Products of tree automata and temporal logic

Products of tree automata with an application to temporal logic

Zoltán Ésik

Joint work with Szabolcs Iván
Plan

• We associate a branching temporal logic $\text{FTL}(\mathcal{L})$ with each class $\mathcal{L}$ of (regular) tree languages.

• Provide an algebraic characterization of the expressive power of the logics $\text{FTL}(\mathcal{L})$ under some natural assumptions.

• We establish a formal connection between temporal logics and algebra.

• Applications: Szabolcs Iván
Ranked Alphabets and Trees

Definition A rank type \( R \) is a finite set of natural numbers containing 0. A ranked alphabet \( \Sigma \) of rank type \( R \) is a disjoint union of finite nonempty sets \( \Sigma_n, n \in R \).

We fix a rank type \( R \) and only consider ranked alphabets of rank type \( R \).

Notation When \( \Sigma \) is a ranked alphabet, the set of all ground \( \Sigma \)-terms or closed \( \Sigma \)-terms is denoted \( T_\Sigma \). It is well-known that \( T_\Sigma \) is the carrier of the initial \( \Sigma \)-algebra.

We will keep calling terms as trees.
Products of tree automata and temporal logic

Tree Automata

Suppose that $\Sigma$ is a ranked alphabet.

A $\Sigma$-tree automaton is a finite $\Sigma$-algebra that has no nontrivial subalgebra.

When $A$ is a $\Sigma$-algebra each $t \in T_\Sigma$ evaluates to an element $t_A \in A$. We say that a language $L \subseteq T_\Sigma$ is recognizable by a $\Sigma$-tree automaton $A$ if there is a set $F \subseteq A$ with

$$L = \{ t \in T_\Sigma : t_A \in F \}.$$  

A tree language $L \subseteq T_\Sigma$ is called regular if it is recognizable by a tree automaton.
Our logic contains formulas over each ranked set $\Sigma$.

- For each letter $\sigma \in \Sigma$, $p_\sigma$ is a formula.
- If $\varphi$ and $\psi$ are formulas, then $\neg \varphi$, $\varphi \lor \psi$, $\varphi \land \psi$, ... are formulas.
- If $\Delta$ is a ranked set, $(\varphi_\delta)_{\delta \in \Delta}$ is a family of formulas, $K \subseteq T\Delta$ is a pattern language, then

$$K(\delta \mapsto \varphi_\delta)_{\delta \in \Delta}$$

is a formula.

$L$ a class of regular languages – $\text{FTL}(L)$
$K$ a class of finite tree automata – $\text{FTL}(K)$
Suppose that \( \varphi \) is a formula over \( \Sigma \) and \( t \in T_\Sigma \). Then \( t \models \varphi \) if

1. \( \varphi = p_\sigma \) and the root of \( t \) is labeled \( \sigma \), or

2. \( \varphi = \neg \psi \) and \( t \models \psi \) does not hold, or \( \varphi = \psi \lor \psi' \) and \( t \models \psi \) or \( t \models \psi' \) holds, or, ..., or

3. \( \varphi = K(\delta \mapsto \varphi_\delta)_{\delta \in \Delta} \) and the characteristic tree \( \hat{t} \in T_\Delta \) determined by \( t \) and \( (\varphi_\delta)_{\delta \in \Delta} \) is in \( K \).
Semantics 2

\( \hat{t} \) differs from \( t \) only in the labeling of the vertices.

Suppose that \( v \) is a vertex labeled in \( t \) by a letter \( \sigma \in \Sigma_n \). Then the label of \( v \) in \( \hat{t} \) is the first letter \( \delta \in \Delta_n \) such that the subtree \( t|_{v} \) of \( t \) rooted at \( v \) satisfies \( \varphi_\delta \), \( t|_{v} \models \varphi_\delta \), if there is such a letter. Otherwise the label is the last letter in \( \Delta_n \).

We assume a fixed lexicographic order on each ranked alphabet. (But the ordering is not important ...)

6
Products of tree automata and temporal logic

The Characteristic Tree

$t \in T_{\Sigma}$

$\hat{t} \in T_{\Delta}$

$s \models \varphi_\delta$
Example

$\Delta_n = \{\uparrow n, \downarrow n\}, \ n \in R.$

$L_{EF^*} = \{s \in T_\Delta : \exists v \ \exists n \ s(v) = \uparrow n\}$

$L_{EG^*} = \{s \in T_\Delta : \exists p \ \forall v \in p \ \exists n \ s(v) = \uparrow n\}

$\text{Then for any } t \in T_\Sigma, \ t \models L_{EF^*}(\uparrow n \mapsto \varphi, \downarrow n \mapsto \neg \varphi)_{n \in R} \text{ iff for some vertex } v, \ t|_v \models \varphi: \text{ EF modality.}$

And $t \models L_{EG^*}(\uparrow n \mapsto \varphi, \downarrow n \mapsto \neg \varphi)_{n \in R} \text{ iff there is a path } p \text{ such that for each vertex } v \in p, \ t|_v \models \varphi: \text{ EG modality.}$
Earlier Result

**Theorem** Suppose that $\mathcal{L}$ is a class of regular tree languages with:

1. Each quotient of any language in $\mathcal{L}$ is definable in $\text{FTL}(\mathcal{L})$.
2. The “next modalities are expressible in $\text{FTL}(\mathcal{L})$”.

Then a language $L \subseteq T_\Sigma$ is definable in $\text{FTL}(\mathcal{L})$ iff it is regular and its minimal automaton is in the least pseudovariety closed under the cascade product which contains the 2-state reset automaton $D_0$ and the minimal automata of the languages in $\mathcal{L}$.
This Talk

**Theorem** Suppose that $\mathcal{L}$ is a class of regular tree languages such that each quotient of any language in $\mathcal{L}$ is definable in $\text{FTL}(\mathcal{L})$. Then a language $L \subseteq T_\Sigma$ is definable in $\text{FTL}(\mathcal{L})$ iff it is regular and its minimal automaton is in the least pseudovariety closed under the Moore product containing the 2-state reset automaton $D_0$ and the minimal automata of the languages in $\mathcal{L}$.

**Corollary** Suppose that $\mathcal{K}$ is a class of finite tree automata. Then a language $L \subseteq T_\Sigma$ is definable in $\text{FTL}(\mathcal{K})$ iff it is regular and its minimal automaton is in the least pseudovariety closed under the Moore product containing $D_0$ and $\mathcal{K}$. 
Cascade Product (Ricci)

Let $A$ be a $\Sigma$-tree automaton, $B$ a $\Delta$-tree automaton, and $\alpha$ a family of functions $\alpha_n : A^n \times \Sigma_n \to \Delta_n$, $n \in \mathbb{R}$. The cascade product $A \times_{\alpha} B$ determined by $\alpha$ is the least subalgebra of the $\Sigma$-algebra with carrier $A \times B$ and operations

$$\sigma(((a_1, b_1), \ldots, (a_n, b_n))) = (\sigma(a_1, \ldots, a_n), \delta(b_1, \ldots, b_n)),$$

where $\delta = \alpha_n(a_1, \ldots, a_n, \sigma)$, for all $((a_1, b_1), \ldots, (a_n, b_n)) \in A \times B$, $\sigma \in \Sigma_n$, $n \in \mathbb{R}$.

The direct product is clearly a special case of the cascade product.

The cascade product is closely related to the wreath product of function clones (VanderWerf).
Moore Product

Let $A$ be a $\Sigma$-tree automaton, $B$ a $\Delta$-tree automaton. We call a cascade product $A \times_\alpha B$ a Moore product if there exists a family of functions $\beta_n : A \times \Sigma_n \to \Delta_n$, $n \in R$ such that

$$\alpha_n(a_1, \ldots, a_n, \sigma) = \beta_n(\sigma(a_1, \ldots, a_n), \sigma)$$

for all $a_1, \ldots, a_n \in A$ and $\sigma \in \Sigma_n$, $n \in R$. 
Products of tree automata and temporal logic

\[ \text{CTL} = \text{FTL}(\{E_U\}) \text{ where} \]

\( E_U \) is defined on the set \( \{0, 1\} \) by

\[
\begin{align*}
\uparrow_n (b_1, \ldots, b_n) &= 1 \\
\mu_n (b_1, \ldots, b_n) &= b_1 \lor \ldots \lor b_n \\
\downarrow_n (b_1, \ldots, b_n) &= 0,
\end{align*}
\]

for all \( b_1, \ldots, b_n \in \{0, 1\} \), \( n \in R \).

\textbf{Theorem} A tree language is definable in \( \text{CTL} \) iff its minimal tree automaton is in the least pseudovariety containing \( E_U \) which is closed under the cascade product.
Products of tree automata and temporal logic

\[ \text{CTL}(\text{EF}^*) \]

\[ \text{CTL}(\text{EF}^*) = \text{FTL}(\{ \mathbb{E}_{\text{EF}^*} \}) \quad \text{where} \]

\[ \mathbb{E}_{\text{EF}^*} \] is defined on the set \( \{0, 1\} \) by

\[
\begin{align*}
\uparrow^n (b_1, \ldots, b_n) &= 1 \\
\downarrow^n (b_1, \ldots, b_n) &= b_1 \lor \ldots \lor b_n
\end{align*}
\]

for all \( b_1, \ldots, b_n \in \{0, 1\}, \ n \in \mathbb{N} \).

\textbf{Theorem} A tree language is definable in \( \text{CTL}(\text{EF}^*) \) iff its minimal tree automaton is in the least pseudovariety containing \( \mathbb{E}_{\text{EF}^*} \) and \( \mathbb{D}_0 \) which is closed under the Moore product.
Wreath Product of Clones

Suppose that \((A, S)\) and \((B, T)\) are function clones. We define a clone on the set \(A \times B\), denoted \((A \times B, S \circ w T)\). The \(n\)-ary functions in \(S \circ w T\) are the functions \((s, f)\), where \(s : A^n \to A\) in \(S\) and \(f\) maps \(A^n\) into the set of \(n\)-ary functions in \(T\), defined by

\[
(s, f)((a_1, b_1), \ldots, (a_n, b_n)) = (s(a_1, \ldots, a_n), f^{(a_1, \ldots, a_n)}(b_1, \ldots, b_n)),
\]

for all \((a_i, b_i) \in A \times B, i \in [n]\).

Then \((s, f) = (s', f')\) iff \(s = s'\) and \(f = f'\). Moreover, the functions \((s, f)\) form a clone: they are closed under composition and contain the projections.