Relating word and tree automata

Zhaowei Xu

State Key Laboratory of Computer Science,
Institute of Software, Chinese Academy of Sciences

May 27, 2013

Outline

Motivation

In this lecture, we consider tree automata that describe derived languages.

Intuitive definition of derived languages:

- *Let L be a language of words.*
- *The derived language of L, denoted L_Δ, consists of all trees all of whose paths are in L.*

The interest in derived languages is that

we want to specify that all the computations of the program satisfying some property.

Motivation of this paper:

- $L_1 = (0+1)^* 1^\omega$ is in $\mathbf{NBW} \backslash \mathbf{DBW}$;
- $\mathbf{NRT} \backslash \mathbf{NBT}$ contains $L_{1\Delta}$;
- $L \in \mathbf{NBW} \backslash \mathbf{DBW}$ iff $L_{1\Delta} \in \mathbf{NRT} \backslash \mathbf{NBT}$?

Outline of the paper

The main result in this paper is

for every word Language L, we have that $L \in NBW \backslash DBW$ iff $L_{1\Delta} \in NRT \backslash NBT$.

And the difficult part in the proof is to show that

if $L_{\Delta} \in NBT$, then $L \in DBW$.

Also, we study the following problems:

- Decide whether the language of a given tree automaton is derivable and show that it is EXPTIME-complete;
- Decide whether the set of trees that satisfy a given branching temporal logic formula is derivable;
- Show how our result can be used in order to obtain inexpressibility results in temporal logic and in order to check the derivability of formulas;
- Discuss the exponential blow-up of the construction and question its optimality.

Outline

Some definitions and notations

Definition. Given a tree automaton $\mathcal{U} = \langle \Sigma, Q, \delta, Q_0, F \rangle$. For $S \subseteq Q$, we denote by $\mathcal{U}^{\mathcal{S}}$ the tree automaton $\langle \Sigma, Q, \delta, S, F \rangle$ and denote by $\mathcal{U}[S]$ the set of trees accepted by $\mathcal{U}^{S}$. A state q of $\mathcal{U}$ is *null* iff $\mathcal{U}[q] = \varnothing$.

Assumption

Assume that $\mathcal{U}[Q_0] \neq \varnothing$ and eliminate all null states and all transitions that involve null states.

Definition. For $S \subseteq Q$ and $a \in \Sigma$, we denote by $\delta_L(S, a)$ the set of states reachable from S by reading a, on the left branch, disregarding what happens on the right branch, i.e.,

$$\delta_L(S, a) = \{q_l : exists\ q_r\ such\ that\ \langle q_l, q_r \rangle \in \bigcup_{q \in S} \delta(q, a)\}.$$

The set $\delta_R(S, a)$ is defined symmetrically for the right. For two states q and q', and $a \in \Sigma$, we say that

$$q'\ is\ a-reachable\ from\ q\ iff\ q' \in \delta_L(q, a) \cup \delta_R(q, a).$$

Some definitions and notations

Definition. For a word language $L \subseteq \Sigma^\omega$, the *derived language* of L, denoted by L_Δ, is the set of all trees all of whose paths are labeled with words in L. Formally,

$$L_\Delta = \{V \in \Sigma_\mathrm{T} : \textit{all paths } \pi \subset T \textit{ satisfy } V(\pi) \in L\}.$$

For a tree language X and a word language L, we say that L derives X iff $X = L_\Delta$. We say that X is derivable iff there exists some word language L such that L derives X.

Claim. For a word language L and a letter a, let $L^a = \{\sigma : a \cdot \sigma \in L\}$. Let $\mathcal{U}$ be a tree automaton, S a subset of the states of $\mathcal{U}$, and let $\mathcal{U}[S] = L_\Delta$. It is easy to achieve that

$$\mathcal{U}[\delta_L(S,a)] \bigcup \mathcal{U}[\delta_R(S,a)] = \mathcal{U}[\delta_L(S,a)] = \mathcal{U}[\delta_R(S,a)] = L^a_\Delta.$$

Definition. Let $\mathcal{U}[S] = L_\Delta$. For a set $S' \subseteq S$, a letter a, and a direction $d \in \{left, right\}$, we say that S' $d-covers$ $\langle S, a\rangle$, iff $\mathcal{U}[\delta_d(S',a)] = L^a_\Delta$.

An important lemma

Lemma

Let $\mathcal{U}$ be a tree automaton, S a subset of the states of $\mathcal{U}$, and let $\mathcal{U}[S] = L_\Delta$. Then, for every $S' \subseteq S$, and letter a, either S' left-covers $\langle S, a \rangle$ or $S \backslash S'$ right-covers $\langle S, a \rangle$.

Proof.

If S' does not left-cover $\langle S, a \rangle$, there exists a tree $V \in L_\Delta^a \backslash \mathcal{U}[\delta_L(S', a)]$. Consider all trees that have a as their root, V as the left subtree, and some tree in L_Δ^a as the right subtree. All these trees are in L_Δ, yet none of them is in $\mathcal{U}[S']$. Hence, as $L_\Delta = \mathcal{U}[S]$, they are all in $\mathcal{U}[S \backslash S']$. Therefore, since their right subtree is an arbitrary tree in L_Δ^a, it must be that $S \backslash S'$ right-covers $\langle S, a \rangle$. □

Outline

From NBT to DBW

Theorem

If $L \subseteq \Sigma^\omega$ is such that L_Δ is recognized by a nondeterministic Büchi tree automaton, then L is recognized by a deterministic Büchi word automaton.

Proof.

First we give the construction, then prove the correctness.
Given a **NBT** $\mathcal{U} = \langle \Sigma, Q, \delta, Q_0, F \rangle$ that recognizes L_Δ, we construct a **DBW** $\mathcal{A} = \langle \Sigma, 2^Q \times \{0,1\}, \nu, \langle Q_0, 1 \rangle, 2^Q \times \{1\} \rangle$ that recognizes L. Define ν as follows.

- For a state $q = \langle S, g \rangle$ with $S \neq \varnothing$ and $g \in \{0,1\}$, for all $a \in \Sigma$,
  - If $S \cap F$ left-covers $\langle S, a \rangle$, then $\nu(q,a) = \langle \delta_L(S \cap F, a), 1 \rangle$.
  - Otherwise, by the preceding lemma, $S \backslash F$ right-covers $\langle S, a \rangle$, in which case $\nu(q,a) = \langle \delta_R(S \backslash F, a), 0 \rangle$.
- For a state $q = \langle \varnothing, g \rangle$ with $g \in \{0,1\}$, we define $\nu(q,a) = \varnothing$ for all $a \in \Sigma$.

□

From NBT to DBW

Proof.

Given a **NBT** $\mathcal{U} = \langle \Sigma, Q, \delta, Q_0, F \rangle$ that recognizes L_Δ, we construct a **DBW** $\mathcal{A} = \langle \Sigma, 2^Q \times \{0,1\}, \nu, \langle Q_0, 1 \rangle, 2^Q \times \{1\} \rangle$ that recognizes L. Define ν as follows.

- For a state $q = \langle S, g \rangle$ with $S \neq \varnothing$ and $g \in \{0,1\}$, for all $a \in \Sigma$,
  - If $S \cap F$ left-covers $\langle S, a \rangle$, then $\nu(q,a) = \langle \delta_L(S \cap F, a), 1 \rangle$.
  - Otherwise, by the preceding lemma, $S \backslash F$ right-covers $\langle S, a \rangle$, in which case $\nu(q,a) = \langle \delta_R(S \backslash F, a), 0 \rangle$.
- For a state $q = \langle \varnothing, g \rangle$ with $g \in \{0,1\}$, we define $\nu(q,a) = \varnothing$ for all $a \in \Sigma$.

Before the correctness proof, we need a preparation.

- In each step of $\mathcal{A}$, its run on a word $\sigma \in \Sigma^\omega$ either *gets stuck*, or take a *left move*, or take a *right move*. This fixes, for any word σ on which the run does not get stuck, an infinite path $\pi_\sigma \subset T$.
- Consider a node $x \in \pi_\sigma$. The node x has two subtrees. One subtree contains the suffix of π_σ. We say that this subtree *continues with* π_σ. The other subtree is disjoint with π_σ. We say that this subtree *quits* π_σ.

□

From NBT to DBW

Proof.

Given a word $\sigma \in \Sigma^\omega$, we first show that if $\mathcal{A}$ accepts σ, then $\sigma \in L$.

- Let $r = \langle S_0, g_0 \rangle, \langle S_1, g_1 \rangle, \langle S_2, g_2 \rangle, \ldots$ be the accepting run of $\mathcal{A}$ on σ. Since r is accepting, it does not get stuck and there are infinitely many j's with $g_j = 1$.
- Accordingly, we can construct a (not necessarily binary) Q-labeled tree based on r. By König's lemma, we can therefore pick a sequence $r' = q_0, q_1, \ldots$ such that for all $j \geq 0$, we have that $q_j \in S_j$, q_{j+1} is σ_j-reachable from q_j , and there are infinitely many j's with $q_j \in F$.
- We can easily define a tree V according to r', accepted by $\mathcal{U}$, in which $V(\pi_\sigma) = \sigma$. As $\mathcal{U}$ recognizes L_Δ, this implies that $\sigma \in L$.

□

From NBT to DBW

Proof.

Next, we show that if $\mathcal{A}$ does not accept σ, then $\sigma \notin L$. Let $r = \langle S_0, g_0 \rangle$, $\langle S_1, g_1 \rangle$, $\langle S_2, g_2 \rangle, \dots$ be the rejecting run of $\mathcal{A}$ on σ. Consider two cases.

Case 1: r gets stuck.

- For a word $\tau \in \Sigma^\omega$ and $j \geq 0$, let V_τ^j be the tree derived from $\{\tau^j\}$. For all $\tau \in L$ and for all $j \geq 0$ for which τ agrees with σ on their first j letters, we have that $V_\tau^j \in \mathcal{U}[S_j]$. (by induction on j)
- Assume now, by way of contradiction, that $\sigma \in L$. Then, by the above, there exists $j \geq 0$ for which both $S_j = \varnothing$ and $V_\sigma^j \in \mathcal{U}[S_j]$. A contradiction.

□

From NBT to DBW

Proof.

Next, we show that if $\mathcal{A}$ does not accept σ, then $\sigma \notin L$. Let $r = \langle S_0, g_0 \rangle$, $\langle S_1, g_1 \rangle$, $\langle S_2, g_2 \rangle, \ldots$ be the rejecting run of $\mathcal{A}$ on σ. Consider two cases.

Case 2: r does not get stuck, i.e., for all $j \geq 0$, $S_j \neq \varnothing$.
Main idea: We show that there exists a tree V, rejected by $\mathcal{U}$, such that $V(\pi_\sigma) = \sigma$ and all other paths are labeled with words in L. It follows that $\sigma \notin L$.
The construction of V: Define V according to r, proceeding over π_σ and for all $j \geq 0$, we have $V(\pi_\sigma[j]) = \sigma_j$. The subtree that quits π_σ in level j, denoted by V_j, is defined as follows:

- If $S_j \cap F$ left-covers $\langle S_j, \sigma_j \rangle$, we chose as the right subtree some tree in $\mathcal{U}[\delta_L(S_j \cap F, \sigma_j)]$.
- Otherwise (in which case $S_j \backslash F$ right-covers $\langle S_j, \sigma_j \rangle$), we chose as the left subtree some tree in $\mathcal{U}[\delta_R(S_j \backslash F, \sigma_j)] \backslash \mathcal{U}[\delta_L(S_j \cap F, \sigma_j)]$; i.e., a tree that causes V not to be accepted by runs r with $r(\pi_\sigma[j]) \in S_j \cap F$.

□

From NBT to DBW

Proof.

Next, we show that if $\mathcal{A}$ does not accept σ, then $\sigma \notin L$. Let $r = \langle S_0, g_0 \rangle$, $\langle S_1, g_1 \rangle$, $\langle S_2, g_2 \rangle, \ldots$ be the rejecting run of $\mathcal{A}$ on σ. Consider two cases.

Case 2: r does not get stuck, i.e., for all $j \geq 0$, $S_j \neq \emptyset$.
Main idea: We show that there exists a tree V, rejected by $\mathcal{U}$, such that $V(\pi_\sigma) = \sigma$ and all other paths are labeled with words in L. It follows that $\sigma \notin L$.
all other paths are labeled with words in L: To see this, note that

- each such other path has some finite prefix $\sigma_0 \cdot \sigma_1 \cdots \sigma_j$ that agrees with σ and has a suffix that continues as a path in V_j.
- by the definition of V, all the subtrees V_j that quit π_σ satisfy $V_j \in \mathcal{U}[S_{j+1}]$.
- for all $i \geq 0$, all trees Y in $\mathcal{U}[S_i]$, and all paths $\tau \subset T$, we have that $\sigma_0 \cdot \sigma_1 \cdots \sigma_{i-1} \cdot Y(\tau) \in L$. (by induction on i)

□

From NBT to DBW

Proof.

Next, we show that if $\mathcal{A}$ does not accept σ, then $\sigma \notin L$. Let $r = \langle S_0, g_0 \rangle$, $\langle S_1, g_1 \rangle$, $\langle S_2, g_2 \rangle, \ldots$ be the rejecting run of $\mathcal{A}$ on σ. Consider two cases.

Case 2: r does not get stuck, i.e., for all $j \geq 0$, $S_j \neq \emptyset$.
Main idea: We show that there exists a tree V, rejected by $\mathcal{U}$, such that $V(\pi_\sigma) = \sigma$ and all other paths are labeled with words in L. It follows that $\sigma \notin L$.
V **is rejected by** $\mathcal{U}$: Let b be a run of $\mathcal{U}$ on V and let $q_0, q_1, q_2, \ldots$ be the sequence of states that b visits along π_σ. We say that a state q_j *agrees with* ν if the following holds.

- $S_j \cap F$ left-covers $\langle S_j, \sigma_j \rangle$ and $q_j \in S_j \cap F$, or
- $S_j \cap F$ does not left-cover $\langle S_j, \sigma_j \rangle$ and $q_j \in S_j \setminus F$.

We say that a run b agrees with ν iff almost all the states along π_σ agree with ν. That is, if there exists $k \geq 0$ for which all states q_j with $j \geq k$ agree with ν. □

From NBT to DBW

Proof.

Next, we show that if $\mathcal{A}$ does not accept σ, then $\sigma \notin L$. Let $r = \langle S_0, g_0 \rangle$, $\langle S_1, g_1 \rangle$, $\langle S_2, g_2 \rangle, \ldots$ be the rejecting run of $\mathcal{A}$ on σ. Consider two cases.

Case 2: r does not get stuck, i.e., for all $j \geq 0$, $S_j \neq \emptyset$.
Main idea: We show that there exists a tree V, rejected by $\mathcal{U}$, such that $V(\pi_\sigma) = \sigma$ and all other paths are labeled with words in L. It follows that $\sigma \notin L$.
V **is rejected by** $\mathcal{U}$: In order to show that no run of $\mathcal{U}$ accepts V, we prove the following two claims:
Claim 1. For every run b on a tree V with $V[\pi_\sigma] = \sigma$, if b agrees with ν then b is a rejecting run.
Claim 2. If a run b accepts V, then there exist a tree V' and an accepting run b' of $\mathcal{U}$ on V', such that $V'[\pi_\sigma] = \sigma$ and b' agrees with ν.

By the above two claims, there exists no accepting run of $\mathcal{U}$ on V.

□

From NBT to DBW

Proof.

Next, we show that if $\mathcal{A}$ does not accept σ, then $\sigma \notin L$. Let $r = \langle S_0, g_0 \rangle$, $\langle S_1, g_1 \rangle$, $\langle S_2, g_2 \rangle, \ldots$ be the rejecting run of $\mathcal{A}$ on σ. Consider two cases.

Claim 1. For every run b on a tree V with $V[\pi_\sigma] = \sigma$, if b agrees with ν then b is a rejecting run.

Proof of Claim 1: Let $q_0, q_1, \ldots$ be the sequence of states that b visits along π_σ. Since b agrees with ν, then there exists $k \geq 0$ such that for every $j \geq k$, it is possible that q_j is in F only when $S_j \cap F$ left-covers $\langle S_j, \sigma_j \rangle$. But r is a rejecting run, b can visit only finitely many states in F along π_σ. Hence, it is a rejecting run.

□

From NBT to DBW

Proof.

Next, we show that if $\mathcal{A}$ does not accept σ, then $\sigma \notin L$. Let $r = \langle S_0, g_0 \rangle$, $\langle S_1, g_1 \rangle$, $\langle S_2, g_2 \rangle, \ldots$ be the rejecting run of $\mathcal{A}$ on σ. Consider two cases.

Claim 2. If a run b accepts V, then there exist a tree V' and an accepting run b' of $\mathcal{U}$ on V', such that $V'[\pi_\sigma] = \sigma$ and b' agrees with ν.
Proof of Claim 2: Consider a state q_j that appears in the run b along π_σ. We get the following two facts.

- If $j = 0$ or q_{j-1} agrees with ν, and q_j does not agree with ν, then $S_j \bigcap F$ left-covers $\langle S_j, \sigma_j \rangle$ and $q_j \in S_j \backslash F$.
- Since r is a rejecting run, then there exists $k \geq 0$ such that for all $j \geq k$, we have that $S_j \bigcap F$ does not left-cover $\langle S_j, \sigma_j \rangle$.

Thus, if $k = 0$ or if q_{k-1} agrees with ν, then so do all q_j for $j \geq k$.

If $k = 0$, then b agrees with ν. We define $b' = b$, $V' = V$.

□

From NBT to DBW

Proof.

Next, we show that if $\mathcal{A}$ does not accept σ, then $\sigma \notin L$. Let $r = \langle S_0, g_0 \rangle$, $\langle S_1, g_1 \rangle$, $\langle S_2, g_2 \rangle, \ldots$ be the rejecting run of $\mathcal{A}$ on σ. Consider two cases.

Claim 2. If a run b accepts V, then there exist a tree V' and an accepting run b' of $\mathcal{U}$ on V', such that $V'[\pi_\sigma] = \sigma$ and b' agrees with ν.
Proof of Claim 2:
Otherwise, consider the set S_k.

- We first show that if b accepts V, then for every $j \geq 0$, the subtree $V^{\pi_\sigma[j]}$ is in $\mathcal{U}[S_j]$ (by induction on j).
- Since $S_k \backslash F$ right-covers $\langle S_k, \sigma_k \rangle$, there exists $q'_k \in S_k \backslash F$ for which there exist q and q' such that $\langle q', q \rangle \in \delta(q'_k, \sigma_k)$ and $V^{\pi_\sigma[k+1]}$ ($\in \mathcal{U}[S_{k+1}]$) is in $\mathcal{U}[q]$.
- Let V' have some tree in $\mathcal{U}[q']$ as the left tree of $\pi_\sigma[k]$(instead V_k was there in V) and $b'(\pi_\sigma[k]) = q'_k$. Then b' continues on the left and right subtrees with some accepting run. To make sure b' is a legal run, we can climb up π_σ and repair b' further until reaching the root.

Outline

Relating word and tree automata

Definition 1. Given a deterministic word automaton $\mathcal{A} = \langle \Sigma, Q, \delta, Q^0, F \rangle$, let $\mathcal{A}_t = \langle \Sigma, Q, \delta_t, Q^0, F \rangle$ be the tree automaton where for every $q \in Q$ and $a \in \Sigma$ with $\delta(q, a) = q'$, we have $\delta_t(q, a) = \langle q', q' \rangle$.

Definition 2. Given a tree automaton $\mathcal{U} = \langle \Sigma, Q, \delta, Q^0, F \rangle$, we define the word automaton $\mathcal{U}_w = \langle \Sigma, Q, \delta_w, Q^0, F \rangle$, where for every $q \in Q$ and $a \in \Sigma$, we have $\delta_w(q, a) = \{q' : q'$ is a-reachable from q in $\delta\}$.

Lemma 1. For every deterministic word automaton $\mathcal{A}$ and word language L, if $\mathcal{A}$ recognizes L, then $\mathcal{A}_t$ recognizes L_Δ.

Lemma 2. For every tree automaton $\mathcal{U}$ and word language L, if $\mathcal{U}$ recognizes L_Δ, then $\mathcal{U}_w$ recognizes L.

Proof.

We first prove that if $\sigma \in L$ then $\mathcal{U}_w$ accepts σ. Let V_σ be the tree derived from $\{\sigma\}$. Since $V_\sigma \in L_\Delta$, there exists an accepting run r of $\mathcal{U}$ on it. It is easy to see that each path of r suggests a legal and accepting run of $\mathcal{U}_w$ on σ.

Assume now that $\mathcal{U}_w$ accepts σ. It is easy to see that then, we can construct a tree V such that V has a path labeled σ and V is accepted by $\mathcal{U}$. Hence, it must be that $\sigma \in L$. □

Relating word and tree automata

Theorem

For every word language L,

$$L \in NBW \backslash DBW \Leftrightarrow L_{\Delta} \in NRT \backslash NBT.$$

Proof.

We prove the following four claims. The ⇒ direction follows from the first two claims and the ⇐ direction follows from the last two.

1. $L \in NBW \Rightarrow L_{\Delta} \in NRT$.
2. $L_{\Delta} \in NBT \Rightarrow L \in DBW$.
3. $L_{\Delta} \in NRT \Rightarrow L \in NBW$.
4. $L \in DBW \Rightarrow L_{\Delta} \in NBT$.

As NBW=DRW, Claim 1 follows immediately from Lemma 1. Claim 2 follows from the main theorem. Claim 3 follows from Lemma 2 and the fact that NBW=NRW. Finally Lemma 1 implies Claim 4. □

Outline

Derivability and complexity

Theorem. For a Büchi tree automaton $\mathcal{U}$, checking whether $L(\mathcal{U})$ is derivable is EXPTIME-complete.

Proof.

We start with the upper bound. Let $\mathcal{U}$ be an NBT with n states and let $\mathcal{A}$ be the DBW constructed from $\mathcal{U}$ in the main theorem. The size of A is $2^{O(n)}$.
Claim. $L(\mathcal{U}) = L(\mathcal{A})_\Delta$ iff $L(U)$ is derivable.
$\Rightarrow$. Trivial.
$\Leftarrow$. Follows immediately from the main theorem.
Thus, checking the derivability of $\mathcal{U}$ can be reduced to checking the equivalence of $L(\mathcal{U})$ and $L(\mathcal{A})_\Delta$.

□

Derivability and complexity

Theorem. For a Büchi tree automaton $\mathcal{U}$, checking whether $L(\mathcal{U})$ is derivable is EXPTIME-complete.

Proof.

$L(\mathcal{U}) \subseteq L(\mathcal{A})_\Delta$:

- Let $\mathcal{C}$ be an NBW that complements $\mathcal{A}$ with size $2^{O(n)}$.
- Extend $\mathcal{C}$ to an NBT $\mathcal{B}$ of size $2^{O(n)}$ that accepts a tree $\langle T, V\rangle$ iff there exists a path $\pi \subseteq T$ such that $V(\pi) \in L(\mathcal{C})$. So, $L(\mathcal{B}) = \overline{L(\mathcal{A})_\Delta}$.
- Construct $\mathcal{U} \times \mathcal{B}$ as NBT of size $|\mathcal{U}| \cdot |\mathcal{B}|$, such that $L(\mathcal{U} \times \mathcal{B}) = L(\mathcal{U}) \cap L(\mathcal{B})$.
- Since the nonemptiness problem for NBT can be solved in quadratic time, the check can be performed in time exponential in n.

□

Derivability and complexity

Theorem. For a Büchi tree automaton $\mathcal{U}$, checking whether $L(\mathcal{U})$ is derivable is EXPTIME-complete.

Proof.

$L(\mathcal{A})_\Delta \subseteq L(\mathcal{U})$

- Expand $\mathcal{A}$ to an NBT $\mathcal{B}$ of size $2^{O(n)}$ that accepts $L(\mathcal{A})_\Delta$.
- Let $\mathcal{C}$ be a Streett tree automaton that complements $\mathcal{U}$, having $2^{O(nlogn)}$ states and $O(n)$ pairs.
- Let $\mathcal{D}$ be the product $\mathcal{B}$ and $\mathcal{C}$, being a Streett automaton with $2^{O(nlogn)}$ states and $O(n)$ pairs.
- Since the nonemptiness problem for nondeterministic Streett tree automata can be solved in time polynomial in the number of states and exponential in the number of pairs, the check can be performed in time exponential in n.

□

Derivability and complexity

Theorem. For a Büchi tree automaton $\mathcal{U}$, checking whether $L(\mathcal{U})$ is derivable is EXPTIME-complete.

Proof.

For the lower bound, we do a reduction from alternating linear-space Turing machines. Given a machine T, we construct an NBT U, of size linear in T, such that $L(\mathcal{U})$ is derivable iff the machine T does not accept the empty tape. □

Results on derivability and expressibility

Definition. A formula ψ of the branching temporal logic CTL^* is derivable iff the set of trees that satisfy ψ is derivable.

Fact. The problem of deciding whether a given CTL^* formula is derivable is in 2EXPTIME.

Proof.

First, we give a claim.

Claim. A CTL^* formula is derivable iff it is equivalent to the LTL formula obtained by eliminating its path quantifiers.

Then, the proof is divided into the following three steps:

- Given a CTL^* formula ψ, let $\psi_l in$ be the LTL formula obtained from by eliminating its path quantifiers.
- By the above claim, ψ is derivable iff the CTL^* formula $\psi \leftrightarrow A\psi_{lin}$ is valid.
- validity of CTL^* is 2EXPTIME-complete.

□

Results on derivability and expressibility

Fact. The CTL^* formula $AFGp$ can not be expressed in $AFMC$.

Proof.

The proof sketch is listed as follows.

- Formulas of $AFMC$ can be translated to NBT.
- The LTL formula FGp can not be translated to a DBW
- By the main theorem, it follow immediately from the above.

□

Fact. A linear-time property can be specified in $AFMC$ iff it can be recognized by a deterministic Büchi automaton.

Outline

A conjecture on the complexity of the construction

Given a nondeterministic Büchi tree automaton $\mathcal{U}$ with n states that recognizes L_Δ, we can construct a deterministic Büchi word automaton $\mathcal{A}$ with 2^{n+1} states that recognizes L.
Question the optimality of the construction and conjecture that a better, perhaps even linear, construction is possible. Our conjecture is supported by the following two evidences.

- An improved construction is possible for the case of finite trees and even trees accepted by a weak automaton.
- The translation from an NBW for L to an NBT for L^k_Δ may be exponential, just like the translation from NBW to DBW.

Thus, the computational advantage of nondeterminism in the case of derivable languages is not clear.