

# Rewriting Higher-Order Stack Trees

Vincent Penelle

CRIStAL, 21 janvier 2016

# Table of Contents

- 1 Introduction
- 2 Stack Trees
- 3 Properties of Stack Trees
- 4 Perspectives

# Context

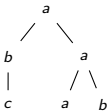
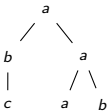
- Study of infinite graphs: presentations, logic and languages.
- Model Checking over some classes of infinite graphs: which logic theories (FO,FO[ $\rightarrow^*$ ],MSO,...) are decidable ?

# Context

- Study of infinite graphs: presentations, logic and languages.
- Model Checking over some classes of infinite graphs: which logic theories (FO,  $\text{FO}[\rightarrow^*]$ , MSO, ...) are decidable ?

Configuration graphs of HOPDA <i>[Caucal02] &amp; [Carayol,Wohrle03]</i>	?	Tree automatic of order $n$ <i>[Colcombet,Loding07]</i>
Configuration graphs of PDA <i>[Muller,Shupp85]</i>	Ground tree rewriting graphs <i>[Dauchet,Tison90]</i>	Tree automatic <i>[Khossainov,Nerode94]</i>
MSO	$\text{FO}[\rightarrow^*]$	FO

# Internal representations

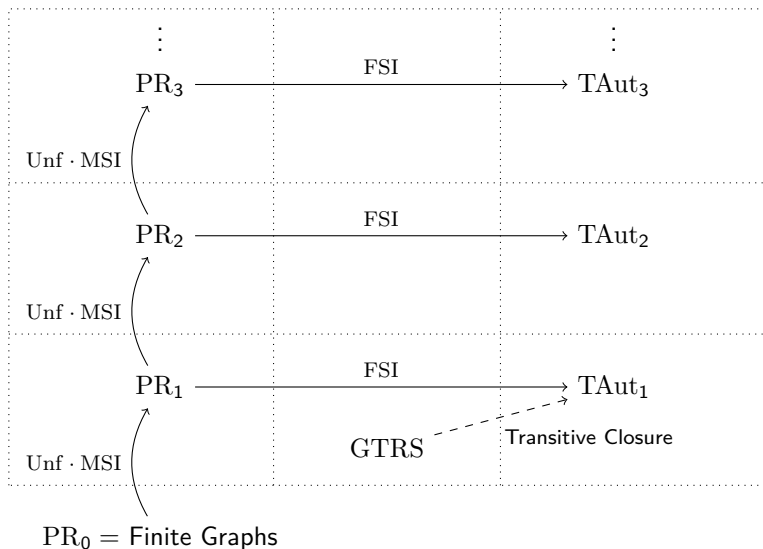
$[[\dots]_{n-1}[\dots]_{n-1}]_n$ $\vdots$ $[[ba]_1[ba]_1[ba]_1]_2$	?	?
$[abaaa]$		

MSO

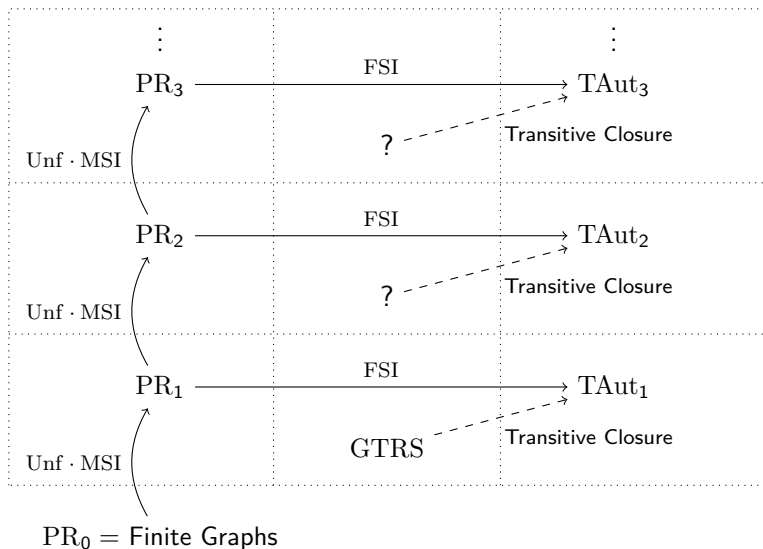
FO[ $\rightarrow^*$ ]

FO

# External Representations



# External Representations



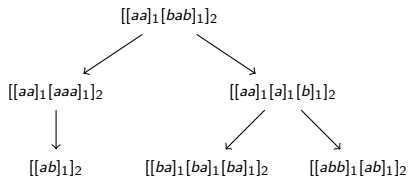
# Table of Contents

- 1 Introduction
- 2 Stack Trees**
- 3 Properties of Stack Trees
- 4 Perspectives



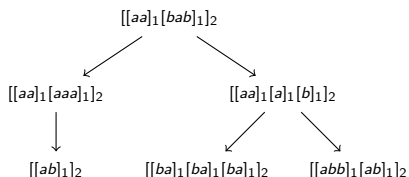
# Higher-Order Stack Trees

A  $n$ -stack tree is a tree labelled by  $(n - 1)$ -stacks.



# Higher-Order Stack Trees

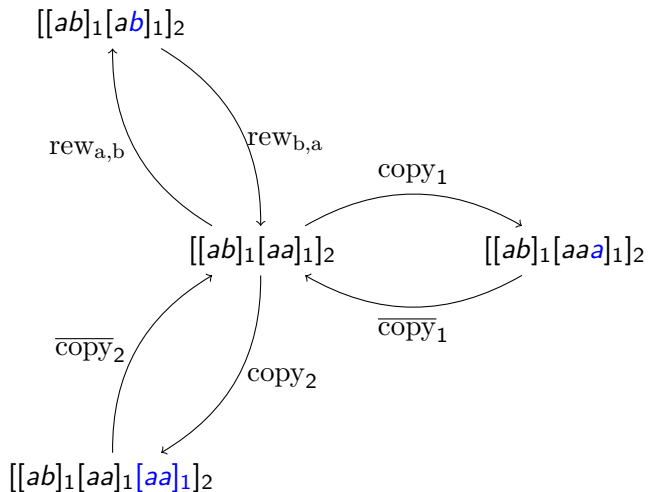
A  $n$ -stack tree is a tree labelled by  $(n - 1)$ -stacks.



A unary  $n$ -stack tree is a  $n$ -stack.

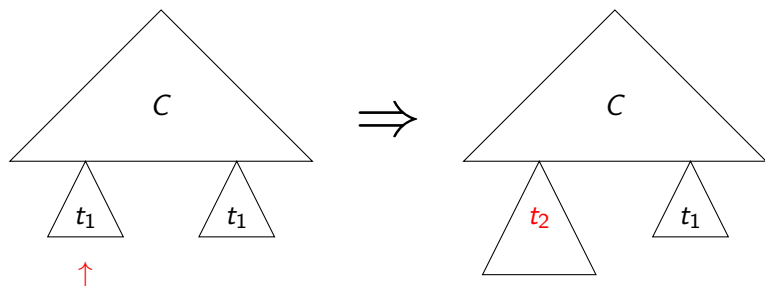
A 1-stack tree is a tree labelled by  $\Sigma$ .

# Higher-order Stack Operations



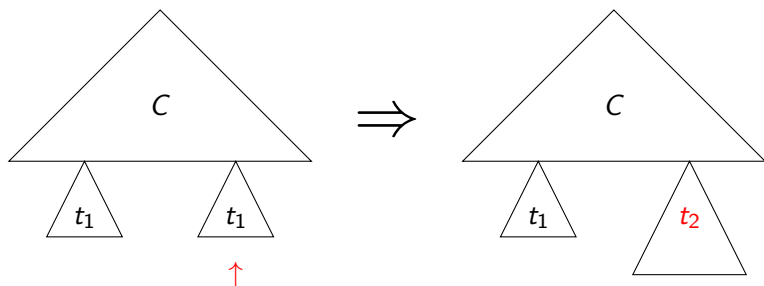
# Ground Tree Operation

A ground tree operation is a pair of trees  $(t_1, t_2)$



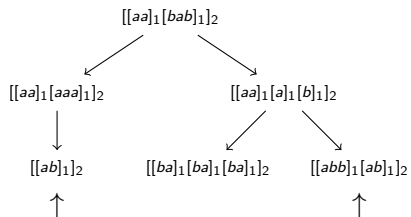
# Ground Tree Operation

A ground tree operation is a pair of trees  $(t_1, t_2)$



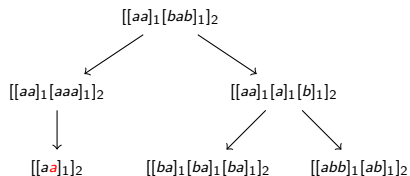
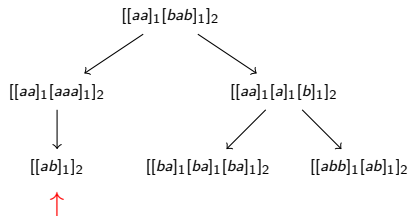
# Basic Operations: Level 0

Rewriting operations over  $\Sigma$ :  $\text{rew}_{b,a}$



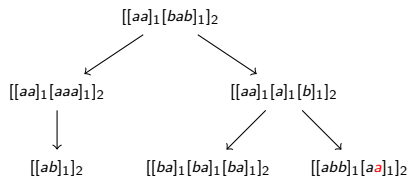
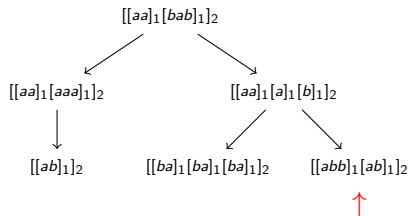
# Basic Operations: Level 0

Rewriting operations over  $\Sigma$ :  $\text{rew}_{b,a}$



# Basic Operations: Level 0

Rewriting operations over  $\Sigma$ :  $\text{rew}_{b,a}$



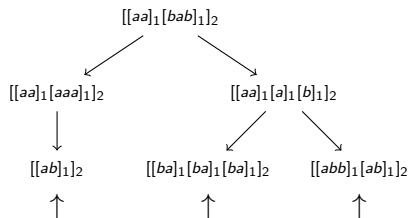


# Basic Operations: Level $i < n$

Copy operations:  $\text{copy}_i, \overline{\text{copy}}_i$

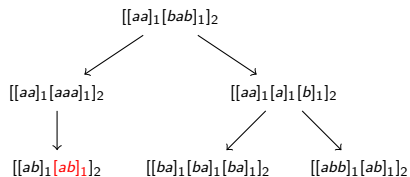
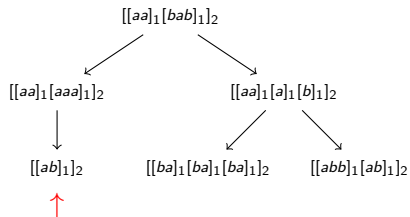
# Basic Operations: Level $i < n$

Copy operations:  $\text{copy}_2$



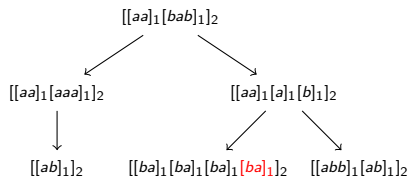
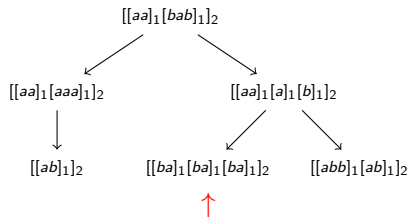
# Basic Operations: Level $i < n$

Copy operations:  $\text{copy}_2$



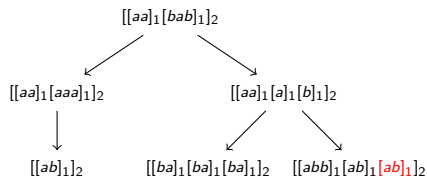
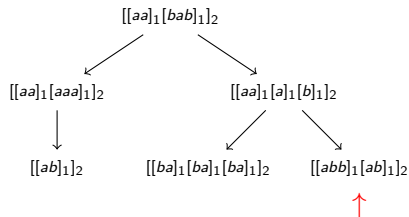
# Basic Operations: Level $i < n$

Copy operations:  $\text{copy}_2$



# Basic Operations: Level $i < n$

Copy operations:  $\text{copy}_2$

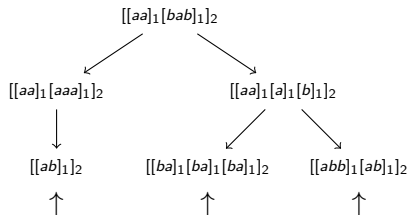


# Basic Operations: Level $n$

Tree copy operations:  $\text{copy}_n^i, \overline{\text{copy}}_n^i$

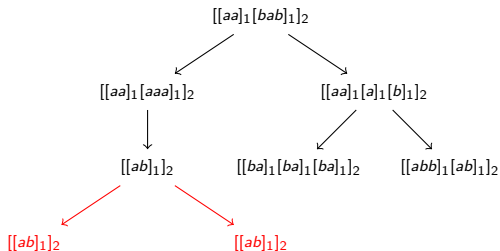
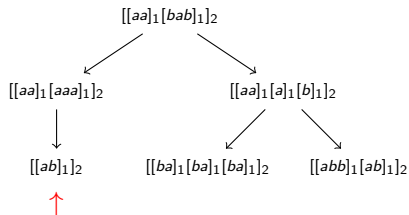
# Basic Operations: Level $n$

Tree copy operations:  $\text{copy}_3^2$



# Basic Operations: Level $n$

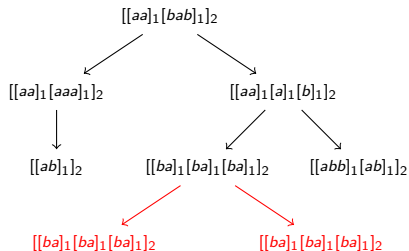
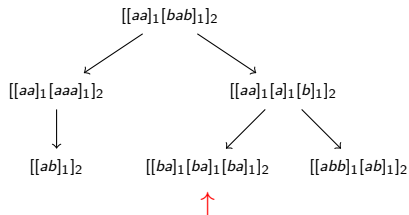
Tree copy operations:  $\text{copy}^2_3$





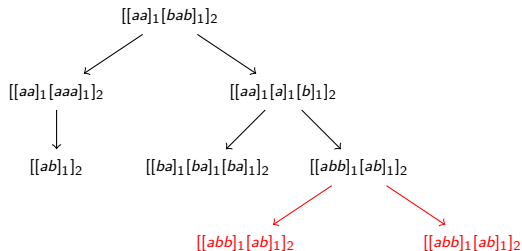
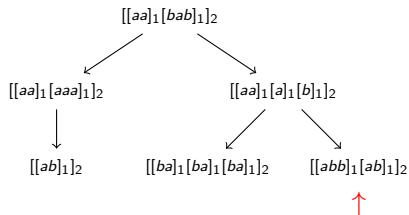
# Basic Operations: Level $n$

Tree copy operations:  $\text{copy}_3^2$



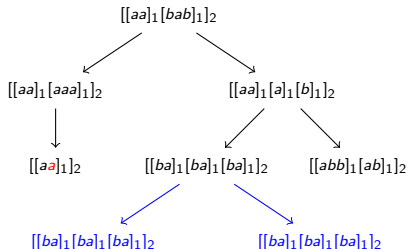
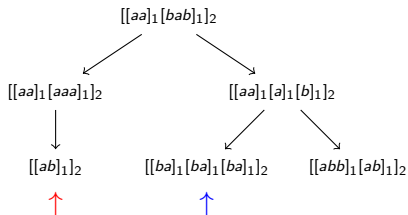
# Basic Operations: Level $n$

Tree copy operations:  $\text{copy}^2_3$



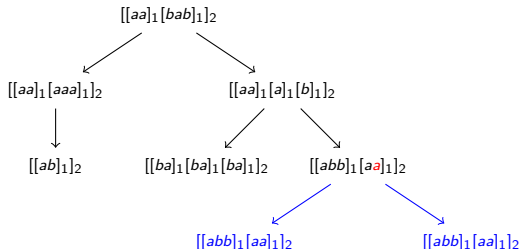
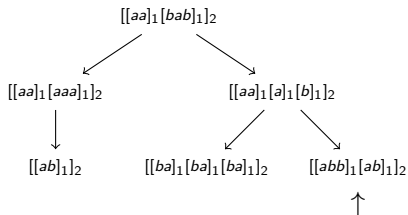
# Composition

$\text{rew}_{b,a} \circ \text{copy}_3^2$ :

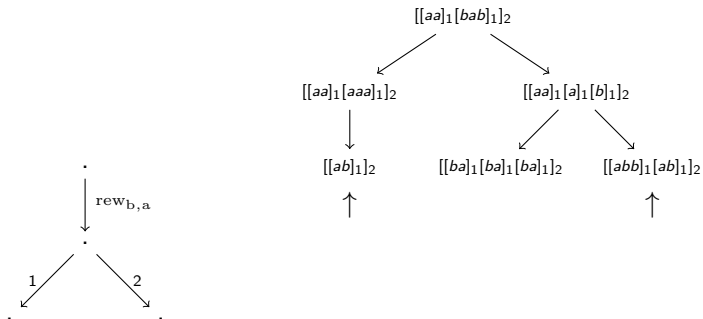


# Composition

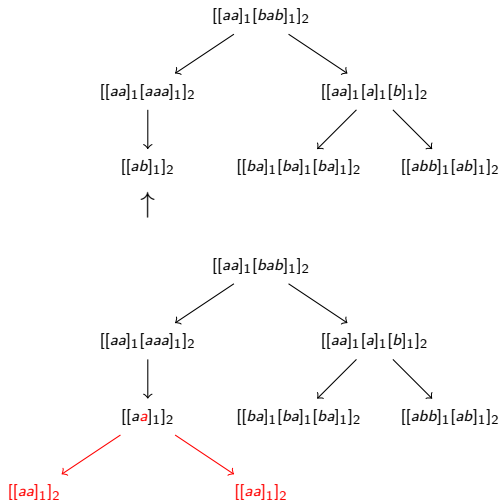
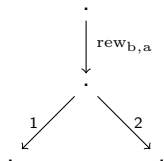
$\text{rew}_{b,a} \circ \text{copy}_3^2$ :



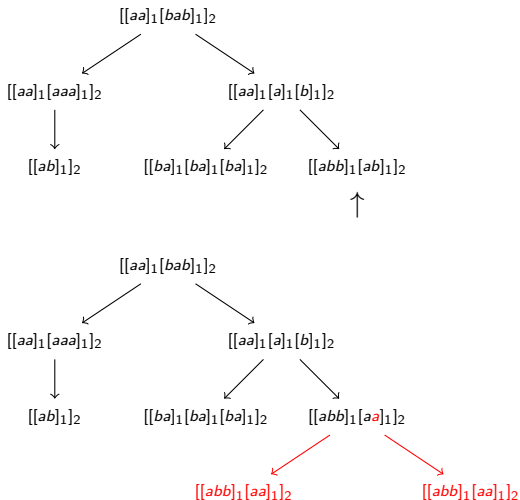
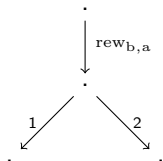
# Compound Operations



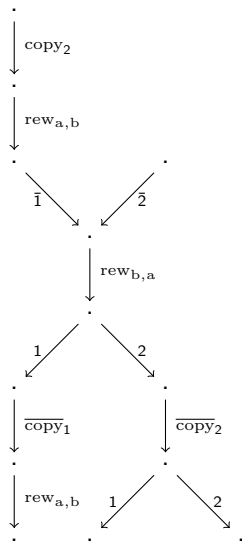
# Compound Operations



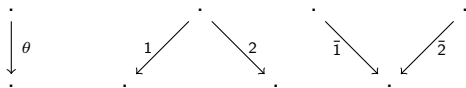
# Compound Operations



# Compound Operations



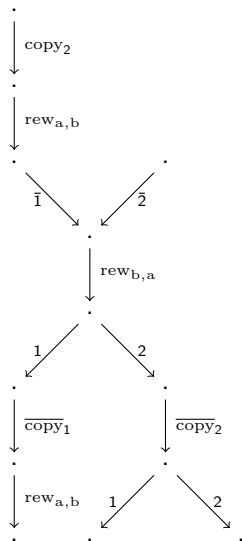
- Characterised by DAGs obtained by concatenations of DAGs whose edges represent basic operations:



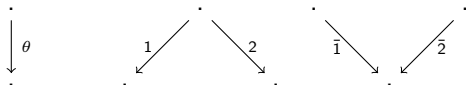
- Only connected operations



# Compound Operations

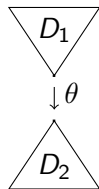


- Characterised by DAGs obtained by concatenations of DAGs whose edges represent basic operations:

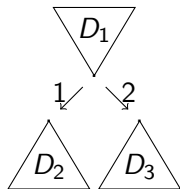


- Only connected operations
- Level 1  $\rightarrow$  ground tree rewriting rules
- Unary trees  $\rightarrow$  finite composition of higher-order pushdown operations

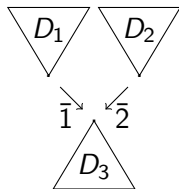
# Compound Operations



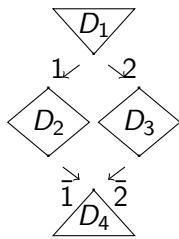
(a)



(b)



(c)



(d)

# Table of Contents

- 1 Introduction
- 2 Stack Trees
- 3 Properties of Stack Trees**
- 4 Perspectives

# Main Result

Given a set of compound operations  $R$ , its rewriting graph  $\mathcal{G}_R$  is:

- $V_{\mathcal{G}_R} = ST_n$
- $E_{\mathcal{G}_R} = \{(t, r, t') \mid r \in R \wedge r(t, t')\}$

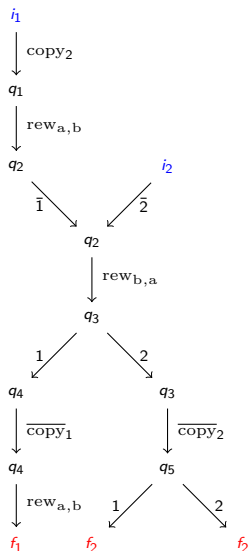
## Theorem

*Given a finite set of compound operations  $R$ , its rewriting graph has a decidable  $FO[\rightarrow^*]$  theory.*

Proof ingredients:

- Notion of recognisability over compound operations
- Finite set interpretation of every stack-tree rewriting graph into a graph with a decidable MSO-theory (the level  $n$  treegraph)

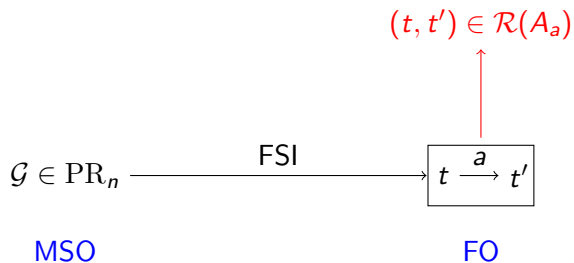
# Operation automata



- Automaton:  $\mathcal{A} = (Q, I, F, \Delta)$ .
- Rules of form:  
 $(q_1, \theta, q_2), (q_1, (q_2, q_3)), ((q_1, q_2), q_3)$ .
- A run is a labelling of a DAG. It is accepting if it is consistent with  $\Delta$ , input nodes are labelled by initial states and output nodes by final states.
- *Rec* the set languages of operation automata is closed by union, intersection and iteration.
- Normalisation notion over automata.

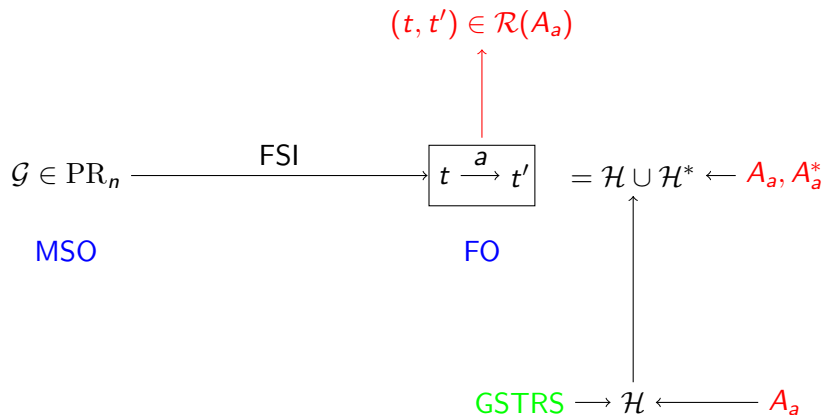
# Idea of the proof

$A_a$  operation automaton for  $a \in \Sigma$ .



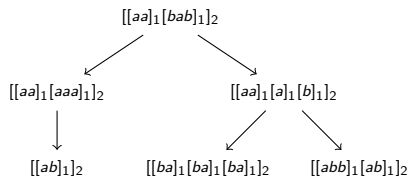
# Idea of the proof

$A_a$  operation automaton for  $a \in \Sigma$ .



# Coding of Stack Trees

We interpret a  $n$ -stack tree as the set of  $n$ -stacks representing the paths from the root to each leaf, in which we added the direction of the path and the number of sons of each node on the path.

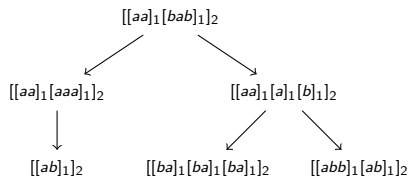


$[[[aa]1[bab21]1]2][[aa]1[aaa11]1]2][[ab]1]2]_3$



# Coding of Stack Trees

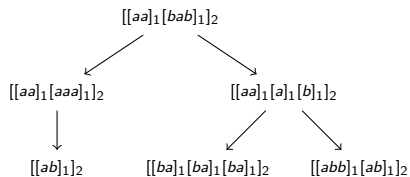
We interpret a  $n$ -stack tree as the set of  $n$ -stacks representing the paths from the root to each leaf, in which we added the direction of the path and the number of sons of each node on the path.



$[[[aa]1[bab22]1]2[[aa]1[a]1[b21]1]2[[ba]1[ba]1[ba]1]2]3$

# Coding of Stack Trees

We interpret a  $n$ -stack tree as the set of  $n$ -stacks representing the paths from the root to each leaf, in which we added the direction of the path and the number of sons of each node on the path.



$[[[aa]_1[bab]_{22}]_1]_2 [[aa]_1[a]_1[b]_{22}]_1]_2 [[abb]_1[ab]_1]_2]_3$

# Table of Contents

- 1 Introduction
- 2 Stack Trees
- 3 Properties of Stack Trees
- 4 Perspectives

# Languages and Strictness

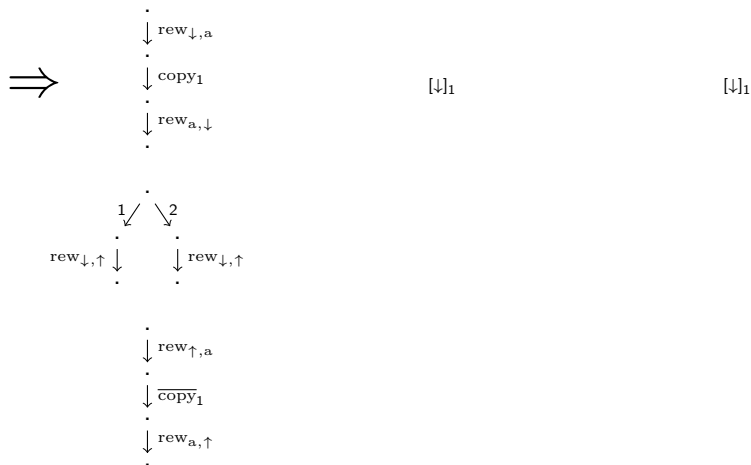
Configuration graphs of HOPDA <i>Indexed Languages</i>	Ground stack tree rewriting graphs ?	Tree automatic of order $n$ ?
Configuration graphs of PDA <i>Context-free Languages</i>	Ground tree rewriting graphs ?	Tree automatic <i>Languages of E-time</i>
Strict hierarchy	?	Strict hierarchy

# Languages and Strictness

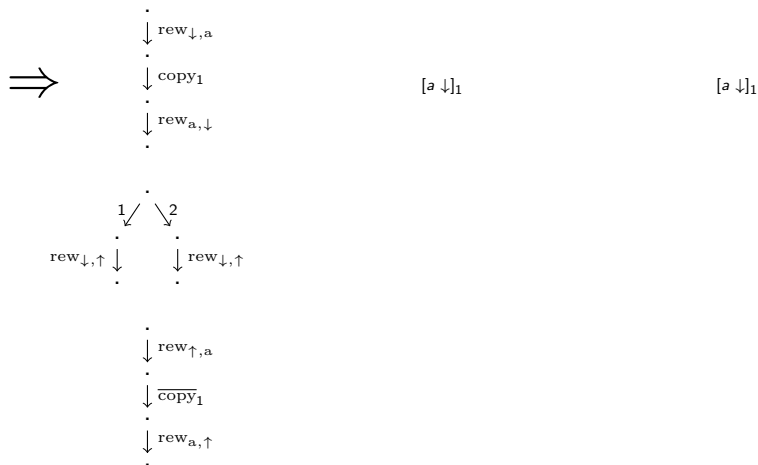
Configuration graphs of HOPDA <i>Indexed Languages</i>	Ground stack tree rewriting graphs ?	Tree automatic of order $n$ ?
Configuration graphs of PDA <i>Context-free Languages</i>	Ground tree rewriting graphs ?	Tree automatic <i>Languages of E-time</i>
Strict hierarchy	?	Strict hierarchy

And the equational representation?

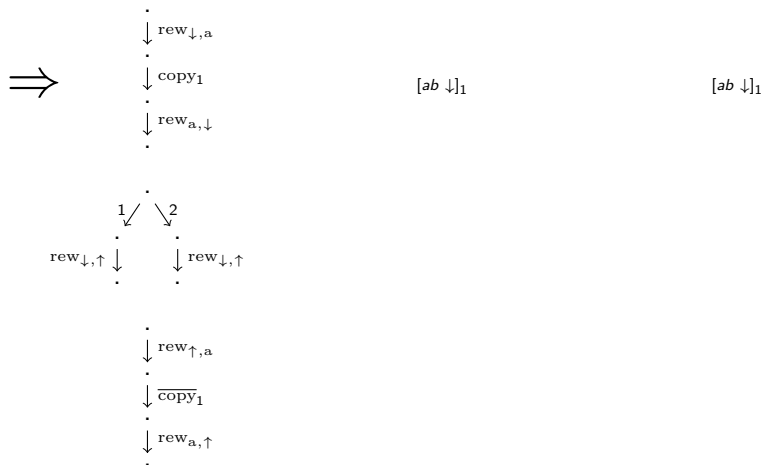
# Exemple of a language: $\{u \sqcup u \mid u \in \Sigma^*\}$



# Exemple of a language: $\{u \sqcup u \mid u \in \Sigma^*\}$

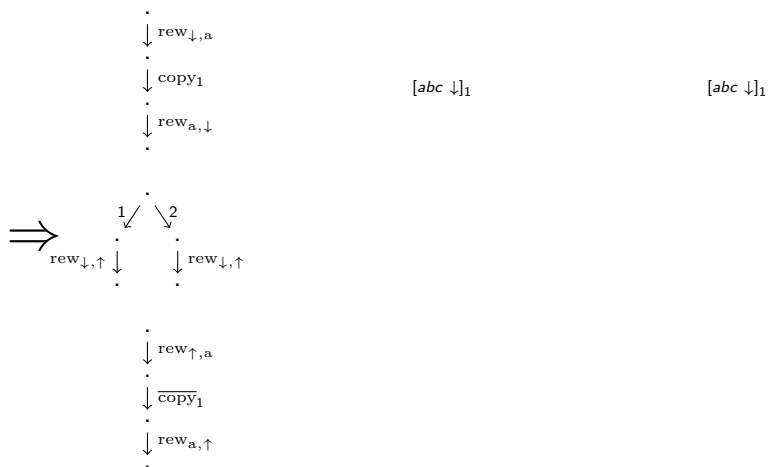


# Exemple of a language: $\{u \sqcup u \mid u \in \Sigma^*\}$

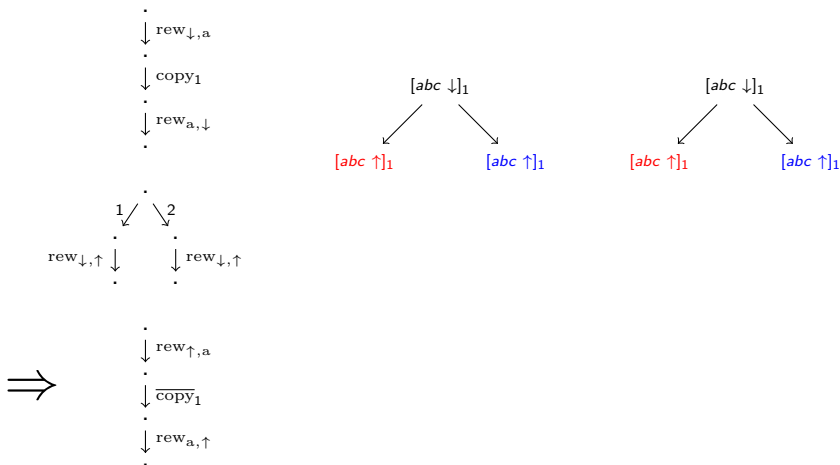




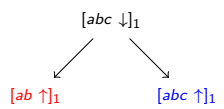
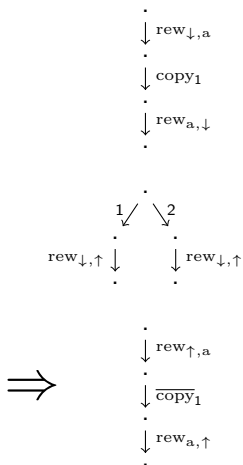
# Exemple of a language: $\{u \sqcup u \mid u \in \Sigma^*\}$



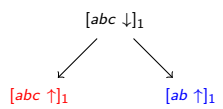
# Exemple of a language: $\{u \sqcup u \mid u \in \Sigma^*\}$



# Exemple of a language: $\{u \sqcup u \mid u \in \Sigma^*\}$

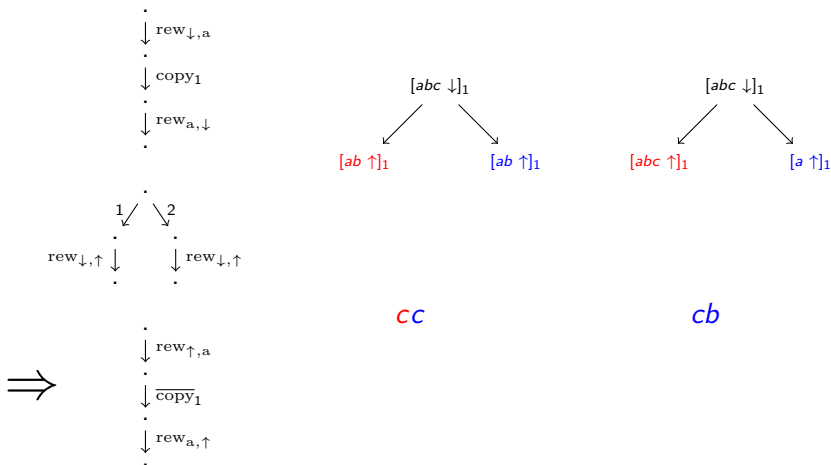


C

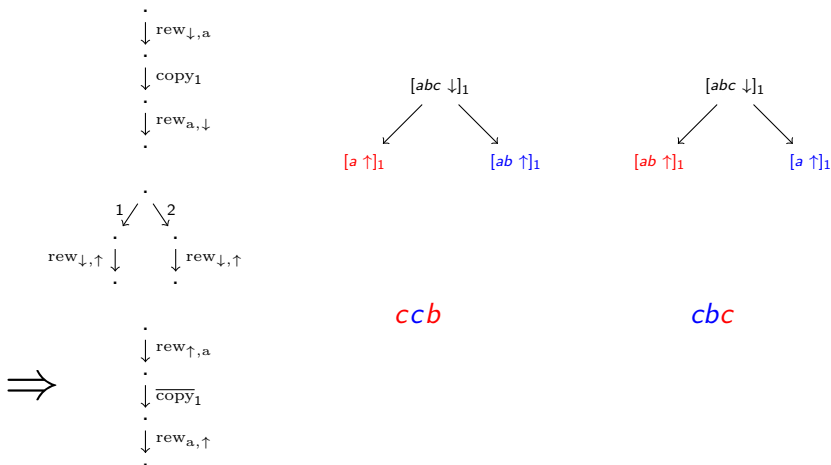


C

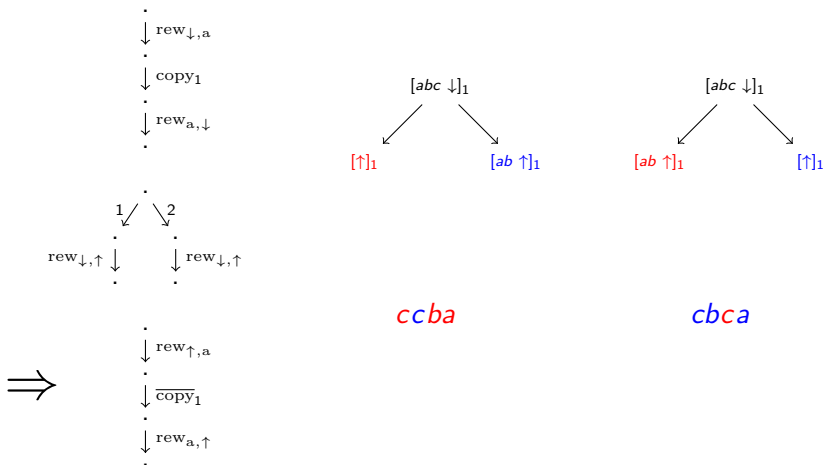
# Exemple of a language: $\{u \sqcup u \mid u \in \Sigma^*\}$



# Exemple of a language: $\{u \sqcup u \mid u \in \Sigma^*\}$

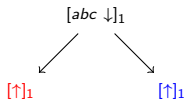
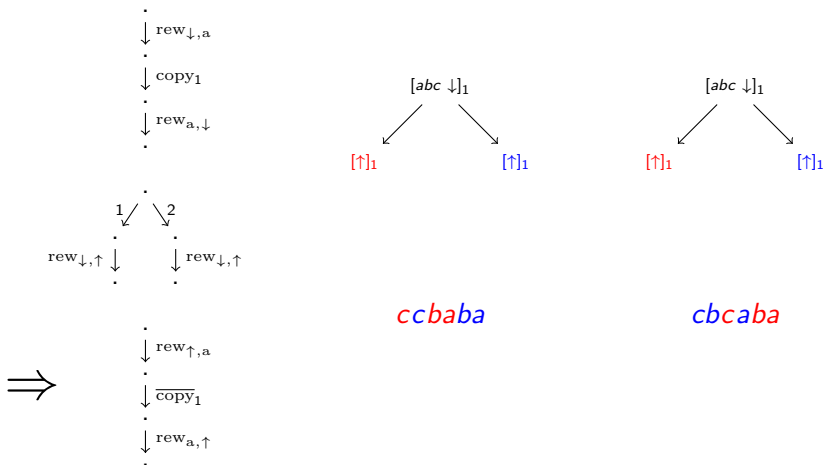


# Exemple of a language: $\{u \sqcup u \mid u \in \Sigma^*\}$

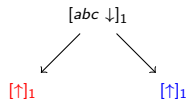




# Exemple of a language: $\{u \sqcup u \mid u \in \Sigma^*\}$



*ccbaba*



*cbcaba*



# Higher-Order Trees

