Tree Automata and Applications

Marius

December

Bottom-up tree automata

Definition

A **Bottom-up tree automata (NFTA)** is an automaton with rules $f(q_0, ..., q_{n-1}) \to q$.

Intuition

The idea is to successively reduce term t to a single state, starting from the leave
What about non-closed terms ?

Pumping Lemma

Let L recognizable.
Then there exists k such that for all terms t such that $\mathcal{H}(t) > k$, there exists contexts C, D and a term u s.t

- D is not trivial [i.e not a variable]
- $t = C[D[u]]$
- $\forall n \in \mathbb{N}, C[D^n[u]] \in L$S.

Top-down tree automata

Definition

A **Top-down tree automata (T-NFTA)** is an automaton with rules $q(f(t_0...t_{n-1})) \to (q_0(t_0), ..., q_{n-1}(t_{n-1}))$.

Intuition

The states are applied from the root to the leaves a bit like a morphism.

Equivalence

$NFTA = DFTA = T - NFTA$

All those tree automata are closed under boolean operations.

Homomorphism

Definition

A **Tree Homomorphism** is a mapping $h\mathcal{F}_n \rightarrow T(\mathcal{F}, x_0..x_{n-1})$
Extension to tree : $h(f(a,b)) = h(f)[x_0 \rightarrow a, x_1 \rightarrow b]$

Properties

h is linear if $\forall f, h(f)$ is linear [each variable appears once at most].

Recognizability

Linear homomorphisms preserve recognazibility
Inverse homomorphisms preserve recognazibility
Non-linear does not in general [see
$h(f) = f'(x_0, x_0), h(g) = g(a), L = \{f(g^n(a))\}$]

Definition

The **path language** of $t = f(g(a,b)$ is $\pi(t) = \{f1g1a, f1g2b\}$. [can easily be extended to languages.] The **path closure** of L is the set of terms that one can build with $\pi(L)$ L is **path-closed** is $L = pc(L)$

Theorem

L is path-closed $\Leftrightarrow$ L is recognizable by a T-DFTA.

Definition

A **Congruence on terms** is an equivalence relation compatible with $\mathcal{F}$

A congruence **saturates L** if $u \sim v$ implies $u \in L \Leftrightarrow v \in L$

$u \sim_L v$ if and only if $\forall C, C[u] \in L \Leftrightarrow C[v] \in L$

Myhill-Nerode Theorem

The tree following propositions are equivalent

- L is recognizable
- L is saturated by a congruence of finite index
- $\sim_L$ is of finite index

Application

Proof of counter-example of non-linear homomorphisms

Definition

The **front** of $t = f(a, g(b, a), c)$ is $ft(t) = abac$ [Formalism use positions]
A **Visibly Pushdown Automata (VPA)** is a pushdown automata where the size of the word pushed in the stack relies only on the letter read.

Propositions

Let L be recognizable.

- $fr(L)$ is context-free.
- L is recognizable by a VPA

Definitions

Second-Order Logic quantifies over relations
Monadic Logic quantifies only over sets
Weak Second-Order quantifies only over finite sets
Weak MSO over with k successors is **MSO**($<_1$ ... $<_k$)

Intuition

One have a formula over a tree. [e.g $\exists x, x \in S_g$ i.e g appears in t]
The trees recognized by this formula are $\{(t, v) | v(t)$ satisfies $\phi\}$
Here, v is a valuation to positions and the subterm at position p is $(t[p], (v(x) = p))_{x \in \mathcal{X}})$

Theorem

L is recognizable if and only if there exists ϕ in WSkS such that $L = L(\phi)$.

Definition

An **unranked** tree may have an arbitrary number of children

A **bottom-up hedge automata (NHA)** is an automaton with rules $a(q_0 + (q_1.q_2)^*) \rightarrow q$. **UTL = Weak MSO(child,next)** [Unranked Tree Logic]

Properties

NHA can be determinized

Unranked tree can be mapped one-to-one with ranked tree using a phony symbol and reading the term on the leaves.

Recognizability is then equivalent.

UTL=NHA