Pebbles at WTA

Automata with Nested Pebbles and FO Logic with Transitive Closure

9.6’06 Bonn
bottom-up tree automata

REG
walking along the tree

evaluates and/or trees!

cf. two-way finite state automaton
example: tree traversal

walk along edges, moves based on
- state
- node label
- child number
  (= incoming edge)
example: tree traversal

is right child

from right

ch₂:up

is leaf

lab TF

to left child

lab \lor: down₁

from parent

down₂

is left child

from left

ch₁:up

walk along edges, moves based on

- state
- node label
- child number

(= incoming edge)
tool: systematic tree-traversal
tree-walking automata

Doner; Thatcher & Wright

Monadic Second-Order logic

MSO=REG

(bottom-up) tree automata

First-Order

FO [≤]

LFO [≤]

dTw

(TW)

(deterministic) Tree-Walking automata
‘branching structure’ of even length

Bojańczyk & Colcombet

TW ⊂ REG
‘branching structure’ of even length

Bojańczyk & Colcombet

$TW \subseteq REG$

not by TWA

but $FO(1)$
‘branching structure’ of even length

Bojańczyk & Colcombet

\[ TW \subset REG \]

not by TWA
but FO (!)

\[(aa)^*\]
tree-walking automata

Doner; Thatcher & Wright

Monadic Second-Order logic

MSO=REG

(bottom-up) tree automata

First-Order

FO [≤]

LFO [S]

dTw

(deterministic) Tree-Walking automata

TW
tree-walking automata

PTW

dPTW

FO

LFO

dTw

TW

MSO=REG

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)

\[
\text{PTW}^1 \subseteq \text{REG}
\]
‘branching structure’ of even length

using a pebble

not by TWA
but FO (!)

(aa)*
FO + (deterministic) transitive closure

MSO = REG

FO + TC

FO + posTC = PTW

nested pebbles

FO + dTC = dPTW

FO

dTw

LFO

TC(FO + mod)

Neven & Schwentick
transitive closure

\[ \varphi(u,v) \]

\[ \varphi^*(u,v) \text{ unary tc} \]

\[ \text{deterministic tc: } \varphi \text{ functional} \]

\[ \varphi(u,v,z) \]
- **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, ...
经典结果：[非]确定性对数空间

Immerman

First-Order Logic + transitive closure

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

arity \( k \) \( \Longleftrightarrow \) \( k \) heads

Bargury & Makowsky

Multi-Head Automata (two-way)
fits in our framework
on strings, trees, grids, toruses, mazes, ...

First-Order Logic + transitive closure
$\varphi^*(x,y)$

Multi-Head Automata + ‘nested pebbles’

arity $k$ $\iff$ $k$ heads
(but this is not a talk on trees only)
single head on trees

FO + (deterministic) transitive closure

FO + posTC = PTW

FO + dTC = dPTW

FO

LFO

dTw

TW

FO + TC

MSO = REG

nested pebbles
main result

\[ \textit{FO} + d_{\text{TC}} = d_{\text{PTW}} \]

proof summary
manager style

( deterministic,
single head,
unary tc,
on trees )
(1) logic to nested pebbles

$$\text{lab}_a(x)$$
$$\text{edg}_i(x, y)$$

$$x \leq y$$
$$x = y$$

$$\neg \land \lor \land$$
$$\forall x \ \exists x$$

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

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

always halting
free variables ~ fixed pebbles

$$x \leq y$$
$$\forall x \ \varphi(x) \ \mathcal{A}_\varphi$$
The figure illustrates the $(1_{\text{ctd}})$ transitive closure of a graph. The transitive closure $\varphi^*(u,v)$ is shown explicitly on the left, with functional but implicit $\varphi$ on the right. The process is described as tree walking, implicit $\varphi$-tree, reconstruct locally backwards, and attributed to Sipser.
i single move $\varphi_{pq}(u, v)$

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

free variables for pebbles

ii computation $\sim$ tc with states

Kleene: removing states finite aut to reg expr
(2\text{ctd}) dropping pebbles

\[ \phi^n_{pq}(u,v) = \phi^{(n-1)\#}_{p',q'}(u,v) \]

replacing \(x_n\) by \(u\)
FO + (deterministic) transitive closure

FO + posTC = PTW

FO + dTC = dPTW

MSO = REG

nested pebbles
FO + (deterministic) transitive closure

FO + dTC = dPTW

FO + posTC = PTW

FO+TC

Buchi

MSO = REG

nested pebbles

2-way automata

dTw

LFO

FO

Shepherdson
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
nested pebbles to logic

\[ \text{lab}_a(x), \quad \text{edg}_i(x,y) \]

\[ x \leq y, \quad x = y \]

\[ \neg, \land, \lor, \forall x, \exists x \]

\[ \phi^*(x,y) \]

\[ \text{dPTW}^k \subseteq \text{FO+dTC}^k \]

for families of graphs (i.e. with fixed label alphabets)
\[ \text{FO} + \text{dTC}^k = \text{dPTW}^k \]

for families of \textit{searchable} graphs with a ‘guide’

\textbf{guide: single} head, deterministic with pebbles visits each node (at least) once & halts

\( (\forall x) \ \text{lab}_0(x) \)

unranked trees, grids, toruses, …

2 pebbles
two heads!
(not nested)
searching with many heads

\[
\text{FO} + d\text{TC}^k = d\text{PTW}^k
\]

for families of \textit{k-searchable} graphs

\begin{align*}
\textit{k heads}, & \text{ deterministic} \\
\text{with pebbles} & \\
\text{visits each node (at least) once} & \\
\text{& halts} & \\
\textit{additional instruction} & \\
\text{move head to pebble} &
\end{align*}

Cook & Rackoff

‘Jumping Automata’

not all graphs

mazes
finally: work to do ...

open for single head on trees:

- $dPTW \subset PTW \subset REG$
- $FO+dTC \subset FO+posTC \subset FO+TC \subset MSO$
- pebble hierarchy
- type of pebbles: strong vs. weak
- alternation
finally: work to do ...

Bojańczyk, Samuelides, Schwentick, Segoufin

MSO = REG

FO + posTC = PTW

FO + dTC = dPTW

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 …
thank you …

Joost Engelfriet
Hendrik Jan Hoogeboom
Leiden NL

‘tossing Pebbles’