ACTS
CMI Chennai
10-2-2015

Reasoning with reflexive-transitive path logics

Diego Figueira
CNRS, LaBRI

data word

b	c	a	b	c	c	a	a	b	c	b
3	1	5	1	1	1	4	4	5	1	4

data word

b	c	a	b	c	c	a	a	b	c	b
3	1	5	1	1	1	4	4	5	1	4

$\in (\mathbf{A} \times \mathbf{D})^*$

D: infinite domain

A: finite alphabet

data word

b	c	a	b	c	c	a	a	b	c	b
3	1	5	1	1	1	4	4	5	1	4

$\in (\mathbf{A} \times \mathbf{D})^*$

- **D** → infinite domain
- **A** → finite alphabet

Reasoning with logics on data words:

high complexity
or
limited expressive power

data word

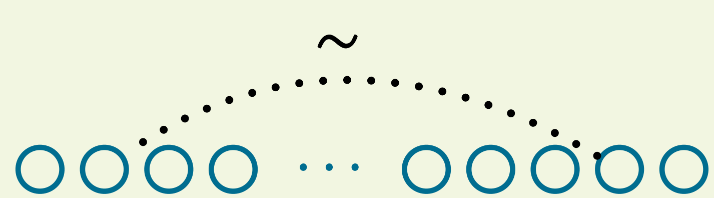
b	c	a	b	c	c	a	a	b	c	b
3	1	5	1	1	1	4	4	5	1	4

$\in (\mathbf{A} \times \mathbf{D})^*$

infinite domain
finite alphabet

Reasoning with logics on data words: **high complexity** or **limited expressive power**

Things become ugly as soon as:



Logics for data words

Logic	SAT	
$FO^2(<,+1,\sim)$	~ PN-reach	[Bojańczyk & al.]
$FO^2(<,\sim)$	NExpTime-c	[Bojańczyk & al.]
$LTL^{\downarrow}(F, U, X)$	Decidable, non-PR hard	[Demri, Lazić]
$LTL^{\downarrow}(F)$	Decidable, non-PR hard	[F, Segoufin]
$LTL^{\downarrow}(F, F^{-1})$	Undecidable	[F, Segoufin]
BasicDataLTL	~ PN-reach	[Kara & al]
LRV	2ExpSpace-c	[Demri, F, Praveen]
LRV + P	~ PN-reach	[Demri, F, Praveen]

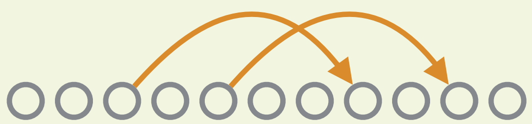
Logics for data words

Logic	SAT	
$FO^2(<,+1,\sim)$	~ PN-reach	[Bojańczyk & al.]
$FO^2(<,\sim)$	NExpTime-c	[Bojańczyk & al.]
$LTL^{\downarrow}(F, U, X)$	Decidable, non-PR hard	[Demri, Lazić]
$LTL^{\downarrow}(F)$	Decidable, non-PR hard	[F, Segoufin]
$LTL^{\downarrow}(F, F^{-1})$	Undecidable	[F, Segoufin]
BasicDataLTL	~ PN-reach	[Kara & al]
LRV	2ExpSpace-c	[Demri, F, Praveen]
LRV + P	~ PN-reach	[Demri, F, Praveen]

Path logics

XPath on data words

path expressions

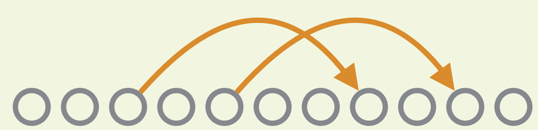


node expressions

Path logics

XPath on data words

path expressions



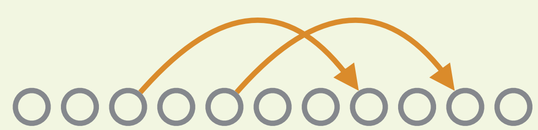
$$\alpha,\beta ::= \varepsilon \mid \alpha\beta \mid \alpha[\phi] \mid o \qquad o \in \{ \longrightarrow, \longrightarrow^{+}, \longrightarrow^{*}, \longleftarrow, {}^{+}\!\longleftarrow, {}^{*}\!\longleftarrow \}$$

node expressions

Path logics

XPath on data words

path expressions



$$\alpha,\beta ::= \varepsilon \mid \alpha\beta \mid \alpha[\phi] \mid \boldsymbol{o} \qquad \boldsymbol{o} \in \{ \longrightarrow, \longrightarrow^{+}, \longrightarrow^{*}, \longleftarrow, {}^{+}\!\longleftarrow, {}^{*}\!\longleftarrow \}$$

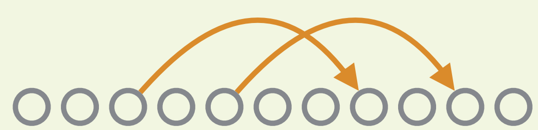
node expressions

$$\phi,\psi ::= \mathbf{a} \mid \neg\phi \mid \phi\wedge\psi \mid \langle\alpha=\beta\rangle \mid \langle\alpha\neq\beta\rangle \mid \alpha? \qquad \mathbf{a}\in\mathbf{A}$$

Path logics

XPath on data words

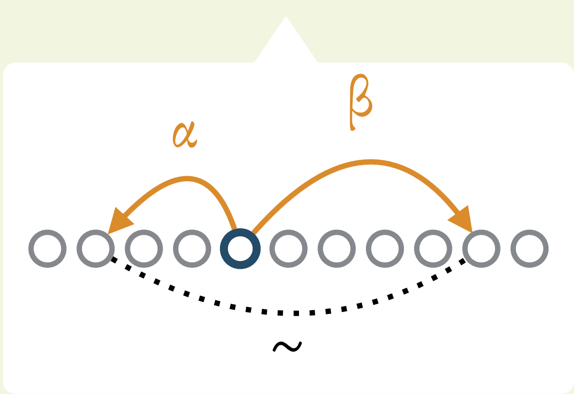
path expressions



$$\alpha,\beta ::= \varepsilon \mid \alpha\beta \mid \alpha[\phi] \mid \boldsymbol{o} \qquad \boldsymbol{o} \in \{\longrightarrow, \longrightarrow^{+}, \longrightarrow^{*}, \longleftarrow, {}^{+}\!\longleftarrow, {}^{*}\!\longleftarrow\}$$

node expressions

$$\phi,\psi ::= \mathbf{a} \mid \neg\phi \mid \phi\wedge\psi \mid \langle\alpha=\beta\rangle \mid \langle\alpha\neq\beta\rangle \mid \alpha? \qquad \mathbf{a}\in\mathbf{A}$$



Path logics

XPath on data words

b	c	a	b	c	b	a	a	b	c	b
3	1	5	1	1	1	4	4	5	1	4

Path logics

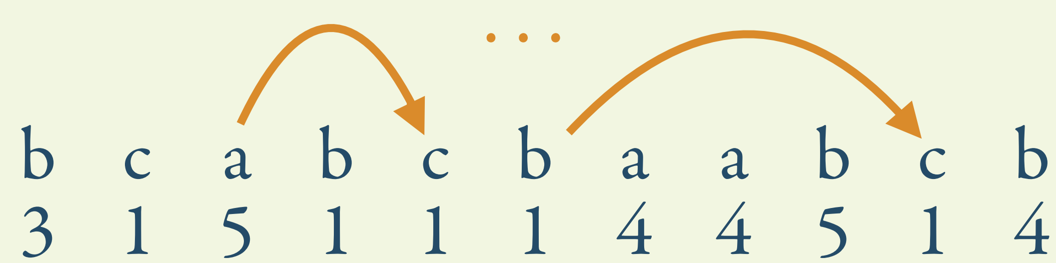
XPath on data words

b	c	a	b	c	b	a	a	b	c	b
3	1	5	1	1	1	4	4	5	1	4

eg: $\longrightarrow^*[\mathbf{b}]\longrightarrow[\mathbf{c}]$

Path logics

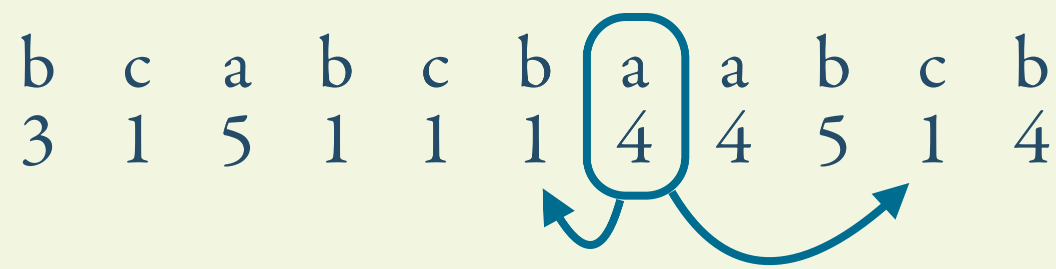
XPath on data words



eg: $\longrightarrow^{*}[\mathbf{b}]\longrightarrow[\mathbf{c}]$

Path logics

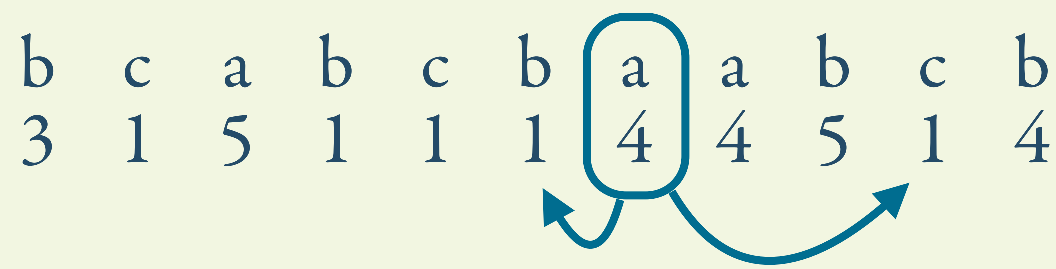
XPath on data words



eg: $\langle [\mathbf{b}]\longleftarrow \; = \; \longrightarrow^{*}[\mathbf{b}]\longrightarrow[\mathbf{c}] \rangle$

Path logics

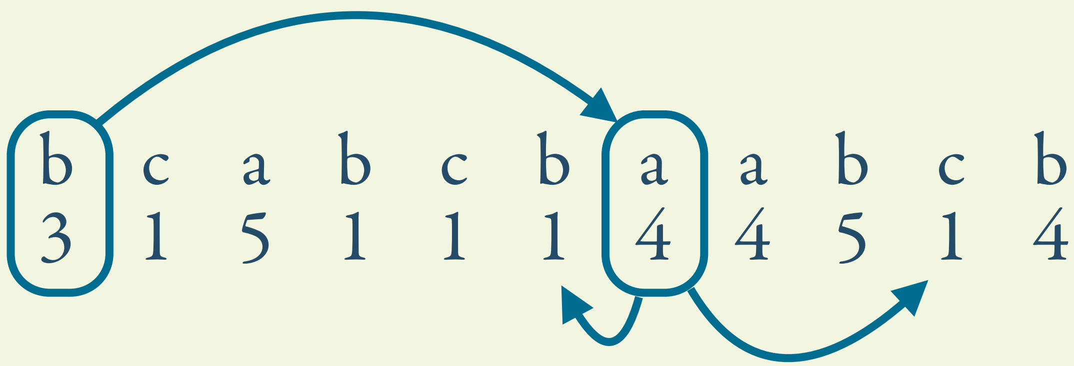
XPath on data words



eg: $\longrightarrow^*[\mathbf{a} \wedge \langle [\mathbf{b}]\longleftarrow \; = \; \longrightarrow^*[\mathbf{b}] \longrightarrow[\mathbf{c}] \rangle]$?

Path logics

XPath on data words



eg: $\longrightarrow^*[\mathbf{a} \wedge \langle [\mathbf{b}]\longleftarrow \;=\; \longrightarrow^*[\mathbf{b}]\longrightarrow[\mathbf{c}] \rangle]$?

Satisfiability of XPath on data words

XPath($\longrightarrow^{+}$, $^{+}\longleftarrow$): undecidable ♣

♠ [Demri, Lazić, 2006]

♣ [F., Segoufin, 2009]

Satisfiability of XPath on data words

XPath($\longrightarrow^{+}$, ${}^{+}\!\longleftarrow$): undecidable ♣

XPath($\longrightarrow^{+}$,${}^{*}\!\longleftarrow$): undecidable ♣

XPath($\longrightarrow$, $\longrightarrow^{*}$,${}^{*}\!\longleftarrow$): undecidable ♣

♠ [Demri, Lazić, 2006]

♣ [F., Segoufin, 2009]

Satisfiability of XPath on data words

XPath($\longrightarrow^{+}$, $^{+}\!\longleftarrow$): undecidable ♣

XPath($\longrightarrow^{+}$,$^{*}\!\longleftarrow$): undecidable ♣

XPath($\longrightarrow$, $\longrightarrow^{*}$,$^{*}\!\longleftarrow$): undecidable ♣

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

♠ [Demri, Lazić, 2006]

♣ [F., Segoufin, 2009]

Satisfiability of XPath on data words

XPath($\longrightarrow^{+}$, $^{+}\!\!\longleftarrow$): undecidable ♣

XPath($\longrightarrow^{+}$,$^{*}\!\!\longleftarrow$): undecidable ♣

XPath($\longrightarrow$, $\longrightarrow^{*}$,$^{*}\!\!\longleftarrow$): undecidable ♣

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

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

♠ [Demri, Lazić, 2006]

♣ [F., Segoufin, 2009]

Satisfiability of XPath on data words

XPath($\longrightarrow^+$, ${}^+\!\longleftarrow$): undecidable ♣

XPath($\longrightarrow^+$, ${}^*\!\longleftarrow$): undecidable ♣

XPath($\longrightarrow$, $\longrightarrow^*$, ${}^*\!\longleftarrow$): undecidable ♣

XPath($\longrightarrow^+$): decidable, non-PR ♠ ♣

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

why?

♠ [Demri, Lazić, 2006]

♣ [F., Segoufin, 2009]

Satisfiability of XPath on data words

XPath($\longrightarrow^+$, ${}^+\!\longleftarrow$): undecidable ♣

XPath($\longrightarrow^+$,${}^*\!\longleftarrow$): undecidable ♣

XPath($\longrightarrow$, $\longrightarrow^*$,${}^*\!\longleftarrow$): undecidable ♣

XPath($\longrightarrow^+$): decidable, non-PR ♠ ♣

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

why?

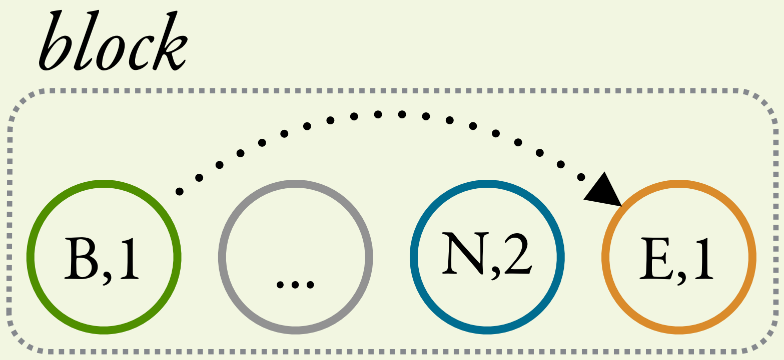
SAT-XPath($\longrightarrow$,$\longrightarrow^+$)

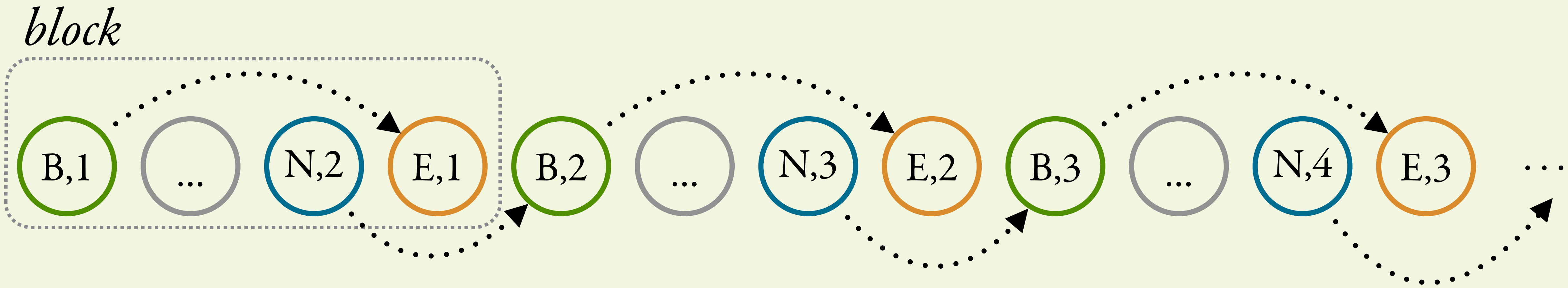
$\rightsquigarrow$

SAT-XPath($\longrightarrow^+$)

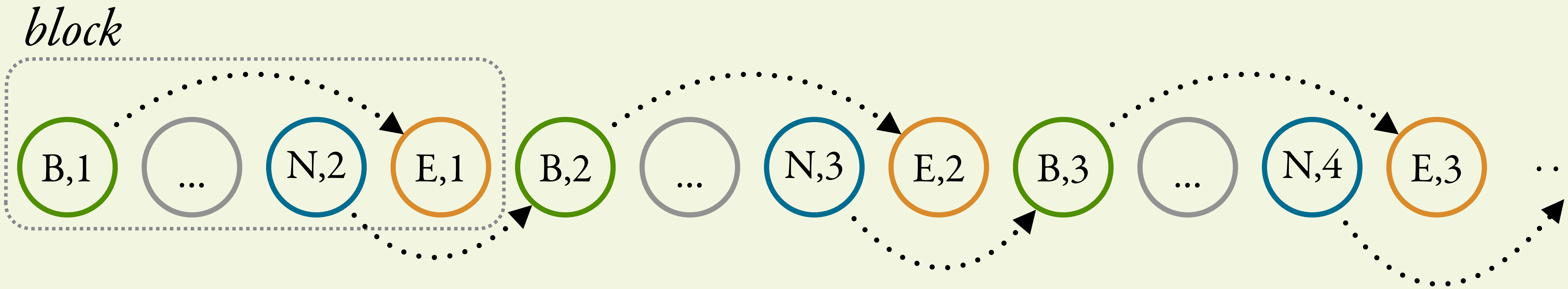
♠ [Demri, Lazić, 2006]

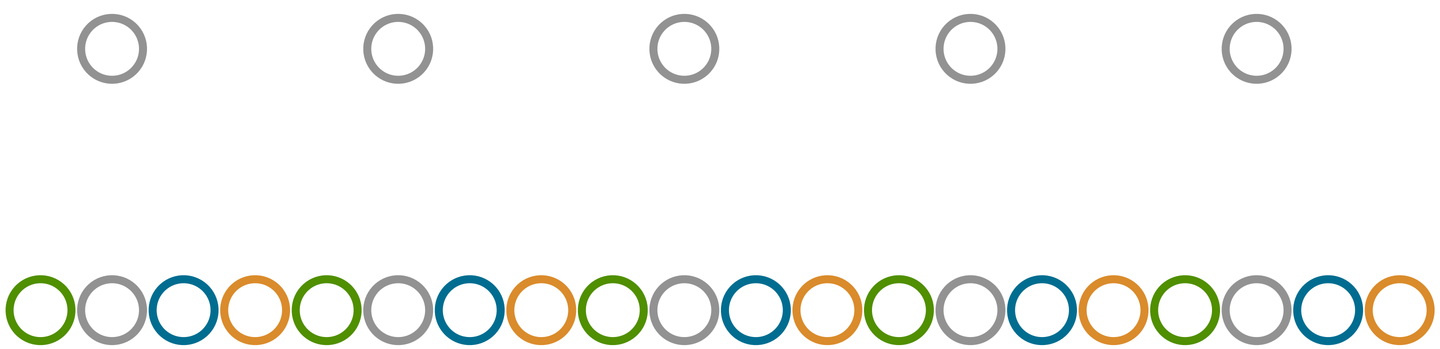
♣ [F., Segoufin, 2009]


block
B,1
...
N,2
E,1



block
B,1
...
N,2
E,1
B,2
...
N,3
E,2
B,3
...
N,4
E,3

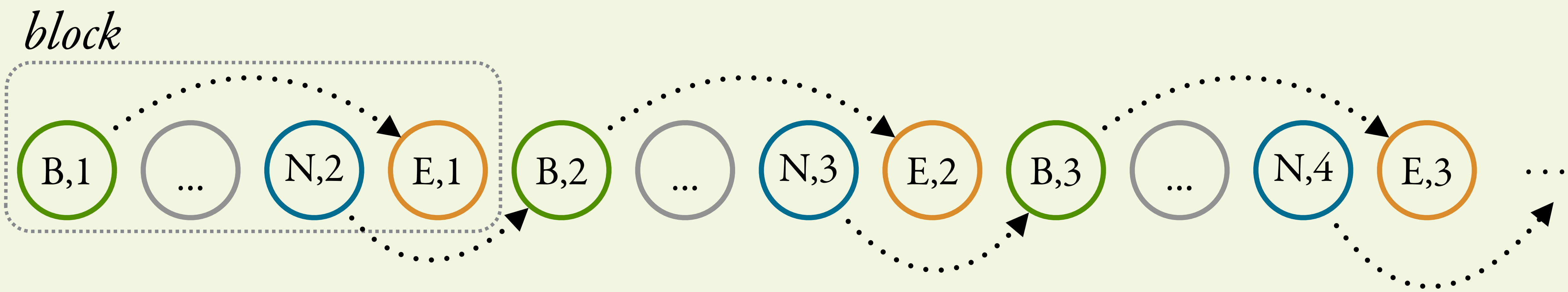




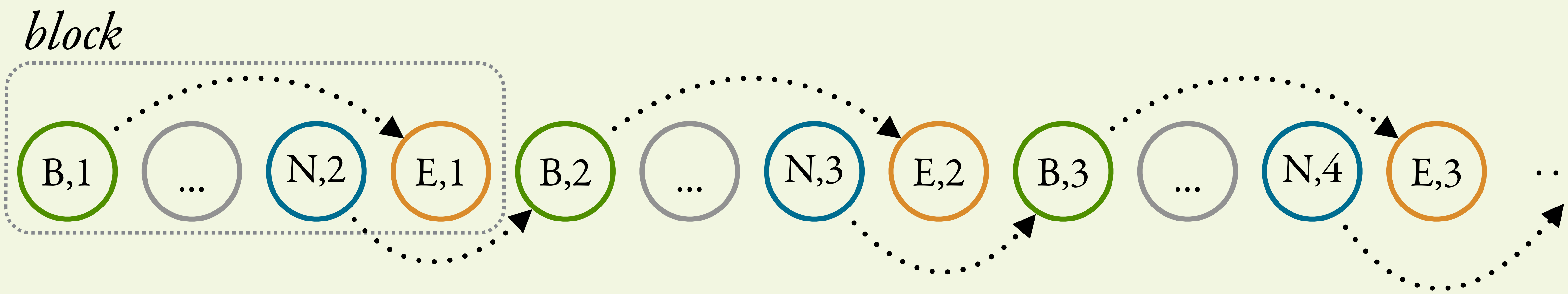
$\models \phi \in \mathrm{XPath}(\longrightarrow, \longrightarrow+)$

$\Updownarrow$

$\models \phi' \in \mathrm{XPath}(\longrightarrow+)$

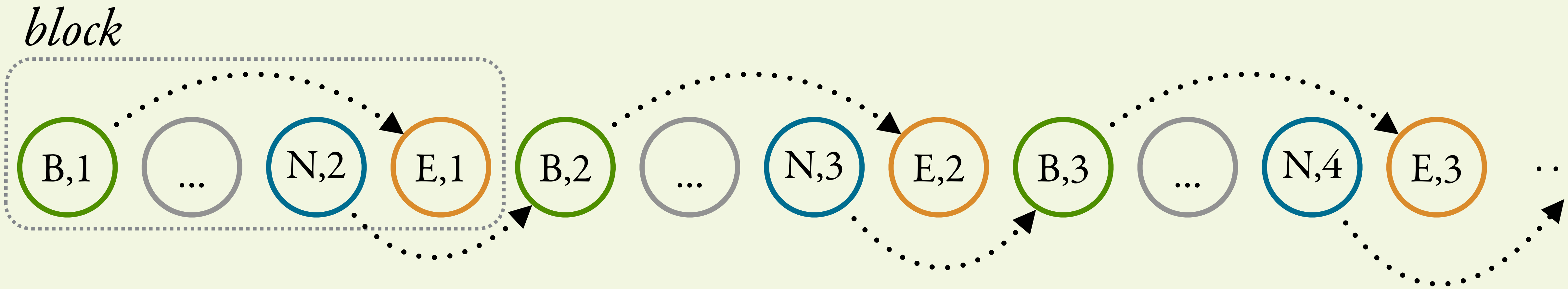


No more than one B,x , N,y , E,x per block



No more than one (B,x) , (N,y) , (E,x) per block

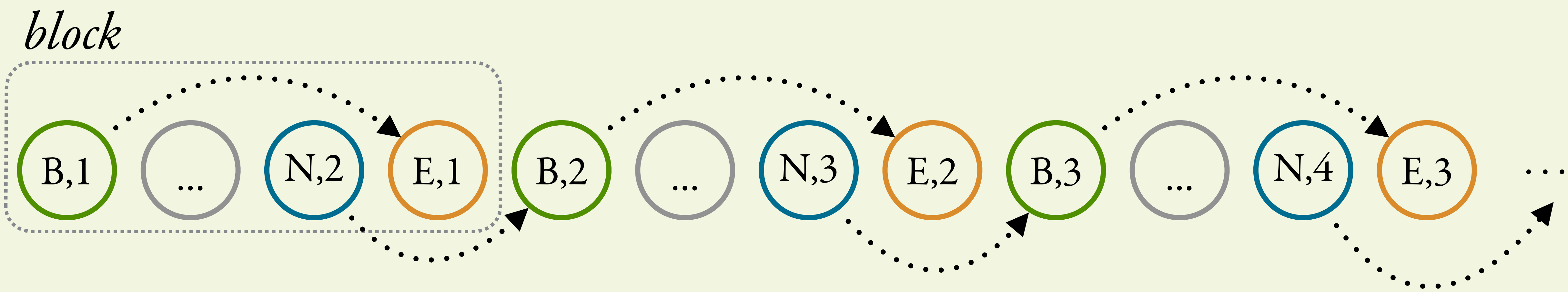
All (B,x) have different data (resp (N,y) , (E,x))



No more than one B,*x* , N,*y* , E,*x* per block

All B,*x* have different data (resp N,*y* , E,*x*)

N,*y* points to next block

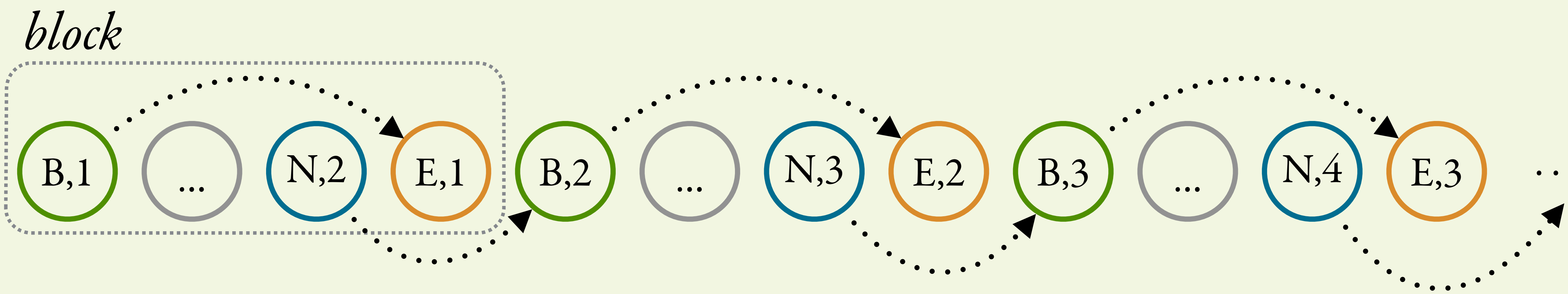


No more than one B,x , N,y , E,x per block

All B,x have different data (resp N,y , E,x)

N,y points to next block

For every B,x there is an E,x with same datum



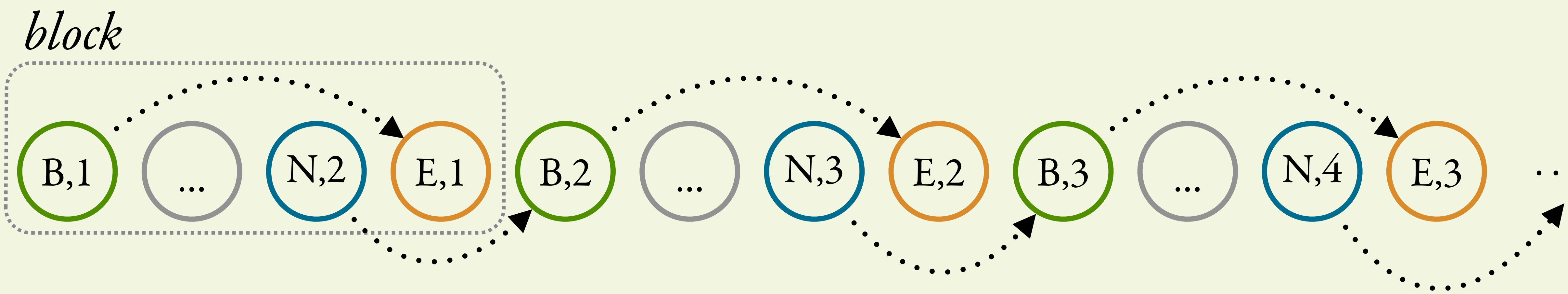
No more than one (B,x), (N,y), (E,x) per block

All (B,x) have different data (resp (N,y), (E,x))

(N,y) points to next block

For every (B,x) there is an (E,x) with same datum

Order in the block: (B,x) (...) (N,y) (E,x)



No more than one (B,x), (N,y), (E,x) per block

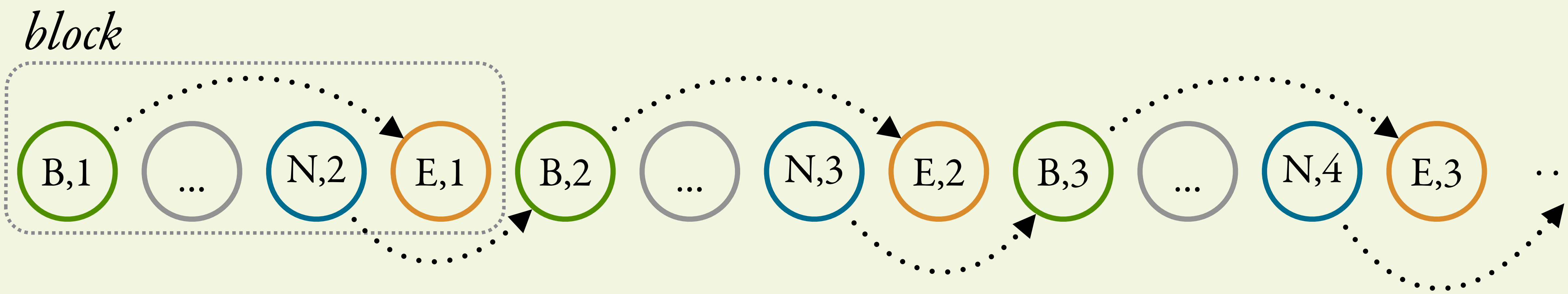
All (B,x) have different data (resp (N,y), (E,x))

(N,y) points to next block

For every (B,x) there is an (E,x) with same datum

Order in the block: (B,x) (...) (N,y) (E,x)

$$\text{next-block}(\phi) := \langle \varepsilon = \longrightarrow^{*}[N \wedge \langle \varepsilon = \longrightarrow^{*}[B \wedge \phi] \rangle]\longrightarrow^{*}[E]\rangle$$



$\longrightarrow^+$ $\longrightarrow^*$

✔ No more than one (B,x), (N,y), (E,x) per block

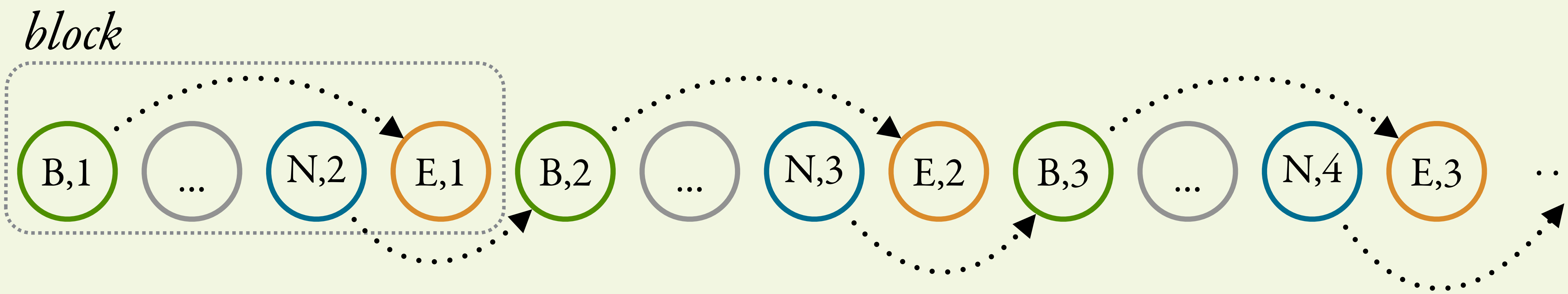
✔ All (B,x) have different data (resp (N,y), (E,x))

✔ (N,y) points to next block

✔ For every (B,x) there is an (E,x) with same datum

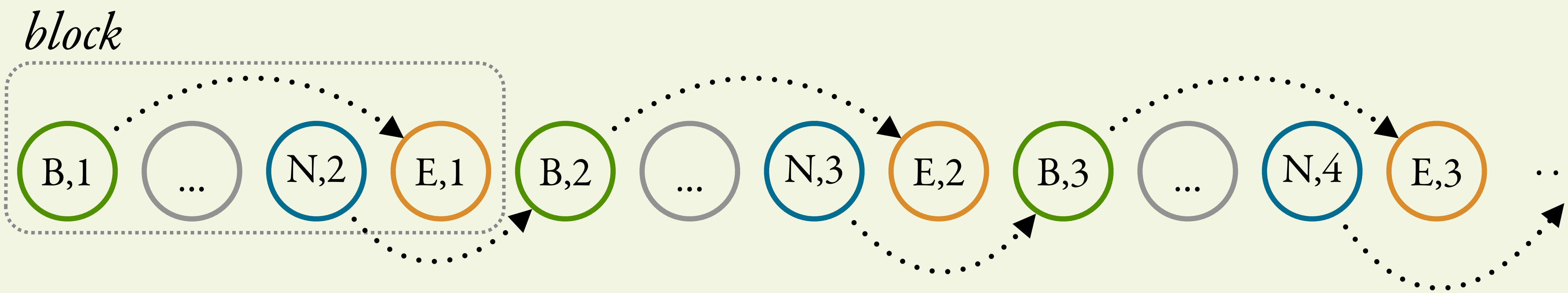
✔ Order in the block: (B,x) (...) (N,y) (E,x)

$$\text{next-block}(\phi) := \langle \varepsilon = \longrightarrow^*[N \wedge \langle \varepsilon = \longrightarrow^*[B \wedge \phi] \rangle]\longrightarrow^*[E]\rangle$$



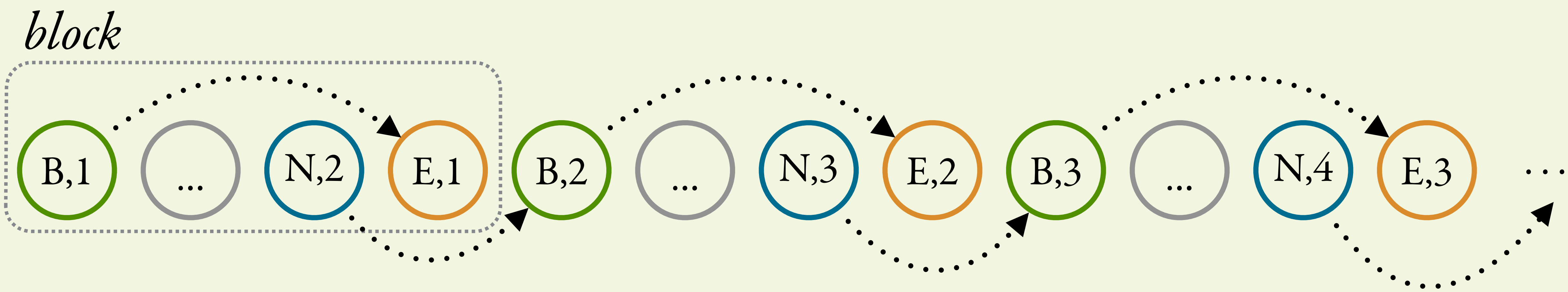
$\longrightarrow^+$	$\longrightarrow^*$	
✔		No more than one (B,x) , (N,y) , (E,x) per block
✔		All (B,x) have different data (resp (N,y) , (E,x))
✔		(N,y) points to next block
✔	✔	For every (B,x) there is an (E,x) with same datum
✔	✔	Order in the block: (B,x) (...) (N,y) (E,x)

$$\text{next-block}(\phi) := \langle \varepsilon = \longrightarrow^*[N \wedge \langle \varepsilon = \longrightarrow^*[B \wedge \phi] \rangle]\longrightarrow^*[E]\rangle$$



$\longrightarrow^+$	$\longrightarrow^*$	
✔	✘	No more than one (B,x), (N,y), (E,x) per block
✔	✘	All (B,x) have different data (resp (N,y), (E,x))
✔		(N,y) points to next block
✔	✔	For every (B,x) there is an (E,x) with same datum
✔	✔	Order in the block: (B,x) (...) (N,y) (E,x)

$$\text{next-block}(\phi) := \langle \varepsilon = \longrightarrow^*[N \wedge \langle \varepsilon = \longrightarrow^*[B \wedge \phi] \rangle]\longrightarrow^*[E]\rangle$$



$\longrightarrow^+$	$\longrightarrow^*$	
✔	✘	No more than one B,x , N,y , E,x per block
✔	✘	All B,x have different data (resp N,y , E,x)
✔	✘	N,y points to next block
✔	✔	For every B,x there is an E,x with same datum
✔	✔	Order in the block: B,x ... N,y E,x

$$\text{next-block}(\phi) := \langle \varepsilon = \longrightarrow^*[N \wedge \langle \varepsilon = \longrightarrow^*[B \wedge \phi] \rangle]\longrightarrow^*[E]\rangle$$

Satisfiability of XPath on data words

XPath($\longrightarrow^+$, $^+\!\longleftarrow$): undecidable ♣

XPath($\longrightarrow^+$,$^*\!\longleftarrow$): undecidable ♣

XPath($\longrightarrow$, $\longrightarrow^*$,$^*\!\longleftarrow$): undecidable ♣

XPath($\longrightarrow^+$): decidable, non-PR ♠ ♣

In particular, any fragment with $\longrightarrow^+$ or $^+\!\longleftarrow$ is **undecidable** or has a **non-PR complexity**

What about XPath($\longrightarrow^*$)?

♠ [Demri, Lazić, 2006]

♣ [F., Segoufin, 2009]

Satisfiability of XPath on data words

XPath($\longrightarrow^+$, $^+\longleftarrow$): undecidable ♣

XPath($\longrightarrow^+$,$^*\longleftarrow$): undecidable ♣

XPath($\longrightarrow$, $\longrightarrow^*$,$^*\longleftarrow$): undecidable ♣

XPath($\longrightarrow^+$): decidable, non-PR ♠ ♣

In particular, any fragment with $\longrightarrow^+$ or $^+\longleftarrow$ is **undecidable** or has a **non-PR complexity**

What about XPath($\longrightarrow^*$)?

XPath($\longrightarrow^*$) is **decidable** in 2ExpSpace ♦

XPath($\longrightarrow^*$,$^*\longleftarrow$) is **decidable** in 2ExpSpace ♦

♦ [F., 2011]

♠ [Demri, Lazić, 2006]

♣ [F., Segoufin, 2009]

Satisfiability of XPath on data words

XPath($\longrightarrow^+$, $^+\!\longleftarrow$): undecidable ♣

XPath($\longrightarrow^+$,$^*\!\longleftarrow$): undecidable ♣

XPath($\longrightarrow$, $\longrightarrow^*$,$^*\!\longleftarrow$): undecidable ♣

XPath($\longrightarrow^+$): decidable, non-PR ♠ ♣

In particular, any fragment with $\longrightarrow^+$ or $^+\!\longleftarrow$ is **undecidable** or has a **non-PR complexity**

What about XPath($\longrightarrow^*$)?

XPath($\longrightarrow^*$) is **decidable** in 2ExpSpace ♦

XPath($\longrightarrow^*$,$^*\!\longleftarrow$) is **decidable** in 2ExpSpace ♦

♦ [F., 2011]

♠ [Demri, Lazić, 2006]

♣ [F., Segoufin, 2009]

why?

SAT-XPath($\longrightarrow^*$,$^*\longleftarrow$) decidable in 2ExpSpace

Proof idea:

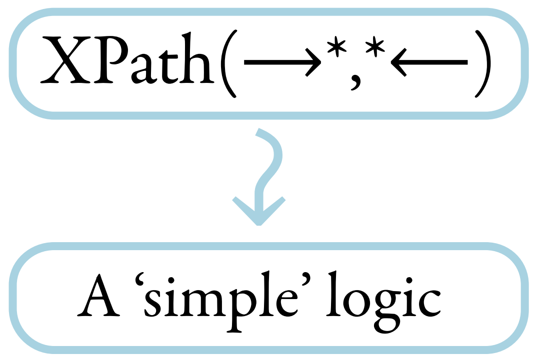
XPath($\longrightarrow^*$,$^*\longleftarrow$)

⇓

A 'simple' logic ...in *'Scott Normal Form'*

SAT-XPath($\longrightarrow^*$,$^*\longleftarrow$) decidable in 2ExpSpace

Proof idea:



...in *'Scott Normal Form'*

- There is a position labelled '**a**'.

SAT-XPath($\longrightarrow^*$,$^*\longleftarrow$) decidable in 2ExpSpace

Proof idea:

XPath($\longrightarrow^*$,$^*\longleftarrow$)

A 'simple' logic ...in *'Scott Normal Form'*

- There is a position labelled '**a**'.
- For every position labelled '**a**', a BC of:

$$\langle\ [\phi_n]^*\!\longleftarrow \cdots {}^*\!\longleftarrow[\phi_1]^*\!\longleftarrow[\phi_0] \ = \ \longrightarrow^*[\psi_0]\longrightarrow^*[\psi_1]\longrightarrow^* \cdots \longrightarrow^*[\psi_m]\ \rangle$$

$\phi_i, \psi_i \in \mathrm{BC}(\mathbf{A})$

$d \in \mathbf{D}$

SAT-XPath($\longrightarrow^*$,$^*\longleftarrow$) decidable in 2ExpSpace

Proof idea:

XPath($\longrightarrow^*$,$^*\longleftarrow$)

$\downarrow$

A ‘simple’ logic ...in *‘Scott Normal Form’*

- There is a position labelled ‘**a**’.
- For every position labelled ‘**a**’, a BC of:

$$\langle\, [\phi_n]^*\!\longleftarrow \cdots {}^*\!\longleftarrow [\phi_1]^*\!\longleftarrow [\phi_0] \;=\; \longrightarrow^*[\psi_0]\longrightarrow^*[\psi_1]\longrightarrow^* \cdots \longrightarrow^*[\psi_m] \,\rangle$$

$$\langle\, [\phi_n]\longrightarrow^* \cdots \longrightarrow^*[\phi_1]\longrightarrow^*[\phi_0] \;=\; \longrightarrow^*[\psi_0]\longrightarrow^*[\psi_1]\longrightarrow^* \cdots \longrightarrow^*[\psi_m] \,\rangle$$

$\phi_i, \psi_i \in \mathrm{BC}(\mathbf{A})$

$d \in \mathbf{D}$

SAT-XPath($\longrightarrow^*, {}^*\longleftarrow$) decidable in 2ExpSpace

Proof idea:

XPath($\longrightarrow^*, {}^*\longleftarrow$)

↓

A 'simple' logic ...in *'Scott Normal Form'*

- There is a position labelled '**a**'.
- For every position labelled '**a**', a BC of:

$$\langle\ [\phi_n]^*\!\longleftarrow \cdots {}^*\!\longleftarrow[\phi_1]^*\!\longleftarrow[\phi_0] \ = \ \longrightarrow^*[\psi_0]\longrightarrow^*[\psi_1]\longrightarrow^* \cdots \longrightarrow^*[\psi_m]\ \rangle$$

$$\langle\ [\phi_n]\longrightarrow^* \cdots \longrightarrow^*[\phi_1]\longrightarrow^*[\phi_0] \ = \ \longrightarrow^*[\psi_0]\longrightarrow^*[\psi_1]\longrightarrow^* \cdots \longrightarrow^*[\psi_m]\ \rangle$$

$$\langle\ d \ = \ \longrightarrow^*[\psi_0]\longrightarrow^*[\psi_1]\longrightarrow^* \cdots \longrightarrow^*[\psi_m]\ \rangle$$

$\phi_i, \psi_i \in \mathrm{BC}(\mathbf{A})$

$d \in \mathbf{D}$

SAT-XPath($\longrightarrow^*$,$^*\longleftarrow$) decidable in 2ExpSpace

Proof idea:

XPath($\longrightarrow^*$,$^*\longleftarrow$)

↓

A 'simple' logic ...in *'Scott Normal Form'*

- There is a position labelled '**a**'.
- For every position labelled '**a**', a BC of:

$$\langle\ [\phi_n]^*\!\longleftarrow \cdots {}^*\!\longleftarrow[\phi_1]^*\!\longleftarrow[\phi_0]\ =\ \longrightarrow^*[\psi_0]\longrightarrow^*[\psi_1]\longrightarrow^* \cdots \longrightarrow^*[\psi_m]\ \rangle$$

$$\langle\ [\phi_n]\longrightarrow^* \cdots \longrightarrow^*[\phi_1]\longrightarrow^*[\phi_0]\ =\ \longrightarrow^*[\psi_0]\longrightarrow^*[\psi_1]\longrightarrow^* \cdots \longrightarrow^*[\psi_m]\ \rangle$$

$$\langle\ d\ =\ \longrightarrow^*[\psi_0]\longrightarrow^*[\psi_1]\longrightarrow^* \cdots \longrightarrow^*[\psi_m]\ \rangle$$

$$\vdots$$

$\phi_i, \psi_i \in \mathrm{BC}(\mathbf{A})$

$d \in \mathbf{D}$

SAT-XPath($\longrightarrow^*, {}^*\!\longleftarrow$) decidable in 2ExpSpace

Proof idea:

XPath($\longrightarrow^*, {}^*\!\longleftarrow$)

↓

A 'simple' logic ...in *'Scott Normal Form'*

- There is a position labelled '**a**'.
- For every position labelled '**a**', a BC of:

$$\langle\, [\phi_n]^*\!\longleftarrow \cdots {}^*\!\longleftarrow [\phi_1]^*\!\longleftarrow [\phi_0] \overset{\neq}{=} \longrightarrow^*[\psi_0]\longrightarrow^*[\psi_1]\longrightarrow^* \cdots \longrightarrow^*[\psi_m]\,\rangle$$

$$\langle\, [\phi_n]\longrightarrow^* \cdots \longrightarrow^*[\phi_1]\longrightarrow^*[\phi_0] \overset{\neq}{=} \longrightarrow^*[\psi_0]\longrightarrow^*[\psi_1]\longrightarrow^* \cdots \longrightarrow^*[\psi_m]\,\rangle$$

$$\langle\, d \overset{\neq}{=} \longrightarrow^*[\psi_0]\longrightarrow^*[\psi_1]\longrightarrow^* \cdots \longrightarrow^*[\psi_m]\,\rangle$$

$$\vdots$$

$\phi_i, \psi_i \in \mathrm{BC}(\mathbf{A})$

$d \in \mathbf{D}$

SAT-XPath($\longrightarrow^*$,$^*\!\longleftarrow$) decidable in 2ExpSpace

Proof idea:

XPath($\longrightarrow^*$,$^*\!\longleftarrow$)

A 'simple' logic ...in *'Scott Normal Form'*

- There is a position labelled '**a**'.
- For every position labelled '**a**', a BC of:

"$\overleftarrow{\phi_n \bullet \cdots \bullet \phi_1 \bullet \phi_0} = \overrightarrow{\psi_0 \bullet \psi_1 \bullet \cdots \bullet \psi_m}$"

$$\langle\ [\phi_n]^*\!\longleftarrow \cdots {}^*\!\longleftarrow[\phi_1]^*\!\longleftarrow[\phi_0] \ \overset{\neq}{=}\ \longrightarrow^*[\psi_0]\longrightarrow^*[\psi_1]\longrightarrow^* \cdots \longrightarrow^*[\psi_m]\ \rangle$$

$$\langle\ [\phi_n]\longrightarrow^* \cdots \longrightarrow^*[\phi_1]\longrightarrow^*[\phi_0] \ \overset{\neq}{=}\ \longrightarrow^*[\psi_0]\longrightarrow^*[\psi_1]\longrightarrow^* \cdots \longrightarrow^*[\psi_m]\ \rangle$$

$$\langle\ d \ \overset{\neq}{=}\ \longrightarrow^*[\psi_0]\longrightarrow^*[\psi_1]\longrightarrow^* \cdots \longrightarrow^*[\psi_m]\ \rangle$$

$$\vdots$$

$\phi_i, \psi_i \in \mathrm{BC}(\mathbf{A})$

$d \in \mathbf{D}$

ϕ

there is only one dv under a **c**

for every **a**, there is a **b** accessible via a **c** with the same dv

there is a position labeled **c**

subpaths of ϕ = {**c** • **b**, **c**, **b**}

ϕ

there is only one dv under a **c**

for every **a**, there is a **b** accessible via a **c** with the same dv

there is a position labeled **c**

subpaths of ϕ = {**c** • **b**, **c**, **b**}

w

a	b	b	a	b	c	a	b	c	c	a	a	b	c	b	$\models \phi$
1	2	4	4	3	1	5	1	1	1	4	4	5	1	4	

ϕ

there is only one dv under a **c**

for every **a**, there is a **b** accessible via a **c** with the same dv

there is a position labeled **c**

subpaths of ϕ = {**c** • **b**, **c**, **b**}

w

a	b	b	b	a	a	b	c	a	b	c	c	a	a	a	a	b	c	b	$\models \phi$
1	2	4	9	4	9	3	1	5	1	1	1	4	9	4	9	5	1	4	

ϕ

there is only one dv under a **c**

for every **a**, there is a **b** accessible via a **c** with the same dv

there is a position labeled **c**

subpaths of ϕ = {**c** • **b**, **c**, **b**}

w

a	b	b	a	b	c	[a	b	c	c]	a	a	b	c	b
1	2	4	4	3	1	5	1	1	1	4	4	5	1	4
						i			j					

$\models \phi$

ϕ

there is only one dv under a **c**

for every **a**, there is a **b** accessible via a **c** with the same dv

there is a position labeled **c**

subpaths of ϕ = {**c** • **b**, **c**, **b**}

w

a	b	b	a	b	c	[a	b	c	c]	a	a	b	c	b	$\models \phi$
1	2	4	4	3	1	[5	1	1	1]	4	4	5	1	4	
						i			j						

π

profile
- internal — reachable data
  - from the left [-->]
  - from the right [<--]
- external — reachable data
  - from the left <--[ ]
  - from the right [ ]-->

ϕ

there is only one dv under a **c**

for every **a**, there is a **b** accessible via a **c** with the same dv

there is a position labeled **c**

subpaths of ϕ = {**c** • **b**, **c**, **b**}

w

a	b	b	a	b	c	[a	b	c	c]	a	a	b	c	b	$\models \phi$
1	2	4	4	3	1	[5	1	1	1]	4	4	5	1	4	
					i				j						

π

profile
- internal
  - reachable data
    - from the left [-->]
    - from the right [<--]
- external
  - reachable data
    - from the left <--[ ]
    - from the right [ ]-->

abs(w,i,j) = π =

[-->] : { (1,c), (1,b) }

[<--] : { (1,c), (1,c•b), (1,b) }

[ ]--> : { (5,b), (4,b), (4,c•b), (1,c) }

<--[ ] : { (1,c), (3,c•b), (4,c•b), (4,b), (3,b) (2,b) }

ϕ

there is only one dv under a **c**

for every **a**, there is a **b** accessible via a **c** with the same dv

there is a position labeled **c**

subpaths of ϕ = {**c** • **b**, **c**, **b**}

w

a	b	b	a	b	c	[a	b	c	c]	a	a	b	c	b
1	2	4	4	3	1	5	1	1	1	4	4	5	1	4

i j $\models \phi$

π

profile

- internal
  - reachable data
    - from the left [-->]
    - from the right [<--]
- external
  - reachable data
    - from the left <--[ ]
    - from the right [ ]-->

abs(w,i,j) = π =

[-->] : { (1,c), (1,b) }

[<--] : { (1,c), (1,c•b), (1,b) }

[ ]--> : { (5,b), (4,b), (4,c•b), (1,c) }

<--[ ] : { (1,c), (3,c•b), (4,c•b), (4,b), (3,b) (2,b) }

$\leq$

ϕ

there is only one dv under a **c**

for every **a**, there is a **b** accessible via a **c** with the same dv

there is a position labeled **c**

subpaths of ϕ = {**c • b**, **c**, **b**}

w

a	b	b	b	a	a	b	c	[a	b	c	c]	a	a	a	a	b	c	b		
1	2	4	9	4	9	3	1	[5	1	1	1]	4	9	4	9	5	1	4		
								i			j									

$\models \phi$

π

profile
- internal
  - reachable data
    - from the left [-->]
    - from the right [<--]
- external
  - reachable data
    - from the left <--[ ]
    - from the right [ ]-->

abs(w,i,j) = π =

[-->] : { (1,c), (1,b) } ∪ { }

[<--] : { (1,c), (1,c•b), (1,b) } ∪ { }

[ ]--> : { (5,b), (4,b), (4,c•b), (1,c) } ∪ { (9,b), (9,c•b) }

<--[ ] : { (1,c), (3,c•b), (4,c•b), (4,b), (3,b) (2,b) } ∪ { (9,c•b), (9,b) }

$\leq$

ϕ

there is only one dv under a **c**

for every **a**, there is a **b** accessible via a **c** with the same dv

there is a position labeled **c**

subpaths of ϕ = {**c** • **b**, **c**, **b**}

w

a	b	b	b	a	a	b	c	[a	b	c	c]	a	a	a	a	b	c	b
1	2	4	9	4	9	3	1	[5	1	1	1]	4	9	4	9	5	1	4
								i			j							

$\models \phi$

π

profile
- internal
  - reachable data
    - from the left [-->]
    - from the right [<--]
- external
  - reachable data
    - from the left <--[ ]
    - from the right [ ]-->

abs(w,i,j) = π =

[-->] : { (1,c), (1,b) } ∪ { }

[<--] : { (1,c), (1,c•b), (1,b) } ∪ { }

[ ]--> : { (5,b), (4,b), (4,c•b), (1,c) } ∪ { (9,b), (9,c•b) }

<--[ ] : { (1,c), (3,c•b), (4,c•b), (4,b), (3,b) (2,b) } ∪ { (9,c•b), (9,b) }

$\leq$

Induces a **wqo** ≤ on the profiles

→ every $\pi_1, \pi_2, \pi_3, \ldots$ has some $i<j$ with $\pi_i \leq \pi_j$

Atomic profiles

a	b	b	a	b	c	[a]	b	c	c	a	a	b	c	b
1	2	4	4	3	1	[5]	1	1	1	4	4	5	1	4

$\pi = \mathrm{abs}(\boldsymbol{w}, i, i{+}1)$ (compatible with ϕ)

Atomic$_\phi$ = all ϕ-abstractions of positions within data words

Concatenation of profiles

a	b	b	a	b	c	a	b	c	c	a	a	b	c	b
1	2	4	4	3	1	5	1	1	1	4	4	5	1	4

Atomic profiles

a	b	b	a	b	c	[a]	b	c	c	a	a	b	c	b
1	2	4	4	3	1	[5]	1	1	1	4	4	5	1	4

$\pi = \text{abs}(\boldsymbol{w}, i, i+1)$ (compatible with ϕ)

Atomic$_\phi$ = all ϕ-abstractions of positions within data words

Concatenation of profiles

a	b	b	[a	b	c]	a	b	c	c	a	a	b	c	b
1	2	4	[4	3	1]	5	1	1	1	4	4	5	1	4

π_1

Atomic profiles

a	b	b	a	b	c	[a]	b	c	c	a	a	b	c	b
1	2	4	4	3	1	[5]	1	1	1	4	4	5	1	4

$\pi = \mathrm{abs}(w, i, i{+}1)$ (compatible with ϕ)

Atomic$_\phi$ = all ϕ-abstractions of positions within data words

Concatenation of profiles

a	b	b	[a	b	c]	[a	b	c	c]	a	a	b	c	b
1	2	4	4	3	1	5	1	1	1	4	4	5	1	4

π_1 π_2

Atomic profiles

a	b	b	a	b	c	[a]	b	c	c	a	a	b	c	b
1	2	4	4	3	1	[5]	1	1	1	4	4	5	1	4

$\pi \;=\; \mathrm{abs}(\boldsymbol{w}, i, i{+}1)$ (compatible with ϕ)

Atomic$_\phi$ = all ϕ-abstractions of positions within data words

Concatenation of profiles

a	b	b	[[a	b	c]	[a	b	c	c]]	a	a	b	c	b
1	2	4	[[4	3	1]	[5	1	1	1]]	4	4	5	1	4

$\pi_3 \;=\; \pi_1 \cdot \pi_2$

Atomic profiles

a	b	b	a	b	c	[a]	b	c	c	a	a	b	c	b
1	2	4	4	3	1	[5]	1	1	1	4	4	5	1	4

$\pi = \mathrm{abs}(w, i, i+1)$ (compatible with ϕ)

$\textbf{Atomic}_\phi$ = all ϕ-abstractions of positions within data words

Concatenation of profiles

a	b	b	[[a	b	c]	[a	b	c	c]]	a	a	b	c	b
1	2	4	[[4	3	1]	[5	1	1	1]]	4	4	5	1	4

$$\pi_3 = \pi_1 \cdot \pi_2$$

a) a profile is an abstraction of a model ⇔ it is derivable (from atomic profiles) and complete

Der

[ ]--> : ∅

<--[ ] : ∅

Atomic profiles

a	b	b	a	b	c	[a]	b	c	c	a	a	b	c	b
1	2	4	4	3	1	[5]	1	1	1	4	4	5	1	4

$\pi = \text{abs}(w, i, i+1)$ (compatible with ϕ)

Atomic$_\phi$ = all ϕ-abstractions of positions within data words

Concatenation of profiles

a	b	b	[[a	b	c]	[a	b	c	c]]	a	a	b	c	b
1	2	4	[[4	3	1]	[5	1	1	1]]	4	4	5	1	4

$\pi_3 = \pi_1 \cdot \pi_2$

a) a profile is an abstraction of a model ⇔ it is derivable (from atomic profiles) and complete

Der

[ ]--> : ∅

<--[ ] : ∅

b) abs(w,0,|w|) determines whether $w \models \phi$

Atomic profiles

a	b	b	a	b	c	[a]	b	c	c	a	a	b	c	b
1	2	4	4	3	1	[5]	1	1	1	4	4	5	1	4

$\pi = \mathrm{abs}(w, i, i{+}1)$ (compatible with ϕ)

Atomic$_\phi$ = all ϕ-abstractions of positions within data words

Concatenation of profiles

a	b	b	[[a	b	c]	[a	b	c	c]]	a	a	b	c	b
1	2	4	[[4	3	1]	[5	1	1	1]]	4	4	5	1	4

$\pi_3 = \pi_1 \cdot \pi_2$

a) a profile is an abstraction of a model ⇔ it is derivable (from atomic profiles) and complete

Der

[ ]--> : ∅

<--[ ] : ∅

b) abs(w,0,|w|) determines whether $w \models \phi$

$a{+}b$: SAT(ϕ) ⇔ There is complete $\pi \in$ **Der** so that $\pi \models \phi$

Atomic profiles

a	b	b	a	b	c	[a]	b	c	c	a	a	b	c	b
1	2	4	4	3	1	[5]	1	1	1	4	4	5	1	4

$\pi = \mathrm{abs}(\boldsymbol{w}, i, i{+}1)$ (compatible with ϕ)

Atomic$_\phi$ = all ϕ-abstractions of positions within data words

Concatenation of profiles

a	b	b	[[a	b	c]	[a	b	c	c]]	a	a	b	c	b
1	2	4	[[4	3	1]	[5	1	1	1]]	4	4	5	1	4

$\pi_3 = \pi_1 \cdot \pi_2$

a) a profile is an abstraction of a model ⇔ it is derivable (from atomic profiles) and complete

Der

[ ]--→ : ∅

←--[ ] : ∅

b) $\mathrm{abs}(\boldsymbol{w},0,|\boldsymbol{w}|)$ determines whether $w \models \phi$

$a{+}b$: SAT(ϕ) ⇔ There is complete $\pi \in$ **Der** so that $\pi \models \phi$

How to compute **Der***?*

1 By property before: $\mathbf{Der} = \uparrow \mathbf{Der}$

$$\{ \pi \mid \pi' \leq \pi, \pi' \in \mathbf{Der} \}$$

1 By property before: **Der** = $\uparrow$**Der**

$\{ \pi \mid \pi' \leq \pi , \pi' \in \mathbf{Der} \}$

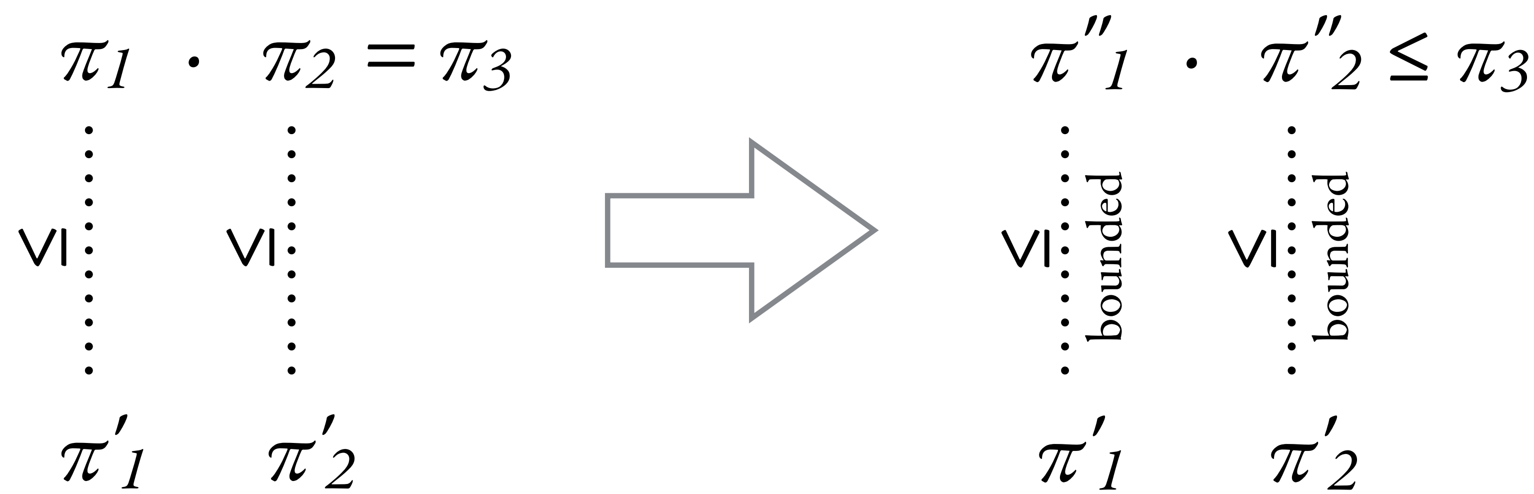
2 MIN(**Atomic**) is finite and computable

1 By property before: **Der** = ↑**Der**

$\{ \pi \mid \pi' \leq \pi, \pi' \in \mathbf{Der} \}$

2 MIN(**Atomic**) is finite and computable

3

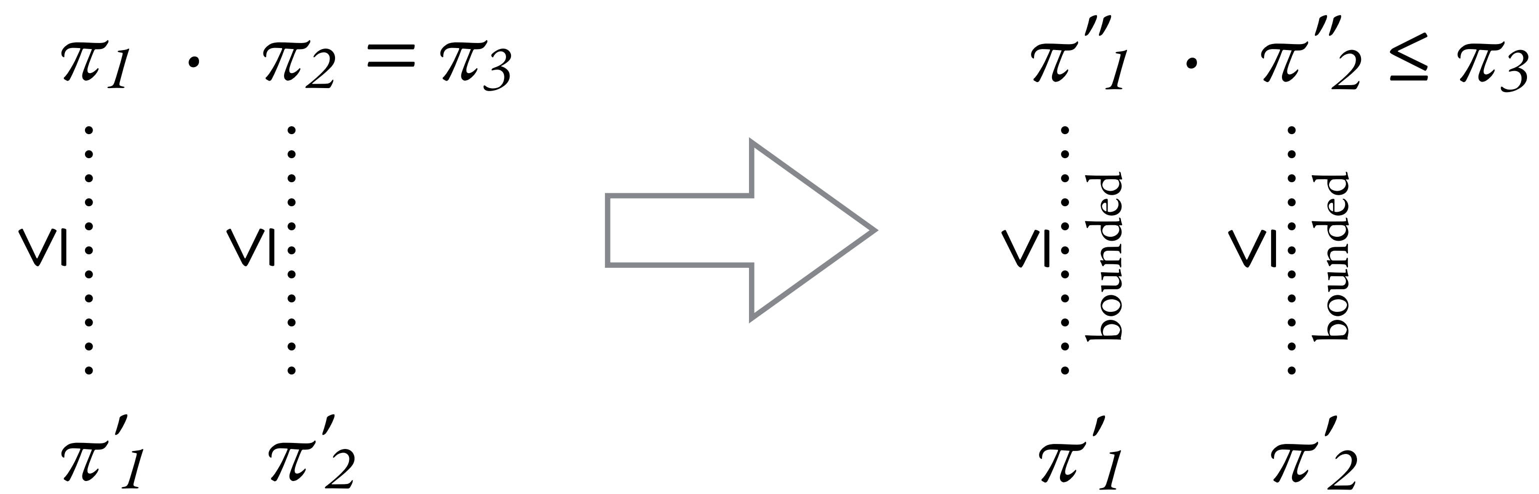


1 By property before: **Der** = ↑**Der**

$\{\pi \mid \pi' \leq \pi, \pi' \in \textbf{Der}\}$

2 MIN(**Atomic**) is finite and computable

3



1+2 =

```
R0 = MIN( Atomic ) ;
while (Ri ≠ Ri+1) :
    Ri+1 = MIN(Ri ∪ ↑Ri · ↑Ri)

return( Ri );
```

computes MIN(Der)

1 By property before: **Der** = ↑**Der**

$\uparrow \mathbf{Der} = \{ \pi \mid \pi' \leq \pi , \pi' \in \mathbf{Der} \}$

2 MIN(**Atomic**) is finite and computable

3

$$\pi_1 \cdot \pi_2 = \pi_3 \quad \Rightarrow \quad \pi''_1 \cdot \pi''_2 \leq \pi_3$$

$\pi_1 \geq \pi'_1$, $\pi_2 \geq \pi'_2$ ⇒ $\pi''_1 \geq \pi'_1$ (bounded), $\pi''_2 \geq \pi'_2$ (bounded)

1+2 =

```
R0 = MIN( Atomic ) ;
while (Ri ≠ Ri+1) :
    Ri+1 = MIN(Ri ∪ ↑Ri · ↑Ri)

return( Ri );
```

computes MIN(Der)

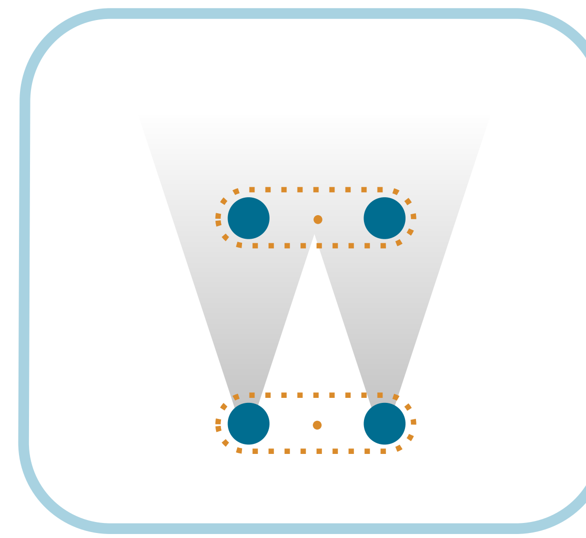
1 By property before: **Der** = ↑**Der**

$\{ \pi \mid \pi' \leq \pi , \pi' \in \textbf{Der} \}$

2 MIN(**Atomic**) is finite and computable

3

$$\pi_1 \cdot \pi_2 = \pi_3 \quad (\pi_1 \geq \pi'_1,\ \pi_2 \geq \pi'_2) \quad \Rightarrow \quad \pi''_1 \cdot \pi''_2 \leq \pi_3 \quad (\pi''_1 \geq \pi'_1 \text{ bounded},\ \pi''_2 \geq \pi'_2 \text{ bounded})$$



1+2 =

```
R0 = MIN( Atomic ) ;
while (Ri ≠ Ri+1) :
    Ri+1 = MIN(Ri ∪ ↑Ri · ↑Ri)

return( Ri );
```

computes MIN(Der)

1 By property before: **Der** = ↑**Der**

$$\{ \pi \mid \pi' \leq \pi , \pi' \in \textbf{Der} \}$$

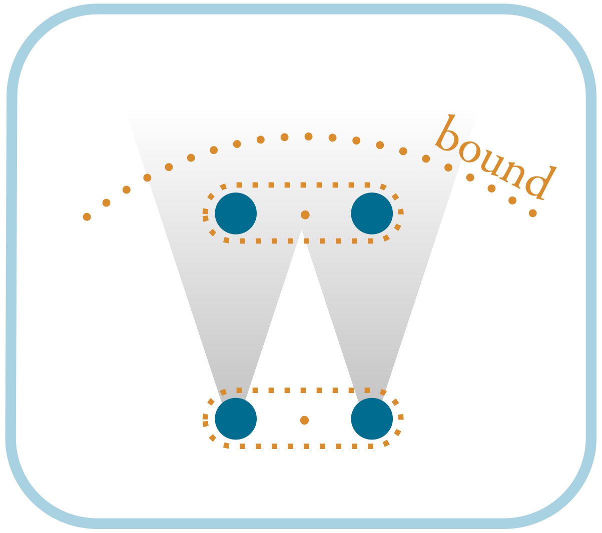
2 MIN(**Atomic**) is finite and computable

3

$$\pi_1 \cdot \pi_2 = \pi_3 \qquad \pi'_1 \leq \pi_1 \quad \pi'_2 \leq \pi_2$$

⇒

$$\pi''_1 \cdot \pi''_2 \leq \pi_3 \qquad \pi'_1 \leq \pi''_1 \text{ (bounded)} \quad \pi'_2 \leq \pi''_2 \text{ (bounded)}$$



1+2 =

```
R0 = MIN( Atomic ) ;
while (Ri ≠ Ri+1) :
    Ri+1 = MIN(Ri ∪ ↑Ri · ↑Ri)

return( Ri );
```

computes MIN(Der)

1 By property before: **Der** = ↑**Der**

$\dots\dots\blacktriangleright \{ \pi \mid \pi' \leq \pi , \pi' \in \mathbf{Der} \}$

2 MIN(**Atomic**) is finite and computable

3

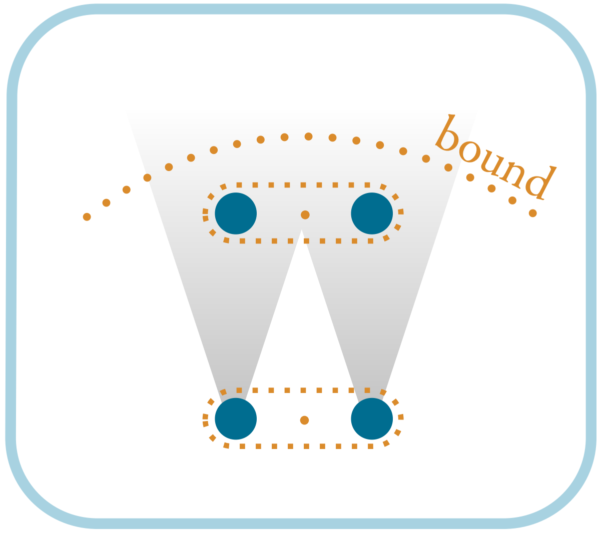
$$\pi_1 \cdot \pi_2 = \pi_3$$

$\pi'_1 \leq \pi_1$, $\pi'_2 \leq \pi_2$

⇒

$$\pi''_1 \cdot \pi''_2 \leq \pi_3$$

$\pi'_1 \leq \pi''_1$ (bounded), $\pi'_2 \leq \pi''_2$ (bounded)



1+2+3=

R_0 = MIN(**Atomic**) ;
while ($R_i \neq R_{i+1}$) :
    R_{i+1} = MIN($R_i \cup \uparrow R_i \cdot \uparrow R_i$)
        = MIN($R_i \cup \uparrow_{bounded} R_i \cdot \uparrow_{bounded} R_i$);
return(R_i);

computes MIN(Der)

Complexity? In principle, **non-primitive recursive.**

Complexity? In principle, **non-primitive recursive.**

Observation: MIN(↓abs(w,i,$|w|$)) determines whether $w \models \phi$

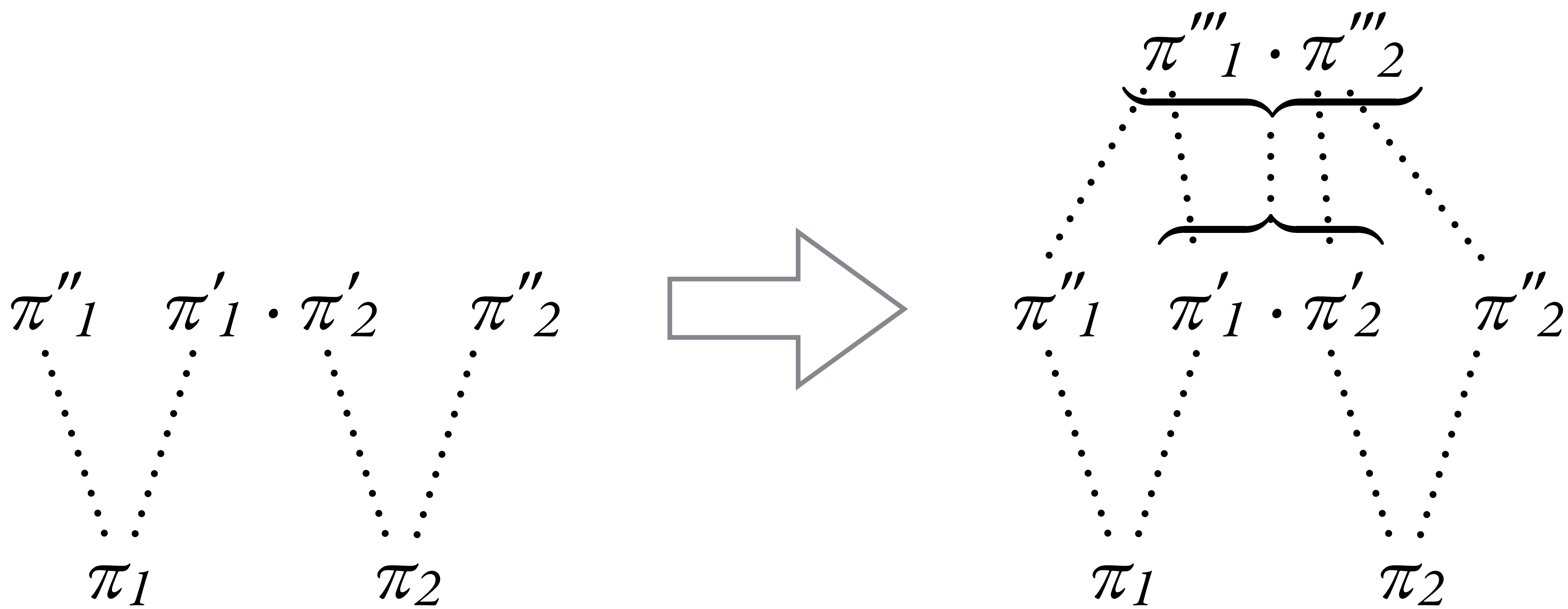
⇒ no need to compute MIN(**Der**), it suffices to compute MIN(↓**Der**).

Complexity? In principle, **non-primitive recursive.**

Observation: MIN(↓abs(w,i,|w|)) determines whether $w \models \phi$

⇒ no need to compute MIN(**Der**), it suffices to compute MIN(↓**Der**).

4

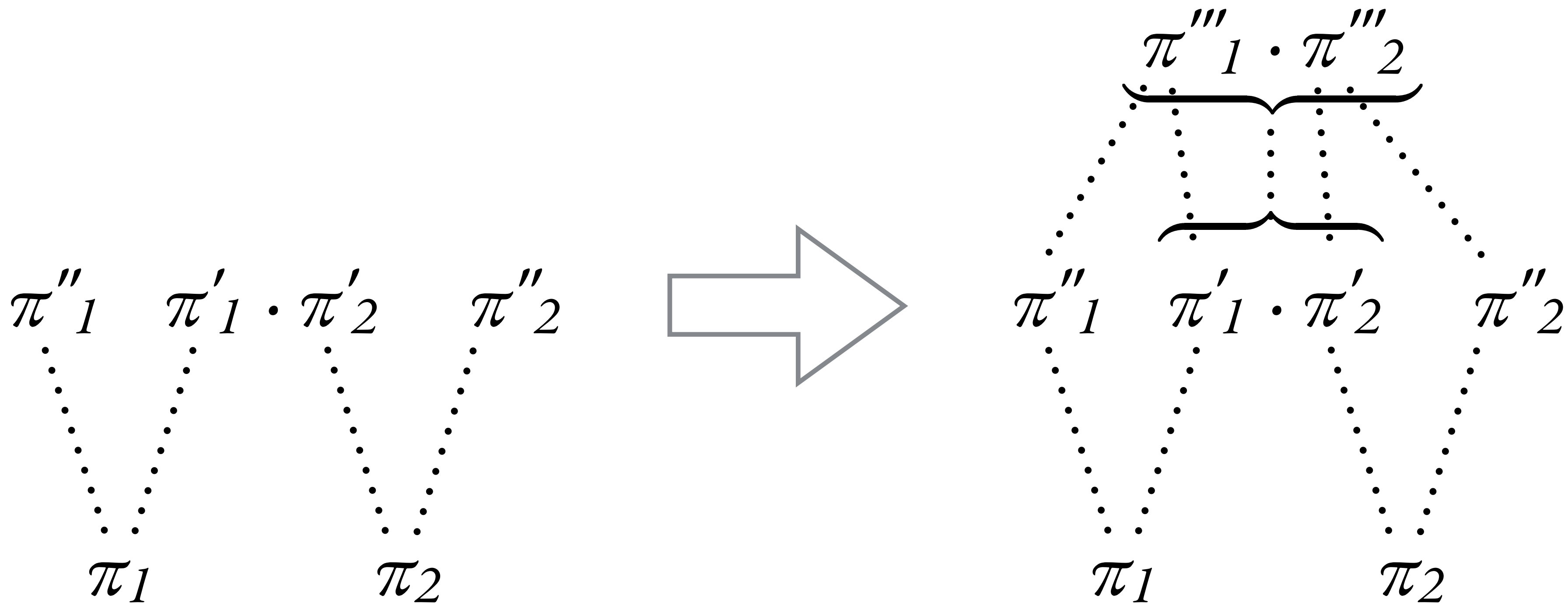


Complexity? In principle, **non-primitive recursive.**

Observation: MIN(↓abs(w,i,|w|)) determines whether $w \models \phi$

⇒ no need to compute MIN(**Der**), it suffices to compute MIN(↓**Der**).

4



1+2+3+4 =

R_0 = MIN(**Atomic**) ;
while ($R_i \neq R_{i+1}$) :
  R_{i+1} = MIN↓($R_i \cup \uparrow_{\text{bounded}} R_i \cdot \uparrow_{\text{bounded}} R_i$);
return(R_i);

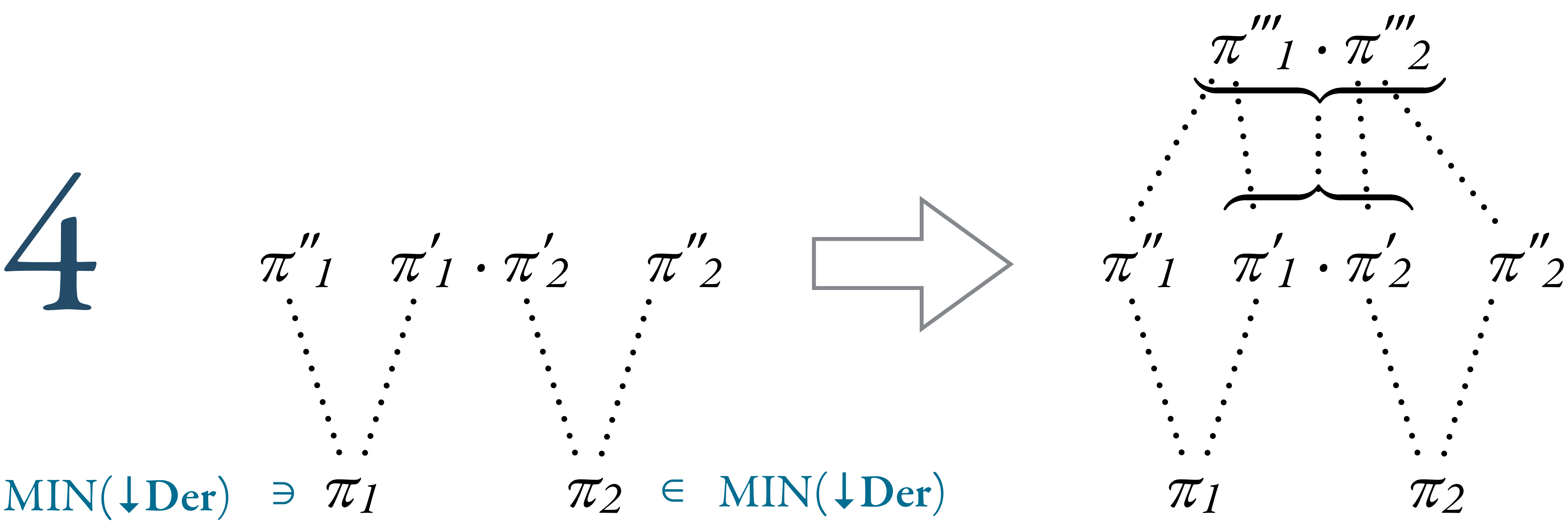
computes MIN(↓Der)

Complexity? In principle, **non-primitive recursive.**

Observation: MIN(↓abs(w,i,|w|)) determines whether $w \models \phi$

⇒ no need to compute MIN(**Der**), it suffices to compute MIN(↓**Der**).

4



1+2+3+4 =

R_0 = MIN(**Atomic**) ;
while ($R_i \neq R_{i+1}$) :
    R_{i+1} = MIN↓($R_i \cup \uparrow_{bounded} R_i \cdot \uparrow_{bounded} R_i$);
return(R_i);

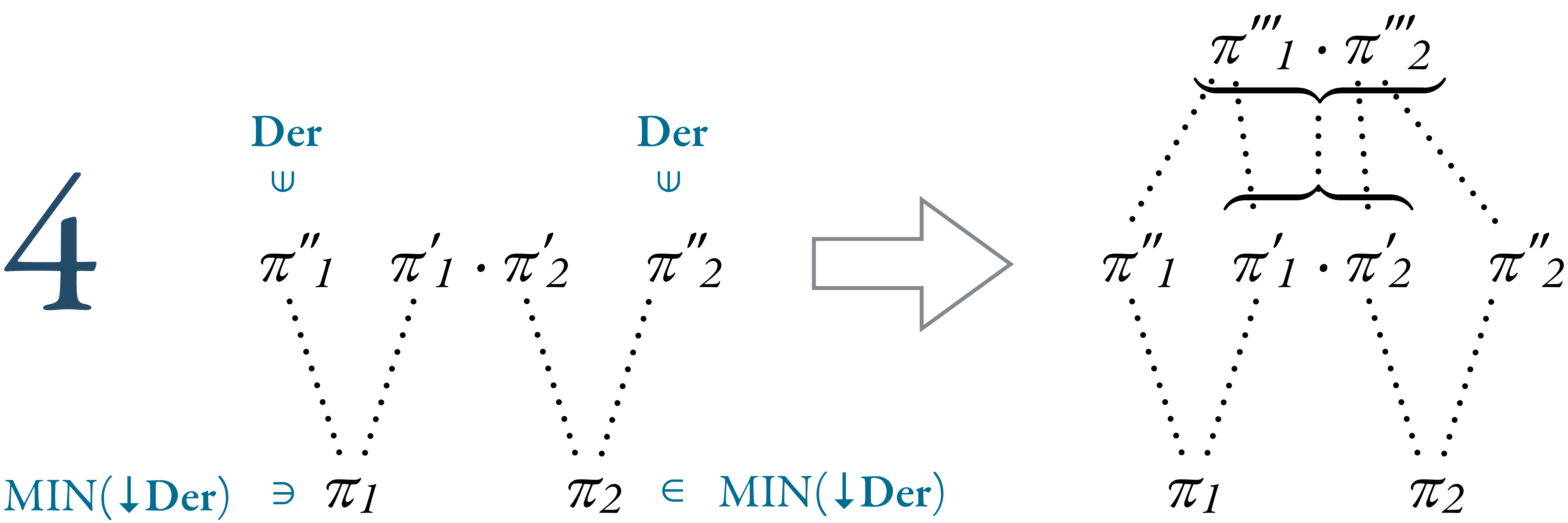
computes MIN(↓Der)

Complexity? In principle, **non-primitive recursive.**

Observation: $\mathrm{MIN}(\downarrow \mathrm{abs}(w, i, |w|))$ determines whether $w \models \phi$

$\Rightarrow$ no need to compute MIN(**Der**), it suffices to compute MIN(↓**Der**).

4



1+2+3+4 =

$R_0 = \mathrm{MIN}($ **Atomic** $)$;
while $(R_i \neq R_{i+1})$:
    $R_{i+1} = \mathrm{MIN}{\downarrow}(R_i \cup \uparrow_{\mathrm{bounded}} R_i \cdot \uparrow_{\mathrm{bounded}} R_i)$;
return(R_i);

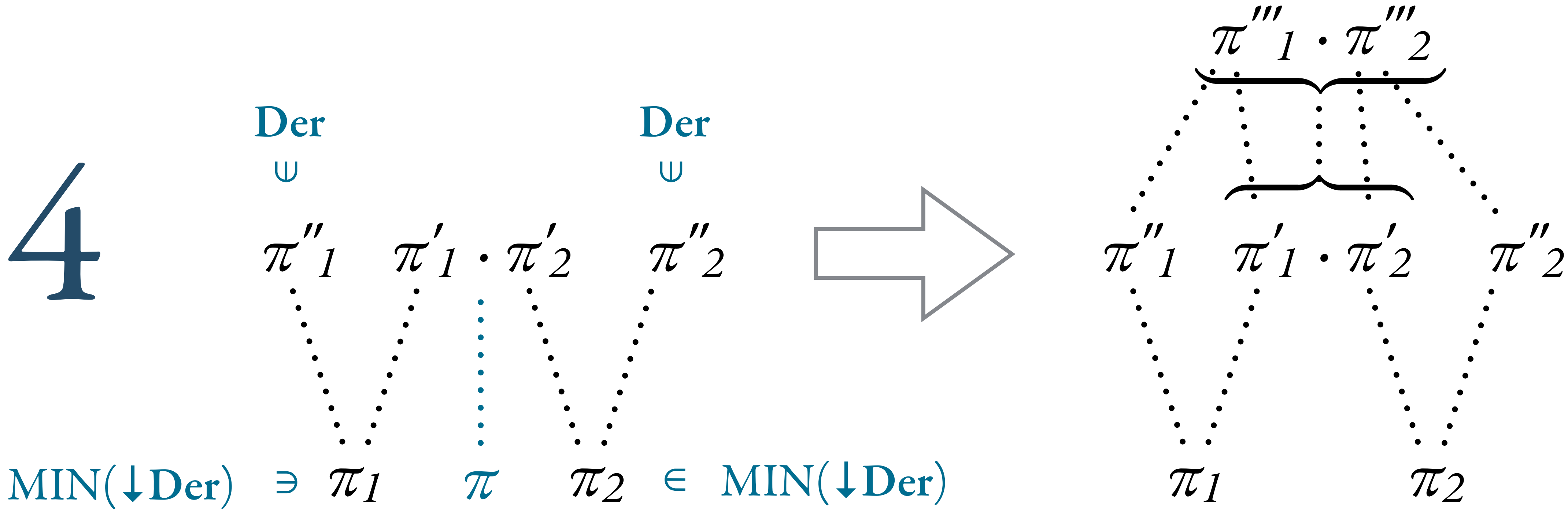
computes MIN(↓Der)

Complexity? In principle, **non-primitive recursive.**

Observation: MIN(↓abs(w,i,|w|)) determines whether $w \models \phi$

⇒ no need to compute MIN(**Der**), it suffices to compute MIN(↓**Der**).

4



1+2+3+4 =

R_0 = MIN(**Atomic**) ;
while ($R_i \neq R_{i+1}$) :
    R_{i+1} = MIN↓($R_i \cup \uparrow_{bounded} R_i \cdot \uparrow_{bounded} R_i$);
return(R_i);

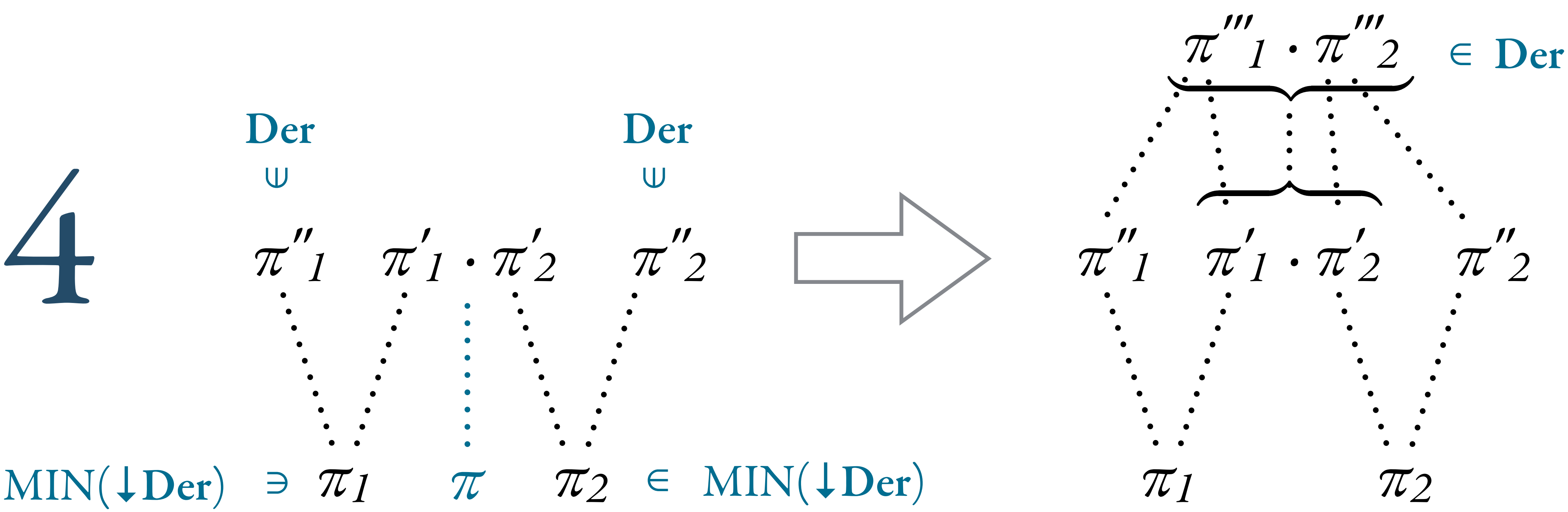
computes MIN(↓Der)

Complexity? In principle, **non-primitive recursive.**

Observation: $\mathrm{MIN}(\downarrow \mathrm{abs}(w,i,|w|))$ determines whether $w \models \phi$

$\Rightarrow$ no need to compute MIN(**Der**), it suffices to compute MIN(↓**Der**).

4



1+2+3+4 =

```
R0 = MIN( Atomic ) ;
while (Ri ≠ Ri+1) :
    Ri+1 = MIN↓(Ri ∪ ↑boundedRi · ↑boundedRi);
return( Ri );
```

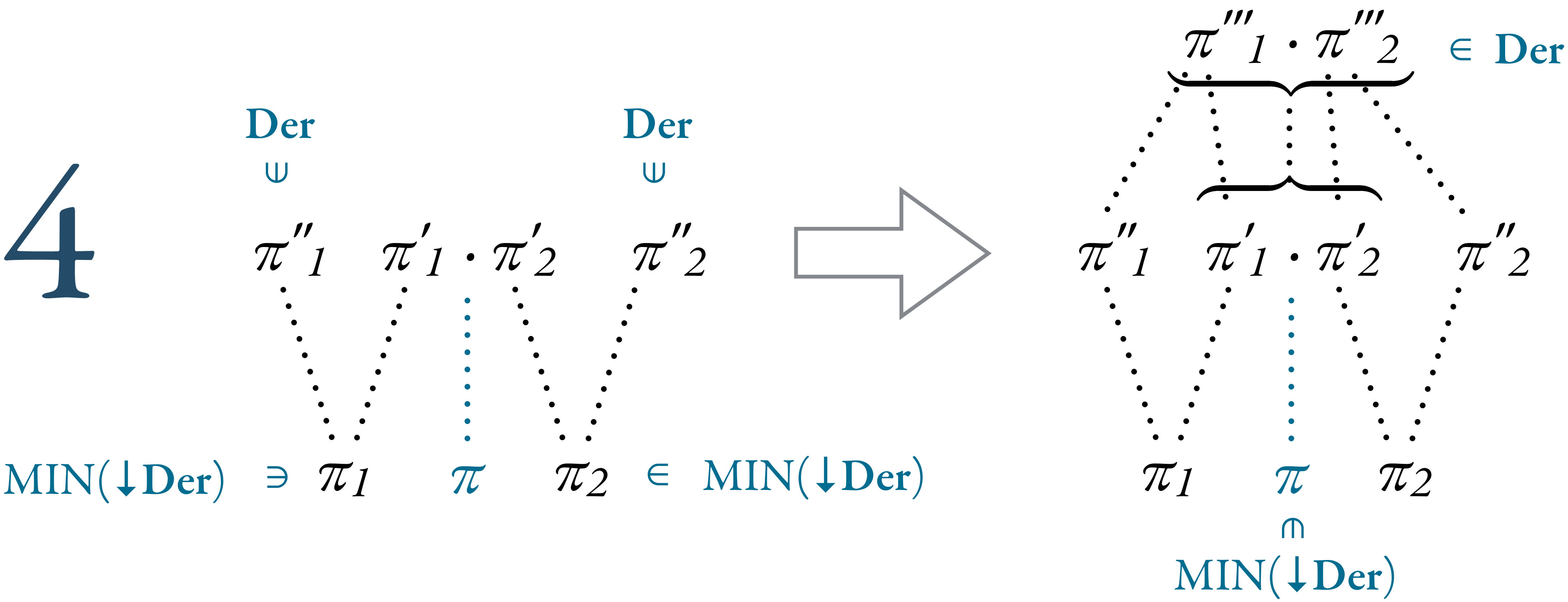
computes MIN(↓Der)

Complexity? In principle, **non-primitive recursive.**

Observation: $\mathrm{MIN}(\downarrow \mathrm{abs}(w, i, |w|))$ determines whether $w \vDash \phi$

$\Rightarrow$ no need to compute MIN(**Der**), it suffices to compute MIN(↓**Der**).

4



1+2+3+4 =

R_0 = MIN(**Atomic**) ;
while ($R_i \neq R_{i+1}$) :
  R_{i+1} = MIN↓($R_i \cup \uparrow_{\mathrm{bounded}} R_i \cdot \uparrow_{\mathrm{bounded}} R_i$);
return(R_i);

computes MIN(↓Der)

Complexity?

2^{ϕ} many MIN(**Profiles**) $\Rightarrow$ 2ExpSpace procedure

Complexity?

2^{ϕ} many MIN(**Profiles**) $\Rightarrow$ 2ExpSpace procedure

Caveat

ϕ

there is only one dv under a **c**

for every **a**, there is a **b** accessible via a **c** with the same dv

there is a position labeled **c**

w

a	b	b	a	b	c	a	b	c	c	a	a	b	c	b
1	2	4	4	3	1	5	1	1	1	4	4	5	1	4

$\models \phi$

Complexity?

2^{ϕ} many MIN(**Profiles**) $\Rightarrow$ 2ExpSpace procedure

Caveat

ϕ

there is only one dv under a **c**

for every **a**, there is a **b** accessible via a **c** with the same dv

there is a position labeled **c**

w

a	b	b	a	b	c	a	b	c	c	c	c	a	a	b	c	b	
1	2	4	4	3	1	5	1	1	9	1	9	4	4	5	1	4	$\not\models \phi$

Complexity?

2^{ϕ} many MIN(**Profiles**) $\Rightarrow$ 2ExpSpace procedure

Caveat

ϕ

there is only one dv under a **c**
for every **a**, there is a **b** accessible via a **c** with the same dv
there is a position labeled **c**

w

a	b	b	a	b	c	a	b	[c]	c	c	c	a	a	b	c	b
1	2	4	4	3	1	5	1	[1]	9	1	9	4	4	5	1	4

$\not\models \phi$

Atomic$_{\phi}$

Complexity?

2^{ϕ} many MIN(**Profiles**) $\Rightarrow$ 2ExpSpace procedure

Caveat

ϕ

there is only one dv under a **c**
for every **a**, there is a **b** accessible via a **c** with the same dv
there is a position labeled **c**

w

a	b	b	a	b	c	a	b	[c]	c	c	c	a	a	b	c	b	$\not\models \phi$
1	2	4	4	3	1	5	1	[1]	9	1	9	4	4	5	1	4	

$\not\pitchfork$

Atomic$_{\phi}$

But: There are only polynomially many 'conflicting' data values.
We can treat them as 'constants'.

From words to trees

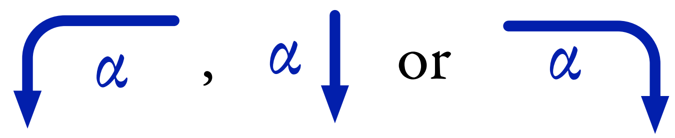
Satisfiability for XPath($^*\!\!\leftarrow$, $\downarrow_*$,$\rightarrow^*$) is decidable in 2ExpSpace.♦

♦ [F., 2013]

From words to trees

Satisfiability for XPath($^*\!\leftarrow$, $\downarrow_*$, $\rightarrow^*$) is decidable in 2ExpSpace.♦

Composed paths



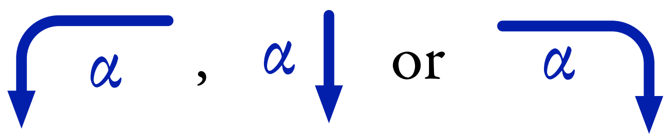
eg: $\longrightarrow^*[a]\longrightarrow^*[b]\downarrow_*[c]$

♦ [F., 2013]

From words to trees

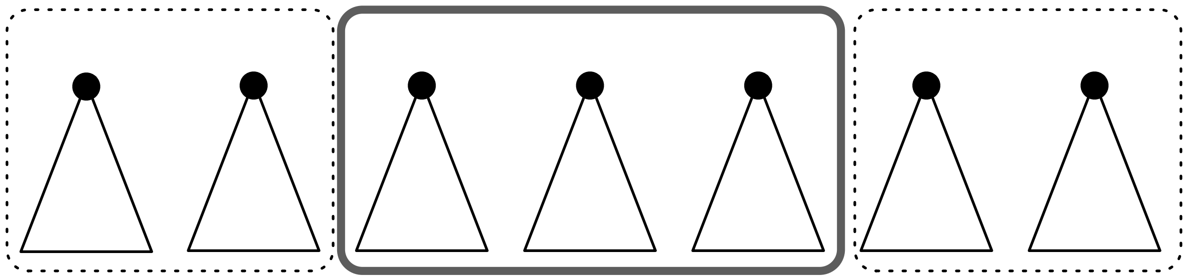
Satisfiability for XPath($^*\!\!\longleftarrow$, $\downarrow_*$, $\longrightarrow^*$) is decidable in 2ExpSpace.♦

Composed paths



eg: $\longrightarrow^*$[a]$\longrightarrow^*$[b]$\downarrow_*$[c]

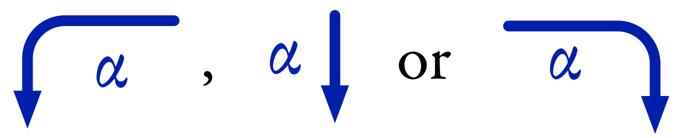
Forest profiles



♦ [F., 2013]

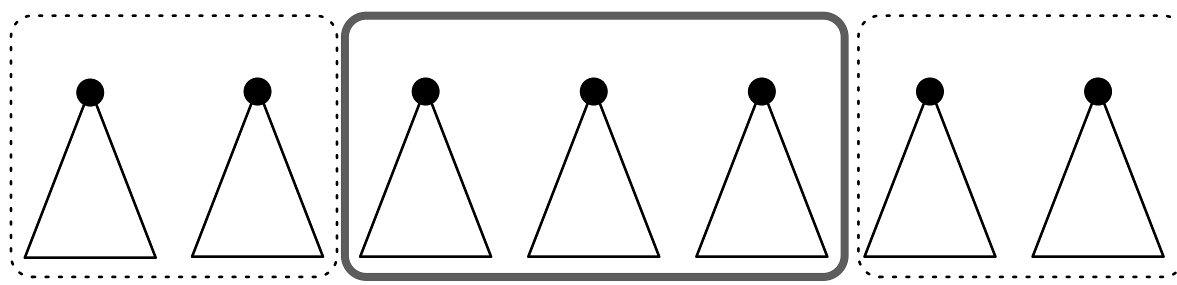
From words to trees

Satisfiability for XPath($^*\!\!\leftarrow$, $\downarrow_*$, $\longrightarrow^*$) is decidable in 2ExpSpace.♦

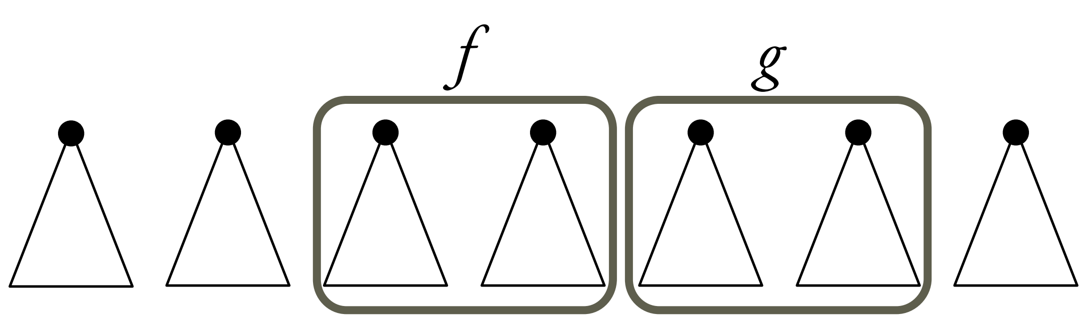
Composed paths 

eg: $\longrightarrow^*[a]\longrightarrow^*[b]\downarrow_*[c]$

Forest profiles



Two-operator algebra

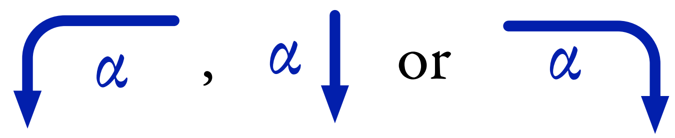


♦ [F., 2013]

From words to trees

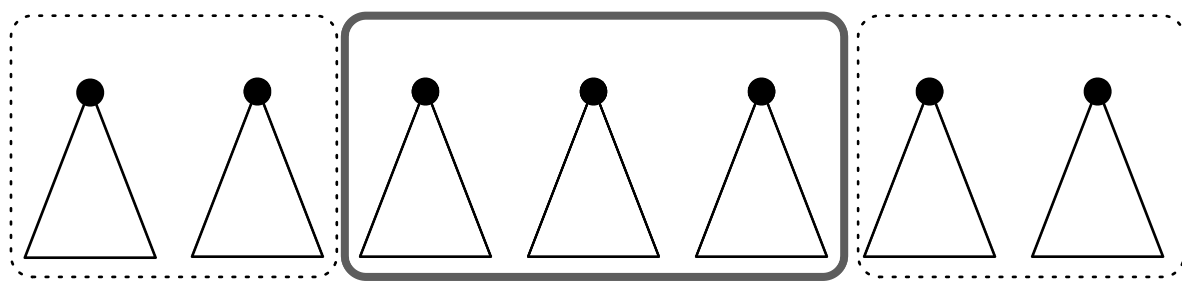
Satisfiability for XPath($^*\!\longleftarrow$, $\downarrow_*$, $\longrightarrow^*$) is decidable in 2ExpSpace.♦

Composed paths

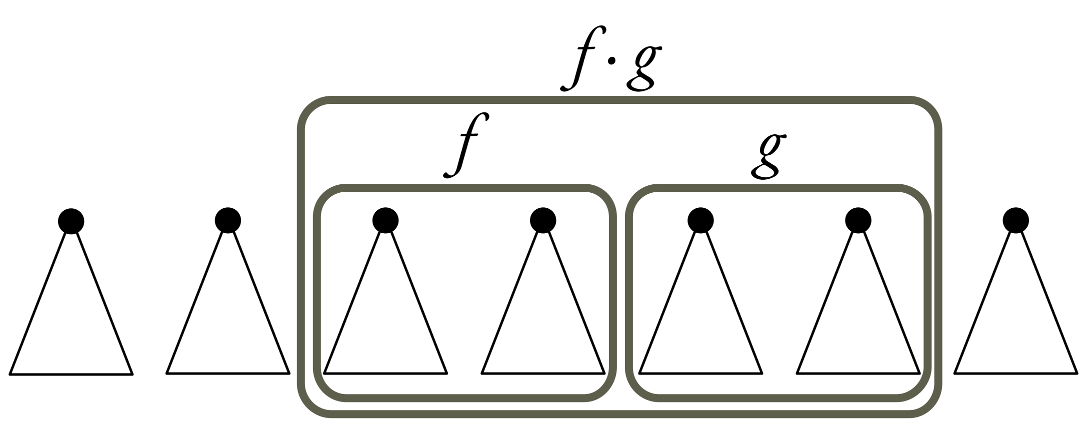


eg: $\longrightarrow^*[a]\longrightarrow^*[b]\downarrow_*[c]$

Forest profiles



Two-operator algebra

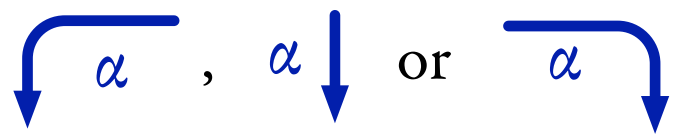


♦ [F., 2013]

From words to trees

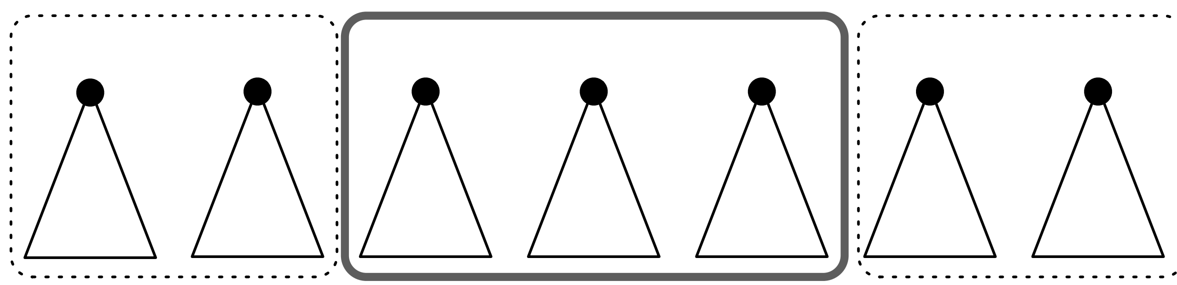
Satisfiability for XPath($^*\!\longleftarrow, \downarrow_*, \longrightarrow^*$) is decidable in 2ExpSpace.♦

Composed paths

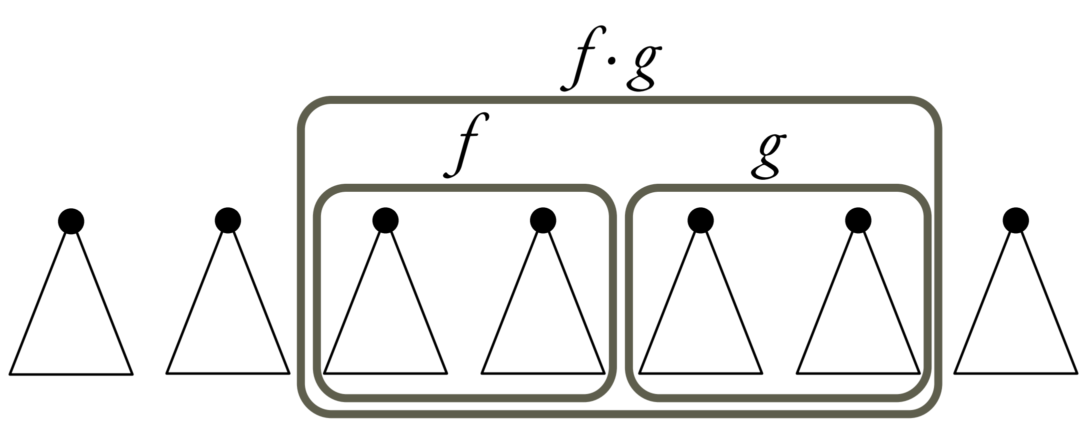


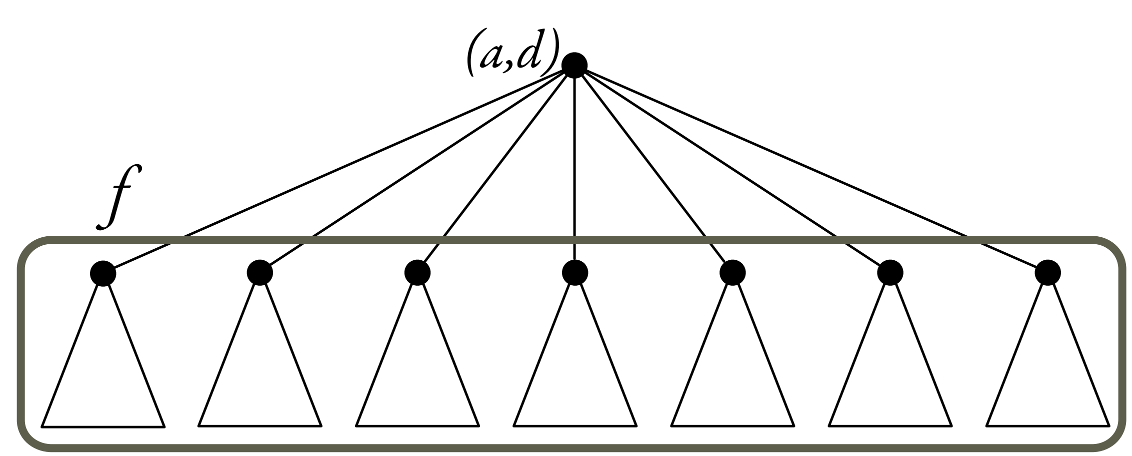
eg: $\longrightarrow^*[a]\longrightarrow^*[b]\downarrow_*[c]$

Forest profiles



Two-operator algebra



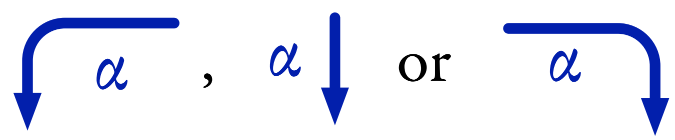


♦ [F., 2013]

From words to trees

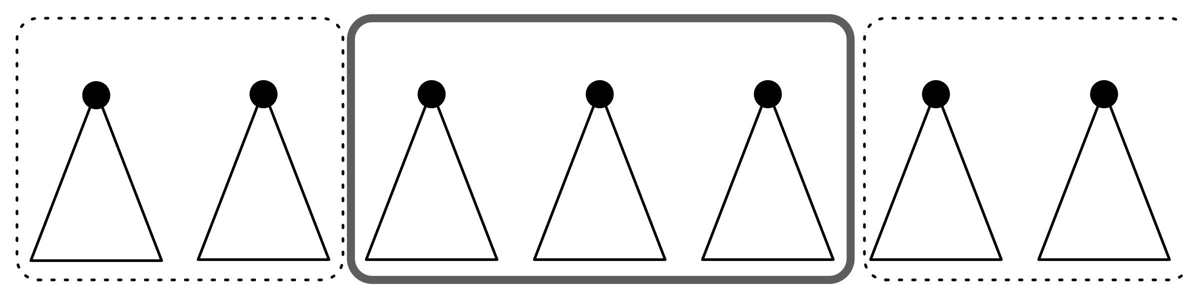
Satisfiability for XPath($^*\!\!\leftarrow, \downarrow_*, \rightarrow^*$) is decidable in 2ExpSpace.♦

Composed paths

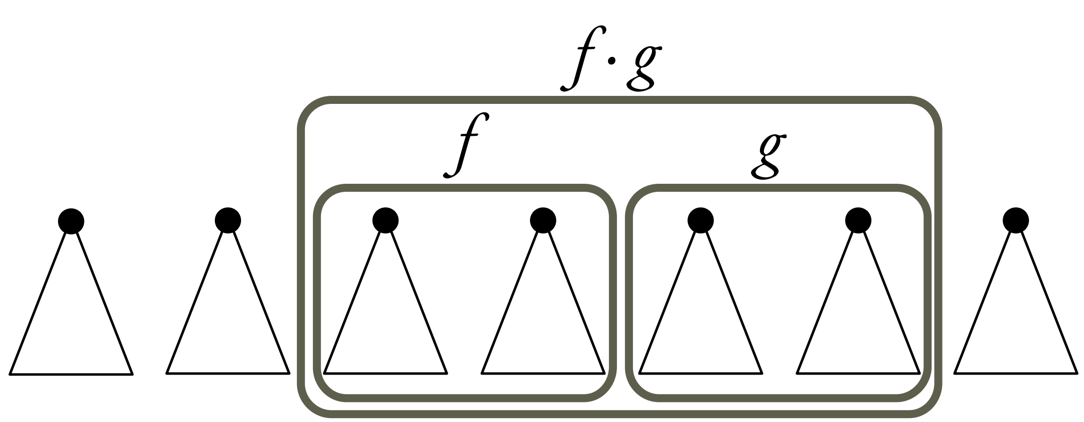


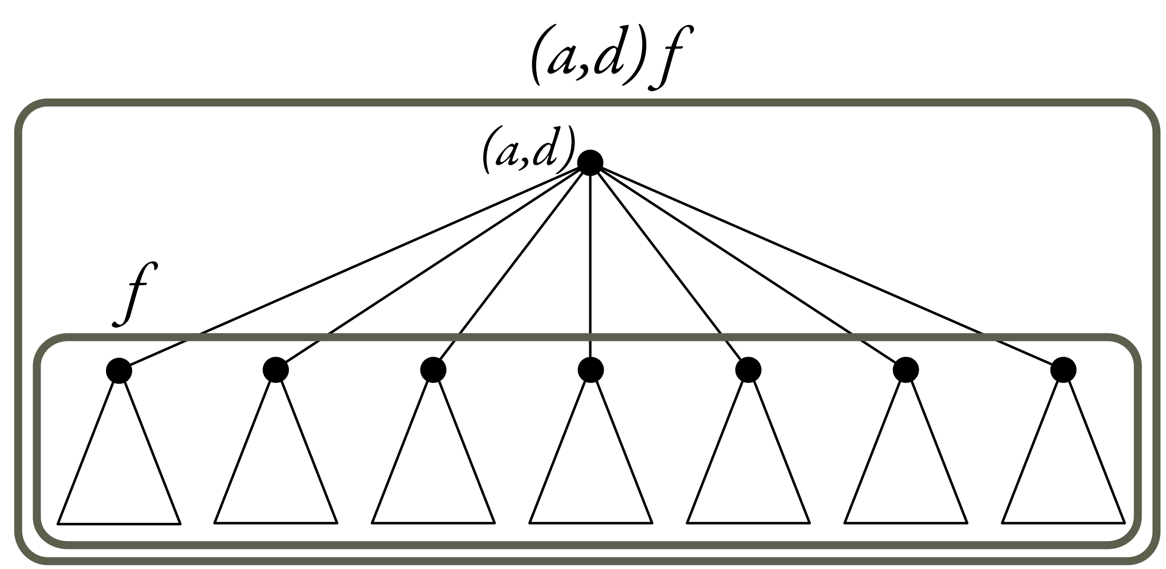
eg: $\longrightarrow^*[a]\longrightarrow^*[b]\downarrow_*[c]$

Forest profiles



Two-operator algebra





♦ [F., 2013]

Final remarks

Changing $\longrightarrow^+$ by $\longrightarrow^+$: undecidable

Adding $\uparrow^*$: non-PR

Adding domain-dependant relations?

Adding $\downarrow$: still decidable?

Complexity: 3ExpSpace (2ExpSpace in normal form)

Final remarks

Changing $\longrightarrow^+$ by $\longrightarrow^+$: undecidable

Adding $\uparrow^*$: non-PR

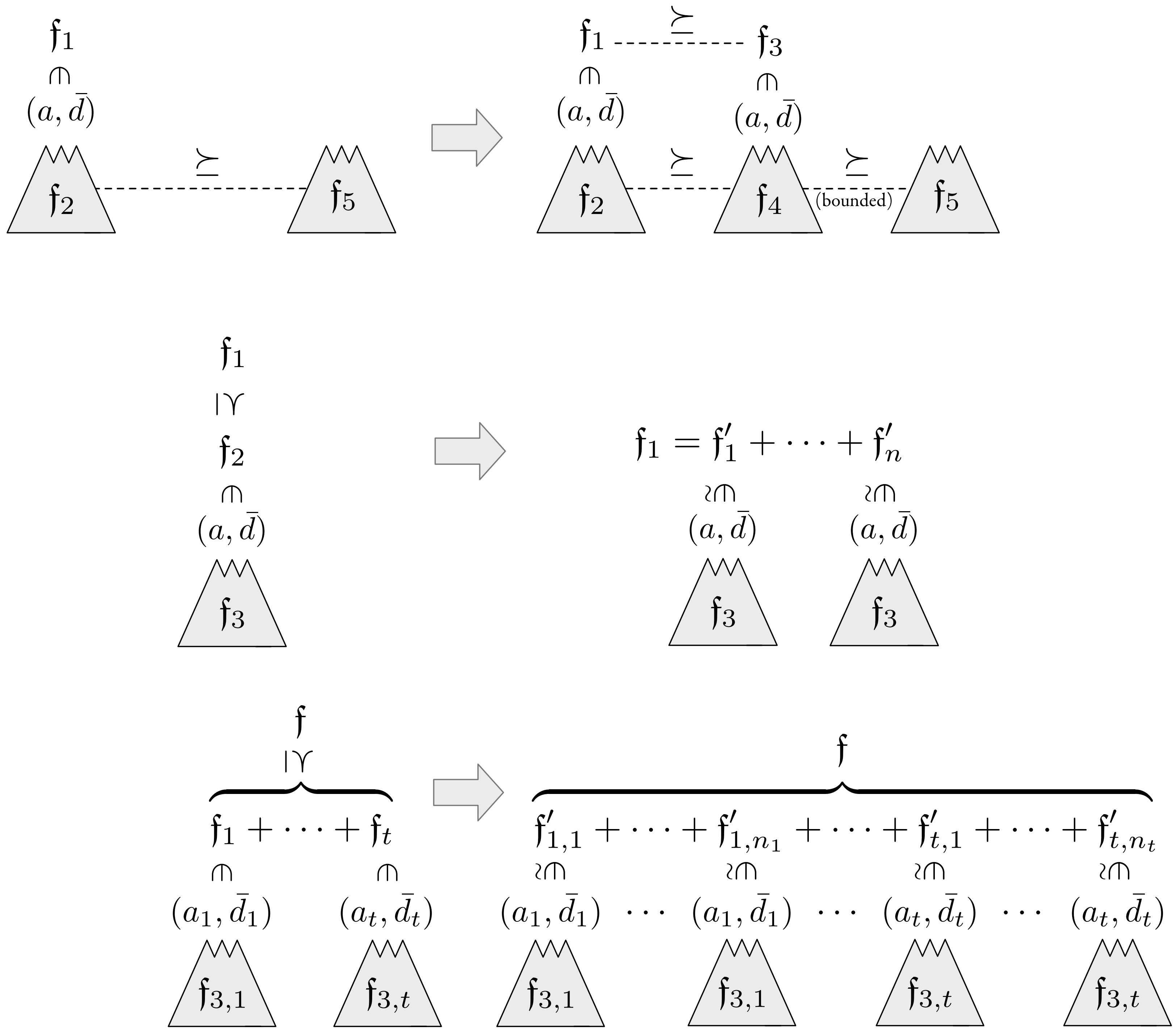
Adding domain-dependant relations?

Adding $\downarrow$: still decidable?

Complexity: 3ExpSpace (2ExpSpace in normal form)

thank you!

Etc.



$\mathfrak{f}_1$
$(a, \bar{d})$
$\mathfrak{f}_2$
$\mathfrak{f}_5$
$\mathfrak{f}_3$
$\mathfrak{f}_4$
(bounded)
$\mathfrak{f}_1 = \mathfrak{f}'_1 + \cdots + \mathfrak{f}'_n$
$\mathfrak{f}$
$\mathfrak{f}_1 + \cdots + \mathfrak{f}_t$
$(a_1, \bar{d}_1)$
$(a_t, \bar{d}_t)$
$\mathfrak{f}_{3,1}$
$\mathfrak{f}_{3,t}$
$\mathfrak{f}'_{1,1} + \cdots + \mathfrak{f}'_{1,n_1} + \cdots + \mathfrak{f}'_{t,1} + \cdots + \mathfrak{f}'_{t,n_t}$