Higher-order schemes and morphic words

Laurent Braud

Irisa, Rennes

August 30, 2006

Morphic words

Let be $\tau, \sigma, \Sigma, a \in \Sigma$ such as

- $\tau(a) = au$

Morphic words

Let be $\tau, \sigma, \Sigma, a \in \Sigma$ such as

- $\tau(a) = au$

$$
\begin{array}{rcl}
\tau^0(a) & = & a \\
\tau(a) & = & au \\
\tau^2(a) & = & au\tau(u) \\
\tau^3(a) & = & au\tau(u)\tau^2(u) \\
 & \cdots & \\
\tau^\omega(a) & = & au\tau(u)\tau^2(u)\tau^3(u)\ldots
\end{array}
$$

Morphic words

Let be $\tau, \sigma, \Sigma, a \in \Sigma$ such as

- $\tau(a) = au$

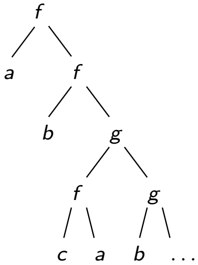
$$\begin{array}{rcl} \tau^0(a) & = & a \\ \tau(a) & = & au \\ \tau^2(a) & = & au\tau(u) \\ \tau^3(a) & = & au\tau(u)\tau^2(u) \\ & \cdots & \\ \tau^\omega(a) & = & au\tau(u)\tau^2(u)\tau^3(u)\dots \end{array}$$

- σ renames the result.

The infinite words $\sigma(\tau^\omega(a))$ are the *morphic words*.

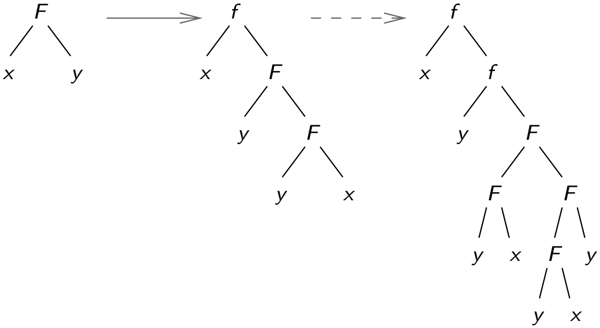
Yielding / Harvesting

A deterministic tree may be *harvested* by picking leaves from left to right, if possible.



Schemes

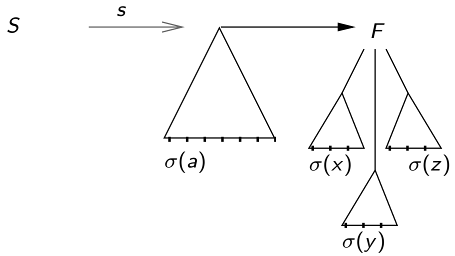
A (first-order) *scheme* is a deterministic term grammar.

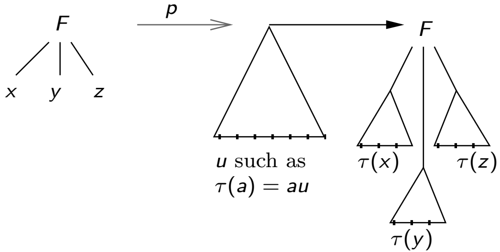


First result

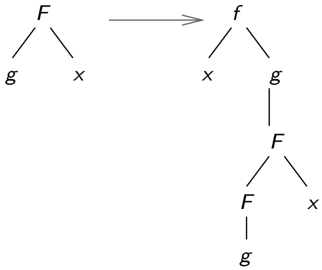
Proposition

Schemes yield exactly morphic words.

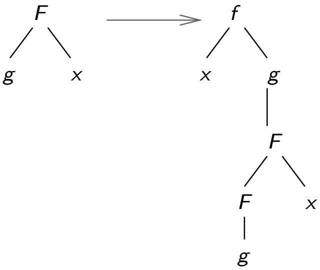

S
s
F
σ(a)
σ(x)
σ(z)
σ(y)


F
x y z
p
F
u such as
τ(a) = au
τ(x)
τ(z)
τ(y)

Schemes at second level



Schemes at second level

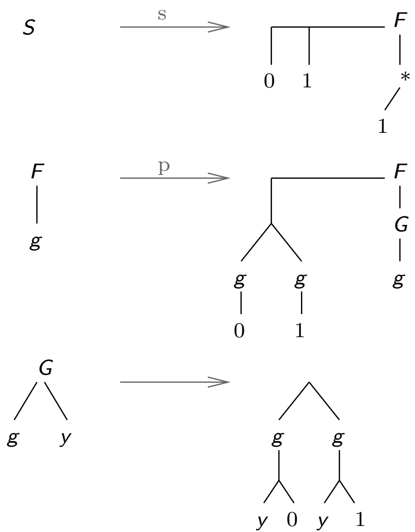


$$
\begin{array}{lcl}
x & : & o \\
g & : & o \rightarrow o \\
f & : & o \rightarrow o \rightarrow o \\
F & : & (o \rightarrow o) \rightarrow o \rightarrow o
\end{array}
$$

The Champernowne word

This word is obtained by enumerating numbers in order and concatenating them. Respectively in decimal an binary, it begins by

$$
\begin{array}{l}
01234567891011121314\ldots \\
0110111001011101111000\ldots
\end{array}
$$



$$
\begin{array}{rcl}
0, 1 & : & o \\
g & : & o \to o \\
* & : & o \to o \to o \\
G & : & (o \to o) \to o \to o \\
F & : & (o \to o) \to o
\end{array}
$$

Introducing second-order morphic words

A *term word* (tw) is a finite sequence of terms.
There are still

- τ, σ, **term** morphisms,
- letters Σ_0, but also functions Σ_1,
- And a root letter $a \in \Sigma_0$.

Introducing second-order morphic words

A *term word* (tw) is a finite sequence of terms.
There are still

- τ, σ, **term** morphisms,
- letters Σ_0, but also functions Σ_1,
- And a root letter $a \in \Sigma_0$.

$$
\begin{array}{lrcll}
 & \tau(a) & = & au & \text{où } u \text{ tw of } \Sigma_0 \cup \Sigma_1 \\
b \in \Sigma_0 : & \tau(b) & = & u_b & \text{tw of } \Sigma_0 \cup \Sigma_1 \\
b \in \Sigma_1 : & \tau(b(z_1, \ldots, z_n)) & = & u_b & \text{tw of } \Sigma_0 \cup \Sigma_1 \cup \{z_1, \ldots, z_n\} \\
\hline
b \in \Sigma_0 : & \sigma(b) & = & v_b & \in \Sigma_0^* \\
b \in \Sigma_1 : & \sigma(b(z_1, \ldots, z_n)) & = & v_b & \in (\Sigma_0 \cup \{z_1, \ldots, z_n\})^*
\end{array}
$$

$$
\begin{array}{rcl}
\tau(b(t_1, \ldots, t_n)) & = & u_b[\tau(t_i)/z_i] \\
\sigma(b(t_1, \ldots, t_n)) & = & u_b[\sigma(t_i)/z_i]
\end{array}
$$

Champernowne is a second-order morphic word

$\Sigma_0 = \{a, 0, 1\}, \Sigma_1 = \{g^1\}$:

$$
\begin{array}{rrcl}
\tau : & a & \mapsto & ag(0)g(1) \\
 & g(z) & \mapsto & g(z0)g(z1) \\
\sigma : & a & \mapsto & 01 \\
 & g(z) & \mapsto & 1z
\end{array}
$$

Champernowne is a second-order morphic word

$\Sigma_0 = \{a, 0, 1\}, \Sigma_1 = \{g^1\}$:

$$
\begin{array}{rrcl}
\tau : & a & \mapsto & ag(0)g(1) \\
 & g(z) & \mapsto & g(z0)g(z1) \\
\sigma : & a & \mapsto & 01 \\
 & g(z) & \mapsto & 1z
\end{array}
$$

$$\tau^2(a) = \overbrace{a \quad g(0)g(1)}^{\tau(a)}\overbrace{g(00)g(01)}^{\tau(g(0))}\overbrace{g(10)g(11)}^{\tau(g(1))}$$

Champernowne is a second-order morphic word

$\Sigma_0 = \{a, 0, 1\}, \Sigma_1 = \{g^1\}$:

$$\begin{array}{rrcl} \tau : & a & \mapsto & ag(0)g(1) \\ & g(z) & \mapsto & g(z0)g(z1) \\ \sigma : & a & \mapsto & 01 \\ & g(z) & \mapsto & 1z \end{array}$$

$$\begin{array}{rcl} \tau^2(a) & = & \overbrace{a \quad g(0)g(1)}^{\tau(a)}\overbrace{g(00)g(01)}^{\tau(g(0))}\overbrace{g(10)g(11)}^{\tau(g(1))} \\ \sigma(\tau^2(a)) & = & 01 \quad 10 \quad 11 \quad 100 \quad 101 \quad 110 \quad 111 \end{array}$$

As wished,

Proposition

Second-order schemes yield exactly second-order morphic words.

Widening the background

The Caucal hierarchy :

Widening the background

The Caucal hierarchy :

- monadic interpretations

$$I(G) = \{p \xrightarrow{a} q | \phi_a(p, q)\}$$

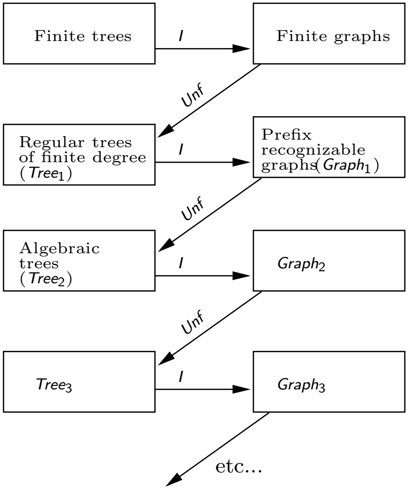
Widening the background

The Caucal hierarchy :

- monadic interpretations

$$I(G) = \{p \xrightarrow{a} q | \phi_a(p, q)\}$$

- unfoldings


Finite trees
I
Finite graphs
Unf
Regular trees of finite degree (Tree1)
I
Prefix recognizable graphs(Graph1)
Unf
Algebraic trees (Tree2)
I
Graph2
Unf
Tree3
I
Graph3
etc...

Word graphs

Proposition

Morphic words belong to $Graph_2$.
Second-order morphic words belong to $Graph_3$.