FO definable regular tree languages

A (hopefully) correct version of a STACS'05 paper.

Luc Segoufin (INRIA, Paris XI)

Joint work with Michael Benedikt.

The word case

Σ finite alphabet.

Word models: $\langle [1 \cdots n], (P_a)_{a \in \Sigma}, S \rangle$

P_a unary for positions with an "a"

S binary is the successor relation: $\{ \ (i, i+1) \ \}$

φ sentence $\longrightarrow$ L_φ language

The word case

MSO $\equiv$ REG [Büchi 60]

FO($<$) $\equiv$ *-free $\equiv$ aperiodic [Sch. 65, McNaughton-Papert 71]

$$\exists l \forall u, v, w \in \Sigma^* \quad u \cdot \mathbf{v}^{\mathbf{l}} \cdot w \in L \quad \text{iff} \quad u \cdot \mathbf{v}^{\mathbf{l+1}} \cdot w \in L$$

FO $\equiv$ LTT $\equiv$ aperiodic + swap condition [Beauquier-Pin 89]

$\forall u, v, w, s, s', e, f \in \Sigma^*$ e and f idempotents

$$u \cdot \mathrm{e} \cdot s \cdot \mathrm{f} \cdot v \cdot \mathrm{e} \cdot s' \cdot \mathrm{f} \cdot w \in L$$

iff

$$u \cdot \mathrm{e} \cdot s' \cdot \mathrm{f} \cdot v \cdot \mathrm{e} \cdot s \cdot \mathrm{f} \cdot w \in L$$

The word case

MSO $\equiv$ REG [Büchi 60]

decidable characterizations

FO($<$) $\equiv$ *-free $\equiv$ aperiodic [Sch. 65, McNaughton-Papert 71]

$$\exists l \forall u, v, w \in \Sigma^* \quad u \cdot \mathbf{v^l} \cdot w \in L \quad \text{iff} \quad u \cdot \mathbf{v^{l+1}} \cdot w \in L$$

FO $\equiv$ LTT $\equiv$ aperiodic + swap condition [Beauquier-Pin 89]

$\forall u, v, w, s, s', e, f \in \Sigma^*$ — e and f idempotents

$$u \cdot \mathrm{e} \cdot s \cdot \mathrm{f} \cdot v \cdot \mathrm{e} \cdot s' \cdot \mathrm{f} \cdot w \in L$$

iff

$$u \cdot \mathrm{e} \cdot s' \cdot \mathrm{f} \cdot v \cdot \mathrm{e} \cdot s \cdot \mathrm{f} \cdot w \in L$$

Ranked trees

Σ finite alphabet, k rank.

Tree models: $\langle I, (P_a)_{a \in \Sigma}, (S_i)_{1 \leq i \leq k} \rangle$

$I \subseteq \{1, \cdots, k\}^*$ closed under prefix

P_a unary for positions with label "a"

S_i binary is the i^{th} successor relation

φ sentence $\longrightarrow$ L_φ tree language

Ranked trees

MSO $\equiv$ REG [Thatcher-Wright 68, Doner 70]

Ranked trees

MSO $\equiv$ REG [Thatcher-Wright 68, Doner 70]

FO($<$) $\subsetneq$ *-free [Potthoff-Thomas 93]

FO($<$) $\subsetneq$ aperiodic [Heuter 88, Potthoff 95]

FO $\equiv$ LTT $\equiv$?

Ranked trees

MSO $\equiv$ REG [Thatcher-Wright 68, Doner 70]

FO($<$) $\subsetneq$ *-free [Potthoff-Thomas 93]

FO($<$) $\subsetneq$ aperiodic [Heuter 88, Potthoff 95]

FO $\equiv$ LTT $\equiv$? This talk: decidable characterization

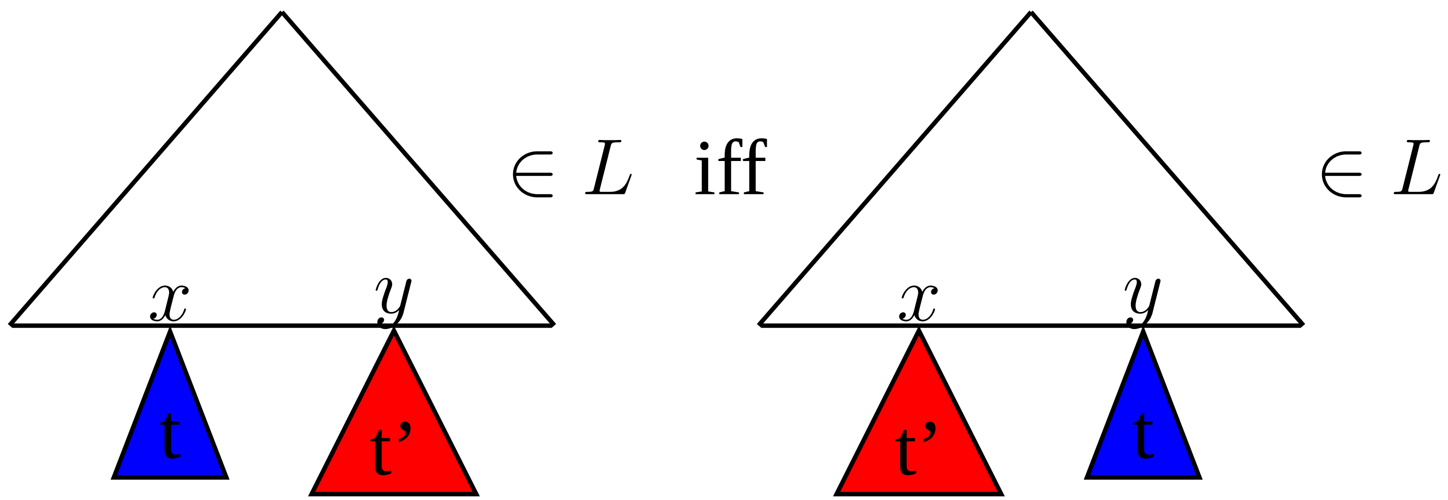
some notations

Two trees are k-similar if they agree up to depth k.

Two nodes of a tree are k-similar if the subtrees rooted at those nodes are.

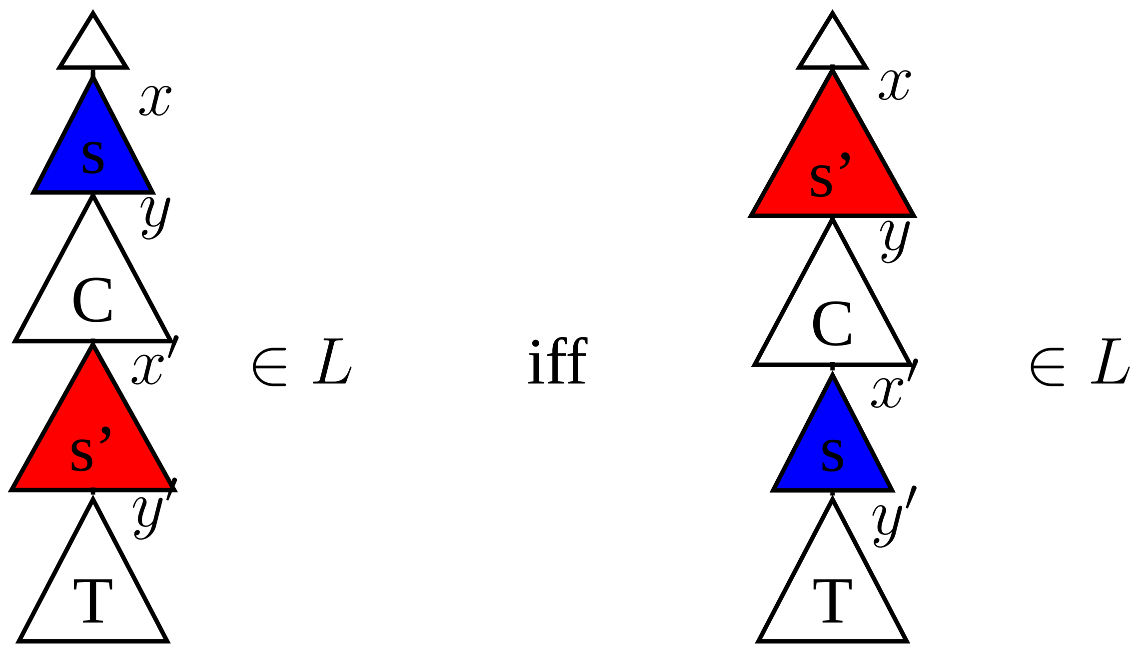
k-horizontal-swap closure

Whenever x and y are k-similar.



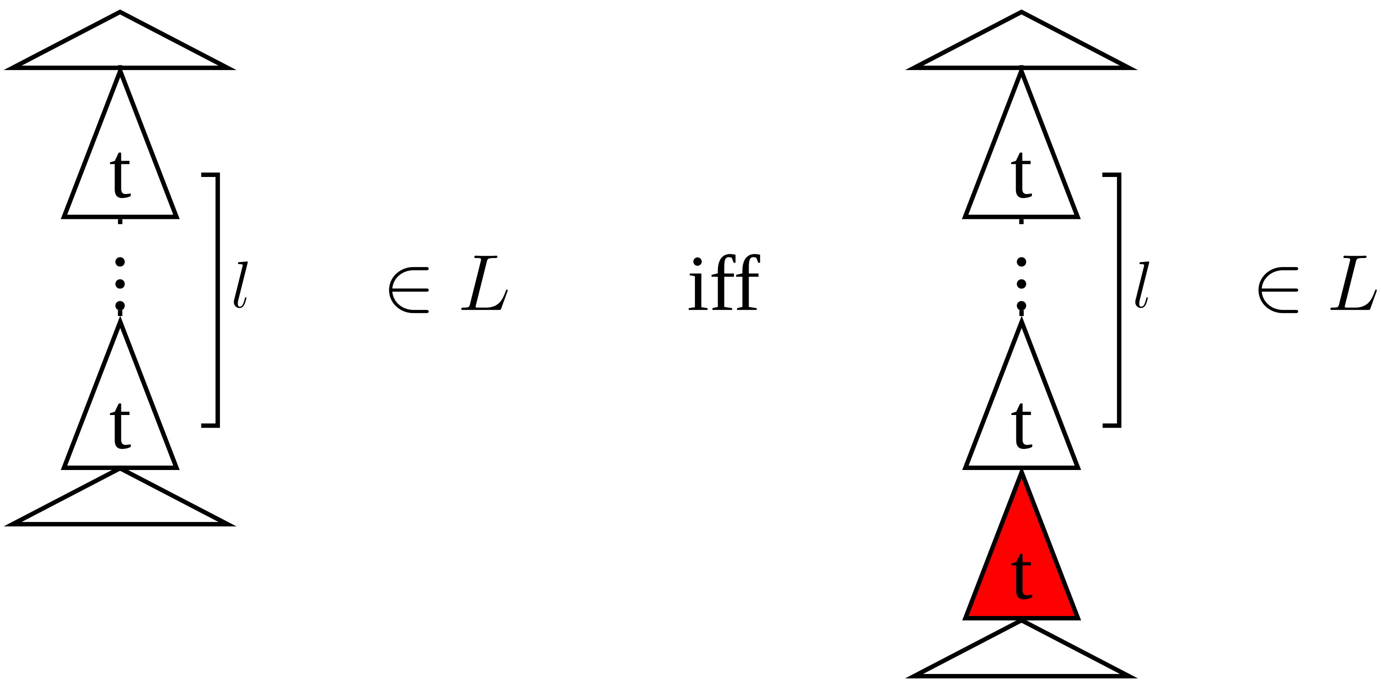
k-vertical-swap closure

Whenever x, x' are k-similar and y, y' are k-similar.



Aperiodicity

$\exists l \in \mathbb{N}$



Theorem A regular language L is definable in FO iff $\exists k \in \mathbb{N}$ such that L is

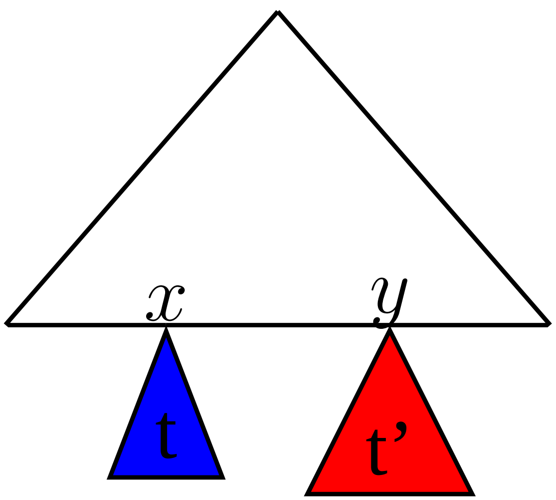
- aperiodic
- closed under k-horizontal-swaps
- closed under k-vertical-swaps.

proof of the left to right direction

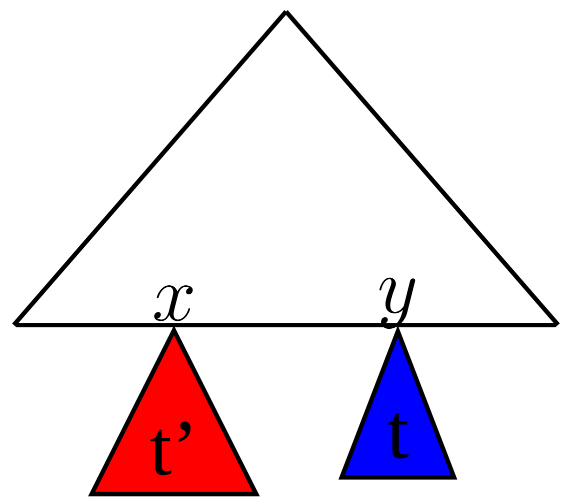
L regular, definable in FO by φ_L.

Let k be the locality rank of φ_L. Then φ_L cannot distinguish between structures having the same k-neighborhoods.

This implies that L is closed under k-swaps.



have the same k-neighborhoods whenever x and y are k-similar



proof of the reverse direction

L regular language, A min. det. automata recognizing L. Assume L is aperiodic and closed under k-types.

- k-similarity has finite index: k-types.
- $t =_d^k t'$: same number of k-types up to threshold d.
- $t \leq_d^k t'$: moreover t' has always more k-types than t.
- $t <_d^k t'$: t' has always strictly more k-types than t.
- $t \hookrightarrow_k t'$: if t can be topologically embedded into t' with a mapping preserving k-types.
- $T \subset_k t'$: if T is a subtree of t' preserving k-types.

Lemma 1. For K *large enough,* $s \equiv_K t \Rightarrow s =_d^{k+1} t$

Lemma 1. For K *large enough*, $s \equiv_K t \Rightarrow s =_d^{k+1} t$

Proof: k-types can be described in FO

Lemma 1. For K *large enough,* $s \equiv_K t \Rightarrow s =_d^{k+1} t$

Lemma 2. For d *large enough,* $s =_d^{k+1} t \Rightarrow s \leq_{d'}^{k+1} t'$

$t \in L$ **iff** $t' \in L$

Lemma 1. For K *large enough,* $s \equiv_K t \Rightarrow s =_d^{k+1} t$

Lemma 2. For d *large enough,* $s =_d^{k+1} t \Rightarrow s \leq_{d'}^{k+1} t'$

$t \in L$ **iff** $t' \in L$

Proof: Pumping argument.

Lemma 1. For K *large enough,* $s \equiv_K t \Rightarrow s =_d^{k+1} t$

Lemma 2. For d *large enough,* $s =_d^{k+1} t \Rightarrow s \leq_{d'}^{k+1} t'$

$t \in L$ **iff** $t' \in L$

Lemma 3. $s \leq_{d'}^{k+1} t \Rightarrow s \hookrightarrow_k t'$

$t =_\infty^{k+1} t'$ **and** $t \in L$ **iff** $t' \in L$

Lemma 1. For K *large enough,* $s \equiv_K t \Rightarrow s =_d^{k+1} t$

Lemma 2. For d *large enough,* $s =_d^{k+1} t \Rightarrow s \leq_{d'}^{k+1} t'$

$t \in L$ **iff** $t' \in L$

Lemma 3. $s \leq_{d'}^{k+1} t \Rightarrow s \hookrightarrow_k t'$

$t =_\infty^{k+1} t'$ **and** $t \in L$ **iff** $t' \in L$

Proof: Difficult. By induction on the tree s. Uses only k-swaps.

Lemma 1. For K *large enough,* $s \equiv_K t \Rightarrow s =_d^{k+1} t$

Lemma 2. For d *large enough,* $s =_d^{k+1} t \Rightarrow s \leq_{d'}^{k+1} t'$

$t \in L$ **iff** $t' \in L$

Lemma 3. $s \leq_{d'}^{k+1} t \Rightarrow s \hookrightarrow_k t'$

$t =_\infty^{k+1} t'$ **and** $t \in L$ **iff** $t' \in L$

Lemma 4. $S <_{d'}^{k+1} t \Rightarrow S \subset_k t'$

$t =_\infty^{k+1} t'$ **and** $t \in L$ **iff** $t' \in L$

Lemma 1. For K *large enough,* $s \equiv_K t \Rightarrow s =_d^{k+1} t$

Lemma 2. For d *large enough,* $s =_d^{k+1} t \Rightarrow s \leq_{d'}^{k+1} t'$

$t \in L$ **iff** $t' \in L$

Lemma 3. $s \leq_{d'}^{k+1} t \Rightarrow s \hookrightarrow_k t'$

$t =_\infty^{k+1} t'$ **and** $t \in L$ **iff** $t' \in L$

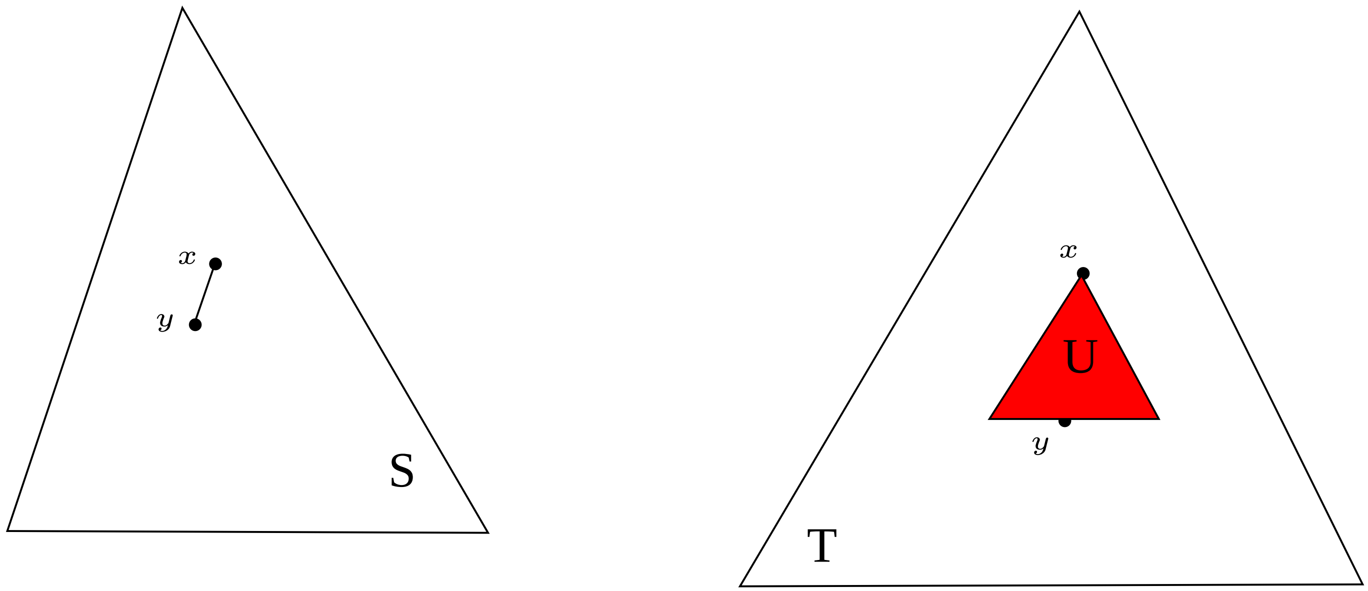
Lemma 4. $S <_{d'}^{k+1} t \Rightarrow S \subset_k t'$

$t =_\infty^{k+1} t'$ **and** $t \in L$ **iff** $t' \in L$

Proof: Same as in Lemma 3. During the induction one extra available k-type is enough to guarantee that the embedding is tight.

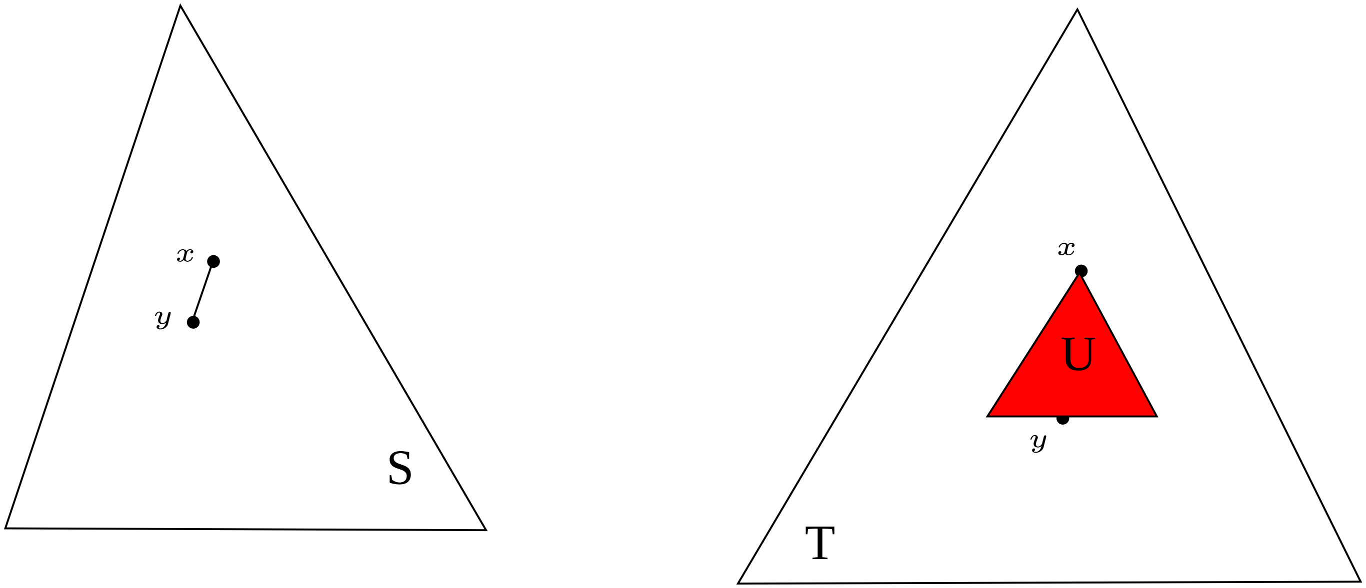
After Lemma 3 we have: $s \hookrightarrow_k t$ and $s =_{d'}^{k+1} t$.

We want to show $s \in L$ iff $t \in L$.



After Lemma 3 we have: $s \hookrightarrow_k t$ and $s =_{d'}^{k+1} t$.

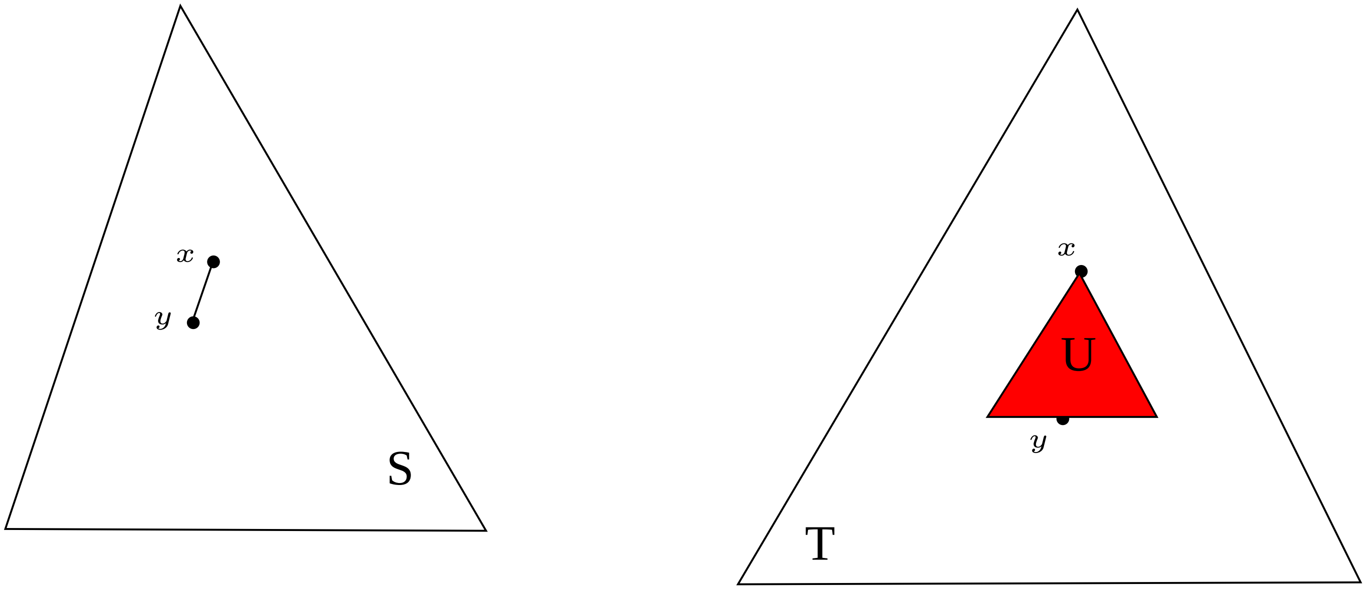
We want to show $s \in L$ iff $t \in L$.



- Using pumping, U can be "minimized" without changing its set of $(k+1)$-types.
- Fix $d' > \min(|U|).l$.
- All $(k+1)$-types of U^l appear strictly more in s than in U^l.

After Lemma 3 we have: $s \hookrightarrow_k t$ and $s =_{d'}^{k+1} t$.

We want to show $s \in L$ iff $t \in L$.

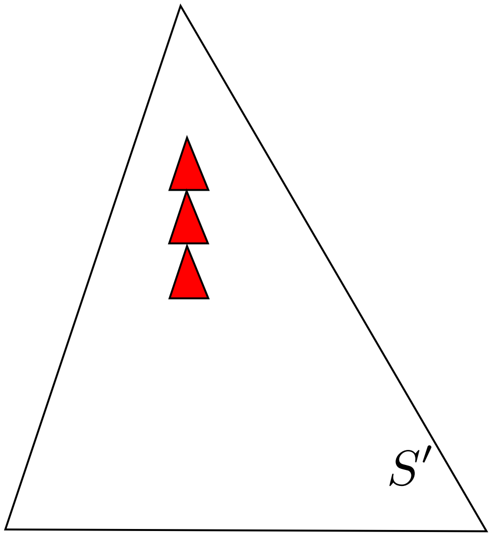


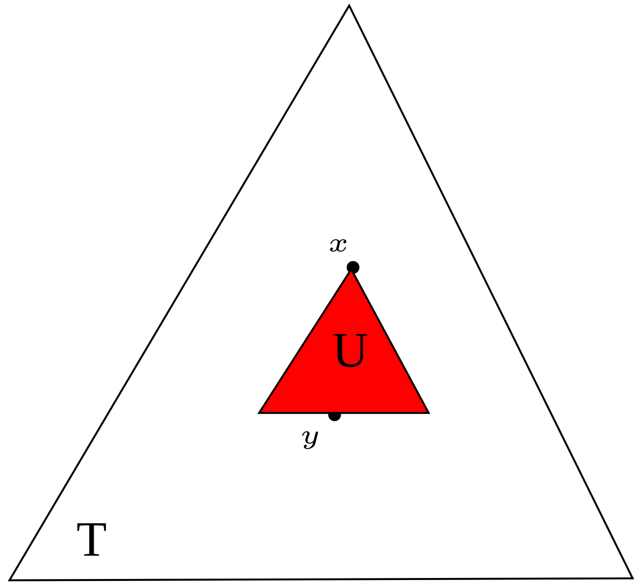
- Using pumping, U can be "minimized" without changing its set of $(k+1)$-types.
- Fix $d' > \min(|U|).l$.
- All $(k+1)$-types of U^l appear strictly more in s than in U^l.

$\Rightarrow$ We can apply Lemma 4 to U^l.

After Lemma 3 we have: $s \hookrightarrow_k t$ and $s =_{d'}^{k+1} t$.

We want to show $s \in L$ iff $t \in L$.



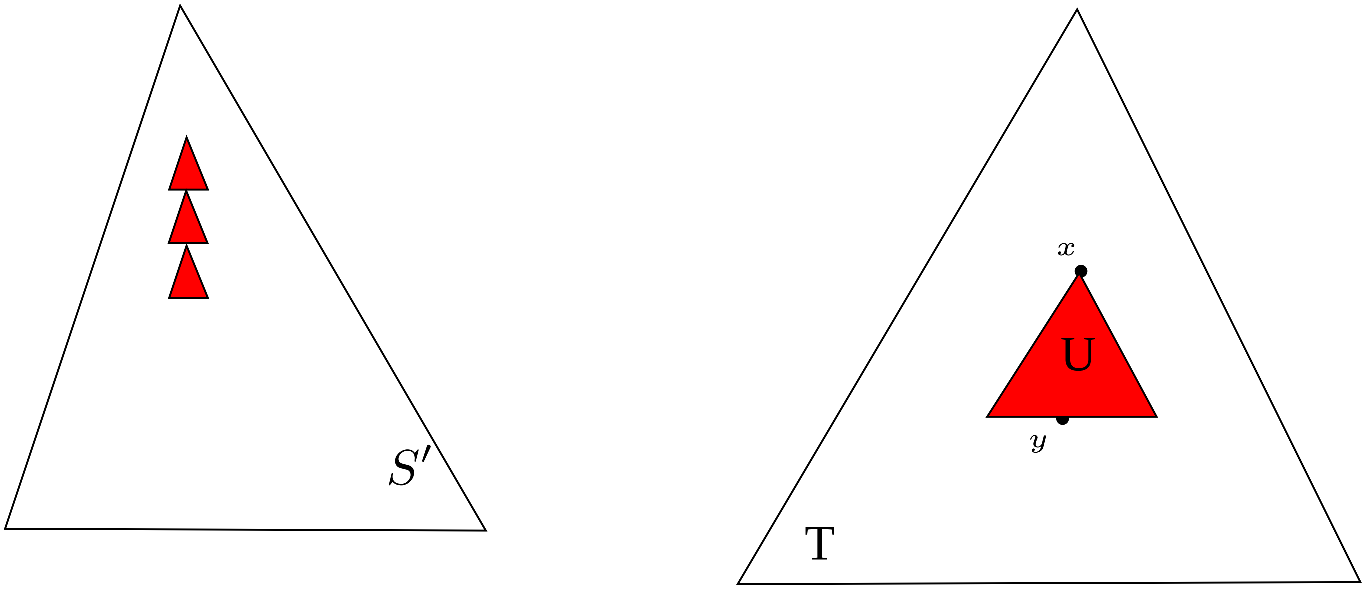


- Using pumping, U can be "minimized" without changing its set of $(k+1)$-types.
- Fix $d' > \min(|U|).l$.
- All $(k+1)$-types of U^l appear strictly more in s than in U^l.

$\Rightarrow$ We can apply Lemma 4 to U^l.

After Lemma 3 we have: $s \hookrightarrow_k t$ and $s =_{d'}^{k+1} t$.

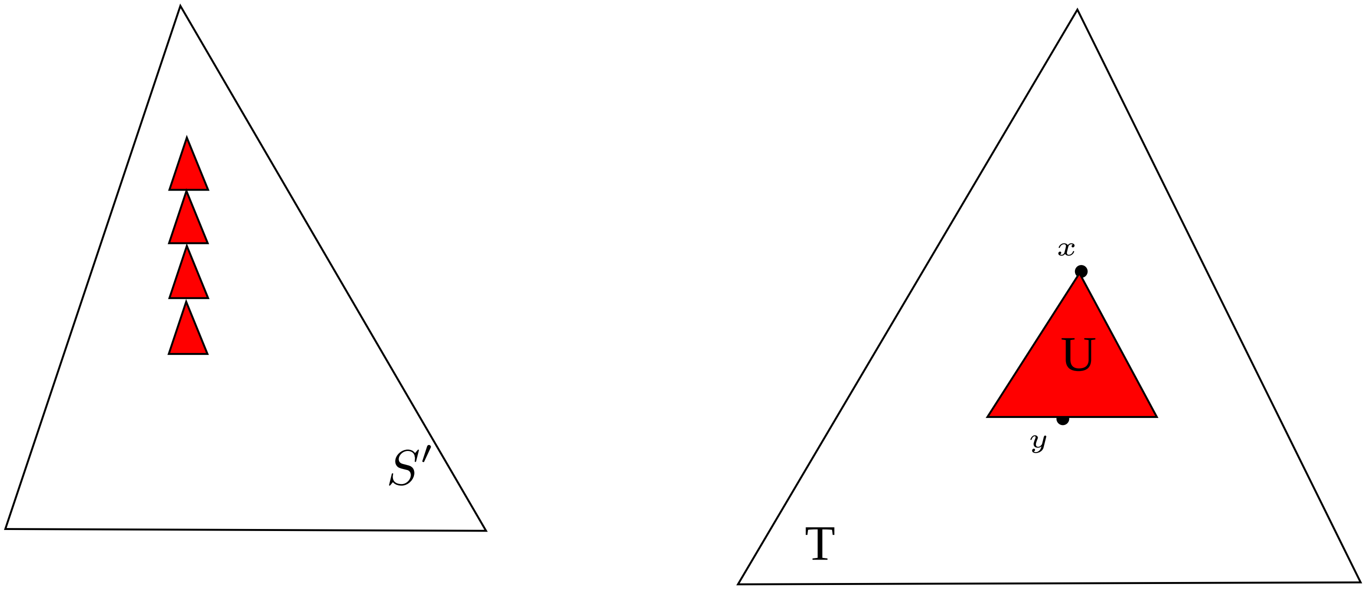
We want to show $s \in L$ iff $t \in L$.



- We can now use aperiodicity of L...

After Lemma 3 we have: $s \hookrightarrow_k t$ and $s =_{d'}^{k+1} t$.

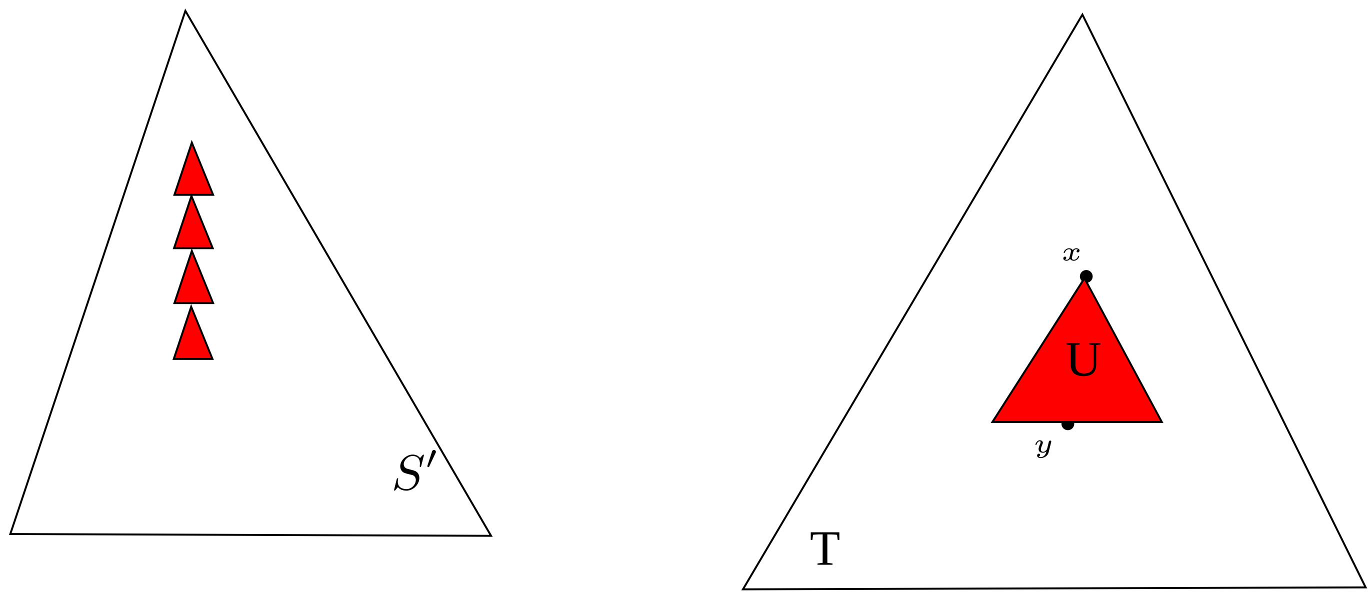
We want to show $s \in L$ iff $t \in L$.



- We can now use aperiodicity of L...

After Lemma 3 we have: $s \hookrightarrow_k t$ and $s =_{d'}^{k+1} t$.

We want to show $s \in L$ iff $t \in L$.

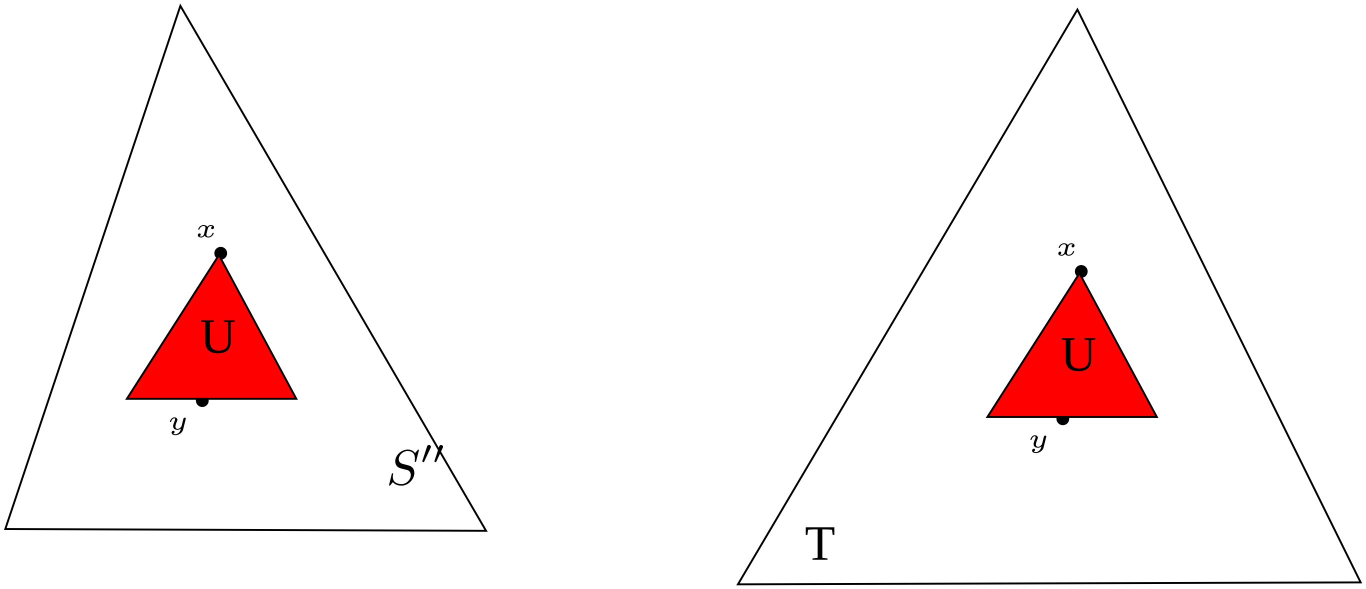


- We can now use aperiodicity of L...

- ... and undo the "inclusion" of U^l.

After Lemma 3 we have: $s \hookrightarrow_k t$ and $s =_{d'}^{k+1} t$.

We want to show $s \in L$ iff $t \in L$.

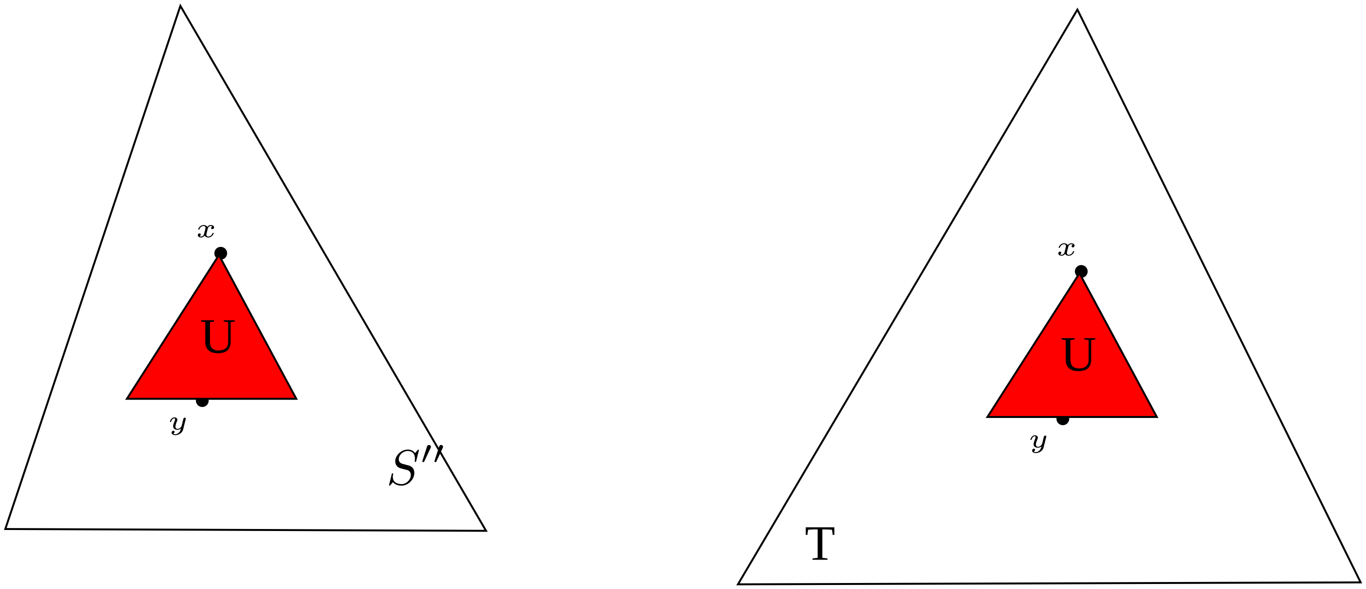


- We can now use aperiodicity of L...

- ... and undo the “inclusion” of U^l.

After Lemma 3 we have: $s \hookrightarrow_k t$ and $s =_{d'}^{k+1} t$.

We want to show $s \in L$ iff $t \in L$.



- We can now use aperiodicity of L...

- ... and undo the "inclusion" of U^l.
- Eventually the two trees are equal.

This showed, assuming L regular, aperiodic and closed under k-swaps:

$$\exists K, \quad s \equiv_K t \Rightarrow s \in L \text{ iff } t \in L$$

Therefore L is a union of classes of $\equiv_K$.

Classical FMT shows that:

- For each K, $\equiv_K$ has finite index,
- each class of $\equiv_K$ is definable in FO.

This prove the Theorem.

Application : decidability

Theorem. It is decidable (in PTIME) from a deterministic minimal bottom-up automaton whether the language it recognize is definable in FO

Proof: check that horizontal swap, vertical swap and aperiodicity holds.

- Known: aperiodicity is decidable (l can be assumed polynomial).
- Easy: for each k decide whether L is closed under k swaps.
- Claim: k can be assumed polynomial.

L closed under k-swaps implies L closed under $(n^2 + 1)$-swaps.

Proof: pumping argument.

Extensions

- The characterization extends to unranked trees.
- Removing the aperiodicity condition characterizes the logic FO+MOD, which extends FO with modulo quantifiers.

Open problems

- Decidable characterization of LT (no modulo counting of the local neighborhoods).
- Decidable characterization of FO($<$).
- Find a "reasonable" formulation of the swaps in the algebra that will be presented in the next talk.