PEBBLES

or

Automata with Nested Pebbles

capture

FO Logic with Transitive Closure

Joost Engelfriet
Hendrik Jan Hoogeboom
Leiden NL

DSPACE(log *n*)

deterministic logarithmic space
(strings)

Immerman

First-Order Logic
+ transitive closure

Multi-Head Automata

$\varphi^*(\underline{x},\underline{y})$

arity k $\Longleftrightarrow$ k heads

this talk

on trees, grids, ...

First-Order Logic
+ transitive closure

Multi-Head Automata
+ ‘nested pebbles’

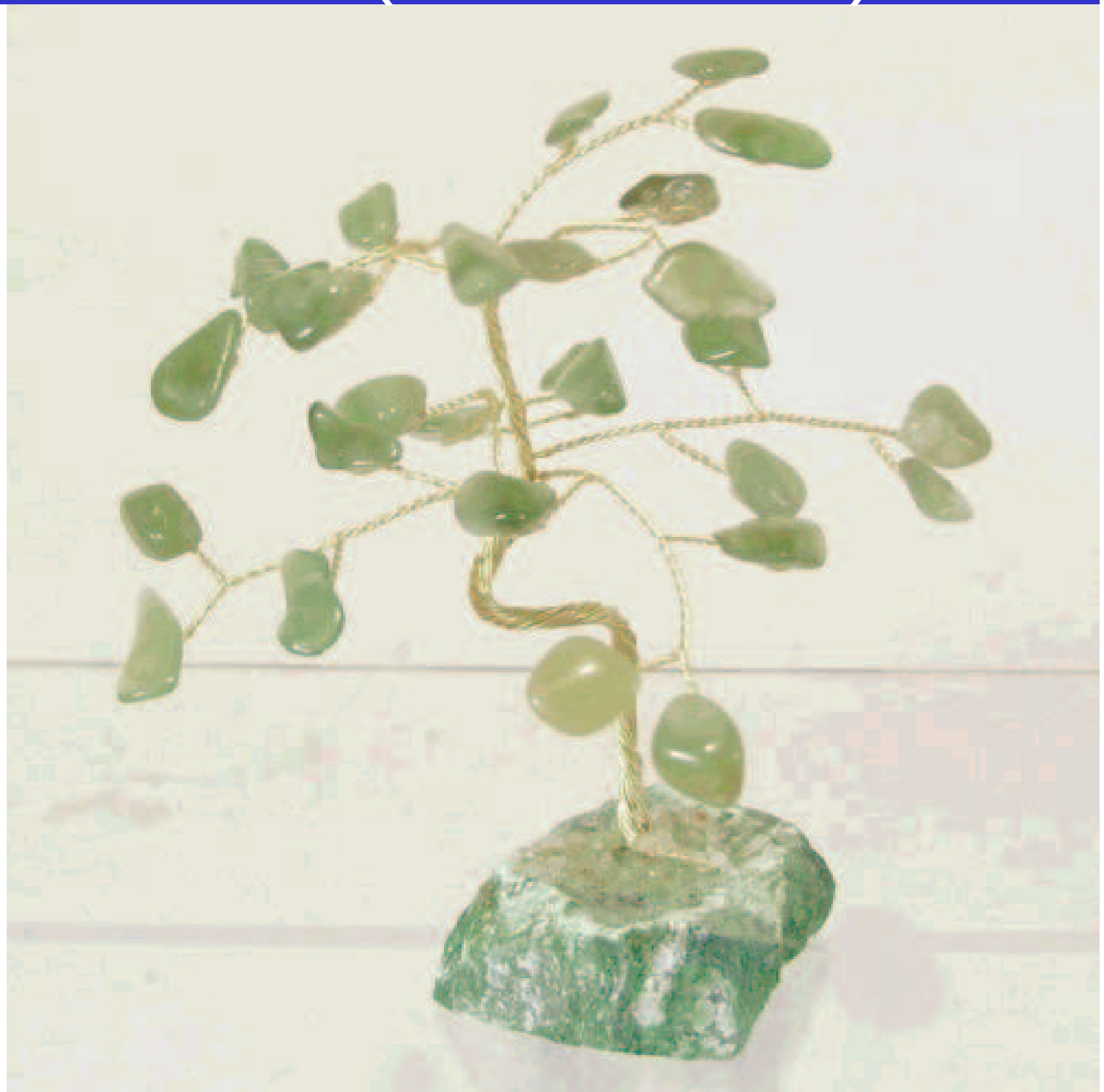
unary ⟺ single head

trees
k=1

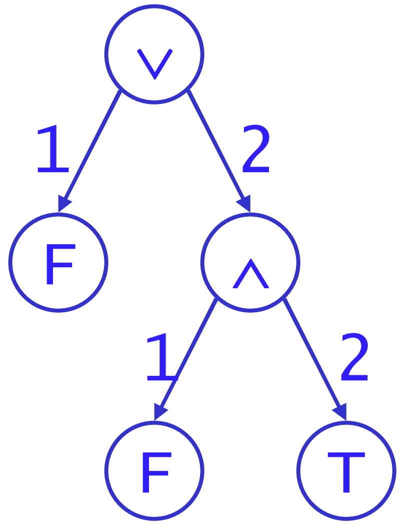
background

- transitive closure
  - k-ary tc vs. many heads
  - strings, trees, n-dim grids, …
- XML document transformation
  - single head on (unranked) trees
- graph exploration
  - many heads on graphs
  - grids, toruses, mazes, …

(automata on) trees



formal model

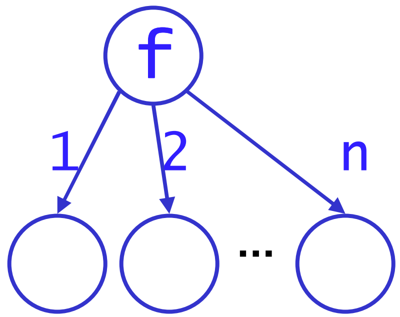


F∨(F∧T)

∨F∧FT

ranked trees

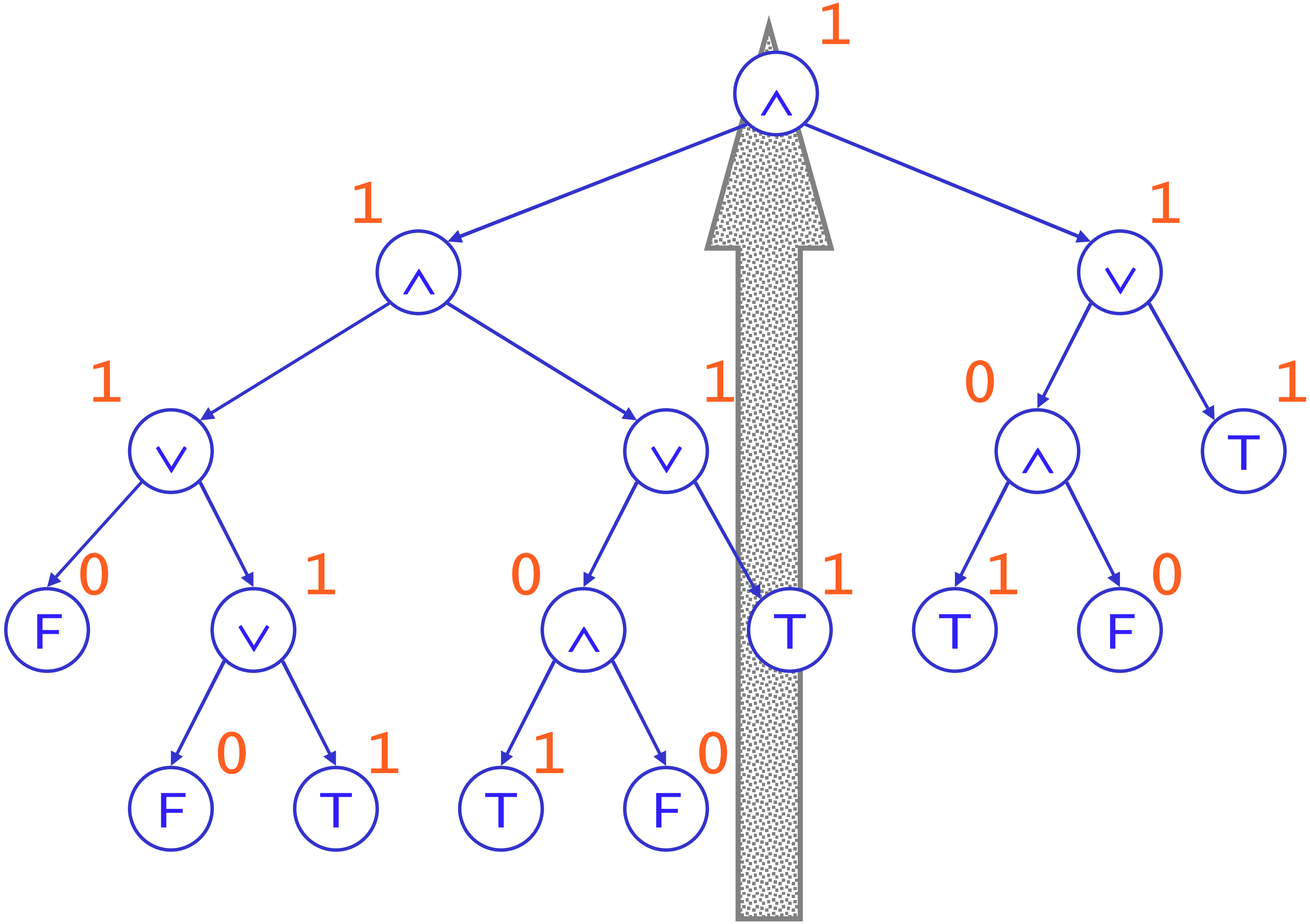
~ terms



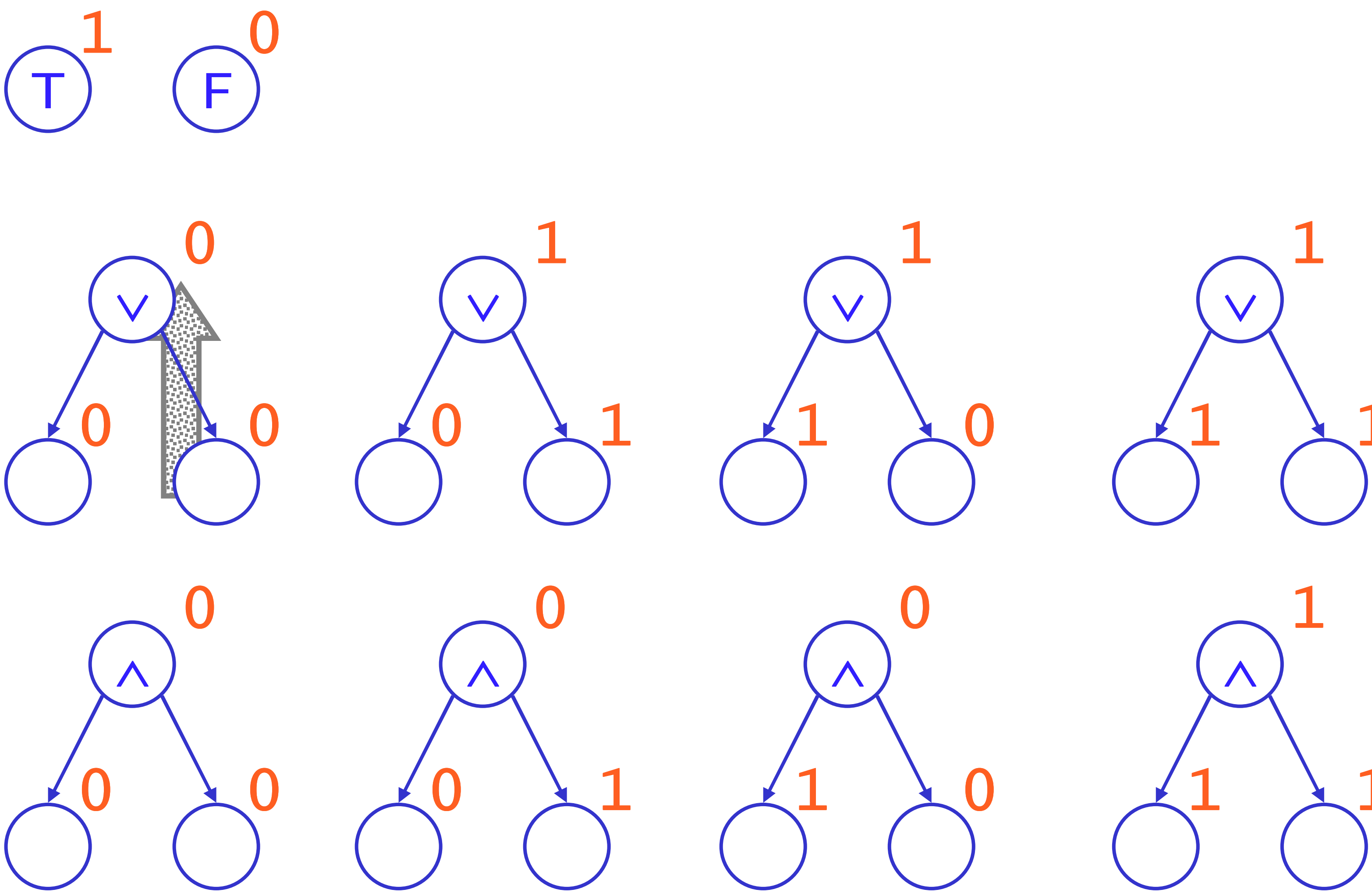
n-ary f

$fx_1x_2\ldots x_n$

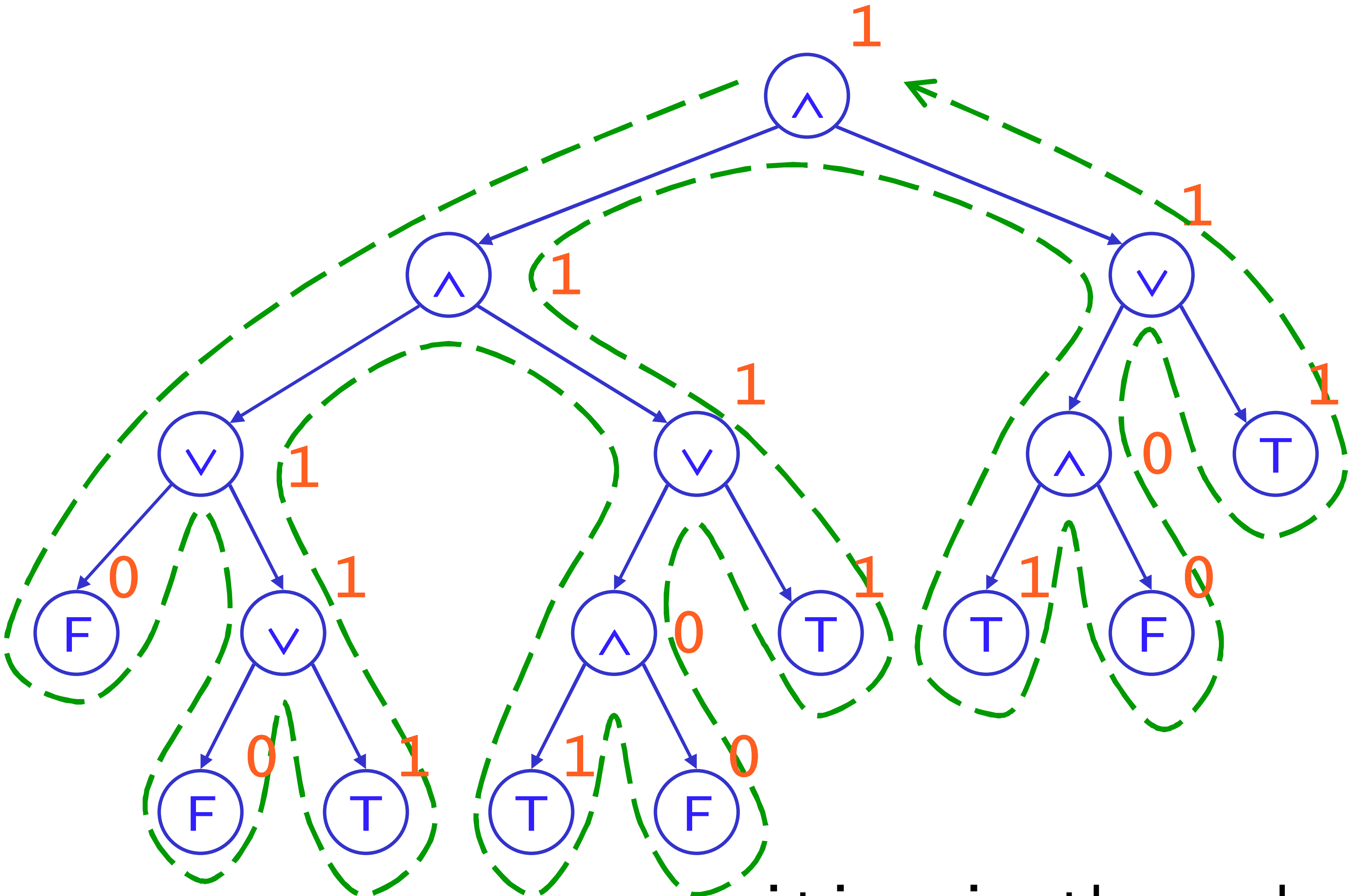
bottom-up evaluation



bottom-up tree automaton

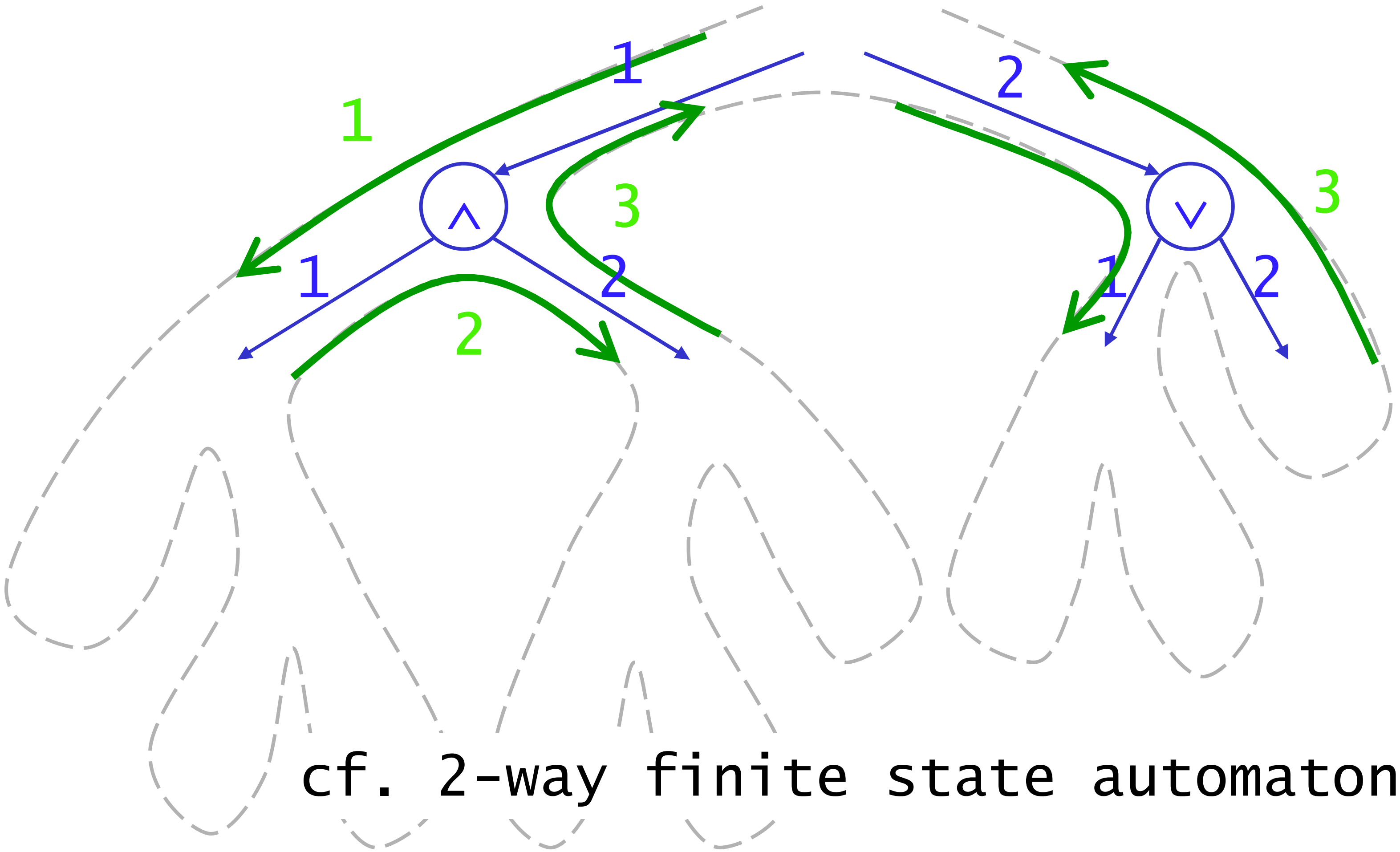


postorder evaluation



… writing in the nodes

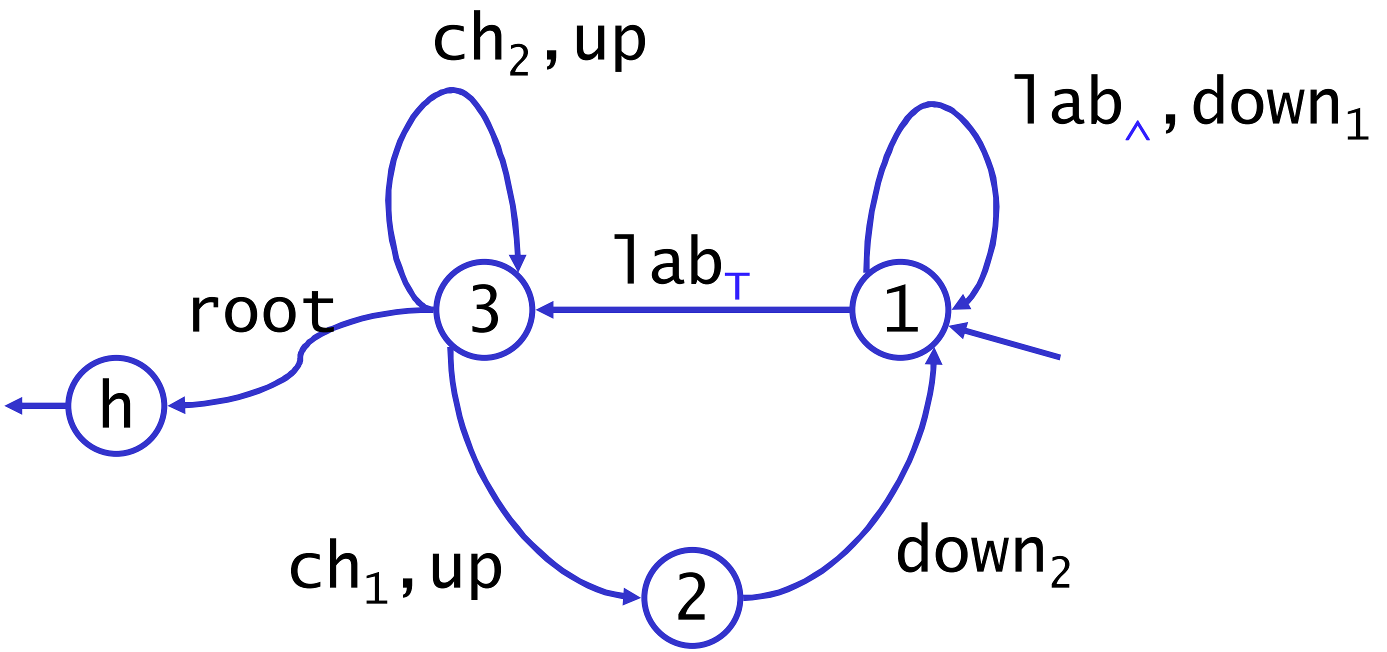
walking along the edges



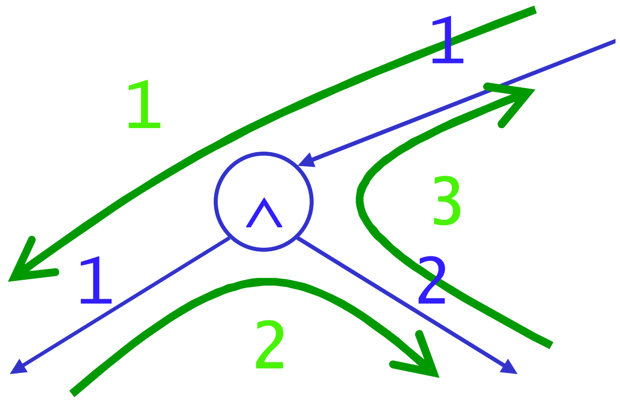
cf. 2-way finite state automaton

tree traversal

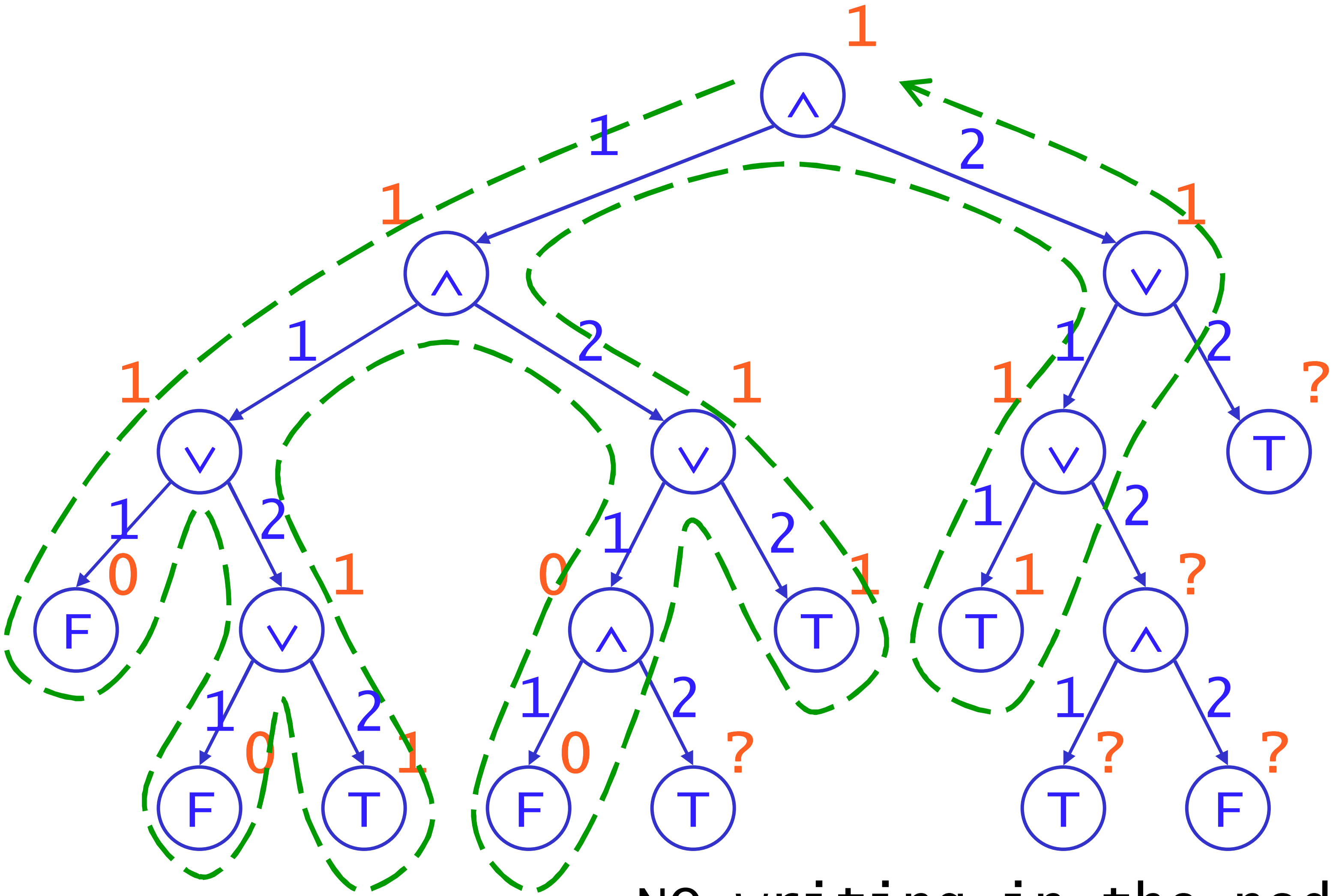
TWA



first child
incoming edge



boolean tree-walk



… NO writing in the nodes

basic models

MSO
second-order logic

REG
tree automata

‘recursive’ bottom-up

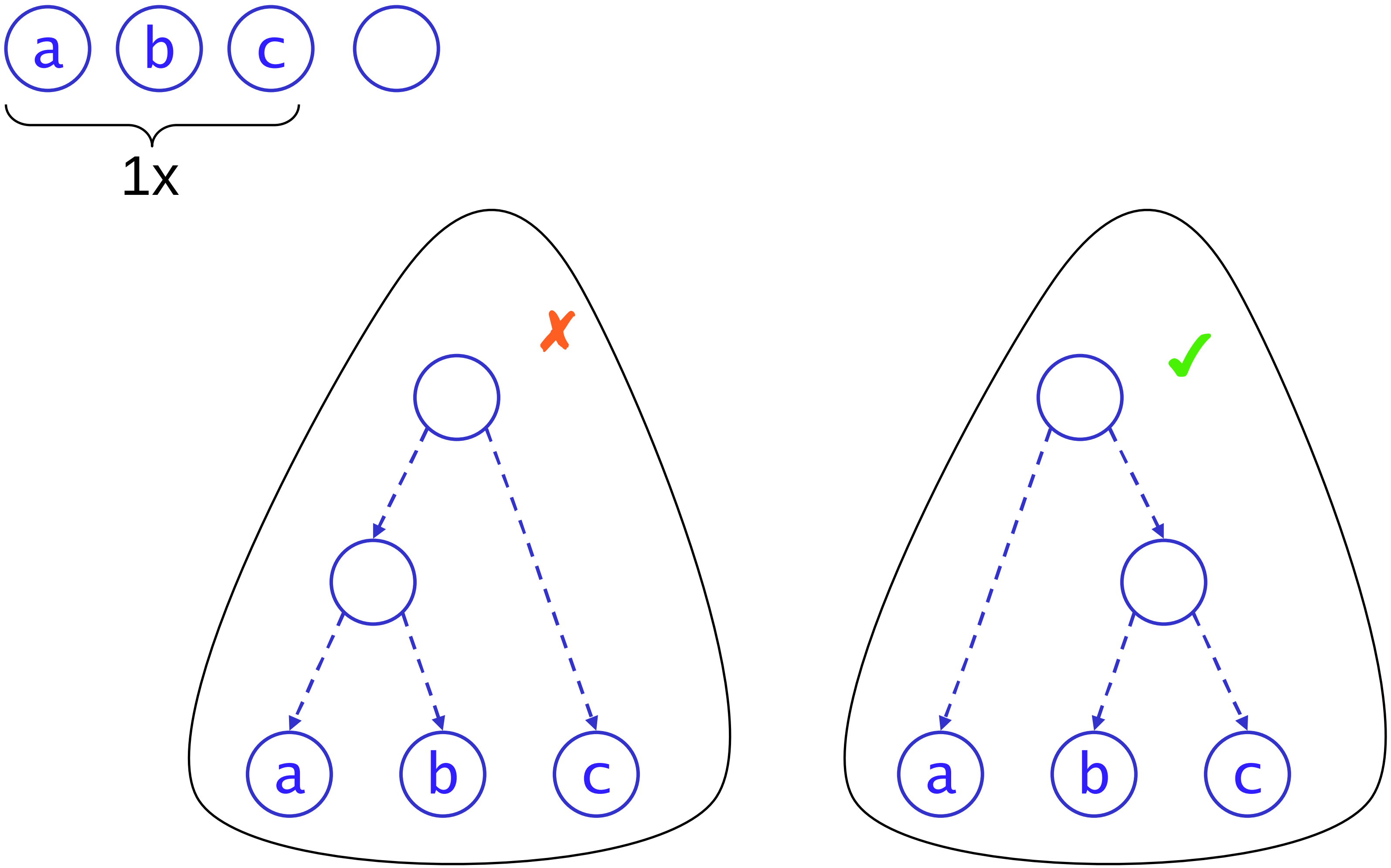
FO
first-order logic

TW
tree-walking automata

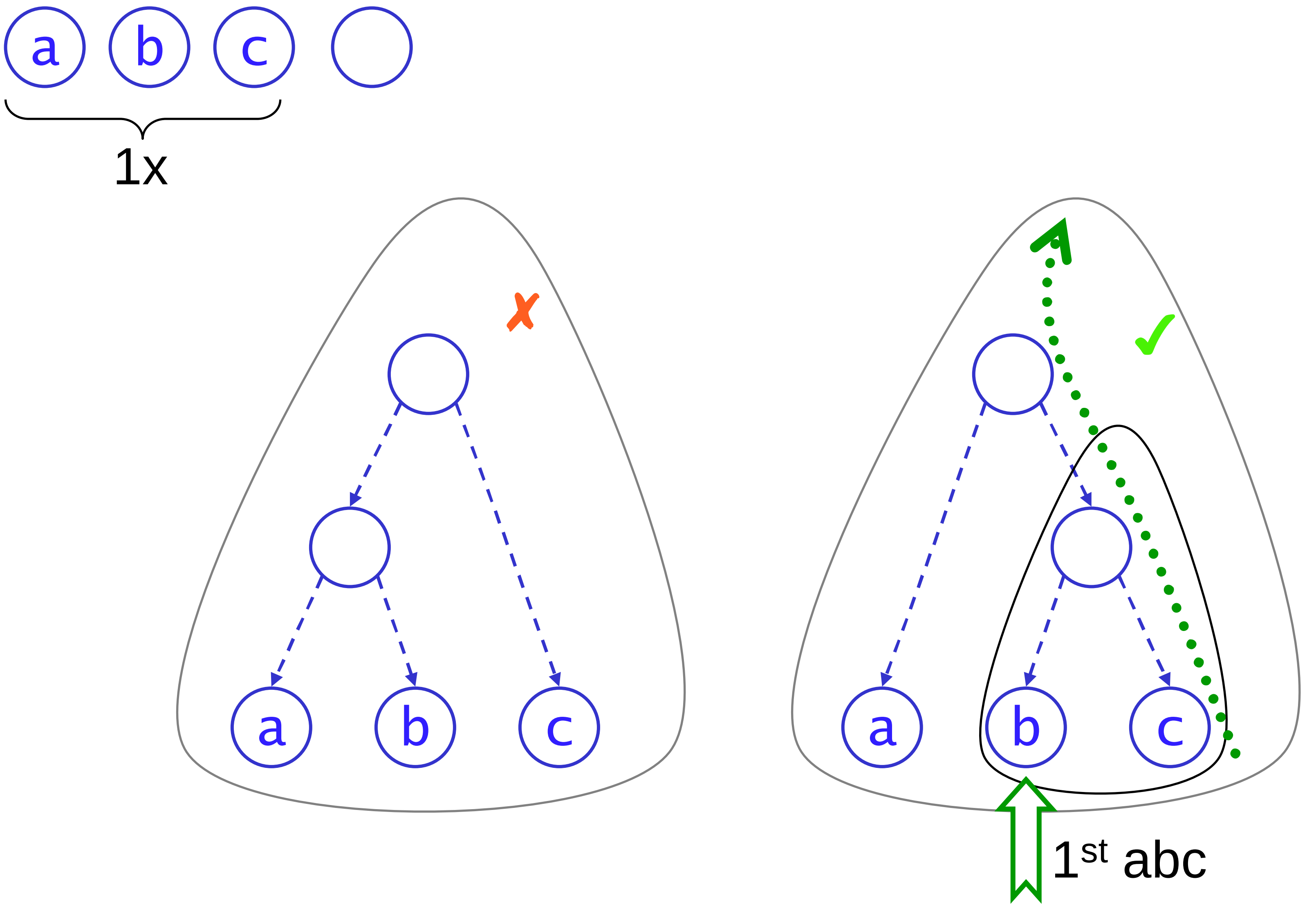
‘sequential’ (this talk)

‘a(bc)’ not by deterministic twa

Bojanczyk & Colcombet - ICALP 2004



‘a(bc)’ by *non*deterministic twa



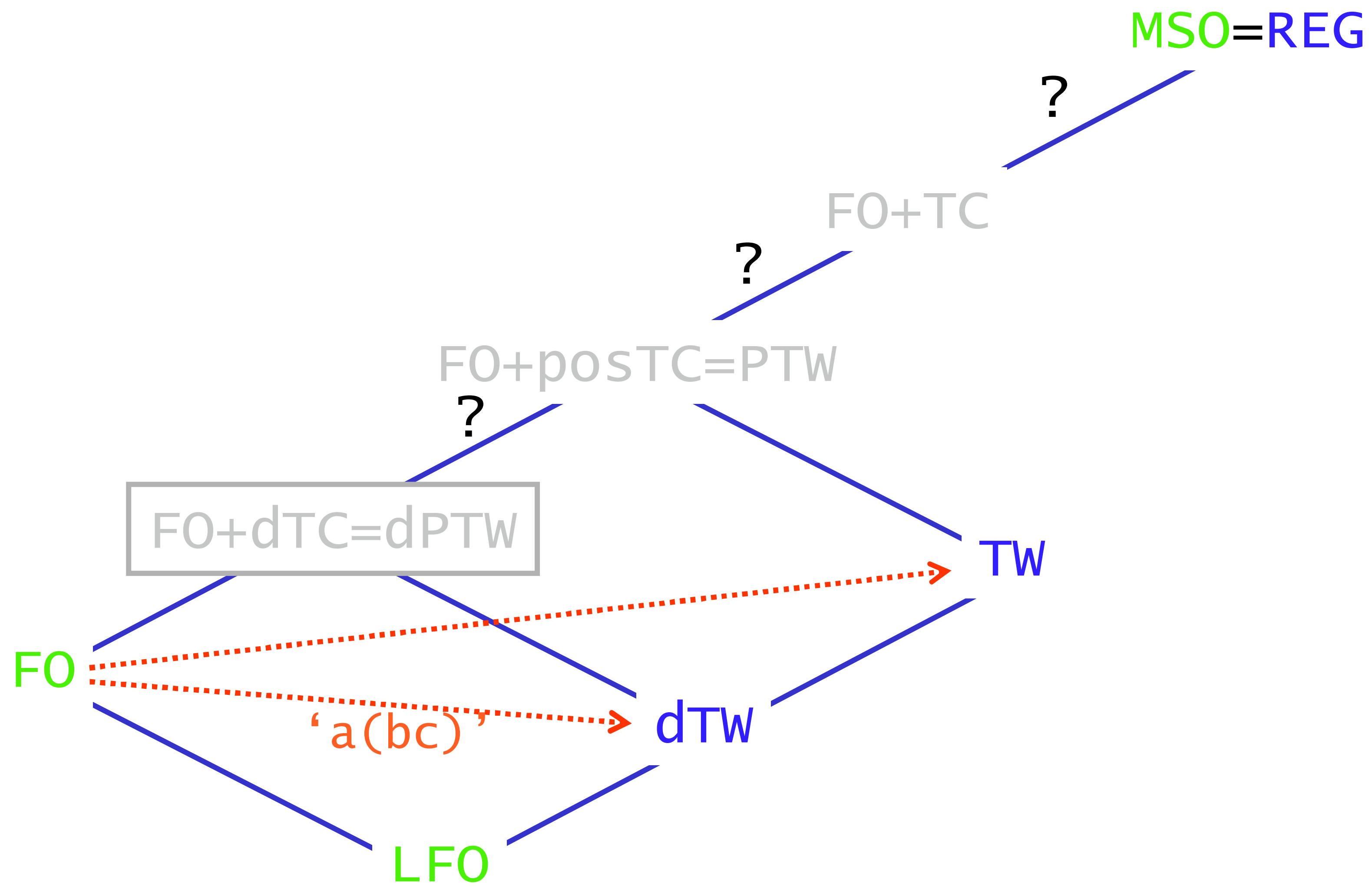
basic models

MSO
second-order logic $=$ REG
tree automata

Doner; Thatcher & Wright

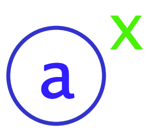
FO
first-order logic $\leftrightarrow$ TW
tree-walking automata

single head on trees



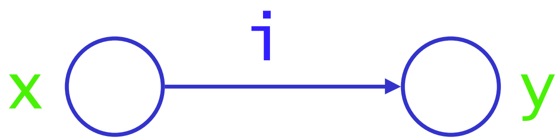
first-order logic for trees

variables x



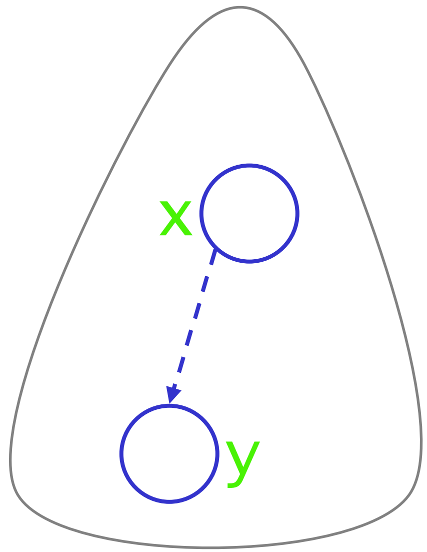
$lab_a(x)$

$edg_i(x,y)$



$x \leq y$

$x = y$

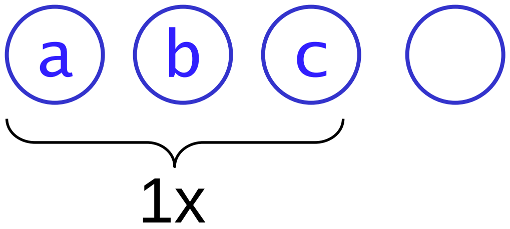


$\neg \quad \wedge \quad \vee \quad \forall x \quad \exists x$

$t(u_1, \ldots, u_n) \models \varphi(x_1, \ldots, x_n) \qquad t \models \varphi$

closed formula φ tree language $L(\varphi)$

'a(bc)' as FO tree language

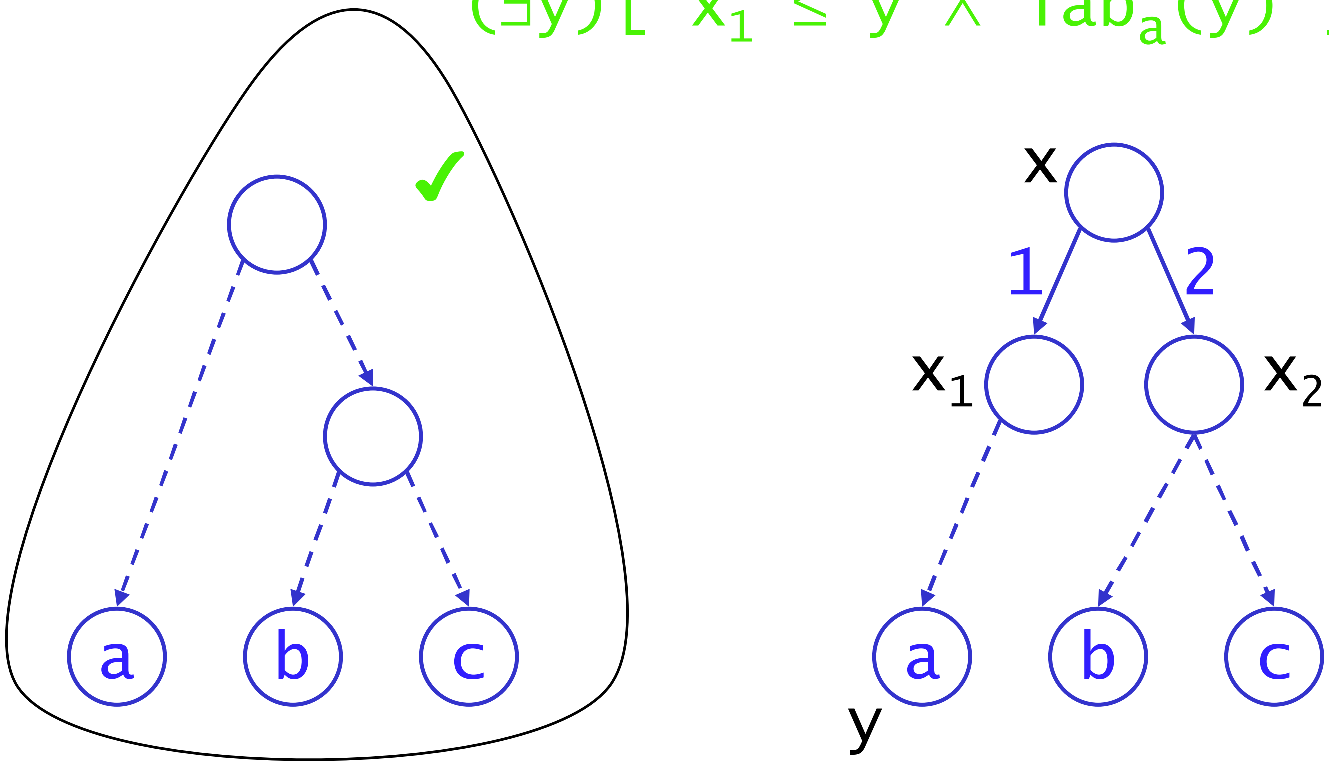


abc each once as leaf …

$(\exists x)(\exists x_1)(\exists x_2)$

$edg_1(x,x_1) \wedge edg_2(x,x_2) \wedge$

$(\exists y)[\ x_1 \leq y \wedge lab_a(y)\ ]$ …



extensions

MSO
second-order logic

REG
tree automata

FO+dTC
transitive closure

dPTW
nested pebbles

FO
first-order logic

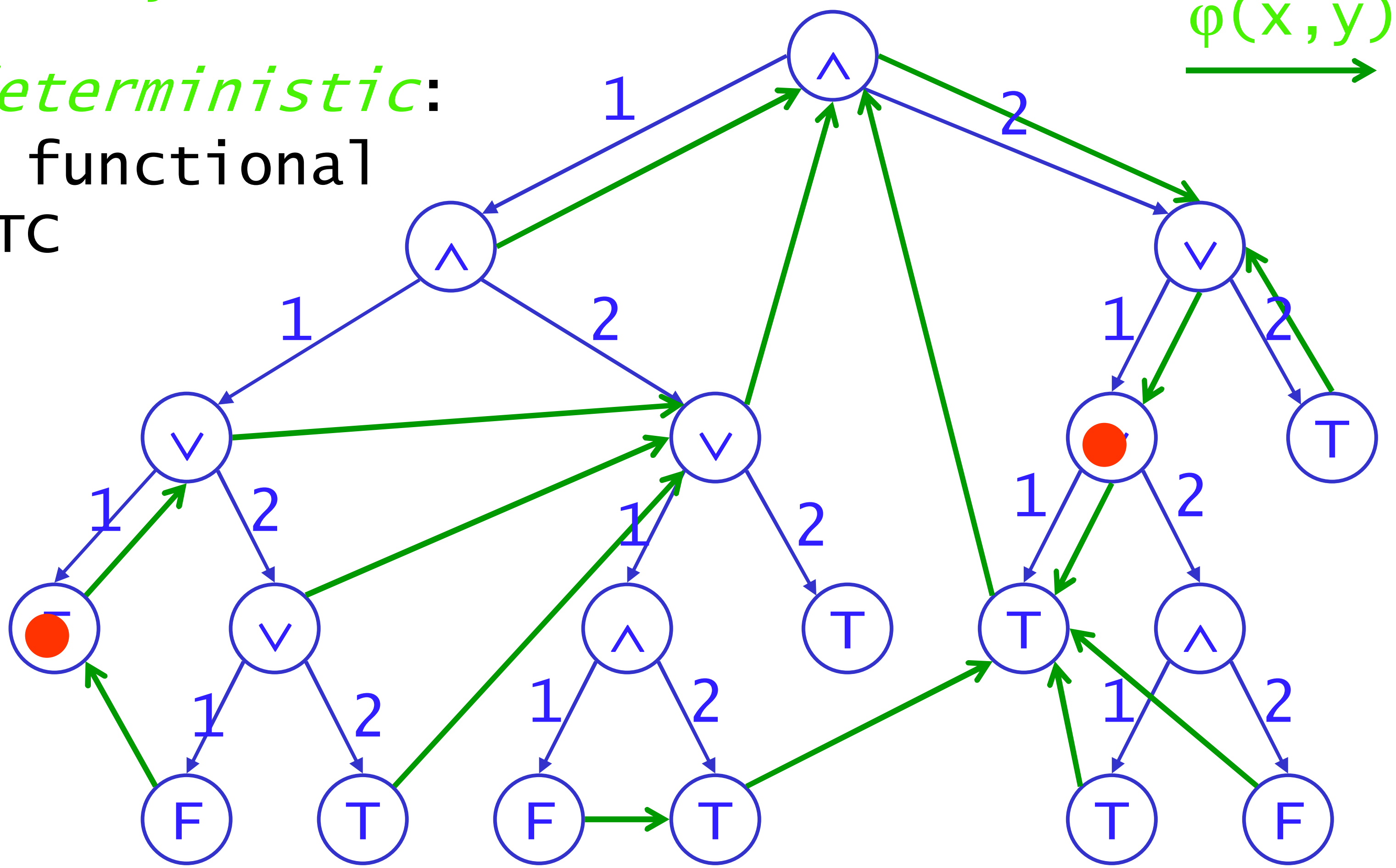
TW
tree-walking automata

determinism

(unary) transitive closure

φ*(x,y)

deterministic:
φ functional
dTC

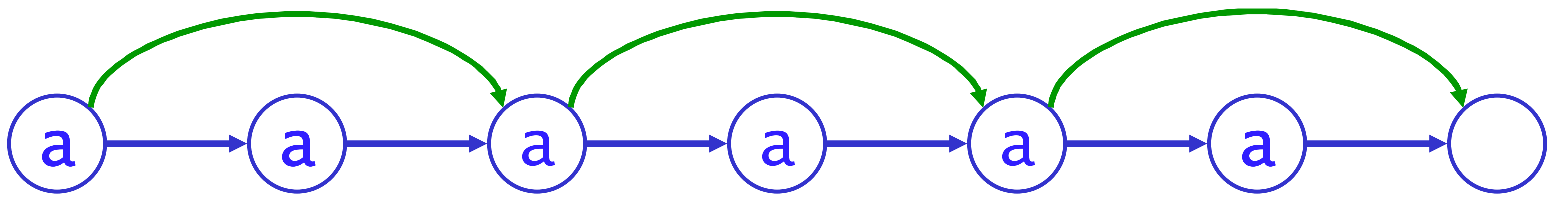


new

transitive closure

$\varphi^*(x,y)$

$\xrightarrow{\varphi(x,y)}$



(aa)*

extensions

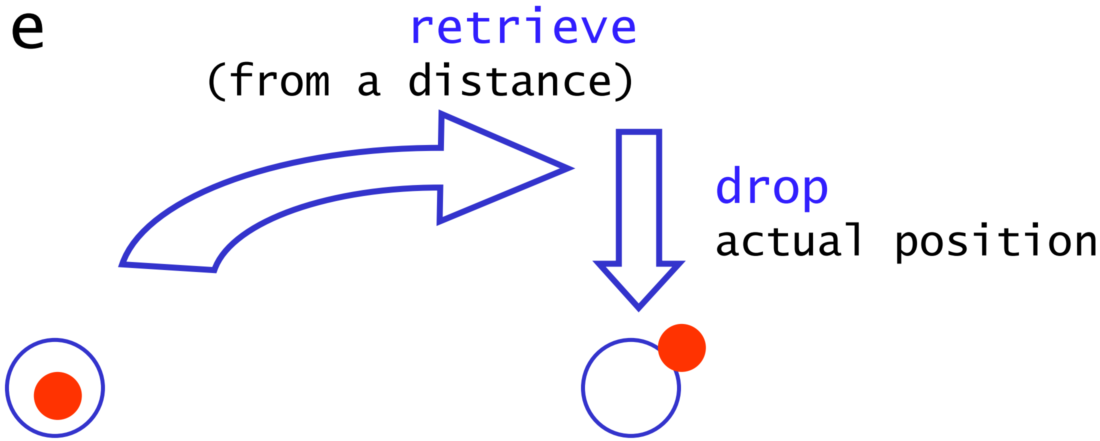
MSO	REG
second-order logic	tree automata
FO+dTC	dPTW
transitive closure	nested pebbles
FO	TW
first-order logic	tree-walking automata

determinism

nested pebbles

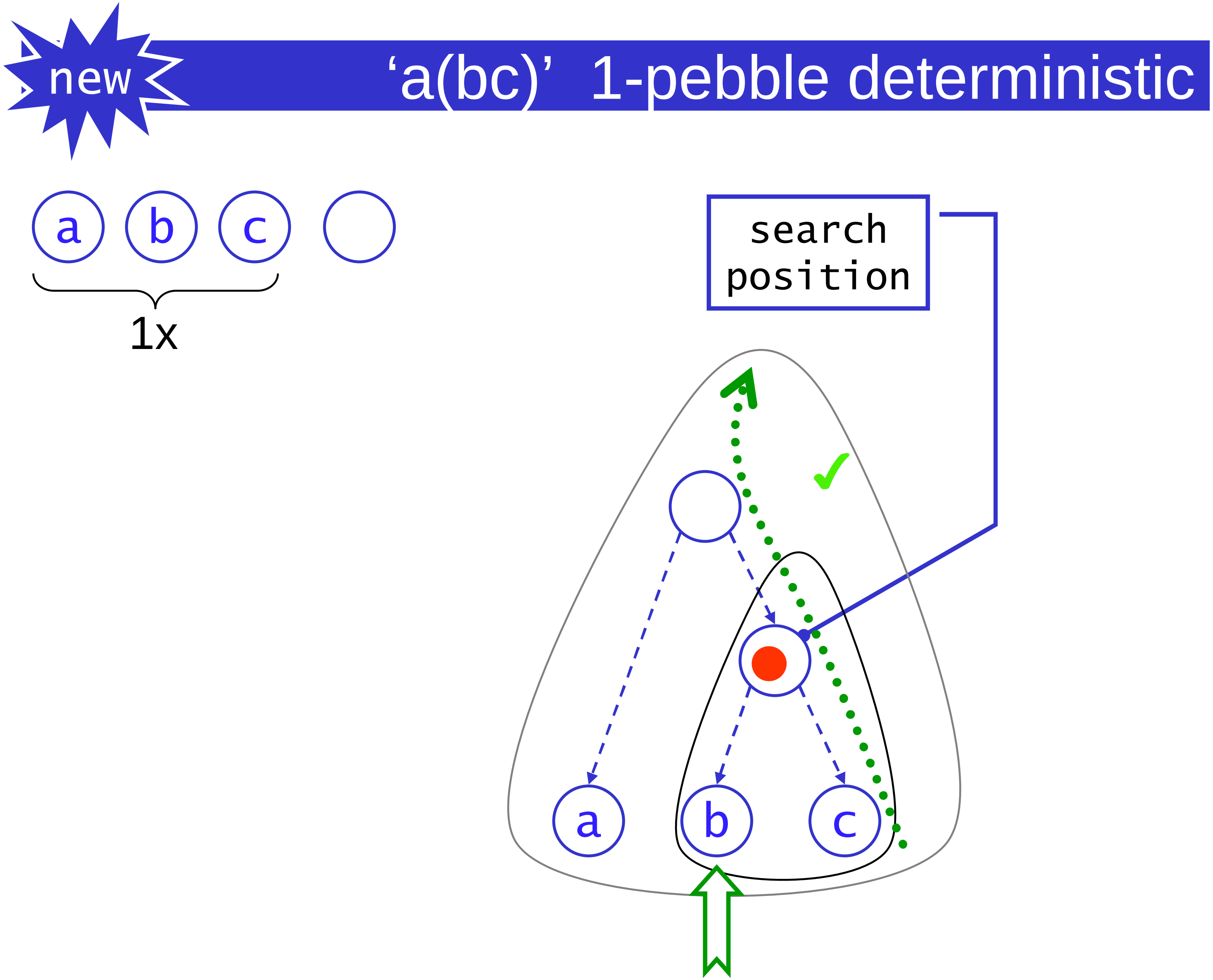
pebble: marks a node

- ‘abstract’ markers
  rather than ‘physical’
- *nested*: FIFO
- coloured
- remain visible

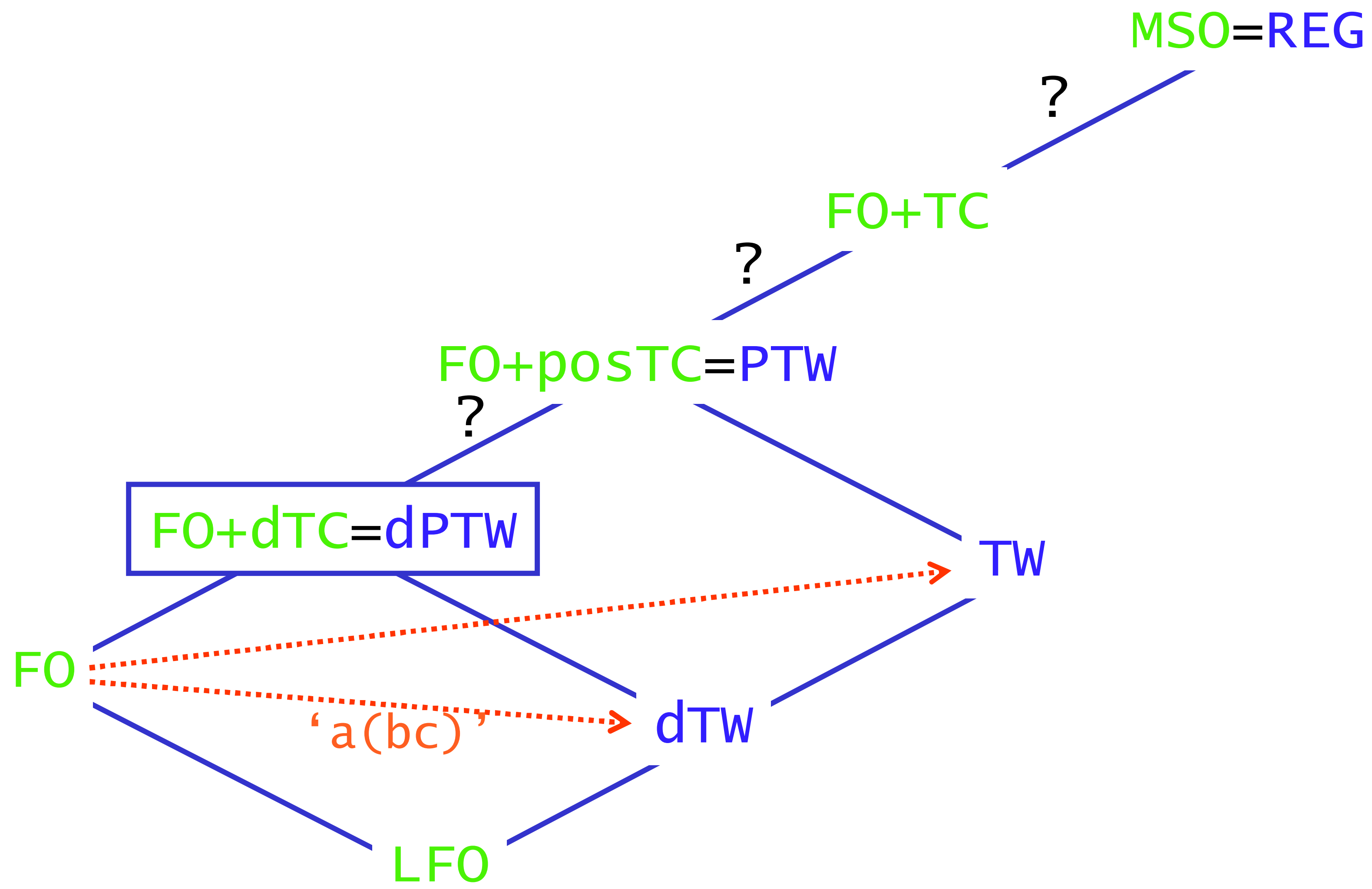


‘regular’ extension: $\boxed{PTW^{1} \subseteq REG}$

‘a(bc)’ 1-pebble deterministic

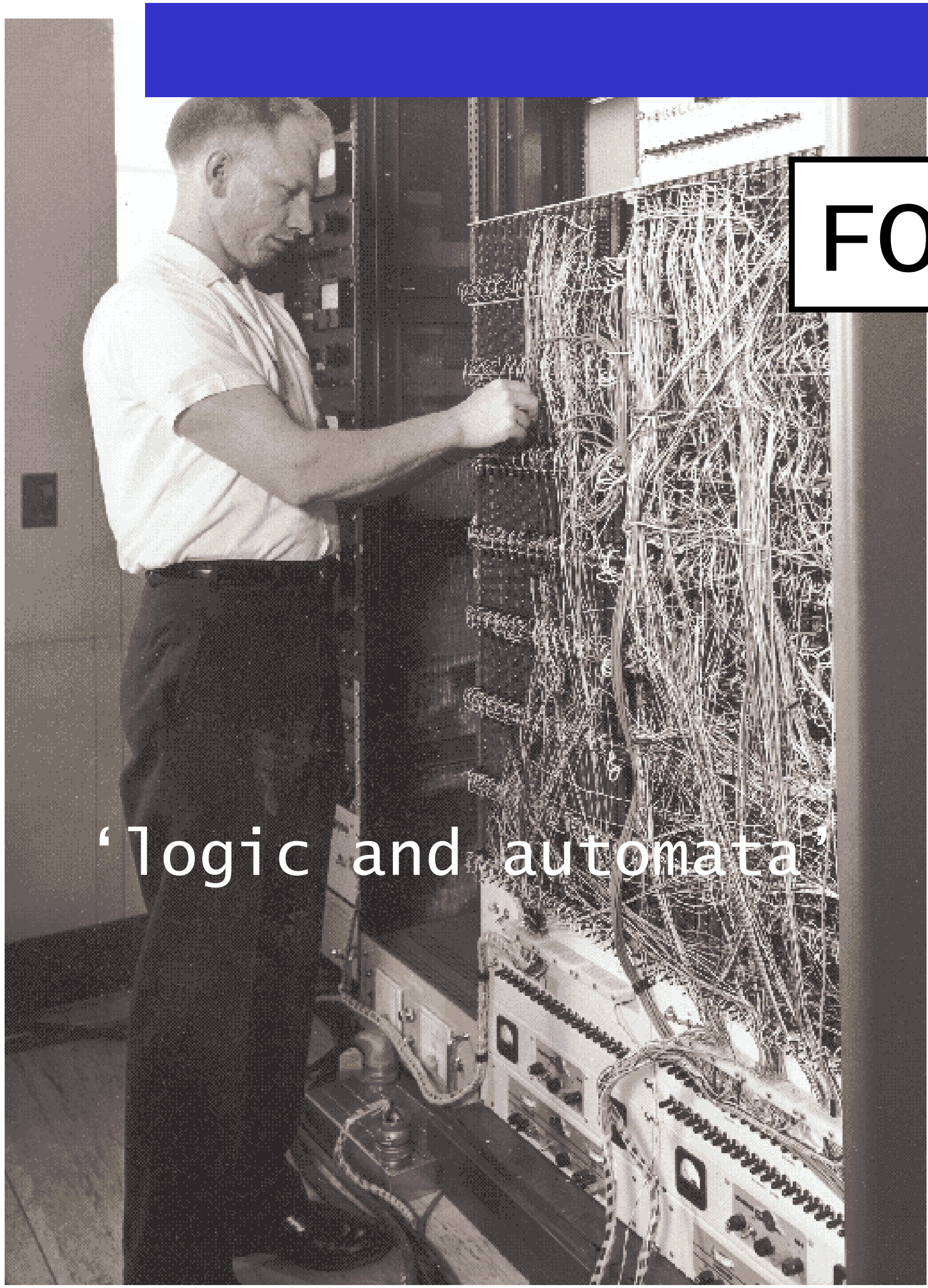


single head on trees



main result

$$\mathrm{FO}+\mathrm{dTC}^k = \mathrm{dPTW}^k$$



logic to nested pebbles

$\mathtt{lab}_a(x)$
$\mathtt{edg}_i(x,y)$
$x \leq y$
$x = y$

$\neg \quad \wedge \quad \vee$

$$\left.\begin{array}{l}\forall x \quad \exists x \\ \\ \varphi^*(x,y)\end{array}\right\}$$

$$\boxed{\mathrm{FO{+}dTC}^k \subseteq \mathrm{dPTW}^k}$$

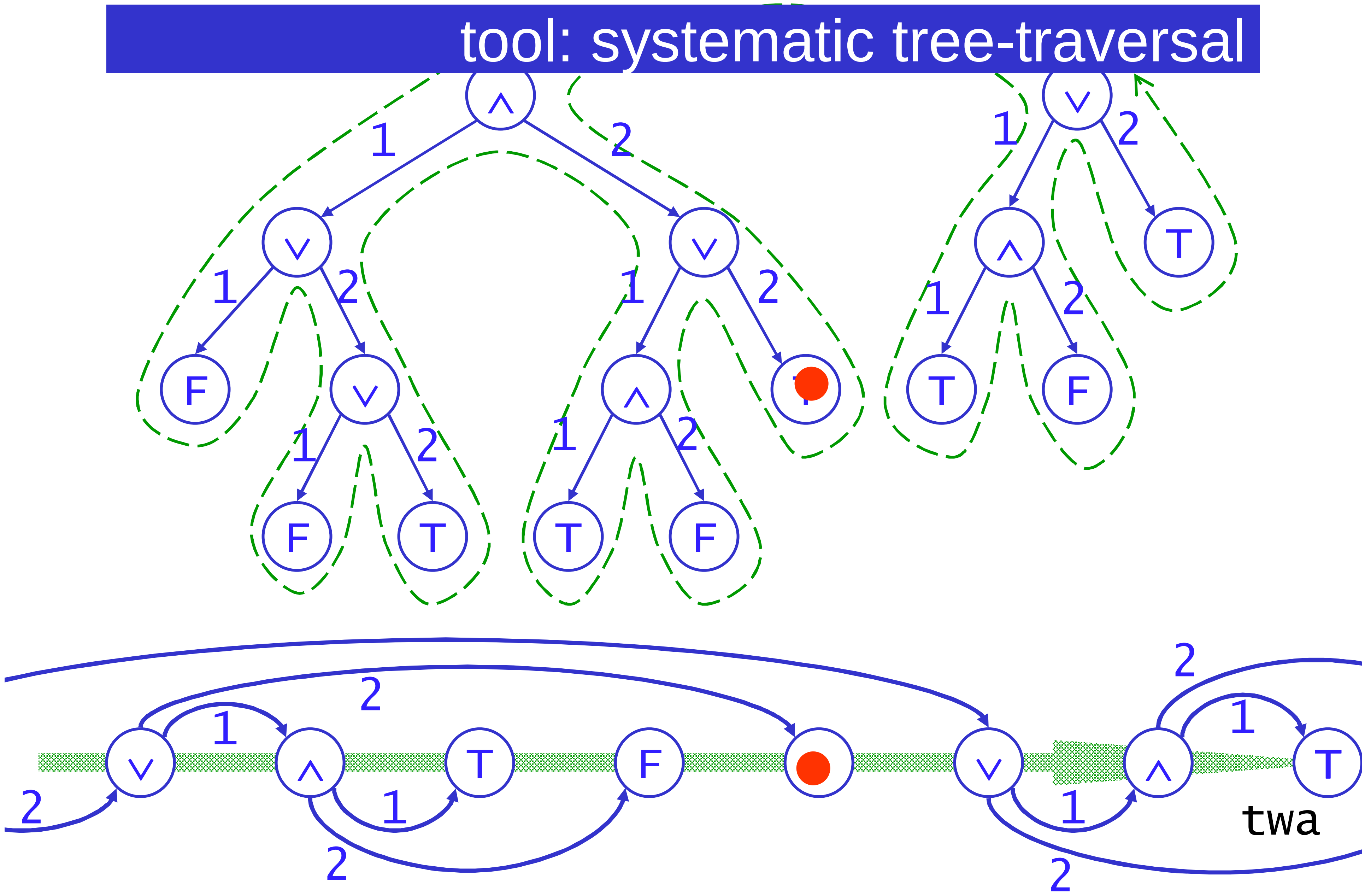
$$\varphi \rightarrow \mathcal{A}$$

by induction
always halting (!)

free variables ~
fixed pebbles

FO+dTC $\subseteq$ dPTW

tool: systematic tree-traversal



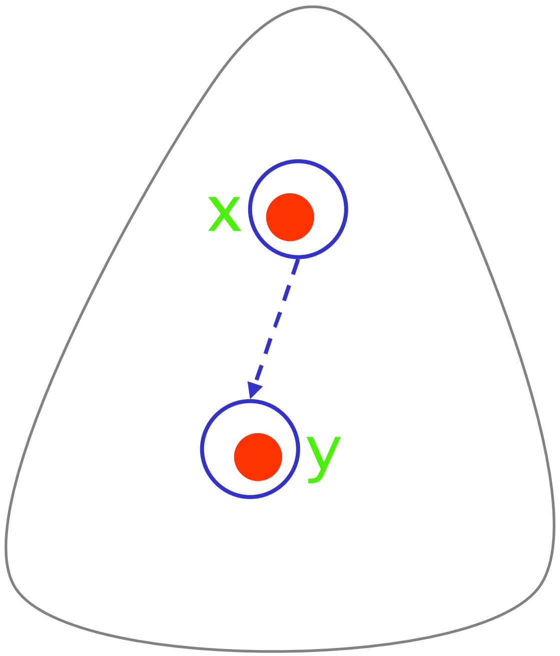
atomic formulas, connectives, ...

always halting
free variables ~
fixed pebbles

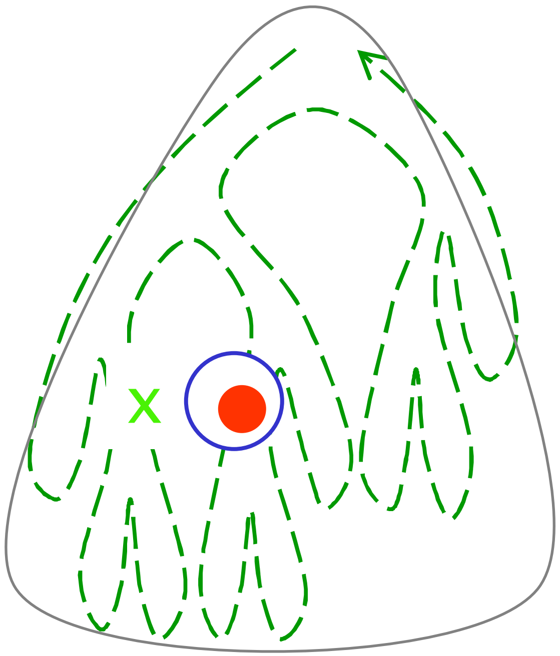
$lab_a(x)$
$edg_i(x,y)$
$x \leq y$
$x = y$

$\neg \;\; \wedge \;\; \vee$
$\forall x \;\; \exists x$

$\varphi^*(x,y)$



$x \leq y$



$\forall x \;\; \varphi(x)$

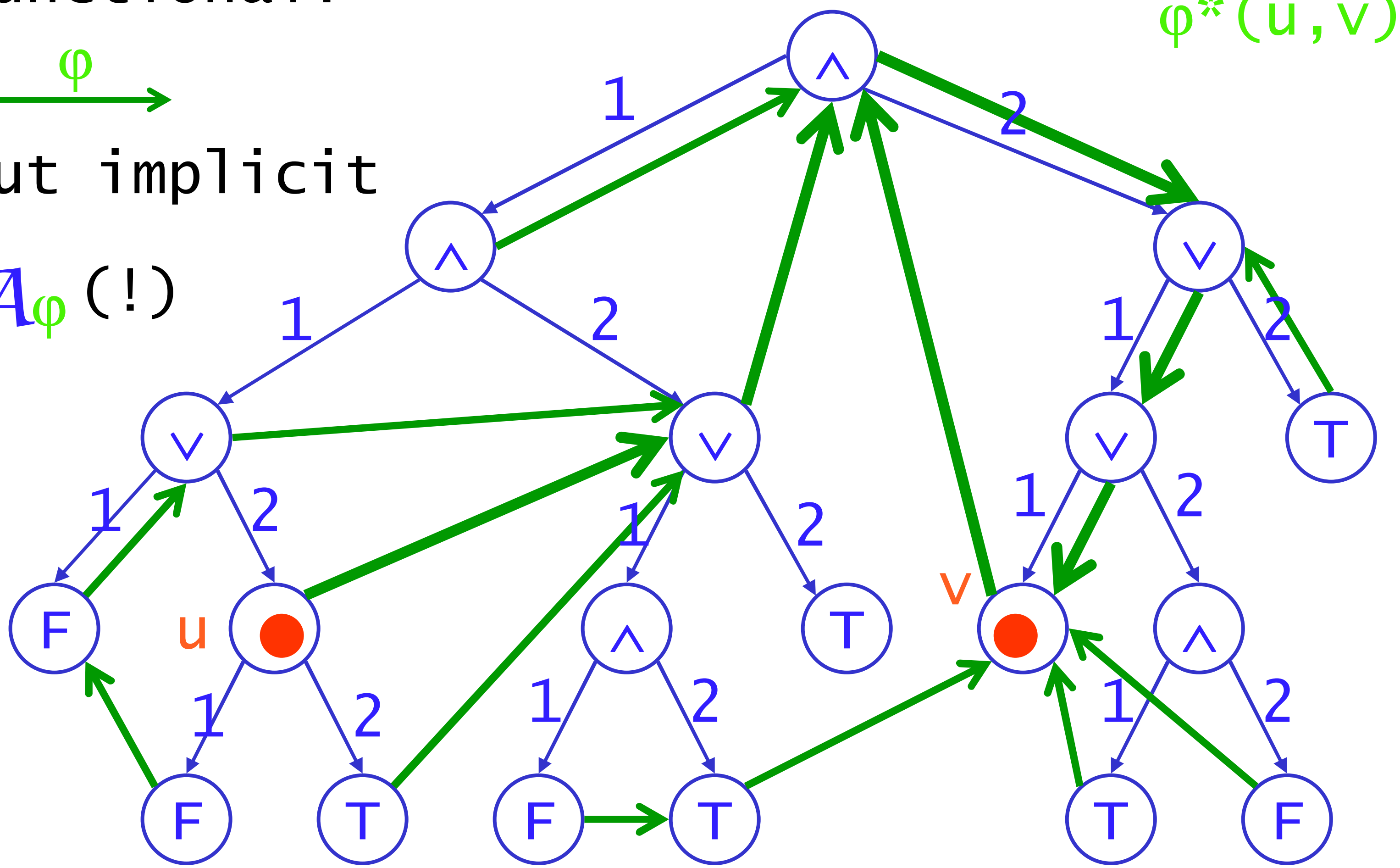
transitive closure

functional!

$\xrightarrow{\varphi}$

but implicit

$\mathcal{A}_\varphi$ (!)

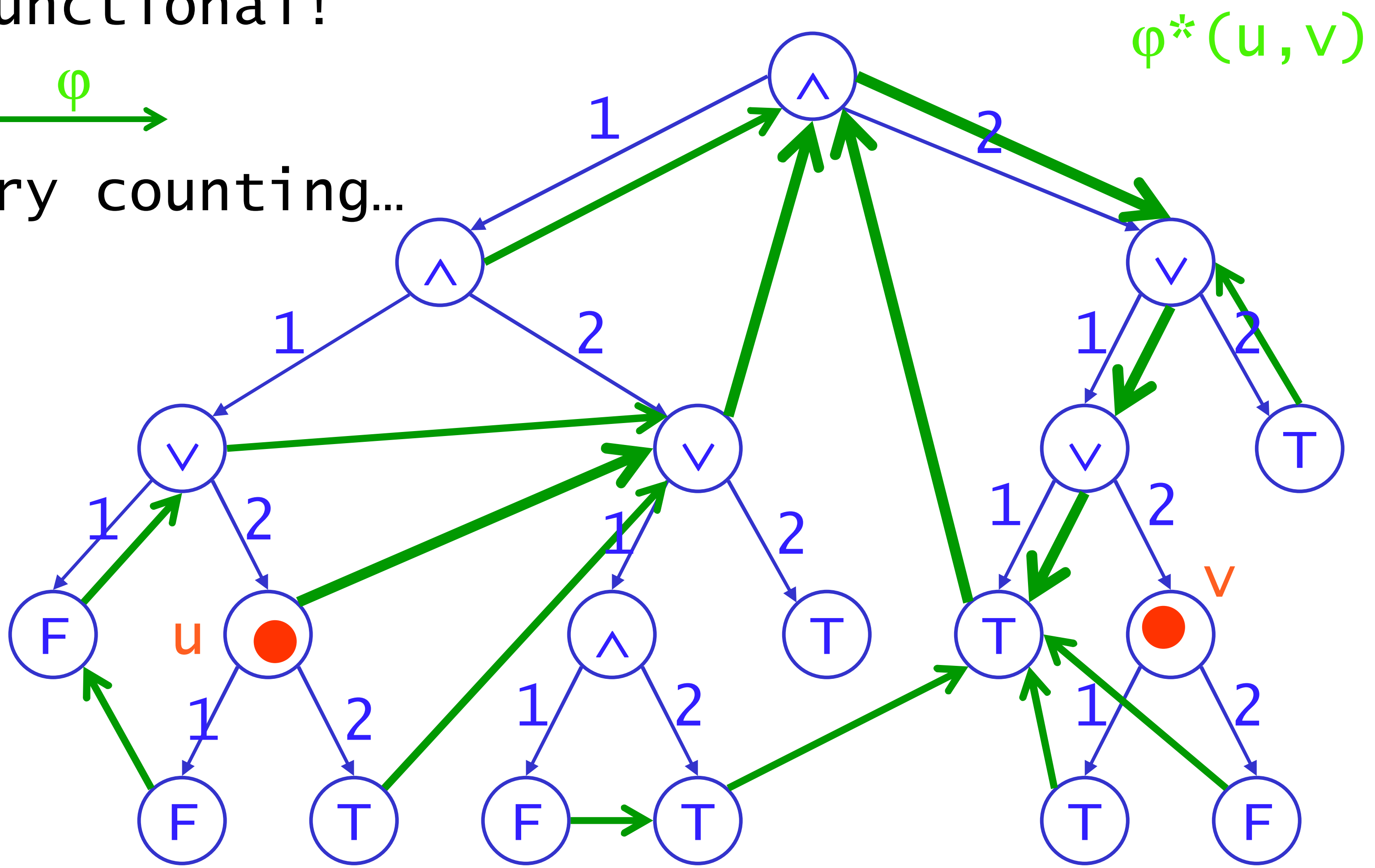


transitive closure

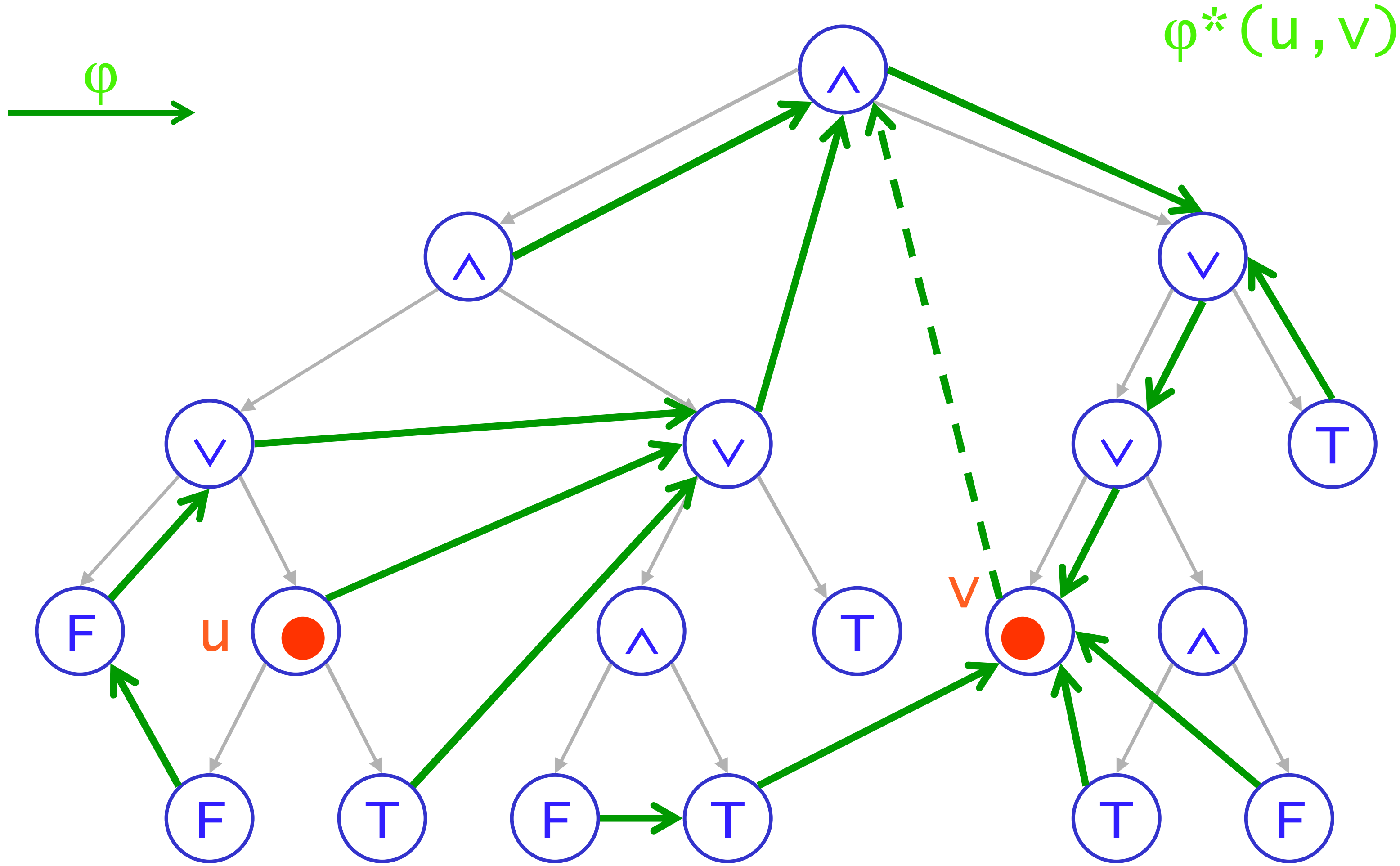
functional!

$\xrightarrow{\varphi}$

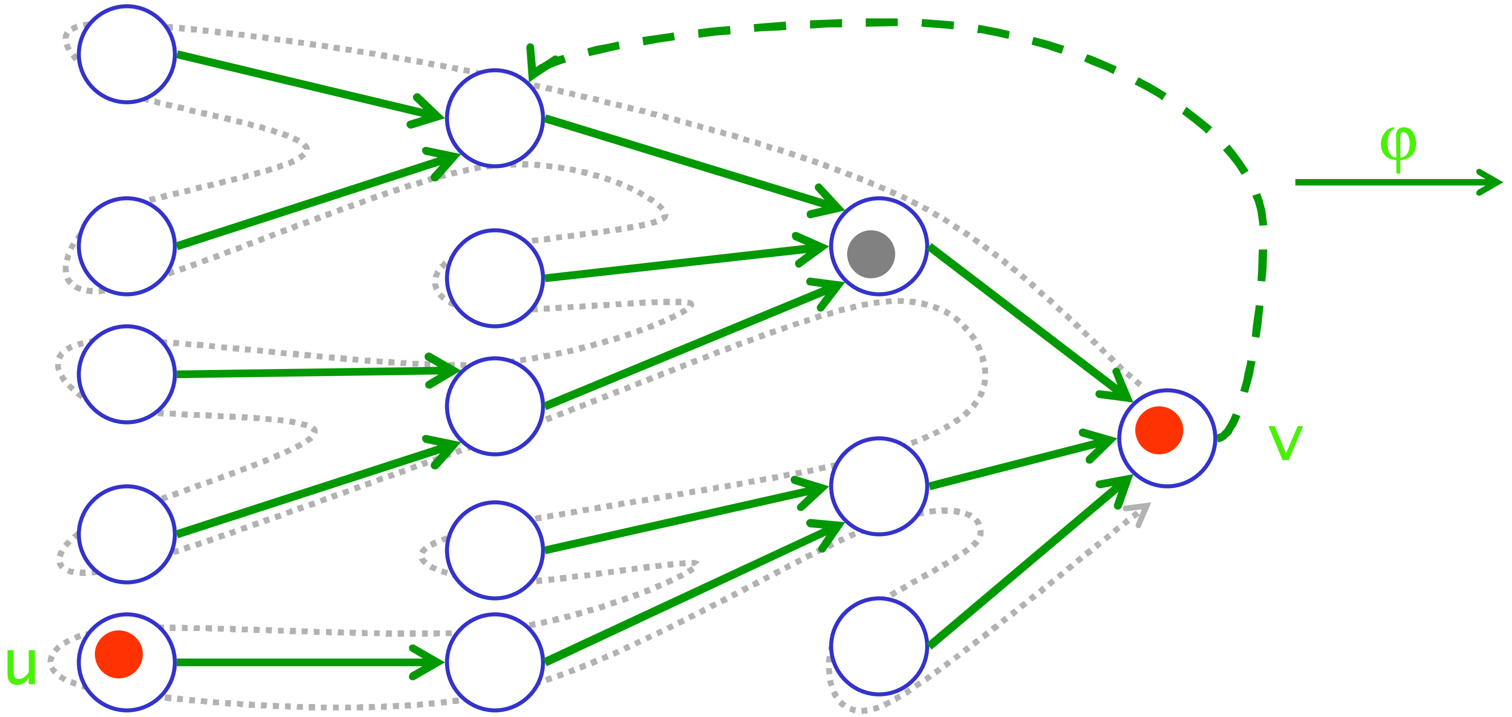
try counting…

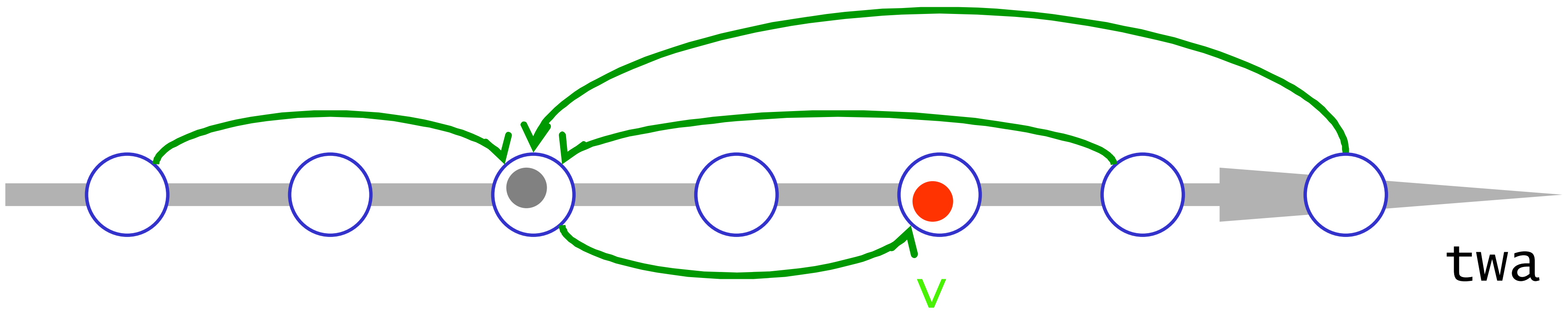


φ-tree



walking the φ-tree backwards





twa

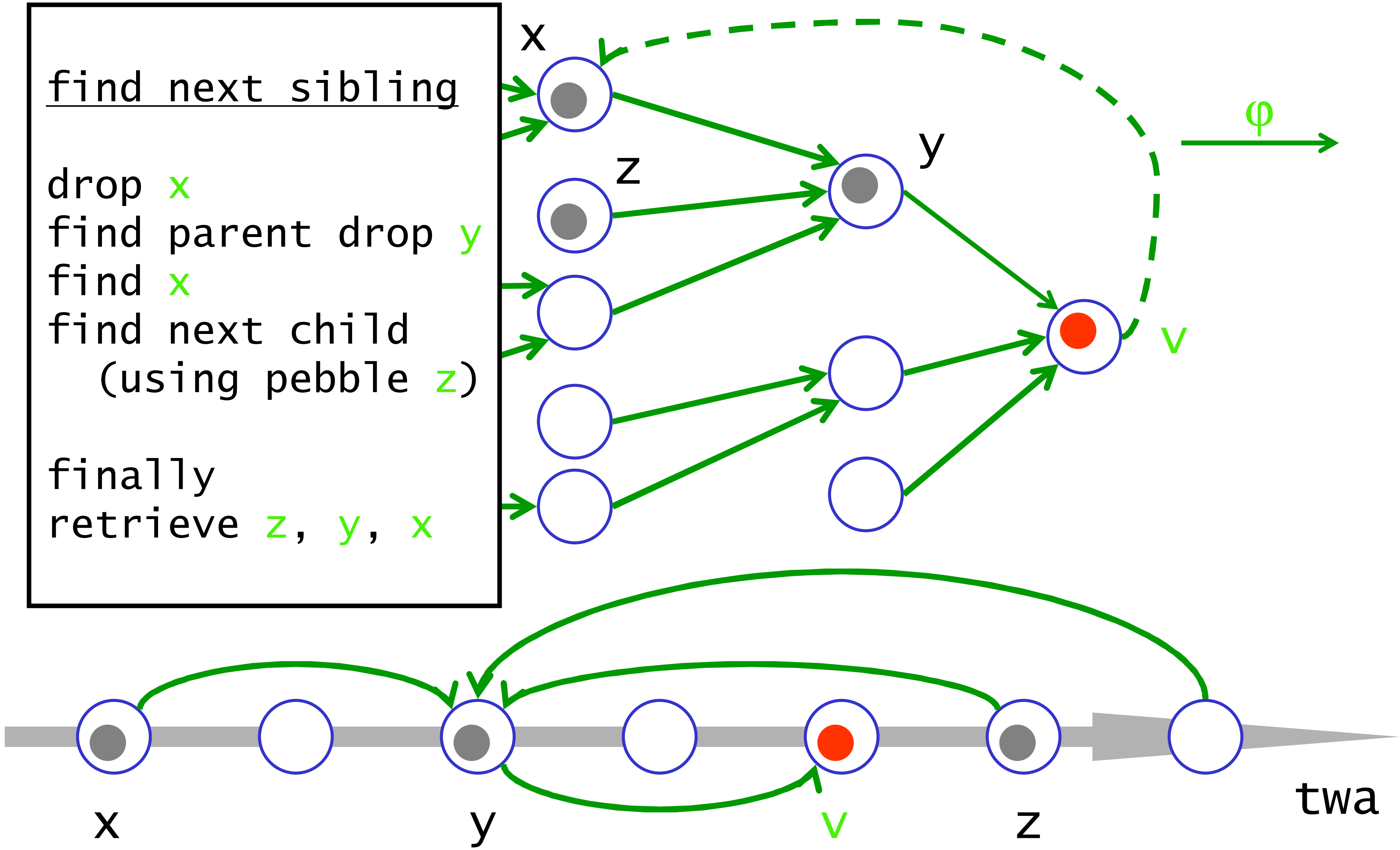
… but implicit

walking the φ-tree backwards

<u>find next sibling</u>

drop x
find parent drop y
find x
find next child
(using pebble z)

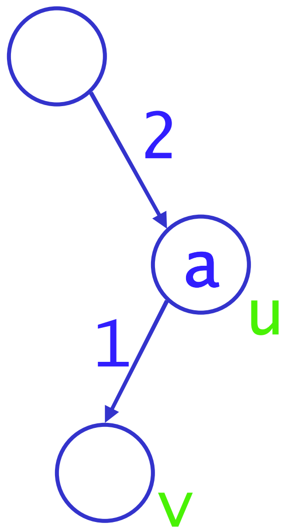
finally
retrieve z, y, x



nested pebbles to logic

$$\mathrm{dPTW}^k \subseteq \mathrm{FO{+}dTC}^k$$

<u>single move</u> from state p to q:
in node u with label a,
second child, not pebble x_3,
then move to first child v



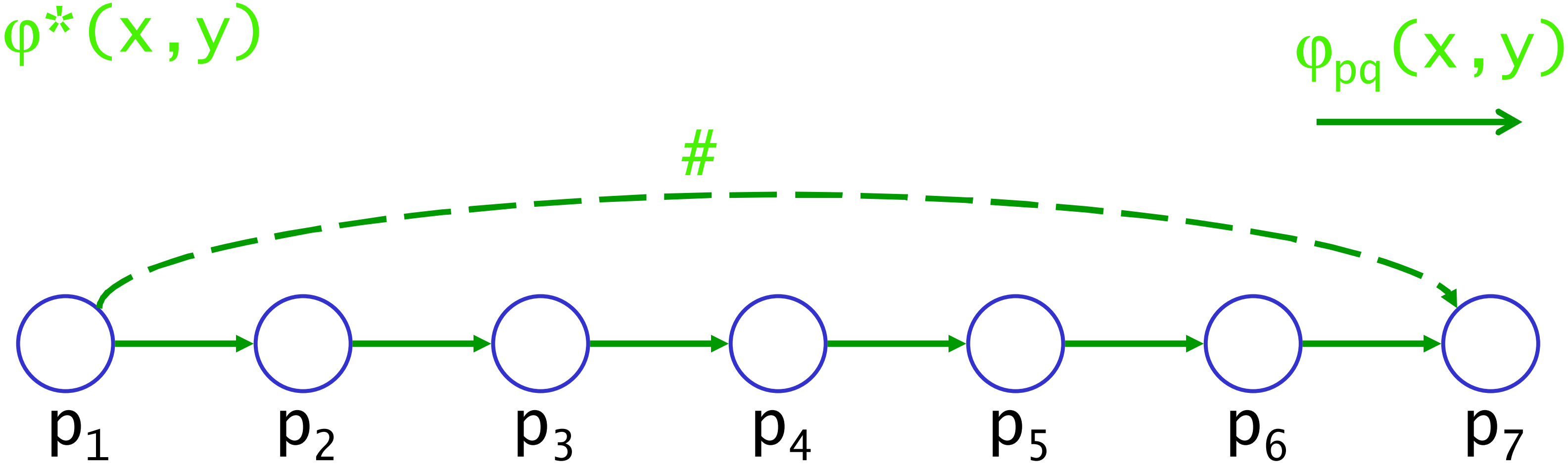
$$\varphi_{pq}(u,v) = \mathrm{lab}_a(u) \wedge \exists u'\,\mathrm{edg}_2(u',u) \wedge u \neq x_3 \wedge \mathrm{edg}_1(u,v)$$

free variables for pebbles

- computation ~ TC with states
- induction on pebbles

transitive closure 'with states'

$\varphi^*(x,y)$



$$\Phi = \begin{pmatrix} \varphi_{pq}(x,y) \end{pmatrix} \qquad \Phi^{\#} = \begin{pmatrix} \varphi^{\#}_{pq}(x,y) \end{pmatrix}$$

single steps

deterministic
(functional & exclusive)

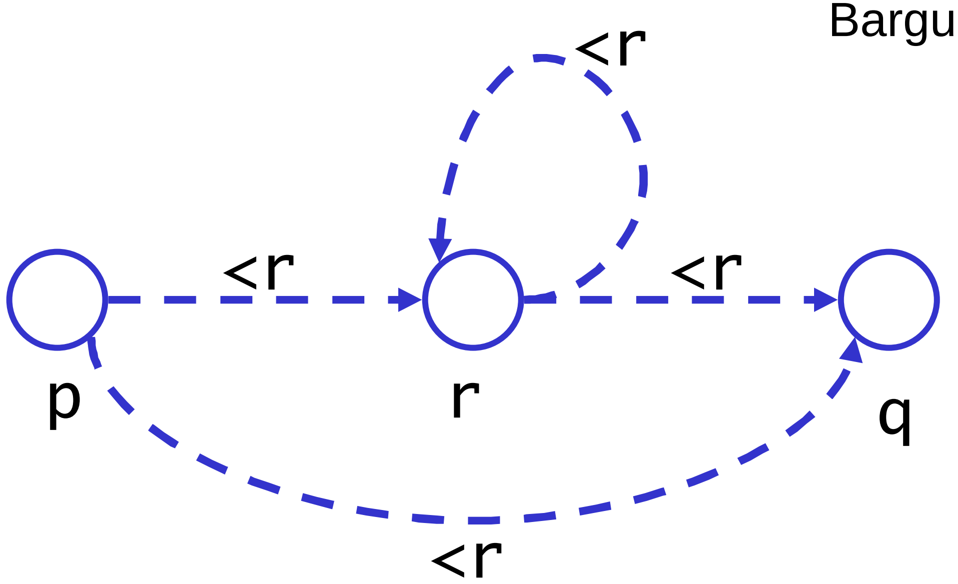
computation closure

deterministic
wrt final states

Kleene

Bargury & Makowsky

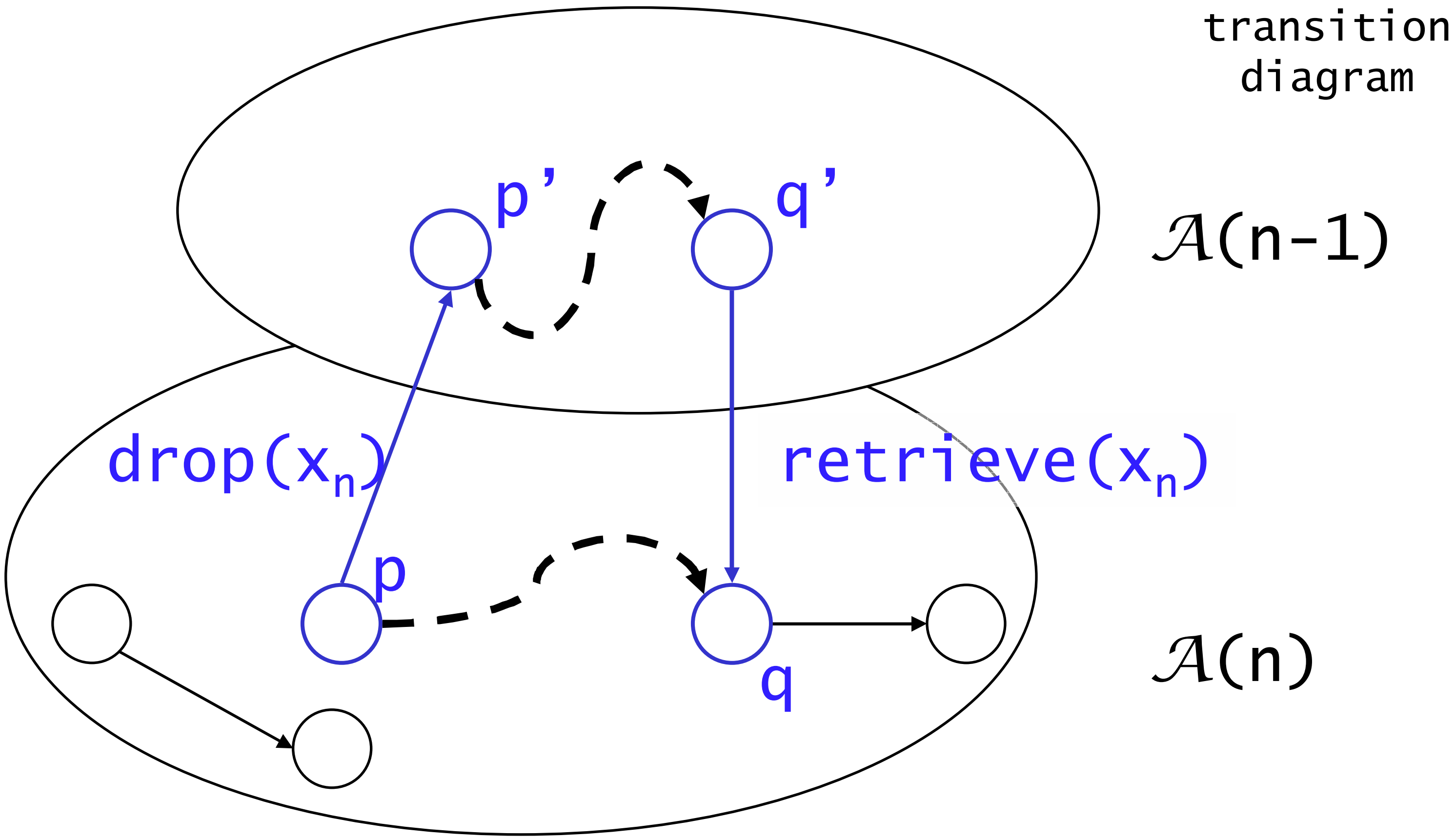
$\boxed{\leq r}$



$$R^{r}_{pq} = R^{r-1}_{pq} \cup R^{r-1}_{pr} \cdot R^{r-1\,*}_{rr} \cdot R^{r-1}_{rq}$$

$$\varphi^{r}_{pr}(x,y) = \varphi^{r-1}_{pr}(x,y) \vee$$
$$(\exists x'y')\ \varphi^{r-1}_{pr}(x,x') \wedge (\varphi^{r-1}_{rr})^{*}(x',y') \wedge \varphi^{r-1}_{rq}(y',y)$$

getting rid of the pebbles



$$\varphi^{n}_{pq}(u,v) = \varphi^{(n-1)\#}_{p'q'}(u,v)$$

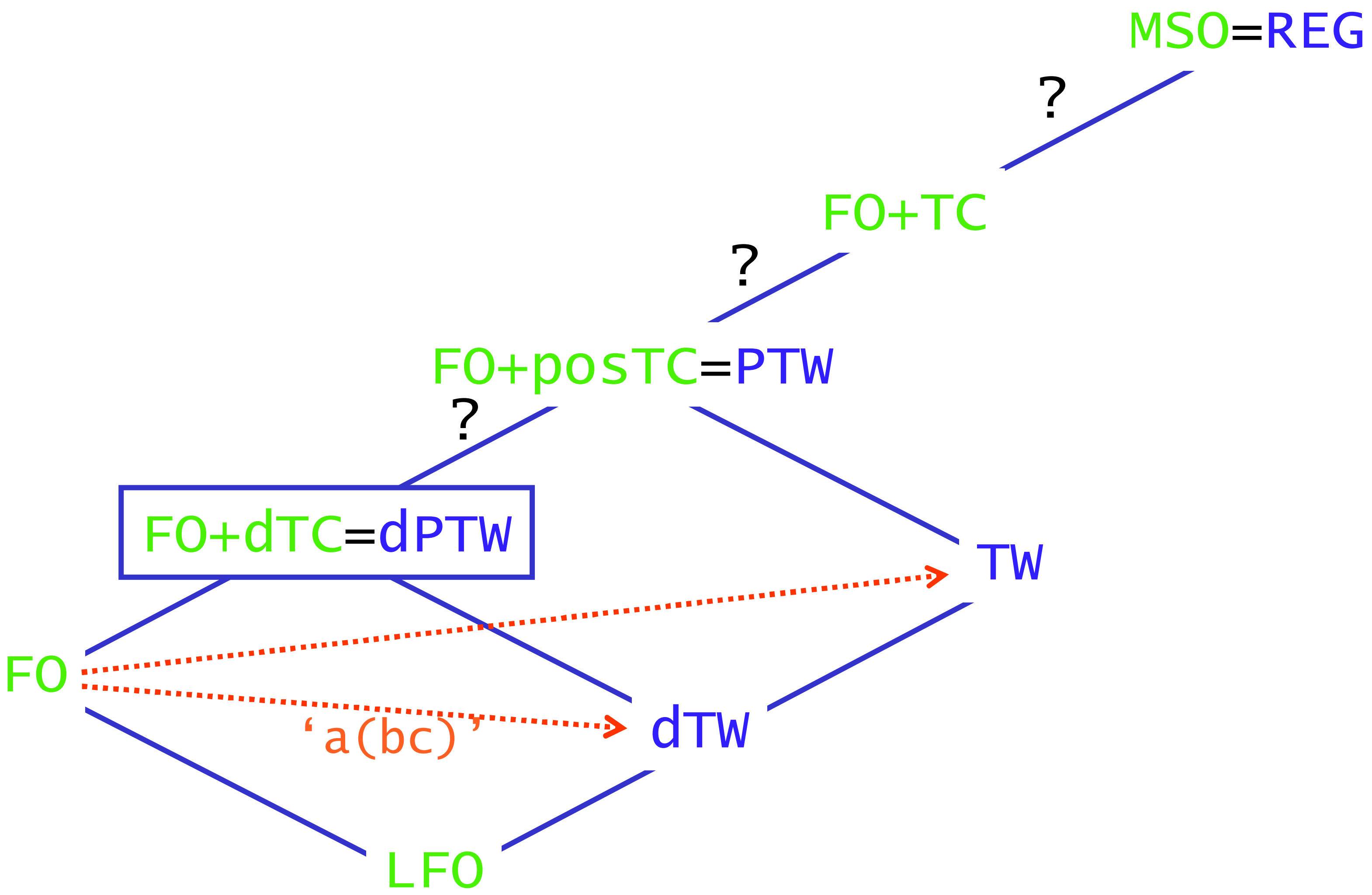
replacing x_n by u

results

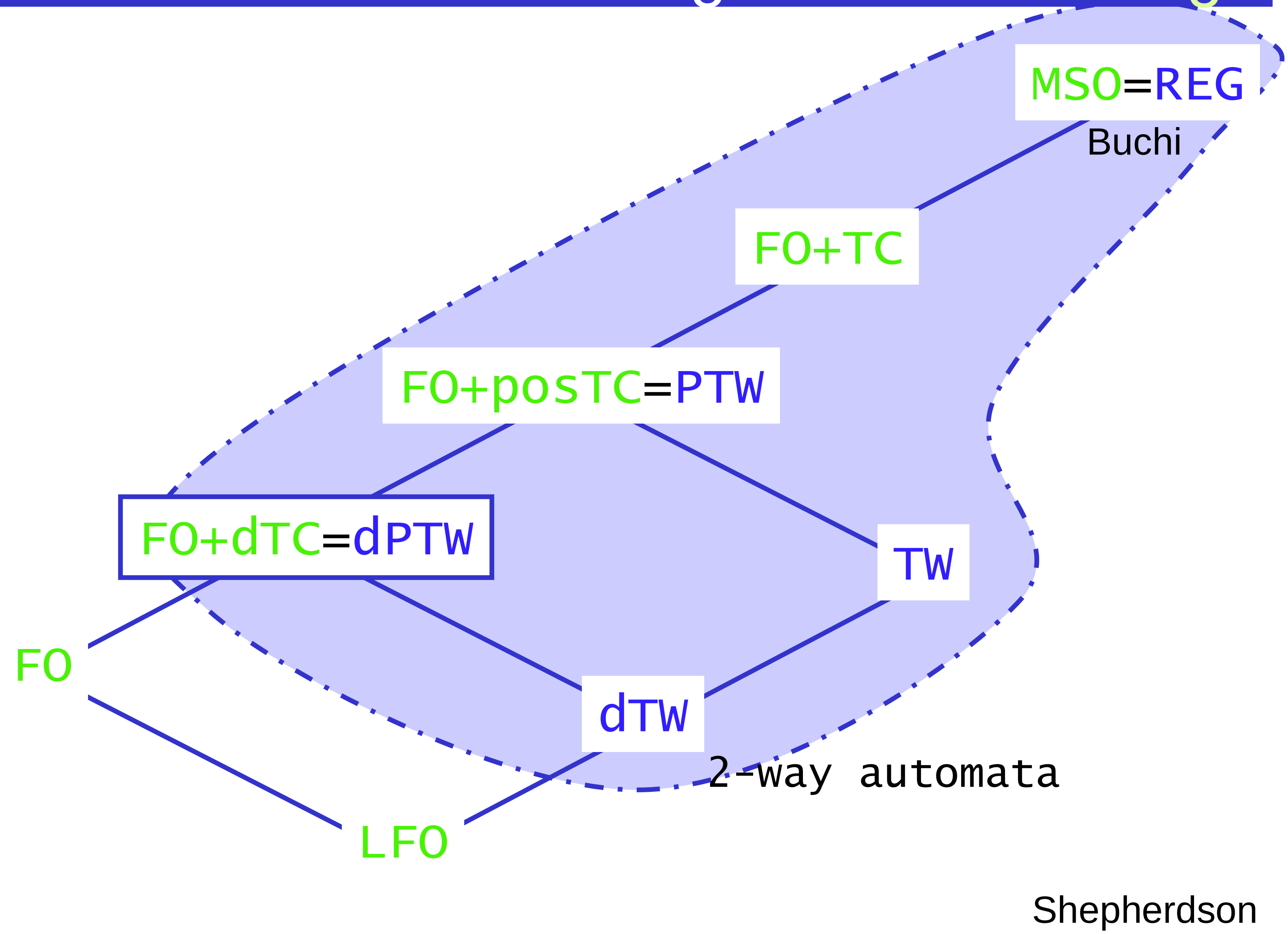
MSO second-order logic	=	REG tree automata
FO+posTC	=	PTW
FO+dTC transitive closure	=	dPTW nested pebbles
FO first-order logic		TW tree-walking automata

determinism

single head on trees

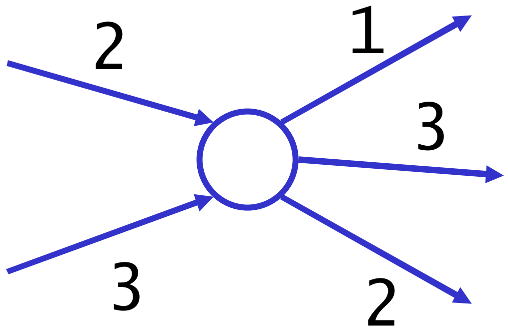


single head on strings

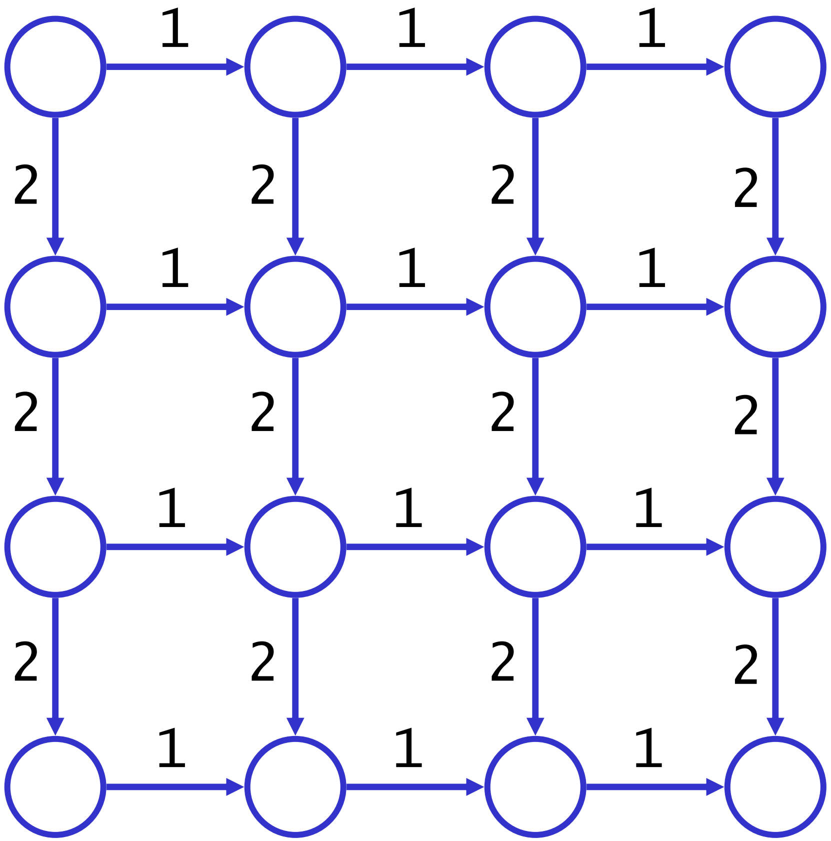


from trees to graphs

graphs



locally injective



grid, torus

nested pebbles to logic

$\mathtt{lab}_a(x)$
$\mathtt{edg}_i(x,y)$
$\cancel{x \le y}$
$x = y$

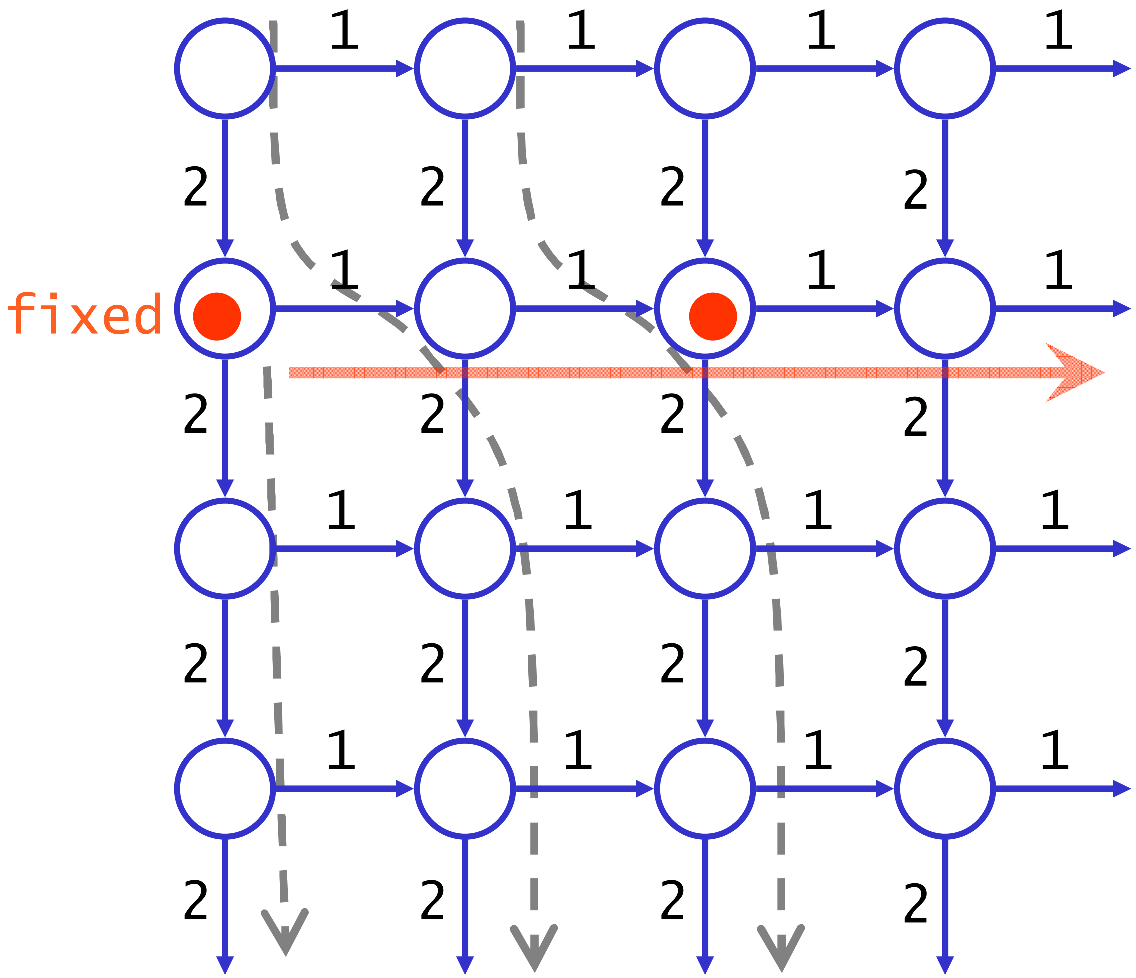
$\neg \quad \wedge \quad \vee$
$\forall x \quad \exists x$

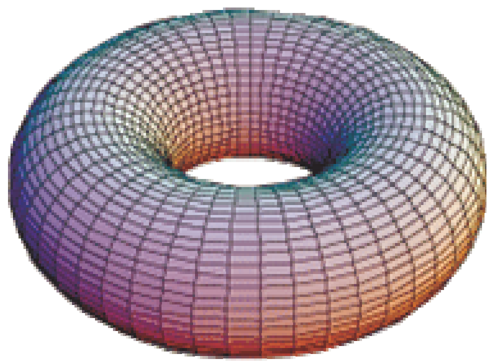
$\varphi^*(x,y)$

$$\boxed{\mathrm{dPTW}^k \subseteq \mathrm{FO{+}dTC}^k}$$

for families of graphs

walking the torus





two pebbles

(nested)

graphs with a guide

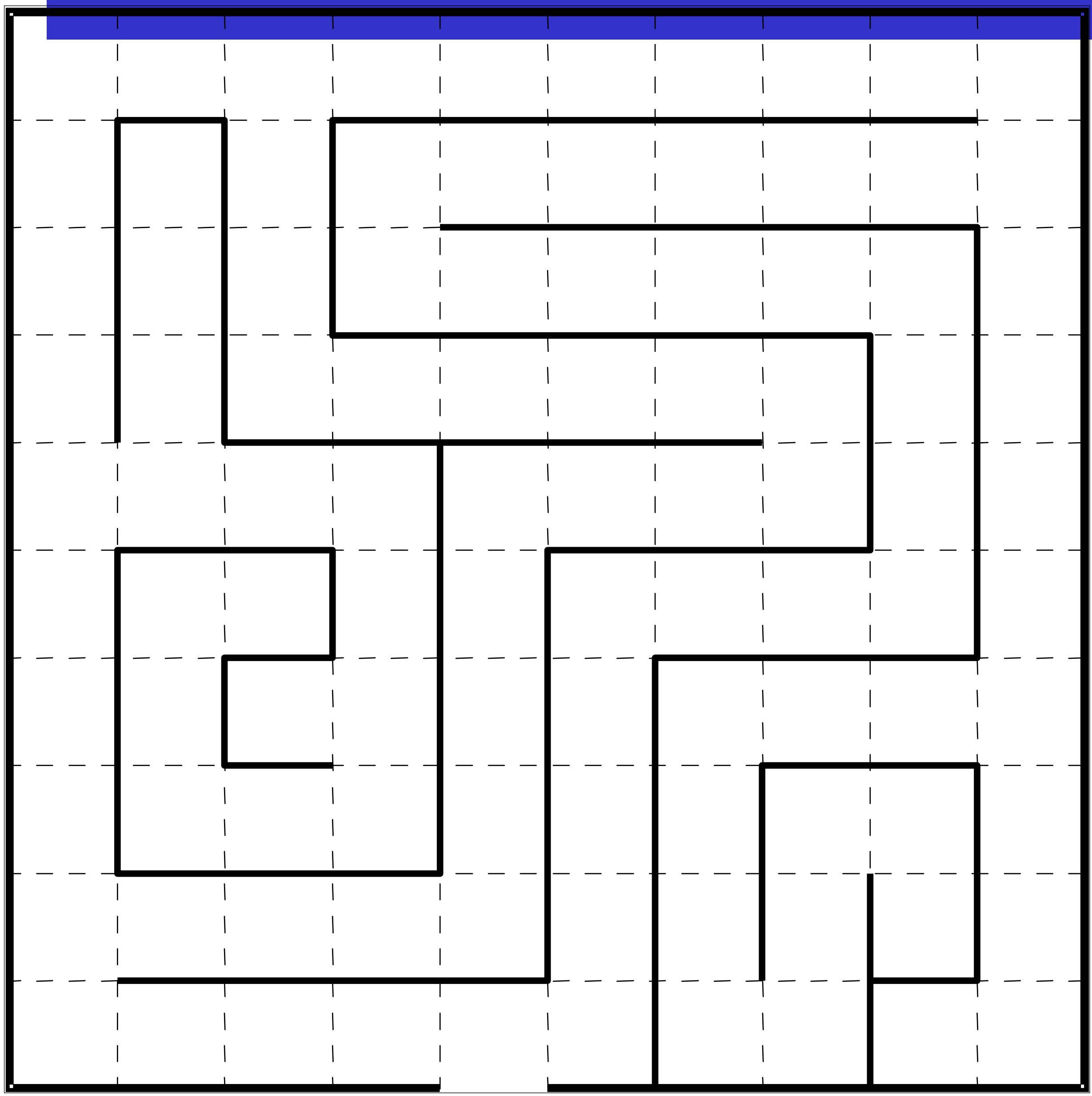
$$\boxed{FO{+}dTC^k = dPTW^k}$$

for families of *searchable* graphs
with a 'guide'

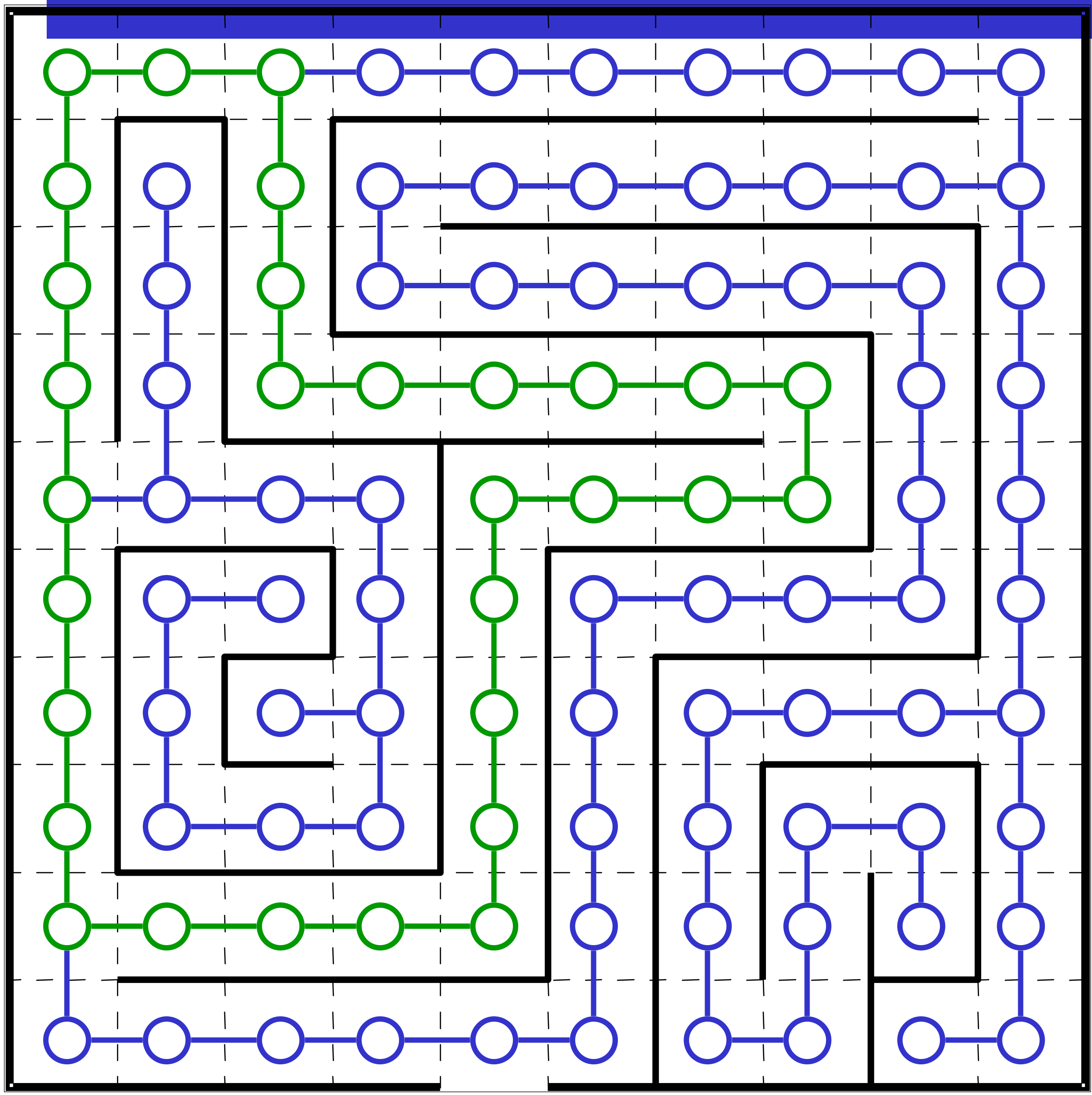
single head, deterministic, with pebbles
visits each node (at least) once
& halts

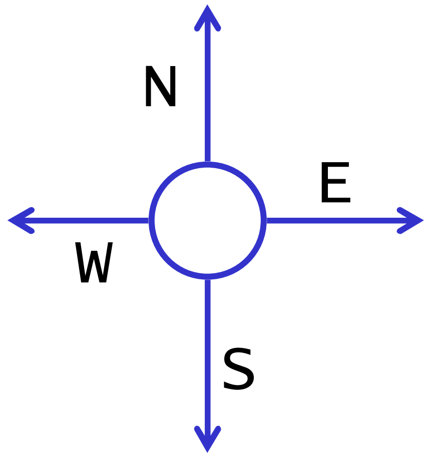
$(\forall x)\ lab_0(x)$

unranked trees, grids, toruses, …
2 pebbles



mazes





Blum & Kozen

two heads!

(*not* nested)

searching with many heads

$$\boxed{FO{+}dTC^k = dPTW^k}$$

for families of *k-searchable* graphs

k heads, deterministic, with pebbles
visits each node (at least) once & halts

additionally
automata: move head to pebble

mazes
not all graphs

Cook & Rackoff
'Jumping Automata'

as promised

on strings, trees, grids, toruses, mazes, …

$$\boxed{\mathrm{FO{+}dTC}^k = \mathrm{dPTW}^k}$$

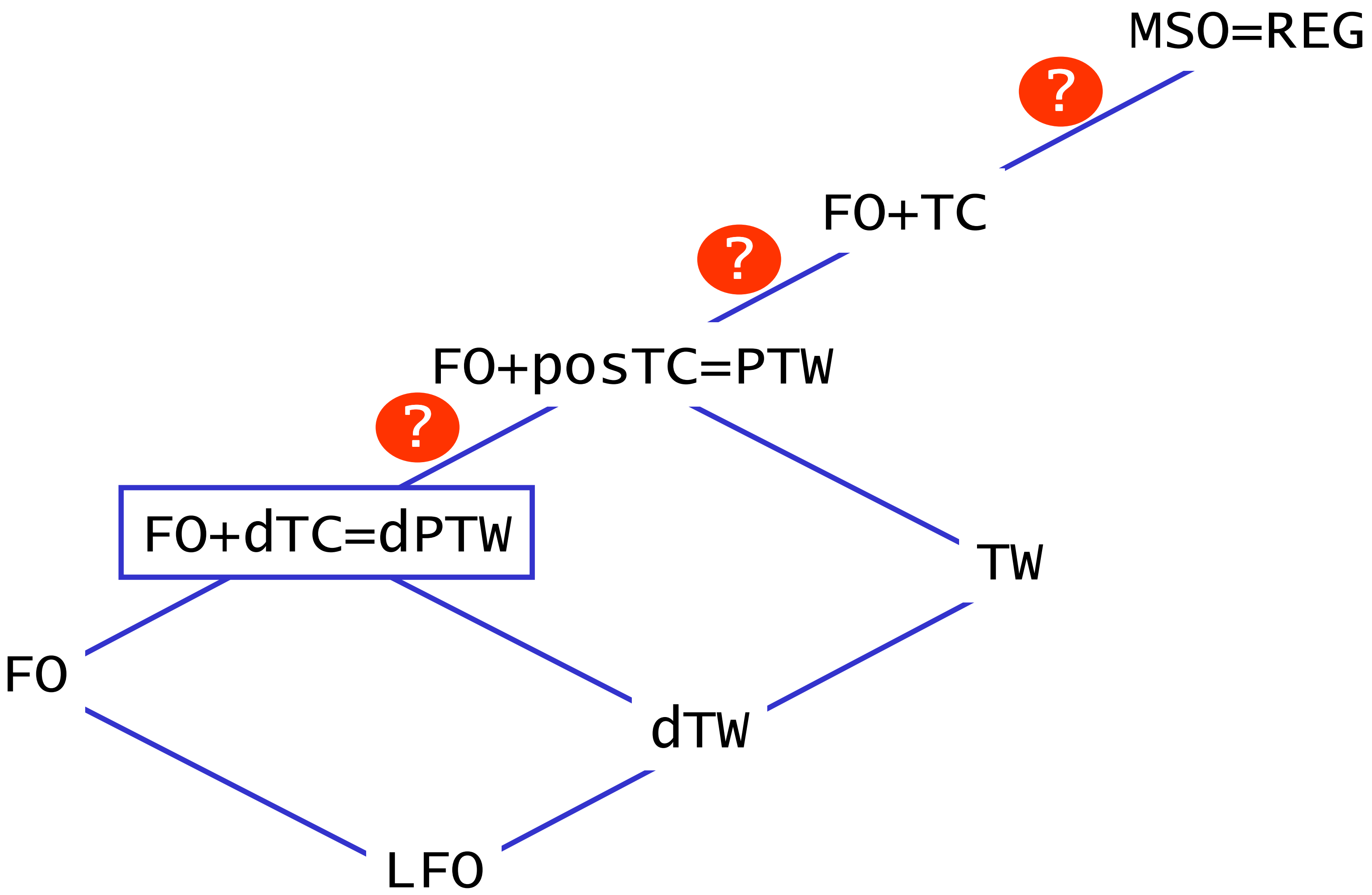
First-Order Logic
+ transitive closure

Multi-Head Automata
+ nested pebbles

arity k ⟺ k heads

- ? hierarchy for k
- ? type of pebbles physical vs. abstract
- ? alternation

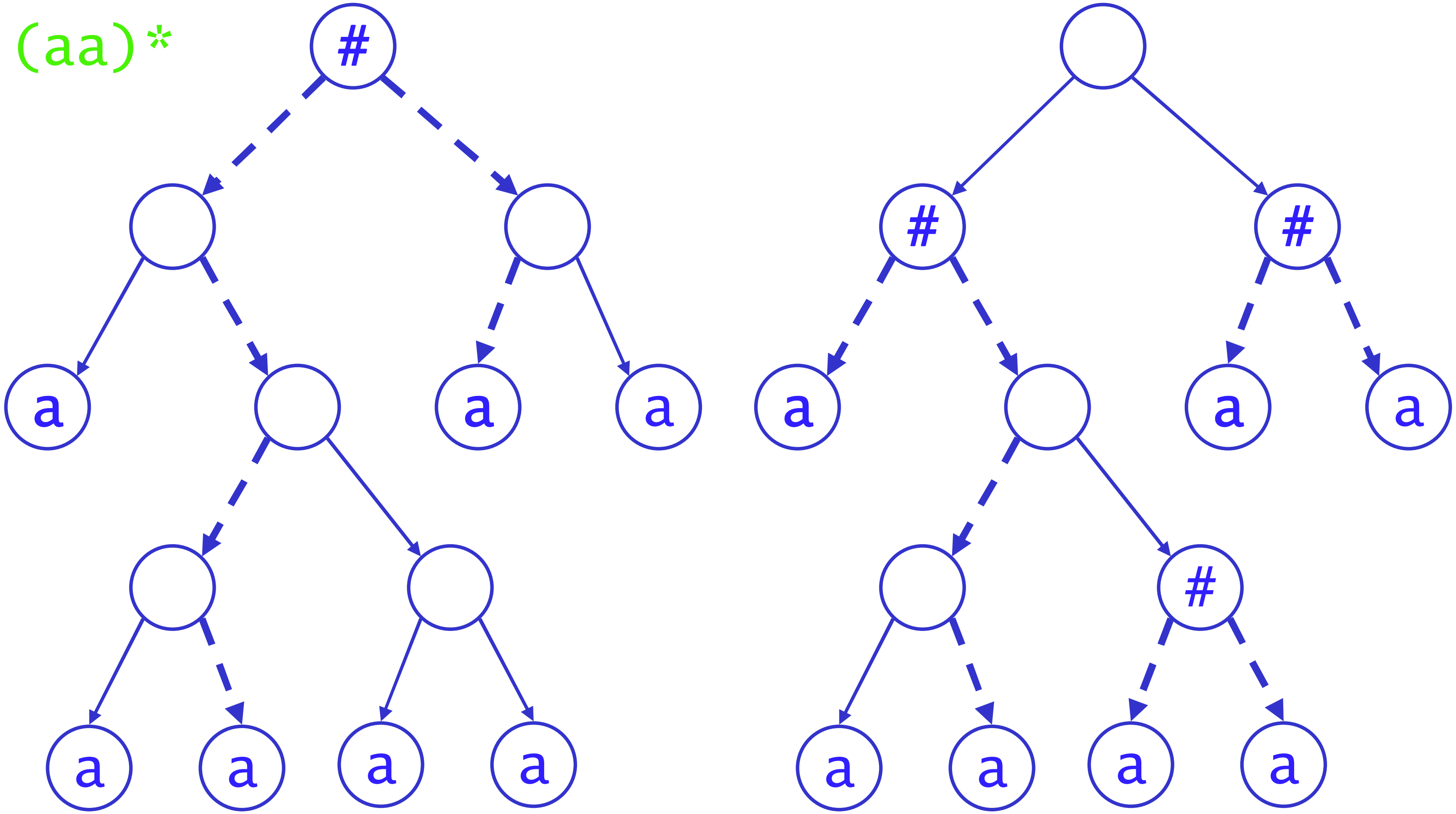
single head on trees




‘tossing Pebbles’
thank you …

paths of even length

Bojanczyk & Colcombet



now as ‘structure’ relative to a-leaves