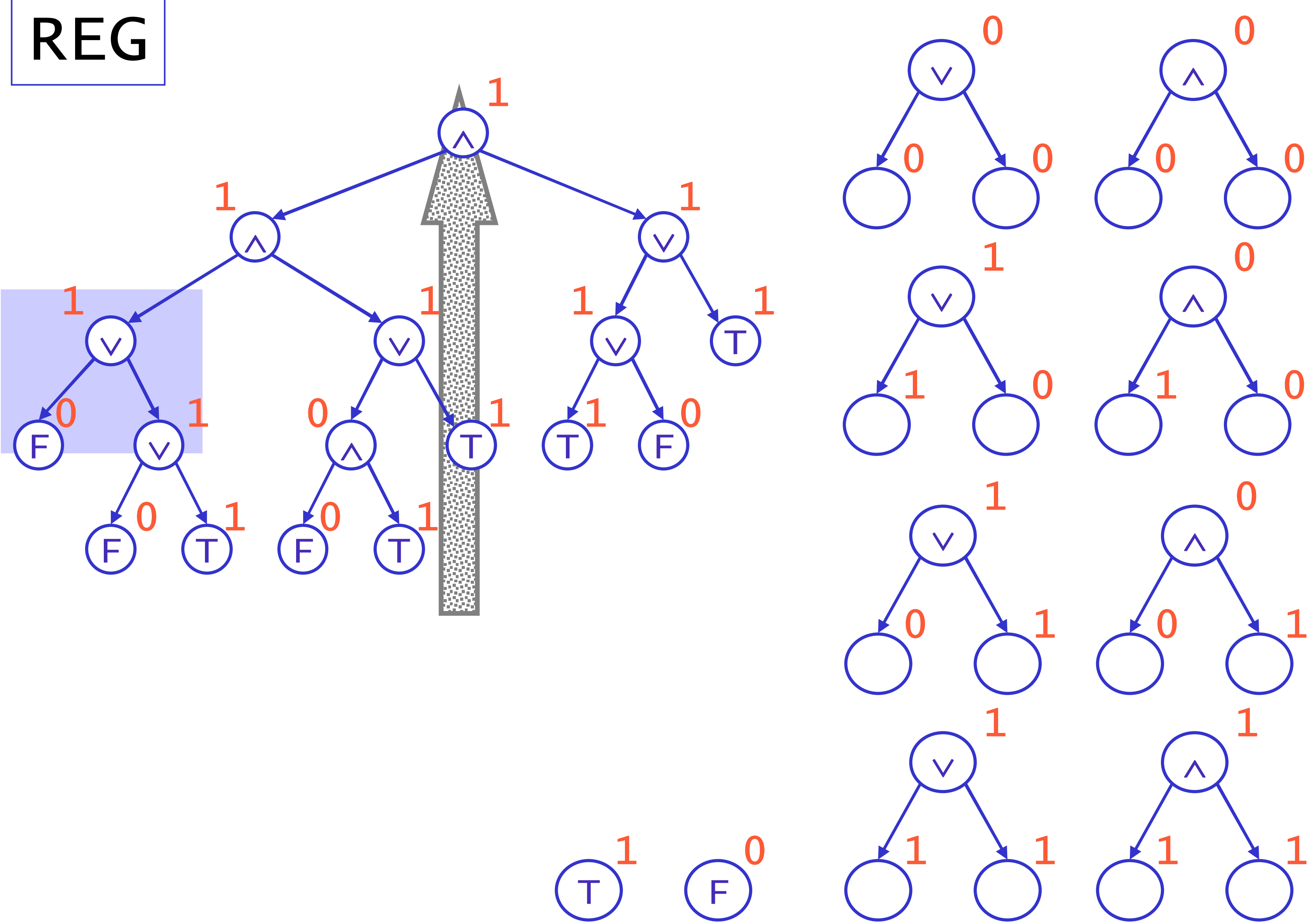
Automata with **Nested Pebbles** and FO Logic with **Transitive Closure**

9.6'06 Bonn

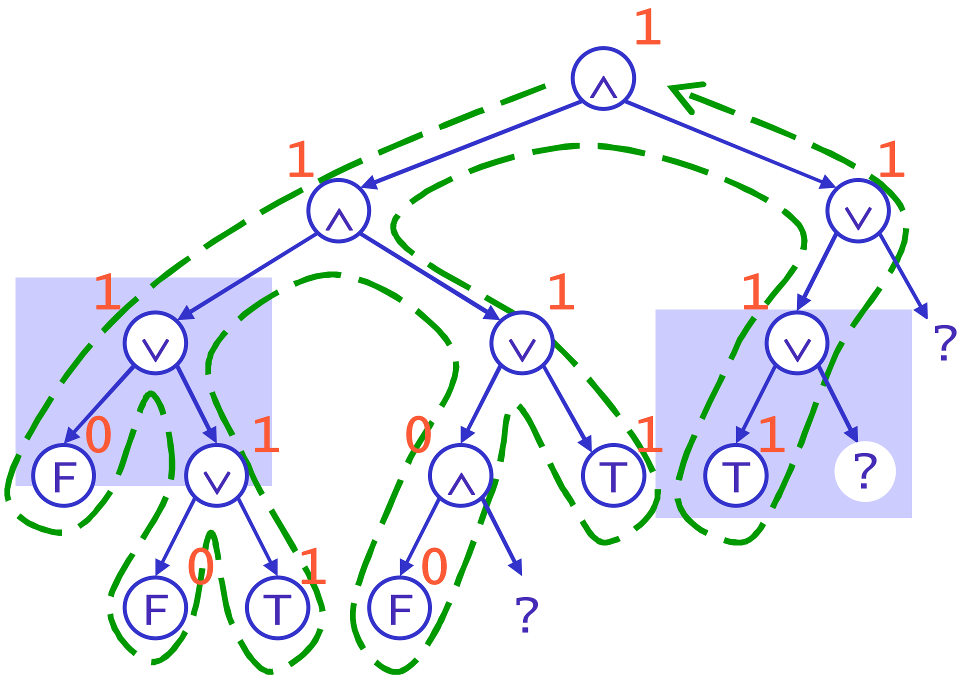
Joost Engelfriet
Hendrik Jan Hoogeboom
Leiden NL

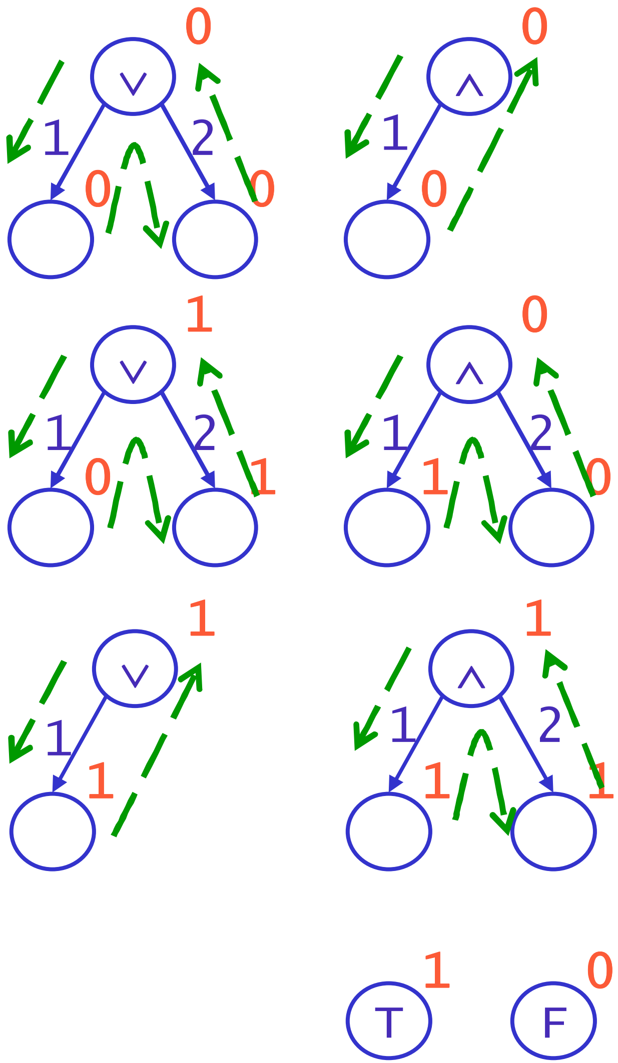
bottom-up tree automata

REG



walking along the tree



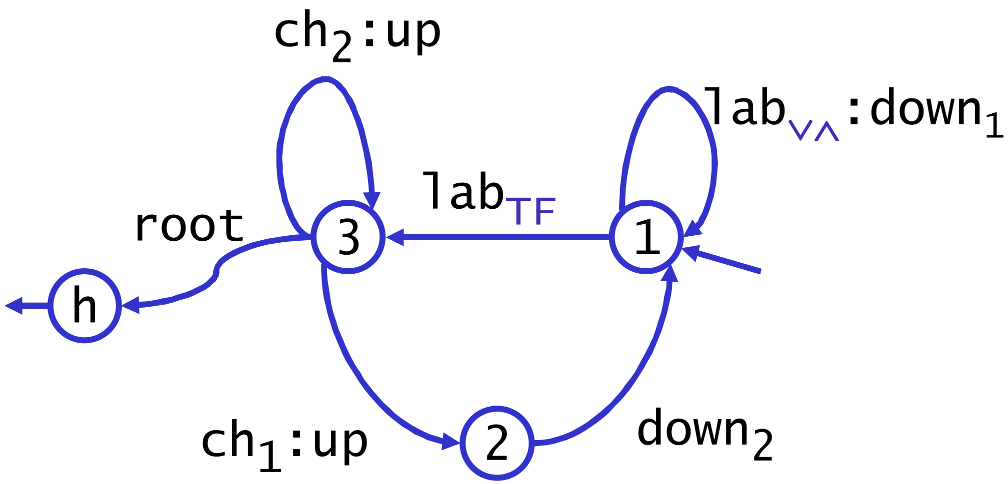


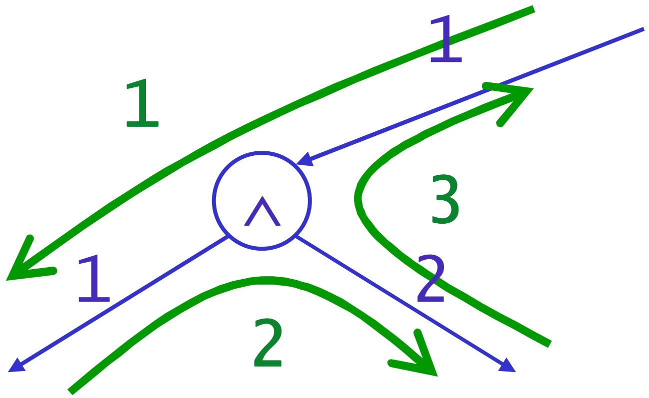
evaluates and/or trees !

cf. two-way finite state automaton

example: tree traversal

TW





walk along edges, moves based on

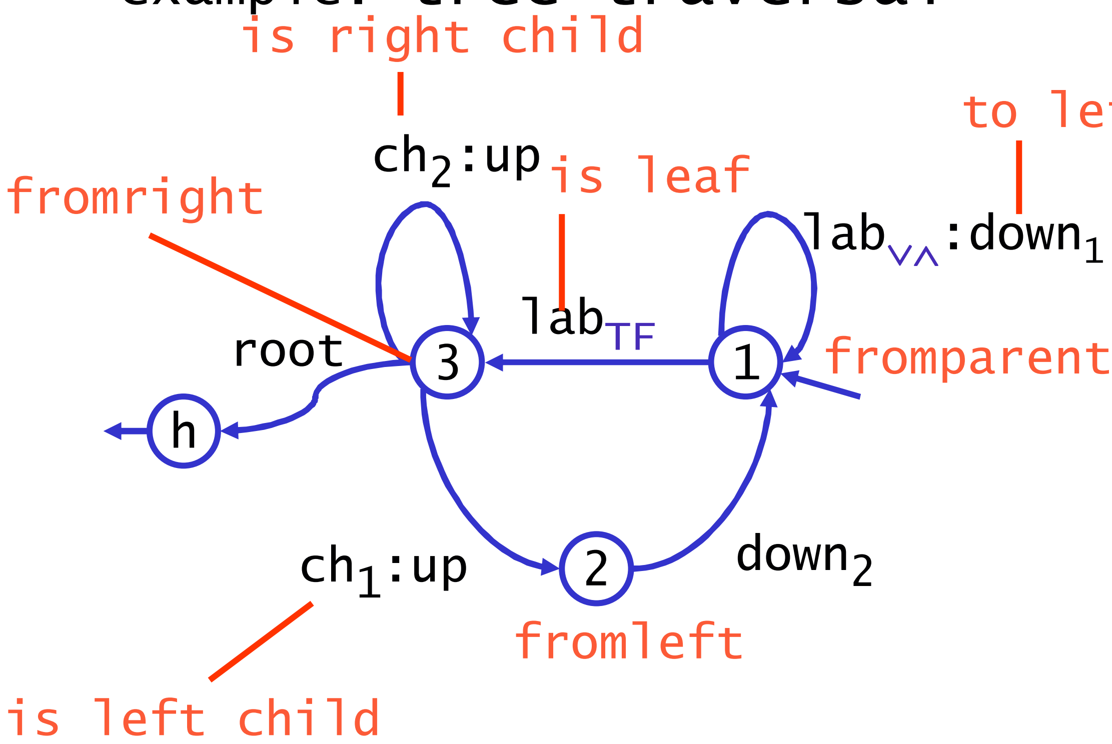
- state
- node label
- child number
  (= incoming edge)

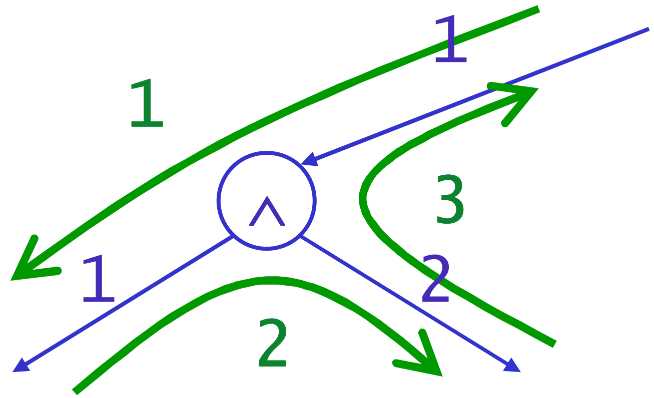
see sheets Mikołaj Bojańczyk

tree-walking automaton

TW

example: tree traversal

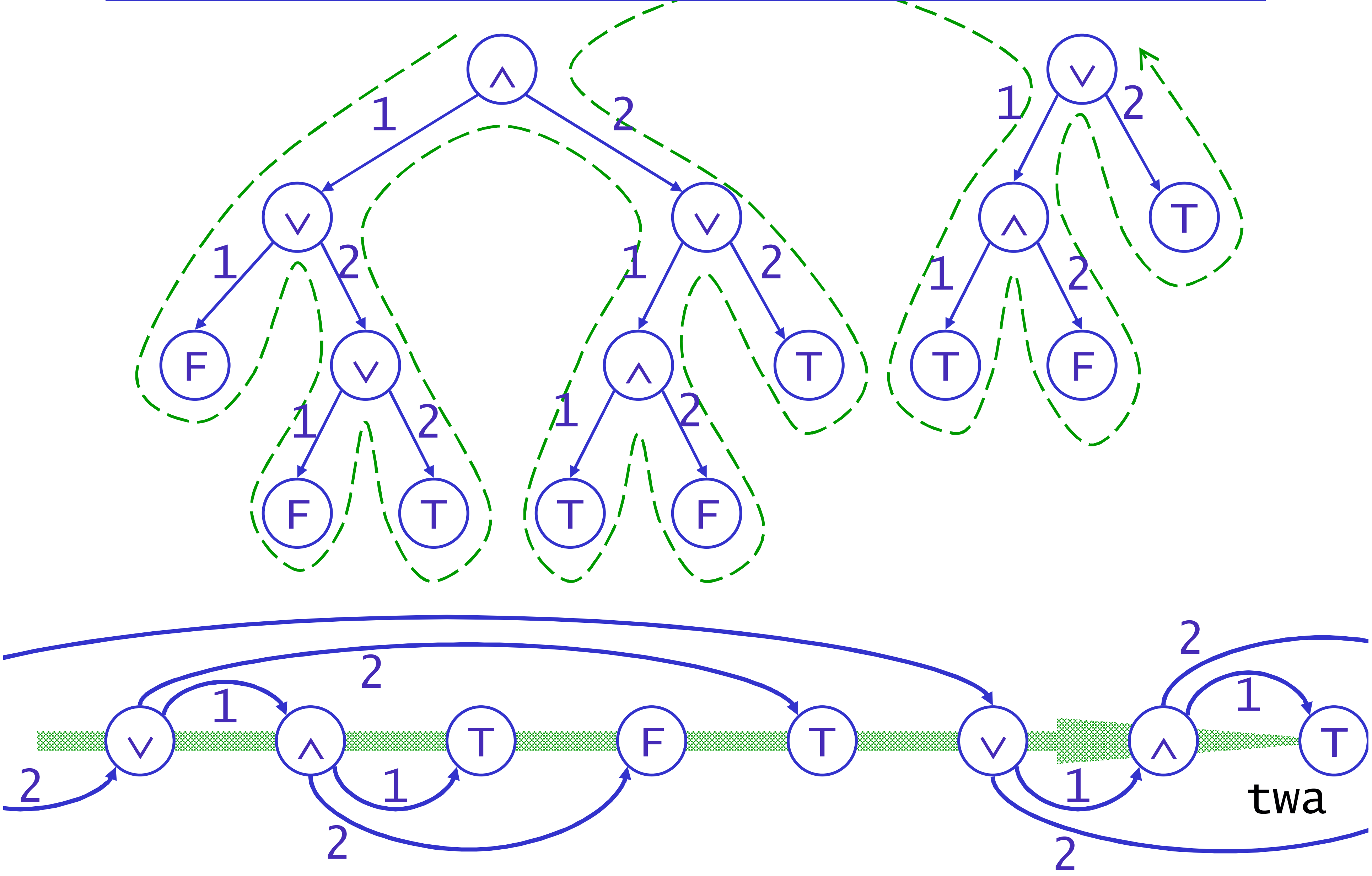




walk along edges, moves based on

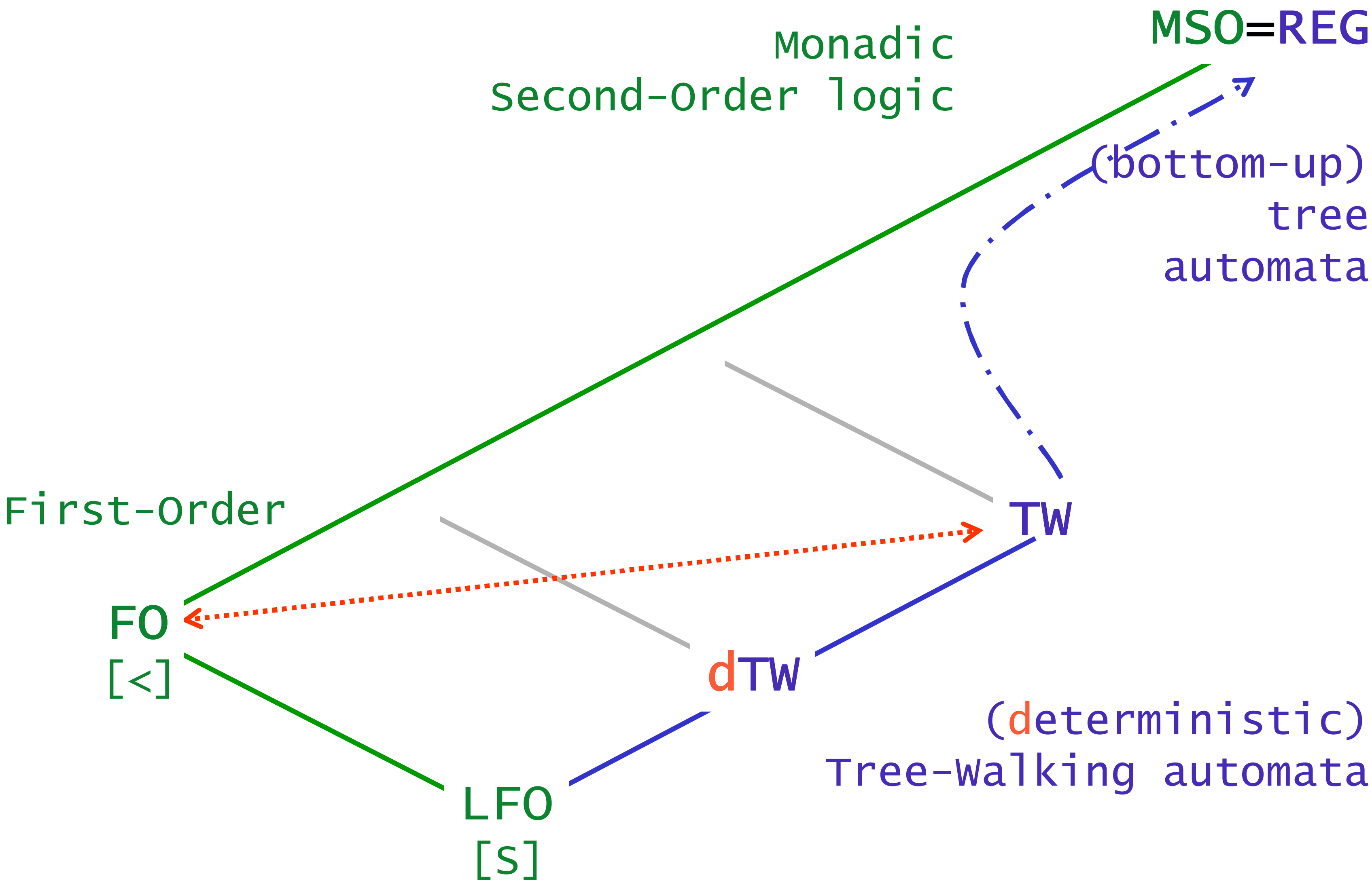
- state
- node label
- child number
  (= incoming edge)

tool: systematic tree-traversal



tree-walking automata

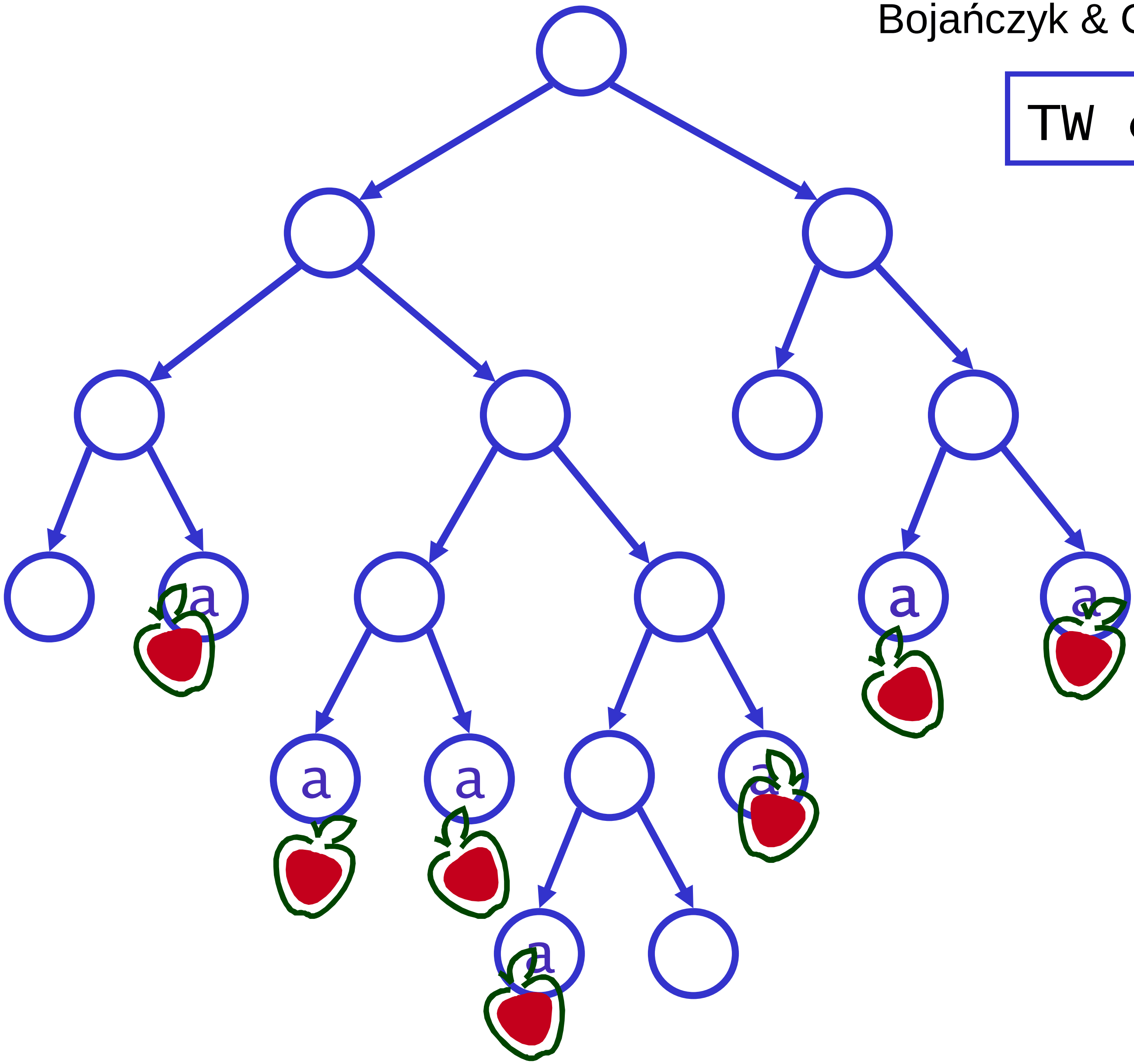
Doner; Thatcher & Wright



‘branching structure’ of even length

Bojańczyk & Colcombet

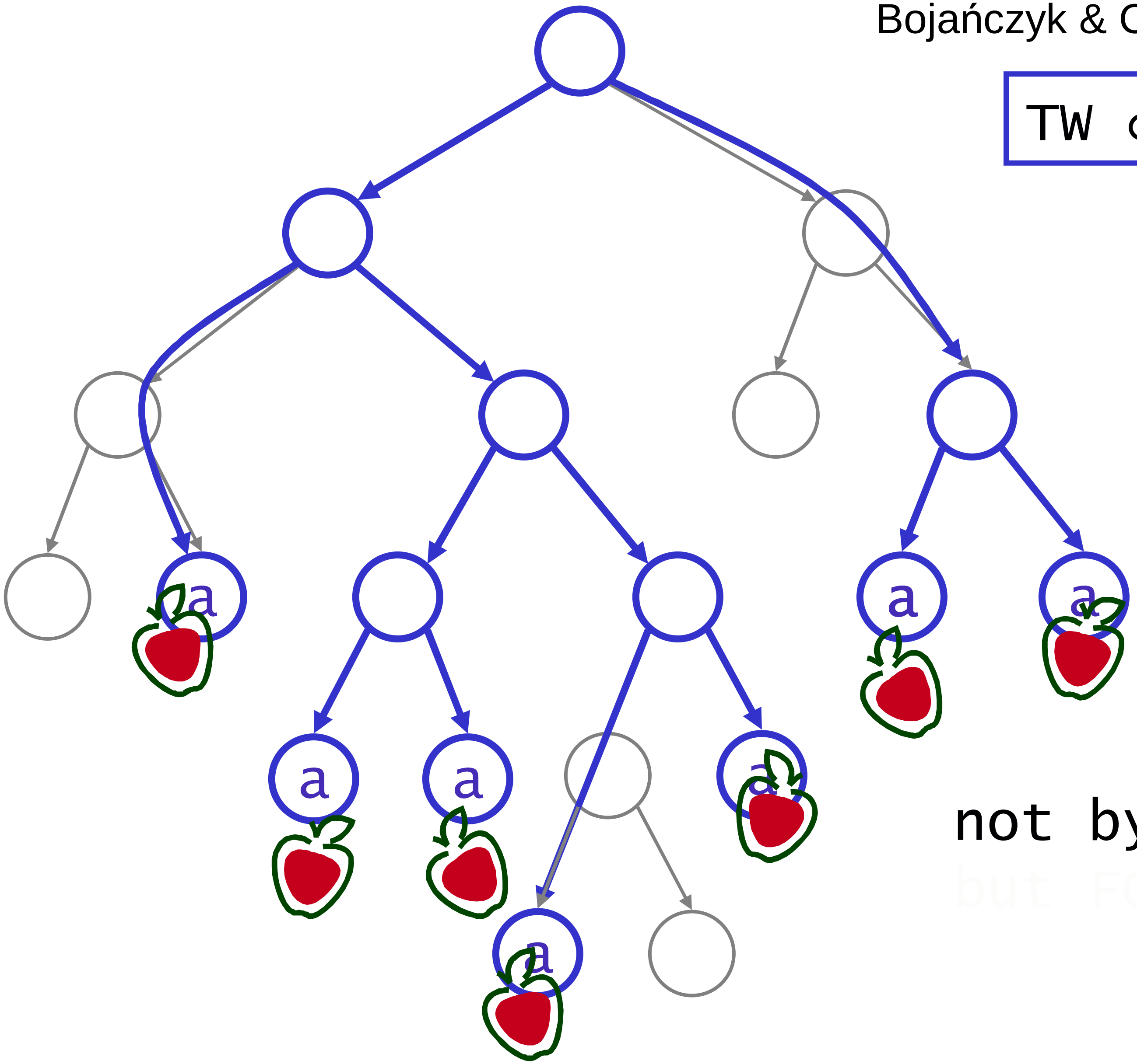
$\boxed{TW \subset REG}$



‘branching structure’ of even length

Bojańczyk & Colcombet

$\boxed{TW \subset REG}$



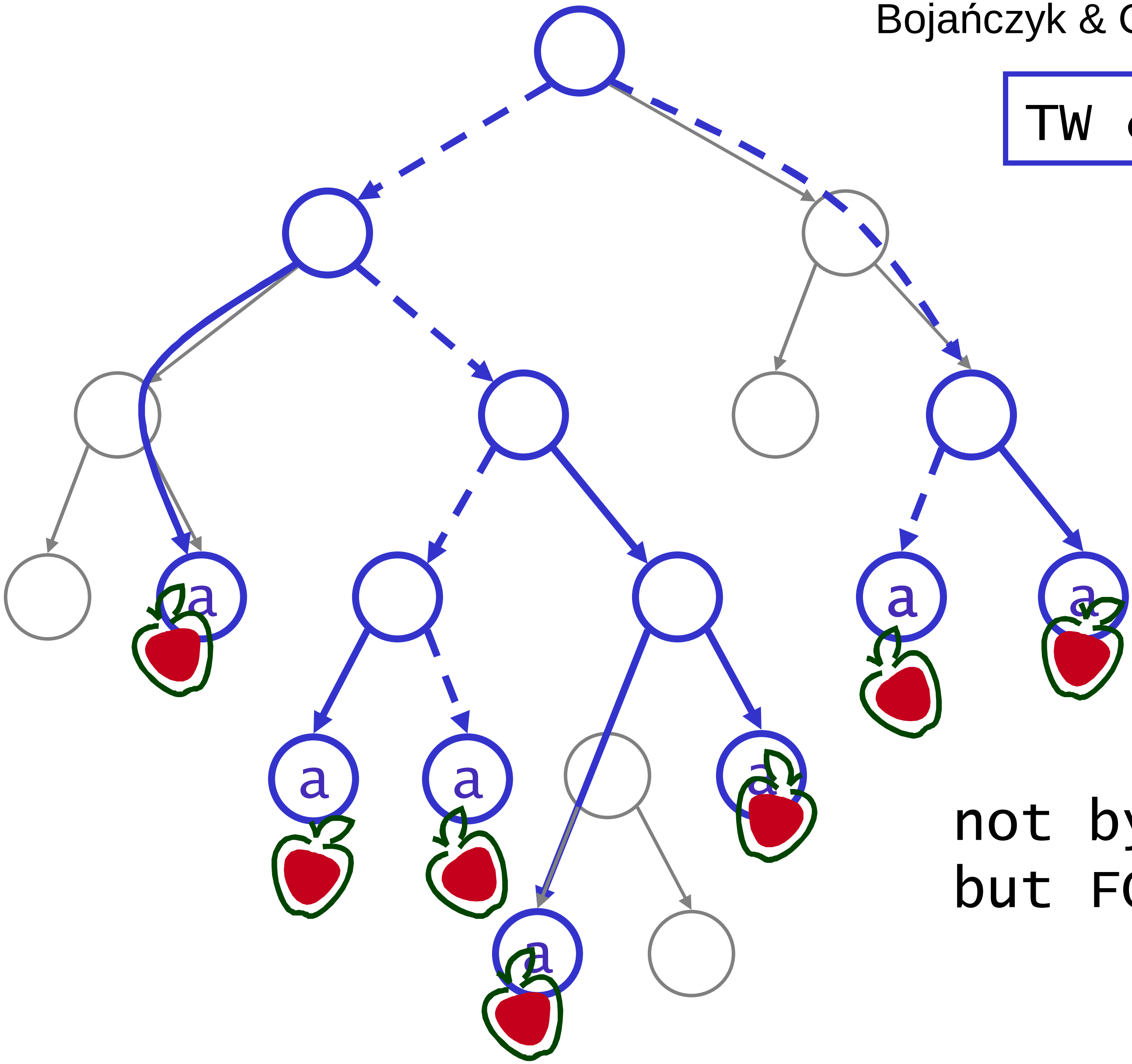
not by TWA

but FO (!)

‘branching structure’ of even length

Bojańczyk & Colcombet

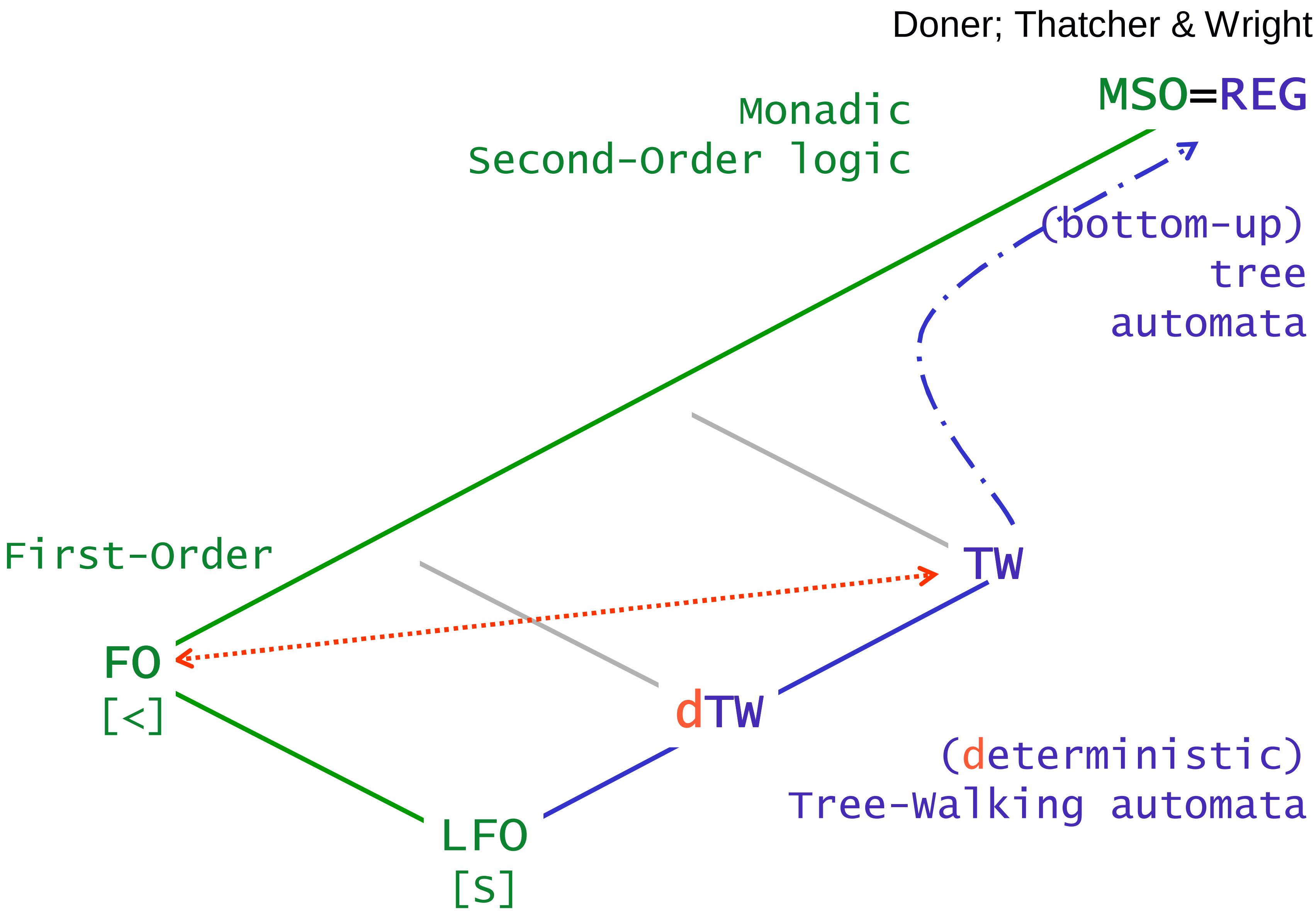
TW ⊂ REG



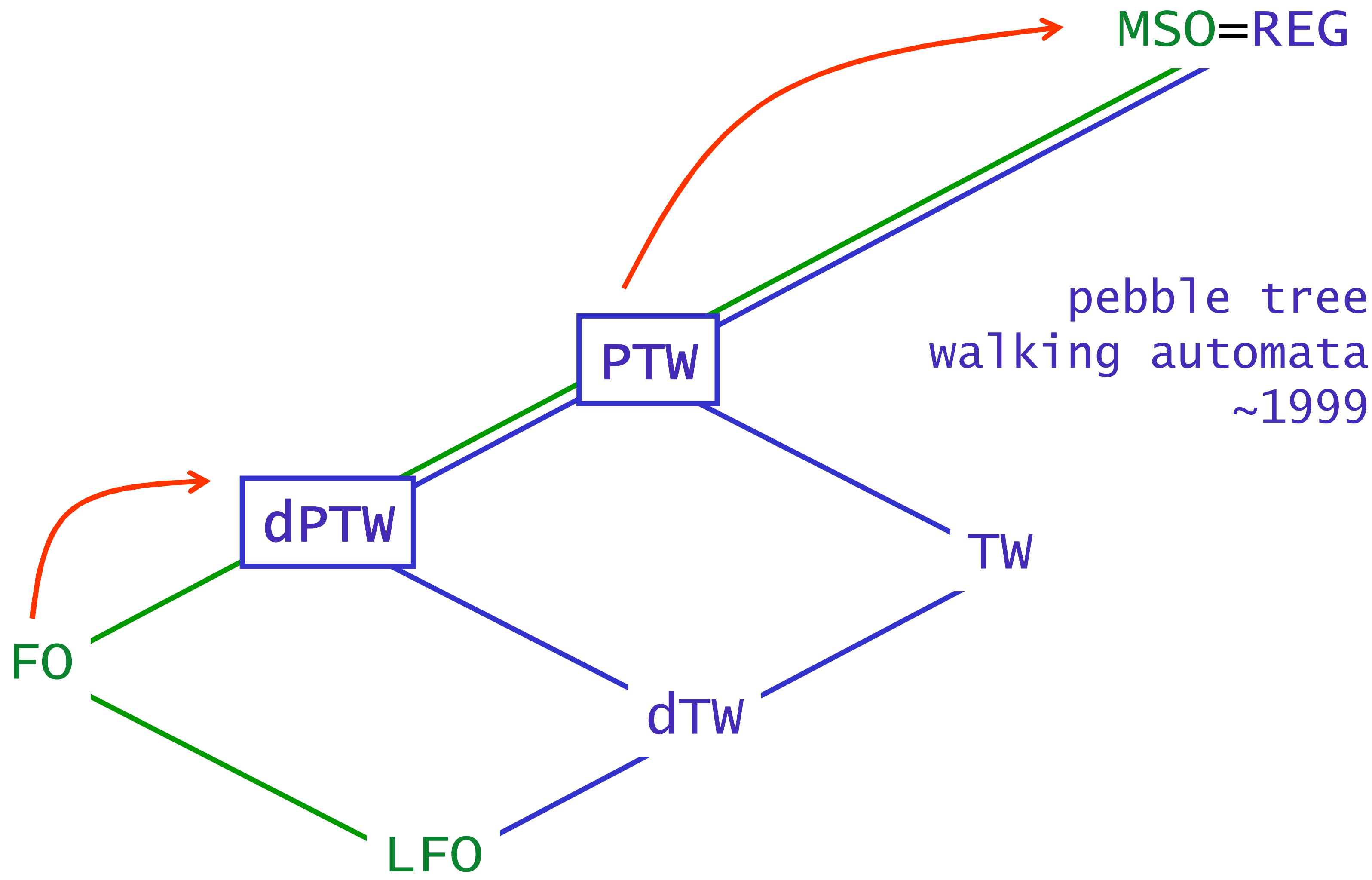
not by TWA
but FO (!)

(aa)*

tree-walking automata



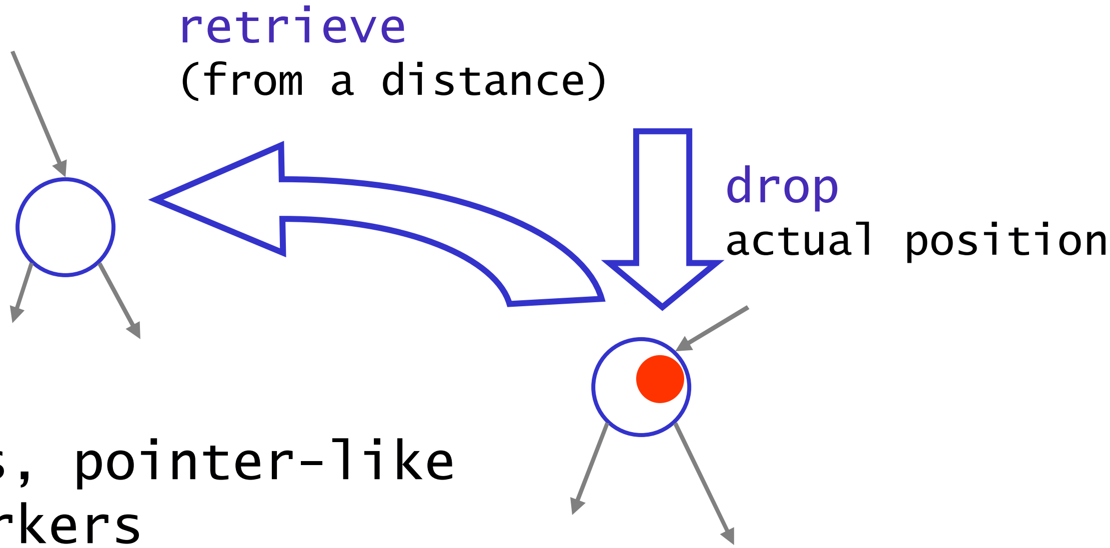
tree-walking automata



pebble tree
walking automata
~1999

adding nested pebbles

pebble: marks a node

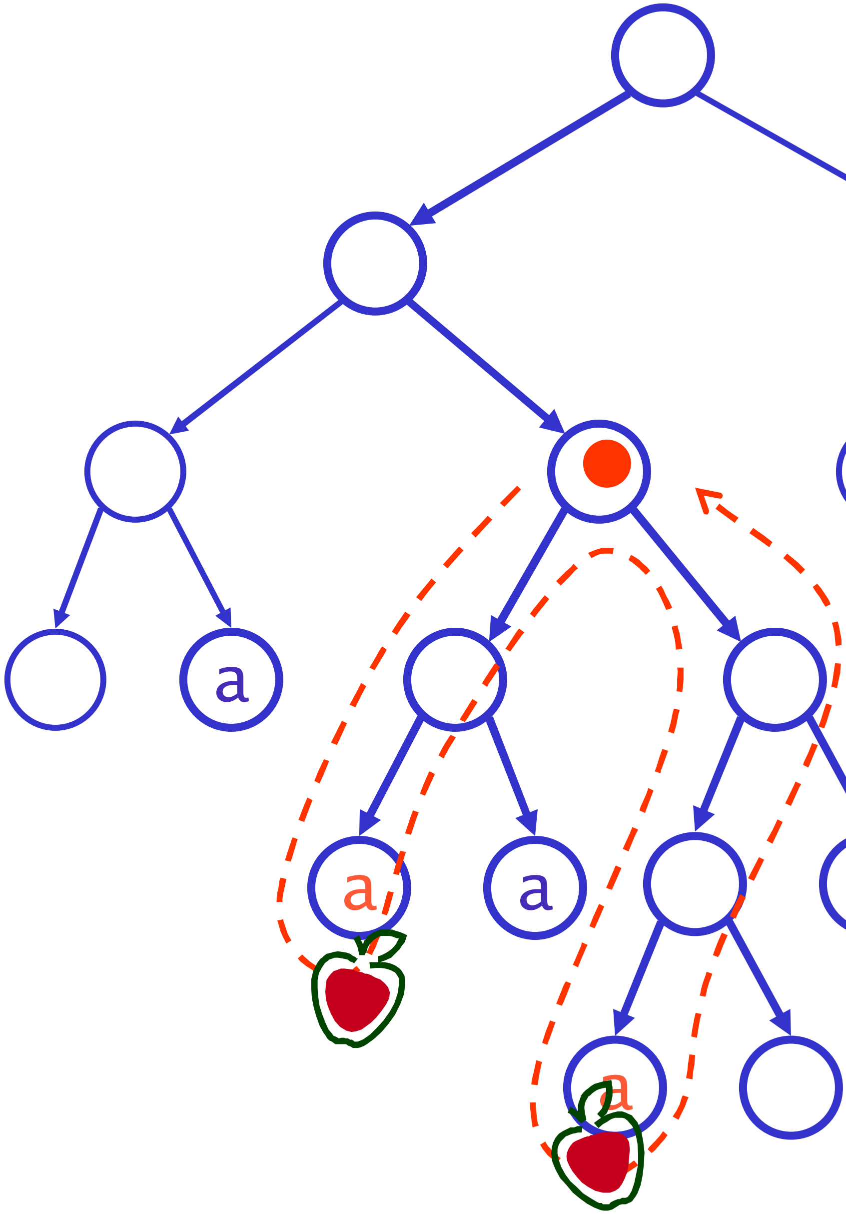


- strong pebbles, pointer-like
  ‘abstract’ markers
  rather than ‘physical’
- *nested lifetimes* LIFO
- fixed number for automaton
- can be distinguished

‘regular’ extension
(for single head on trees)

‘branching structure’ of even length

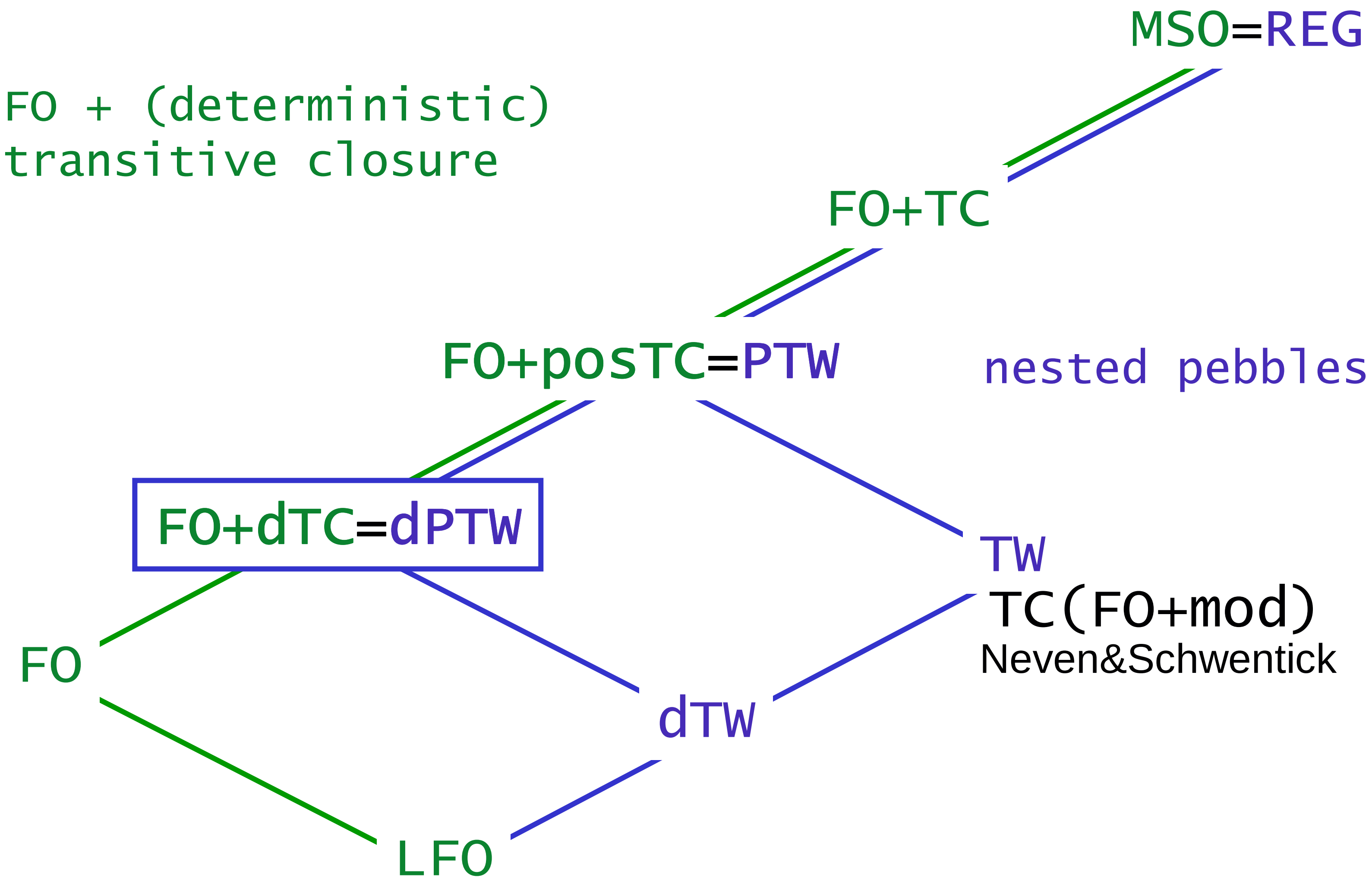
using a pebble



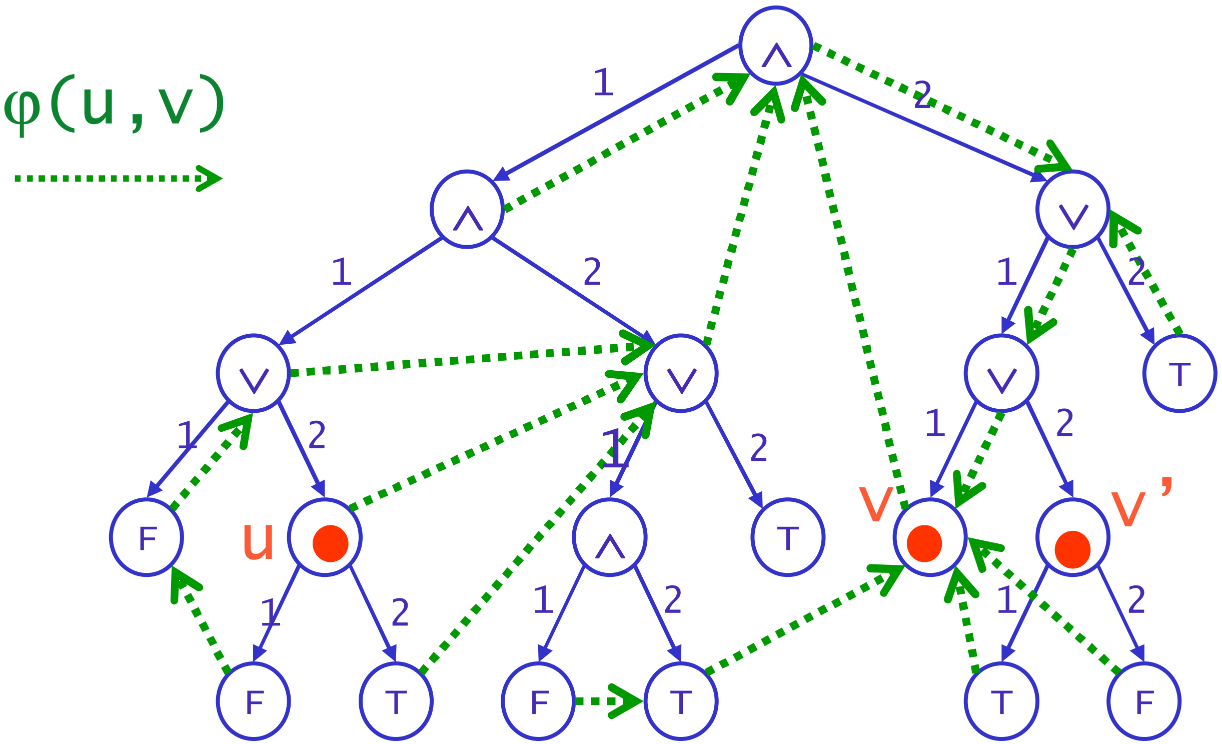
not by TWA
but FO (!)

(aa)*

this talk


MSO=REG
FO + (deterministic)
transitive closure
FO+TC
FO+posTC=PTW
nested pebbles
FO+dTC=dPTW
TW
TC(FO+mod)
Neven&Schwentick
FO
dTW
LFO

$\varphi(u,v)$



$\varphi^*(u,v)$ unary tc

deterministic tc: φ functional

$\varphi(u,v,z)$

background

- XML document transformation
  - single head on (unranked) trees
- transitive closure vs. automata
  - descriptive complexity
  - strings, trees, n-dim grids, …
- graph exploration
  - many heads on graphs ‘robots’
  - grids, toruses, mazes, …

classic result for strings

[non]deterministic logarithmic space

Immerman

First-Order Logic
+ transitive closure

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

Multi-Head Automata
(two-way)

arity k ⟺ k heads

Bargury&Makowsky

fits in our framework

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

First-Order Logic
+ transitive closure

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

Multi-Head Automata
+ 'nested pebbles'

arity k $\Longleftrightarrow$ k heads

we (WTA) mostly like…

on trees,

single head

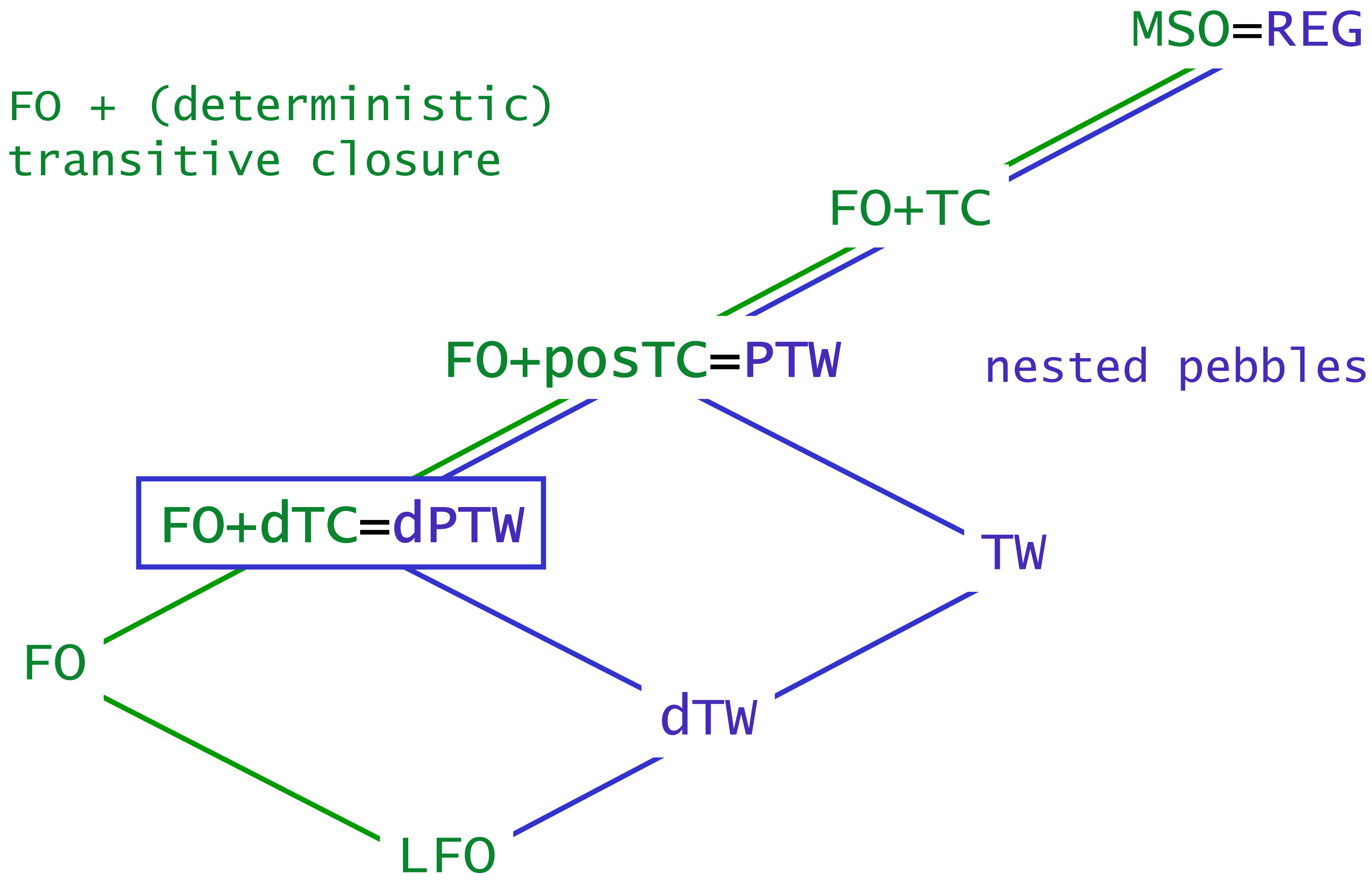
First-Order Logic
+ transitive closure

unary $\varphi^*(x,y)$

Multi-Head Automata
+ ‘nested pebbles’

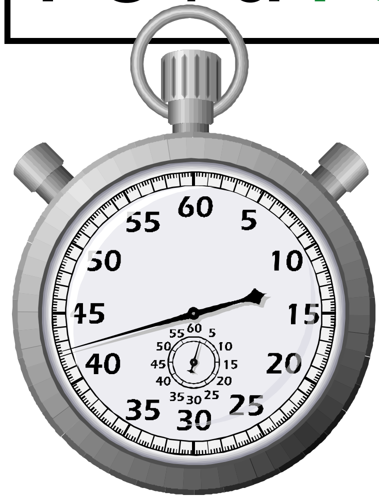
(but this is not a talk on trees only)

single head on trees



main result

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



proof summary
manager style

(deterministic,
single head,
unary tc,
on trees)

FO+dTC $\subseteq$ dPTW

(1) logic to nested pebbles

$\varphi \rightarrow \mathcal{A}$

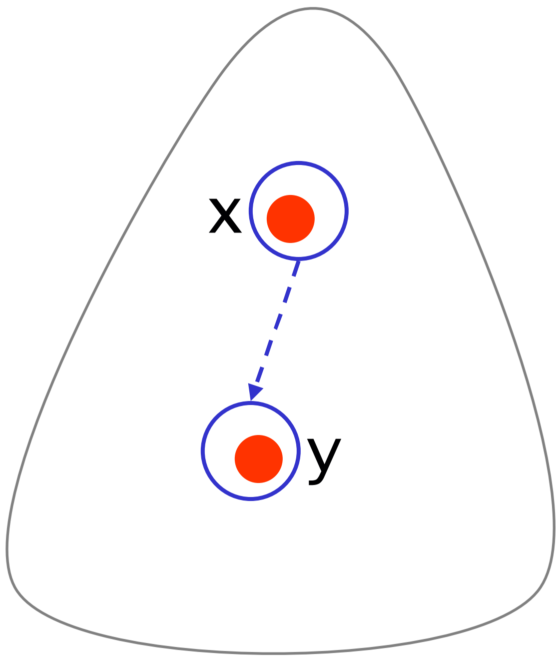
always halting
free variables ~
fixed pebbles

$lab_a(x)$
$edg_i(x,y)$

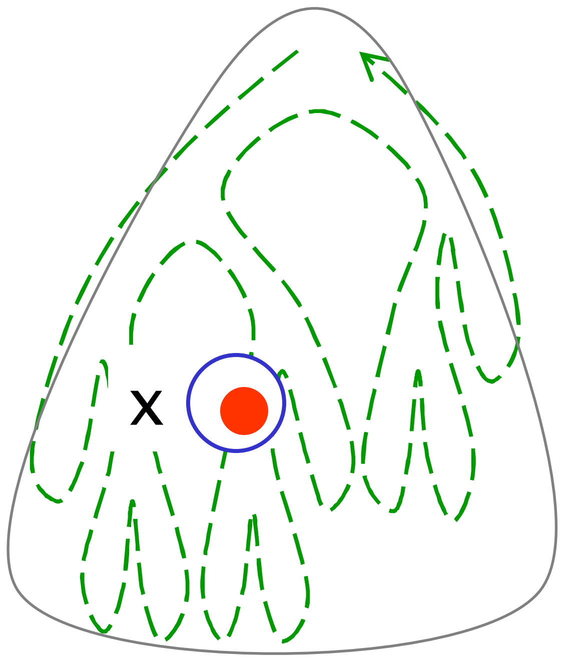
$x \leq y$
$x = y$

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

$\varphi^*(x,y)$

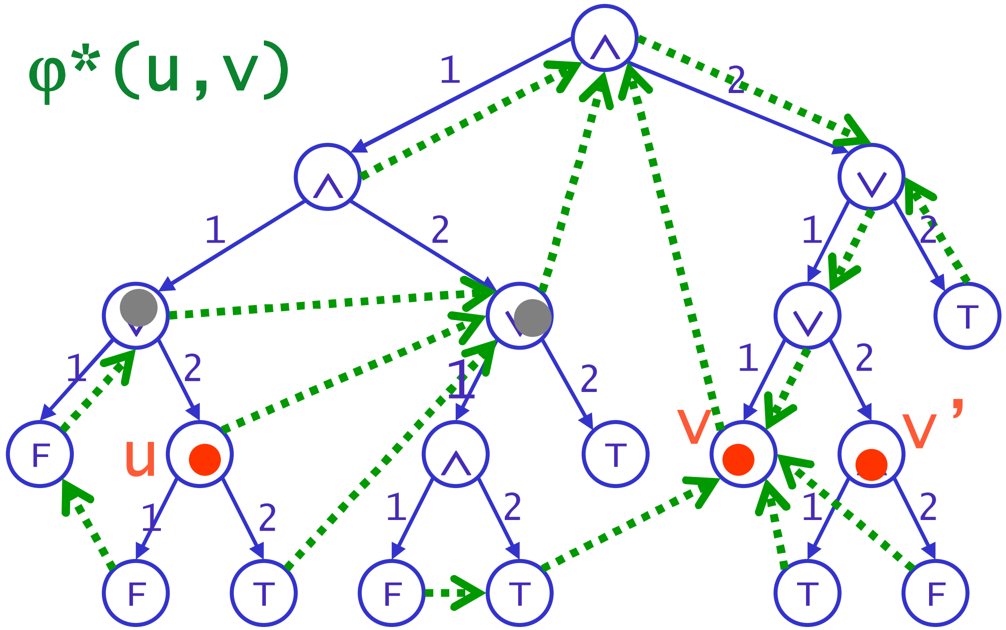


$x \leq y$



$\forall x \ \varphi(x) \quad \mathcal{A}_\varphi$

(1_{ctd}) transitive closure



functional!

but implicit

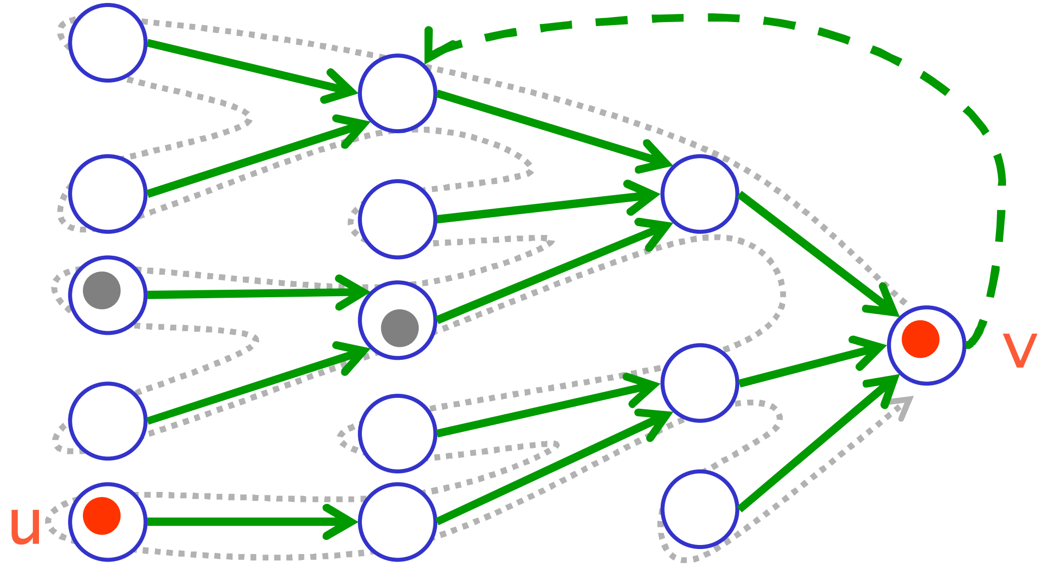
$\mathcal{A}_\varphi$

tree walking

implicit φ-tree

reconstruct locally

backwards! Sipser



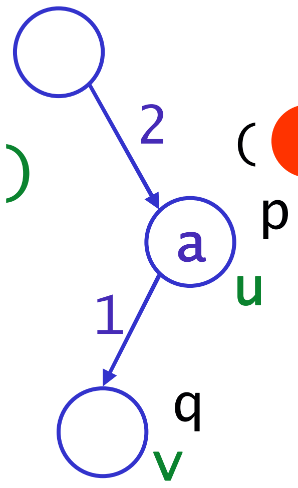
$dPTW \subseteq FO{+}dTC$

(2) nested pebbles to logic

i single move $\varphi_{pq}(u,v)$

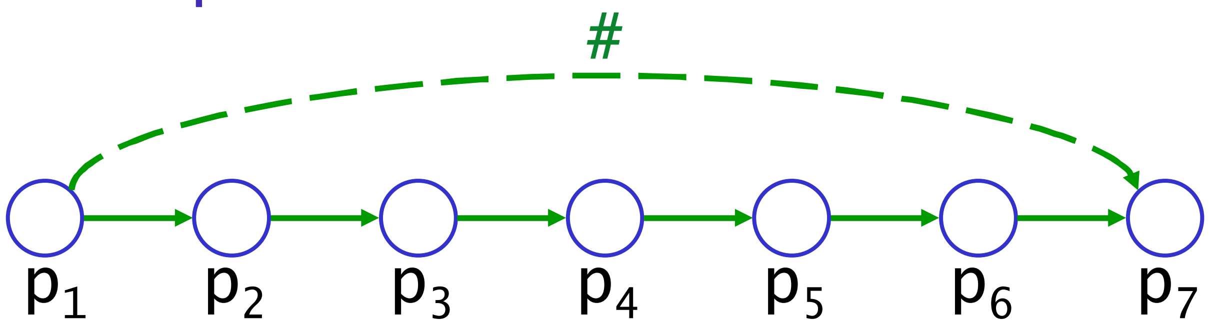
$$lab_a(u) \wedge (\exists u')edg_2(u',u)$$
$$\wedge\ u \neq x_3 \ \wedge\ edg_1(u,v)$$

free variables for pebbles



(3 *not* here)

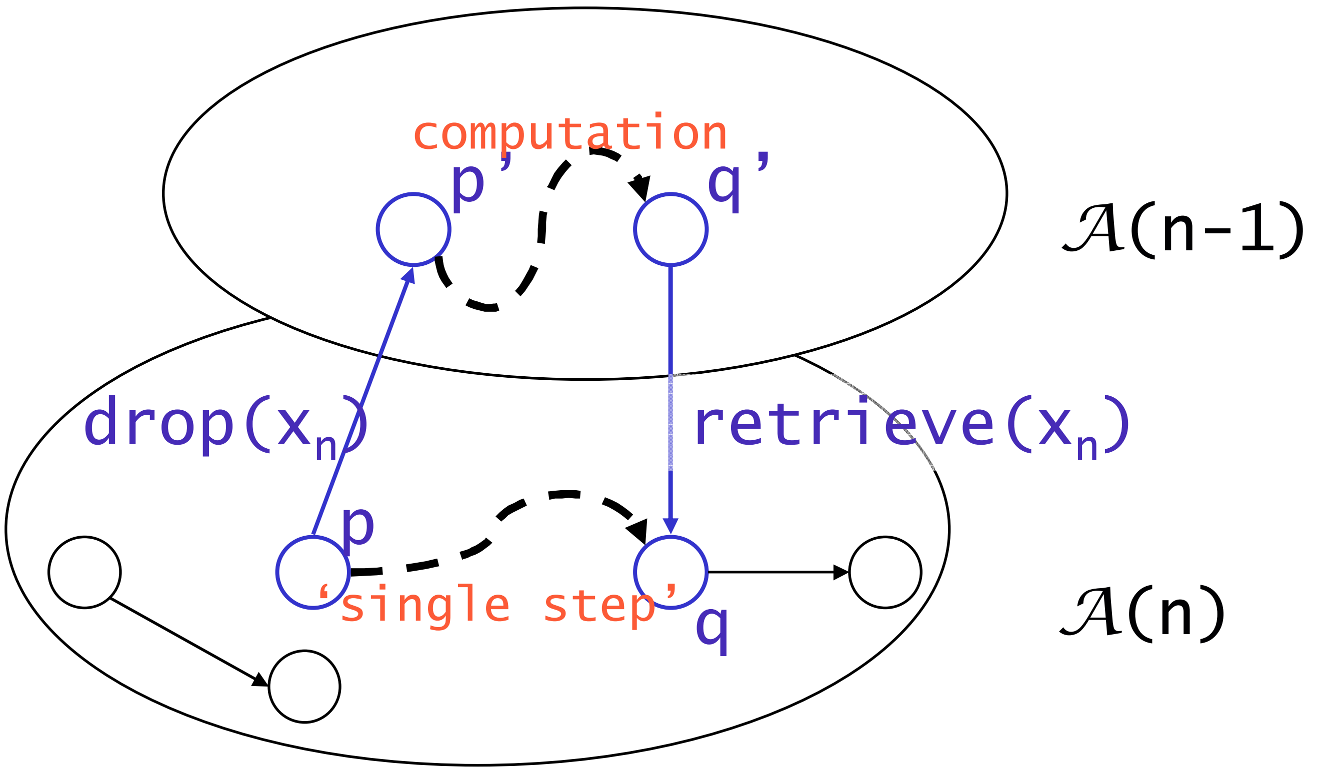
ii computation ~ tc with states



$\varphi_{pq}(x,y)$ $\longrightarrow$

Kleene: removing states finite aut to reg expr dPTW $\subseteq$ FO+dTC

(2ctd) dropping pebbles

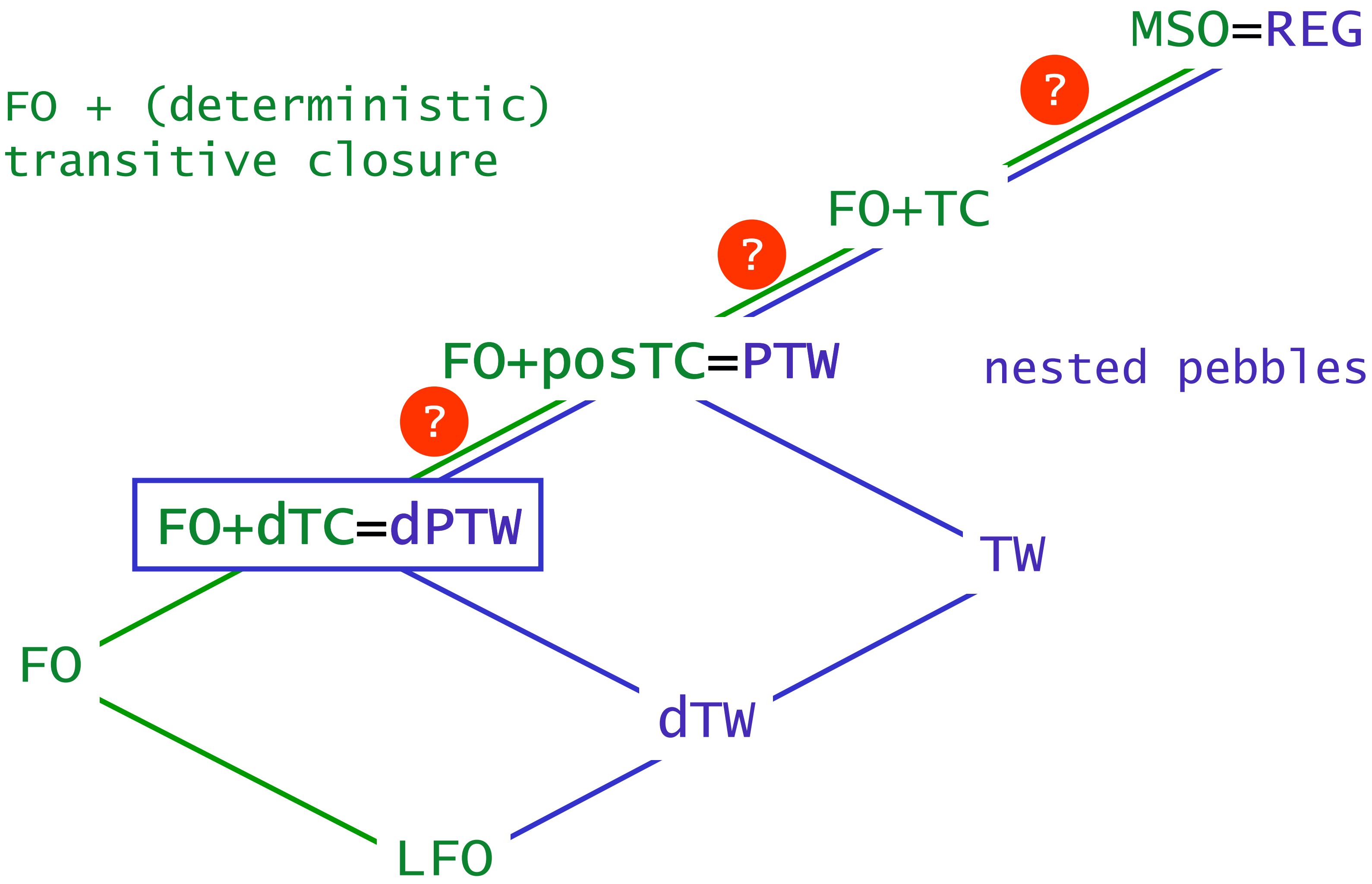


transition diagram

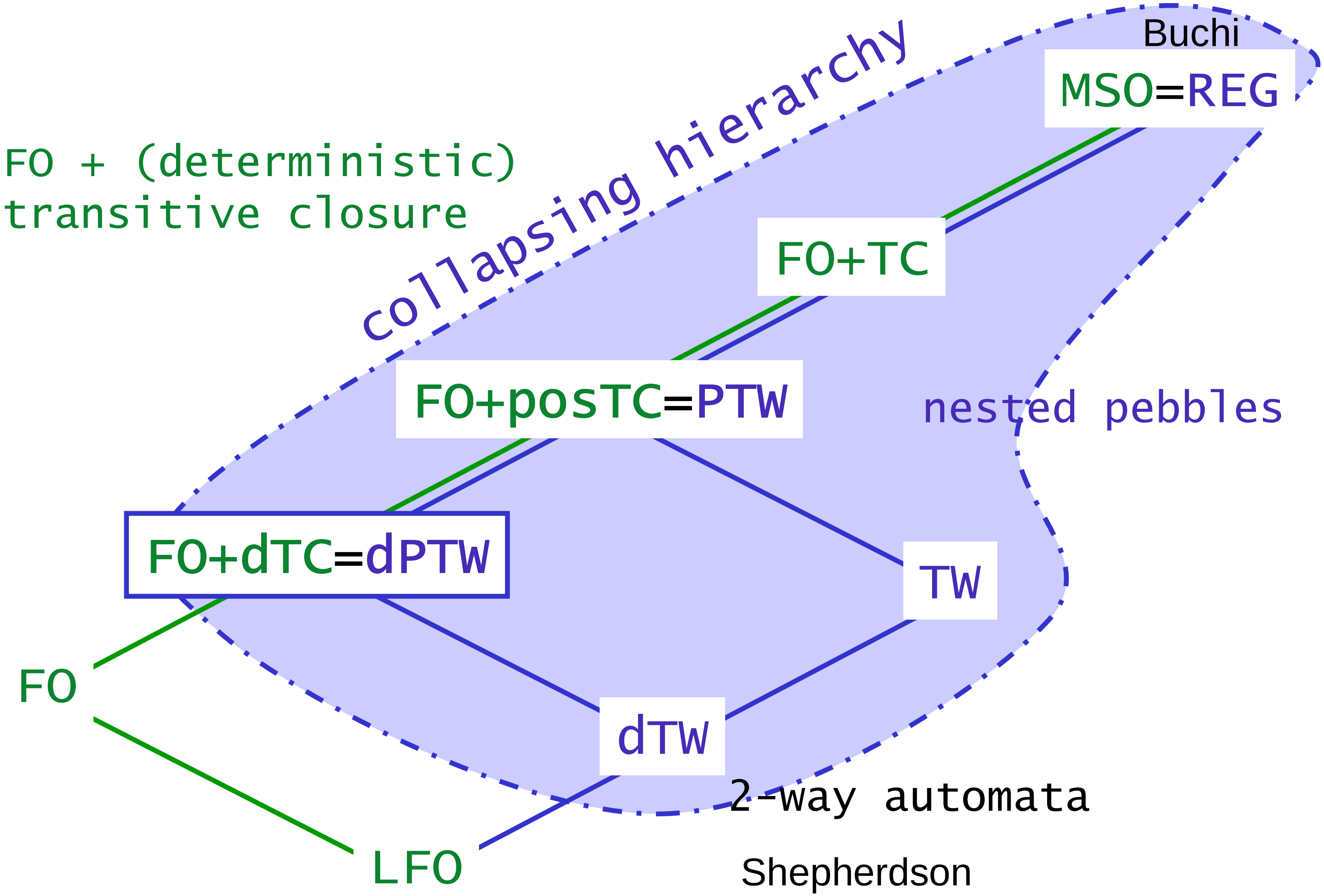
$$\varphi^{n}_{pq}(u,v) = \varphi^{(n-1)\#}_{p'q'}(u,v)$$

replacing x_n by u

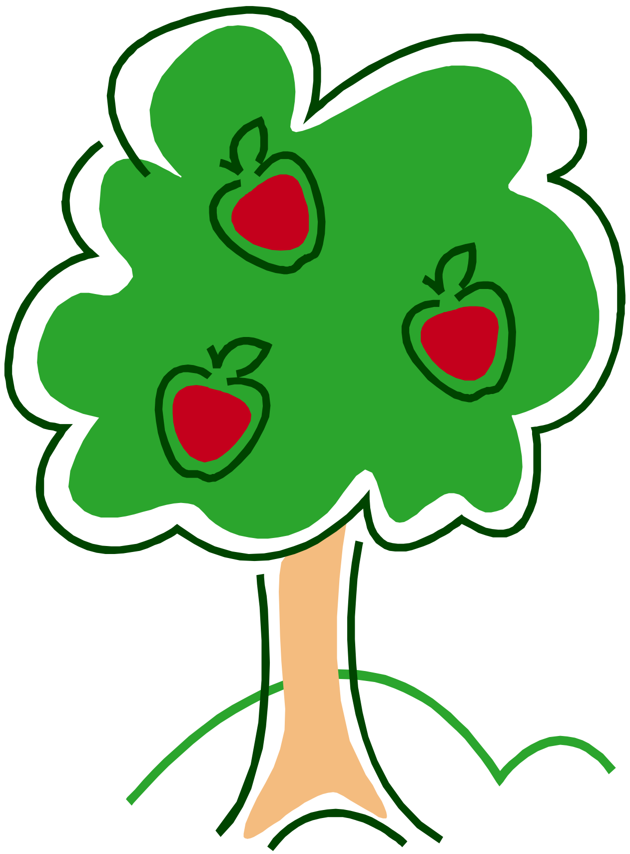
single head on trees



single head on strings

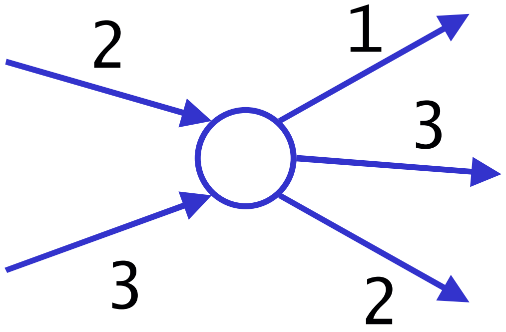


note

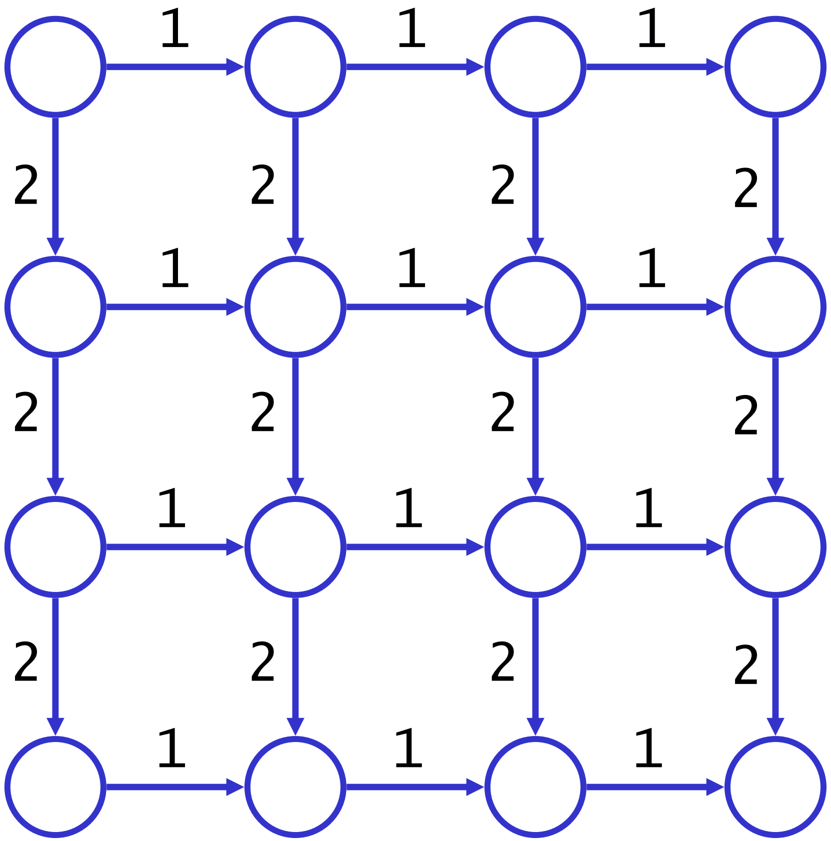


The following slides on graphs were not shown during the presentation. They were designed to illustrate that our result is valid for more general families that have a ‘guide’, a (pebble) automaton that visits all nodes and halts. Note the torus (one head two pebble guide) and the maze (two heads). Only small adaptations to either the logical or automaton framework are necessary.

from trees to graphs



locally injective



grid, torus

$lab_a(x)$
$edg_i(x,y)$

$\cancel{x \leq y}$
$x = y$

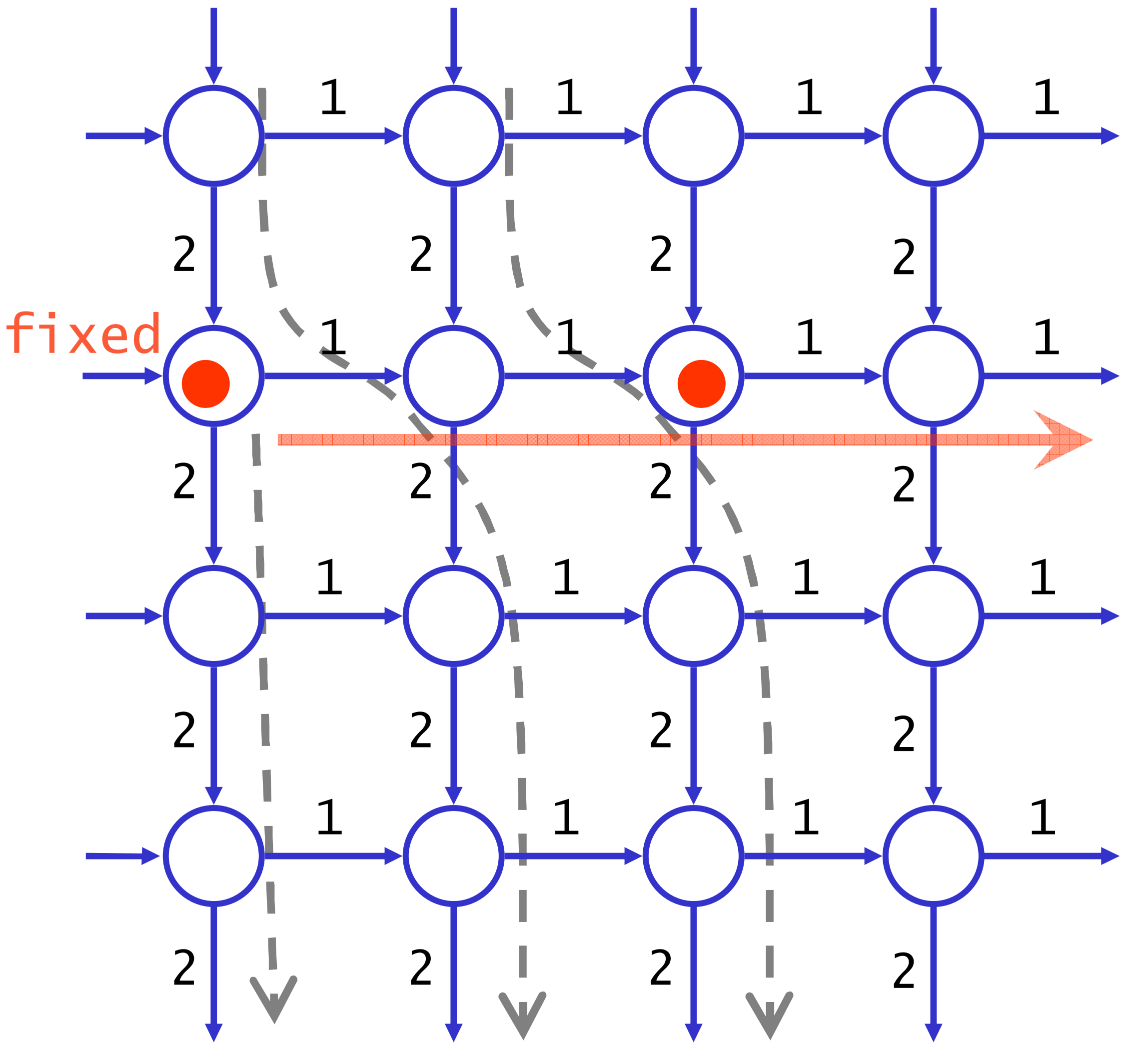
$\neg \quad \wedge \quad \vee$
$\forall x \quad \exists x$

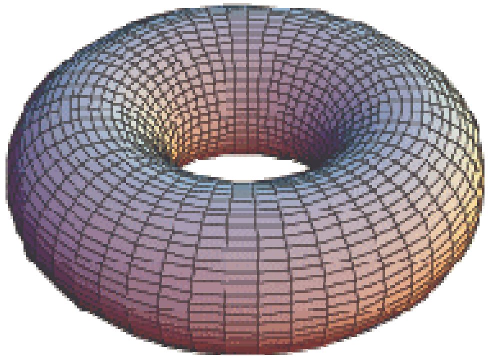
$\varphi^*(x,y)$

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

for families of graphs
(i.e. with fixed label alphabets)

walking the torus





two pebbles

(nested)

graphs with a guide

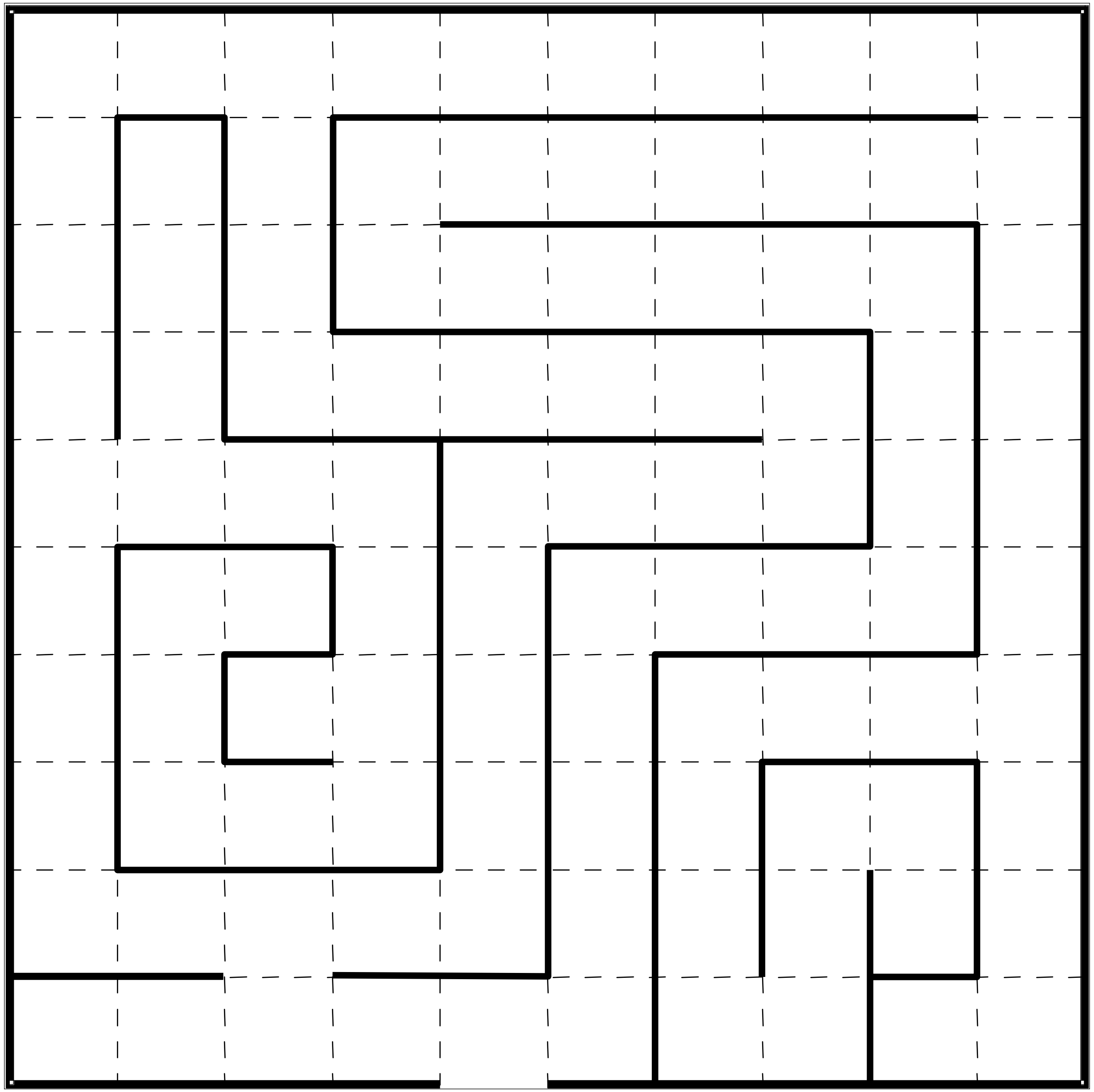
$$\boxed{\mathrm{FO{+}dTC}^k = \mathrm{dPTW}^k}$$

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

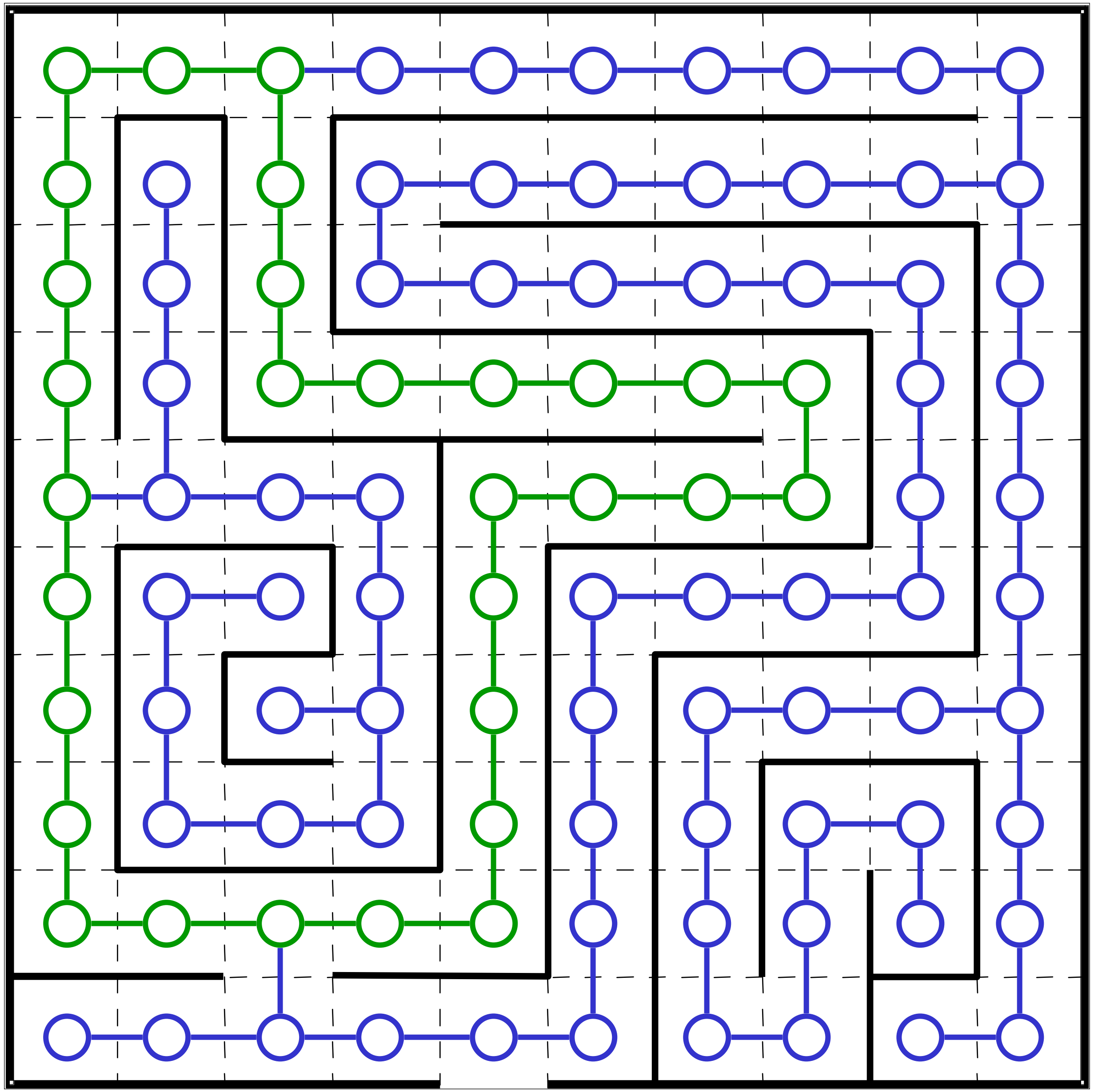
guide: *single* head, deterministic
with pebbles
visits each node (at least) once
& halts

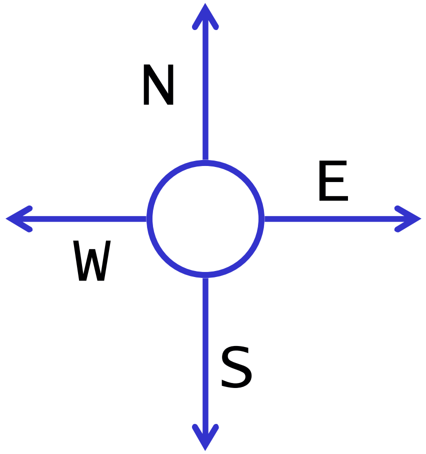
$(\forall x)\ lab_0(x)$

unranked trees, grids, toruses, …
2 pebbles

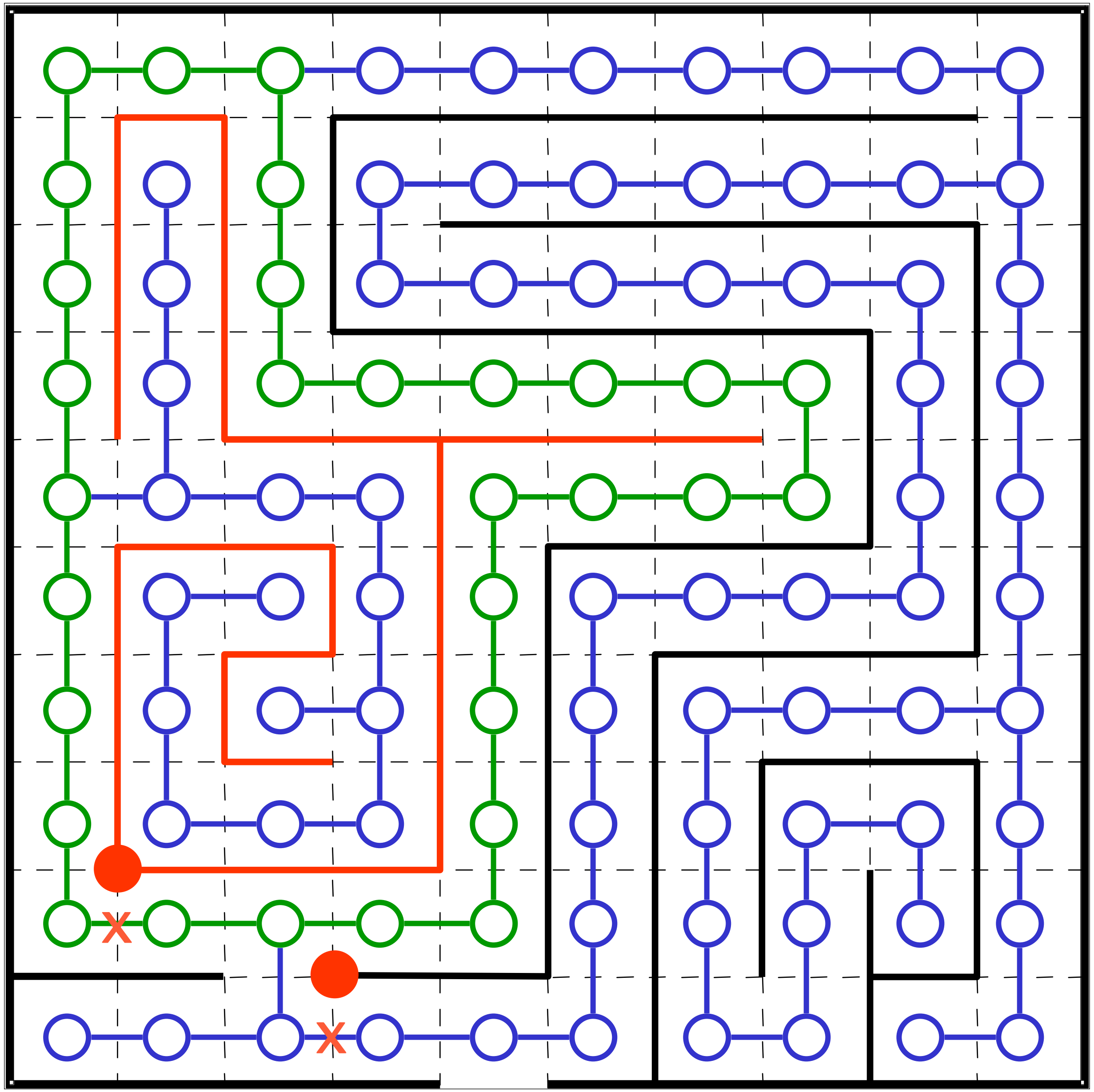


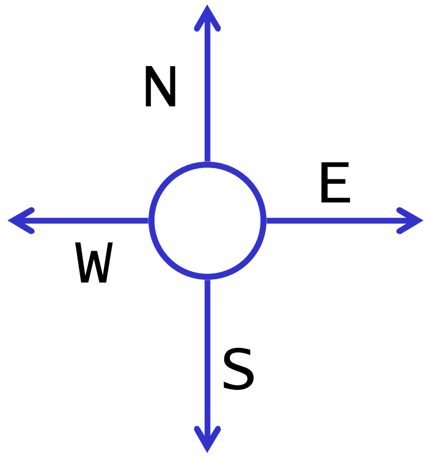
mazes





mazes





Blum & Kozen

two heads!

(*not* nested)

searching with many heads

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

for families of *k-searchable* graphs

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

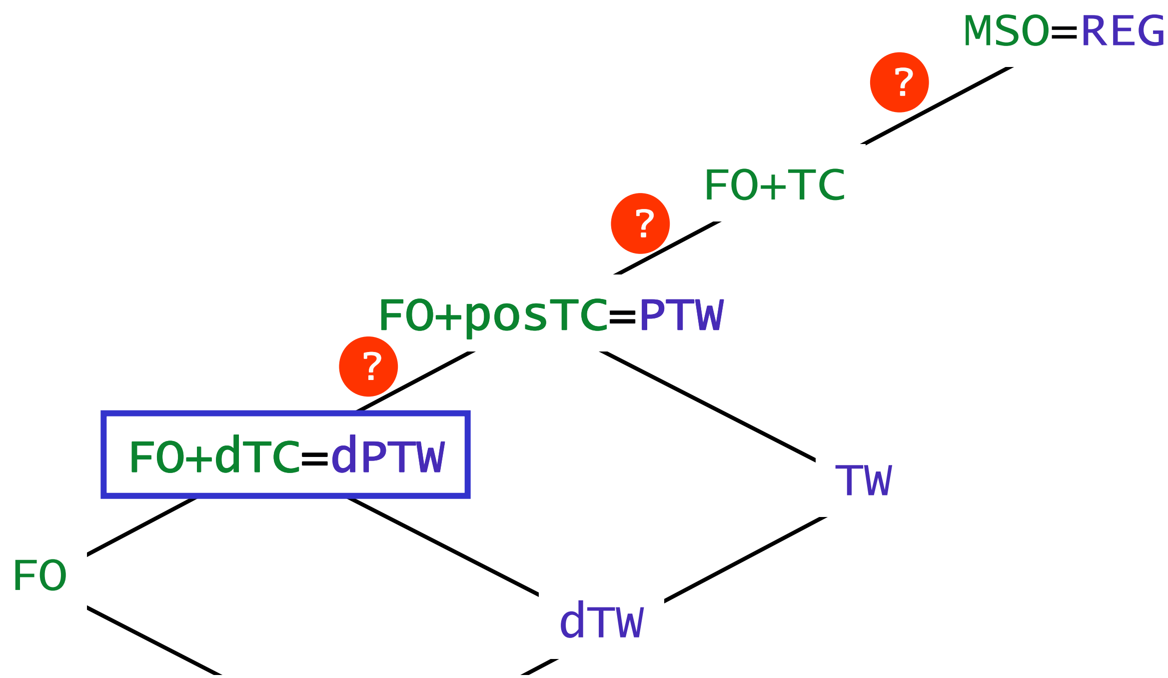
additional instruction
move head to pebble

Cook & Rackoff
'Jumping Automata'

mazes
not all graphs

finally: work to do ...

open for single head on trees:

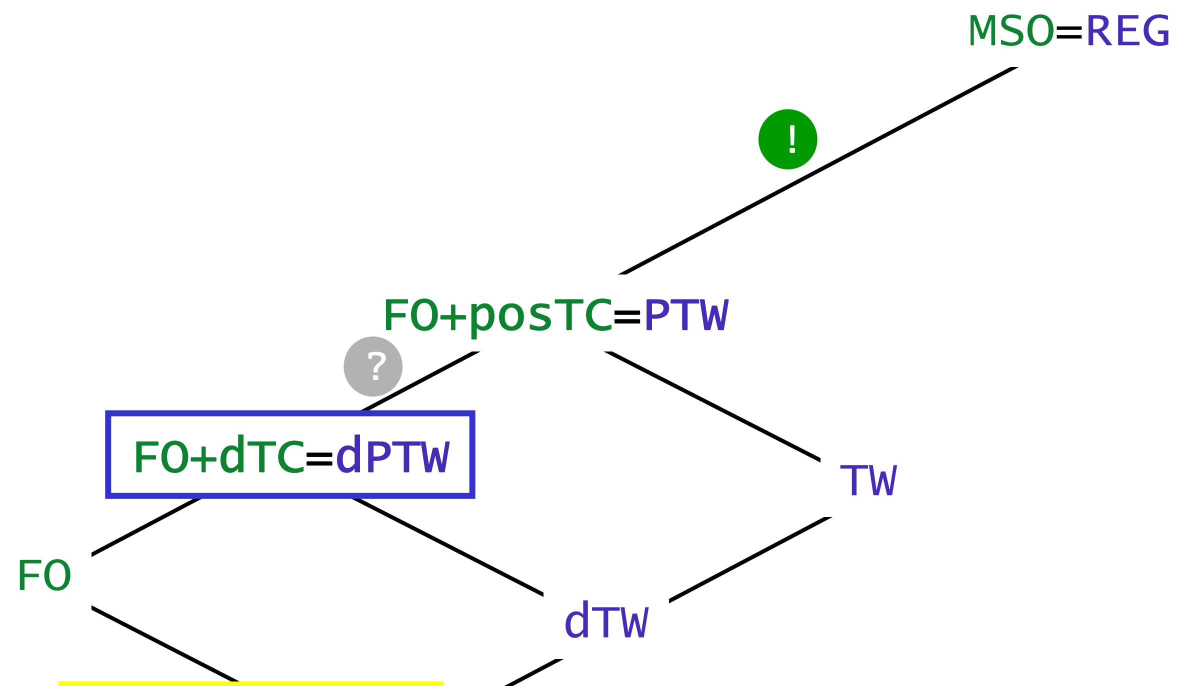


- ? dPTW ⊂ PTW ⊂ REG
- ? FO+dTC ⊂ FO+posTC ⊂ FO+TC ⊂ MSO
- ? pebble hierarchy
- ? type of pebbles strong vs. weak
- ? alternation

finally: work to do …

see ICALP'06

Bojańczyk, Samuelides, Schwentick, Segoufin

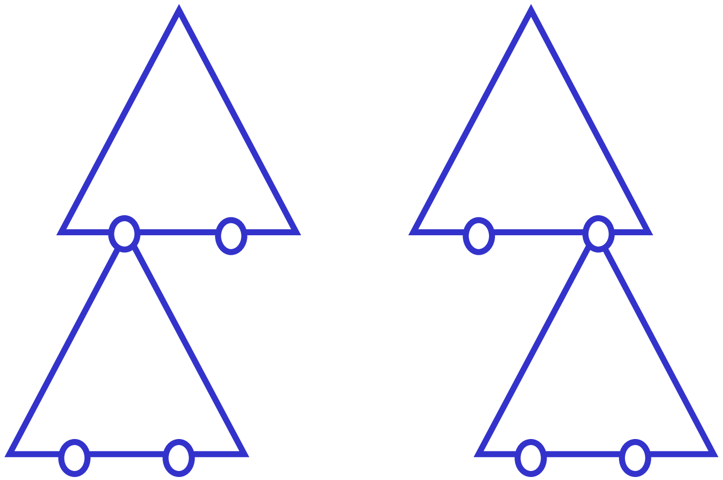


- ! dPTW ⊆ PTW ⊂ REG
- ? FO+dTC ⊆ FO+posTC ⊆ FO+TC ⊆ MSO
- ! pebble hierarchy
- ! type of pebbles physical vs. abstract
- ? alternation

many heads? graphs?

finally: work to do …

because … we forgot about trees



Bojańczyk, Samuelides, Schwentick, Segoufin
ICALP'06 and next talk …

‘tossing Pebbles’

Joost Engelfriet
Hendrik Jan Hoogeboom
Leiden NL

thank you …