

# Reasoning with reflexive-transitive path logics

Diego Figueira  
CNRS & LaBRI

# *reasoning on data words*

*data word*     $\begin{matrix} a & b & a & b & a & b & b & b & b \\ 2 & 4 & 7 & 6 & 9 & 3 & 9 & 7 & 6 \end{matrix} \quad \begin{matrix} a & b & b \\ 2 & 2 & 9 \end{matrix}$      $\in (\{a, b\} \times \mathbb{N})^*$

# *reasoning on data words*

*data word*     $\begin{matrix} a & b & a & b & a & b & b & b & b & a & b & b \\ 2 & 4 & 7 & 6 & 9 & 3 & 9 & 7 & 6 & 2 & 2 & 9 \end{matrix}$      $\in (\{a, b\} \times \mathbb{N})^*$

“for every  $a$  there is a  $b$  with same data value to its right”

*w*hat does it represent?

# *reasoning on data words*

<i>data word</i>	$r(A)$ 4	$r(B)$ 7	$w(A)$ 4	$w(B)$ 7	<i>commit</i> 4	$r(A)$ 7	$w(A)$ 7
------------------	-------------	-------------	-------------	-------------	--------------------	-------------	-------------

*“between  $w(A)$  and *commit* of process P there are no  $w(A)$  from any other process P”*

*execution of concurrent processes,*

*W*hat does it represent?

# *reasoning on data words*

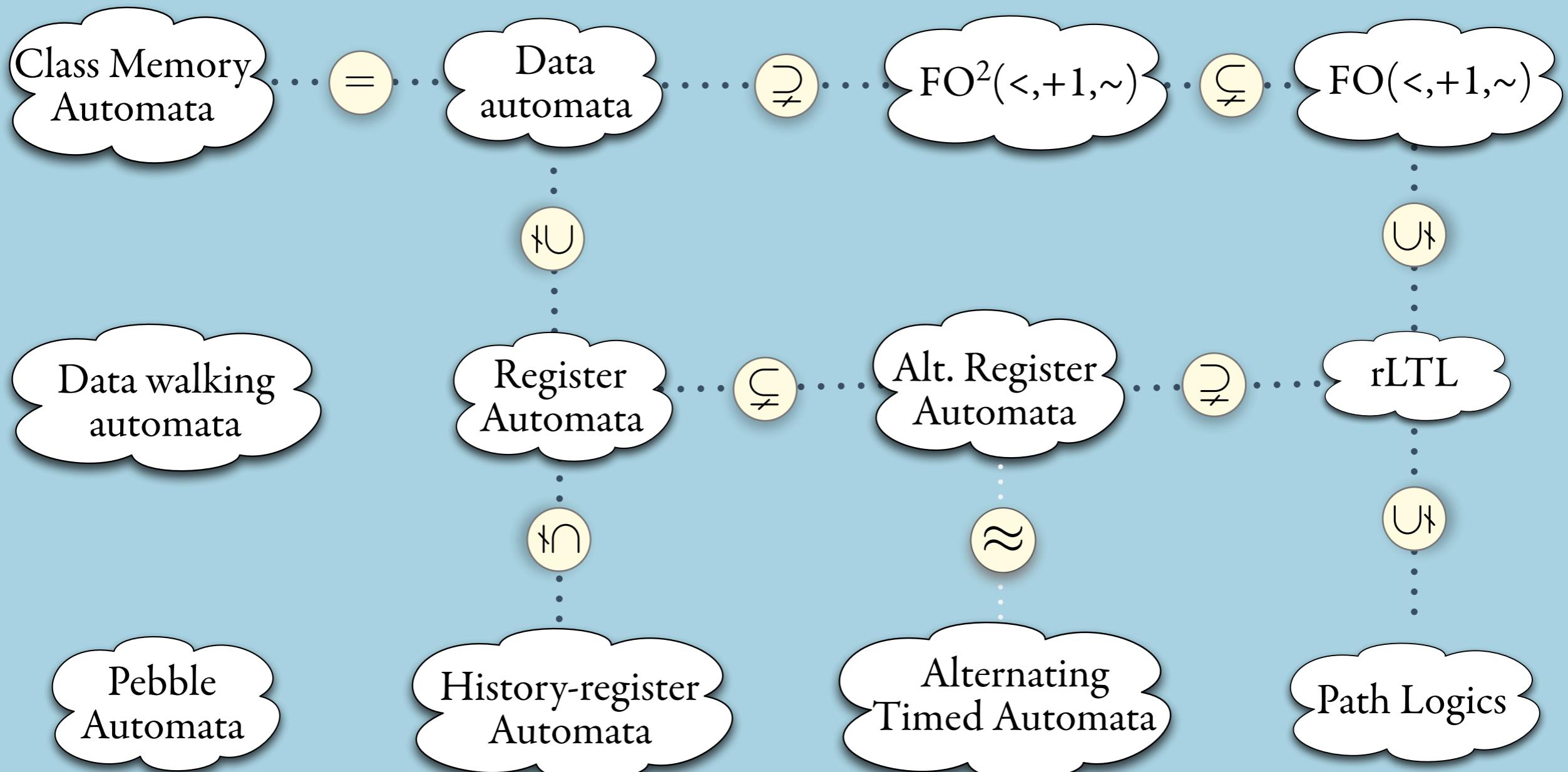
<i>data word</i>	$r(A)$ 4	$r(B)$ 7	$w(A)$ 4	$w(B)$ 7	<i>commit</i> 4	$r(A)$ 7	$w(A)$ 7
------------------	-------------	-------------	-------------	-------------	--------------------	-------------	-------------

“between  $w(A)$  and *commit* of process  $P$  there are no  $w(A)$  from any other process  $P'$ ”

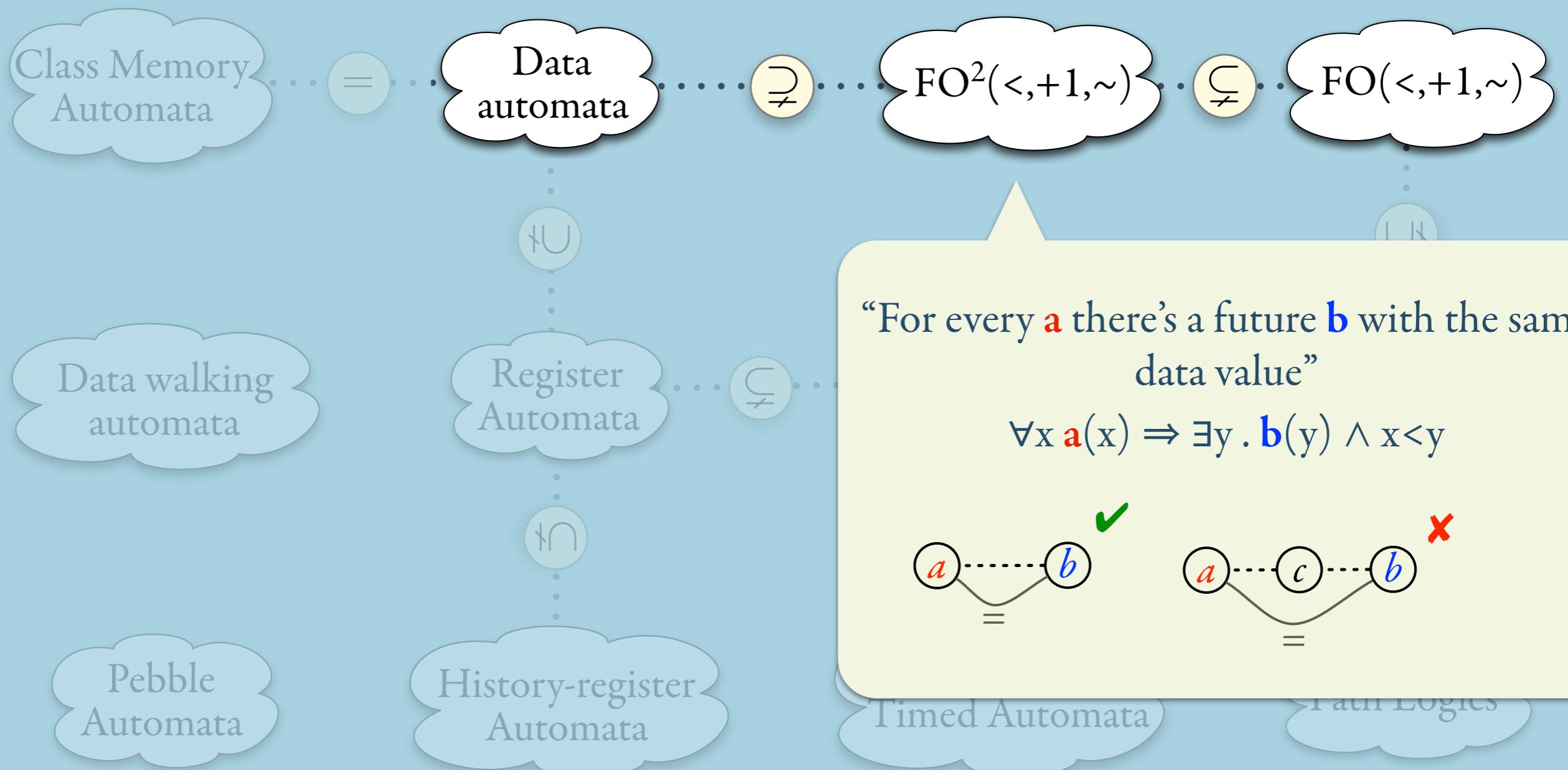
*W*hat does it represent?

execution of concurrent processes,  
usage of some unbounded resources,  
timed words,  
temporal databases,  
runs of counter automata (or inf. state aut.),  
...

# Formalisms for data words



# Formalisms for data words



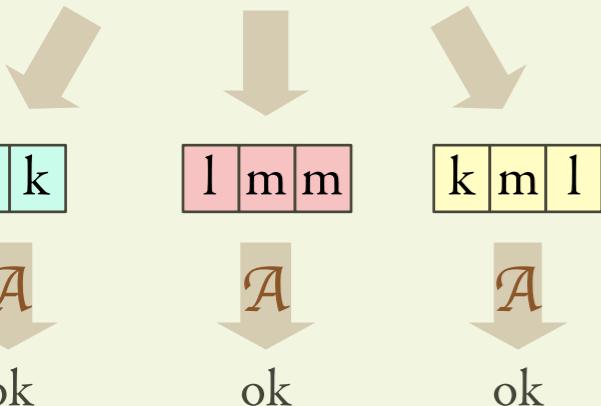
# Formalisms for data words

*Data Automata*  
 $\langle \mathcal{T}, \mathcal{A} \rangle$

a | b | a | c | b | a | b | b

$\downarrow \mathcal{T}$

k | l | k | m | l | m | k | m



Data  
automata

$\mathcal{D}$

$\mathcal{FO}^2(<, +1, \sim)$

$\mathcal{FO}(<, +1, \sim)$

Register  
Automata

$\mathcal{U}$

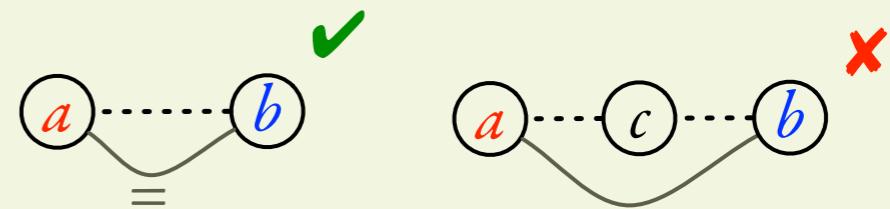
$\mathcal{C}$

$\mathcal{L}$

History-register  
Automata

“For every **a** there’s a future **b** with the same data value”

$$\forall x \, \mathbf{a}(x) \Rightarrow \exists y . \, \mathbf{b}(y) \wedge x < y$$



Timed Automata

Path Logics

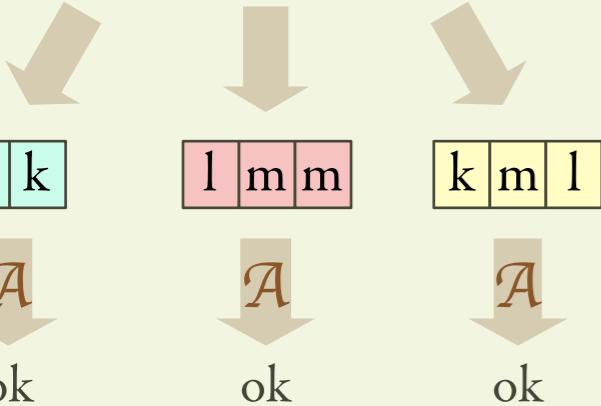
# Formalisms for data words

*Data Automata*  
 $\langle \mathcal{T}, \mathcal{A} \rangle$

a | b | a | c | b | a | b | b



k | l | k | m | l | m | k | m



Data  
automata

emptiness =  
reachability problem  
for Petri Nets.

History-register  
Automata



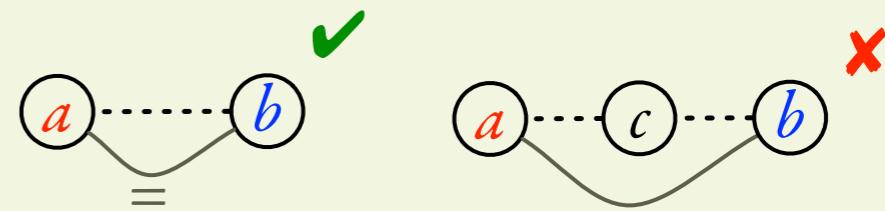
$\text{FO}^2(<, +1, \sim)$



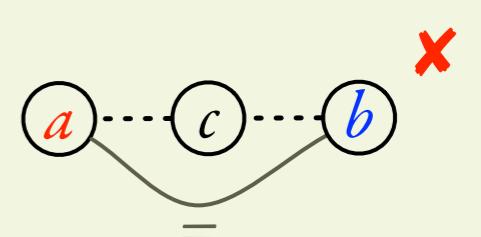
$\text{FO}(<, +1, \sim)$

“For every **a** there’s a future **b** with the same data value”

$$\forall x \, \mathbf{a}(x) \Rightarrow \exists y . \mathbf{b}(y) \wedge x < y$$

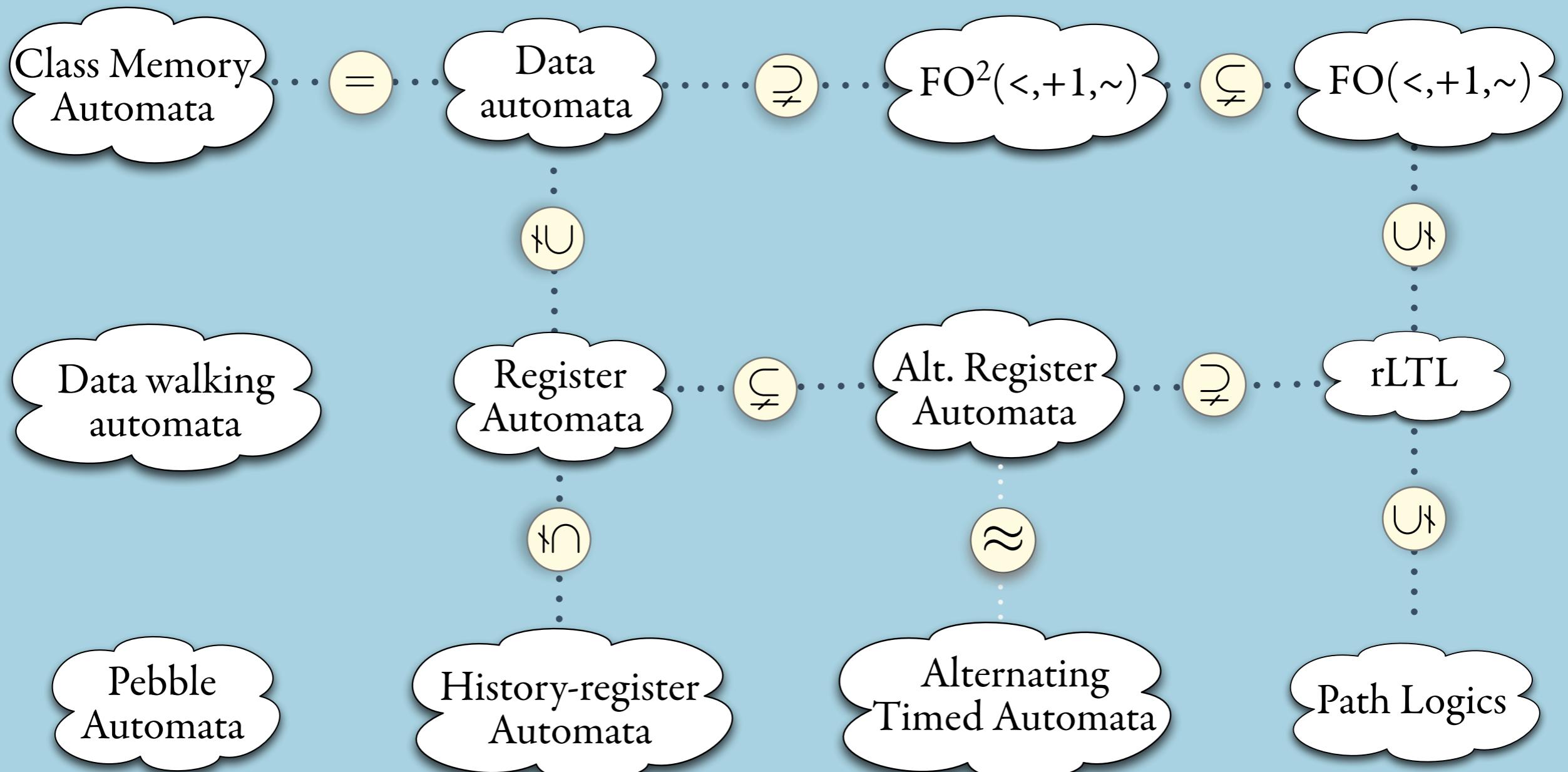


Timed Automata



Path Logics

# Formalisms for data words



# *data word*

b c a b c c a a b c b  
3 1 5 1 1 1 4 4 5 1 4     $\in (A \times D)^*$   
⋮ ⋮ ..... → infinite domain  
⋮ ..... → finite alphabet

Reasoning with logics on data words:

high complexity

or

limited expressive power

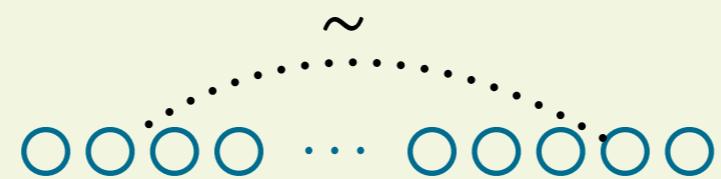
Things become ugly as soon as:

ooo ~ ooo ... ooo

# Complexity of logics for data words

A “real logic”:  $\wedge \vee \neg$

Capable to do distant tests:



# Complexity of logics for data words

## Logic

$\text{FO}^2(<, +1, \sim)$

$\text{FO}^2(<, \sim)$

$\text{LTL}^\downarrow(F, U, X)$

$\text{LTL}^\downarrow(F)$

$\text{LTL}^\downarrow(F, F^{-1})$

BasicDataLTL

LRV

LRV + P

## SAT

$\sim \text{PN-reach}$

$\text{NExpTime-c}$

Decidable, non-PR hard

Decidable, non-PR hard

Undecidable

$\sim \text{PN-reach}$

$2\text{ExpSpace-c}$

$\sim \text{PN-reach}$

[Bojańczyk & al.]

[Bojańczyk & al.]

[Demri, Lazić]

[F, Segoufin]

[F, Segoufin]

[Kara & al]

[Demri, F, Praveen]

[Demri, F, Praveen]

# Complexity of logics for data words

## Logic

$\text{FO}^2(<, +1, \sim)$

## SAT

$\sim$  PN-reach

[Bojańczyk & al.]

$\text{FO}^2(<, \sim)$

NExpTime-c

[Bojańczyk & al.]

$\text{LTL}^\downarrow(F, U, X)$

Decidable, non-PR hard

[Demri, Lazić]

$\text{LTL}^\downarrow(F)$

Decidable, non-PR hard

[F, Segoufin]

$\text{LTL}^\downarrow(F, F^{-1})$

Undecidable

[F, Segoufin]

BasicDataLTL

$\sim$  PN-reach

[Kara & al]

LRV

2ExpSpace-c

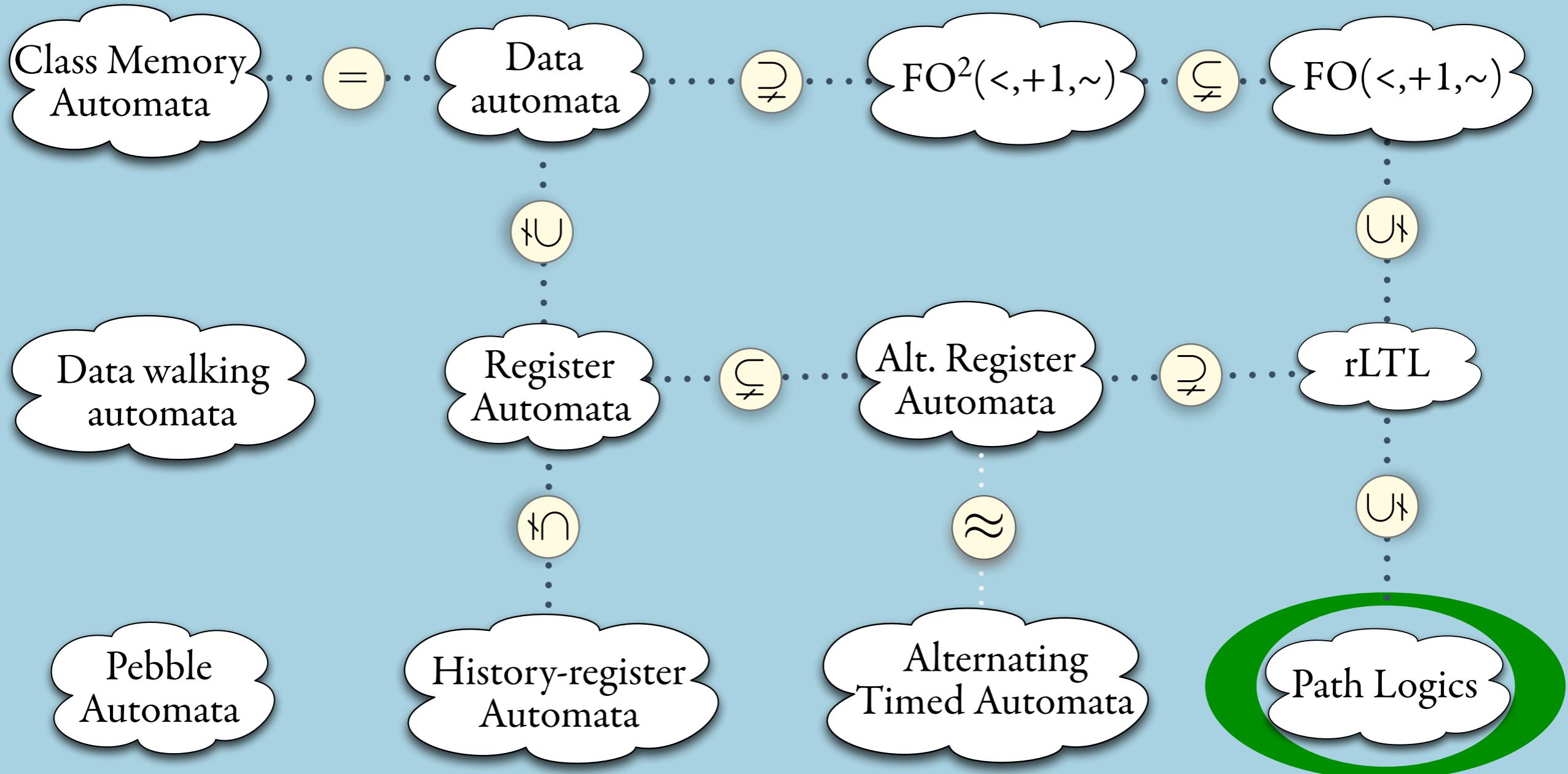
[Demri, F, Praveen]

LRV + P

$\sim$  PN-reach

[Demri, F, Praveen]

# Formalisms for data words



# Path logics

- Path = sequence of contiguous positions
- Set of properties  $\mathbf{P}$  of paths (eg. regular languages)

*“I can travel through to a position with the **same data value** through a path  $\pi$  with property  $p \in \mathbf{P}$ . ”*

Example: PDL  $\langle a^*b \rangle \phi$  (From the current point I can access a point through a **path** in  $a^*b$  where  $\phi$  holds)

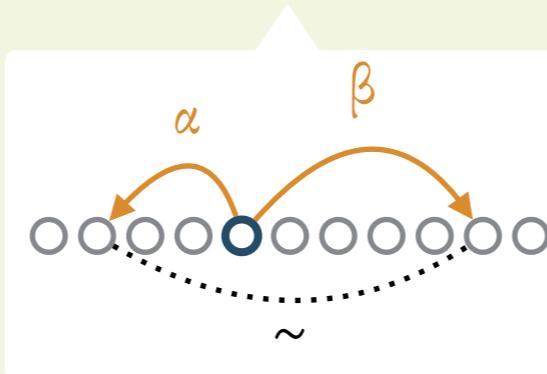
# Path logics

XPath on data words

path expressions

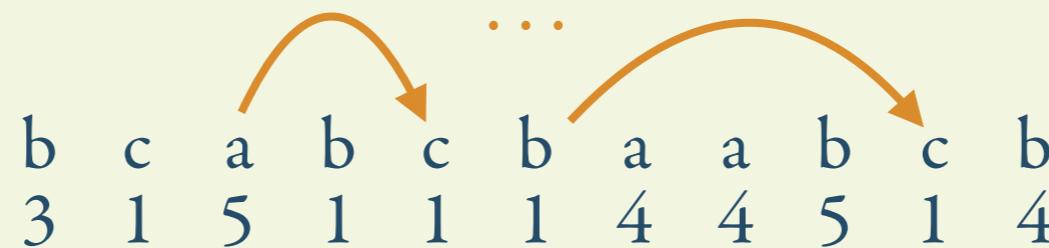

$$\alpha, \beta ::= \varepsilon \mid \alpha\beta \mid \alpha[\phi] \mid o \quad o \in \{ \rightarrow, \rightarrow^+, \rightarrow^*, \leftarrow, +\leftarrow, *+\leftarrow \}$$

node expressions


$$\phi, \psi ::= a \mid \neg\phi \mid \phi \wedge \psi \mid \langle \alpha = \beta \rangle \mid \langle \alpha \neq \beta \rangle \mid \alpha? \quad a \in A$$


# Path logics

XPath on data words

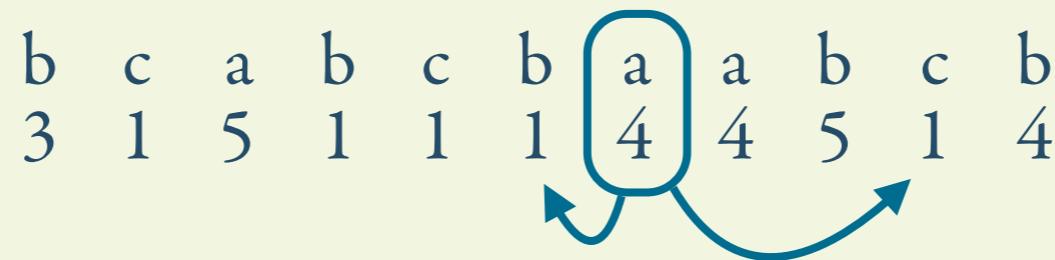


eg:

$\rightarrow^*[b] \rightarrow [c]$

# Path logics

XPath on data words

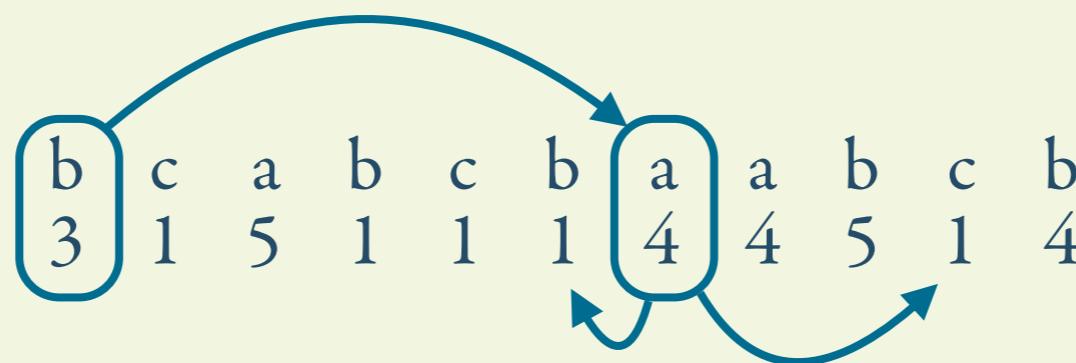


eg:

$$\langle \xleftarrow{[b]} = \xrightarrow{* [b]} \xrightarrow{[c]} \rangle$$

# Path logics

XPath on data words



eg:  $\rightarrow^*[a \wedge \leftarrow[b] = \rightarrow^*[b] \rightarrow[c]]$

# Satisfiability of XPath on data words

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

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

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

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]

# Satisfiability of XPath on data words

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

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

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

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

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



why?

♠ [Demri, Lazić,

2006]

♣ [F., Segoufin, 2009]

# Satisfiability of XPath on data words

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

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

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

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

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

*why?*

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



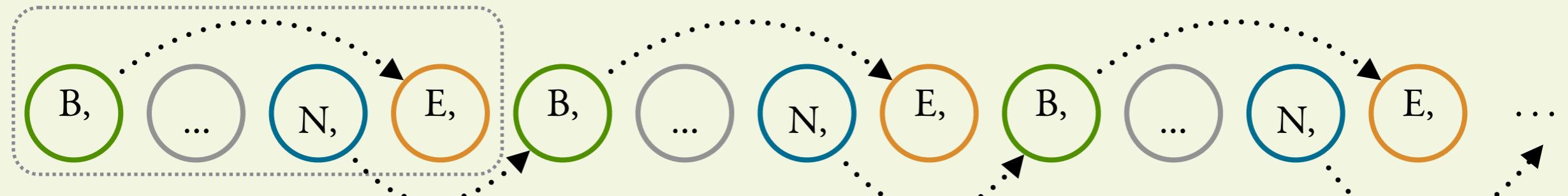
SAT-XPath( $\rightarrow^+$ )

♠ [Demri, Lazić,

2006]

♣ [F., Segoufin, 2009]

*block*



Q

○

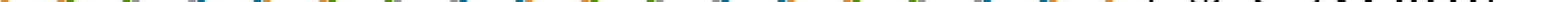
○

○

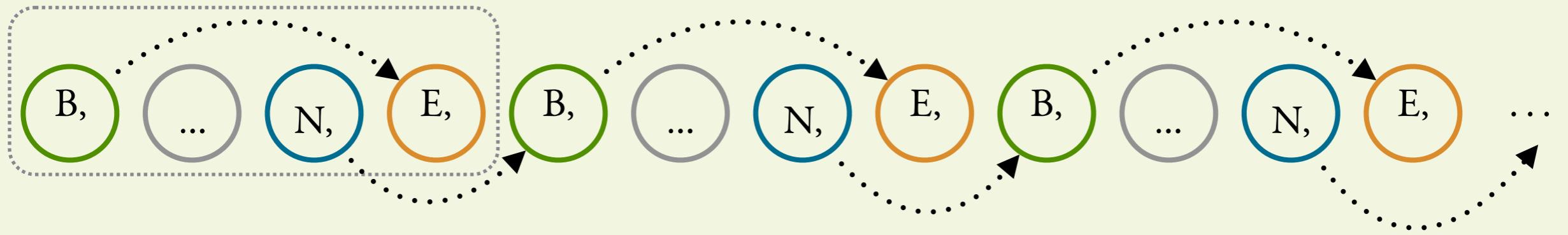
Q

$\models \phi \in \text{XPath}(\rightarrow, \rightarrow^+)$

11

  $\models \phi' \in \text{XPath}(\rightarrow +)$

*block*



No more than one  $B_x$ ,  $N_y$ ,  $E_x$  per block

All  $B_x$  have different data (resp  $N_y$ ,  $E_x$ )

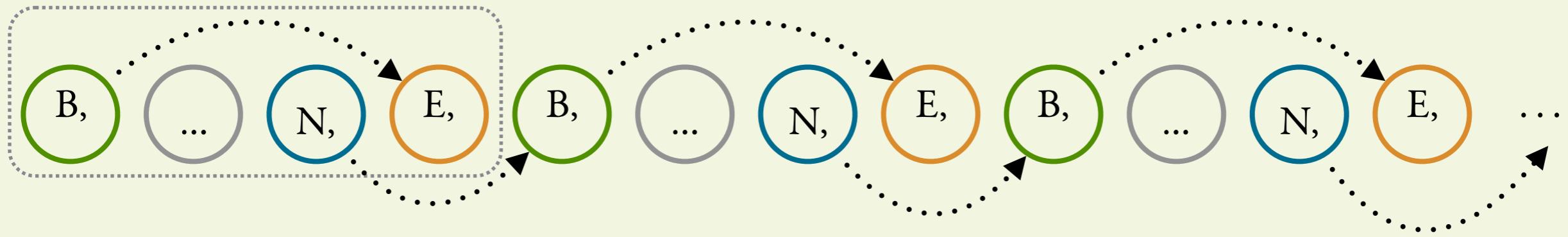
$N_y$  points to next block

For every  $B_x$  there is an  $E_x$  with same datum

Order in the block:  $B_x$   $\dots$   $N_y$   $E_x$

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

*block*



$\longrightarrow^+ \longrightarrow^*$

✓ ✗ 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:  $B_x$   $\dots$   $N_y$   $E_x$

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

# Satisfiability of XPath on data words

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

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

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

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

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

What about XPath( $\rightarrow^*$ )?    XPath( $\rightarrow^*$ ) is **decidable** in 2ExpSpace ♦

♦ [Đorđe Lazić,  
2006]

♣ [F. Segoufin, 2009]

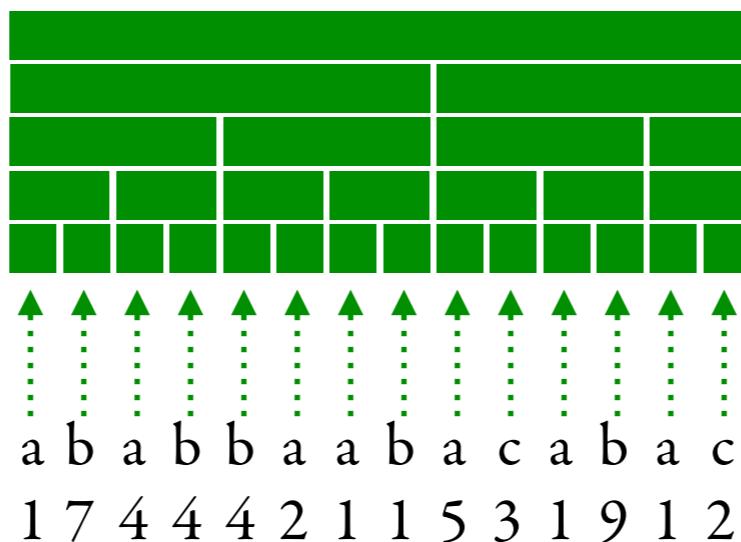
*why?*

# SAT-XPath( $\rightarrow^*, * \leftarrow$ ) decidable in 2ExpSpace

Proof idea:

We work with abstractions of sub-words of a data word satisfying  $\phi$

There must be a  
 $(b, 1)$  to the right



“Every a has a b with the same dv to the right”

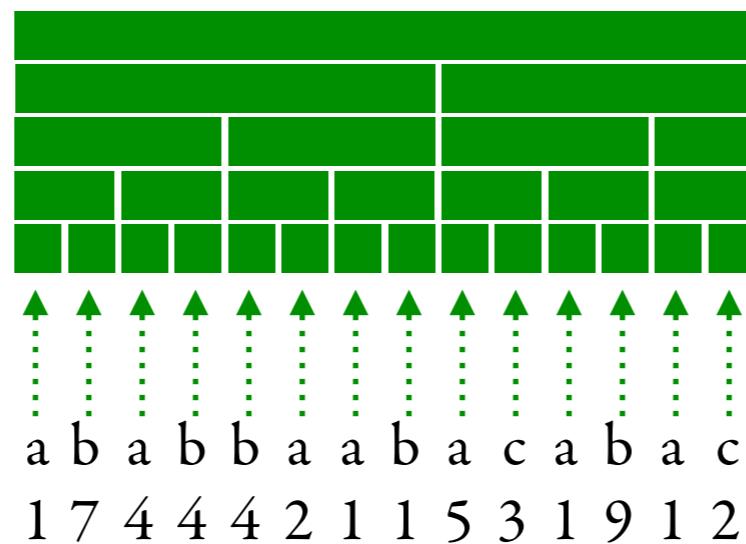
Each position has its “atomic profile”  
Concatenation between profiles

- what's inside
- demands to the left
- demands to the right

# SAT-XPath( $\rightarrow^*, * \leftarrow$ ) decidable in 2ExpSpace

Proof idea:

We work with abstractions of sub-words of a data word satisfying  $\phi$



Each position has its “atomic profile”  
Concatenation between profiles



Find a way of concatenating atomic profiles so that it has a certain property

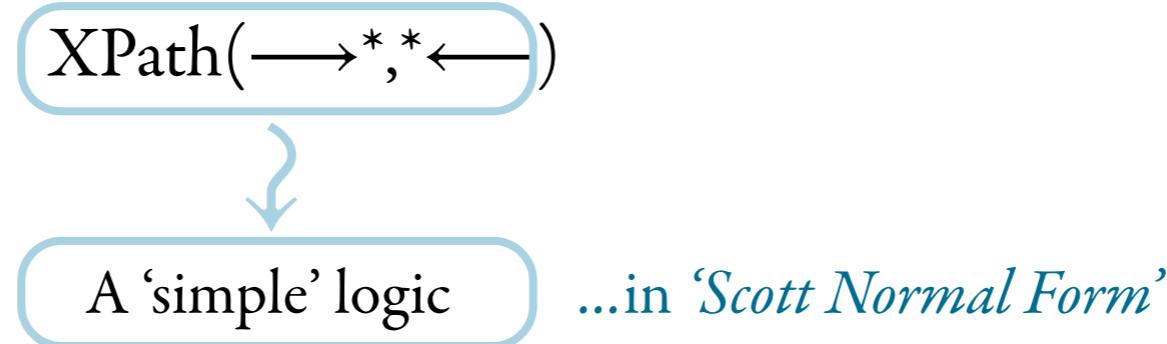
**Problem:** profiles are infinite

**Solution:** well-behaved ordering between profiles preserving sat:

if  $\pi$  abstracts a part of a model of  $\phi$  then so does  $\pi' \geq \pi$

# SAT-XPath( $\rightarrow^*, * \leftarrow$ ) decidable in 2ExpSpace

Proof idea:



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

$$\langle [\phi_n]^* \leftarrow \dots \leftarrow [\phi_1]^* \leftarrow [\phi_0] = \rightarrow^* [\psi_0] \rightarrow^* [\psi_1] \rightarrow^* \dots \rightarrow^* [\psi_m] \rangle$$

$$\langle [\phi_n] \rightarrow^* \dots \rightarrow^* [\phi_1] \rightarrow^* [\phi_0] = \rightarrow^* [\psi_0] \rightarrow^* [\psi_1] \rightarrow^* \dots \rightarrow^* [\psi_m] \rangle$$

$$\langle d = \rightarrow^* [\psi_0] \rightarrow^* [\psi_1] \rightarrow^* \dots \rightarrow^* [\psi_m] \rangle$$

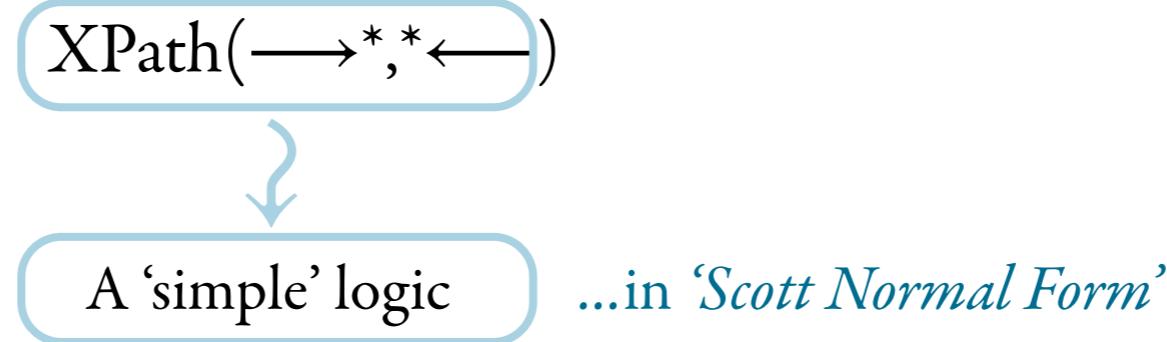
⋮

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

$d \in D$

# SAT-XPath( $\rightarrow^*, * \leftarrow$ ) decidable in 2ExpSpace

Proof idea:



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

$$\langle [\phi_n]^* \leftarrow \dots \leftarrow [\phi_1]^* \leftarrow [\phi_0]^\neq = \rightarrow^* [\psi_0] \rightarrow^* [\psi_1] \rightarrow^* \dots \rightarrow^* [\psi_m] \rangle$$

$$\langle [\phi_n] \rightarrow^* \dots \rightarrow^* [\phi_1] \rightarrow^* [\phi_0]^\neq = \rightarrow^* [\psi_0] \rightarrow^* [\psi_1] \rightarrow^* \dots \rightarrow^* [\psi_m] \rangle$$

$$\langle d = \rightarrow^\neq \rightarrow^* [\psi_0] \rightarrow^* [\psi_1] \rightarrow^* \dots \rightarrow^* [\psi_m] \rangle$$

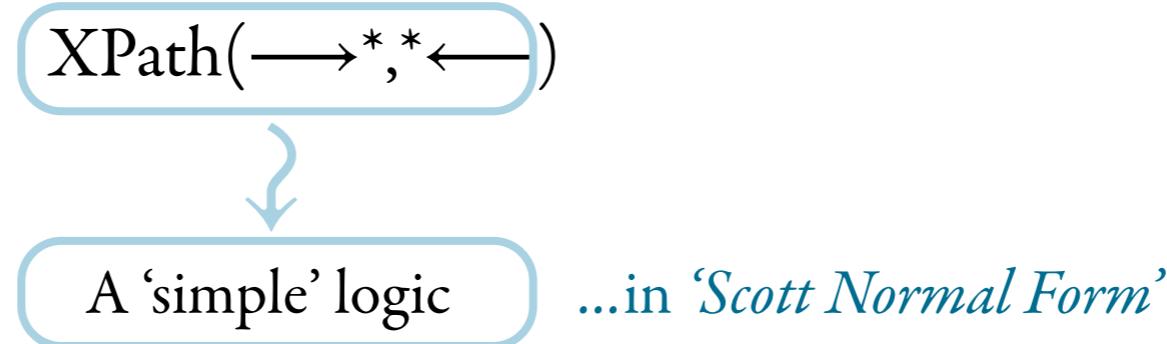
⋮

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

$d \in D$

# SAT-XPath( $\rightarrow^*, * \leftarrow$ ) decidable in 2ExpSpace

Proof idea:



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

$$\text{“} \xleftarrow{\phi_n \bullet \dots \bullet \phi_1 \bullet \phi_0} = \xrightarrow{\psi_0 \bullet \psi_1 \bullet \dots \bullet \psi_m} \text{”}$$

$$\langle [\phi_n]^* \leftarrow \dots \leftarrow [\phi_1]^* \leftarrow [\phi_0]^\neq = \rightarrow^* [\psi_0] \rightarrow^* [\psi_1] \rightarrow^* \dots \rightarrow^* [\psi_m] \rangle$$

$$\langle [\phi_n] \rightarrow^* \dots \rightarrow^* [\phi_1] \rightarrow^* [\phi_0]^\neq = \rightarrow^* [\psi_0] \rightarrow^* [\psi_1] \rightarrow^* \dots \rightarrow^* [\psi_m] \rangle$$

$$\langle d = \xrightarrow{\neq}^* [\psi_0] \rightarrow^* [\psi_1] \rightarrow^* \dots \rightarrow^* [\psi_m] \rangle$$

⋮

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

$d \in D$



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  $\phi = \{c \bullet b, c, b\}$

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

$\models \phi$



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  $\phi = \{c \bullet b, c, b\}$

*w*

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



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  $\phi = \{c \bullet b, c, b\}$

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

$\models \phi$



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  $\phi = \{c \bullet b, c, b\}$

*w*

$$\begin{array}{ccccccccc} a & b & b & a & b & c & [a & b & c & c] & a & a & b & c & b \\ 1 & 2 & 4 & 4 & 3 & 1 & 5 & 1 & 1 & 1 & 4 & 4 & 5 & 1 & 4 \end{array} \models \phi$$

*i*                                    *j*



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  $\phi = \{c \bullet b, c, b\}$

**w**

$$\begin{array}{ccccccccc} a & b & b & a & b & c & [a & b & c & c] & a & a & b & c & b \\ 1 & 2 & 4 & 4 & 3 & 1 & 5 & 1 & 1 & 4 & 4 & 5 & 1 & 4 \end{array} \models \phi$$

*i*                                    *j*

internal  
profile

reachable data from the left  $\rightarrow$   
from the right  $\leftarrow$

external

reachable data from the left  $\leftarrow [ ]$   
from the right  $[ ] \rightarrow$

**π**



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  $\phi = \{c \bullet b, c, b\}$

**w**

$$\begin{array}{ccccccccc} a & b & b & a & b & c & a & a & b \\ 1 & 2 & 4 & 4 & 3 & 1 & 5 & 4 & 5 \\ & & & & & \left[ \begin{array}{cccc} a & b & c & c \\ 5 & 1 & 1 & 1 \end{array} \right] & & & \\ & & & & & i & & j & \end{array} \models \phi$$

internal profile

reachable data from the left  $\rightarrow [ ]$   
from the right  $[ \leftarrow ]$

external

reachable data from the left  $\leftarrow [ ]$   
from the right  $[ ] \rightarrow$

**π**

$$\text{abs}(w, i, j) = \pi = \begin{aligned} & [\rightarrow] : \{(1,c), (1,b)\} \\ & [\leftarrow] : \{(1,c), (1,c \bullet b), (1,b)\} \\ & [ ] \rightarrow : \{(5,b), (4,b), (4,c \bullet b), (1,c)\} \\ & \leftarrow [ ] : \{(1,c), (3,c \bullet b), (4,c \bullet b), (4,b), (3,b), (2,b)\} \end{aligned}$$



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  $\phi = \{c \bullet b, c, b\}$

**w**

$$\begin{array}{ccccccccc} a & b & b & a & b & c & a & a & b \\ 1 & 2 & 4 & 4 & 3 & 1 & 5 & 4 & 5 \end{array} \left[ \begin{array}{cccc} a & b & c & c \\ 5 & 1 & 1 & 1 \end{array} \right] \begin{array}{ccccccccc} b & c & b & a & a & b & c & b \\ 4 & 1 & 4 & 5 & 1 & 4 \end{array} \models \phi$$

*i*                            *j*

internal  
profile

reachable data from the left [→]  
from the right [←]

external

reachable data from the left ←[ ]  
from the right [ ]→

**π**

$$\text{abs}(w,i,j) = \pi = \begin{aligned} [\rightarrow] : & \{(1,c), (1,b)\} \\ [\leftarrow] : & \{(1,c), (1,c \bullet b), (1,b)\} \\ [ ]\rightarrow : & \{(5,b), (4,b), (4,c \bullet b), (1,c)\} \\ \leftarrow[ ] : & \{(1,c), (3,c \bullet b), (4,c \bullet b), (4,b), (3,b), (2,b)\} \end{aligned}$$

≤



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  $\phi = \{c \bullet b, c, b\}$

$w$

$$\begin{array}{ccccccccc} a & b & b & \textcolor{red}{b} & a & \textcolor{red}{a} & b & c & \\ 1 & 2 & 4 & \textcolor{red}{9} & 4 & \textcolor{red}{9} & 3 & 1 & \\ & & & & & & & \left[ \begin{array}{cccc} a & b & c & c \\ 5 & 1 & 1 & 1 \end{array} \right] & \\ & & & & & & & i & j \\ & & & & & & & & \end{array} \quad \begin{array}{ccccccccc} a & \textcolor{red}{a} & a & \textcolor{red}{a} & b & c & b & \textcolor{red}{a} & \\ 4 & \textcolor{red}{9} & 4 & \textcolor{red}{9} & 5 & 1 & 4 & \textcolor{red}{9} & \\ & & & & & & & & \end{array} \models \phi$$

internal profile

reachable data from the left  $\rightarrow [ ]$   
from the right  $[ \leftarrow ]$

external

reachable data from the left  $\leftarrow [ ]$   
from the right  $[ ] \rightarrow$

$\pi$

$\text{abs}(w, i, j) = \pi =$

$$\begin{aligned} & [\rightarrow] : \{ (1,c), (1,b) \} \cup \{ \} \\ & [\leftarrow] : \{ (1,c), (1,c \bullet b), (1,b) \} \cup \{ \} \\ & [ ] \rightarrow : \{ (5,b), (4,b), (4,c \bullet b), (1,c) \} \cup \{ (9,b), (9,c \bullet b) \} \\ & \leftarrow [ ] : \{ (1,c), (3,c \bullet b), (4,c \bullet b), (4,b), (3,b), (2,b) \} \cup \{ (9,c \bullet b), (9,b) \} \end{aligned}$$

$\leq$



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  $\phi = \{c \bullet b, c, b\}$

**w**

$$\begin{array}{ccccccccc} a & b & b & \textcolor{red}{b} & a & \textcolor{red}{a} & b & c & a \\ 1 & 2 & 4 & \textcolor{red}{9} & 4 & \textcolor{red}{9} & 3 & 1 & 5 \end{array} \left[ \begin{array}{cccc} a & b & c & c \\ 5 & 1 & 1 & 1 \end{array} \right] \begin{array}{ccccccccc} a & \textcolor{red}{a} & a & \textcolor{red}{a} & b & c & b & \textcolor{red}{a} \\ 4 & \textcolor{red}{9} & 4 & \textcolor{red}{9} & 5 & 1 & 4 & \textcolor{red}{9} \end{array} \models \phi$$

*i*                            *j*

internal profile

reachable data from the left  $\rightarrow [ ]$   
from the right  $[ \leftarrow ]$

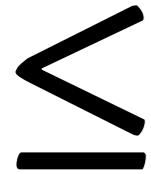
external

reachable data from the left  $\leftarrow [ ]$   
from the right  $[ ] \rightarrow$

**π**

$\text{abs}(w, i, j) = \pi =$

$$\begin{aligned} & [\rightarrow] : \{ (1,c), (1,b) \} \cup \{ \} \\ & [\leftarrow] : \{ (1,c), (1,c \bullet b), (1,b) \} \cup \{ \} \\ & [ ] \rightarrow : \{ (5,b), (4,b), (4,c \bullet b), (1,c) \} \cup \{ (9,b), (9,c \bullet b) \} \\ & \leftarrow [ ] : \{ (1,c), (3,c \bullet b), (4,c \bullet b), (4,b), (3,b), (2,b) \} \cup \{ (9,c \bullet b), (9,b) \} \end{aligned}$$



Induces a **wqo**  $\leq$  on the profiles

.....  $\rightarrow$  every  $\pi_1, \pi_2, \pi_3, \dots$  has some  $i < j$  with  $\pi_i \leq \pi_j$

## Atomic profiles

$$\begin{array}{ccccccccc} a & b & b & a & b & c & \left[ \begin{matrix} a \\ 5 \end{matrix} \right] & b & c \\ 1 & 2 & 4 & 4 & 3 & 1 & & 1 & 1 \end{array} \quad \begin{array}{ccccccccc} c & a & a & b & c & b \\ 1 & 4 & 4 & 5 & 1 & 4 \end{array}$$
$$\pi = \text{abs}(w, i, i+1) \quad (\text{compatible with } \phi)$$

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

## Concatenation of profiles

$$\begin{array}{ccccccccc} a & b & b & a & b & c & a & b & c \\ 1 & 2 & 4 & 4 & 3 & 1 & 5 & 1 & 1 \end{array} \quad \begin{array}{ccccccccc} c & a & a & b & c & b \\ 1 & 4 & 4 & 5 & 1 & 4 \end{array}$$

## Atomic profiles

$$\begin{array}{ccccccccc} a & b & b & a & b & c & \left[ \begin{matrix} a \\ 5 \end{matrix} \right] & b & c \\ 1 & 2 & 4 & 4 & 3 & 1 & & 1 & 1 \end{array} \quad \begin{array}{ccccccccc} c & a & a & b & c & b \\ 1 & 4 & 4 & 5 & 1 & 4 \end{array}$$
$$\pi = \text{abs}(w, i, i+1) \quad (\text{compatible with } \phi)$$

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

## Concatenation of profiles

$$\begin{array}{ccccccccc} a & b & b & \left[ \begin{matrix} a & b & c \\ 4 & 3 & 1 \end{matrix} \right] & a & b & c & c & a \\ 1 & 2 & 4 & & 5 & 1 & 1 & 1 & 4 \end{array} \quad \begin{array}{ccccccccc} a & a & b & c & b \\ 4 & 4 & 5 & 1 & 4 \end{array}$$
$$\pi_I$$

## Atomic profiles

$$\begin{array}{ccccccccc} a & b & b & a & b & c & \left[ \begin{matrix} a \\ 5 \end{matrix} \right] & b & c \\ 1 & 2 & 4 & 4 & 3 & 1 & & 1 & 1 \end{array} \quad \begin{array}{ccccccccc} c & a & a & b & c & b \\ 1 & 4 & 4 & 5 & 1 & 4 \end{array}$$
$$\pi = \text{abs}(w, i, i+1) \quad (\text{compatible with } \phi)$$

**Atomic<sub>φ</sub>** = all φ-abstractions of positions within data words

## Concatenation of profiles

$$\begin{array}{ccccccccc} a & b & b & \left[ \begin{matrix} a & b & c \\ 4 & 3 & 1 \end{matrix} \right] & \left[ \begin{matrix} a & b & c & c \\ 5 & 1 & 1 & 1 \end{matrix} \right] & a & a & b & c & b \\ 1 & 2 & 4 & & & 4 & 4 & 5 & 1 & 4 \end{array}$$
$$\pi_1 \quad \pi_2$$

## Atomic profiles

$$\begin{array}{ccccccccc} a & b & b & a & b & c & \left[ \begin{matrix} a \\ 5 \end{matrix} \right] & b & c \\ 1 & 2 & 4 & 4 & 3 & 1 & & 1 & 1 \end{array} \quad \begin{array}{ccccccccc} c & a & a & b & c & b \\ 1 & 4 & 4 & 5 & 1 & 4 \end{array}$$
$$\pi = \text{abs}(w, i, i+1) \quad (\text{compatible with } \phi)$$

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

## Concatenation of profiles

$$\begin{array}{ccccccccc} a & b & b & \left[ \begin{matrix} a & b & c \\ 4 & 3 & 1 \end{matrix} \right] & \left[ \begin{matrix} a & b & c & c \\ 5 & 1 & 1 & 1 \end{matrix} \right] & a & a & b & c & b \\ 1 & 2 & 4 & & & 4 & 4 & 5 & 1 & 4 \end{array}$$
$$\pi_3 = \pi_1 \cdot \pi_2$$

## Atomic profiles

$$\begin{array}{ccccccccc}
 a & b & b & a & b & c & \left[ \begin{matrix} a \\ 5 \end{matrix} \right] & b & c \\
 1 & 2 & 4 & 4 & 3 & 1 & 1 & 1 & 1 \\
 \end{array}
 \quad \pi = \text{abs}(w, i, i+1) \quad (\text{compatible with } \phi)$$

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

## Concatenation of profiles

$$\begin{array}{ccccccccc}
 a & b & b & \left[ \begin{matrix} a & b & c \\ 4 & 3 & 1 \end{matrix} \right] & \left[ \begin{matrix} a & b & c & c \\ 5 & 1 & 1 & 1 \end{matrix} \right] & a & a & b & c & b \\
 1 & 2 & 4 & & & 4 & 4 & 5 & 1 & 4 \\
 \end{array}
 \quad \pi_3 = \pi_1 \cdot \pi_2$$

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

Der

:.....

$\rightarrow : \emptyset$

$\leftarrow : \emptyset$

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

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

## Atomic profiles

$$\begin{array}{ccccccccc}
 a & b & b & a & b & c & \left[ \begin{matrix} a \\ 5 \end{matrix} \right] & b & c \\
 1 & 2 & 4 & 4 & 3 & 1 & 1 & 1 & 1 \\
 \end{array}
 \quad \pi = \text{abs}(w, i, i+1) \quad (\text{compatible with } \phi)$$

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

## Concatenation of profiles

$$\begin{array}{ccccccccc}
 a & b & b & \left[ \begin{matrix} a & b & c \\ 4 & 3 & 1 \end{matrix} \right] & \left[ \begin{matrix} a & b & c & c \\ 5 & 1 & 1 & 1 \end{matrix} \right] & a & a & b & c & b \\
 1 & 2 & 4 & & & 4 & 4 & 5 & 1 & 4 \\
 \end{array}
 \quad \pi_3 = \pi_1 \cdot \pi_2$$

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

Der

$\vdots \dots \dots \rightarrow [ ] \rightarrow : \emptyset$

$\leftarrow [ ] : \emptyset$

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

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

*How to compute Der?*

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

$$\vdots \dots \rightarrow \{ \pi \mid \pi' \leq \pi, \pi' \in \text{Der} \}$$

2  $\text{MIN( Atomic )}$  is finite and computable

$1+2 =$

```
R0 = MIN( Atomic );
while (Ri ≠ Ri+1):
    Ri+1 = MIN(Ri ∪  $\uparrow R_i \cdot \uparrow R_i$ )
```

return( R<sub>i</sub> );

computes  $\text{MIN}(\text{Der})$

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

$$\vdots \dots \rightarrow \{ \pi \mid \pi' \leq \pi, \pi' \in \text{Der} \}$$

2  $\text{MIN( Atomic )}$  is finite and computable



$1+2 =$

```
R0 = MIN( Atomic );  
while (Ri ≠ Ri+1):  
    Ri+1 = MIN(Ri ∪  $\uparrow$ Ri ∙  $\uparrow$ Ri)
```

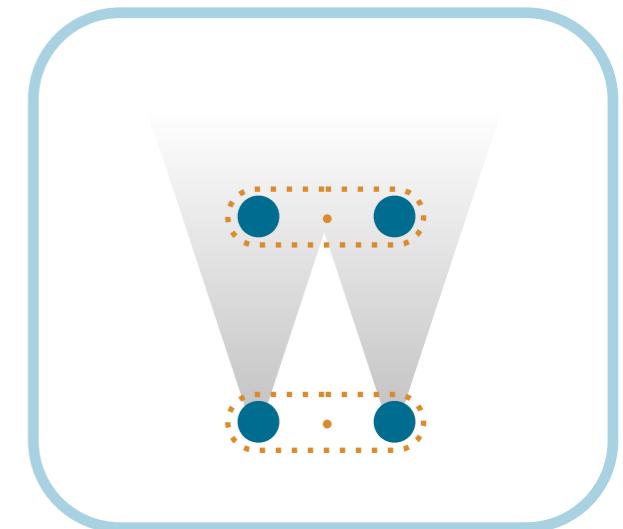
return( R<sub>i</sub> );

computes  $\text{MIN}(\text{Der})$

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

$$\vdots \dots \dots \rightarrow \{ \pi \mid \pi' \leq \pi, \pi' \in \text{Der} \}$$

2  $\text{MIN( Atomic )}$  is finite and computable



$1+2 =$

```
R0 = MIN( Atomic );  
while (Ri ≠ Ri+1) :  
    Ri+1 = MIN(Ri ∪  $\uparrow R_i \cdot \uparrow R_i$ )
```

return( R<sub>i</sub> );

computes  $\text{MIN}(\text{Der})$

1

By property before:  $\text{Der} = \uparrow \text{Der}$

$$\vdots \dots \rightarrow \{ \pi \mid \pi' \leq \pi, \pi' \in \text{Der} \}$$

2

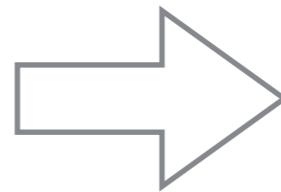
$\text{MIN( Atomic )}$  is finite and computable

3

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

$$\begin{matrix} \vdots \\ \text{VI} \\ \vdots \end{matrix} \quad \begin{matrix} \vdots \\ \text{VI} \\ \vdots \end{matrix}$$

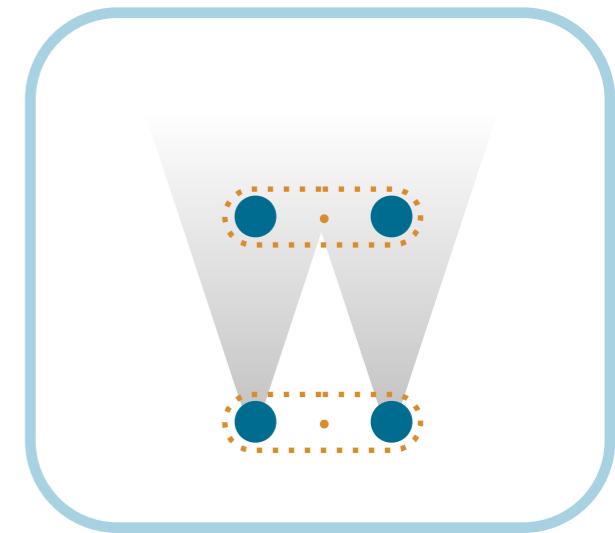
$$\pi'_1 \quad \pi'_2$$



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

$$\begin{matrix} \vdots \\ \text{VI bounded} \\ \vdots \end{matrix} \quad \begin{matrix} \vdots \\ \text{VI bounded} \\ \vdots \end{matrix}$$

$$\pi'_1 \quad \pi'_2$$



1+2

=

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

return( R<sub>i</sub> );

computes  $\text{MIN}(\text{Der})$

1

By property before:  $\text{Der} = \uparrow \text{Der}$

$$\vdots \dots \rightarrow \{ \pi \mid \pi' \leq \pi, \pi' \in \text{Der} \}$$

2

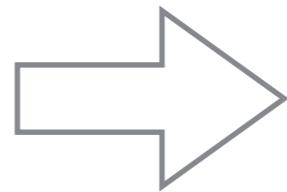
$\text{MIN( Atomic )}$  is finite and computable

3

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

$$\begin{matrix} \vdots \\ \text{VI} \\ \vdots \end{matrix} \quad \begin{matrix} \vdots \\ \text{VI} \\ \vdots \end{matrix}$$

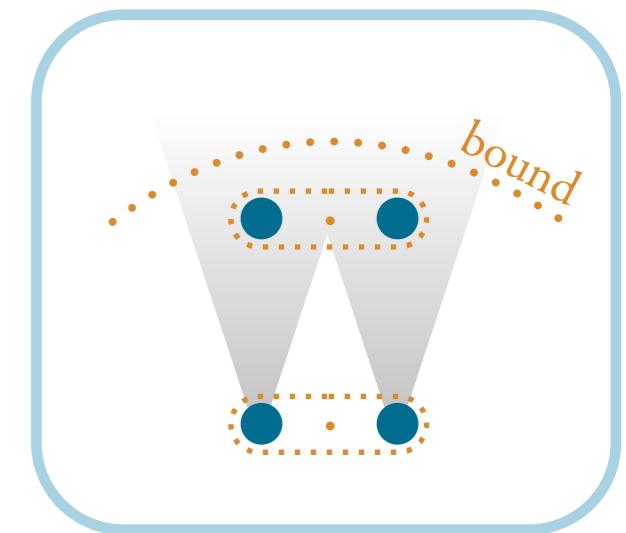
$$\pi'_1 \quad \pi'_2$$



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

$$\begin{matrix} \vdots \\ \text{VI bounded} \\ \vdots \end{matrix} \quad \begin{matrix} \vdots \\ \text{VI bounded} \\ \vdots \end{matrix}$$

$$\pi'_1 \quad \pi'_2$$



1+2

=

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

return( R<sub>i</sub> );

computes  $\text{MIN}(\text{Der})$

1

By property before:  $\text{Der} = \uparrow \text{Der}$

$$\vdots \dots \rightarrow \{ \pi \mid \pi' \leq \pi, \pi' \in \text{Der} \}$$

2

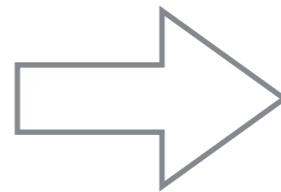
$\text{MIN}(\text{Atomic})$  is finite and computable

3

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

$$\begin{array}{c} \vdots \\ \text{VI} \\ \vdots \end{array} \quad \begin{array}{c} \vdots \\ \text{VI} \\ \vdots \end{array}$$

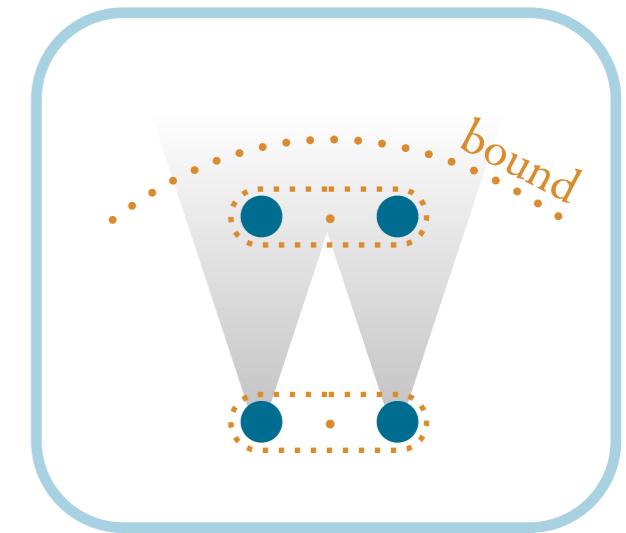
$$\pi'_1 \quad \pi'_2$$



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

$$\begin{array}{c} \vdots \\ \text{VI} \\ \vdots \text{ bounded} \\ \vdots \\ \text{VI} \\ \vdots \text{ bounded} \end{array} \quad \begin{array}{c} \vdots \\ \text{VI} \\ \vdots \text{ bounded} \\ \vdots \\ \text{VI} \\ \vdots \text{ bounded} \end{array}$$

$$\pi'_1 \quad \pi'_2$$



1+2+3=

$$R_0 = \text{MIN}(\text{Atomic});$$

while ( $R_i \neq R_{i+1}$ ) :

$$R_{i+1} = \text{MIN}(R_i \cup \uparrow R_i \cdot \uparrow R_i)$$

$$= \text{MIN}(R_i \cup \uparrow_{\text{bounded}} R_i \cdot \uparrow_{\text{bounded}} R_i);$$

return(  $R_i$  );

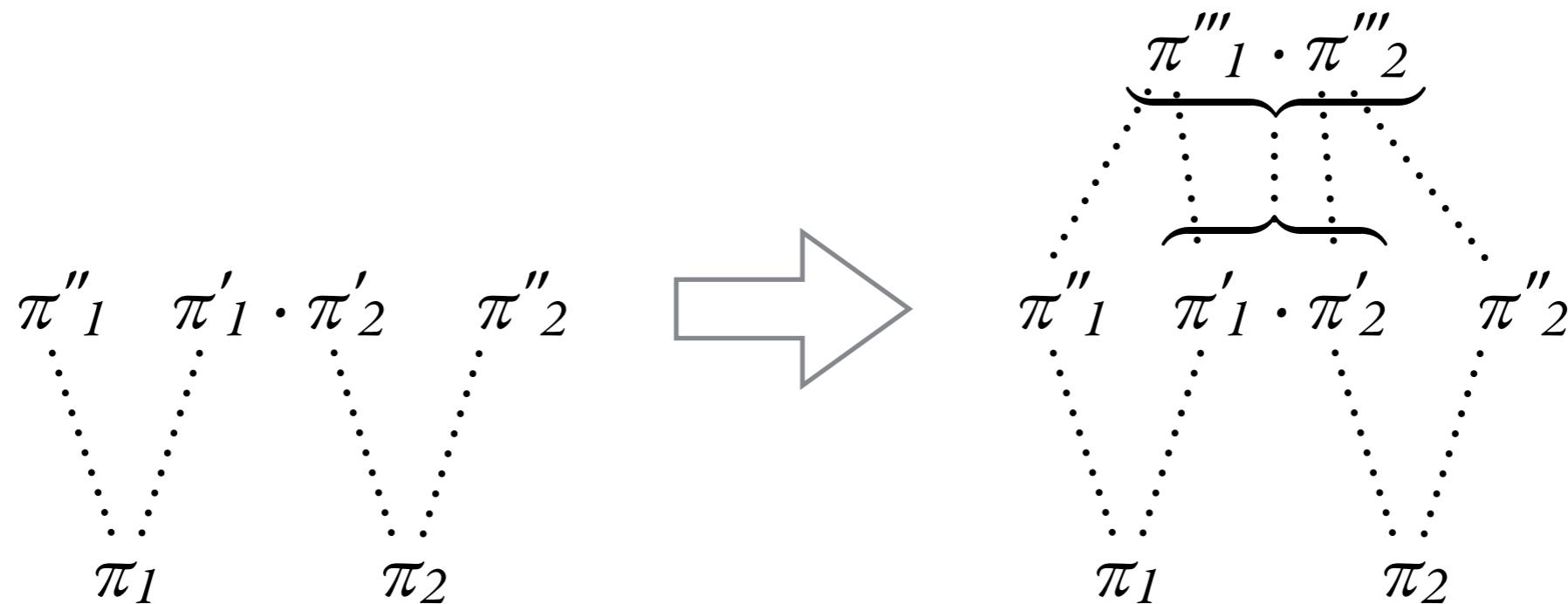
computes  $\text{MIN}(\text{Der})$

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

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

$\Rightarrow$  no need to compute  $\text{MIN}(\text{Der})$ , it suffices to compute  $\text{MIN}(\downarrow \text{Der})$ .

4

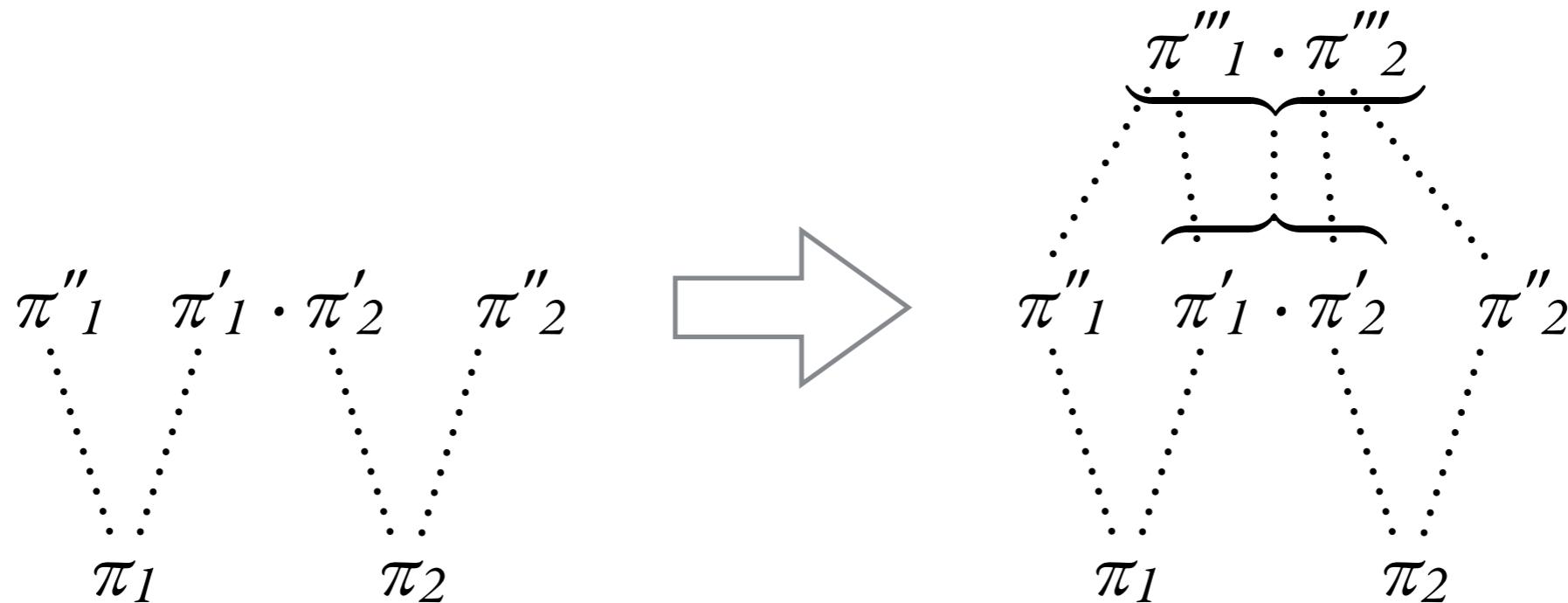


**Complexity?** In principle, **non-primitive recursive**.

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

$\Rightarrow$  no need to compute  $\text{MIN}(\text{Der})$ , it suffices to compute  $\text{MIN}(\downarrow \text{Der})$ .

4



1+2+3+4 =

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

computes  $\text{MIN}(\downarrow \text{Der})$

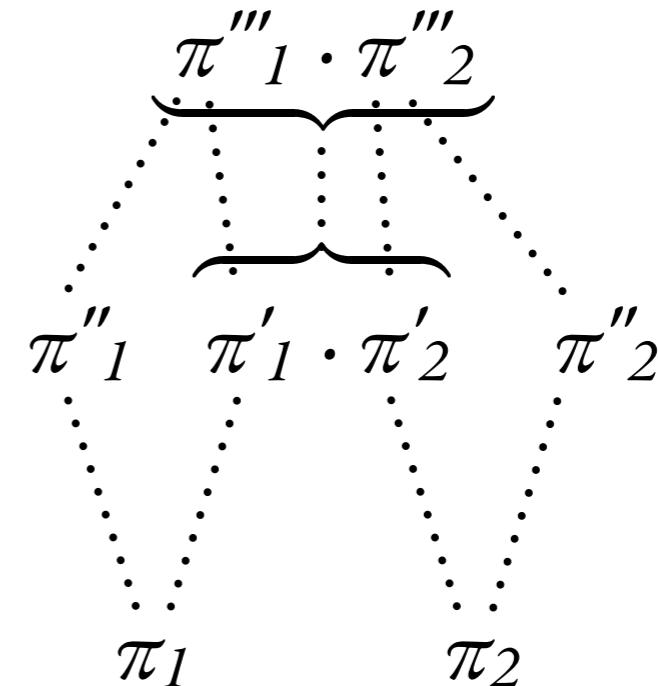
**Complexity?** In principle, **non-primitive recursive**.

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

$\Rightarrow$  no need to compute  $\text{MIN}(\text{Der})$ , it suffices to compute  $\text{MIN}(\downarrow \text{Der})$ .

4

$$\pi''_1 \quad \pi'_1 \cdot \pi'_2 \quad \pi''_2 \quad \rightarrow \quad \text{MIN}(\downarrow \text{Der}) \ni \pi_1 \quad \pi_2 \in \text{MIN}(\downarrow \text{Der})$$



1+2+3+4 =

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

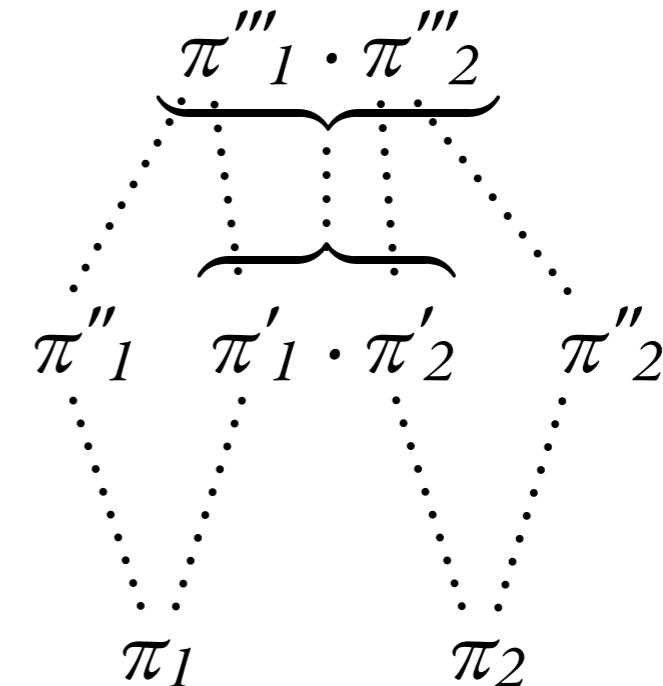
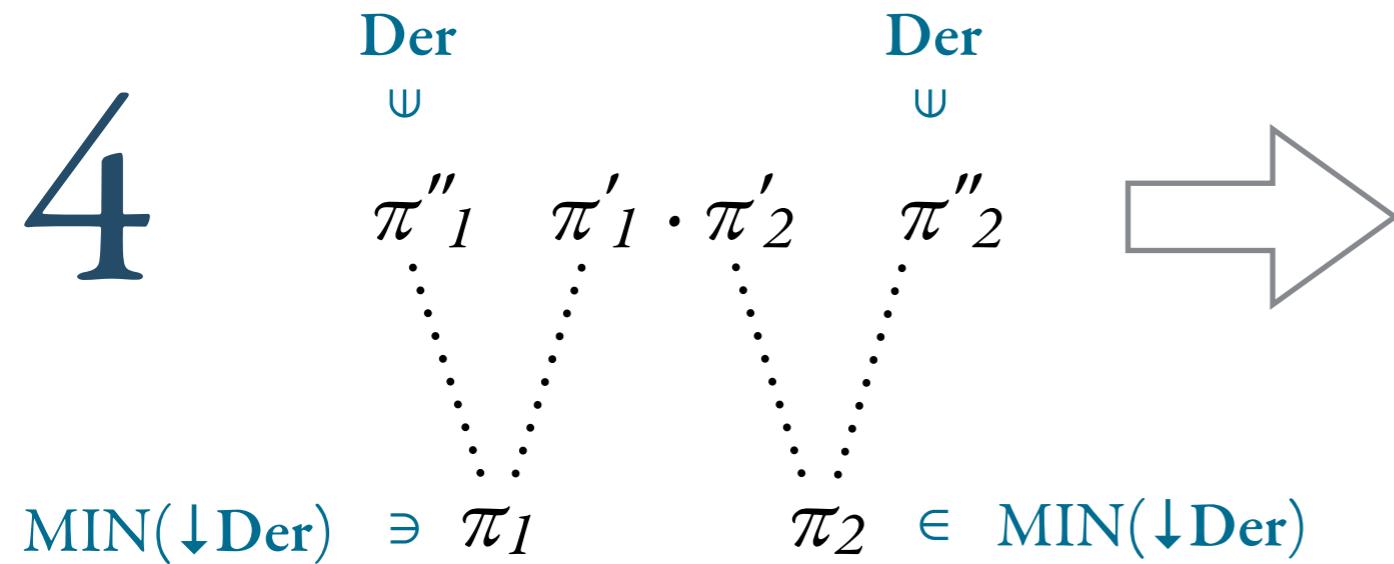
computes  $\text{MIN}(\downarrow \text{Der})$

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

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

$\Rightarrow$  no need to compute  $\text{MIN}(\text{Der})$ , it suffices to compute  $\text{MIN}(\downarrow \text{Der})$ .

4



$1+2+3+4 =$

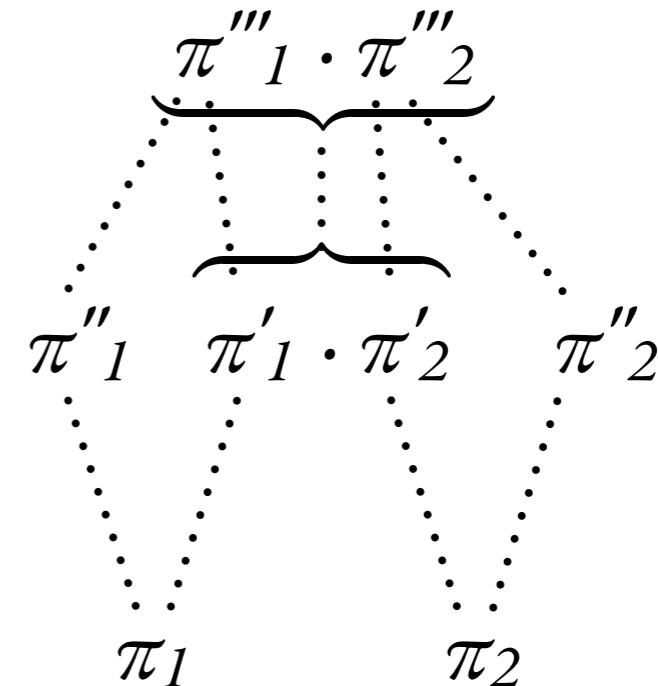
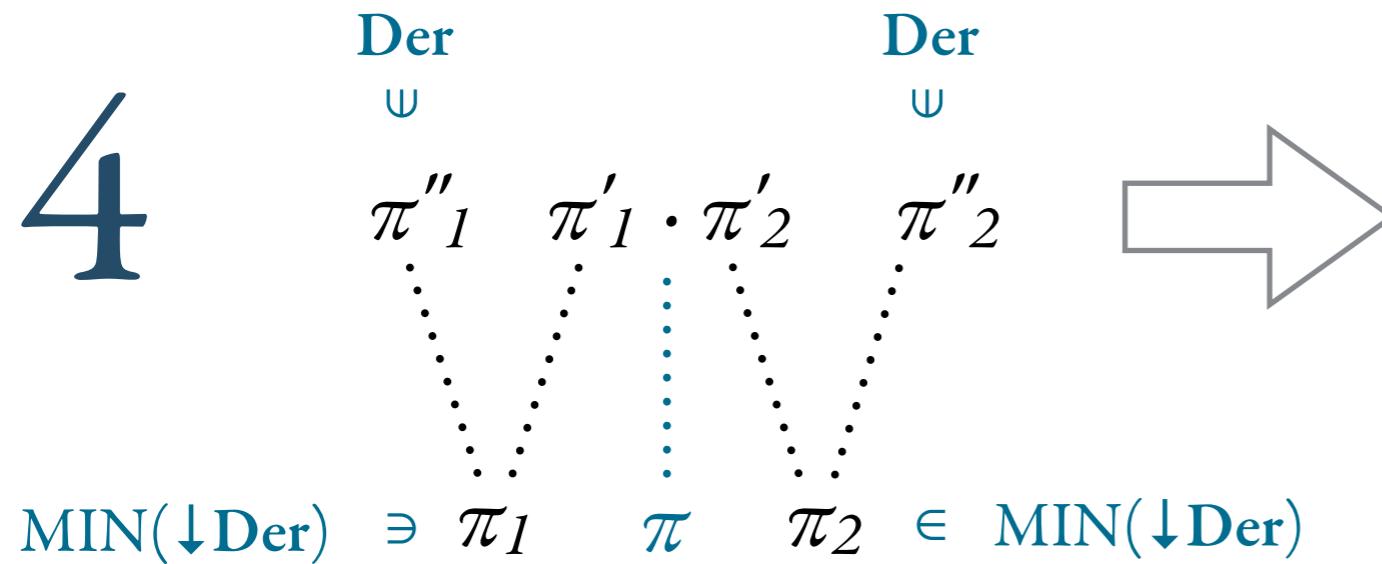
```
R0 = MIN( Atomic );
while (Ri ≠ Ri+1):
    Ri+1 = MIN↓(Ri ∪ ↑boundedRi · ↑boundedRi);
return( Ri );
```

computes  $\text{MIN}(\downarrow \text{Der})$

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

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

$\Rightarrow$  no need to compute  $\text{MIN}(\text{Der})$ , it suffices to compute  $\text{MIN}(\downarrow \text{Der})$ .



$1+2+3+4 =$

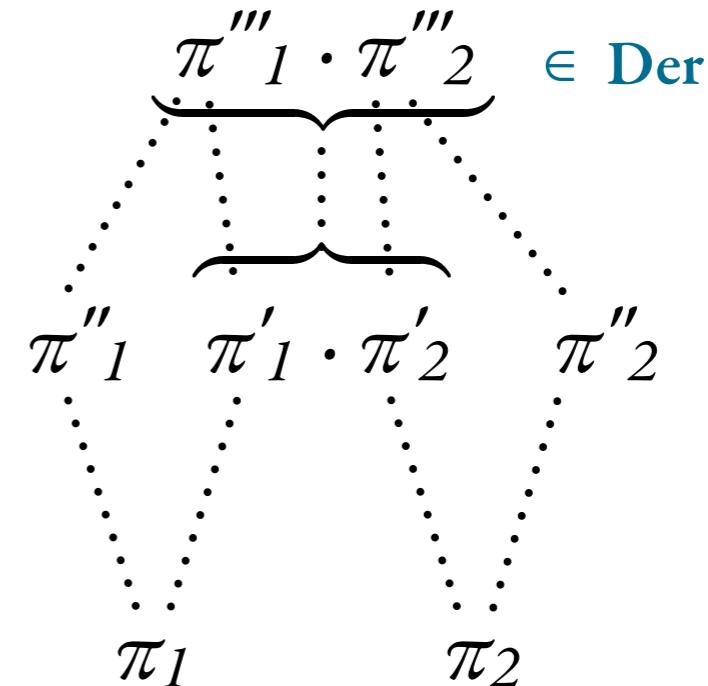
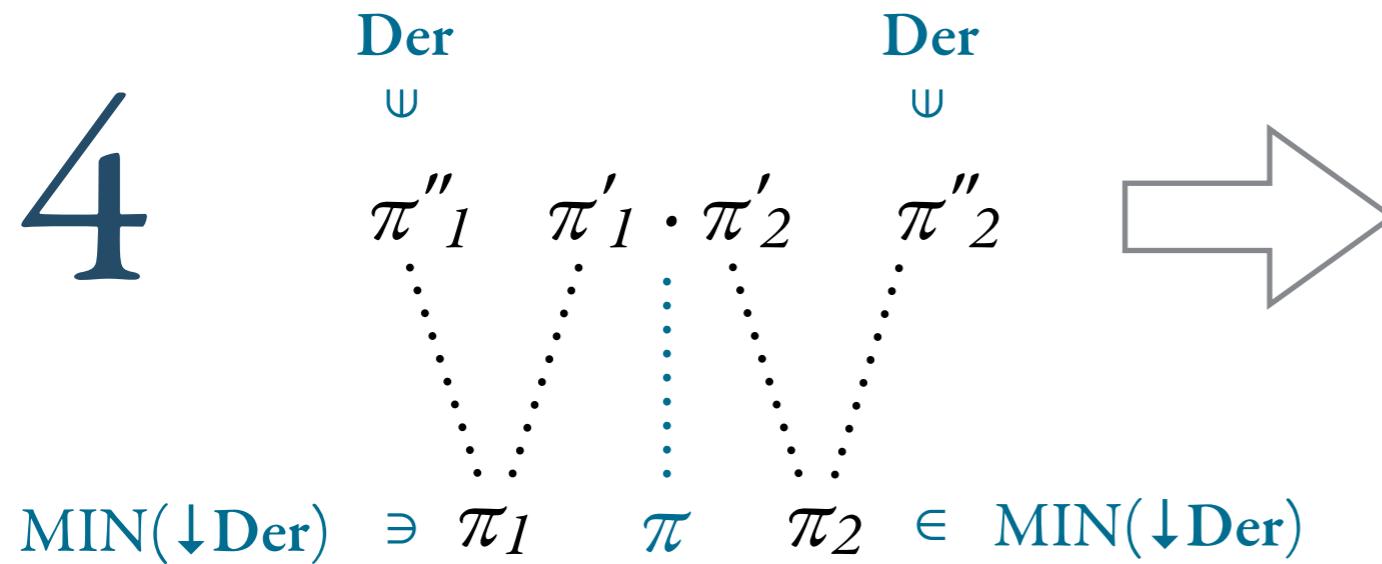
```
R0 = MIN( Atomic );  
while (Ri ≠ Ri+1):  
    Ri+1 = MIN↓(Ri ∪ ↑boundedRi · ↑boundedRi);  
return( Ri );
```

computes  $\text{MIN}(\downarrow \text{Der})$

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

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

$\Rightarrow$  no need to compute  $\text{MIN}(\text{Der})$ , it suffices to compute  $\text{MIN}(\downarrow \text{Der})$ .



1+2+3+4 =

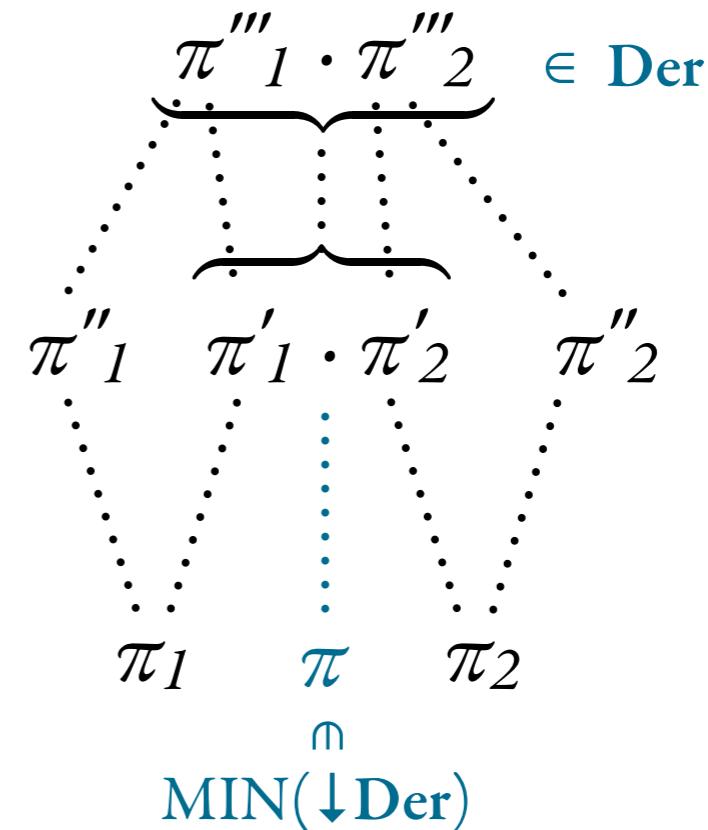
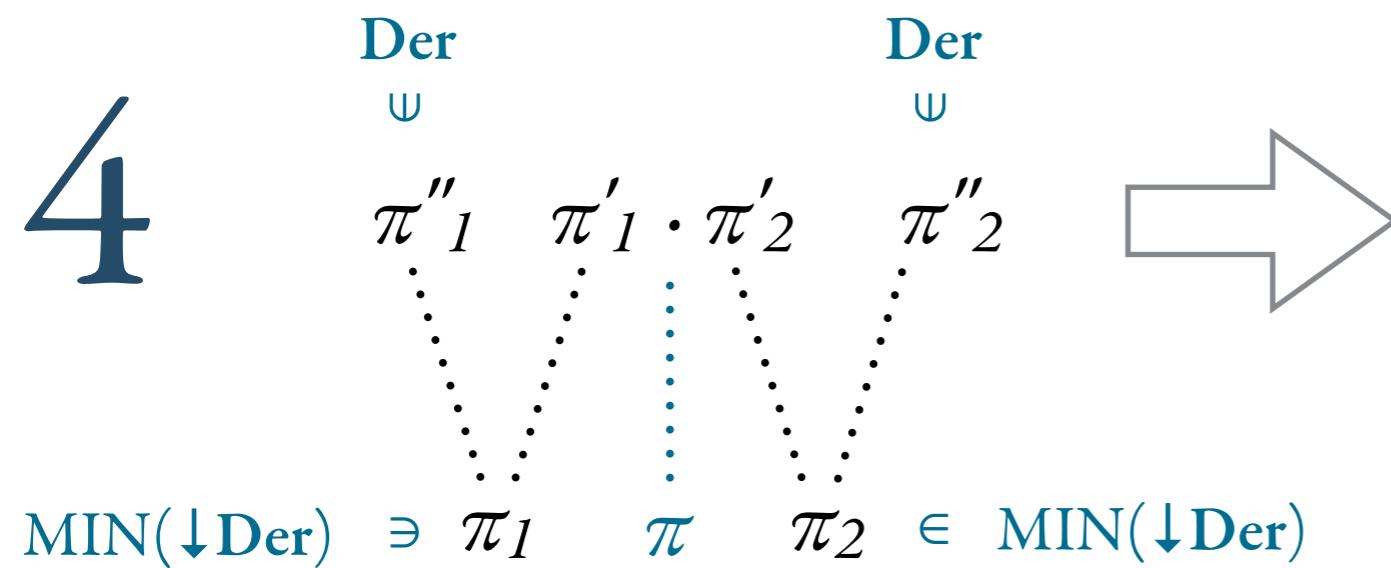
```
R0 = MIN( Atomic );
while (Ri ≠ Ri+1) :
    Ri+1 = MIN↓(Ri ∪ ↑boundedRi · ↑boundedRi);
return( Ri );
```

computes  $\text{MIN}(\downarrow \text{Der})$

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

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

$\Rightarrow$  no need to compute  $\text{MIN}(\text{Der})$ , it suffices to compute  $\text{MIN}(\downarrow \text{Der})$ .



1+2+3+4 =

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

computes  $\text{MIN}(\downarrow \text{Der})$

# Complexity?

$2^\phi$  many MIN(Profiles)  $\Rightarrow$  2ExpSpace procedure

# Complexity?

$2^\phi$  many MIN(Profiles)  $\Rightarrow$  2ExpSpace procedure

## Caveat



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?

$2^\phi$  many MIN(Profiles)  $\Rightarrow$  2ExpSpace procedure

## Caveat



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    $\begin{bmatrix} c \\ 1 \end{bmatrix}$    c   a   a   b   c   b    $\models \phi$

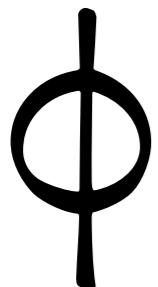
$\sqcap$

**Atomic** $_\phi$

# Complexity?

$2^\phi$  many MIN(Profiles)  $\Rightarrow$  2ExpSpace procedure

## Caveat



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

$\nexists$   
**Atomic** $_{\phi}$

# Complexity?

$2^\phi$  many MIN(Profiles)  $\Rightarrow$  2ExpSpace procedure

## Caveat



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

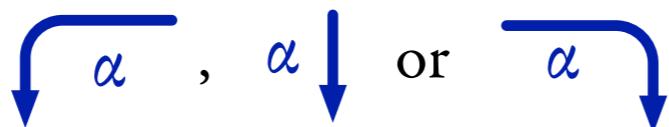
$\sqcap$   
**Atomic** $_\phi$

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

# From words to trees

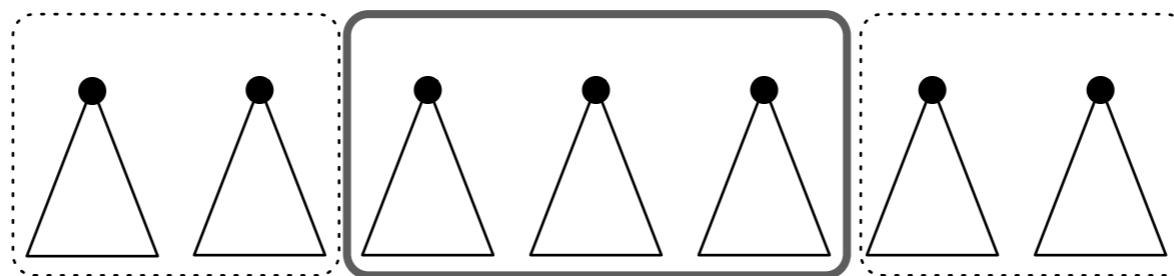
Satisfiability for  $\text{XPath}(*\leftarrow, \downarrow_*, \rightarrow^*)$  is decidable in 2ExpSpace. ♦

Composed paths

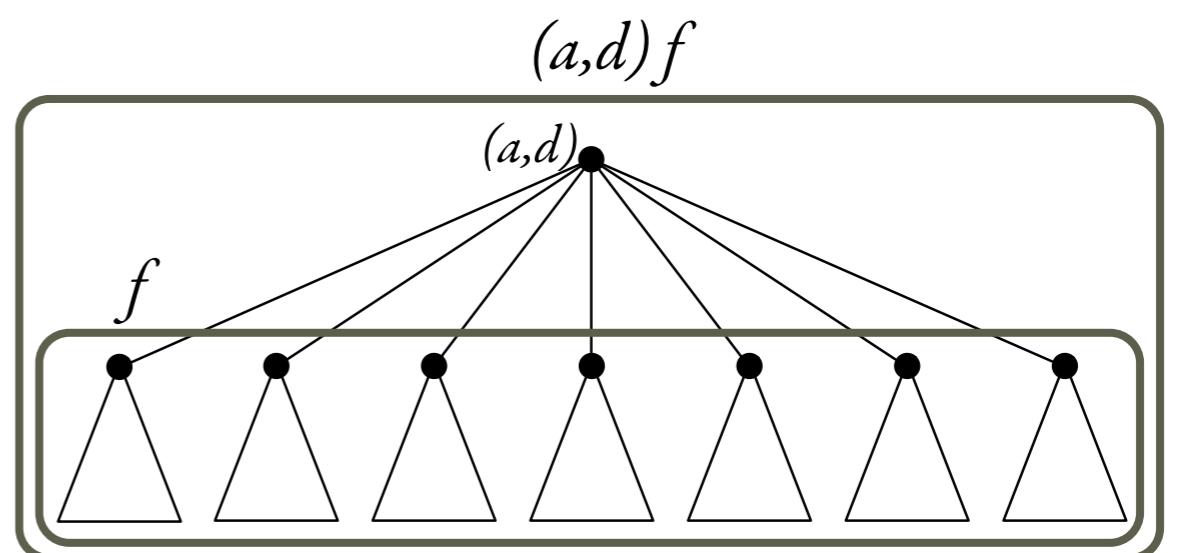
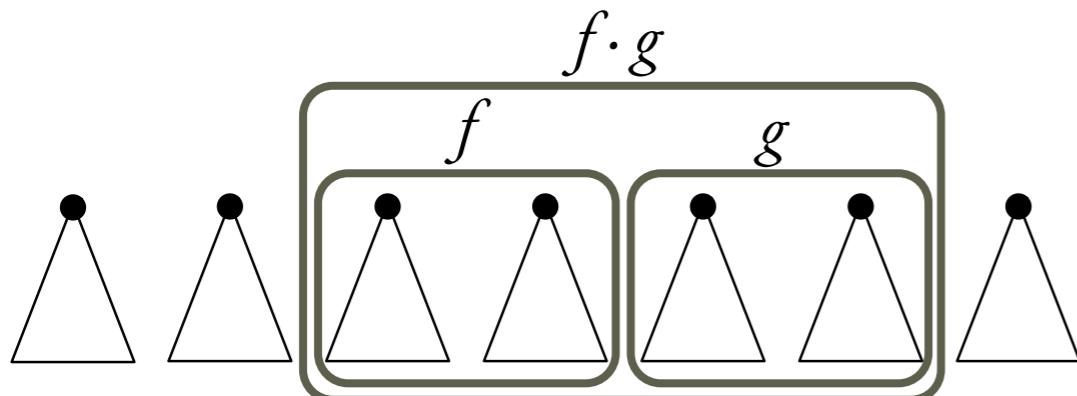


eg:  $\rightarrow^*[a] \rightarrow^*[b] \downarrow_*[c]$

Forest profiles



Two-operator algebra



# Final remarks

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

Adding  $\uparrow^*$ : non-PR

Adding domain-dependant relations?

Adding  $\downarrow$ : still decidable?

Complexity: 3ExpSpace (2ExpSpace in normal form)

thank you!