büchi-automaton, $\mathrm{fix}_z A = (Q_f, q_{0f}, \Delta_f)$, $\nu z.A = (Q_\nu, q_{\nu 0}, \Delta_\nu, F_\nu)$, and write $Q'_\nu = \{(P, P') \in Q_\nu \mid q_0 \notin P\}$.

The Büchi-automaton $\mu z.A$ is defined as $(Q_\mu, q_{\mu 0}, \Delta_\mu, F_\mu)$, where

$$
\begin{aligned}
Q_\mu &= Q_f \uplus Q'_\nu \\
q_{\mu 0} &= q_{0f} \\
\Delta_\mu &= \{P \xrightarrow{Z} (P_1, \ldots, P_k) \in Q_f \times \Sigma \times Q_f^k \mid \exists \delta : P \xrightarrow{Z,\delta} (P_1, \ldots, P_k) \in \Delta_f\} \ \cup \\
&\quad\ \{P \xrightarrow{Z} ((P_1, P_1 \setminus F), \ldots, (P_k, P_k \setminus F)) \in Q_f \times \Sigma \times Q'^k_\nu \mid \\
&\qquad\qquad\qquad\qquad \exists \delta : P \xrightarrow{Z,\delta} (P_1, \ldots, P_k) \in \Delta_f\} \ \cup \\
&\quad\ (\Delta_\nu \ \cap \ Q'_\nu \times \Sigma \times Q'^k_\nu) \\
F_\mu &= F_\nu \cap Q'_\nu
\end{aligned}
$$

□

Proposition 4.24 Let ϕ be a μK-formula, z a μK-variable and A a Büchi-automaton such that $L(\phi) = L(A)$ and A makes reference to z only before accepting states.

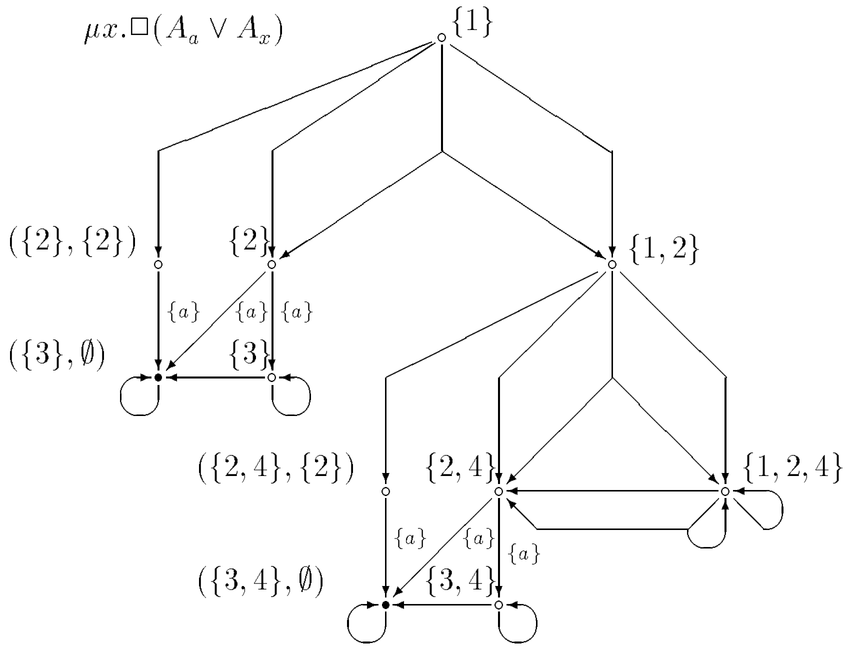


Figure 3: A minimal-fixpoint Büchi-automaton

Then $L(\mu z.\phi) = L(\mu z.A)$. If in addition $\text{free}(\phi) = \text{ref}(A)$, then $\text{free}(\mu z.\phi) = \text{ref}(\mu z.A)$. Furthermore, if z' is a variable such that A makes reference to it only before accepting states, then so does $\mu z.A$.

Proof: From 4.20 and 4.22 as in the proof of 4.16. The validity of the last claim is easy to see from the structure of the automaton $\mu z.A$. $\Box$

Example 4.25 Figure 3 depicts the minimal-fixpoint Büchi-automaton $\mu x.\Box(A_a \vee A_x)$ based on the intermediate automaton $\text{fix}_x\Box(A_a \vee A_x)$ constructed earlier in example 4.10. $\Box$

We can now summarise the mapping from Π_2 to Büchi-automata.

Theorem 4.26 Let ϕ be a μK-formula such that $\phi \in \Pi_2$. Then there exists a Büchi-automaton A_ϕ such that $L(\phi) = L(A_\phi)$.

Proof: By induction on the structure of ϕ: the base case is shown in 4.3 and the induction steps in 4.6, 4.16, 4.17 and 4.24. $\Box$

As a corollary, we can see that in the case of modal mu-calculus the alternation depth hierarchy does not collapse at level Π_2 as in the case of the linear mu-calculus.

Proposition 4.27 There is a $\phi \in \Pi_3$ such that $L(\phi) \neq L(\phi')$ for all $\phi' \in \Pi_2$.

Proof: The property *on every path, a is false only finitely many times* is not Büchi-recognisable [Tho89], i.e. for the corresponding language L, $L \neq L(\phi)$ for all $\phi \in \Pi_2$. But $L = L(\phi)$, where $\phi = \mu x.\nu z.(\Box z \wedge (a \vee \Box x)) \in \Sigma_2 \subseteq \Pi_3$. $\Box$

5 From automata to formulae

In this section we show the other half of the result by providing a mapping from Büchi-automata to modal mu-calculus formulae.

However, let us first notice an essential difference between Büchi-automata and the modal mu-calculus: distinguishing subtrees. In a Büchi-automaton we can distinguish between different subtrees of a node $T(i)$ of an input tree, and to require e.g. that $T^{i \cdot 1}$ satisfies a certain property and that the $T^{i \cdot 2}$ satisfies some other property. However, in modal mu-calculus we can only express the requirements *all subtrees satisfy a property* and *some subtree satisfies a property.*

This means that there is no hope of mapping Büchi-automata faithfully to modal mu-calculus. Instead, this result is shown 'modulo distinguishing subtrees', only up to the extent that the Büchi-automata concerned do not take advantage of their greater expressive power. To this purpose, we define a more relaxed notion of what it means to run an automaton on a tree, called a *pseudo-run*, and use this to define the language accepted by an automaton.

Definition 5.1 Let $A = (Q, q_0, \Delta, F)$ be a Büchi-automaton and $T \in \mathcal{T}$ a tree. A *pseudo-run of A on T* is an infinite tree π labelled with $(\Delta \times [k]^*)$ fulfilling:

- $\pi(\epsilon) = ((q_0, Z, \bar{q}), \epsilon)$ for some Z, $\bar{q}$.
- for all $i \in N^*$, if $\pi(i) = ((q, Z, \bar{q}), i')$, then $Z \subseteq T(i')$
- for all $i \in N^*$, if $\pi(i) = ((q, Z, \bar{q}), i')$, then

$$\begin{array}{rcl} \{\bar{q}_1, \ldots, \bar{q}_k\} & = & \{q' \mid \exists j \in N : \pi(i \cdot j) = ((q', Z', \bar{q}'), j') \text{ for some } Z', \bar{q}', j'\} \\ \{i' \cdot 1, \ldots, i' \cdot k\} & = & \{j' \mid \exists j \in N : \pi(i \cdot j) = ((q', Z', \bar{q}'), j') \text{ for some } q', Z', \bar{q}'\} \end{array}$$

The mappings π^{fr}, π^{lab} and π^{to}, and the concept of an accepting pseudo-run are defined as for ordinary runs.

Furthermore, define $L_{PS}(A) = \{T \in \mathcal{T} \mid A \text{ has an accepting pseudo-run } \pi \text{ on } T\}$. □

The difference between a normal run and a pseudo-run has been illustrated in figure 4. In a normal run, if an automaton is in state q, is currently working on node $T(i)$ of the input tree, and takes a transition $q \xrightarrow{Z} (q_1, \ldots, q_k)$, it runs down each subtree $T^{i \cdot j}$ from node q_j. However, in a pseudo-run the node from which the automaton runs down a particular subtree is less rigidly defined: e.g. the automaton might run down $T^{i \cdot 1}$ from node q_k and down $T^{i \cdot k}$ from node q_1. The only requirement is that for each q_j, the automaton must run down some subtree $T^{i \cdot j'}$ from it, and conversely, that for each subtree $T^{i \cdot j}$, the automaton must run down it from some $q_{j'}$.

We can now formulate the mapping from Büchi-automata to modal mu-calculus formulae.

Definition 5.2 Let $A = (Q, q_0, \Delta, F)$ be a Büchi-automaton. Fix a set $\{x_q \mid q \in Q\}$ of variables such that A does not contain any references to them.

Let us define for each $q \xrightarrow{Z} \bar{q} \in \Delta$ and for each $q \in Q$:

$$\begin{array}{rcl} \psi(q, Z, \bar{q}) & = & \bigwedge_{z \in Z} z \;\wedge\; \Box(\bigvee_{i \in [k]} x_{\bar{q}_i}) \;\wedge\; \bigwedge_{i \in [k]} \Diamond x_{\bar{q}_i} \\ \psi_q & = & \bigvee \{\psi(q, Z, \bar{q}) \mid q \xrightarrow{Z} \bar{q} \in \Delta\} \end{array}$$

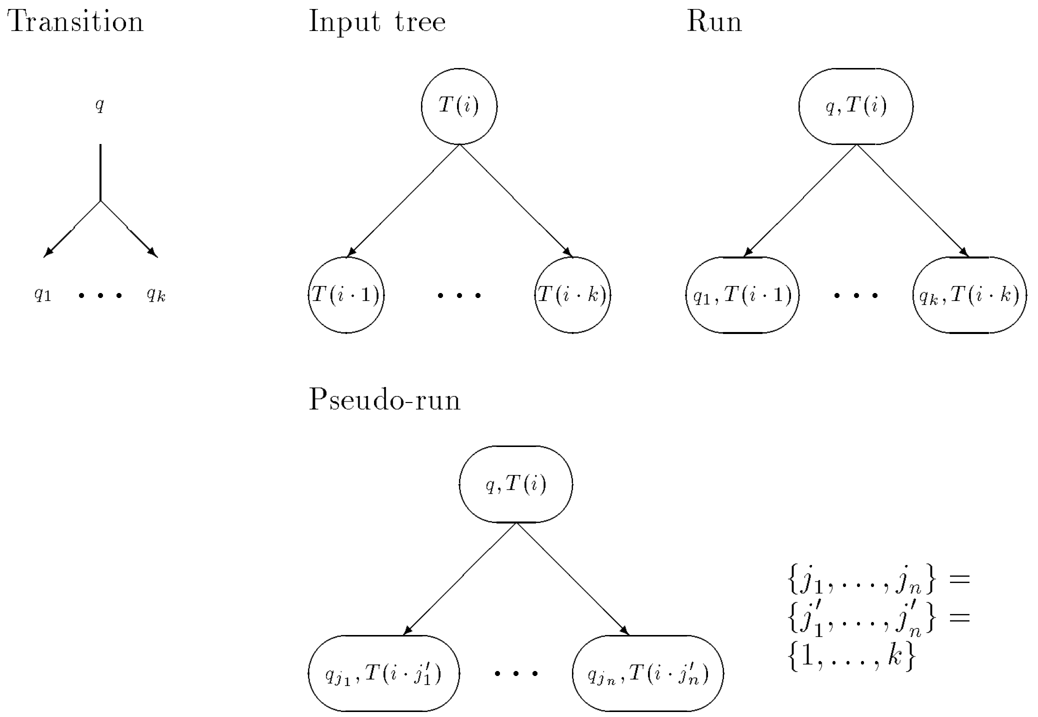


Figure 4: A run vs. a pseudo-run

For all $q \in F \cup \{q_0\}$, let us define χ_q by:

$$
\begin{aligned}
\chi_q &= \psi_q[\chi_{q_1}(\{q_1\})/x_{q_1}, \ldots, \chi_{q_n}(\{q_n\})/x_{q_n}] \quad \text{where } \{q_1, \ldots, q_n\} = Q \setminus F \quad \text{and} \\
\chi_q(P) &= \mu x_q . \psi_q[\chi_{q_1}(P \cup \{q_1\})/x_{q_1}, \ldots, \chi_{q_n}(P \cup \{q_n\})/x_{q_n}] \quad \text{where } \{q_1, \ldots, q_n\} = (Q \setminus F) \setminus P
\end{aligned}
$$

Let us define the formula ϕ_A corresponding to A by:

$$
\begin{aligned}
\phi_A &= \chi_{q_0}[\phi_{q_1}(\{q_1\})/x_{q_1}, \ldots, \phi_{q_n}(\{q_n\})/x_{q_n}] \quad \text{where } \{q_1, \ldots, q_n\} = F \quad \text{and} \\
\phi_q(E) &= \nu x_q . \chi_q[\phi_{q_1}(E \cup \{q_1\})/x_{q_1}, \ldots, \phi_{q_n}(E \cup \{q_n\})/x_{q_n}] \quad \text{where } \{q_1, \ldots, q_n\} = F \setminus E
\end{aligned}
$$

□

Example 5.3 Figure 5 shows a Büchi-automaton for 2-branching trees and some of the formulae used in mapping this automaton to modal mu-calculus. To keep the formulae readable, we have introduced some abbreviated notation. □

Lemma 5.4 $L_{PS}(A) = L(\phi_A)$ for all Büchi-automata A.

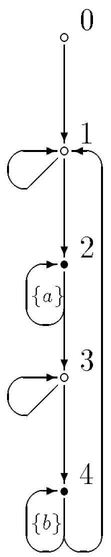
Proof: Take any Büchi-automaton A and any tree $T \in \mathcal{T}$. We want to show that $T \in L_{PS}(A)$ iff $T \in L(\phi_A)$.

From the structure of ϕ_A it can be seen that $T \in L(\phi_A)$ iff there is a tree T' obtained by labelling some of the nodes of T with the maximal fixpoint variables such that

1 $T'(i) \setminus \{x_q \mid q \in F\} = T(i)$ for all $i \in [k]^*$,

2 $T' \in L(\chi_{q_0})$, and

3 if $x_q \in T'(i)$, then $T'^i \in L(\chi_q)$, for all $i \in [k]^*$ and $q \in F$.



Define abbreviations: $\boxslash\{\phi_1, \ldots, \phi_n\} = \Box(\bigvee \phi_i) \wedge \bigwedge \Diamond\phi_i$

$\boxslash\phi = \boxslash\{\phi\}$

Formulae are:

$$
\begin{aligned}
\psi_0 &= \boxslash x_1 \\
\psi_1 &= \boxslash x_1 \vee \boxslash x_2 \\
\psi_2 &= a \wedge \boxslash\{x_2, x_3\} \\
\psi_3 &= \boxslash x_3 \vee \boxslash x_4 \\
\psi_4 &= b \wedge \boxslash\{x_1, x_4\}
\end{aligned}
$$

$$
\begin{aligned}
\chi_0 &= \boxslash\mu x_1.(\boxslash x_1 \vee \boxslash x_2) \\
\chi_2 &= a \wedge \boxslash\{x_2, \mu x_3.(\boxslash x_3 \vee \boxslash x_4)\} \\
\chi_4 &= b \wedge \boxslash\{\mu x_1.(\boxslash x_1 \vee \boxslash x_2), x_4\}
\end{aligned}
$$

$$
\begin{aligned}
\phi_A = \; & \boxslash\mu x_1.(\boxslash x_1 \vee \\
& \quad \boxslash\nu x_2.(a \wedge \\
& \quad\quad \boxslash\{x_2, \mu x_3.(\boxslash x_3 \vee \\
& \quad\quad\quad \boxslash\nu x_4.(b \wedge \\
& \quad\quad\quad\quad \boxslash\{\mu x_1.(\boxslash x_1 \vee \boxslash x_2), x_4\}))\}))
\end{aligned}
$$

Figure 5: Mapping a Büchi-automaton to a formula

Let us show that this holds iff $T \in L_{PS}(A)$, i.e. iff A has an accepting pseudo-run π on T.

" => "

Suppose that there is a tree T' fulfilling the conditions 1–3 above. From the condition 2, we know that $T' \in L(\chi_{q_0})$. Notice that all the fixpoints in χ_{q_0} are minimal, i.e. each of them can be unwound only finitely many times. Looking at the structure of χ_{q_0}, this means that on each path of T' there must be a point so that no minimal fixpoints of χ_{q_0} need to be unwound anywhere below that point in T' in order for χ_{q_0} to hold of the root of T'. What is more, the labels of T' below such a point are irrelevant as far as the validity of χ_{q_0} of the root of T' is concerned. To be slightly more formal, this means that there is a finite prefix-closed set $S \subseteq [k]^*$ such that $T_S \in L(\chi_{q_0})$, where $T_S(i) = T'(i)$ if $i \in S$ and $T_S(i) = \emptyset$ otherwise. Intuitively, S specifies a finite fragment of T' such that χ_{q_0} holding of the root of T' can be verified without need to make reference to nodes outside this fragment.

From the structure of χ_{q_0} we can see that $T_S \in L(\chi_{q_0})$ iff there is a tree T'_S obtained from T_S by labelling some of its nodes with the minimal fixpoint variables such that

- $T'_S(i) \setminus \{x_q \mid q \in Q \setminus F\} = T_S(i)$ for all $i \in S$, and $T'_S(i) = \emptyset$ for all $i \notin S$,
- $T'_S \in L(\psi_{q_0})$, and
- if $x_q \in T'_S(i)$, then $T'^{i}_S \in L(\psi_q)$, for all $i \in S$ and $q \in Q \setminus F$.

From this labelling of T'_S it is easy to construct a finite fragment of a pseudo-run, i.e. a finite tree starting from (q_0, ϵ) and leading to leaves of the type (q, i), where $q \in F$ and $x_q \in T'_S(i) = T_S(i) = T'(i)$.

By condition 3 above, $x_q \in T'(i)$ implies that $T'^{i} \in L(\chi_q)$. This means that for each leave of the fragment labelled with (q, i), we can repeat the construction of the previous paragraphs to yield a finite fragment of a pseudo-run starting from (q, i) and leading to leaves of the type (q', i') where $q' \in F$ and $x_{q'} \in T'(i')$. From these leaves we can repeat the construction again etc. As a limit of these finite fragments we obtain a pseudo-run π of A on T. The pseudo-run π is accepting, since all the leaves of the finite fragments correspond to accepting states, which means that on each path some accepting state must occur infinitely often.

" <= "

Suppose that A has an accepting pseudo-run π on T. Define T' by: $T'(i) = T(i) \cup \{x_q \mid \exists i' : \pi(i') = ((q, Z, \bar{q}), i)$ for some $Z, \bar{q}\}$ for all $i \in [k]^*$. T' clearly fulfils condition 1 above. It is also easy to see from the structure of π that T' fulfils both 2 and 3. $\square$

Lemma 5.5 For all Büchi-automata A, $\phi_A \in \Pi_2$.

Proof: Take any μ-variable z' and any ν-variable z of ϕ_A such that $\phi_A \trianglelefteq \mu z' \triangleleft \nu z$. It is easy to see from the definition of ϕ_A that z' does not occur free in νz. By proposition 3.4, this implies $\phi_A \in \Pi_2$. $\square$

Theorem 5.6 If A is a Büchi-automaton, then there is a $\phi \in \Pi_2$ such that $L(\phi) = L_{PS}(A)$.

Proof: Immediate from 5.5 and 5.4. $\square$

Finally, let us characterise the class of Büchi-automata that correspond exactly to modal mu-calculus formulae in Π_2.

Definition 5.7 Let A be a Büchi-automaton, We say that A *does not distinguish subtrees* iff $L(A) = L_{PS}(A)$. $\square$

Lemma 5.8 For all $\phi \in \Pi_2$, A_ϕ does not distinguish subtrees.

Proof: Straightforward induction on the structure of ϕ. $\square$

Corollary 5.9 If $\phi \in \Pi_2$, then there exists a Büchi-automaton A such that A does not distinguish subtrees and $L(A) = L(\phi)$. Conversely, if A is a Büchi-automaton which does not distinguish subtrees, then there is a $\phi \in \Pi_2$ such that $L(\phi) = L(A)$.

Proof: Immediate from 4.26, 5.8 and 5.6. $\square$

Returning to the issue of the different ways of defining alternation depth, notice that for the formula ϕ_A of 5.3, $\phi_A \in \Pi_2$ but $\phi_A \notin \Pi_2^{EL}$. What is more, it is easy to see from the definition of ϕ_A that in general there is no fixed n such that $\phi_A \in \Pi_n^{EL}$ for all Büchi-automata A.

Acknowledgements

I would like to thank Colin Stirling and Mads Dam for many useful discussions. This work has been funded by the Academy of Finland, the British Council, the CVCP Overseas Research Students Awards Scheme and the Finnish Cultural Foundation.

References

[AN90] Arnold, A. & Niwinski, D.: Fixed Point Characterisation of Büchi Automata on Infinite Trees, in *Journal of Information Processing and Cybernetics (EIK)*, vol. 8-9, 1990, pp. 451-459

[Bra91] Bradfield, J.: *Verifying Temporal Properties of Systems with Applications to Petri Nets*, PhD thesis CST-83-91, University of Edinburgh, Department of Computer Science, 1991

[Dam92] Dam, M.: *Fixpoints of Büchi automata*, Report ECS-LFCS-92-224, University of Edinburgh, Laboratory for Foundations of Computer Science, 1992

[EC80] Emerson, E. A. & Clarke, E. M.: Characterising correctness properties of parallel programs using fixpoints, in *Proceedings of the 7th International Conference on Automata, Languages and Programming*, LNCS vol. 85, Springer-Verlag, 1980, pp. 169-181

[EJ91] Emerson, E. A. & Jutla, C. S.: Tree automata, mu-calculus, and determinacy, in *Proceedings of the 32nd Symposium on Foundations of Computer Science*, 1991, pp. 368-377

[EL86] Emerson, E. A. & Lei, C.L.: Efficient model-checking in fragments of the propositional mu-calculus, in *Proceedings of the First IEEE Symposium on Logic in Computer Science*, 1986, pp. 267-278

[Koz83] Kozen, D.: Results on the propositional μ-calculus, in *Theoretical Computer Science*, vol. 27, 1983, pp. 333-354

[Niw86] Niwinski, D.: On fixed point clones, in *Proceedings of the 13th ICALP*, LNCS vol. 226, Springer-Verlag, 1986, pp. 464-473

[Niw88] Niwinski, D.: Fixed points vs. infinite generation, in *Proceedings of the 3rd IEEE Symposium on Logic in Computer Science* 1988, pp. 402-409

[Sti92] Stirling, C.: Modal and temporal Logics, in Abramsky, S. & Gabbay, D. & Maibaum, T. (eds.): *Handbook of Logic in Computer Science*, Oxford University Press, 1992, pp. 477-563

[Tak86] Takahashi, M.: The greatest fixed-points and rational omega-tree languages, in *Theoretical Computer Science*, vol. 44, 1986, pp. 259-274

[Tho89] Thomas, W.: Automata on infinite objects, in van Leeuwen, J. (ed.): *Handbook of Theoretical Computer Science*, vol. 2, North-Holland, 1989