PODS
New York, USA
26-jun-2013

XPath with transitive axes

Diego Figueira
University of Edinburgh

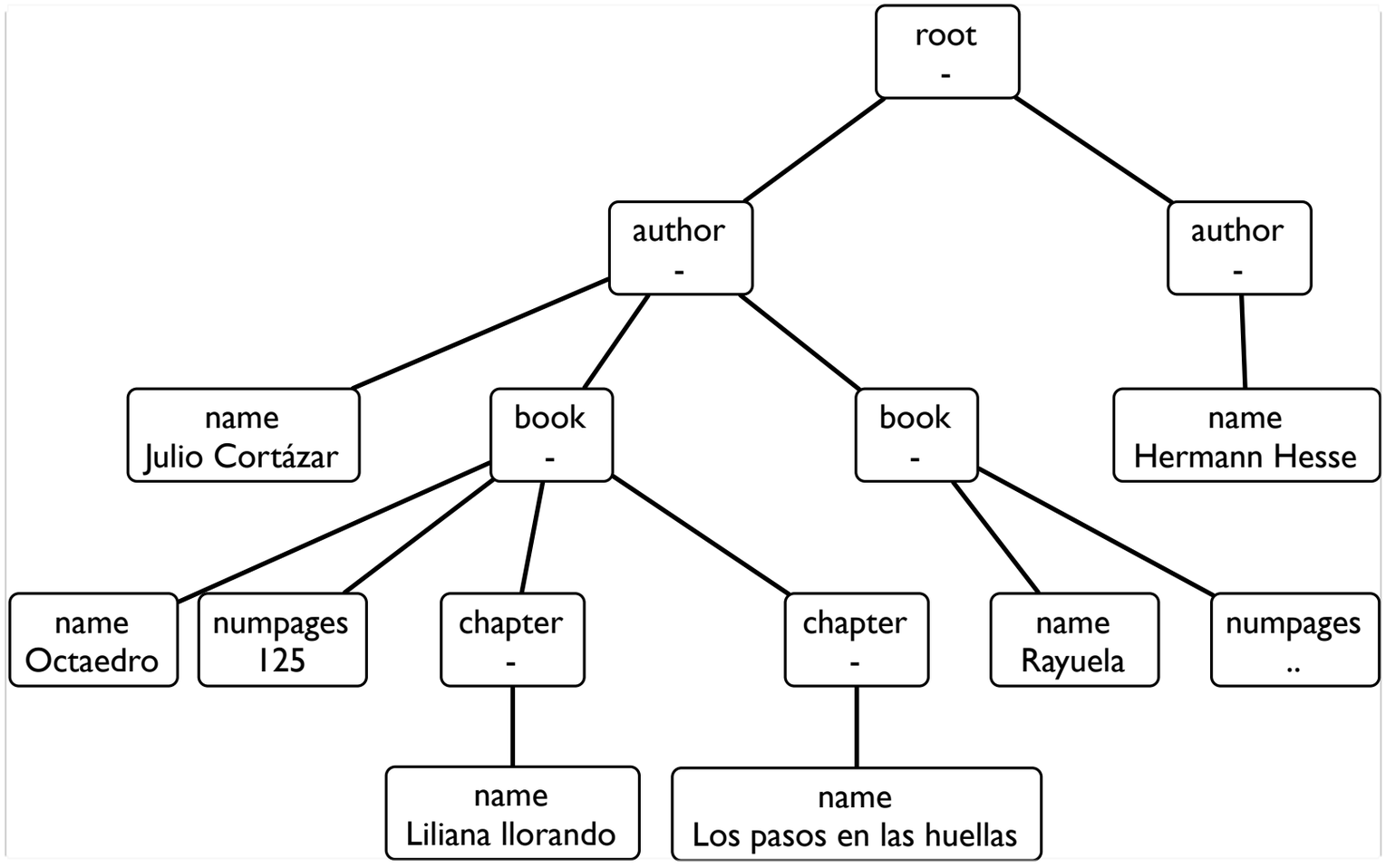
```
<author name = “Julio Cortázar”>
    <book name = “Octaedro” numpages = “125”>
        <chapter name = “Liliana llorando”/>
        <chapter name = “Los pasos en las huellas”/>
    </book>
    <book name= “Rayuela” numpages = “...”>
        ...
    </book>
</author>
<author name = “Hermann Hesse”>
    ...
</author>
```

XML

```
<author name = “Julio Cortázar”>
    <book name = “Octaedro” numpages = “125”>
        <chapter name = “Liliana llorando”/>
        <chapter name = “Los pasos en las huellas”/>
    </book>
    <book name= “Rayuela” numpages = “...”>
        ...
    </book>
</author>
<author name = “Hermann Hesse”>
    ...
</author>
```

XML

data tree



```
<author name = “Julio Cortázar”>
    <book name = “Octaedro” numpages = “125”>
        <chapter name = “Liliana llorando”/>
        <chapter name = “Los pasos en las huellas”/>
    </book>
    <book name= “Rayuela” numpages = “...”>
        ...
    </book>
</author>
<author name = “Hermann Hesse”>
    ...
</author>
```

XML

data tree

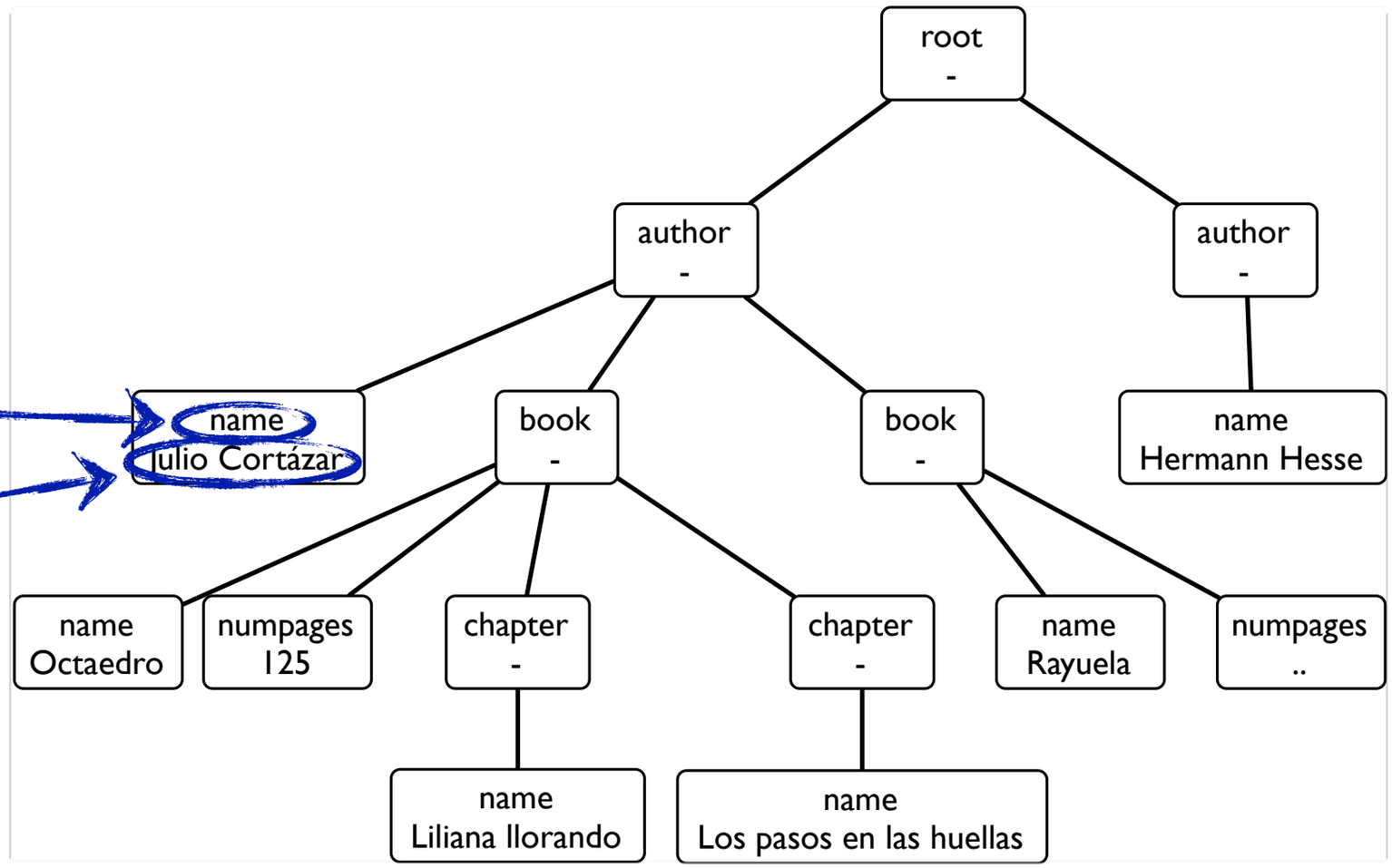
data tree

A finite, unranked tree over

a finite alphabet

&

an infinite domain



Reasoning on *data* trees

Reasoning on *data* trees

satisfiability for logics

Reasoning on *data* trees

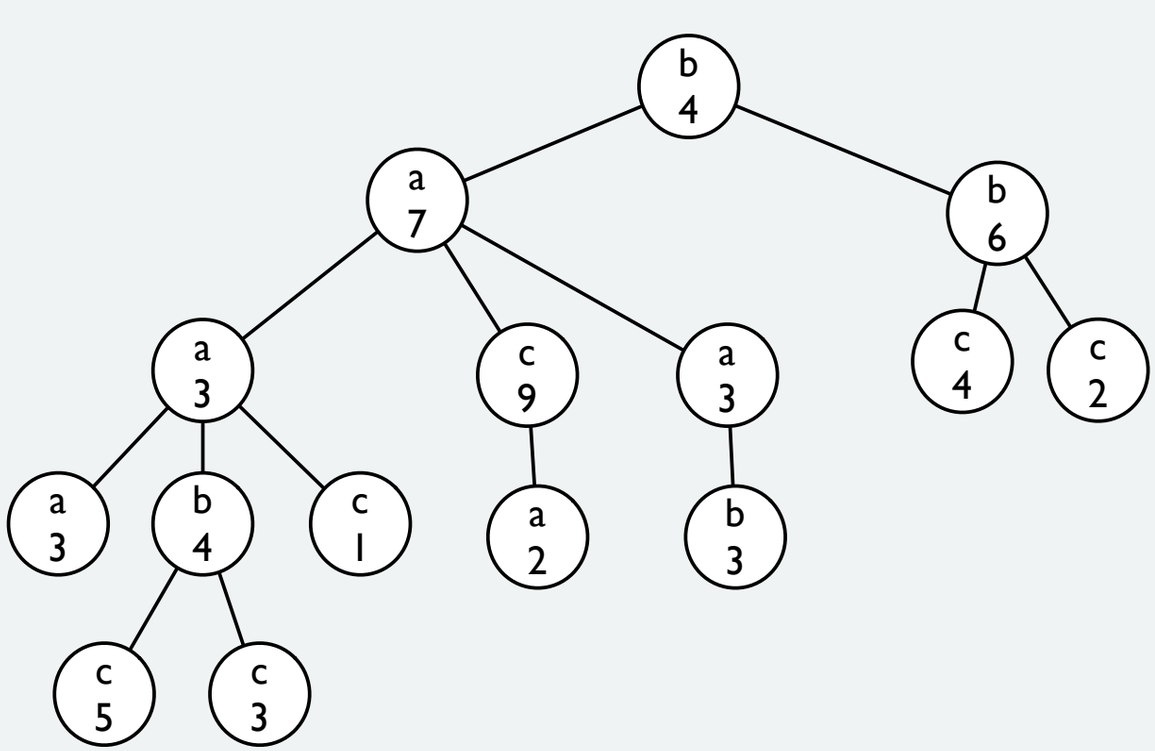
satisfiability for logics navigation & joins

Reasoning on *data* trees

satisfiability for XPath navigation & joins

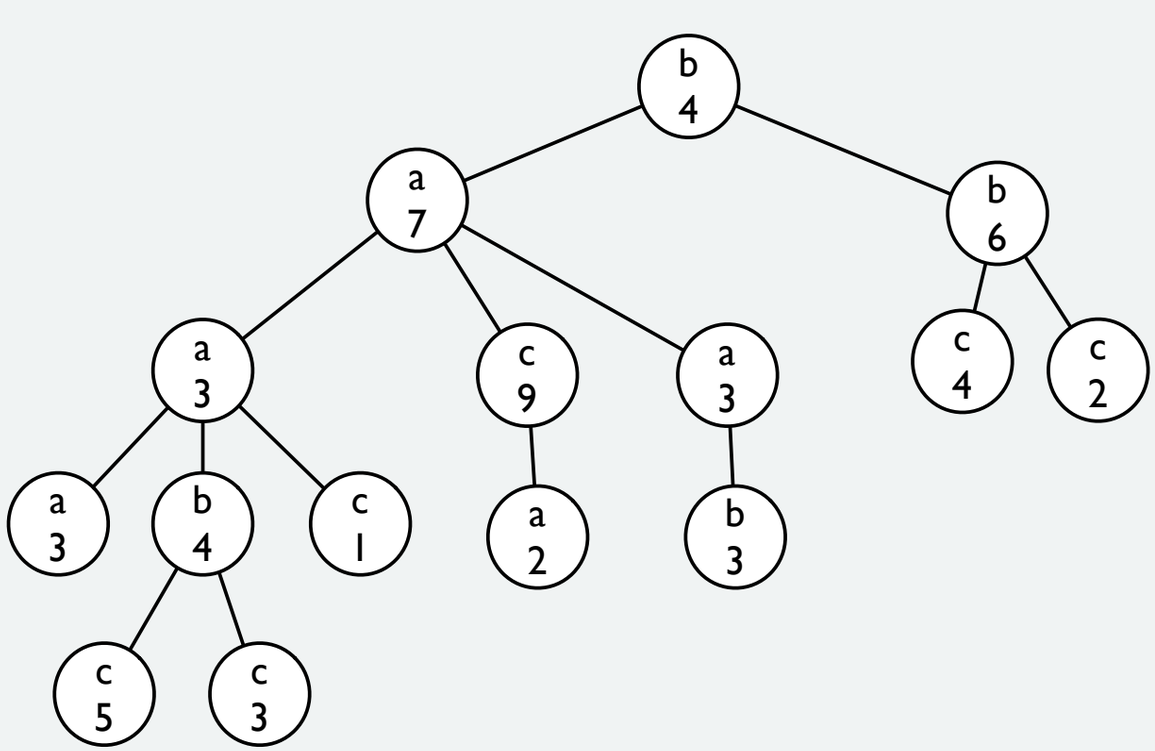
XPath, what's that...?

path exp



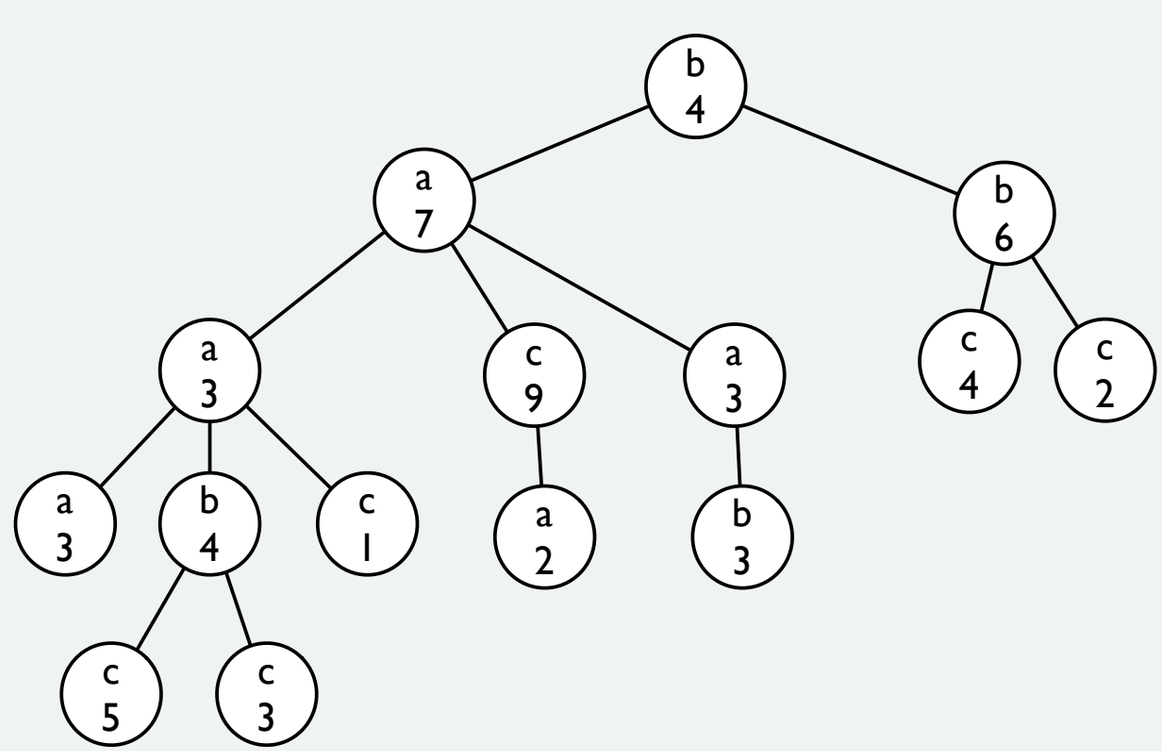
XPath, what's that...?

path exp go to ancestor, go to child, go to right sibling, go to descendant



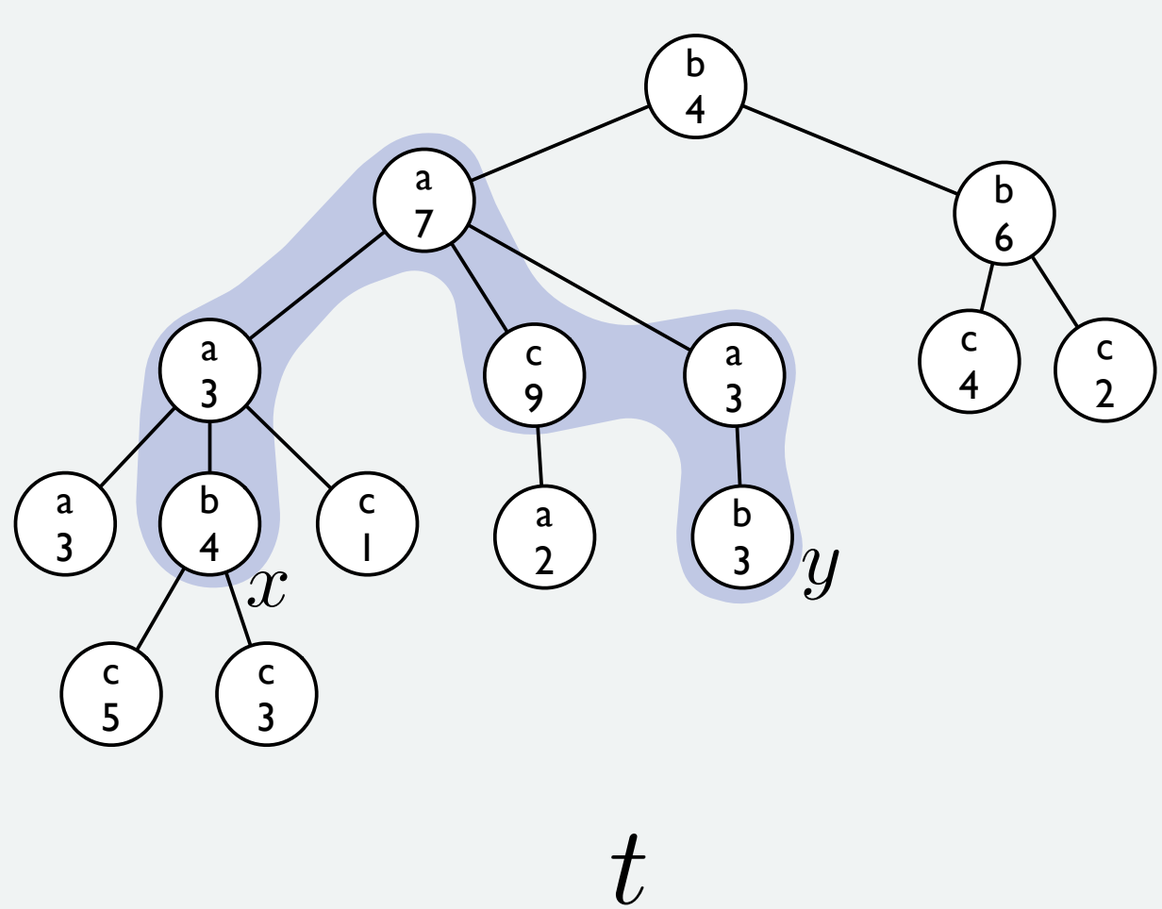
XPath, what's that...?

path exp $\uparrow^* \downarrow \rightarrow \downarrow_*$



XPath, what's that...?

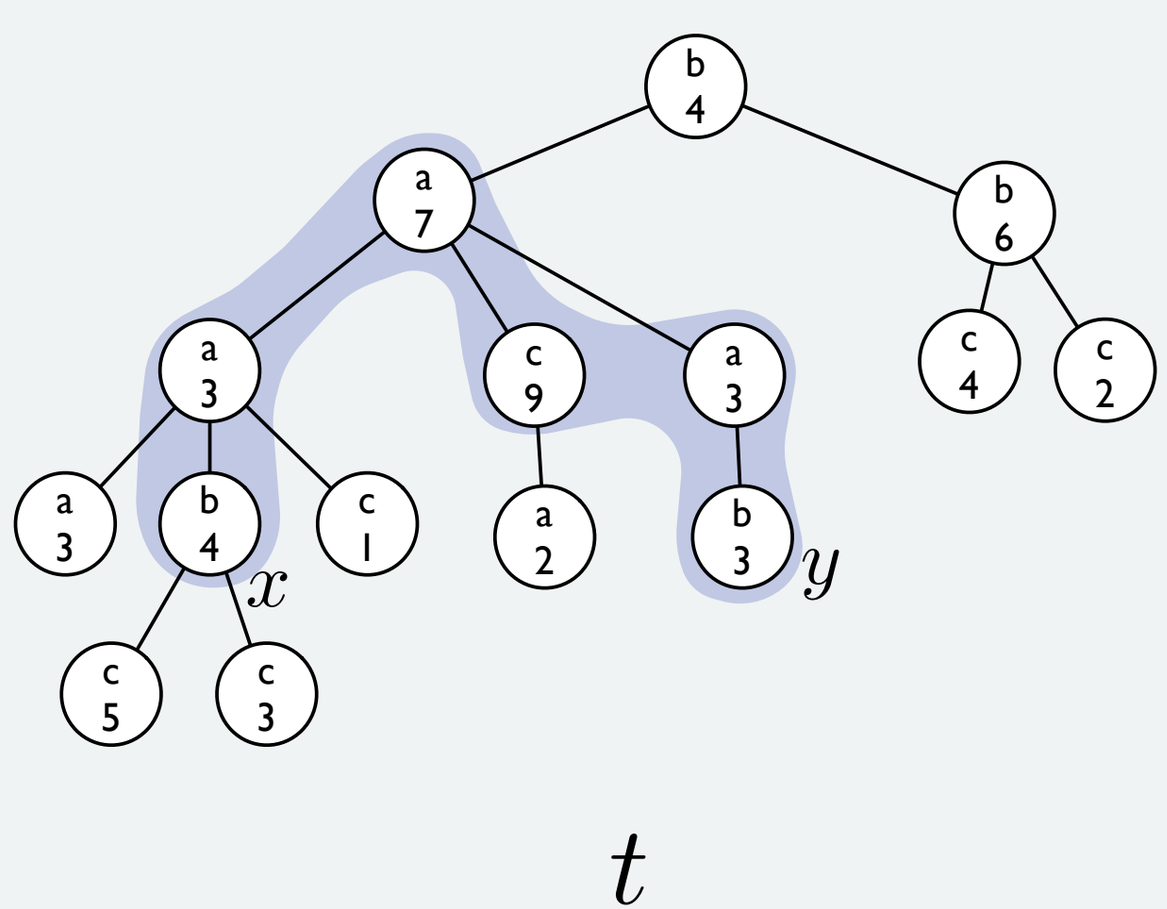
path exp $\qquad t, (x, y) \models \uparrow^* \downarrow \rightarrow \downarrow_*$



t

XPath, what's that...?

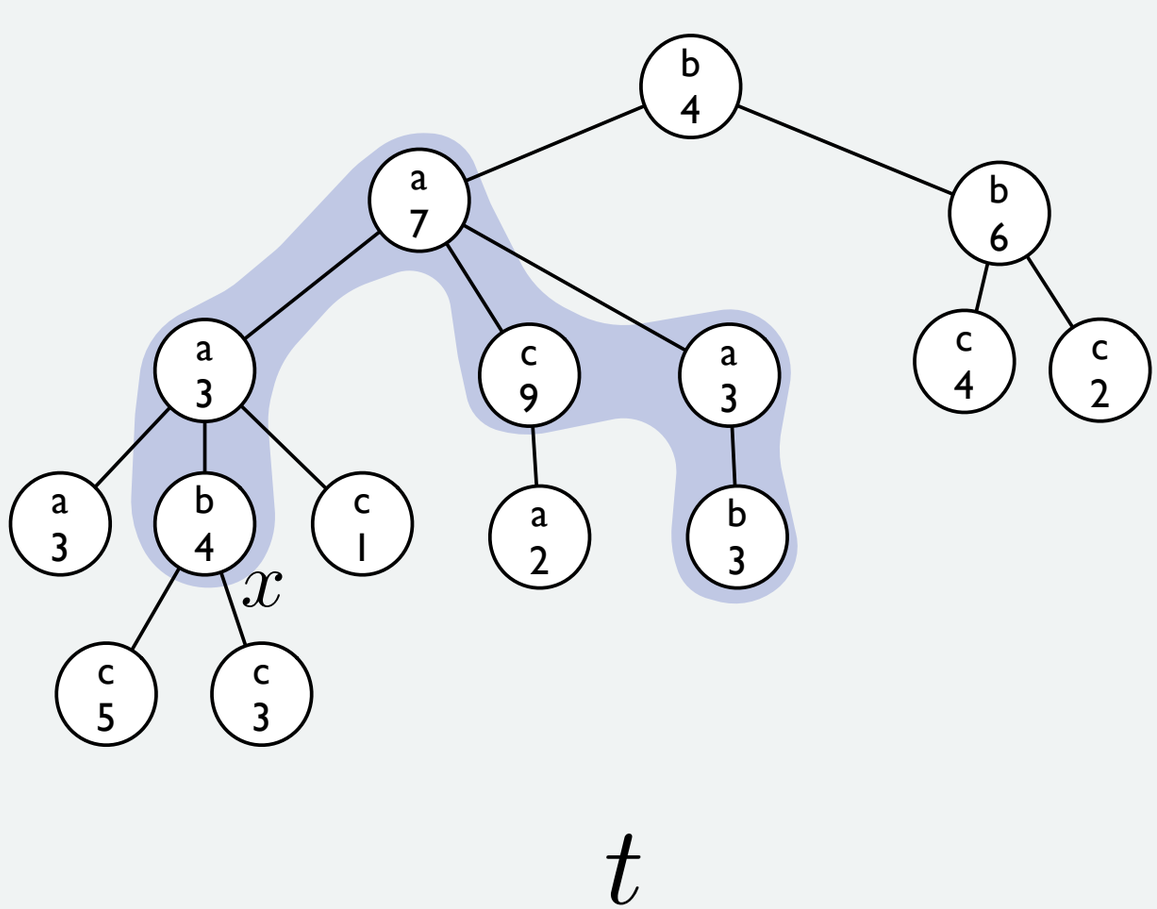
path exp $\qquad t, (x, y) \models \uparrow^* [a] \downarrow [c] \rightarrow \downarrow_* [b]$



t

XPath, what's that...?

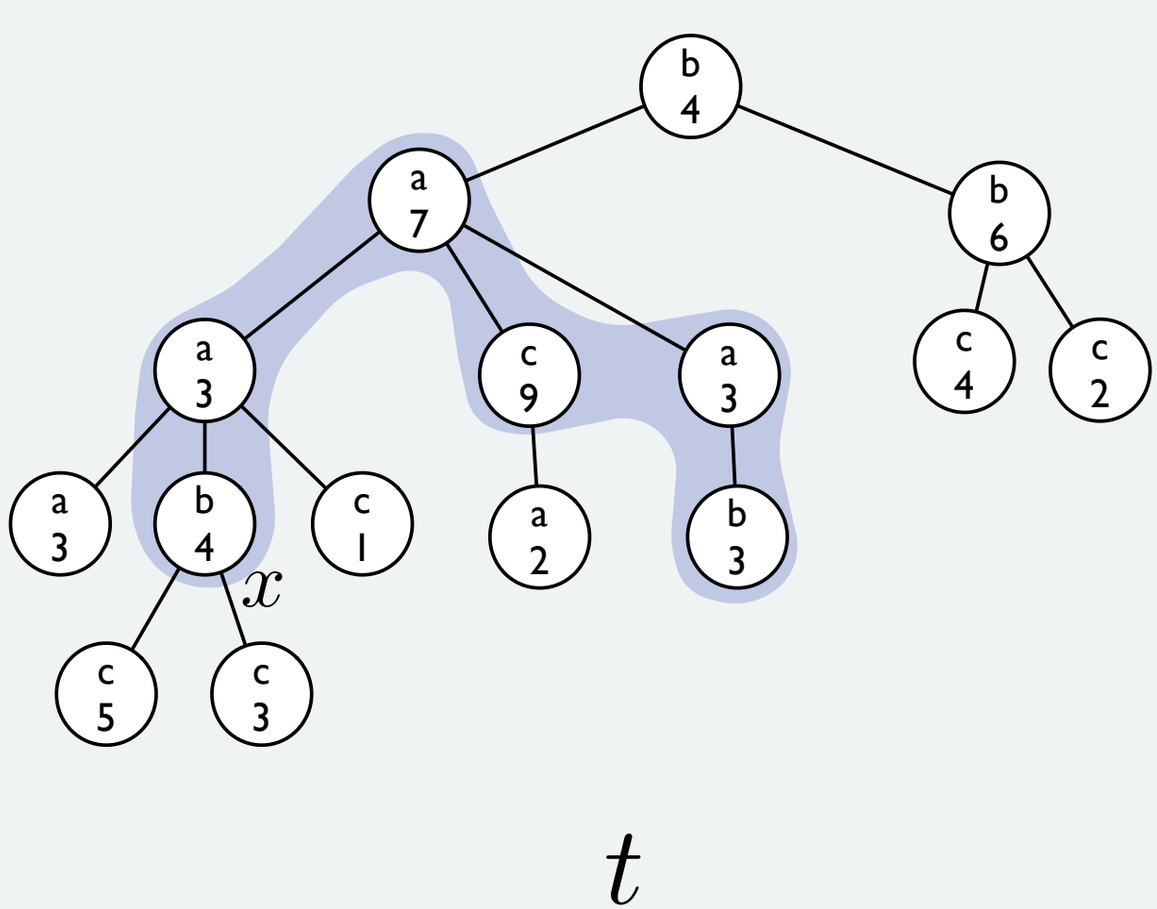
node exp $\qquad t, x \models \langle \uparrow^* [a] \downarrow [c] \rightarrow \downarrow_* [b] \rangle$



t

XPath, what's that...?

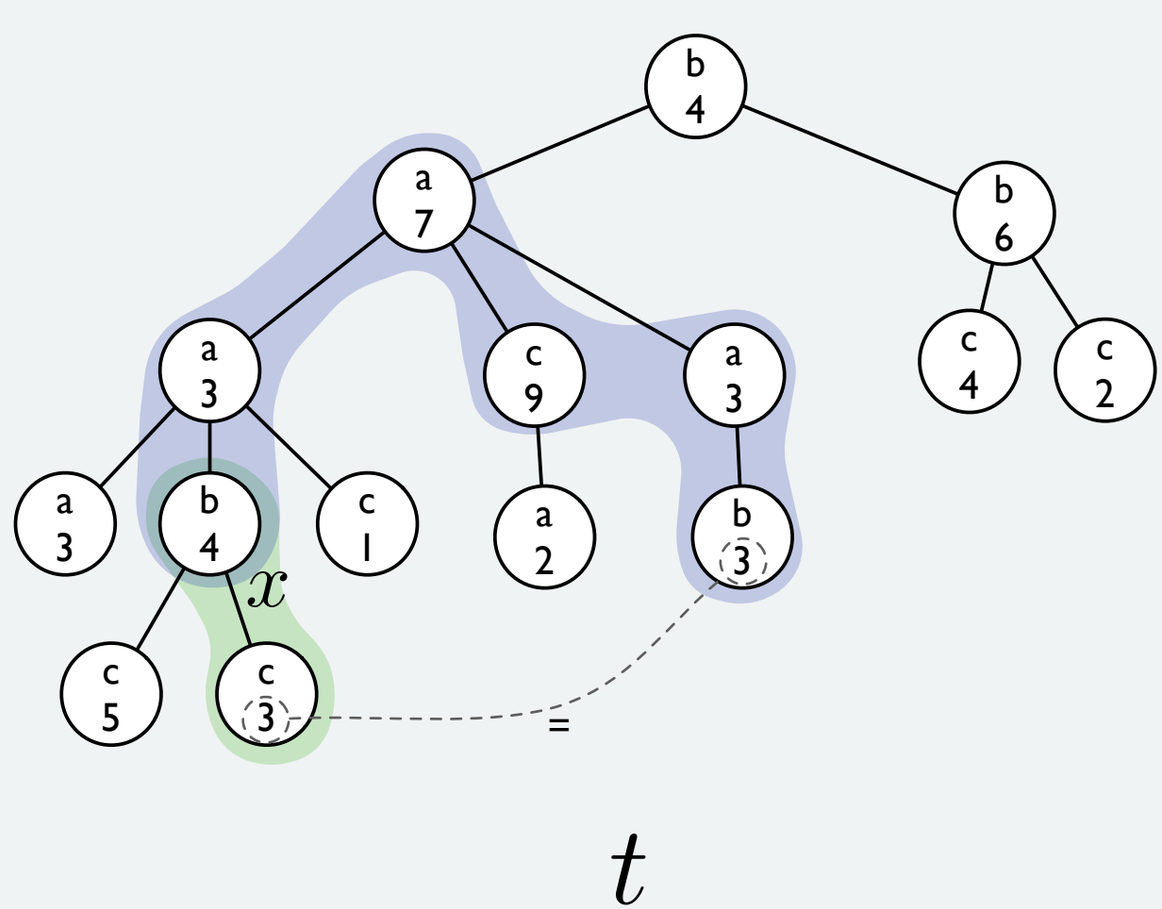
node exp $t, x \models \langle \uparrow^* [a] \downarrow [c] \rightarrow \downarrow_* [\neg \langle \downarrow \rangle \wedge b] \rangle$



t

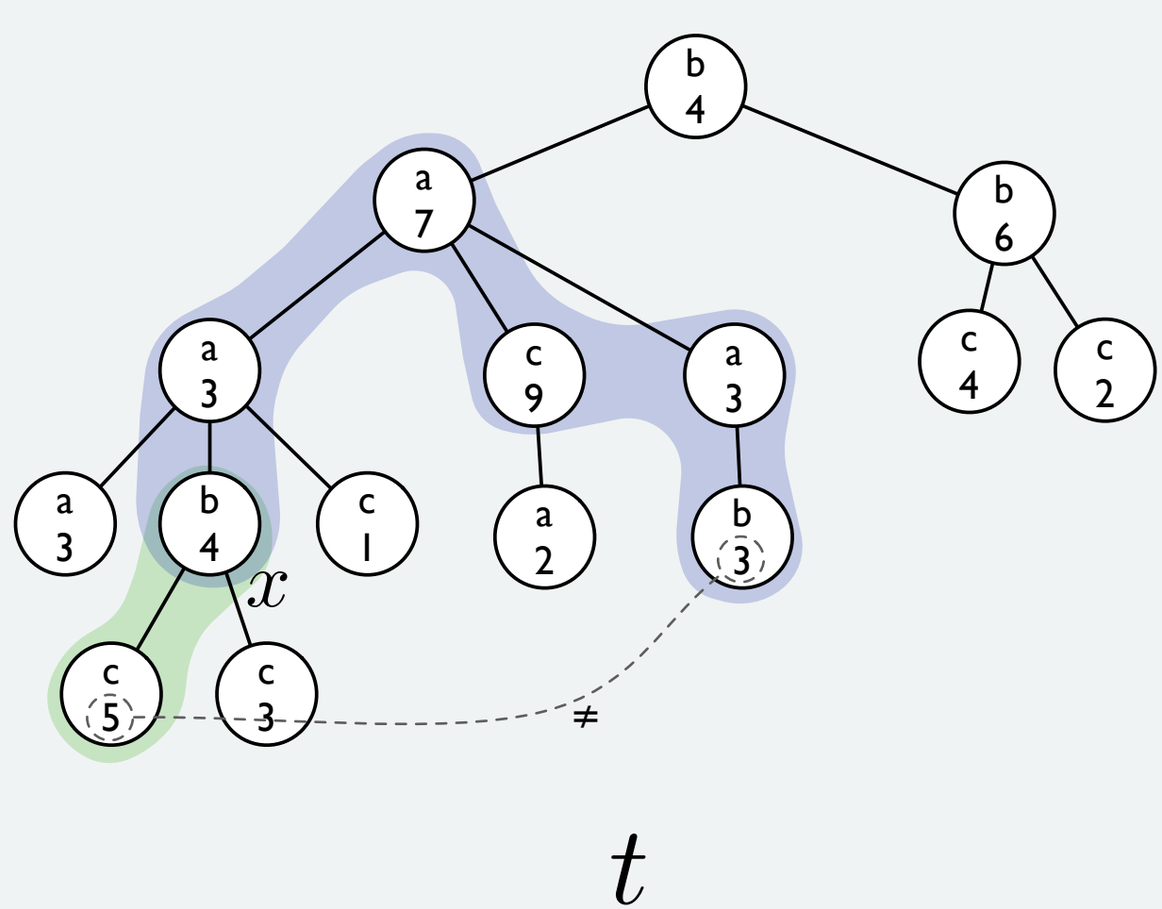
XPath, what's that...?

node exp $t, x \models \langle \downarrow[c] = \uparrow^* [a] \downarrow [c] \rightarrow \downarrow_* [\neg \langle \downarrow \rangle \wedge b] \rangle$



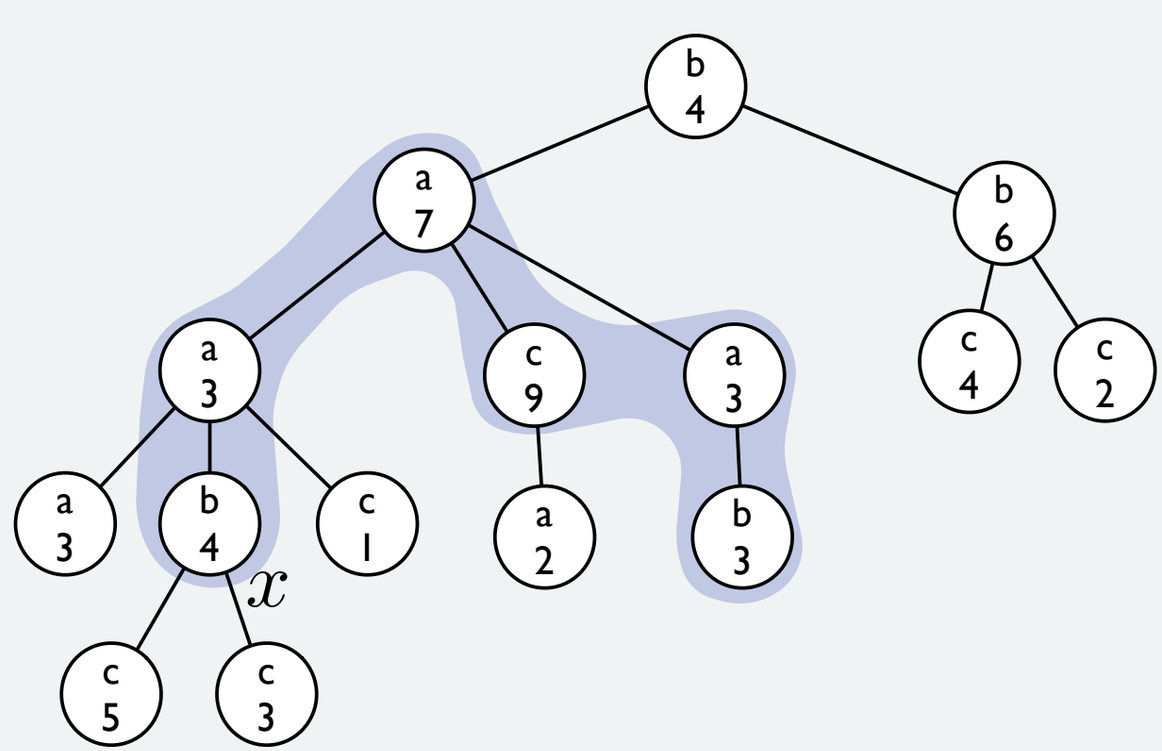
XPath, what's that...?

node exp $t, x \models \langle \downarrow[c] \neq \uparrow^* [a] \downarrow [c] \rightarrow \downarrow_* [\neg \langle \downarrow \rangle \wedge b] \rangle$



XPath, what's that...?

node exp $\quad t, x \models \neg \langle \downarrow[c] \neq \uparrow^* [a] \downarrow [c] \rightarrow \downarrow_* [\neg \langle \downarrow \rangle \wedge b] \rangle$



t

Satisfiability of XPath

Satisfiability of XPath

Undecidable

full XPath ♠

♠ [Geerts, Fan, 2005]

Satisfiability of XPath

Decidable

Forward ♦

$\mathsf{XPath}(\downarrow, \downarrow_*, \rightarrow, \rightarrow^*)$

Vertical ♣

$\mathsf{XPath}(\downarrow, \downarrow_*, \uparrow, \uparrow^*)$

Undecidable

full XPath ♠

♦ [F. 2010]

♣ [F., Segoufin, 2010]

♠ [Geerts, Fan, 2005]

Satisfiability of XPath

ExpTime

Decidable

Undecidable

Downward ♥
XPath($\downarrow, \downarrow_*$)

XPath without data tests ♠

Forward ♦
XPath($\downarrow, \downarrow_*, \rightarrow, \rightarrow^*$)

Vertical ♣
XPath($\downarrow, \downarrow_*, \uparrow, \uparrow^*$)

full XPath ♠

♥ [F. 2009]
♦ [F. 2010]
♣ [F., Segoufin, 2010]
♠ [Geerts, Fan, 2005]

Satisfiability of XPath

NP

Positive-XPath ♠

ExpTime

Downward ♥
XPath($\downarrow, \downarrow_*$)

XPath without data tests ♠

Decidable

Forward ♦
XPath($\downarrow, \downarrow_*, \rightarrow, \rightarrow^*$)

Vertical ♣
XPath($\downarrow, \downarrow_*, \uparrow, \uparrow^*$)

Undecidable

full XPath ♠

♥ [F. 2009]
♦ [F. 2010]
♣ [F., Segoufin, 2010]
♠ [Geerts, Fan, 2005]

Horizontal

$\mathsf{XPath}(\rightarrow, \rightarrow^*, \leftarrow, {}^*\leftarrow)$

$\mathsf{XPath}(\rightarrow^+, {}^+\leftarrow)$: undecidable ♣

$\mathsf{XPath}(\rightarrow^+, {}^*\leftarrow)$: undecidable ♣

$\mathsf{XPath}(\rightarrow, \rightarrow^*, {}^*\leftarrow)$: undecidable ♣

$\mathsf{XPath}(\rightarrow^+)$: decidable, non-PR ♠ ♣

In particular, any fragment with $\rightarrow^+$ or ${}^+\leftarrow$ is **undecidable** or has a **non-PR complexity**

♠ [Demri, Lazić, 2006]

♣ [F., Segoufin, 2009]

Horizontal

$\mathsf{XPath}(\rightarrow, \rightarrow^*, \leftarrow, {}^*\leftarrow)$

$\mathsf{XPath}(\rightarrow^+, {}^+\leftarrow)$: undecidable ♣

$\mathsf{XPath}(\rightarrow^+, {}^*\leftarrow)$: undecidable ♣

$\mathsf{XPath}(\rightarrow, \rightarrow^*, {}^*\leftarrow)$: undecidable ♣

$\mathsf{XPath}(\rightarrow^+)$: decidable, non-PR ♠ ♣

In particular, any fragment with $\rightarrow^+$ or ${}^+\leftarrow$ is **undecidable** or has a **non-PR complexity**

What about $\mathsf{XPath}(\rightarrow^*)$?

♠ [Demri, Lazić, 2006]

♣ [F., Segoufin, 2009]

Horizontal

$\mathsf{XPath}(\rightarrow, \rightarrow^*, \leftarrow, {}^*\leftarrow)$

$\mathsf{XPath}(\rightarrow^+, {}^+\leftarrow)$: undecidable ♣

$\mathsf{XPath}(\rightarrow^+, {}^*\leftarrow)$: undecidable ♣

$\mathsf{XPath}(\rightarrow, \rightarrow^*, {}^*\leftarrow)$: undecidable ♣

$\mathsf{XPath}(\rightarrow^+)$: decidable, non-PR ♠ ♣

In particular, any fragment with $\rightarrow^+$ or ${}^+\leftarrow$ is **undecidable** or has a **non-PR complexity**

What about $\mathsf{XPath}(\rightarrow^*)$?

$\mathsf{XPath}(\rightarrow^*)$ is decidable in 2ExpSpace ♦

$\mathsf{XPath}(\rightarrow^*, {}^*\leftarrow)$ is decidable in 2ExpSpace ♦

♦ [F., 2011]

♠ [Demri, Lazić, 2006]

♣ [F., Segoufin, 2009]

Theorem

Satisfiability for $\mathsf{XPath}(\rightarrow^*, \downarrow_*, {}^*\leftarrow)$ is **decidable.**

Concluding remarks

Changing $\rightarrow^*$ by $\rightarrow^+$: undecidable

Concluding remarks

Changing $\rightarrow^*$ by $\rightarrow^+$: undecidable

Adding $\uparrow^*$: NPR

Concluding remarks

Changing $\rightarrow^*$ by $\rightarrow^+$: undecidable

Adding $\uparrow^*$: NPR

Adding $\downarrow$: still decidable?

Concluding remarks

Changing $\rightarrow^*$ by $\rightarrow^+$: undecidable

Adding $\uparrow^*$: NPR

Adding $\downarrow$: still decidable?

Complexity : 3ExpSpace (2ExpSpace in normal form)

Concluding remarks

Changing $\rightarrow^*$ by $\rightarrow^+$: undecidable

Thank you!

Adding $\uparrow^*$: NPR

Adding $\downarrow$: still decidable?

Complexity : 3ExpSpace (2ExpSpace in normal form)

Satisfiability problem for ϕ

Satisfiability problem for ϕ

Derivation problem for Forest Profiles

Satisfiability problem for ϕ

Derivation problem for Forest Profiles

Can we derive a profile verifying ϕ from the empty profiles using our algebra?

Satisfiability problem for ϕ

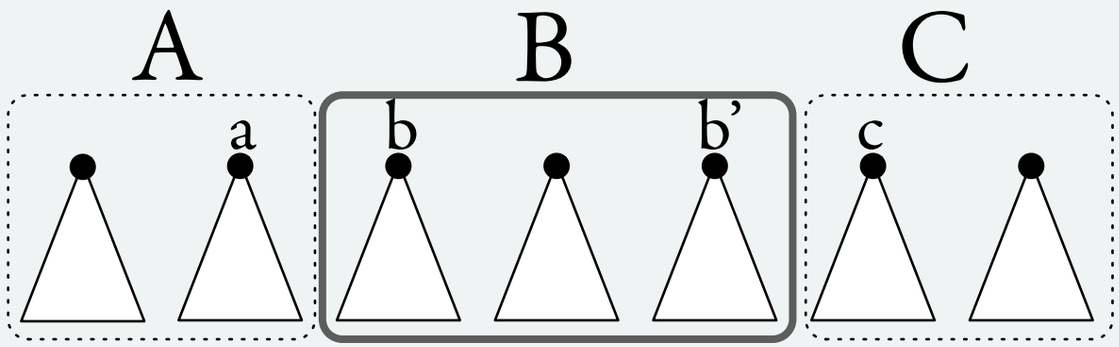
Satisfiability problem for ϕ

Normal form for ϕ

all path expressions α of ϕ are of the form 

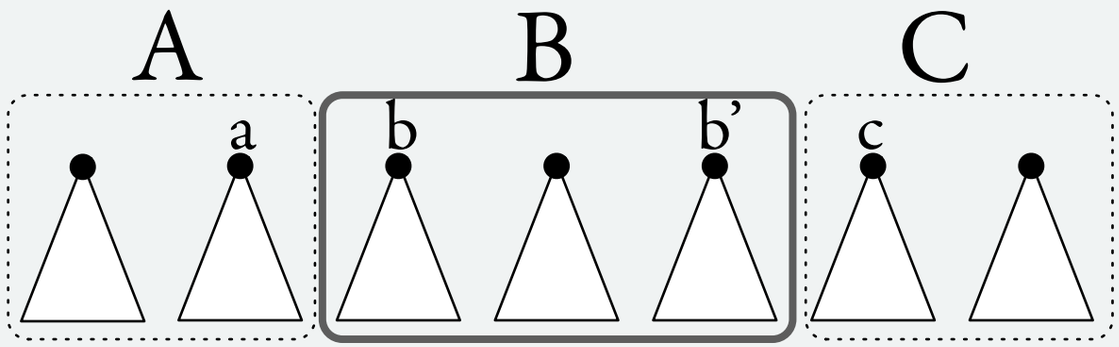
Satisfiability problem for ϕ

Forest profile



Satisfiability problem for ϕ

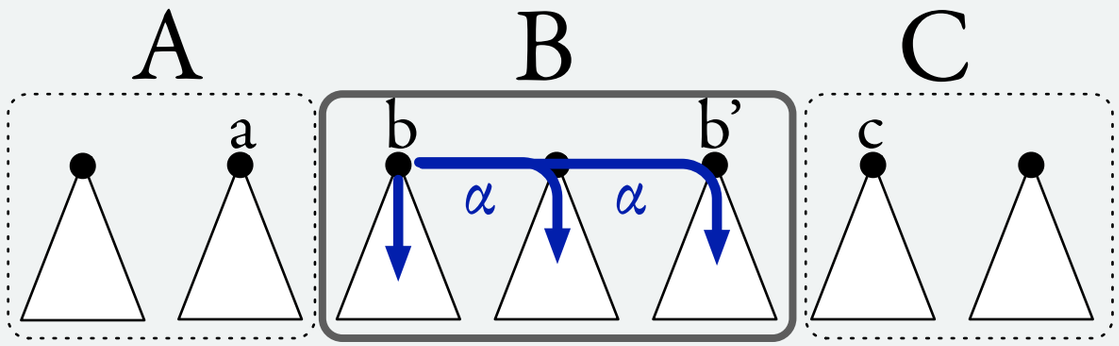
Forest profile



Contains, for every data value d and path formula α
whether we can reach d with with α:

Satisfiability problem for ϕ

Forest profile

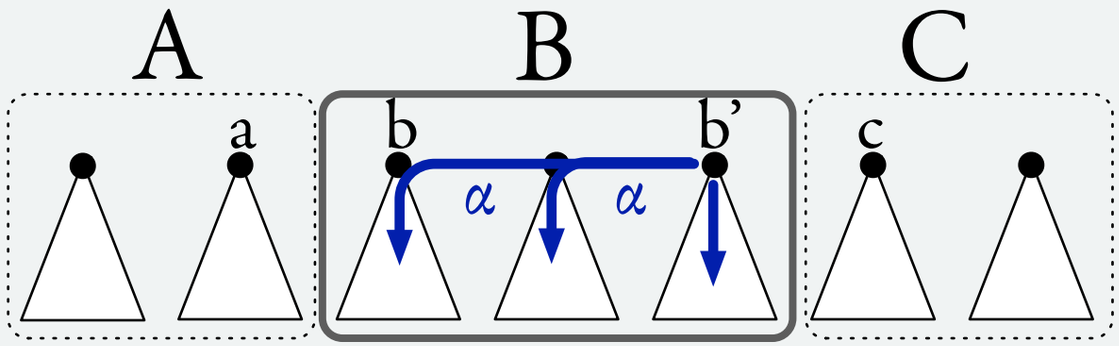


Contains, for every data value d and path formula α whether we can reach d with with α:

- from **b** inside B

Satisfiability problem for ϕ

Forest profile

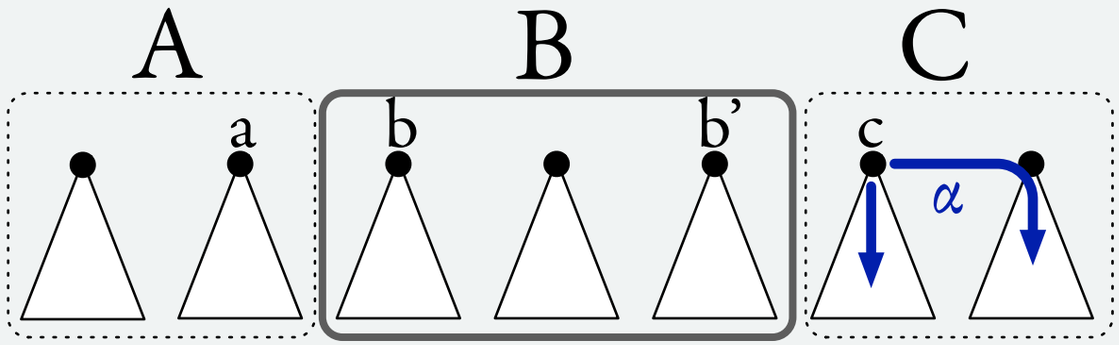


Contains, for every data value d and path formula α whether we can reach d with with α:

- from **b** inside B
- from **b'** inside B

Satisfiability problem for ϕ

Forest profile

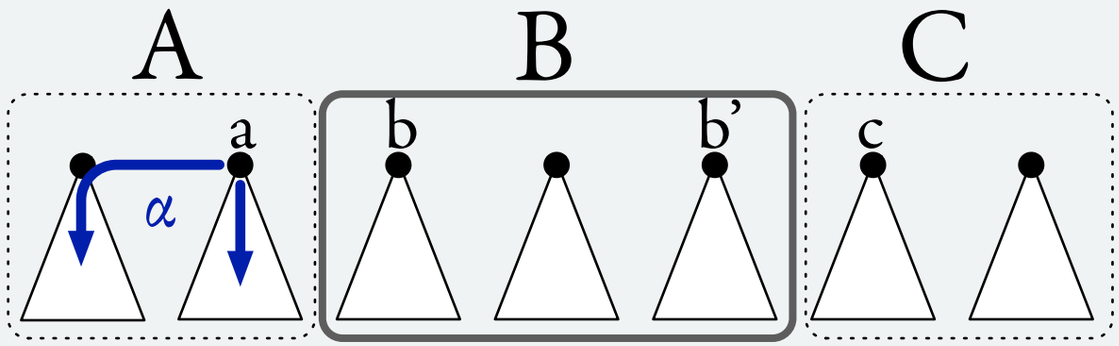


Contains, for every data value *d* and path formula α whether we can reach *d* with with α:

- from **b** inside B
- from **b'** inside B
- from **c** inside C

Satisfiability problem for ϕ

Forest profile

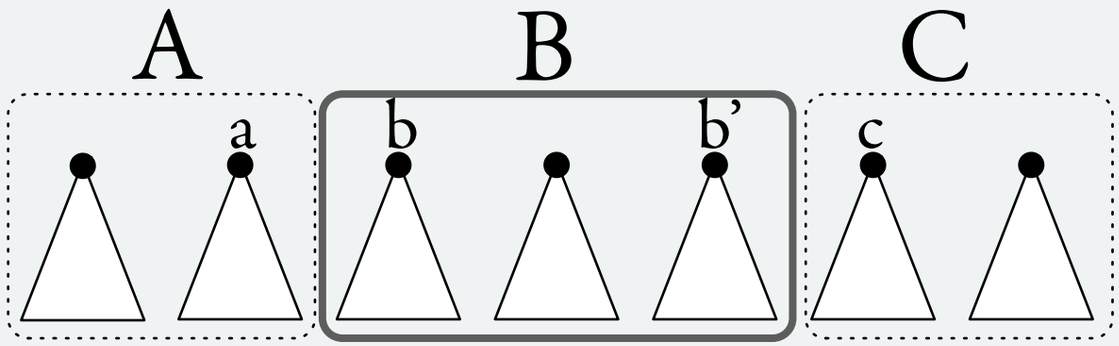


Contains, for every data value d and path formula α whether we can reach d with with α:

- from **b** inside B
- from **b'** inside B
- from **c** inside C
- from **a** inside A

Satisfiability problem for ϕ

Forest profile

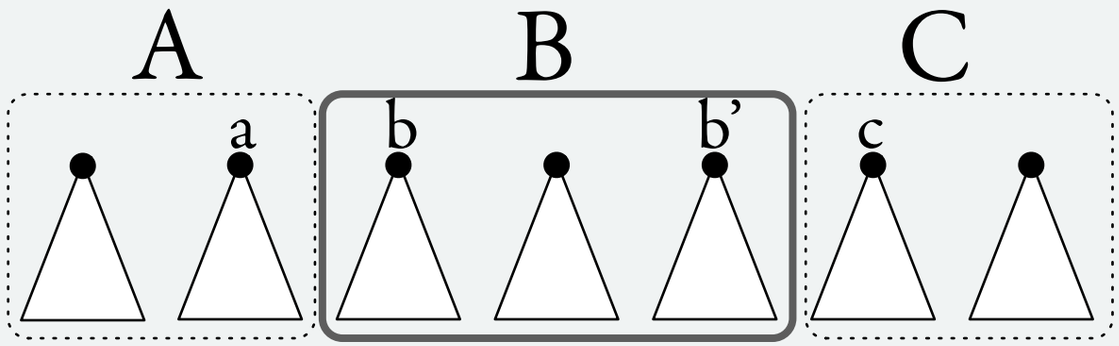


Contains, for every data value d and path formula α whether we can reach d with with α:

- from **b** inside B
- from **b'** inside B
- from **c** inside C
- from **a** inside A

Satisfiability problem for ϕ

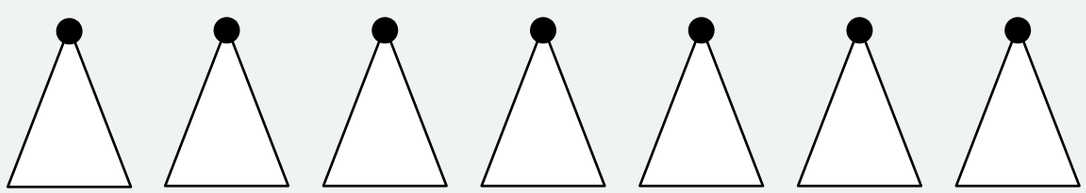
Forest profile



Contains, for every data value d and path formula α whether we can reach d with with α:

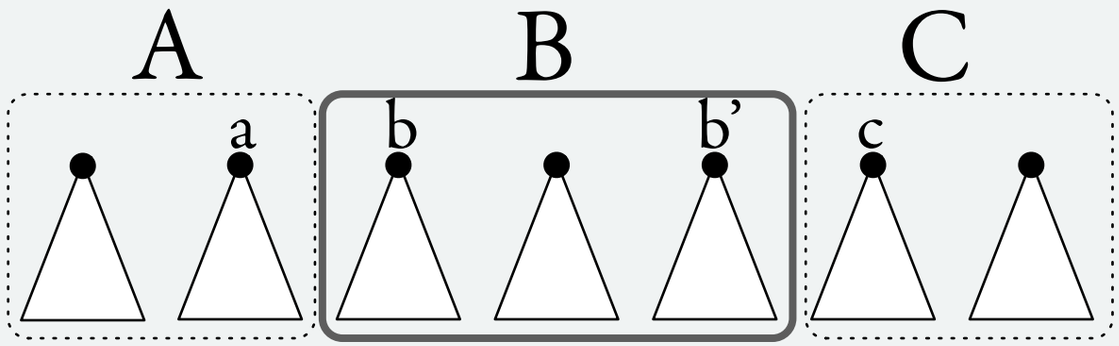
- from **b** inside B
- from **b'** inside B
- from **c** inside C
- from **a** inside A

Algebra



Satisfiability problem for ϕ

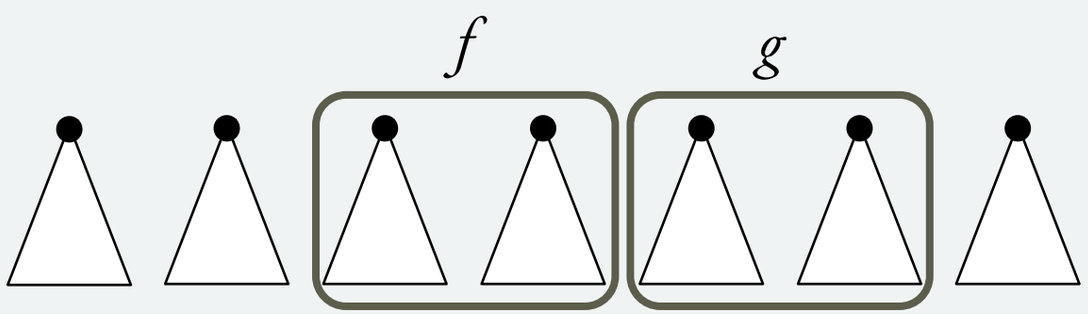
Forest profile



Contains, for every data value d and path formula α whether we can reach d with with α:

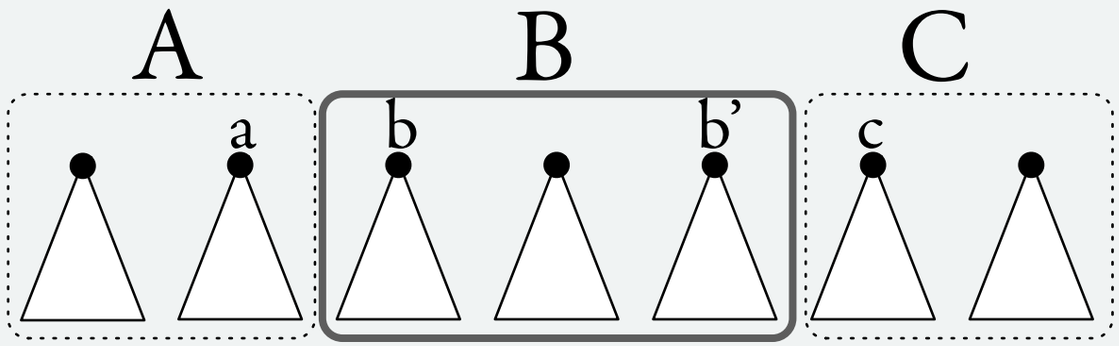
- from **b** inside B
- from **b'** inside B
- from **c** inside C
- from **a** inside A

Algebra



Satisfiability problem for ϕ

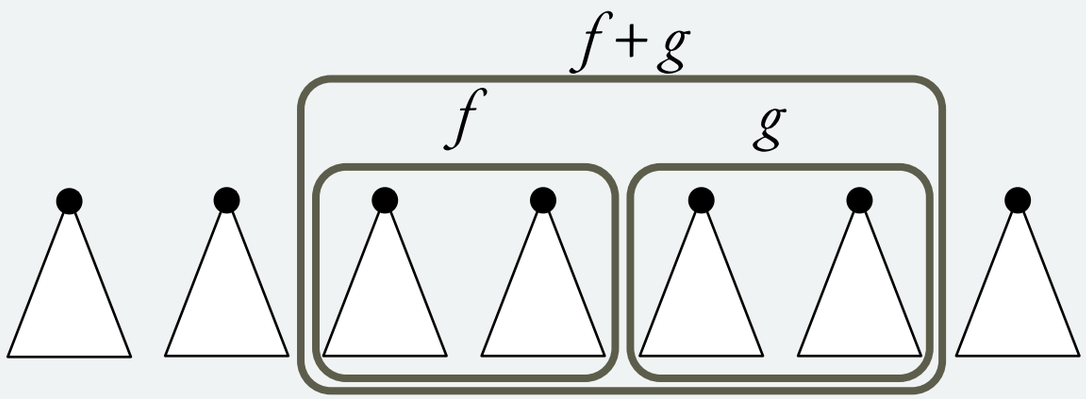
Forest profile



Contains, for every data value d and path formula α whether we can reach d with with α:

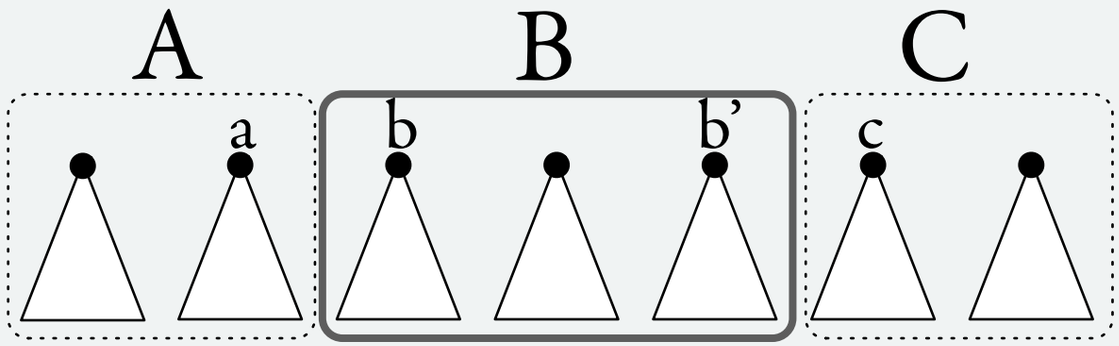
- from **b** inside B
- from **b'** inside B
- from **c** inside C
- from **a** inside A

Algebra



Satisfiability problem for ϕ

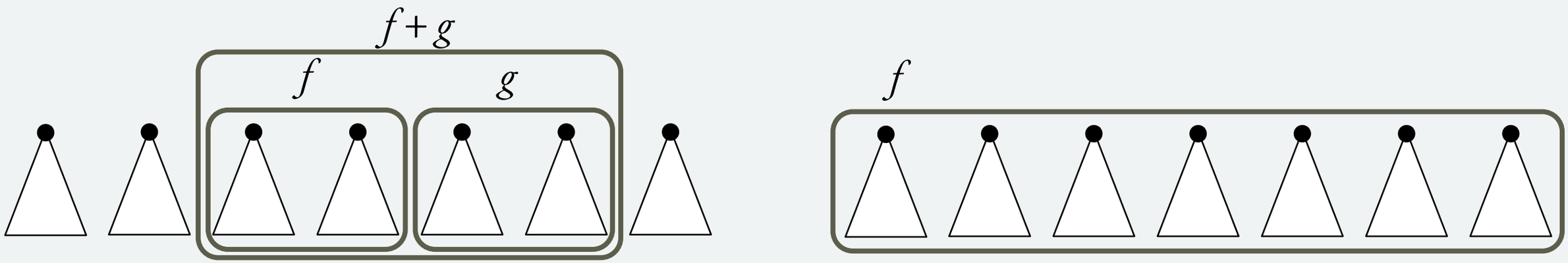
Forest profile



Contains, for every data value d and path formula α whether we can reach d with with α:

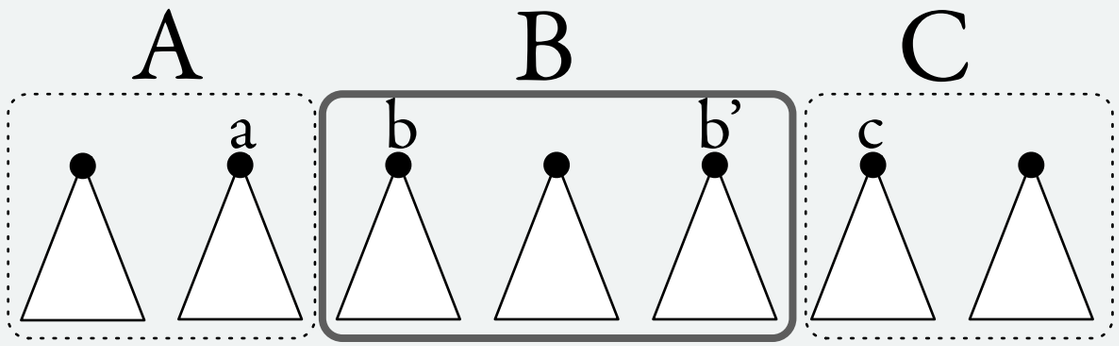
- from **b** inside B
- from **b'** inside B
- from **c** inside C
- from **a** inside A

Algebra



Satisfiability problem for ϕ

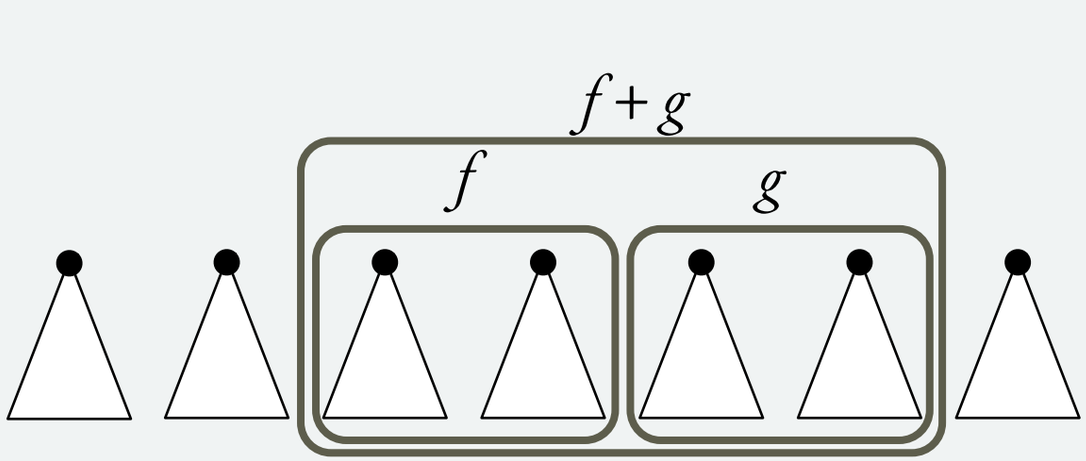
Forest profile

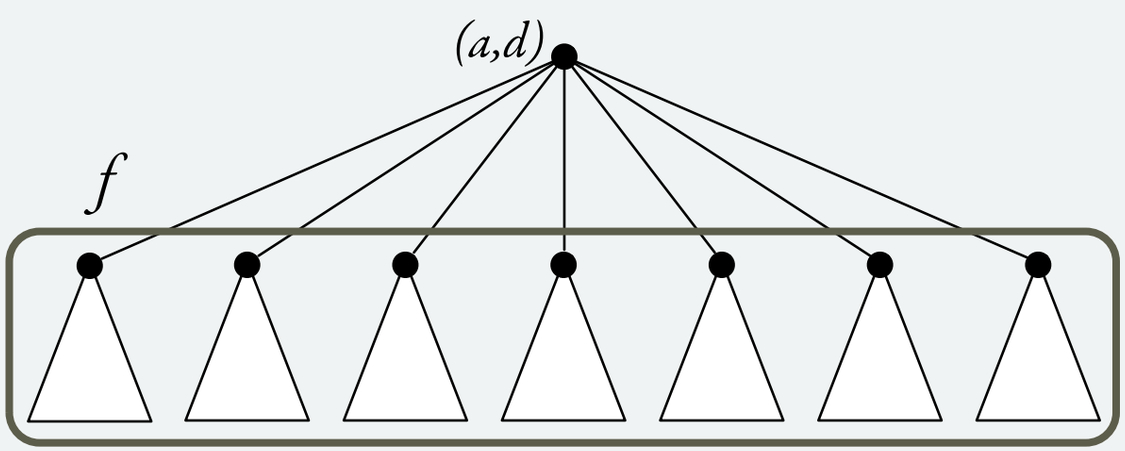


Contains, for every data value d and path formula α whether we can reach d with with α:

- from **b** inside B
- from **b'** inside B
- from **c** inside C
- from **a** inside A

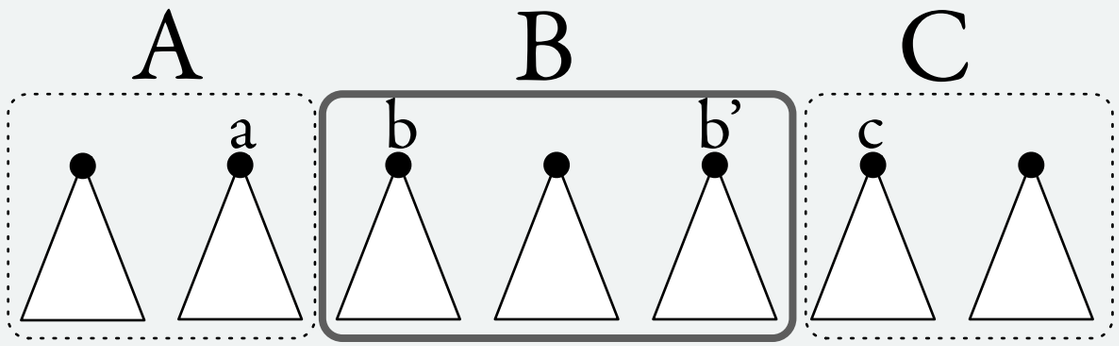
Algebra





Satisfiability problem for ϕ

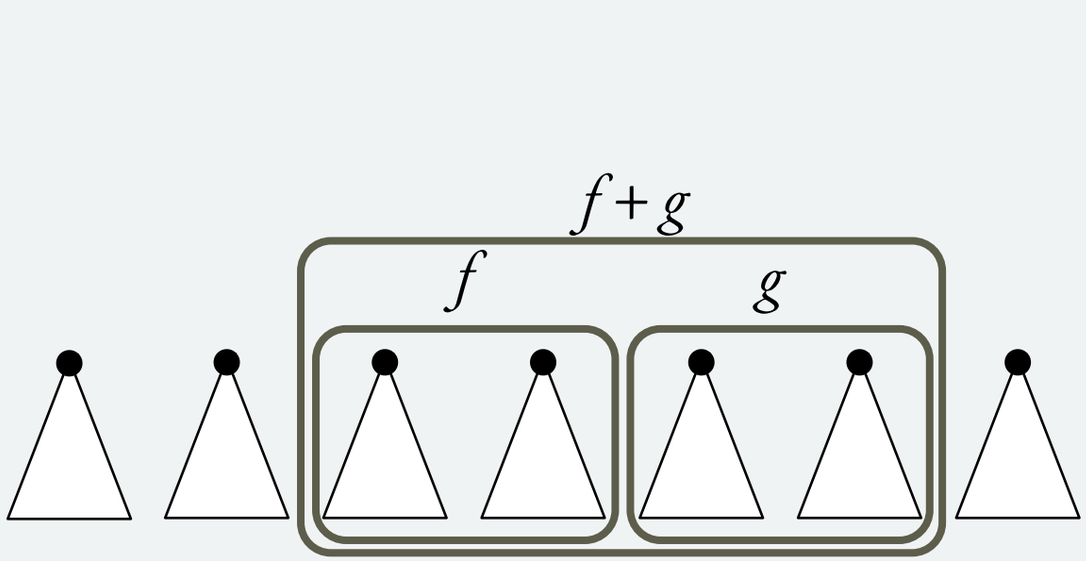
Forest profile

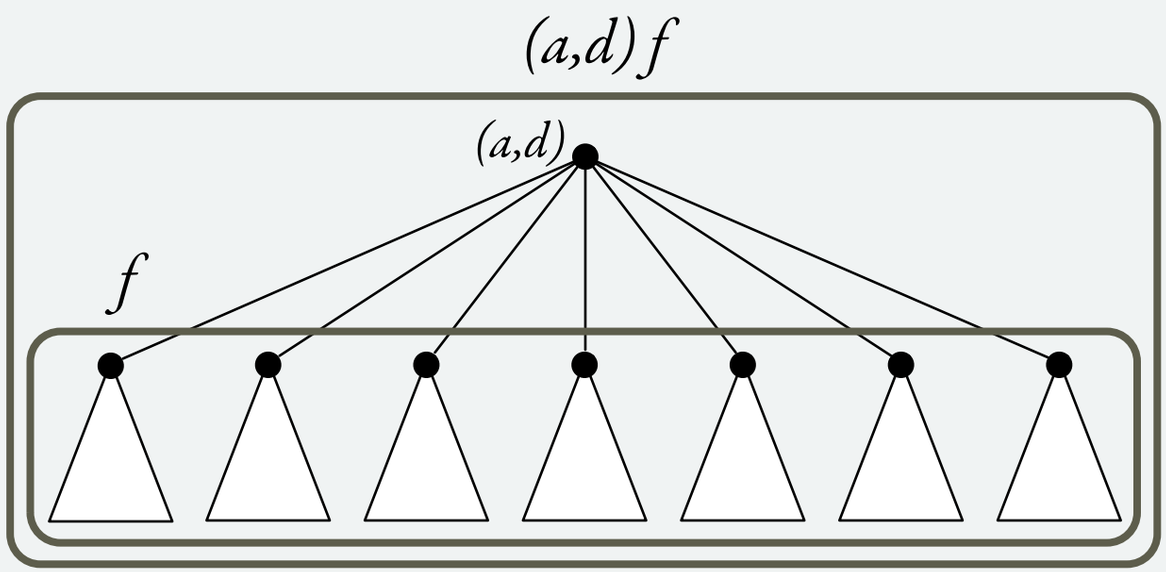


Contains, for every data value d and path formula α whether we can reach d with with α:

- from **b** inside B
- from **b'** inside B
- from **c** inside C
- from **a** inside A

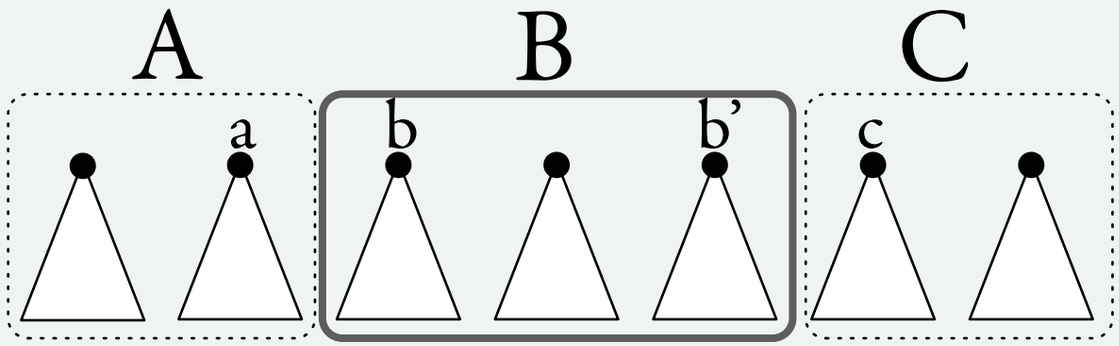
Algebra





Satisfiability problem for ϕ

Forest profile

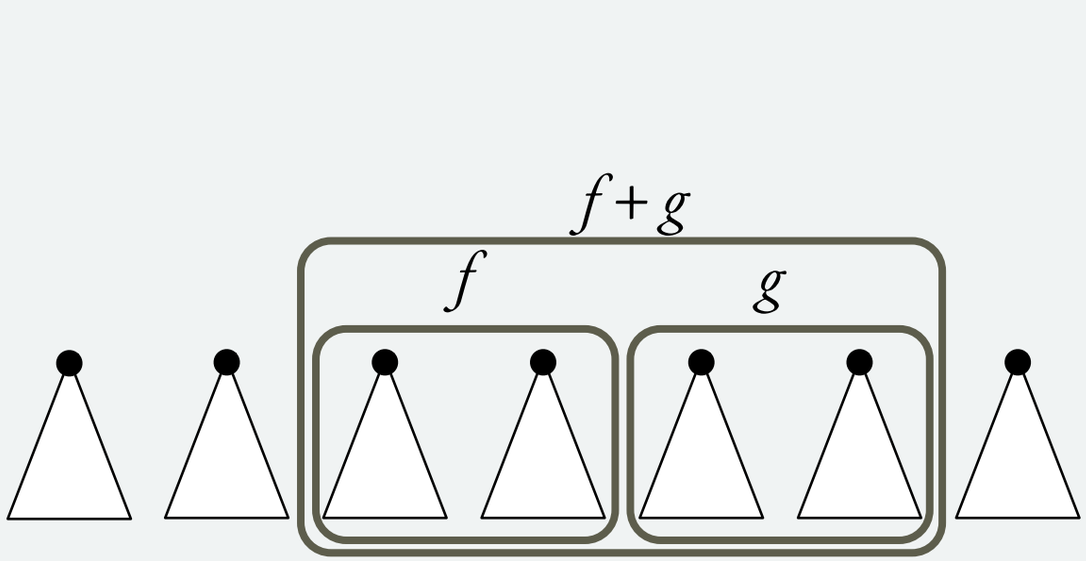


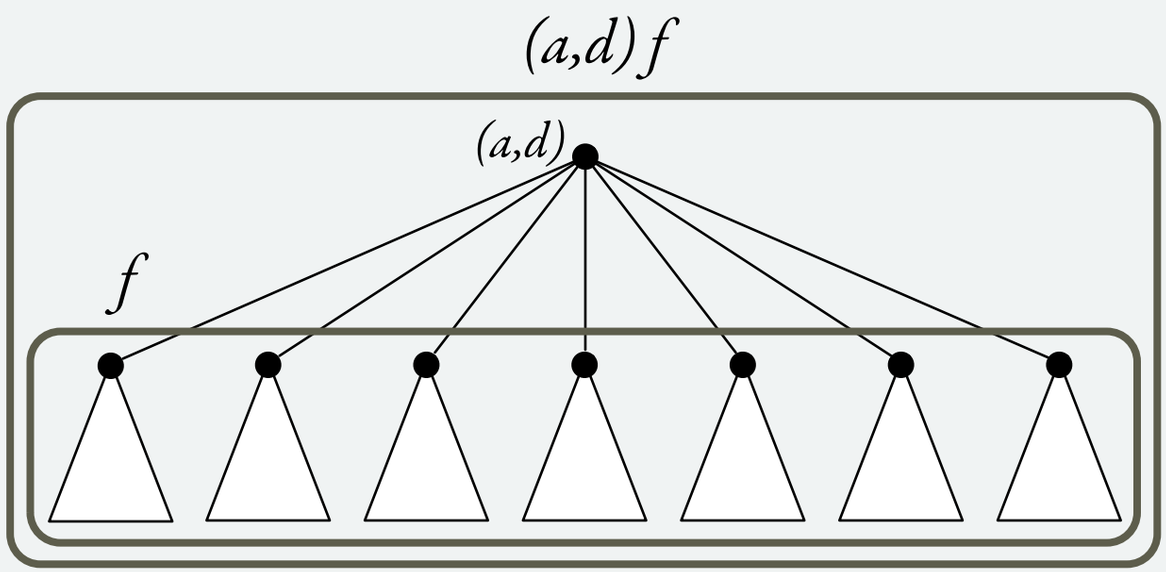
Contains, for every data value d and path formula α whether we can reach d with with α:

- from **b** inside B
- from **b'** inside B
- from **c** inside C
- from **a** inside A

Algebra

consistent profiles = profiles that do not contradict ϕ

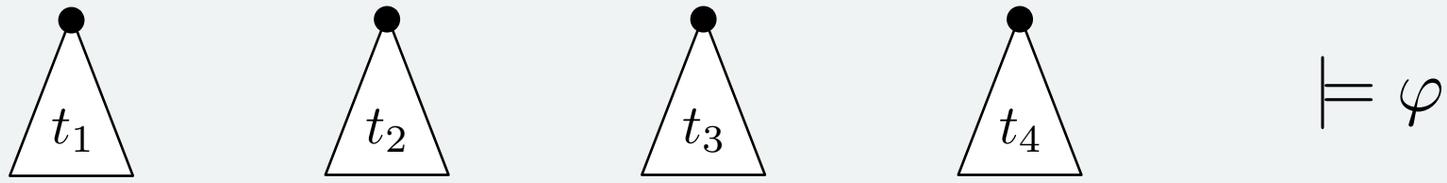




Key closure property

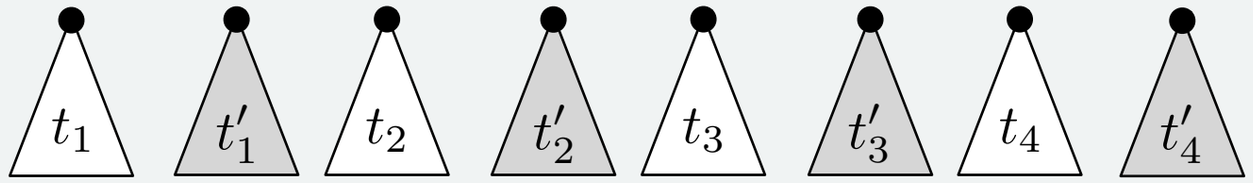
Key closure property

For almost every data value *d:*



Key closure property

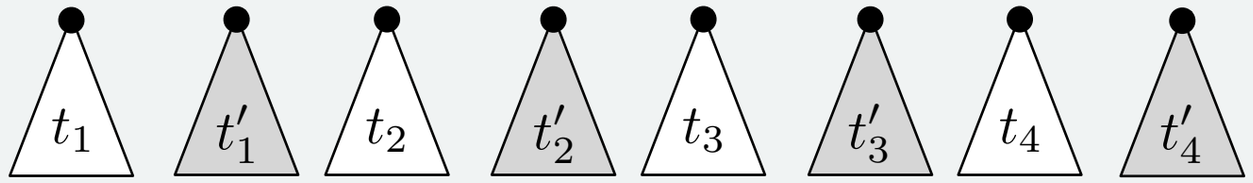
For almost every data value *d:*



where $t'_i = t_i[d \mapsto d']$

Key closure property

For almost every data value *d:*



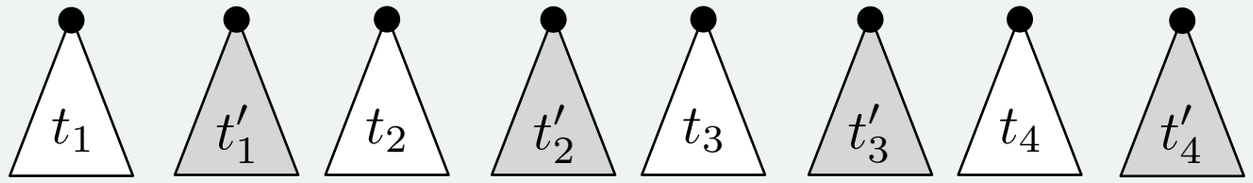
where $t'_i = t_i[d \mapsto d']$

adding  's ⇝ partial order < on FP

closure property ⇝ <-compatibility of FP algebra

Key closure property

For almost every data value *d:*



where $t'_i = t_i[d \mapsto d']$

adding  's ⇝ partial order < on FP

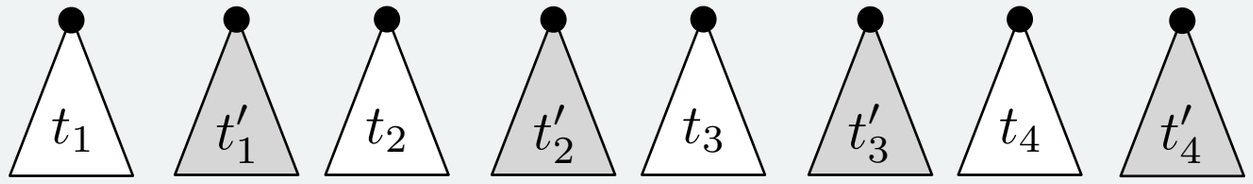
closure property ⇝ <-compatibility of FP algebra

derivation problem

derivation problem
for <-minimal profiles

Key closure property

For almost every data value d:



where $t'_i = t_i[d \mapsto d']$

adding  's $\rightsquigarrow$ partial order < on FP

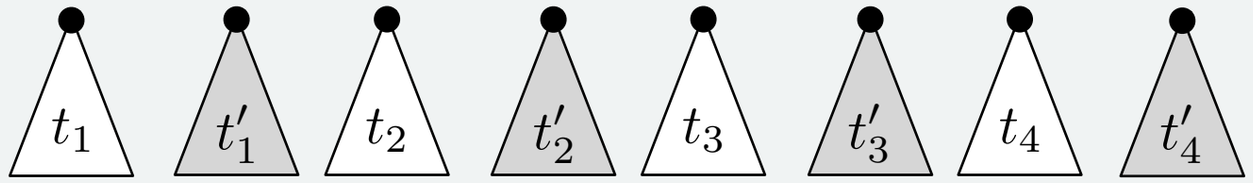
closure property $\rightsquigarrow$ <-compatibility of FP algebra

derivation problem

derivation problem
for <-minimal profiles
bounded extensions of

Key closure property

For almost every data value *d:*



where $t'_i = t_i[d \mapsto d']$

adding  's ⇝ partial order < on FP

closure property ⇝ <-compatibility of FP algebra

derivation problem

↓

derivation problem

for <u>≤-minimal profiles</u>

bounded extensions of

boundedly many!

Satisfiability problem for ϕ

Derivation problem for Forest Profiles

Satisfiability problem for ϕ

Derivation problem for Forest Profiles

Derivation problem over a **finite set** of Forest Profiles

Satisfiability problem for ϕ

Derivation problem for Forest Profiles

Derivation problem over a **finite set** of Forest Profiles

Finite-state reachability algorithm