Minimizing Tree Automata for Unranked Trees

Wim Martens   Joachim Niehren

What and Why?

To study the minimization problem for
deterministic automata over unranked trees.

- Bottom-up deterministic: theoretical interest.
  E.g. do results from
  - deterministic automata on strings
  - bottom-up deterministic automata on ranked trees

  carry over naturally?
- Top-down deterministic: XML schema languages:
  - XML Schema Definitions
  - 1-pass preorder typeable schemas

  Minimization $\equiv$ optimizing the schema.

Goals for Minimization

Requirements:

1. Minimization should be efficient (PTIME)
2. Unique minimal automata would be nice (up to isomorphism)
3. Minimal automata should be small

Minimization

Minimization:
Given an automaton A, integer k.

Does there exist an automaton B such that

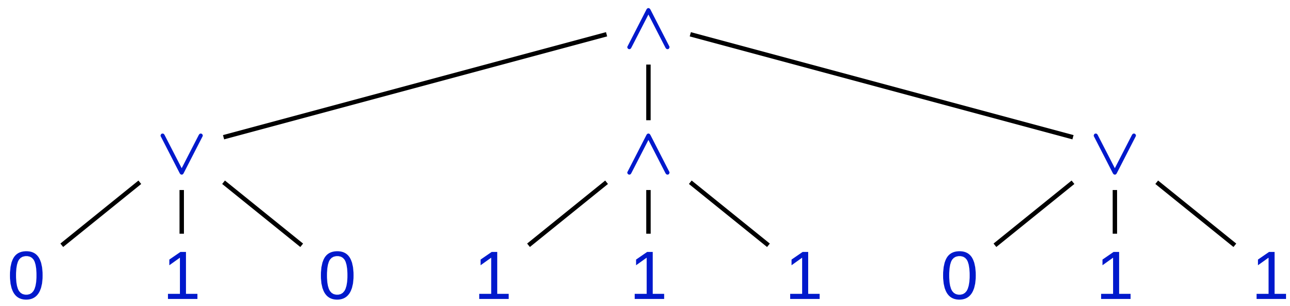
- B is equivalent to A
- the size of B is $\leq k$

Overview

- Unranked Tree Automata (UTAs)
- Minimizing UTAs
- Small Survey on Bottom-up Deterministic TA
- Top-Down Determinism

UTAs - Example

Evaluate Boolean expressions:

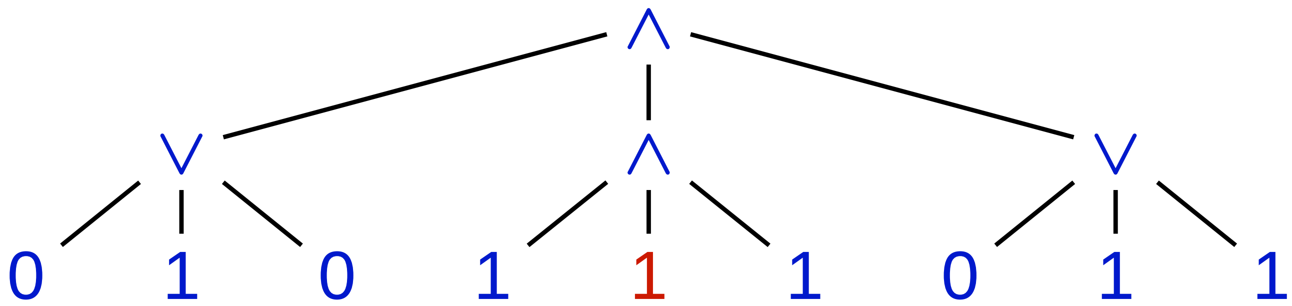


States: $\{t, f\}$

	label		state		language
δ (	1	,	t	) =	ε
δ (	0	,	f	) =	ε
δ (	$\wedge$	,	t	) =	tt^*
δ (	$\wedge$	,	f	) =	$(f\|t)^* f (f\|t)^*$
δ (	$\vee$	,	t	) =	$(f\|t)^* t (f\|t)^*$
δ (	$\vee$	,	f	) =	ff^*

UTAs - Example

Evaluate Boolean expressions:

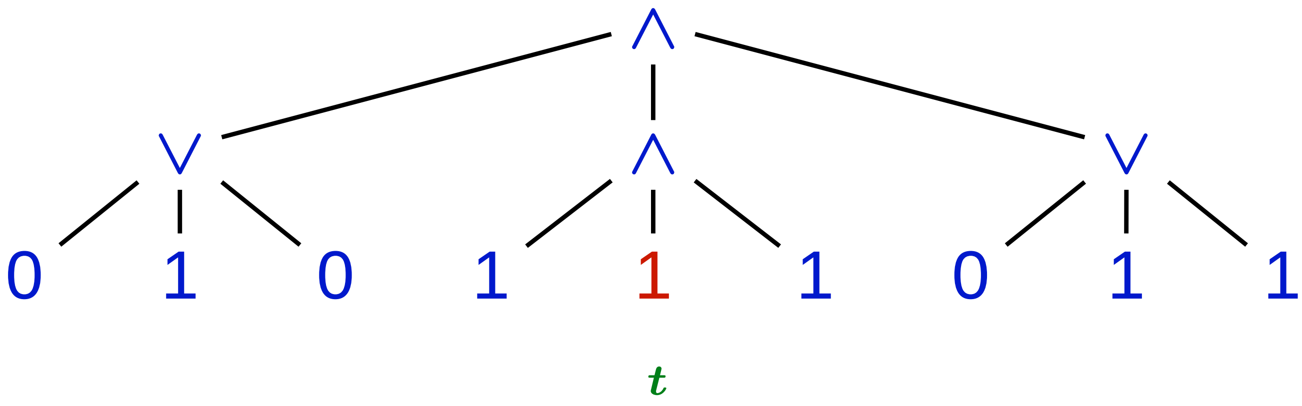


States: $\{t, f\}$

	label	state	language
$\delta($	1	$, \; t$	$) =$ ε
$\delta($	0	$, \; f$	$) =$ ε
$\delta($	$\wedge$	$, \; t$	$) =$ tt^*
$\delta($	$\wedge$	$, \; f$	$) =$ $(f\|t)^* f (f\|t)^*$
$\delta($	$\vee$	$, \; t$	$) =$ $(f\|t)^* t (f\|t)^*$
$\delta($	$\vee$	$, \; f$	$) =$ ff^*

UTAs - Example

Evaluate Boolean expressions:

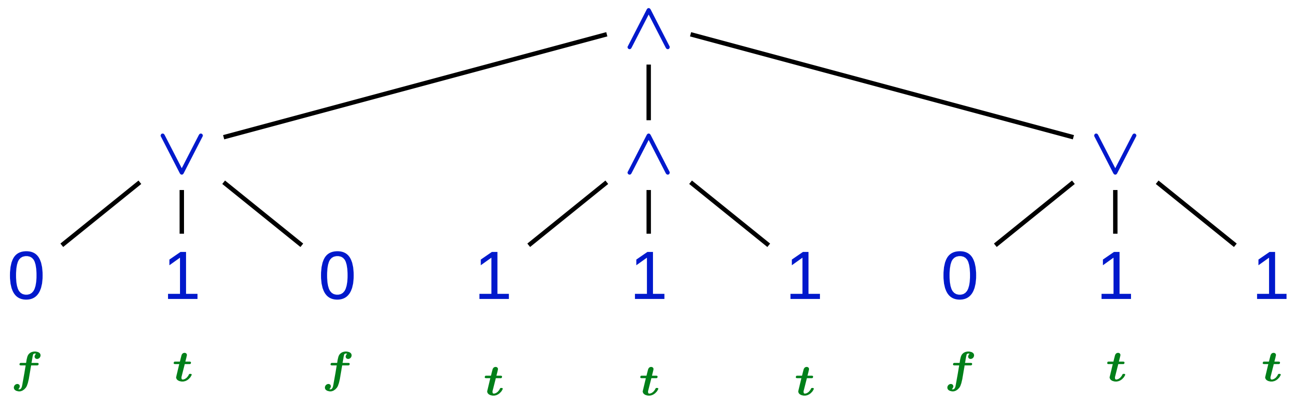


States: $\{t, f\}$

	label		state		language
$\delta($	1	,	t	$) =$	ε
$\delta($	0	,	f	$) =$	ε
$\delta($	$\wedge$	,	t	$) =$	tt^*
$\delta($	$\wedge$	,	f	$) =$	$(f\|t)^* f (f\|t)^*$
$\delta($	$\vee$	,	t	$) =$	$(f\|t)^* t (f\|t)^*$
$\delta($	$\vee$	,	f	$) =$	ff^*

UTAs - Example

Evaluate Boolean expressions:

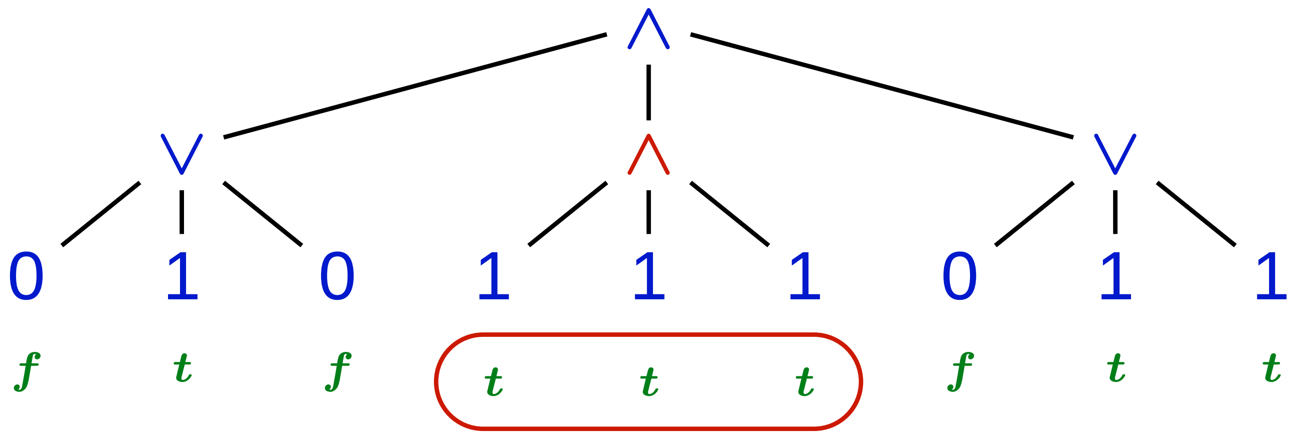


States: $\{t, f\}$

	label		state		language
$\delta($	1	,	t	$) =$	ε
$\delta($	0	,	f	$) =$	ε
$\delta($	$\wedge$	,	t	$) =$	tt^*
$\delta($	$\wedge$	,	f	$) =$	$(f\|t)^* f (f\|t)^*$
$\delta($	$\vee$	,	t	$) =$	$(f\|t)^* t (f\|t)^*$
$\delta($	$\vee$	,	f	$) =$	ff^*

UTAs - Example

Evaluate Boolean expressions:

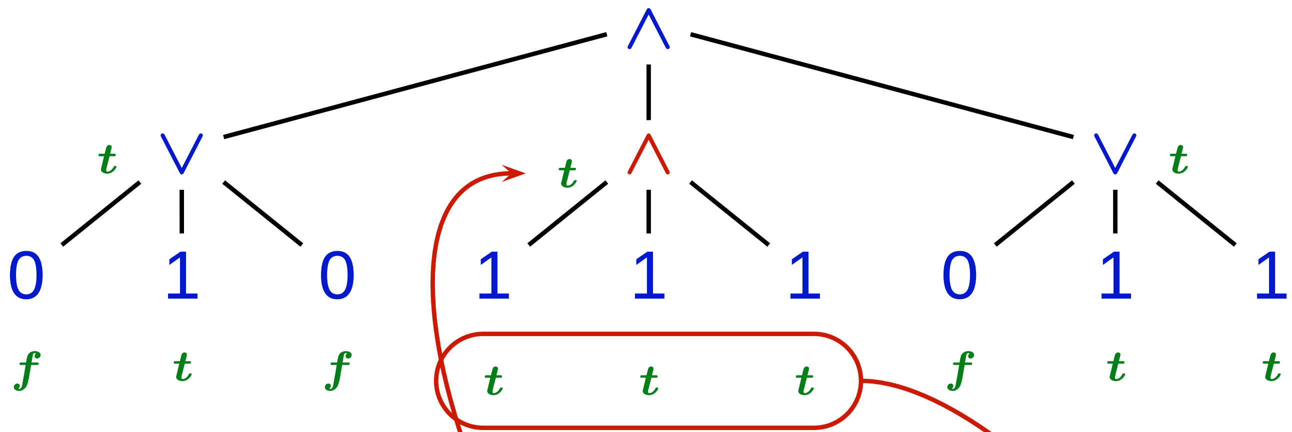


States: $\{t, f\}$

	label		state		language
$\delta($	1	,	t	$) =$	ε
$\delta($	0	,	f	$) =$	ε
$\delta($	$\wedge$	,	t	$) =$	tt^*
$\delta($	$\wedge$	,	f	$) =$	$(f\|t)^* f (f\|t)^*$
$\delta($	$\vee$	,	t	$) =$	$(f\|t)^* t (f\|t)^*$
$\delta($	$\vee$	,	f	$) =$	ff^*

UTAs - Example

Evaluate Boolean expressions:

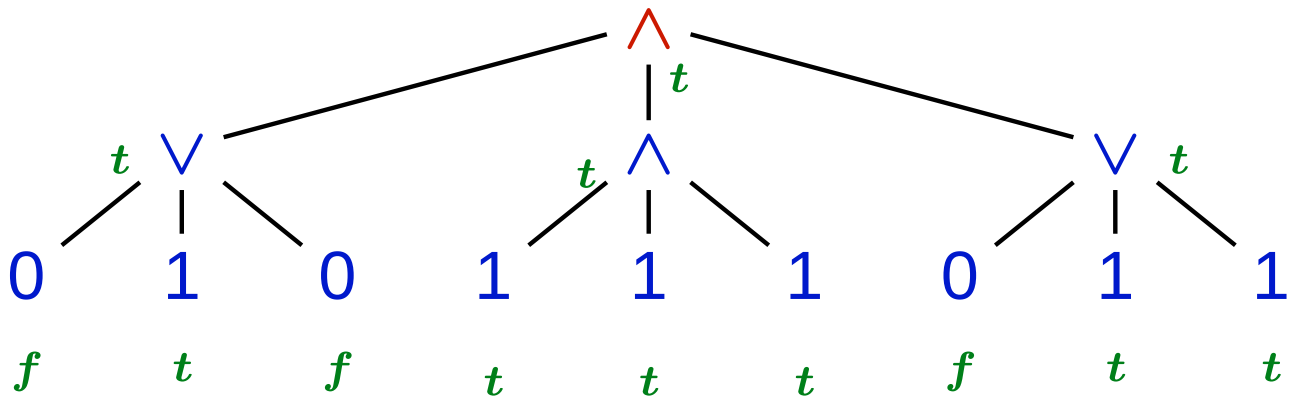


States: $\{t, f\}$

	label		state		language
δ (	1	,	t	) =	ε
δ (	0	,	f	) =	ε
δ (	$\wedge$	,	t	) =	tt^*
δ (	$\wedge$	,	f	) =	$(f\|t)^* f (f\|t)^*$
δ (	$\vee$	,	t	) =	$(f\|t)^* t (f\|t)^*$
δ (	$\vee$	,	f	) =	ff^*

UTAs - Example

Evaluate Boolean expressions:



States: $\{t, f\}$

	label		state		language
δ (	1	,	t	) =	ε
δ (	0	,	f	) =	ε
δ (	$\wedge$	,	t	) =	tt^*
δ (	$\wedge$	,	f	) =	$(f\|t)^* f (f\|t)^*$
δ (	$\vee$	,	t	) =	$(f\|t)^* t (f\|t)^*$
δ (	$\vee$	,	f	) =	ff^*

UTAs by Example

Bottom-up Determinism [BMW 1999]:

	label	state		language
δ (	$\wedge$,	t	) =	tt^*
δ (	$\wedge$,	f	) =	$(f\|t)^* f (f\|t)^*$

If the labels are the same, then the languages are disjoint

Overview

- Unranked Tree Automata (UTAs)
- Minimizing UTAs
- Small Survey on Bottom-up Deterministic TA
- Top-Down Determinism

Minimizing UTAs

- What is the size of a UTA?

Minimizing UTAs

- What is the size of a UTA?

  Take states + internal languages into account

Minimizing UTAs

- What is the size of a UTA?
  Take states + internal languages into account
- Representation of internal languages left open
  NFA, DFA, regular expression, etc.

Minimizing UTAs

- What is the size of a UTA?
  Take states + internal languages into account
- Representation of internal languages left open
  NFA, DFA, regular expression, etc.

Minimizing NFAs, regular expressions is PSPACE-complete

Minimizing UTAs

- What is the size of a UTA?
  Take states + internal languages into account
- Representation of internal languages left open
  NFA, DFA, regular expression, etc.

Minimizing NFAs, regular expressions is PSPACE-complete

As we want efficient minimization,
we represent internal languages by DFAs

Then, size = |states| + $\sum$|states internal DFAs|

Minimizing DUTAs

DUTA: Bottom-up deterministic UTA
with DFAs for internal languages

Minimizing DUTAs

DUTA: Bottom-up deterministic UTA
with DFAs for internal languages

Unfortunately,

Theorem:

- Minimizing DUTAs is NP-complete
- Minimal DUTAs are not unique

This is not what one expects from deterministic automata!

Minimizing DUTAs

DUTA: Bottom-up deterministic UTA
with DFAs for internal languages

Unfortunately,

Theorem:

- Minimizing DUTAs is NP-complete
- Minimal DUTAs are not unique

This is not what one expects from deterministic automata!

- Why NP-hard? / Why not unique?

Minimizing DUTAs

DUTA: Bottom-up deterministic UTA
with DFAs for internal languages

Unfortunately,

Theorem:

- Minimizing DUTAs is NP-complete
- Minimal DUTAs are not unique

This is not what one expects from deterministic automata!

- Why NP-hard? / Why not unique?
  Crux: internal languages can be represented by

  a disjoint union of DFAs

Minimizing DUTAs

Internal languages can be represented by
a disjoint union of DFAs

	label		state		language
δ (	$\wedge$	,	t	) =	tt^*
δ (	$\wedge$	,	f	) =	$(f\|t)^* f (f\|t)^*$

Minimizing DUTAs

Internal languages can be represented by
a disjoint union of DFAs

	label	state		language
$\delta($	$\wedge$	, t	$) =$	tt^*
$\delta($	$\wedge$	, f	$) =$	$(f\|t)^* f (f\|t)^*$

Can be split up into:
even number of t's / odd number of t's

Minimizing DUTAs

Internal languages can be represented by
a disjoint union of DFAs

Lemma:

- Minimizing disjoint unions of DFAs is NP-complete
- Minimal disjoint unions of DFAs are not unique

NP hardness strengthens some results in [Jiang, Ravikumar 1993], [Malcher 2004]

Minimizing DUTAs

Internal languages can be represented by
a disjoint union of DFAs

Lemma:

- Minimizing disjoint unions of DFAs is NP-complete
- Minimal disjoint unions of DFAs are not unique

NP hardness strengthens some results in [Jiang, Ravikumar 1993], [Malcher 2004]

Why is minimization in NP?

Minimizing DUTAs

Internal languages can be represented by
a disjoint union of DFAs

Lemma:

- Minimizing disjoint unions of DFAs is NP-complete
- Minimal disjoint unions of DFAs are not unique

NP hardness strengthens some results in [Jiang, Ravikumar 1993], [Malcher 2004]

Why is minimization in NP?

Guess minimal automaton + check equivalence

Overview

- Unranked Tree Automata (UTAs)
- Minimizing UTAs
- Small Survey on Bottom-up Deterministic TA
- Top-Down Determinism

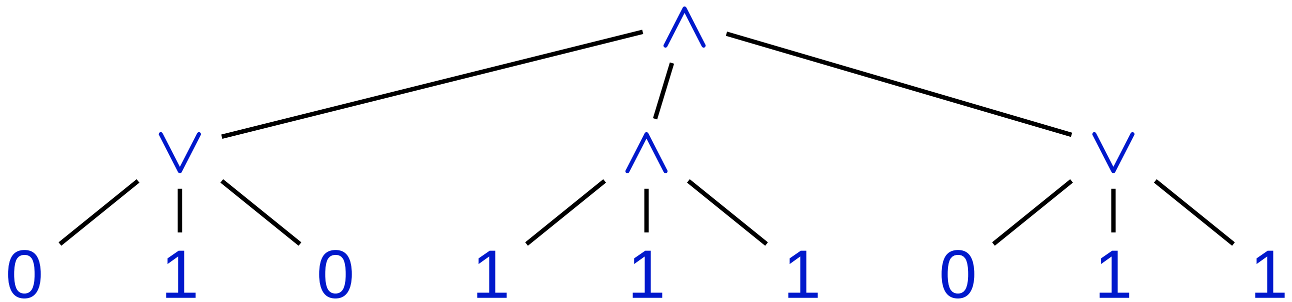
Other Bottom-up Deterministic TA

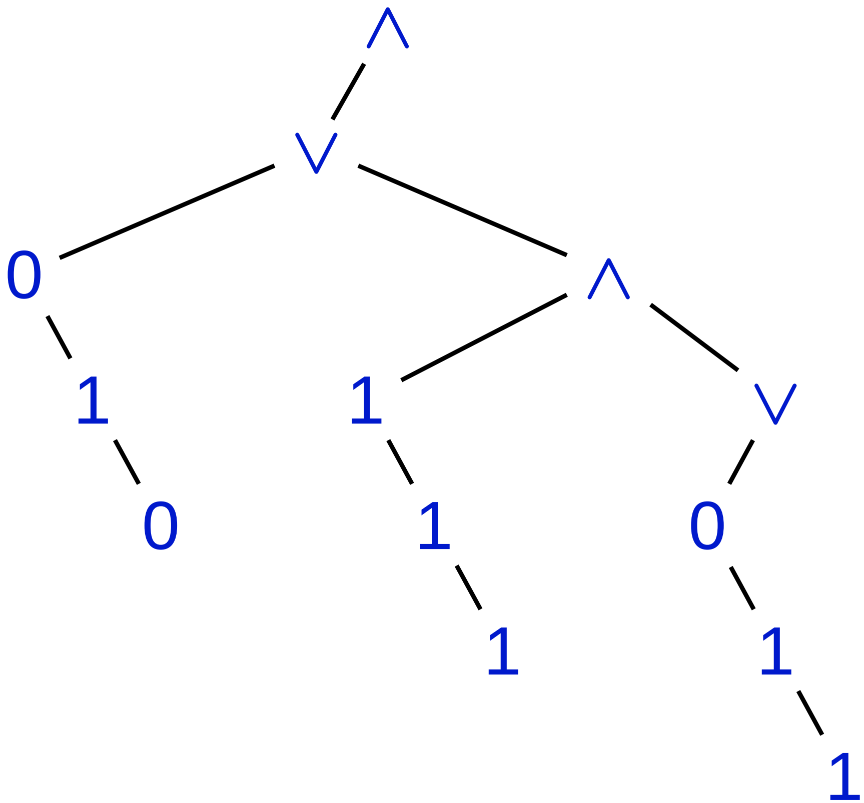
- Automata over FCNS encoding, see e.g. [Frick,Grohe,Koch 2003]
- Parallel UTAs [Raeymaekers 2004, Cristau, Löding, Thomas 2005]
- Stepwise automata [Carmen,Niehren,Tommasi 2004]

Requirements:

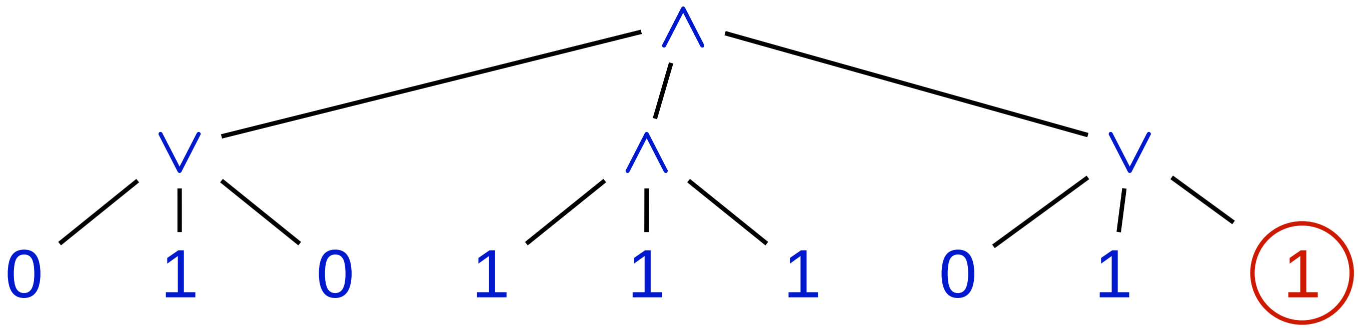
1. Minimization should be efficient –OK
2. Minimal automata should be unique –OK
3. Minimal automata should be small

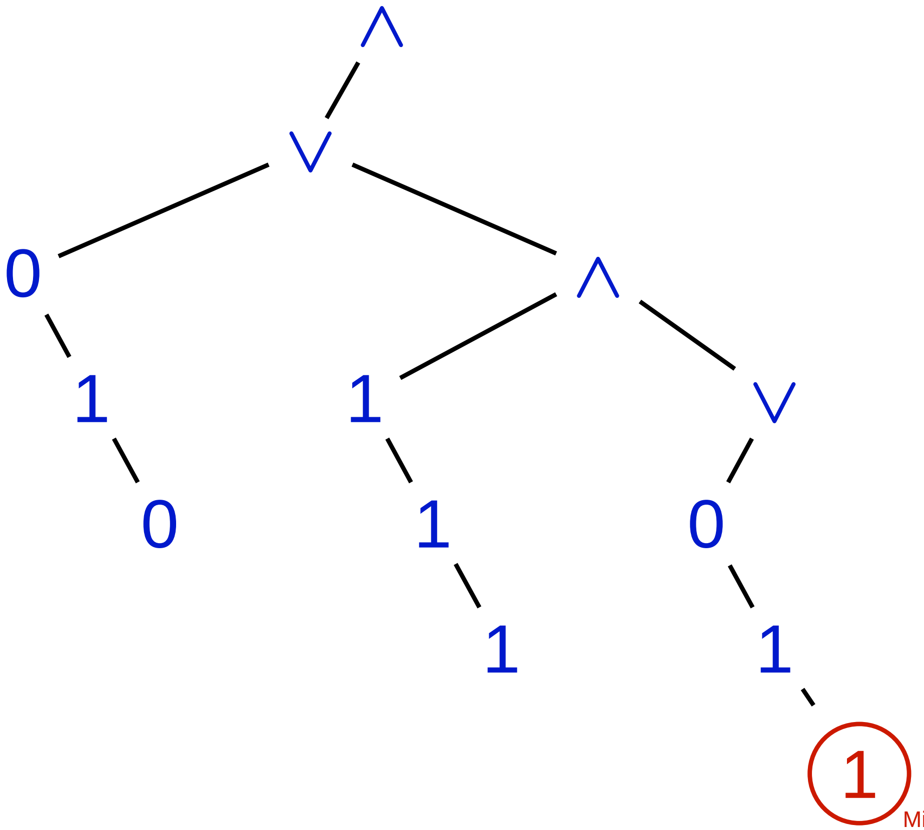
FCNS-encoding



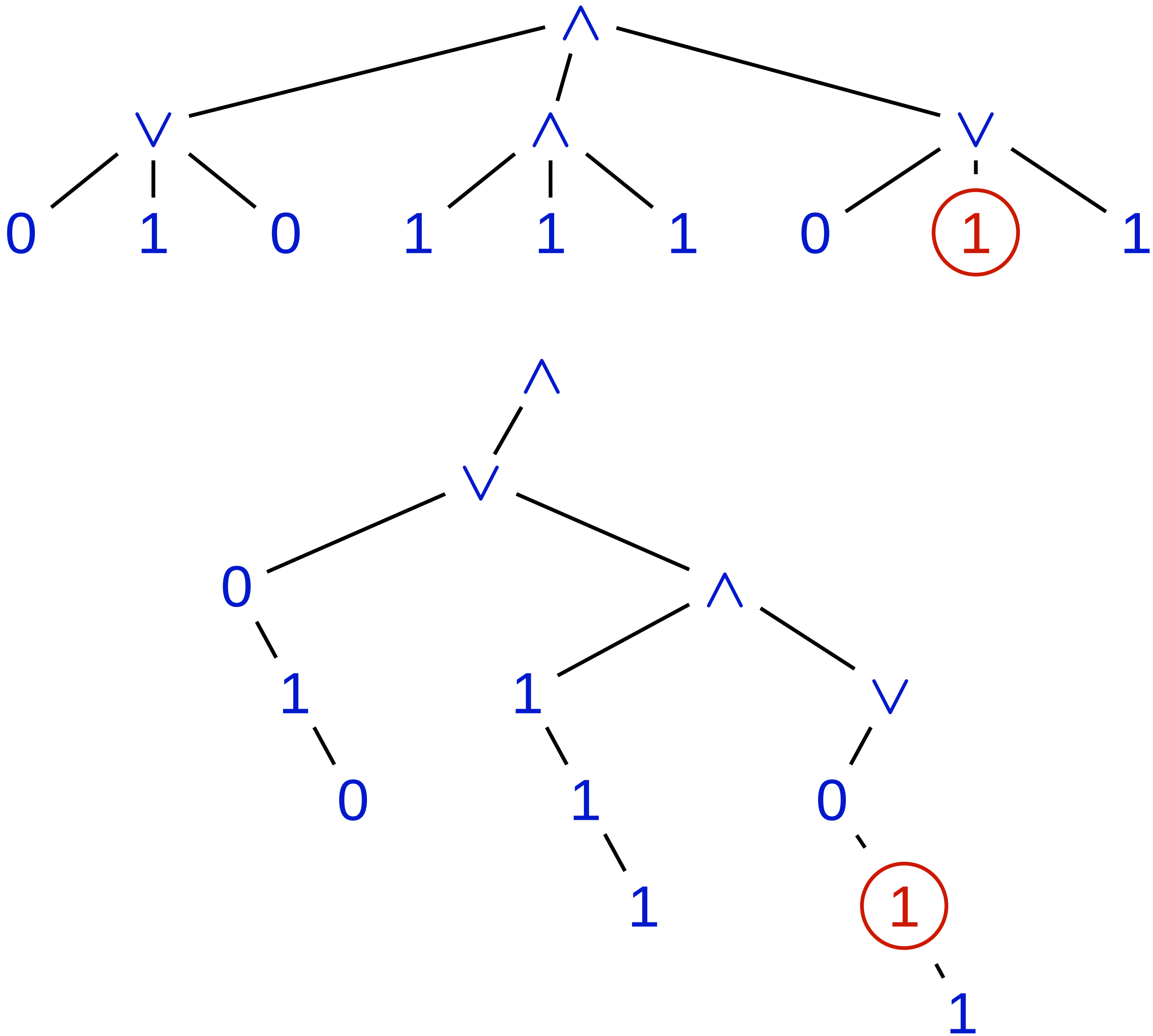


FCNS-encoding

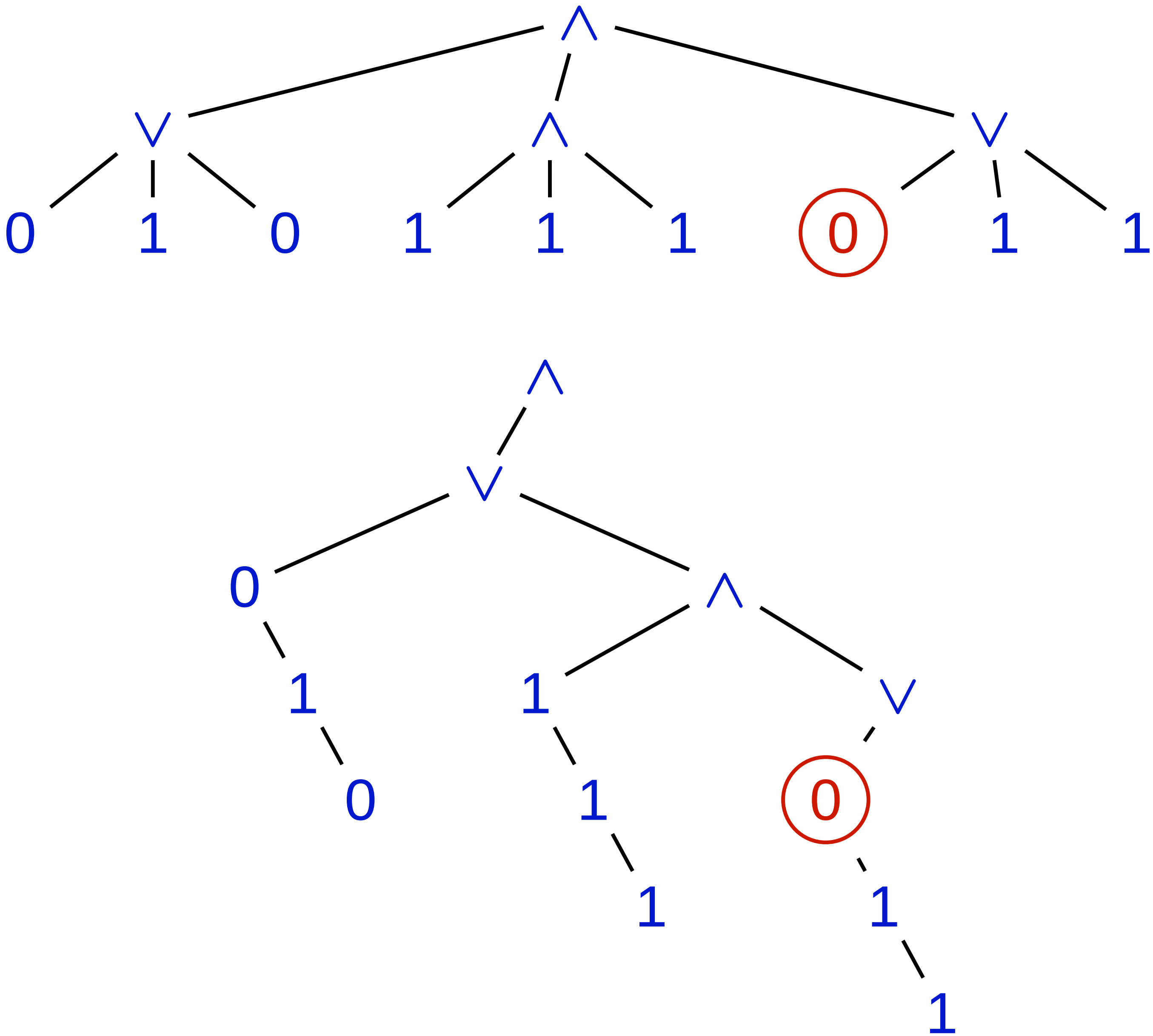




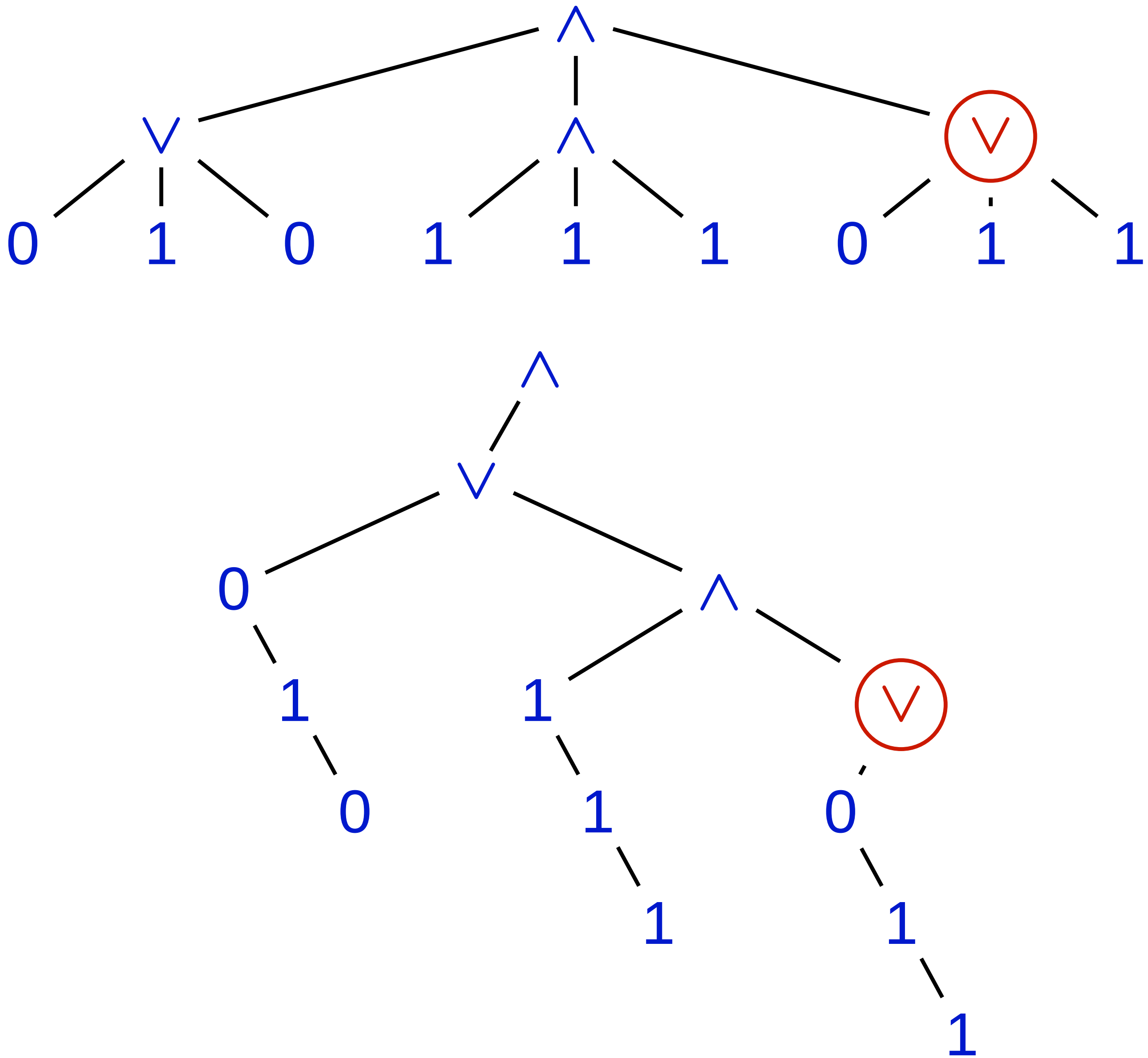
FCNS-encoding



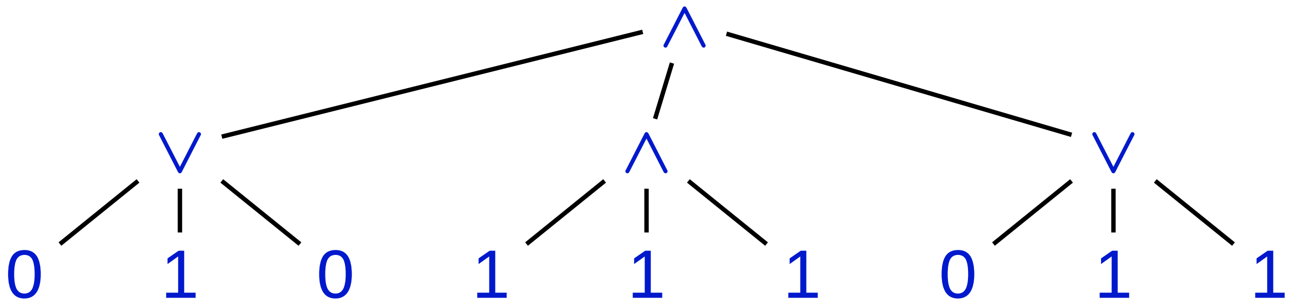
FCNS-encoding



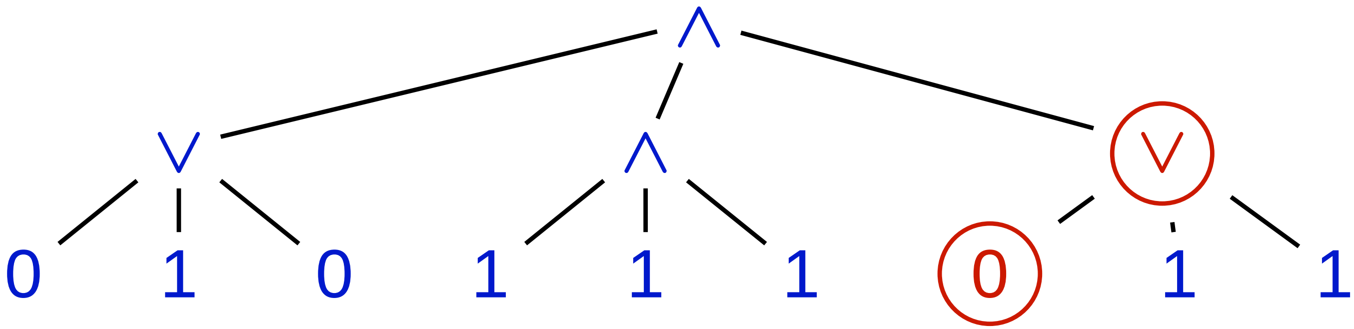
FCNS-encoding



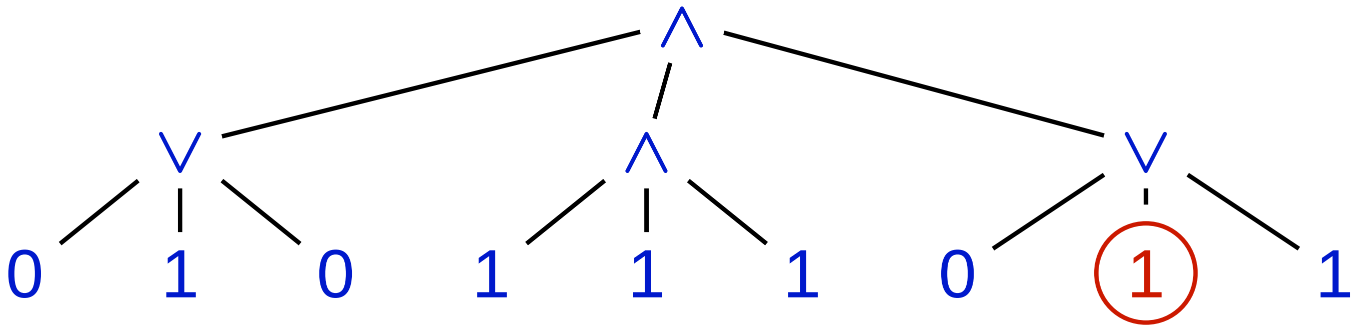
Parallel UTAs and Stepwise Automata



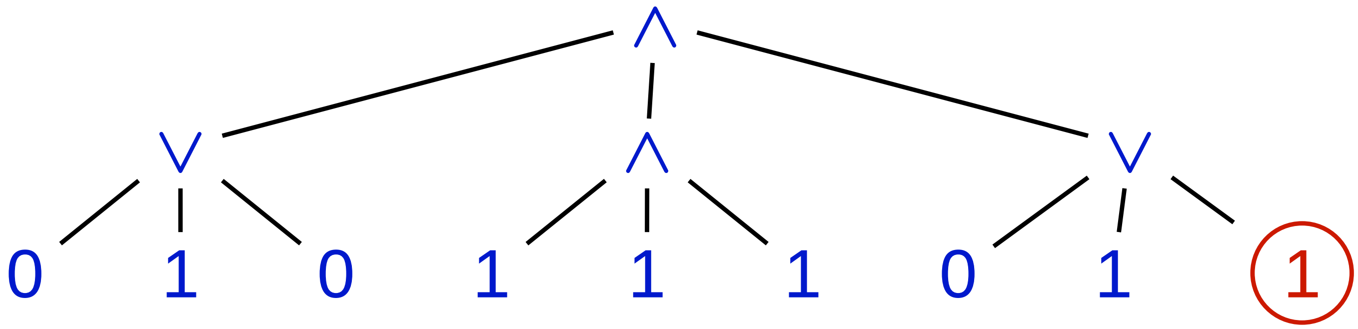
Parallel UTAs and Stepwise Automata



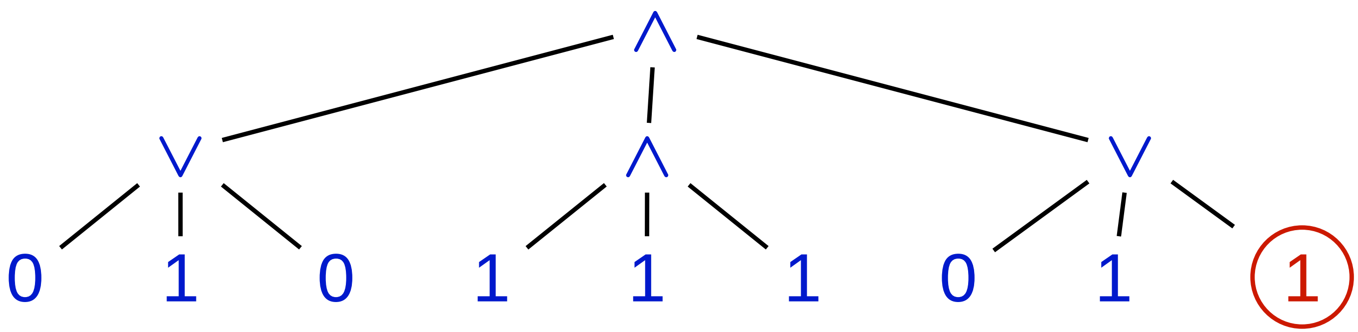
Parallel UTAs and Stepwise Automata



Parallel UTAs and Stepwise Automata



Parallel UTAs and Stepwise Automata



Differences:

- Difference in representation: stepwise automata can be quadratically smaller
- Stepwise automata correspond to ranked automata through an encoding (currying)

Size Comparison

Theorem:
Minimal stepwise tree automata are

- quadratically smaller than minimal Parallel UTAs
- exponentially smaller than minimal FCNS-Automata

in general

Conversely, minimal stepwise automata are never larger than the corresponding minimal Parallel UTA or FCNS-automaton for the same tree language.

Overview

- Unranked Tree Automata (UTAs)
- Minimizing UTAs
- Small Survey on Bottom-up Deterministic TA
- Top-Down Determinism

Restrained Competition

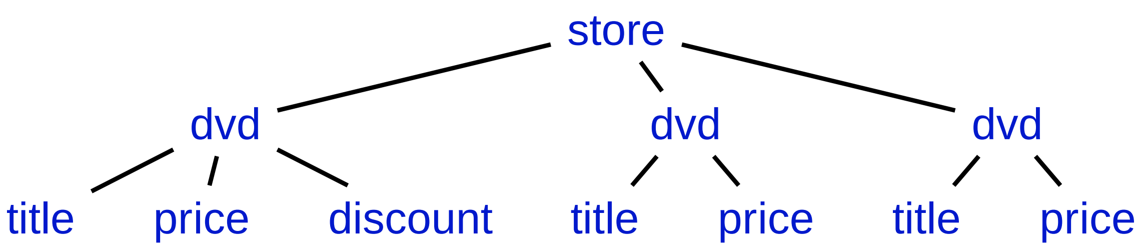
In terms of Extended DTDs:

$$\begin{array}{lcl} \text{store} & \longrightarrow & \text{dvd}^2\ (\text{dvd}^1)^* \\ \text{dvd}^1 & \longrightarrow & \text{title price} \\ \text{dvd}^2 & \longrightarrow & \text{title price discount} \end{array}$$

Restrained Competition

In terms of Extended DTDs:

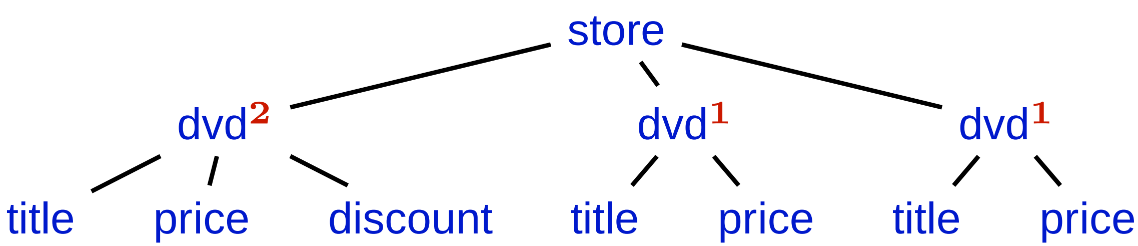
$$\begin{array}{lcl} \text{store} & \longrightarrow & \text{dvd}^2\ (\text{dvd}^1)^* \\ \text{dvd}^1 & \longrightarrow & \text{title price} \\ \text{dvd}^2 & \longrightarrow & \text{title price discount} \end{array}$$



Restrained Competition

In terms of Extended DTDs:

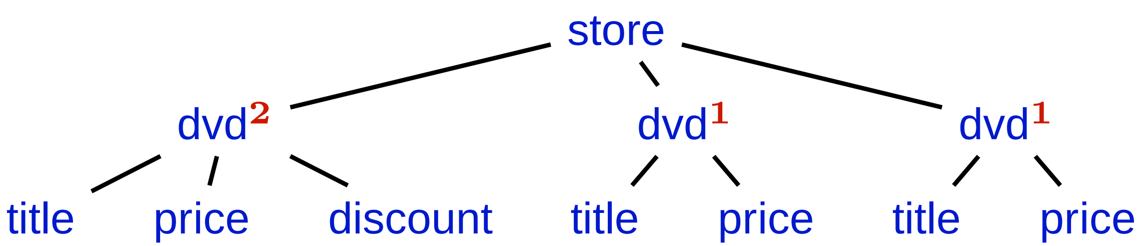
$$\begin{array}{lcl} \text{store} & \longrightarrow & \text{dvd}^2\ (\text{dvd}^1)^* \\ \text{dvd}^1 & \longrightarrow & \text{title price} \\ \text{dvd}^2 & \longrightarrow & \text{title price discount} \end{array}$$



Restrained Competition

In terms of Extended DTDs:

$$
\begin{array}{lcl}
\text{store} & \longrightarrow & \text{dvd}^2\ (\text{dvd}^1)^* \\
\text{dvd}^1 & \longrightarrow & \text{title price} \\
\text{dvd}^2 & \longrightarrow & \text{title price discount}
\end{array}
$$



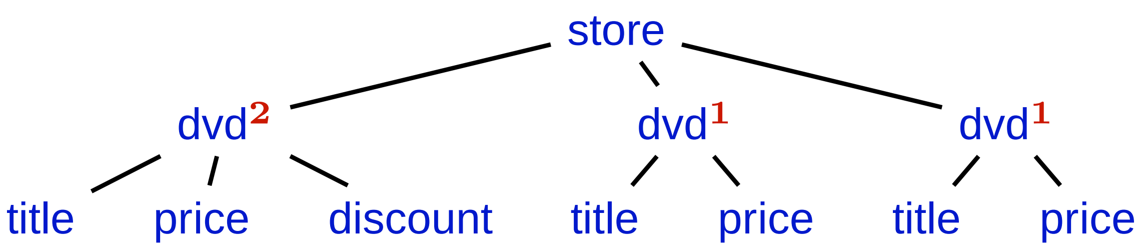
Restrained Competition:

When reading the string from left to right, type of every node should be clear.

Restrained Competition

In terms of Extended DTDs:

$$\begin{array}{lcl} \text{store} & \longrightarrow & \text{dvd}^2\ (\text{dvd}^1)^* \\ \text{dvd}^1 & \longrightarrow & \text{title price} \\ \text{dvd}^2 & \longrightarrow & \text{title price discount} \end{array}$$



Restrained Competition:
When reading the string from left to right, type of every node should be clear.

Examples:

- Single-type extended DTDs (i.e. XML Schema)
- 1-pass preorder typeable EDTDs ($=$ Restrained competition extended DTDs!)

Top-Down Determinism

When horizontal languages are represented by DFAs,

Theorem:

- Restraine Competition DTDs can be minized in PTIME
- Minimal restrained competition EDTDs are unique (up to isomorphism)

Minimization algorithm preserves single-type property.

Corollary:

- Single-type EDTDs can be minized in PTIME
- Minimal single-type EDTDs are unique (up to isomorphism)