Morphic words and recursion schemes

Laurent BRAUD

FREC meeting, LABRI

6 dec 2011

Context

Graphs and trees with decidable MSO-theory :

- pushdown/Caucal hierarchy
- recursion schemes : [Damm], and recently [Knapik-Niwiński-Urzyczyn], [Ong]
- infinite words : ult. periodic, morphic [Carton-Thomas]

Context

Graphs and trees with decidable MSO-theory :

- pushdown/Caucal hierarchy
- recursion schemes : [Damm], and recently [Knapik-Niwiński-Urzyczyn], [Ong]
- infinite words : ult. periodic, morphic [Carton-Thomas]

This talk :

1. relationship between order-1 schemes and morphic words
2. extension to order 2

Recursion schemes : first order

Term grammar with

- terminals $T = \{a, b, f, g \dots\}$,
- nonterminals $N = \{S, F, G \dots\}$,
- a specific starting nonterminal S
- one rewriting rule per nonterminal, using variables $\mathcal{X} = \{x, y \dots\}$.

Every symbol α has fixed arity $\rho(\alpha)$.

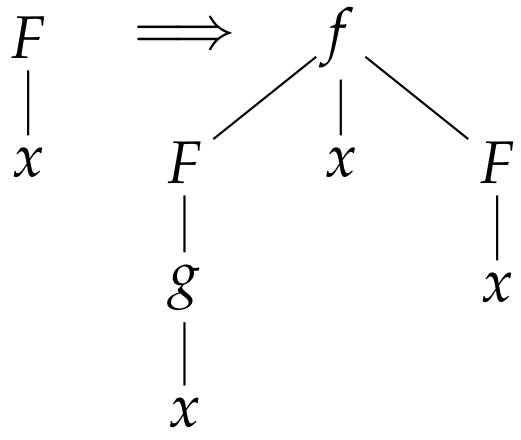
Recursion schemes : first order

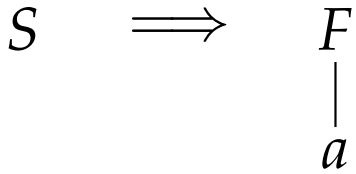
Term grammar with

- terminals $T = \{a, b, f, g \dots\}$,
- nonterminals $N = \{S, F, G \dots\}$,
- a specific starting nonterminal S
- one rewriting rule per nonterminal, using variables $\mathcal{X} = \{x, y \dots\}$.

Every symbol α has fixed arity $\rho(\alpha)$.

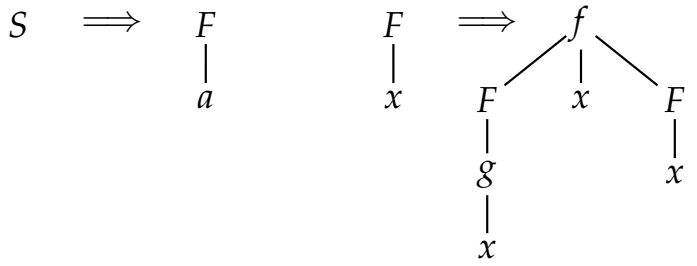
$$T = \{f, g, a\}$$
$$N = \{S, F\}$$



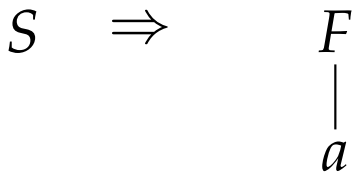


Here $\rho(f) = 3, \rho(F) = \rho(g) = 1, \rho(a) = 0$.

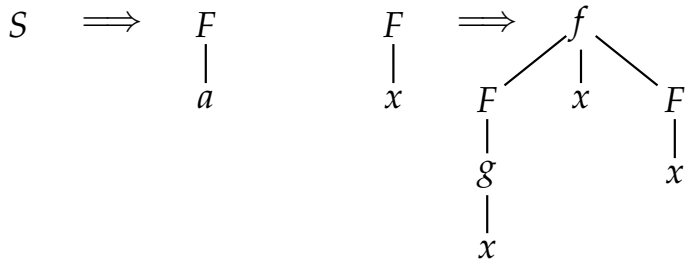
Recursion schemes : first order



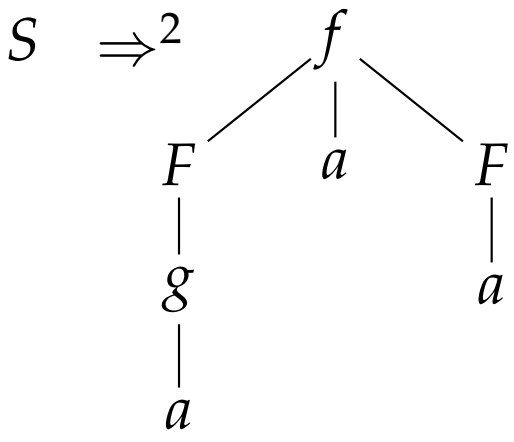
A scheme builds a (possibly) infinite tree of terminals.



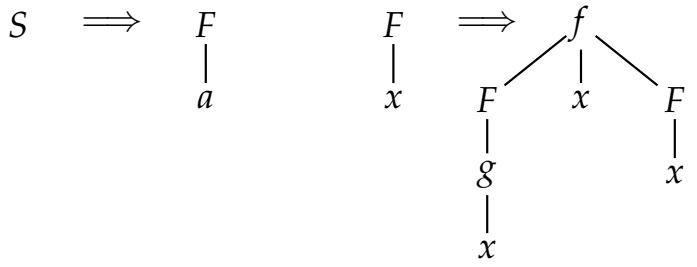
Recursion schemes : first order



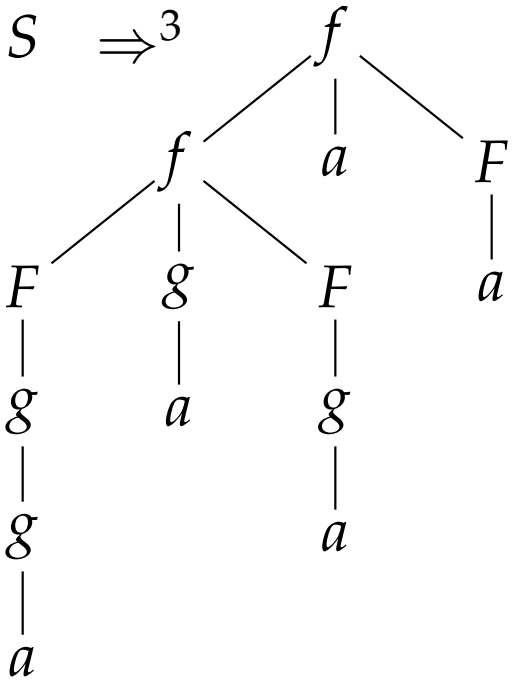
A scheme builds a (possibly) infinite tree of terminals.



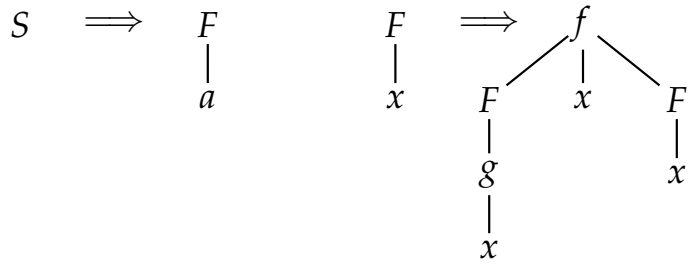
Recursion schemes : first order



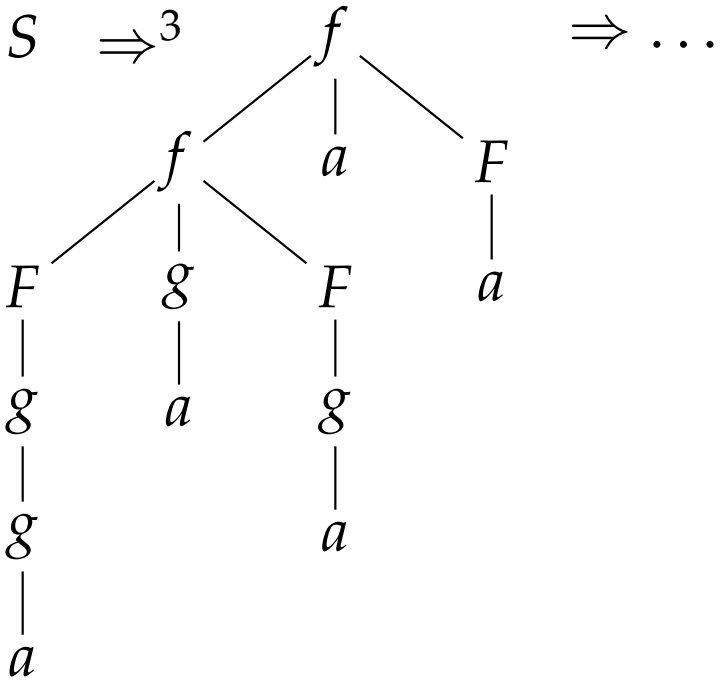
A scheme builds a (possibly) infinite tree of terminals.



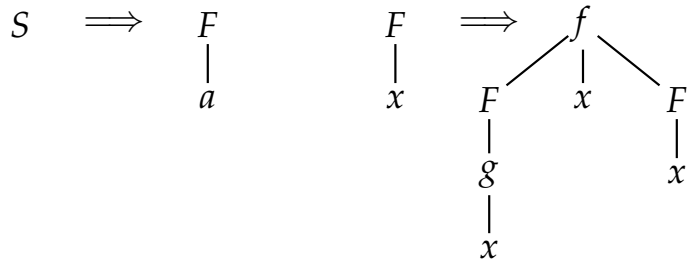
Recursion schemes : first order



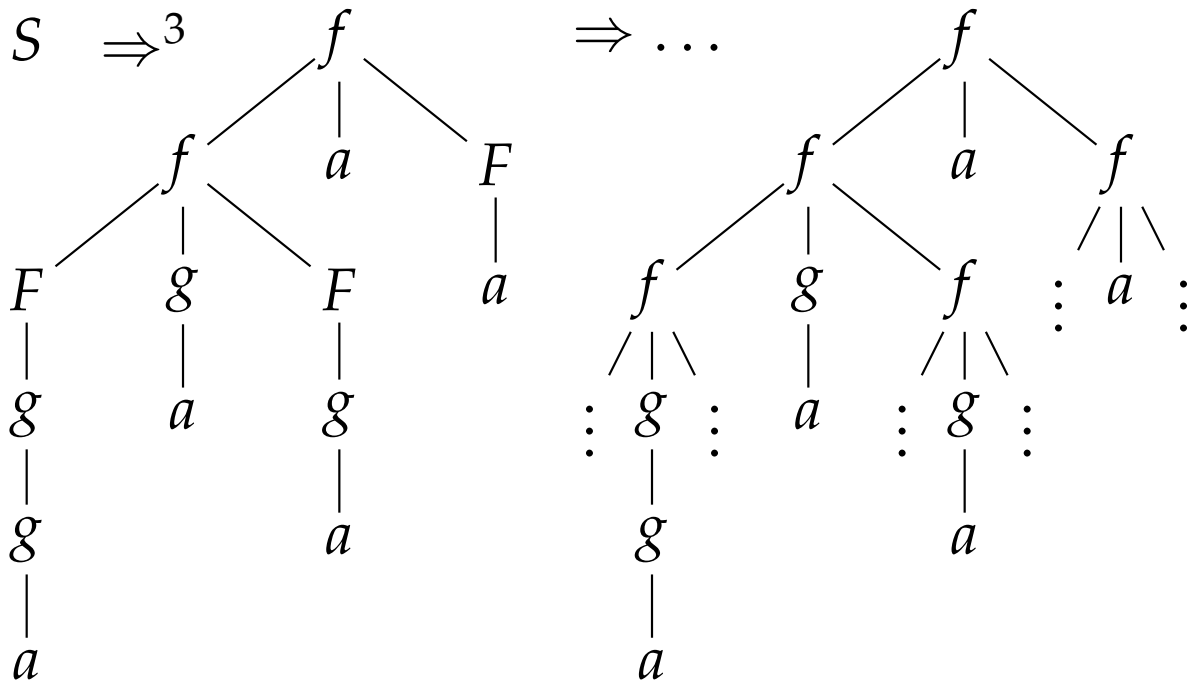
A scheme builds a (possibly) infinite tree of terminals.



Recursion schemes : first order

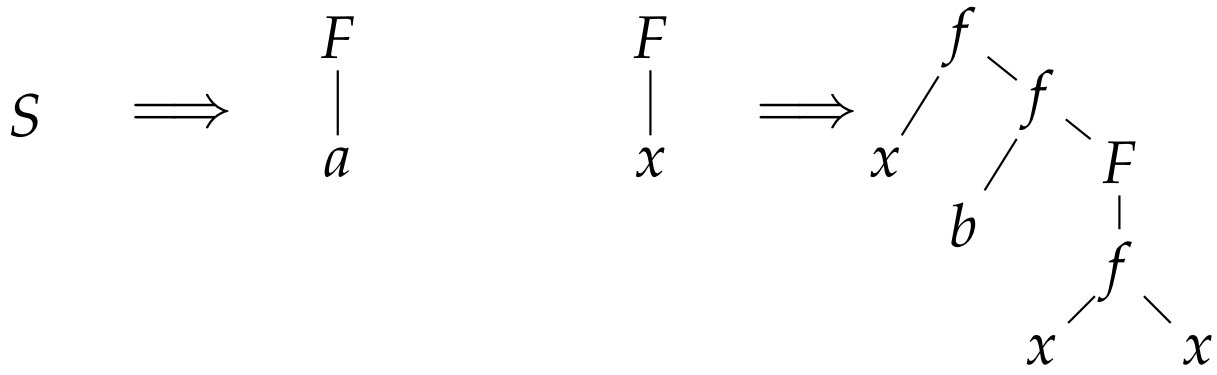


A scheme builds a (possibly) infinite tree of terminals.

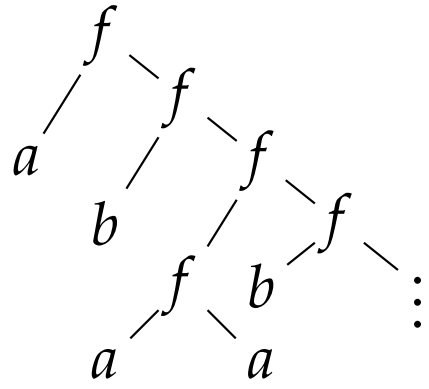


Frontiers of limit trees

We are interested in infinite words that appear in schemes.

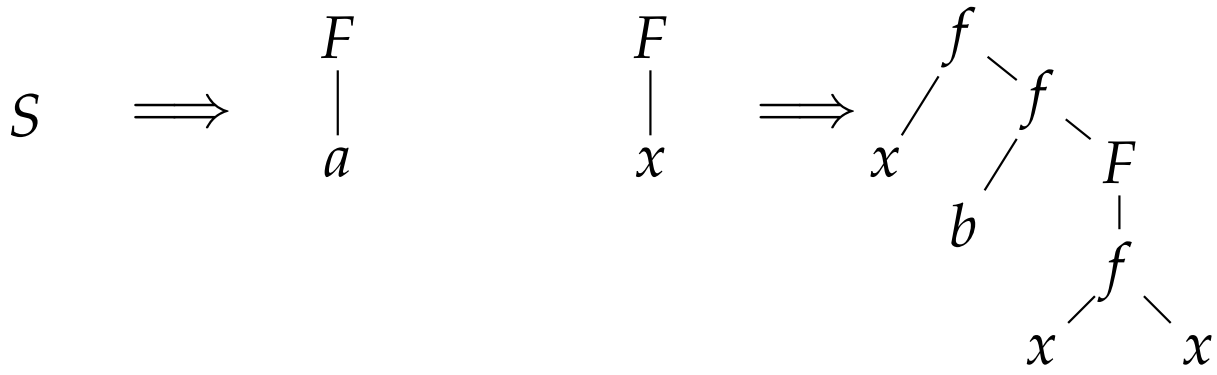


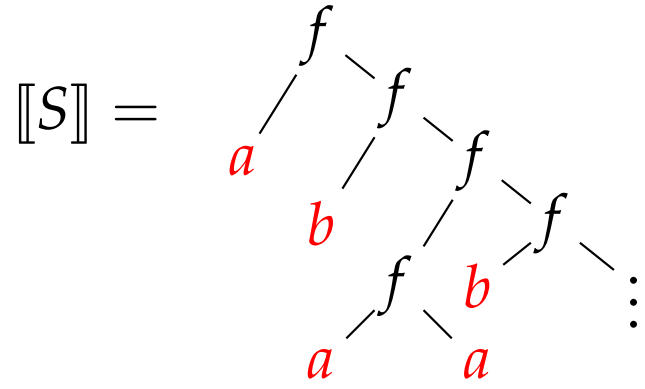
$[\![S]\!] =$



Frontiers of limit trees

We are interested in infinite words that appear in schemes.





$$\mathbf{Fr}(\llbracket S \rrbracket) = abaab \ldots a^{2^i} b \ldots$$

Frontiers of limit trees

Let T be an infinite term and let the *frontier* $\mathbf{Fr}(T)$ be the colored order of leaves in left-right order.

Frontiers of limit trees

Let T be an infinite term and let the *frontier* $\mathbf{Fr}(T)$ be the colored order of leaves in left-right order.
The *ω-frontier* ω-**Fr**(T) is the initial part of $\mathbf{Fr}(T)$ of type ω, when it exists.

Frontiers of limit trees

Let T be an infinite term and let the *frontier* $\mathbf{Fr}(T)$ be the colored order of leaves in left-right order.
The *ω-frontier* ω-$\mathbf{Fr}$(T) is the initial part of $\mathbf{Fr}(T)$ of type ω, when it exists.

Proposition

For any tree generated by an order-1 scheme, there is a tree generated by a order-1 scheme where the rightmost branch is the only infinite branch, and with the same ω-frontier.

The trees with one infinite rightmost branch are called *combs*.

Morphic words

Σ is an alphabet. A *morphism* τ on Σ^* is such that

$$\tau(ab) = \tau(a)\tau(b).$$

Morphic words

Σ is an alphabet. A *morphism* τ on Σ^* is such that

$$\tau(ab) = \tau(a)\tau(b).$$

Let τ be a morphism on Σ^* s.t. there is $a \in \Sigma$ with $\tau(a) \in a\Sigma$.

$$\begin{aligned} \tau(a) &= au \\ \tau^2(a) &= au\tau(u) \\ \tau^3(a) &= au\tau(u)\tau^2(u) \\ &\dots \\ \tau^\omega(a) &= au\tau(u)\dots \end{aligned}$$

Words $\sigma(\tau^\omega(a))$ are *morphic words*, where σ is another morphism.

Morphic words : example

$$\begin{aligned} \tau(a) &= abcc \\ \tau(b) &= b \\ \tau(c) &= cc \end{aligned} \qquad \begin{aligned} \sigma(a) &= a \\ \sigma(b) &= b \\ \sigma(c) &= a \end{aligned}$$

Morphic words : example

$$\begin{array}{lcl} \tau(a) & = & abcc \\ \tau(b) & = & b \\ \tau(c) & = & cc \end{array} \qquad \begin{array}{lcl} \sigma(a) & = & a \\ \sigma(b) & = & b \\ \sigma(c) & = & a \end{array}$$

$$\tau(a) = abcc$$

Morphic words : example

$$\begin{array}{lcl} \tau(a) & = & abcc \\ \tau(b) & = & b \\ \tau(c) & = & cc \end{array} \qquad \begin{array}{lcl} \sigma(a) & = & a \\ \sigma(b) & = & b \\ \sigma(c) & = & a \end{array}$$

$$\begin{array}{rcl} \tau(a) & = & abcc \\ \tau^2(a) & = & abccbcccc \\ & & \dots \end{array}$$

Morphic words : example

$$\begin{array}{lcl} \tau(a) & = & abcc \\ \tau(b) & = & b \\ \tau(c) & = & cc \end{array} \qquad \begin{array}{lcl} \sigma(a) & = & a \\ \sigma(b) & = & b \\ \sigma(c) & = & a \end{array}$$

$$\begin{array}{rcl} \tau(a) & = & abcc \\ \tau^2(a) & = & abccbcccc \\ & & \dots \\ \tau^\omega(a) & = & abccb \dots c^{2^i} b \dots \end{array}$$

Morphic words : example

$$\begin{array}{lcl} \tau(a) & = & abcc \\ \tau(b) & = & b \\ \tau(c) & = & cc \end{array} \qquad \begin{array}{lcl} \sigma(a) & = & a \\ \sigma(b) & = & b \\ \sigma(c) & = & a \end{array}$$

$$\begin{array}{rcl} \tau(a) & = & abcc \\ \tau^2(a) & = & abccbcccc \\ & & \dots \\ \tau^\omega(a) & = & abccb \dots c^{2^i} b \dots \\ \sigma(\tau^\omega(a)) & = & abaab \dots a^{2^i} b \dots \end{array}$$

First result

Theorem

ω-frontiers of limit trees of (order-1) recursion schemes are exactly morphic words.

First result

Theorem

ω-frontiers of limit trees of (order-1) recursion schemes are exactly morphic words.

A nonterminal F has a useless parameter index i when x_i does not appear in $[\![F\overrightarrow{x}]\!]$.

Lemma (usefulness)

For any order-1 scheme, there is an order-1 scheme generating the same tree and where every nonterminal has only useful parameters.

First result

Theorem

ω-frontiers of limit trees of (order-1) recursion schemes are exactly morphic words.

A nonterminal F has a useless parameter index i when x_i does not appear in $[\![F\overrightarrow{x}]\!]$.

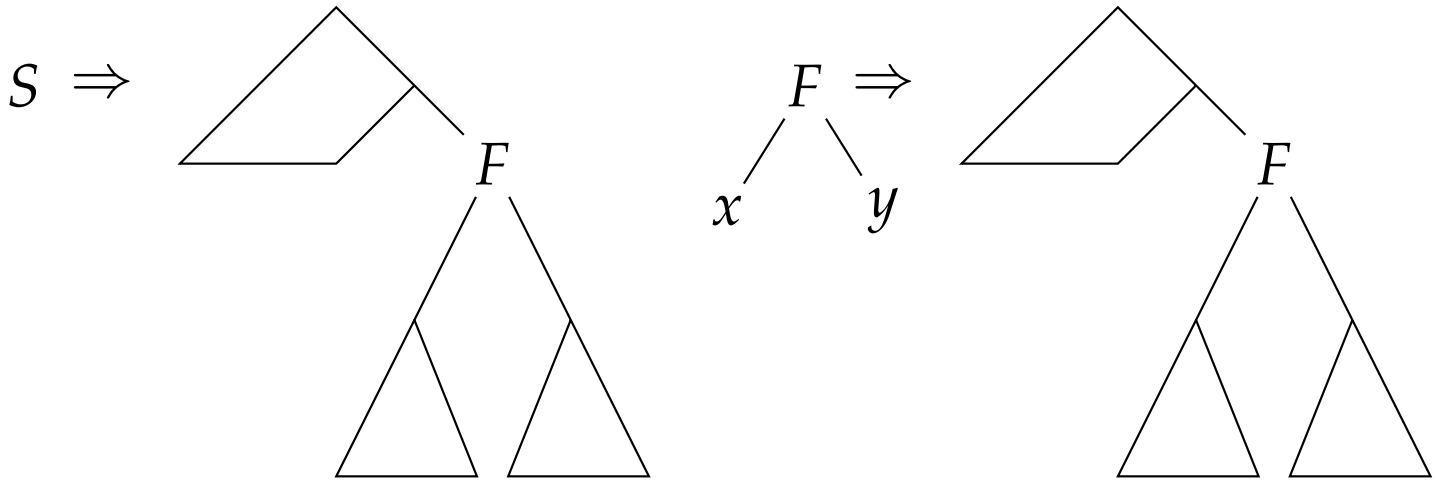
Lemma (usefulness)

For any order-1 scheme, there is an order-1 scheme generating the same tree and where every nonterminal has only useful parameters.

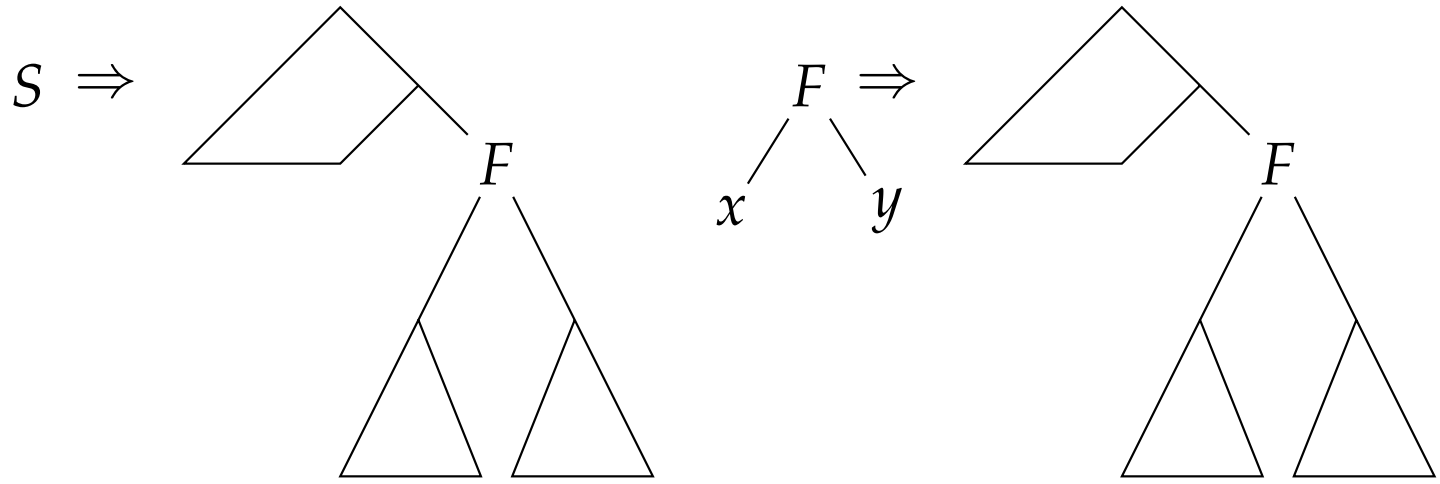
Lemma (linearization)

For any order-1 scheme $\mathfrak{S}$ generating a comb, there is $\mathfrak{S}'$ with only two nonterminals $\{S, R\}$ such that $\mathbf{Fr}(\mathfrak{S}) = \mathbf{Fr}(\mathfrak{S}')$. Moreover, each rewriting rule has exactly one occurence of R and none of S.

Proof sketch

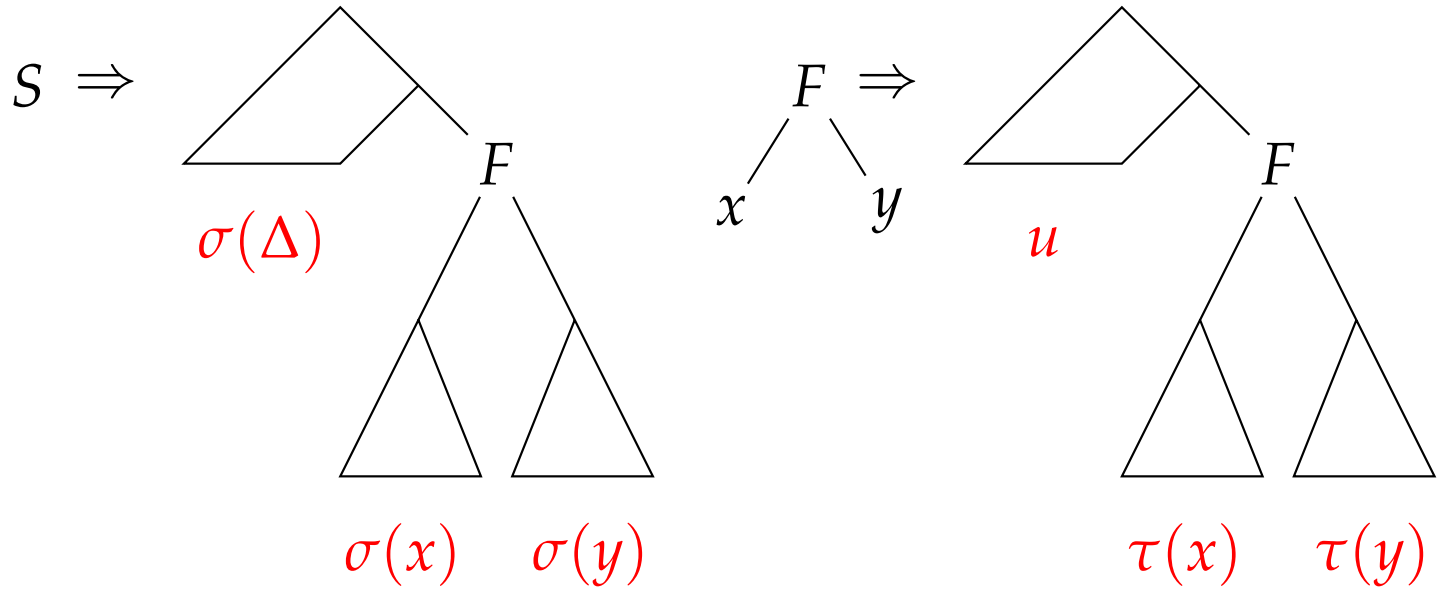


Proof sketch



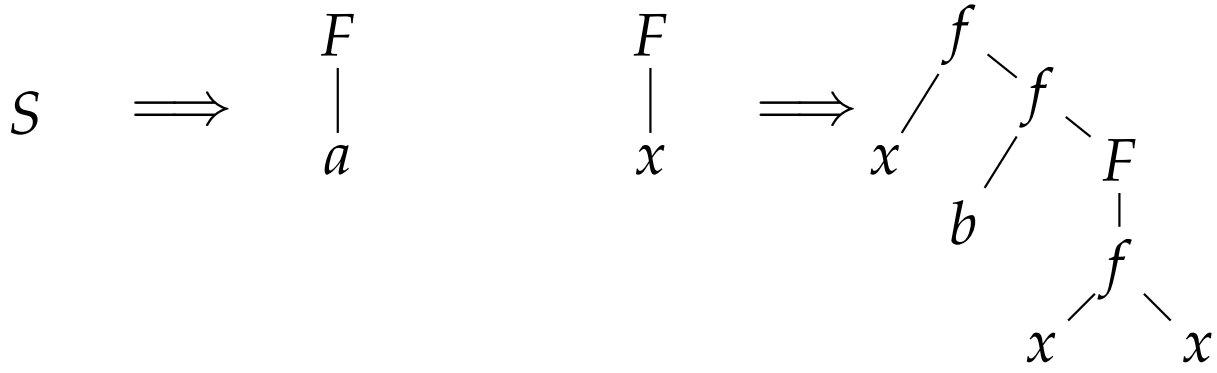
- Letters : $\{c \in T \mid \rho(c) = 0\} \cup \{x, y, \Delta\}$
- $\tau(c) = c$ for all $c \in T$,
- Δ is the root : $\tau(\Delta) = \Delta u$

Proof sketch



- Letters : $\{c \in T \mid \rho(c) = 0\} \cup \{x, y, \Delta\}$
- $\tau(c) = c$ for all $c \in T$,
- Δ is the root : $\tau(\Delta) = \Delta u$

Proof sketch



Letters : $\{\Delta, a, b, x\}$

$$\begin{aligned} \sigma(\Delta) &= \varepsilon & \qquad \tau(\Delta) &= \Delta x b \\ \sigma(a) &= a & \qquad \tau(a) &= a \\ \sigma(b) &= b & \qquad \tau(b) &= b \\ \sigma(x) &= a & \qquad \tau(x) &= xx \end{aligned}$$

Towards next-order morphic words

Can we expect to

- increase subword complexity?

Theorem (Allouche-Shallit)

The number of words of length n in a morphic word is at most $\mathcal{O}(n^2)$.

- increase growth rate?

Theorem (Carton-Thomas)

The sequence of indexes of a given letter in a morphic word is at most $\mathcal{O}(k^n)$ for some k.

Recursion schemes : next order

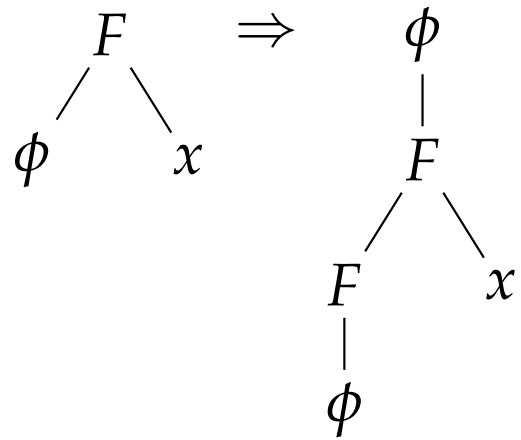
Instead of simply arity, symbols have fixed type (starting with a *base type* **o** :

- terminals $T = \{a, b, f, g \dots\}$,
- nonterminals $N = \{S, F, G \dots\}$,
- a specific starting nonterminal S
- rewriting rules for each nonterminal, using variables $\mathcal{X} = \{x, y, \phi, \psi \dots\}$.

Recursion schemes : next order

Instead of simply arity, symbols have fixed type (starting with a *base type* $\mathbf{o}$:

- terminals $T = \{a, b, f, g \dots\}$,
- nonterminals $N = \{S, F, G \dots\}$,
- a specific starting nonterminal S
- rewriting rules for each nonterminal, using variables $\mathcal{X} = \{x, y, \phi, \psi \dots\}$.

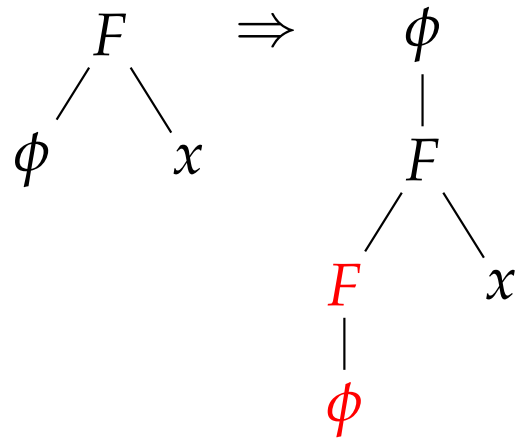


$$x \;:\; \mathbf{o} \qquad\qquad F \;:\; (\mathbf{o} \to \mathbf{o}) \to \mathbf{o} \to \mathbf{o}$$
$$\phi \;:\; \mathbf{o} \to \mathbf{o}$$

Recursion schemes : next order

Instead of simply arity, symbols have fixed type (starting with a *base type* $\mathbf{o}$:

- terminals $T = \{a, b, f, g \dots\}$,
- nonterminals $N = \{S, F, G \dots\}$,
- a specific starting nonterminal S
- rewriting rules for each nonterminal, using variables $\mathcal{X} = \{x, y, \phi, \psi \dots\}$.



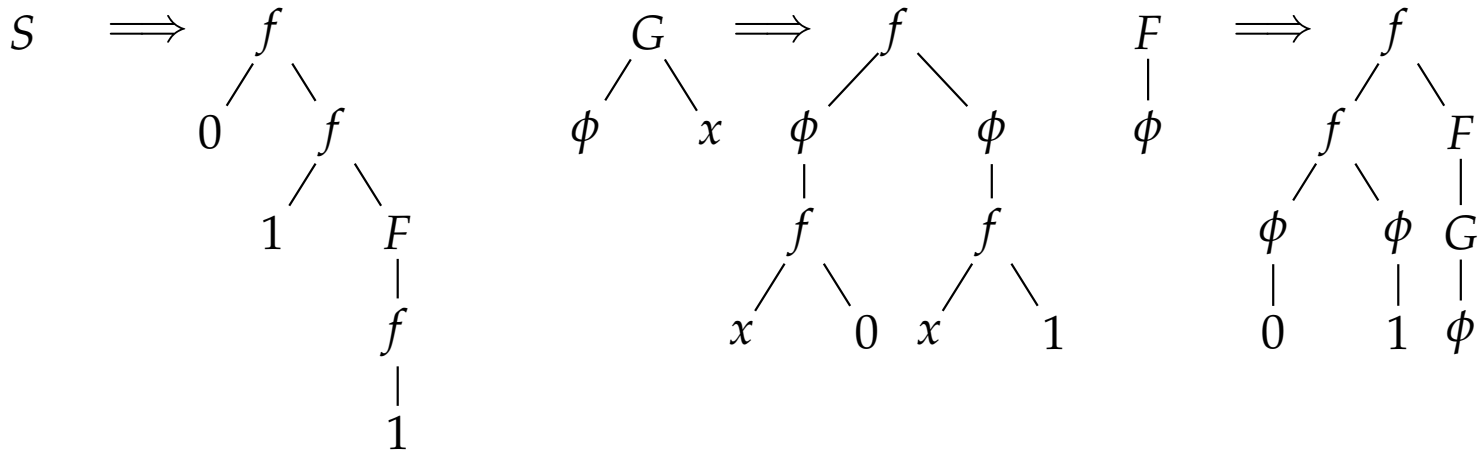
$$x : \mathbf{o} \qquad F : (\mathbf{o} \to \mathbf{o}) \to \mathbf{o} \to \mathbf{o}$$
$$\phi : \mathbf{o} \to \mathbf{o} \qquad F\phi : \mathbf{o} \to \mathbf{o}$$

Champernowne

The *Champernowne's constant* is simply the concatenation of numbers.

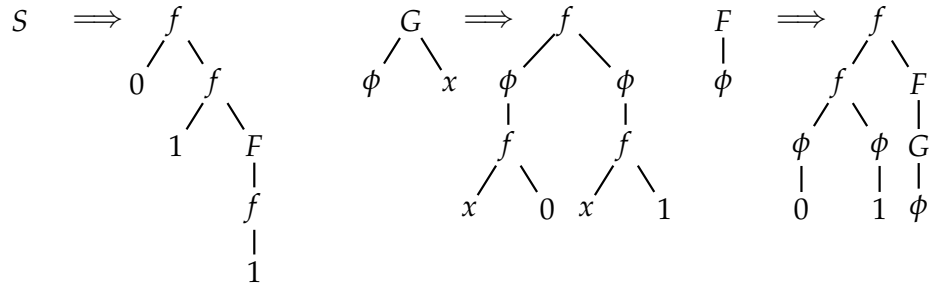
$$0123456778910 11\ldots$$
$$0110111001011 10\ldots$$

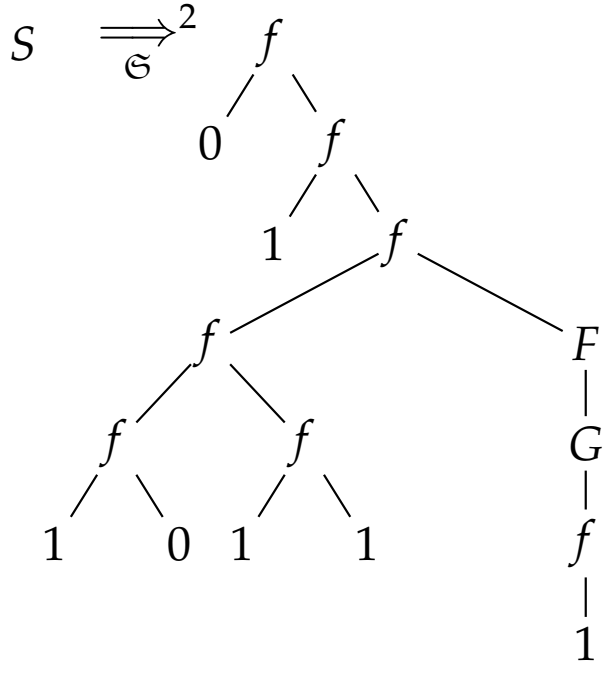
Champernowne : scheme approach



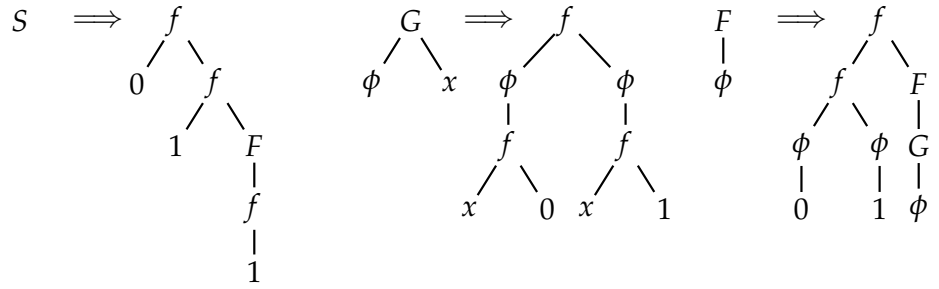
$$
\begin{array}{lcl}
S & : & \mathbf{o} \\
F & : & (\mathbf{o} \to \mathbf{o}) \to \mathbf{o} \\
G & : & (\mathbf{o} \to \mathbf{o}) \to \mathbf{o} \to \mathbf{o}
\end{array}
\qquad
\begin{array}{lcl}
f & : & \mathbf{o} \to \mathbf{o} \to \mathbf{o} \\
0 & : & \mathbf{o} \\
1 & : & \mathbf{o}
\end{array}
$$

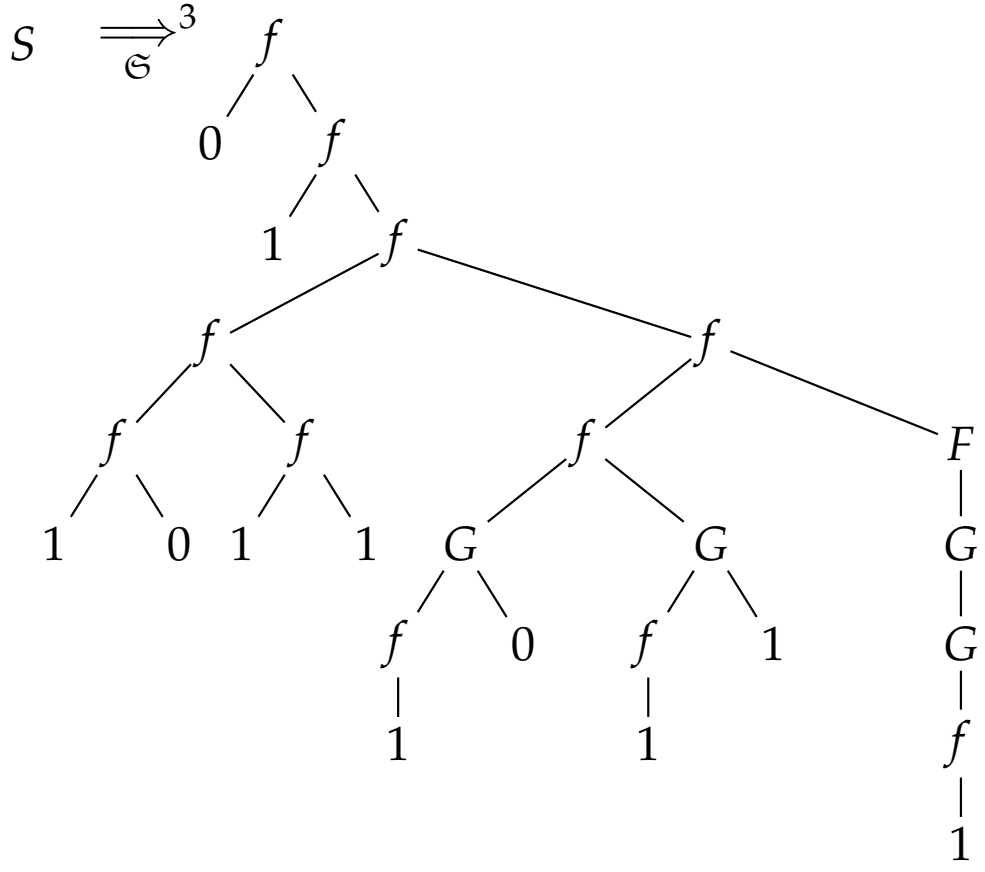
Champernowne : scheme approach



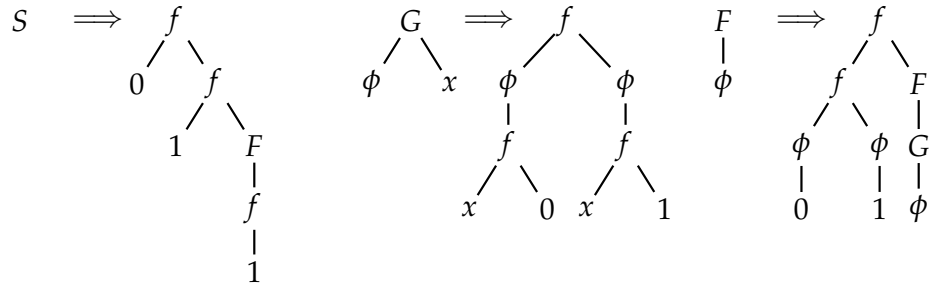


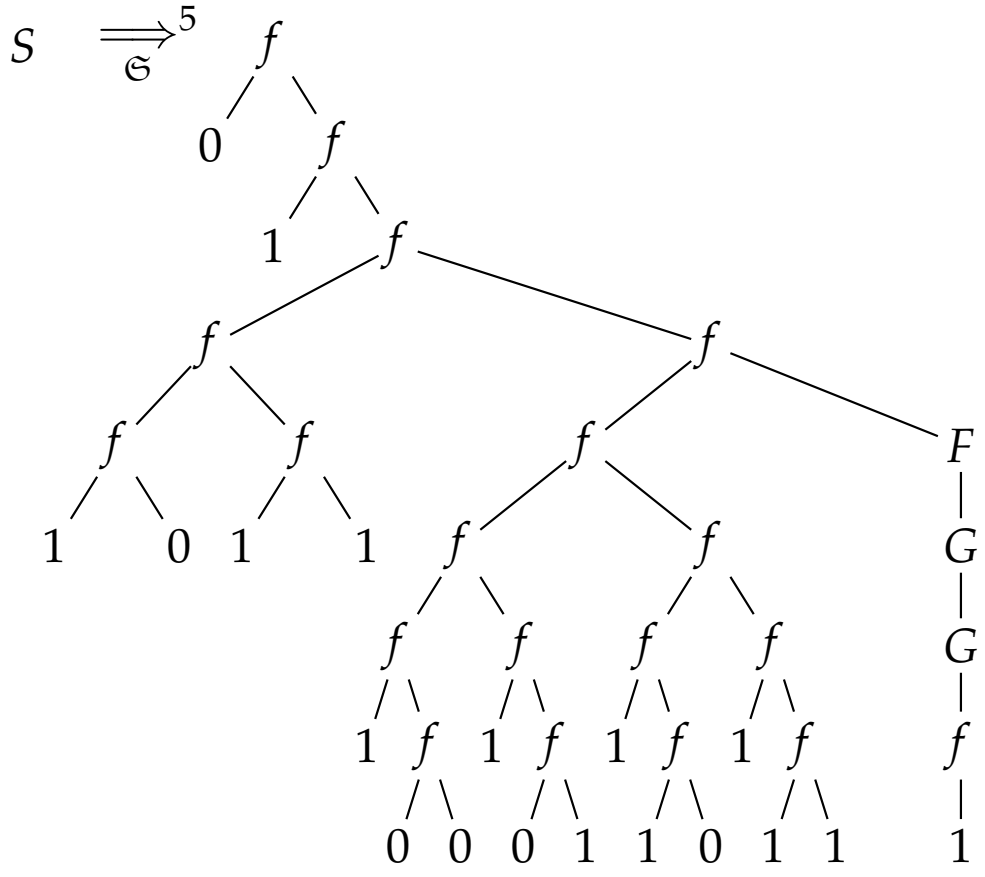
Champernowne : scheme approach





Champernowne : scheme approach





Linearization of order-2 schemes

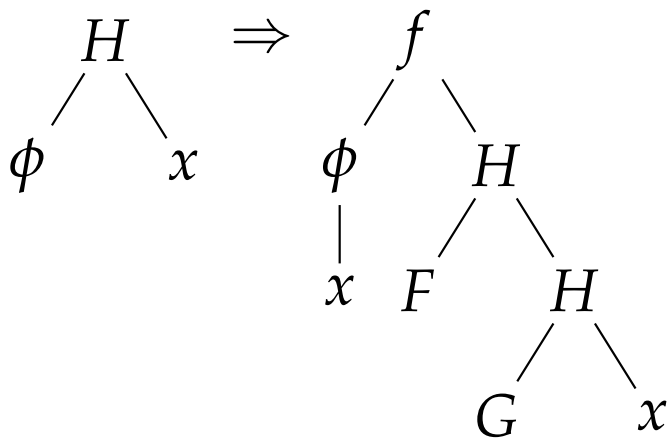
Can we have the same linearization lemma as before?

Linearization of order-2 schemes

Can we have the same linearization lemma as before?

Lemma

For any scheme in $\mathcal{S}_2$, there is a scheme in $\mathcal{S}_2$ with only useful nonterminals generating the same tree.

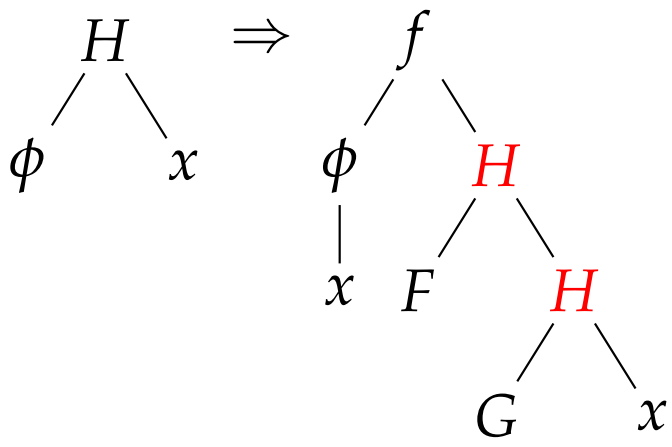


Linearization of order-2 schemes

Can we have the same linearization lemma as before?

Lemma

For any scheme in $\mathcal{S}_2$, there is a scheme in $\mathcal{S}_2$ with only useful nonterminals generating the same tree.



$F : \mathbf{o} \to \mathbf{o}, \quad G : \mathbf{o} \to \mathbf{o}$

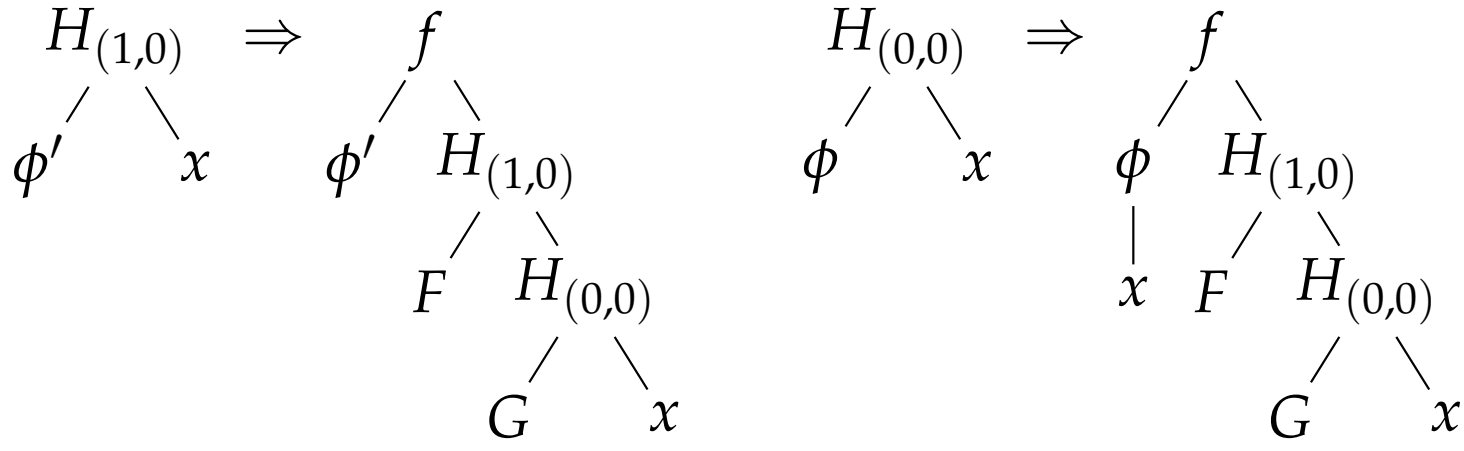
$\rightsquigarrow F : \mathbf{o}$

Linearization of order-2 schemes

Can we have the same linearization lemma as before?

Lemma

For any scheme in $\mathcal{S}_2$, there is a scheme in $\mathcal{S}_2$ with only useful nonterminals generating the same tree.



($H_{\overrightarrow{a}}$ means "argument i has arity recuded by a_i)

Linearization

Nonterminals are separated into

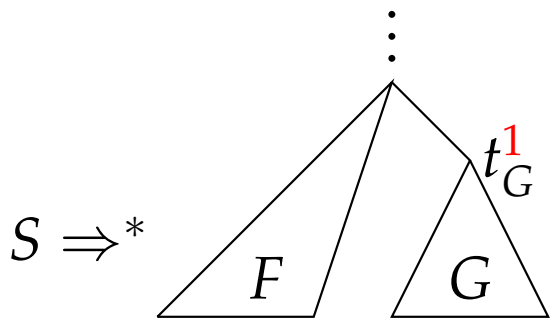
- *semiterminals* : nonterminals rewriting into finite trees,
- ∞-*nonterminals* the other ones.

Lemma (linearization)

For any order-2 scheme $\mathfrak{S}$ *generating a comb, there is* $\mathfrak{S}'$ *with only two* ∞*-nonterminals* $\{S, R\}$ *such that* $\mathbf{Fr}(\mathfrak{S}) = \mathbf{Fr}(\mathfrak{S}')$. *Moreover, their rewriting rules have exactly one occurence of* R *and none of* S.

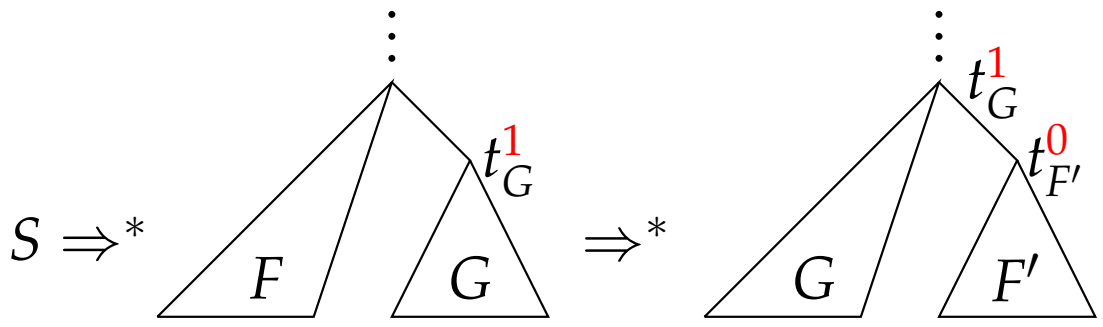
Linearization

Given the shape of the tree, we actually never have two ∞-nonterminals at the same time.

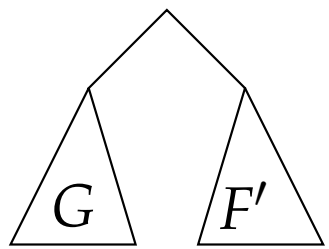


Linearization

Given the shape of the tree, we actually never have two ∞-nonterminals at the same time.

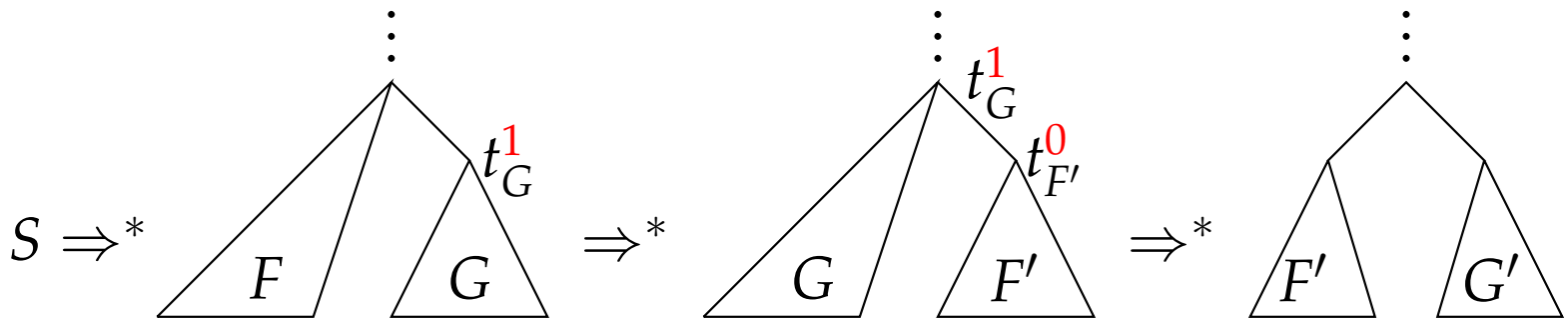


or

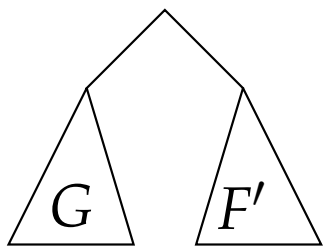


Linearization

Given the shape of the tree, we actually never have two ∞-nonterminals at the same time.



or



Term words

Alphabet $\Sigma = \bigcup_{i=0}^{n} \Sigma_i$ where Σ_0 is called "letters".

$$\theta := \epsilon \mid a \in \Sigma_0 \mid f(\underbrace{\theta, \ldots, \theta}_{i}), f \in \Sigma_i \mid \theta \cdot \theta$$

Term words

Alphabet $\Sigma = \bigcup_{i=0}^{n} \Sigma_i$ where Σ_0 is called "letters".

$$\theta := \epsilon \mid a \in \Sigma_0 \mid f(\underbrace{\theta, \ldots, \theta}_{i}), f \in \Sigma_i \mid \theta \cdot \theta$$

We use variables from $\mathcal{V} = \{z_1, \ldots\}$ to define
$\Sigma(\bar{z}) = \{f(z_1, \ldots, z_k) \mid f \in \Sigma_k\}$.

Term words

Alphabet $\Sigma = \bigcup_{i=0}^{n} \Sigma_i$ where Σ_0 is called "letters".

$$\theta := \epsilon \mid a \in \Sigma_0 \mid f(\underbrace{\theta, \ldots, \theta}_{i}), f \in \Sigma_i \mid \theta \cdot \theta$$

We use variables from $\mathcal{V} = \{z_1, \ldots\}$ to define
$\Sigma(\bar{z}) = \{f(z_1, \ldots, z_k) \mid f \in \Sigma_k\}$.
Let τ, σ be two morphisms on $\Sigma(\bar{z})^*$ w.r.t. concatenation.

for $f \in \Sigma_k$ and $z_1, \ldots, z_k \in \mathcal{V}$,

$$\begin{aligned} \tau(f(z_1, \ldots, z_k)) &\in \mathsf{TW}(\Sigma \cup \{z_1, \ldots, z_k\}) \\ \sigma(f(z_1, \ldots, z_k)) &\in \mathsf{TW}(\Sigma_0 \cup \{z_1, \ldots, z_k\}) \\ &= (\Sigma_0 \cup \{z_1, \ldots, z_k\})^* \end{aligned}$$

2-morphic words

This definition is extended on term words by

$$\begin{aligned} &\text{for } f \in \Sigma_k \\ &\text{and } t_1, \ldots, t_k \in \mathsf{TW}(\Sigma), \\ \tau(f(t_1, \ldots, t_k)) &= \tau(f(z_1, \ldots, z_k))[\forall i, z_i := \tau(t_i)] \\ \sigma(f(t_1, \ldots, t_k)) &= \sigma(f(z_1, \ldots, z_k))[\forall i, z_i := \sigma(t_i)] \end{aligned}$$

Let $\Delta \in \Sigma_0$, words of the form $\sigma(\tau^\omega(\Delta))$ are *2-morphic words*.

Champernowne : 2-morphic words

$\Sigma_0 = \{0, 1\}$ and $\Sigma_1 = \{g\}$.

$$\begin{aligned} \tau(\Delta) &= \Delta g(0) g(1) \\ \tau(g(z)) &= g(z0) g(z1) \\ \sigma(\Delta) &= 01 \\ \sigma(g(z)) &= 1z \end{aligned}$$

In addition $\tau(1) = \sigma(1) = 1$ and $\tau(0) = \sigma(0) = 0$.

$$\tau(\Delta) = \Delta \quad g(0) \quad g(1)$$

Champernowne : 2-morphic words

$\Sigma_0 = \{0, 1\}$ and $\Sigma_1 = \{g\}$.

$$\begin{aligned} \tau(\Delta) &= \Delta g(0) g(1) \\ \tau(g(z)) &= g(z0) g(z1) \\ \sigma(\Delta) &= 01 \\ \sigma(g(z)) &= 1z \end{aligned}$$

In addition $\tau(1) = \sigma(1) = 1$ and $\tau(0) = \sigma(0) = 0$.

$$\begin{aligned} \tau(\Delta) &= \Delta \quad g(0) \quad g(1) \\ \tau^{(2)}(\Delta) &= \Delta \quad g(0) \quad g(1) \quad g(00) \quad g(01) \quad g(10) \quad g(11) \end{aligned}$$

Champernowne : 2-morphic words

$\Sigma_0 = \{0, 1\}$ and $\Sigma_1 = \{g\}$.

$$\begin{aligned} \tau(\Delta) &= \Delta g(0) g(1) \\ \tau(g(z)) &= g(z0) g(z1) \\ \sigma(\Delta) &= 01 \\ \sigma(g(z)) &= 1z \end{aligned}$$

In addition $\tau(1) = \sigma(1) = 1$ and $\tau(0) = \sigma(0) = 0$.

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

Final result

Theorem

The frontiers of combs generated by order-2 schemes are exactly 2-morphic words.

Final result

Theorem

The frontiers of combs generated by order-2 schemes are exactly 2-morphic words.

A *safe* scheme : in every rule $F\,\overrightarrow{x} \Rightarrow T_F$, and every subterm t of T_F, the order of t is lower of equal to any order of x_i inside it.
The proof of the theorem translates words into safe schemes.

Final result

Theorem

The frontiers of combs generated by order-2 schemes are exactly 2-morphic words.

A *safe* scheme : in every rule $F\,\overrightarrow{x} \Rightarrow T_F$, and every subterm t of T_F, the order of t is lower of equal to any order of x_i inside it.
The proof of the theorem translates words into safe schemes.

Moreover, by MSO properties of the pushdown hierarchy, 2-morphic words are also

- ω-frontiers of safe trees,
- paths generated by order-3 safe schemes.

What about unsafe ones?

Consequences

We apply properties of the pushdown hierarchy.

Corollary

For any 2-morphic word w,

- *the MSO theory of w is decidable;*
- *for any MSO-transduction $\mathcal{T}$, if $\mathcal{T}(w)$ is a word, it is a 2-morphic word.*
- *the sequence of indexes of a given letter in a morphic word is at most $\mathcal{O}(2^{2^{Cn}})$ for some C. The bound is tight.*

Other example : characteristic word of $(n!)_{n\geq 0}$, known as the *Liouville constant*.

Conclusion and beyond

This construction builds a new class of graphs for order-2 schemes.

- What about higher orders? can we still linearize ?
- Connexion with classes $\mathbb{S}_k$ of [Fratani-Senizergues], or k-automatic words by [Bárány] ?
- And beyond the pushdown/Caucal/scheme hierarchy? the characteristic word of

$$\left(2^{2^{2^{\cdots}}\}n}\right)_{n\geq 0}$$

has decidable MSO-theory [Thomas]