Pointers and Polynomial Space Functions

Isabel Oitavem

UNL and CMAF

Overview

- Implicit characterizations of classes of computational complexity: a uniform approach.
- Recursion schemes over algebras.
- Implicit characterization of *Pspace*.

Uniform approach

Given a free algebra $\mathbb{A}$ define

$\mathbf{T}_{\mathbb{A}} =$ COMP/REC$_{\mathbb{A}}\{\mathbb{A}$ constructors, destructors, conditional, proj.$\}$

$\mathbf{ST}_{\mathbb{A}} =$ SCOMP/SREC$_{\mathbb{A}}\{\mathbb{A}$ constr, destr., cond., proj. in both tiers$\}$

- $\mathbf{ST}_{\mathbb{N}} =$ *Lspace*
- $\mathbf{ST}_{\mathbb{W}} =$ *Ptime*
- $\mathbf{ST}_{\mathbb{T}} =$ *NC*

Uniform approach — recursion schemes

algebras	constructors	arities
$\mathbb{W}$	$\epsilon, \mathbf{S}_0, \mathbf{S}_1$	0, 1, 1
$\mathbb{T}$	$\mathbf{0}, \mathbf{1}, *$	0, 0, 2

REC$_{\mathbb{W}}$:

$$f(\epsilon, \bar{x}) = g(\epsilon, \bar{x})$$
$$f(\mathbf{S}_0 z, \bar{x}) = h(\mathbf{S}_0 z, \bar{x}, f(z, \bar{x}))$$
$$f(\mathbf{S}_1 z, \bar{x}) = h(\mathbf{S}_1 z, \bar{x}, f(z, \bar{x}))$$

REC$_{\mathbb{T}}$:

$$f(p, \mathbf{0}, \bar{x}) = g(p, \mathbf{0}, \bar{x})$$
$$f(p, \mathbf{1}, \bar{x}) = g(p, \mathbf{1}, \bar{x})$$
$$f(p, u * v, \bar{x}) = h(p, u * v, \bar{x}, f(p * 0, u, \bar{x}), f(p * 1, v, \bar{x}))$$

Recursion schemes — pointers

REC$_{\mathbb{W}}$:

$$f(\epsilon, \bar{x}) = g(\epsilon, \bar{x})$$
$$f(\mathbf{S}_0 z, \bar{x}) = h(\mathbf{S}_0 z, \bar{x}, f(z, \bar{x}))$$
$$f(\mathbf{S}_1 z, \bar{x}) = h(\mathbf{S}_1 z, \bar{x}, f(z, \bar{x}))$$

REC$_{\mathbb{T}}$:

$$f(p, \mathbf{0}, \bar{x}) = g(p, \mathbf{0}, \bar{x})$$
$$f(p, \mathbf{1}, \bar{x}) = g(p, \mathbf{1}, \bar{x})$$
$$f(p, u * v, \bar{x}) = h(p, u * v, \bar{x}, f(p * 0, u, \bar{x}), f(p * 1, v, \bar{x}))$$

REC$_{\mathbb{TW}}$:

$$f(p, \epsilon, \bar{x}) = g(p, \epsilon, \bar{x})$$
$$f(p, \mathbf{S}_0 z, \bar{x}) = h(p, \mathbf{S}_0 z, \bar{x}, f(\mathbf{S}_0 p, z, \bar{x}), f(\mathbf{S}_1 p, z, \bar{x}))$$
$$f(p, \mathbf{S}_1 z, \bar{x}) = h(p, \mathbf{S}_1 z, \bar{x}, f(\mathbf{S}_0 p, z, \bar{x}), f(\mathbf{S}_1 p, z, \bar{x}))$$

Pointers and Pspace

SREC$_{\mathbb{TW}}$:

$$
\begin{aligned}
f(p, \epsilon, \bar{x}; \bar{y}) &= g(p, \epsilon, \bar{x}; \bar{y}) \\
f(p, \mathbf{S}_0 z, \bar{x}; \bar{y}) &= h(p, \mathbf{S}_0 z, \bar{x}; \bar{y}, f(\mathbf{S}_0 p, z, \bar{x}; \bar{y}), f(\mathbf{S}_1 p, z, \bar{x}; \bar{y})) \\
f(p, \mathbf{S}_1 z, \bar{x}; \bar{y}) &= h(p, \mathbf{S}_1 z, \bar{x}; \bar{y}, f(\mathbf{S}_0 p, z, \bar{x}; \bar{y}), f(\mathbf{S}_1 p, z, \bar{x}; \bar{y}))
\end{aligned}
$$

$\mathbf{ST}_{\mathbb{TW}} =$ SCOMP/SREC$_{\mathbb{TW}}$$\{\mathbb{W}$ constr, destr., cond., proj. in both tiers$\}$

$\mathbf{ST}_{\mathbb{TW}} =$ *Pspace*

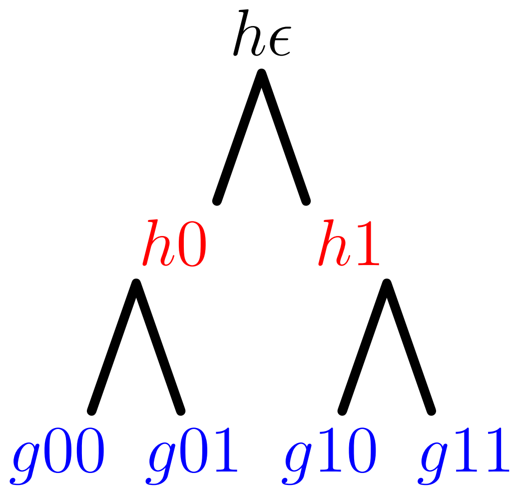
Let f be a function over $\mathbb{W}$. f is computable in polynomial space if, and only if, f is bitwise computable by an ATM in polynomial time, and $|f(w)|$ is polynomial in $|w|$.

Pointers and Pspace

We assume that non-terminating configurations have universal or existential states. To see if a ATM accepts an input x, we define a bottom-up labeling of its computation tree (or part of it) by the following rules: 1) the accepting leaves are labeled 1; 2) any existential node is labeled 1 if at least one of its sons has been labeled 1; 3) any universal node is labeled 1 if all its sons are labeled 1. The machine accepts the input if, and only if, the root is labeled 1.

Pointers and Pspace

Notice that if $f(\epsilon, 11;)$ is defined by recursion (with pointers) on its second input based on g and h, then one has the term $h(\epsilon; h(0; g(00;), g(01;)), h(1; g(10;), g(11;)))$ (some inputs are omitted), which corresponds to the tree



and it is suitable to carry out the bottom-up labeling described above (assuming that non-terminating configurations have two successor configurations).