Fixed points of Büchi automata

Mads Dam*

Department of Computer Science, University of Edinburgh, U.K.

Abstract. We give a new and direct proof of the equivalence between the linear time μ-calculus νTL and Büchi automata. Constructions on automata are given which compute their least and greatest fixed points. Together with other well-known constructions corresponding to the remaining νTL connectives the result is a representation of νTL as Büchi automata which in contrast to previously known constructions is both elementary and compositional. Applications to the problem of completely axiomatising νTL are discussed.

1 Introduction

The relation between automata as devices for recognising behaviours, and fixed points, or equations, as means of characterising them is an important recurring theme in the theory of computation. The ω-regular languages provides an example of particular interest in concurrency theory. They are characterised on the one hand by formulas in the linear time μ-calculus. This logic, known as νTL, augments linear time logic by least and greatest fixed points of formally monotone contexts. The ω-regular languages are also exactly the languages recognised by Büchi automata, finite automata applied to words of infinite length. Both νTL and Büchi automata have had considerable attention as formalisms for specifying and verifying concurrent programs (c.f. [?, ?, ?, ?, ?, ?]).

We suggest examining the connection between νTL and Büchi automata further. Büchi automata at present lacks a structural theory which is usable in practice, for instance for machine implementation or to support equational reasoning. The equivalence with S1S, the monadic second-order theory of successor, is nonelementary [?] and thus offers little concrete assistance. The linear time μ-calculus is potentially much more valuable for this purpose. Fixed points, on the other hand, can be very troublesome in practical use. Already at the second level of alternation formulas can become highly unintelligible. Automata can prove useful aids for visualising fixed point properties.

The value of a compositional, or syntax-directed approach in such an enterprise is well documented. Indeed Büchi's original work on the decidability of S1S, the monadic second-order theory of one successor [?], gave a compositional representation of S1S formulas as automata, representing second-order quantification, in particular, by projection. The present paper can be viewed

* Research supported by SERC grant GR/F 32219. Current address: Swedish Institute of Computer Science, Box 1263, S-164 28 Kista, Sweden. E-mail: mfd@se.sics.

as an adaptation of Büchi's work to νTL, by providing representations for the fixed point quantifiers. That is, given an automaton recognising the language expressed by the νTL-formula ϕ where ϕ is formally monotone in the variable X, we produce automata recognising the least and greatest fixed points, $\mu X.\phi$ and $\nu X.\phi$ respectively, of the operator $\lambda X.\phi$. Of course only one fixed point construction, for instance for greatest fixed points, is needed due to the equivalence $\mu X.\phi \equiv \neg\nu X.\neg\phi[\neg X/X]$. However, the construction for least fixed points generalises the construction for greatest fixed points in a natural way, and by using it the need for explicit complementation of Büchi automata can be dispensed with.

Existing proofs that formulas in νTL define ω-regular languages give constructions of Büchi automata that are either nonelementary because S1S is used as an intermediate step, or noncompositional. The latter is the case, in particular, for the automata-theoretic techniques of e.g. [**?**, **?**]. Their approach is global rather than compositional: The automaton for a formula ϕ is built as the intersection of an automaton that checks local model conditions with the complement of an automaton that checks for non-well-foundedness of a certain regeneration relation.

The paper is organised as follows: In section 2 we introduce νTL, and in section 3 we introduce Büchi automata and show how they can be represented in νTL. This representation is instructive in showing results that do not appear to be widely known, such as the collapse of the fixed point alternation hierarchy (on level $\nu\mu$), and the expressive equivalence of the aconjunctive fragment of νTL with the full language (see [**?**] for a definition of aconjunctivity). The fixed point constructions first builds an intermediate automaton with nonstandard acceptance conditions. This construction is described in section 4, and then in sections 5 and 6 the constructions for greatest and least fixed points are given. Finally, in section 7, we discuss the application of our construction to the problem of completely axiomatising νTL. This is of particular interest since automata-based techniques, despite their success in temporal logic in general, have not so far proved very useful where axiomatisations are concerned. The axiomatisation we have in mind is based on Kozen's axiomatisation of the modal μ-calculus [**?**]. Using our construction Büchi automata can be viewed as normal forms for νTL, suggesting a strategy for proving completeness whereby each formula is proved equivalent to its normal form using only the axioms and rules of inference provided. We have so far used this strategy successfully to prove completeness for the aconjunctive fragment. Our approach is related to Siefke's completeness result for S1S [**?**] and to Kozen's recent completeness result for the algebra of regular events [**?**].

2 The Linear Time μ-calculus

Formulas ϕ, ψ, γ of the linear-time μ-calculus νTL are built from propositional variables X, Y, Z, boolean connectives $\neg$ and $\wedge$, the nexttime operator O, and the least fixed point operator $\mu X.\phi$, subject to the formal monotonicity condition that all free occurrences of X lie in the scope of an even number of negations. Other connectives are derived in the usual way, and in particular greatest fixed points are derived by $\nu X.\phi \stackrel{\Delta}{=} \neg\mu X.\neg\phi[\neg X/X]$. Intuitively, least fixed points are used for eventuality properties, and greatest fixed points for invariants.

Fix a finite set Σ of propositional variables. A *model* $\mathcal{M}$ assigns to each variable $X \in \Sigma$ a subset $\mathcal{M}(X) \subseteq \omega$. Models are extended to arbitrary formulas with free variables in Σ in the following way:

$$\begin{aligned}
\mathcal{M}(\neg\phi) &= \overline{\mathcal{M}(\phi)} \\
\mathcal{M}(\phi \wedge \psi) &= \mathcal{M}(\phi) \cap \mathcal{M}(\psi) \\
\mathcal{M}(\mathrm{O}\phi) &= \{i \mid i+1 \in \mathcal{M}(\phi)\} \\
\mathcal{M}(\mu X.\phi) &= \cap\{A \subseteq \omega \mid \mathcal{M}[X \mapsto A](\phi) \subseteq A\}
\end{aligned}$$

Here $\mathcal{M}[X \mapsto A]$ is the obvious update of $\mathcal{M}$. There is a bijective correspondence between models and ω-words α over the alphabet 2^{Σ}. The model $\mathcal{M}$ determines the ω-word $\alpha_{\mathcal{M}} : i \mapsto \{X \mid i \in \mathcal{M}(X)\}$, and the *language defined by* ϕ is

$$L(\phi) = \{\alpha_{\mathcal{M}} \mid 0 \in \mathcal{M}(\phi)\}. \tag{1}$$

Operations on ω-words α include the *n'th suffix*, α^n, and, where $n \leq m$, the *n, m-segment*, $\alpha(n,m) = \alpha(n) \cdots \alpha(m)$.

3 Büchi Automata

Automata provide an alternative way of defining ω-languages. We use a slightly modified account of Büchi automata, closely related to Alpern and Schneider's use of transition predicates [?]. Fix a finite set Σ of propositional variables. An *atom* over Σ is a pair $a = (a^+, a^-)$ where a^+ and a^- are subsets of Σ. Intuitively, a transition labelled a is enabled when all members of a^+ are true and all members of a^- false. The set of all atoms over Σ is denoted by $\mathrm{At}(\Sigma)$.

A Büchi-automaton (over Σ) is an NFA $\mathcal{A} = (Q, q_0, \{\xrightarrow{a}\}_{a \in \mathrm{At}(\Sigma)}, F)$ where Q is the finite set of states, $q_0 \in Q$ is the initial state, $\xrightarrow{a} \subseteq Q \times Q$ is the transition relation for each $a \in \mathrm{At}(\Sigma)$, and $F \subseteq Q$ is the set of accepting states. We sometimes write $\mathcal{A}(q_0)$ instead of just $\mathcal{A}$ to emphasize the initial state. Let an ω-word α over alphabet 2^{Σ} be given. An (infinite) *run* of $\mathcal{A}$ on α is an ω-word Π over Q s.t. $\Pi(0) = q_0$ and for all $i \geq 0$ there is an atom $a \in \mathrm{At}(\Sigma)$ s.t. $\Pi(i) \xrightarrow{a} \Pi(i+1)$, $a^+ \subseteq \alpha(i)$ and $a^- \cap \alpha(i) = \emptyset$. Finite runs are defined similarly. An infinite run is *successful* if some accepting state in F occurs infinitely often in it, and $\mathcal{A}$ *accepts* α if a successful run of $\mathcal{A}$ on α exists. The language *recognised by* $\mathcal{A}$ is $L(\mathcal{A}) = \{\alpha \mid \mathcal{A} \text{ accepts } \alpha\}$.

Example 1. In all examples here and below formulas are positive in their free propositional variables. The negative component of atoms can consequently be omitted.

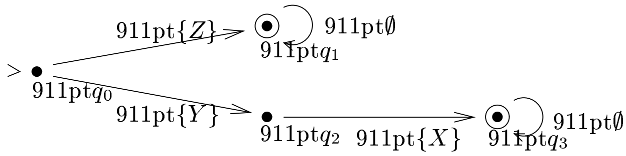

Fig. 1. Büchi automaton $\mathcal{A}_1$ for $Z \vee (Y \wedge \mathrm{O}X)$

1. The automaton $\mathcal{A}_1$ of fig. 1 recognises the language defined by the νTL formula $Z \vee (Y \wedge \mathrm{O}X)$.
2. The automaton $\mathcal{A}_2$ of fig. 2 recognises $\mathrm{O}((\mathrm{O}(\mu Y.X \vee \mathrm{O}Y)) \wedge Z)$, equivalent to the PTL formula $\mathrm{O}((\mathrm{O}FX) \wedge Z)$.

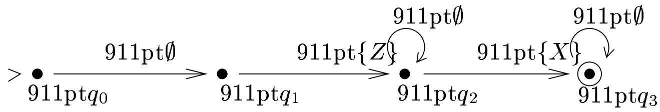

Fig. 2. Büchi automaton $\mathcal{A}_2$ for $\mathrm{O}((\mathrm{O}(\mu Y.X \vee \mathrm{O}Y)) \wedge Z)$

The Büchi automaton $\mathcal{A}$ can be represented as a νTL formula $\mathrm{fm}(\mathcal{A})$ in the following way: Let $F_{\mathcal{A}} = \{q_1, \ldots, q_n\}$ and for each $1 \leq i \leq n$, let $\mathcal{A}_i$ be $\mathcal{A}$ with F replaced by the singleton $\{q_i\}$. Then $L(\mathcal{A}) = \bigcup_{1\leq i}^{n} L(\mathcal{A}_i)$ so we can let $\mathrm{fm}(\mathcal{A}) \stackrel{\Delta}{=} \bigvee_{1\leq i}^{n} \mathrm{fm}(\mathcal{A}_i)$. To represent the $\mathcal{A}_i$, states are represented as fixed point formulas, the unique accepting state as a ν-formula and all other states as μ-formulas. Thus the representation, $\mathrm{fm}_\rho(q)$, of q is really relative to an environment $\rho \subseteq Q$ keeping track of earlier encountered states, and then $\mathrm{fm}(\mathcal{A}) = \mathrm{fm}_\emptyset(q_0)$. For each state q let X_q be a distinguished propositional variable. Atoms are dealt with by defining

$$a.\phi \stackrel{\Delta}{=} \mathrm{O}\phi \wedge \bigwedge a^+ \wedge \bigwedge\{\neg X \mid X \in a^-\} \tag{2}$$

The representation is now defined as follows:

$$\mathrm{fm}_\rho(q) = \begin{cases} X_q & \text{if } q \in \rho \\ \mu X_q. \bigvee\{a.\mathrm{fm}_{\rho\cup\{q\}}(q') \mid q \stackrel{a}{\to} q'\} & \text{if } q \notin \rho \text{ and } q \neq q_i \\ \nu X_q. \bigvee\{a.\mathrm{fm}_{\{q\}}(q') \mid q \stackrel{a}{\to} q'\} & \text{otherwise} \end{cases} \tag{3}$$

We can assume that every state q has a successor, i.e. that there are a and q' such that $q \stackrel{a}{\to} q'$ so that only nonempty disjunctions in (3) are needed. This assumption applies throughout the rest of the paper. The representation is closely related to the translation of ECTL* into the modal μ-calculus of Dam [?] and can be proved correct in the same way.

Theorem 1. *For each Büchi automaton $\mathcal{A}$, $L(\mathcal{A}) = L(\mathrm{fm}(\mathcal{A}))$.* □

4 Intermediate Automata

To derive equivalent Büchi automata from νTL-formulas we give for each connective of νTL a corresponding construction on automata. Each formula can be put in *positive form*, generated by

$$\phi ::= X \mid \neg X \mid \phi_1 \vee \phi_2 \mid \phi_1 \wedge \phi_2 \mid \mathrm{O}\phi \mid \nu X.\phi \mid \mu X.\phi$$

so we only need consider negation applied to propositional variables. It is easy to produce automata $\mathrm{aut}(X)$ and $\mathrm{aut}(\neg X)$ respectively recognising $L(X)$ and $L(\neg X)$, and to produce an automaton $\mathrm{O}\mathcal{A}$ recognising $L(\mathrm{O}\phi)$ when $\mathcal{A}$ recognises $L(\phi)$. Corresponding to the $\vee$ is the sum operation $\mathcal{A}_1 + \mathcal{A}_2$ which adjoins a new initial state to the disjoint sum of the statesets of $\mathcal{A}_1$ and $\mathcal{A}_2$. Corresponding to the $\wedge$ is a product automaton $\mathcal{A}_1 \times \mathcal{A}_2$ which accepts when first an accepting state of $\mathcal{A}_1$ and then of $\mathcal{A}_2$ is encountered (c.f. [?]).

Completing this procedure it thus remains to produce automata $\nu X.\mathcal{A}$ and $\mu X.\mathcal{A}$ for $\nu X.\phi$ and $\mu X.\phi$ respectively when $\mathcal{A} = (Q, q_0, \{\stackrel{a}{\rightarrow}\}_{a \in \mathrm{At}(\Sigma)}, F)$ recognises $L(\phi)$. We assume the following two properties of $\mathcal{A}$:

1. Whenever $q \stackrel{a}{\rightarrow} q'$ then $X \notin a^-$.
2. Whenever $q_0 \stackrel{a}{\rightarrow} q$ then $X \notin a^+$.

The first property reflects the formal monotonicity requirement of X in ϕ and is validated by the inductive construction of $\mathcal{A}$ from ϕ. The second property ensures that occurrences of X in $\mathrm{fm}(\mathcal{A})$ are *guarded*, i.e. occurs only within the scope of the nexttime operator O. It is a straightforward matter to modify an automaton $\mathcal{A}$ such that property 2 is satisfied without affecting the languages recognised by the fixed point automata (c.f. [?]).

The key problem in deriving the fixed point automata is to handle transitions $q \stackrel{a}{\rightarrow} q'$ of $\mathcal{A}$ that involve reference to the recursion variable, i.e. such that $X \in a^+$. In this situation, as part of a fixed point automaton, q gives rise not only to q', but also to a state q'' for which $q_0 \stackrel{a'}{\rightarrow} q''$ for some appropriate a'. We use a subset construction to handle this conjunctive branching of the transition relation. Given $\mathcal{A}$ the procedure detailed below gives an automaton $\mathcal{A}'$, called an *intermediate automaton*. The states of $\mathcal{A}'$ are subsets of Q, and the initial state is the singleton $\{q_0\}$. For the transition relation there are two cases according to whether a reference to the recursion variable is needed or not:

1. (X not referenced). Let $a^+ = a_1^+ \cup \cdots \cup a_m^+$, $a^- = a_1^- \cup \cdots \cup a_m^-$, and $X \notin a^+$. If $q_1 \stackrel{a_1}{\rightarrow} q_1', \ldots, q_m \stackrel{a_m}{\rightarrow} q_m'$ then $\{q_1, \ldots, q_m\} \xrightarrow{(a^+, a^-)} \{q_1', \ldots, q_m'\}$.
2. (X referenced). Let $a^+ = a_1^+ \cup \cdots \cup a_{m+n+1}^+$, $a^- = a_1^- \cup \cdots \cup a_{m+n+1}^-$, $n \geq 1$, and $X \notin a^+$. Suppose
   (a) $q_1 \stackrel{a_1}{\rightarrow} q_1', \ldots, q_m \stackrel{a_m}{\rightarrow} q_m'$,
   (b) $q_{m+1} \xrightarrow{(a_{m+1}^+ \cup \{X\}, a_{m+1}^-)} q_{m+1}', \ldots, q_{m+n} \xrightarrow{(a_{m+n}^+ \cup \{X\}, a_{m+n}^-)} q_{m+n}'$, and
   (c) $q_0 \xrightarrow{a_{m+n+1}} q_{m+n+1}'$.

Then $\{q_1, \ldots, q_{m+n}\} \xrightarrow{(a^+,a^-)} \{q'_1, \ldots, q'_{m+n+1}\}$.

Note that $\rightarrow$ is used for the transition relation in both $\mathcal{A}$ and $\mathcal{A}'$. Ambiguities caused by this are resolved by context.

It remains to equip $\mathcal{A}'$ with appropriate acceptance conditions. For this purpose an analysis of the way individual states in $\mathcal{A}$ are generated along runs of $\mathcal{A}'$ is required. Let S range over subsets of Q and assume that $S \xrightarrow{a} S'$. The successor relation $\Rightarrow \subseteq S \times S'$ is determined in the following way: In case 1 we let $q_i \Rightarrow q'_j$ only if $i = j$, and q'_j is then the *direct successor* of q_i. In case 2 we let $q_i \Rightarrow q'_j$ only if either $i = j$ in which case q'_j is the *direct successor* of q_i, or $m < i \leq m + n$ and $j = m + n + 1$, in which case q'_j is the *indirect successor* of q_i. Consider a run Π through $\mathcal{A}'$ and any word π over states of $\mathcal{A}$ with the property that $\pi(i)$ is defined and a member of $\Pi(i)$ whenever the latter is defined, and whenever $\Pi(i+1)$ is defined then $\pi(i) \Rightarrow \pi(i+1)$ relative to the transition $\Pi(i) \xrightarrow{a} \Pi(i+1)$. We call π a *trail* through Π, written as $\pi \in \Pi$. If $\pi(i+1)$ is the direct successor of $\pi(i)$ for all i for which $\pi(i+1)$ is defined then π is a *direct trail*. Note that each run Π and $q \in \Pi(0)$ determines a unique direct trail $\pi \in \Pi$, the direct trail from q, for which $\pi(0) = q$.

We can now define the acceptance conditions: A trail π is *successful* if π^i is a direct trail for some i and $\pi(j) \in F$ for infinitely many j. An infinite run Π through $\mathcal{A}'$ is *μ-successful* if all $\pi \in \Pi$ are successful, and it is *ν-successful* if all $\pi \in \Pi$ for which π^i is a direct trail for some i are successful.

Theorem 2. *The following statements are equivalent:*

1. $0 \in \mathcal{M}(\nu X.\mathrm{fm}(\mathcal{A}))$.
2. *There is a ν-successful run Π through $\mathcal{A}'$ on $\alpha_{\mathcal{M}}$.* □

Theorem 3. *The following statements are equivalent:*

1. $0 \in \mathcal{M}(\mu X.\mathrm{fm}(\mathcal{A}))$.
2. *There is a μ-succesful run Π through $\mathcal{A}'$ on $\alpha_{\mathcal{M}}$.* □

Theorems 2 and 3 are easily proved using e.g. the model characterisations of [?] or [?].

Example 2. 1. The automaton $\mathcal{A}'_1$ of fig. 3 is the intermediate automaton obtained from $\mathcal{A}_1$ of fig. 1 with respect to the recursion variable X. States that are not accessible from the initial state have for clarity been removed. All infinite runs through $\mathcal{A}'_1$ are ν-successful, and only runs that eventually visits the state $\{q_1, q_3\}$ are μ-successful.

2. Similarly $\mathcal{A}'_2$ of fig 4 is the intermediate automaton obtained from $\mathcal{A}_2$ of fig. 2 with recursion variable Z. Again inaccessible states have been removed. Runs are ν-successful if the transition $\{q_1, q_2, q_3\} \xrightarrow{\{X\}} \{q_1, q_2, q_3\}$ is taken infinitely often. There are no μ-successful runs.

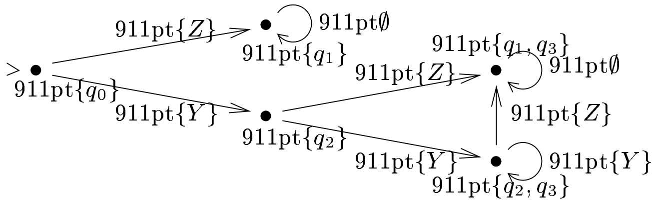

Fig. 3. Intermediate automaton $\mathcal{A}'_1$

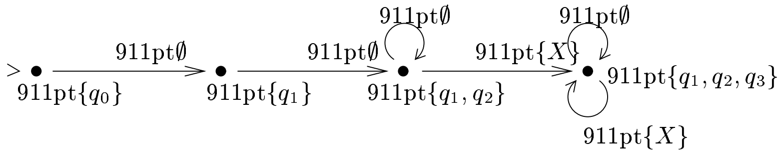

Fig. 4. Intermediate automaton $\mathcal{A}'_2$

5 Greatest Fixed Points

For greatest fixed points Theorem 2 gives rise to a natural idea of resolution of eventualities. Consider a finite run Π from S_1 to S_2 in $\mathcal{A}'$, let $q \in S_1$ and $\pi \in \Pi$ be the direct trail from q. We can view q as resolved at S_2 if $\pi(j)$ is an accepting state for some j. Let then pending(Π) be the subset of S_1 of states that are not resolved at S_2. The idea of the rewriting procedure is embodied by the following easy Lemma:

Lemma 4. *An infinite run Π through $\mathcal{A}'$ is ν-successful iff there is a node S and an infinite, strictly increasing sequence $j_0, j_1, \ldots$ such that for all $k \in \omega$,*

1. *$\Pi(j_k) = S$, and*
2. *$\text{pending}(\Pi(j_k, j_{k+1})) = \emptyset$.* □

For each node S the automaton $\mathcal{A}^\nu_S$ handles the situation where S is visited infinitely often by an infinite run through $\mathcal{A}'$. The desired automaton, $\nu X.\mathcal{A}$, is then built as the sum of the $\mathcal{A}^\nu_S$. The states of each $\mathcal{A}^\nu_S$ are pairs (T, T') where T is a node, and $T' \subseteq T$. The intention is that T' is the set of members of T currently pending. The initial state of $\mathcal{A}^\nu_S$ is the pair $(\{q_0\}, \{q_0\})$, and the single accepting state is the state $(S, \emptyset)$. The transition relation removes pending states as they are resolved, so that there will be a run (of length greater than 1) from $(S, \emptyset)$ to $(S, \emptyset)$ in $\mathcal{A}^\nu_S$ just in case there is a corresponding run Π from S to S in $\mathcal{A}'$ for which pending$(\Pi) = \emptyset$. Formally we let $(T_1, T'_1) \stackrel{a}{\to} (T_2, T'_2)$ iff $T_1 \stackrel{a}{\to} T_2$ in $\mathcal{A}'$, and either

1. T'_1 is nonempty, and T'_2 is the set of all $q_2 \in T_2 - F$ such that q_2 is the direct successor of some $q_1 \in T'_1$, or
2. T'_1 is empty, and then T'_2 is the set of all $q_2 \in T_2 - F$ such that q_2 is the direct successor of some $q_1 \in T_1$.

The correctness of this account is a direct consequence of Lemma 4:

Theorem 5. $L(\nu X.\mathcal{A}) = L(\nu X.\mathrm{fm}(\mathcal{A}))$. □

A pragmatically useful optimisation is that states that are inaccessible from the initial state, or for which an accepting state is inaccessible, can be removed. This modification applies in the examples to follow.

Example 3. The intermediate automaton $\mathcal{A}_2'$ of fig. 4 gives the greatest fixed point automaton $\nu Z.\mathcal{A}_2$ of fig. 5. In νTL the language recognised by $\nu Z.\mathcal{A}_2$ is $\nu Z.\mathrm{O}((\mathrm{O}(\mu Y.X \vee \mathrm{O}Y)) \wedge Z)$ equivalent to the PTL formula $G\mathrm{O}\mathrm{O}FX$ (and indeed GFX), expressing the fairness related property that X holds infinitely often.

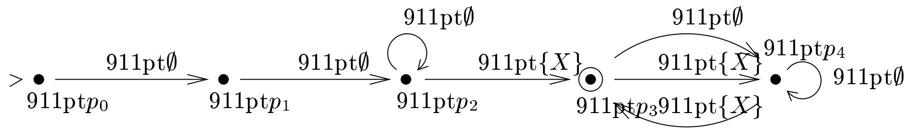

Fig. 5. Büchi automaton $\nu Z.\mathcal{A}_2$

6 Least Fixed Points

For least fixed points we have additionally to take account of trails that do not eventually coincide with a direct trail and are consequently unsuccessful. Let S be any node occurring infinitely often along some infinite run Π through $\mathcal{A}'$. The crucial observation is that it must be possible to order S in a way which prevents trails that are not eventually direct.

Lemma 6. *An infinite run Π through $\mathcal{A}'$ is μ-successful iff there is a node S, a linear order $<$ on S, and an infinite, strictly increasing sequence $j_0, j_1, \ldots$ such that for all $k \in \omega$,*

1. $\Pi(j_k) = S$,
2. $\mathrm{pending}(\Pi(j_k, j_{k+1})) = \emptyset$, *and*
3. *whenever $\pi \in \Pi$ and $\pi(j_k, j_{k+1})$ is not a direct trail then $\pi(j_{k+1}) < \pi(j_k)$.*

Proof. The if direction is easily checked. For the only-if direction assume that Π is μ-successful. Let S be any node visited infinitely often by Π, and let $j_0, j_1, \ldots$ be any infinite, strictly increasing sequence of j_k such that $\Pi(j_k) = S$. For any $q \in S$ and $k \in \omega$ there is some k' such that $q \notin \mathrm{pending}(\Pi(j_k, j_{k'}))$, so as S is finite we can assume both (1) and (2) to be satisfied.

We derive a subsequence and a linear ordering $<$ such that also (3) is satisfied. The ordering $<$ is obtained by defining inductively a numeration $p_0, \ldots, p_m$ of S. For the base case note that there must be some $p_0 \in S$ with the property that for infinitely many k,

$$\text{if } \pi \in \Pi \text{ and } \pi(j_k) = p_0 \text{ then } \pi^{j_k} \text{ is a direct trail.} \qquad (4)$$

For assume this fails to hold. For each $q \in S$ there is some k_q with the property that whenever $k \geq k_q$ then there is a $\pi \in \Pi$ and $k' > k$ such that $\pi(j_k) = q$ and $\pi(j_k, j_{k'})$ is not a direct trail. Let k_0 be largest among $\{k_q \mid q \in S\}$. Pick any $p_0' \in S$. Then we find a $k_1 > k_0$ such that there is a trail $\pi_0 \in \Pi$ where $\pi_0(j_{k_0}, j_{k_1})$ is not direct, and $\pi_0(j_{k_0}) = p_0'$. And we find a $k_2 > k_1$ such that there is a trail $\pi_1 \in \Pi$ where $\pi_1(j_{k_1}, j_{k_2})$ is not direct, and $\pi_1(j_{k_1}) = \pi_0(j_{k_1})$. Continuing ad infinitum an unsuccessful trail through Π is then pieced together. This completes the base case. Note that at the end of the base case we can assume without loss of generality that (4) holds for all $k \in \omega$.

Suppose then we have obtained $p_0, \ldots, p_i$, and let $T_i = \{p_0, \ldots, p_i\}$. If $S = T_i$ we are done. Otherwise there must be some $p_{i+1} \in S - T_i$ such that for infinitely many k,

$$\text{if } \pi \in \Pi, \pi(j_k) = p_{i+1},\ k' > k \text{ and } \pi(j_k, j_{k'}) \text{ is not direct then } \pi(j_{k'}) \in T_i. \quad (5)$$

For if this fails a contradiction is obtained as in the base case. Similarly we can assume here that (5) holds for all $k \in \omega$.

We then define $<$ in the obvious way, by letting $p_i < p_j$ iff $i < j$. It follows that (3) above is satisfied, and the proof is complete. □

Reflecting Lemma 6 the automata $\mathcal{A}^\mu_S$ are built as the sum of automata $\mathcal{A}^\mu_{(S,<)}$ where $<$ is a linear ordering of S. In order to check that $<$ is not violated each automaton $\mathcal{A}^\mu_{(S,<)}$ must take into account the states that are accessible both directly and indirectly. For this purpose we define the sets $\mathrm{dir}(T_1) \subseteq S_2$ and $\mathrm{ind}(T_1) \subseteq S_2$ when $T_1 \subseteq S_1$ and $S_1 \xrightarrow{a} S_2$ in $\mathcal{A}'$:

$$\mathrm{dir}(T_1) = \{p_2 \in S_2 \mid \exists p_1 \in T_1.p_2 \text{ is the direct successor of } p_1\}$$
$$\mathrm{ind}(T_1) = \{p_2 \in S_2 \mid \exists p_1 \in T_1.p_2 \text{ is the indirect successor of } p_1\}$$

The states of $\mathcal{A}^\nu_S$ are augmented by mappings f which given any member q of S produces a pair (T, T') such that T is the subset of the current node which is directly accessible from the last visit to q in S, and T' the subset which is indirectly accessible. The initial state of $\mathcal{A}^\mu_{(S,<)}$ is the state (S, S, f) where f maps each $q \in S$ into the pair $(\{q\}, \emptyset)$. For the transition relation we let $(S_1, S_1', f_1) \xrightarrow{a} (S_2, S_2', f_2)$ iff

1. $(S_1, S_1') \xrightarrow{a} (S_2, S_2')$ in $\mathcal{A}^\nu_S$, and
2. for all $q \in S$, if $f_1(q) = (T_1, T_1')$ then $f_2(q) = (\mathrm{dir}(T_1), \mathrm{dir}(T_1') \cup \mathrm{ind}(T_1) \cup \mathrm{ind}(T_1'))$.

To produce $\mathcal{A}^\mu_{(S,<)}$ it remains to fix the accepting state. For this purpose say that a node (S, S', f) is *consistent* with $<$ if whenever $q \in S$, $f(q) = (T, T')$ and $q' \in T'$ then $q' < q$. The accepting states of $\mathcal{A}^\mu_{(S,<)}$ are all states of the form $(S, \emptyset, f)$ that are consistent with $<$. The automaton $\mu X.\mathcal{A}$ is then obtained from $\mathcal{A}'$ by replacing each node S of $\mathcal{A}'$ with the sum of $\mathcal{A}^\mu_S$ and $\mathcal{A}'(S)$, $\mathcal{A}'$ with initial state S in place of $\{q_0\}$. Thus runs are allowed to violate the ordering for an arbitrarily long initial segment. We obtain:

Theorem 7. $L(\mu X.\mathcal{A}) = L(\mu X.\mathrm{fm}(\mathcal{A}))$. □

Example 4. Fig. 6 shows the least fixed point automaton $\mu X.\mathcal{A}_1$ resulting from the intermediate automaton $\mathcal{A}_1'$ of fig. 3. The language recognised by $\mu X.\mathcal{A}_1$ is $\mu X.Z \vee (Y \wedge \bigcirc X)$ in νTL or YUZ in PTL where U is the strong until-operator that requires Z eventually to hold. The greatest fixed point automaton $\nu X.\mathcal{A}_1$ is obtained by letting in addition the state p_4 of fig 6 be accepting. The corresponding property in PTL is $YU'Z$ where U' is the weak until-operator that allows Z never to hold.

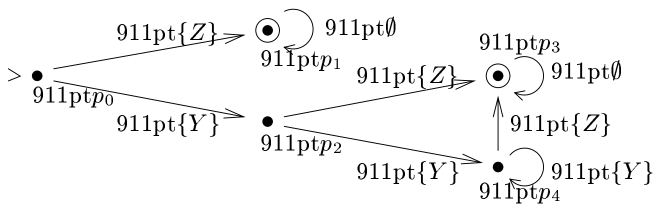

Fig. 6. Büchi automaton $\mu X.\mathcal{A}_1$

A potentially useful optimisation of the least fixed point construction is to introduce a (possibly partial) *ranking* of members of S such that only orderings $<$ need be considered which have the property that if $q_1, q_2 \in S$ are both ranked, and q_1 is of strictly smaller rank than q_2 then $q_1 < q_2$. The ranking can be computed in the following way:

1. If $q \in S$ has the property that no q' is accessible from q such that q' has an indirect successor then q has rank 0.
2. If $q \xrightarrow{(a_1^+ \cup \{X\}, a_1^-)} q'$, q' has rank n, and m is maximal such that whenever $q_0 \xrightarrow{a_2} q''$, $a_1^+ \cap a_2^- = \emptyset$ and $a_1^- \cap a_2^+ = \emptyset$ then q'' has rank m, then q has rank $\max(n, m) + 1$.

7 Applications

Our approach suggests a strategy for obtaining completeness results for νTL. A good candidate for a sound and complete axiomatisation (c.f. [**?**]) adds to some suitable standard axiomatisation of boolean logic and the nexttime operator the axiom $\phi[\mu X.\phi/X] \to \mu X.\phi$ and the rule of fixed point induction:

$$\text{From } \phi[\psi/X] \to \psi \text{ infer } \mu X.\phi \to \psi.$$

We write $\vdash \phi$ if ϕ is provable in an axiomatisation along these lines. Since ϕ and $\mathrm{fm}(\mathrm{aut}(\phi))$ are semantically equivalent, formulas of the form $\mathrm{fm}(\mathcal{A})$ can be viewed as *normal forms* for νTL. Completeness then amounts to showing

1. $\vdash \phi \to \mathrm{fm}(\mathrm{aut}(\phi))$, and
2. if $\mathrm{fm}(\mathcal{A})$ is consistent (i.e. $\not\vdash \neg\mathrm{fm}(\mathcal{A})$) then $L(\mathcal{A}) \neq \emptyset$.

Of these, 2 is not hard to establish. The proof uses an important Lemma due to Kozen [?] which is a proof-theoretic correlate of Winskel's use of relativised fixed points [?].

Lemma 8. *If X is not free in ϕ and $\phi \wedge \mu X.\psi$ is consistent then so is $\phi \wedge \psi[X/\mu X.(\psi \wedge \neg\phi)]$.* □

Using Lemma 8, 2 can be proved by showing that if $\mathrm{fm}(\mathcal{A})$ is consistent then there must be an accepting state in $\mathcal{A}$ which is visited infinitely often along some run. But then it follows that $L(\mathcal{A}) \neq \emptyset$.

Using structural induction 1 can be reduced to showing

(a) $\vdash (\neg)X \to \mathrm{fm}(\mathrm{aut}((\neg)X))$,
(b) $\vdash \mathrm{O}\mathrm{fm}(\mathcal{A}) \to \mathrm{fm}(\mathrm{O}\mathcal{A})$,
(c) $\vdash \mathrm{fm}(\mathcal{A}_1) \vee \mathrm{fm}(\mathcal{A}_2) \to \mathrm{fm}(\mathcal{A}_1 + \mathcal{A}_2)$,
(d) $\vdash \mathrm{fm}(\mathcal{A}_1) \wedge \mathrm{fm}(\mathcal{A}_2) \to \mathrm{fm}(\mathcal{A}_1 \times \mathcal{A}_2)$,
(e) $\vdash \nu X.\mathrm{fm}(\mathcal{A}) \to \mathrm{fm}(\nu X.\mathcal{A})$, and
(f) $\vdash \mu X.\mathrm{fm}(\mathcal{A}) \to \mathrm{fm}(\mu X.\mathcal{A})$.

Of these we have so far only been able to establish (a)–(e). For the aconjunctive fragment, however, our strategy has been more successful. Aconjunctivity is a technical condition due to Kozen [?] which, intuitively, disallows conjunctive branching of the regeneration relation for least fixed point formulas. Since formulas in normal form are aconjunctive, it follows that for νTL the aconjunctive fragment is as expressive as the full language. Completeness for the aconjunctive fragment follows by showing

(i) If $\mu X.\phi$ is aconjunctive then so is $\mu X.\mathrm{fm}(\mathrm{aut}(\phi))$.
(ii) If $\mu X.\mathrm{fm}(\mathcal{A})$ is aconjunctive then $\vdash \mu X.\mathrm{fm}(\mathcal{A}) \to \mathrm{fm}(\mu X.\mathcal{A})$.

Proofs of (i) and (ii) as well as other claims made in this section will be given in the full version of the paper.

8 Concluding Remarks

We have described a syntax-directed procedure for deriving from each νTL formula ϕ an equivalent Büchi automaton $\mathrm{aut}(\phi)$. The construction for greatest fixed points is $2^{\mathcal{O}(n)}$ and for least fixed points $2^{\mathcal{O}(n^2)}$ in the size of $\mathcal{A}$. The overall worst-case complexity of the procedure is thus $2^{\mathcal{O}(n^2)}$. This leaves a small gap to the lower bound which is $2^{\mathcal{O}(n \cdot \log n)}$ [?]. Both Safra [?] and Klarlund [?] have obtained essentially optimal procedures for complementing Büchi automata, and it would be of interest to see if our procedure can be optimised to achieve a $2^{\mathcal{O}(n \cdot \log n)}$ running time. The technical similarities between our work and Klarlunds suggest that this could well be possible. Certainly this running time can be achieved if the construction is modified to use greatest fixed point together with complementation instead of both greatest and least fixed points.