Many-sorted (Co)Algebraic Specification with Base Sets

(http://fldit-www.cs.uni-dortmund.de/∼peter/IFIP2014.pdf)

Peter Padawitz
TU Dortmund

January 30, 2014

More details can be found in:

- *Algebraic Compiler Construction*, course notes in German
- *(Co)Algebras, (Co-)Horn Logic, and (Co)Induction*

Abstract

We present some fundamentals of a uniform approach to specify, implement and reason about (co)algebraic models in a many-sorted setting that covers constant, polynomial and collection types. Three kinds of (infinite-)tree models (finite terms, coterms and continuous trees) Emphasis yield concrete representations (and Haskell implementations) of initial resp. final models.

On the axiomatic side, a format for recursive equations, which define either constructors on a final model or destructors on an initial one, is introduced. We show how the well-known *iterative* equations, which define continuous trees, can be translated into recursive equations so that the unique solvability of the latter implies the unique solvability of the former.

The recursive *Brzozowski* equations define automata whose states are regular expressions and which accept regular languages. We show how this set of equations can be extended by equations representing a non-left-recursive grammar G such that it defines an acceptor of the language of G.

Contents

Syntax

Let S be a set of **sorts**.

An S**-sorted set** is a tuple $A = (A_s)_{s \in S}$ of sets.

We also write A for the union of A_s over all $s \in S$.

An S**-sorted subset** B of A, written as $B \subseteq A$, is an S-sorted set with $B_s \subseteq A_s$ for all $s \in S$.

Given S-sorted sets $A_1, \ldots, A_n$, an S**-sorted relation** $r \subseteq A_1 \times \cdots \times A_n$ is an S-sorted set with $r_s \subseteq A_{1,s} \times \ldots \times A_{n,s}$ for all $s \in S$.

The S-sorted binary relation $\Delta_A = \{\Delta_{A,s} \mid s \in S\}$ is called the **diagonal of** A^2.

Given S-sorted sets A and B, an S**-sorted function** $f : A \rightarrow B$ is an S-sorted set such that for all $s \in S$, f_s is a function from A_s to B_s.

Set^S denotes the category of S-sorted sets and S-sorted functions.

Let S and BS be sets of **sorts** and **base sets**, respectively.

The set $\mathbb{T}(S, BS)$ of **types over** S **and** BS is inductively defined as follows:

- $S \subseteq \mathbb{T}(S, BS)$. (sorts)
- $BS \subseteq \mathbb{T}(S, BS)$. (base sets)
- For all $n > 0$, $e_1, \ldots, e_n \in \mathbb{T}(S, BS)$, $e_1 \times \cdots \times e_n \in \mathbb{T}(S, BS)$. (product types)
- For all $n > 0$, $e_1, \ldots, e_n \in \mathbb{T}(S, BS)$, $e_1 + \cdots + e_n \in \mathbb{T}(S, BS)$. (sum types)
- For all $e \in \mathbb{T}(S, BS)$, $word(e), bag(e), set(e) \in \mathbb{T}(S, BS)$. (collection types over e)
- For all $X \in BS$ and $e \in \mathbb{T}(S, BS)$, $e^X \in \mathbb{T}(S, BS)$. (power types over e)
- For all $e, e' \in \mathbb{T}(S, BS)$ with $e' \notin BS$, $e^{e'} \in \mathbb{T}(S, BS)$. (functional types over e)

A type is **first-order** if it does not contain functional types.

$\mathbb{T}_1(S, BS)$ denotes the set of first-order types over S and BS.

A type is **flat** if it is a sort, a base set or a collection type over a sort or a base set. $\mathbb{FT}(S, BS)$ denotes the set of flat types over S and BS.

A **signature** $\Sigma = (S, BS, F, P)$ consists of

- a finite set S of **sorts**,
- a finite set BS of **base sets**,
- a finite set F of **operations** $f : e \to e'$ where $e, e' \in \mathbb{T}(S, BS)$,
- a finite set P of **predicates** $p : e$ where $e \in \mathbb{T}(S, BS)$.

For all $f : e \to e' \in F$, $dom(f) = e$ resp. $ran(f) = e'$ is the **domain** resp. **range** of f.

For all $p : e \in P$, $dom(p) = e$ is the **domain** of p.

$f \in F$ is a **constructor** if there are $e_1, \ldots, e_n \in \mathbb{FT}(S, BS)$ such that $dom(f) = e_1 \times \cdots \times e_n$ and $ran(f) \in S$.

Σ is **constructive** if F consists of constructors.

$f \in F$ is a **destructor** if there are $e_1, \ldots, e_n \in \mathbb{FT}(S, BS)$ and $X \in BS$ such that $dom(f) \in S$ and $ran(f) = (e_1 + \cdots + e_n)^X$.

Σ is **destructive** if F consists of destructors.

A constructive signature Let CS be a set of sets (of constants).

Reg(CS) ⇝ regular expressions over CS
The singletons among CS form the traditional "alphabet" of "terminal symbols".

$$S = \{reg\}, \quad BS = \{1, CS\}, \quad F = \{ \begin{array}{l} eps : 1 \to reg, \\ mt : 1 \to reg, \\ con : CS \to reg, \\ par : reg \times reg \to reg, \\ seq : reg \times reg \to reg, \\ iter : reg \to reg \ \}. \end{array}$$

A destructive signature Let X and Y be sets.

DAut(X, Y) ⇝ deterministic Moore automata with input from X and output in Y

$$S = \{state\}, \quad BS = \{X, Y\}, \quad F = \{ \begin{array}{l} \delta : state \to state^X, \\ \beta : state \to Y \ \}. \end{array}$$

$Stream(X) =_{def} DAut(1, X)$ ⇝ streams over X

$Acc(X) =_{def} DAut(X, 2)$ ⇝ deterministic acceptors of subsets of X^*

Let V be a $\mathbb{T}(S, BS)$-sorted set of variables.

The $\mathbb{T}(S, BS)$-sorted set $T_\Sigma(V)$ of **Σ-terms over V** is inductively defined as follows:

- For all $s \in S \cup BS$, $V_s \subseteq T_\Sigma(V)_s$.
- For all $X \in BS$, $X \subseteq T_\Sigma(V)_X$.
- For all $n > 1$, $e_1, \dots, e_n \in \mathbb{T}(S, BS)$, $t = (t_1, \dots, t_n) \in T_\Sigma(V)_{e_1 \times \dots \times e_n}$ and $i \in [n]$, $\pi_i t \in T_\Sigma(V)_{e_i}$.
- For all $n > 1$, $e_1, \dots, e_n \in \mathbb{T}(S, BS)$, $i \in [n]$ and $t \in T_\Sigma(V)_{e_i}$, $\iota_i t \in T_\Sigma(V)_{e_1 + \dots + e_n}$.
- For all $n > 1$, $e_1, \dots, e_n \in \mathbb{T}(S, BS)$ and $t_i \in T_\Sigma(V)_{e_i}$, $i \in [n]$, $(t_1, \dots, t_n) \in T_\Sigma(V)_{e_1 \times \dots \times e_n}$.
- For all $c \in \{word, bag, set\}$, $e \in \mathbb{T}(S, BS)$ and $t \in T_\Sigma(V)^*_e$, $c(t) \in T_\Sigma(V)_{c(e)}$.
- For all $f : e \to e' \in F$ and $t \in T_\Sigma(V)_e$, $ft \in T_\Sigma(V)_{e'}$.
- For all $n > 0$, $e_i, e \in \mathbb{T}(S, BS)$, $x_i \in V_{e_i}$ and $t_i \in T_\Sigma(V)_e$, $1 \leq i \leq n$, $\lambda x_1.t_1 | \dots | x_n.t_n \in T_\Sigma(V)_{e^{e_1 + \dots + e_n}}$.
- For all $e, e' \in \mathbb{T}(S, BS)$, $t \in T_\Sigma(V)_{e^{e'}}$ and $u \in T_\Sigma(V)_{e'}$, $t(u) \in T_\Sigma(V)_e$.
- For all $e \in \mathbb{T}(S, BS)$, $t \in T_\Sigma(V)_2$ and $u, v \in T_\Sigma(V)_e$, $ite(t, u, v) \in T_\Sigma(V)_e$.

For all $f : 1 \longrightarrow e \in F$, we write f for the term $f(\epsilon)$. (ϵ is the unique element of 1.)

A Σ-term t is **first-order** if the range of each subterm of t is first-order.

If t does not contain variables or ite, then t is called **ground**.

T_Σ denotes the set of ground Σ-terms.

The set $Fo_\Sigma(V)$ of **Σ-formulas over** V is inductively defined as follows:

- $True, False \in Fo_\Sigma(V)$.
- For all $p : e \in P$ and $t \in T_\Sigma(V)_e$, $p(t) \in Fo_\Sigma(V)$. (Σ-atoms over V)
- For all $\varphi \in Fo_\Sigma(V)$, $\neg\varphi \in Fo_\Sigma(V)$.
- For all $\varphi, \psi \in Fo_\Sigma(V)$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \Rightarrow \psi$, $\varphi \Leftarrow \psi$, $\varphi \Leftrightarrow \psi \in Fo_\Sigma(V)$.
- For all $x \in V$ and $\varphi \in Fo_\Sigma(V)$, $\forall x\varphi$, $\exists x\varphi \in Fo_\Sigma(V)$.

Predicate lifting

For alle $e \in \mathbb{T}_1(S, BS)$, the functor $F_e : Set^S \to Set$ is inductively defined as follows:
For all S-sorted sets A, B, S-sorted functions $h : A \to B$, $s \in S$, $X \in BS$, $n > 1$ and $e, e_1, \dots, e_n \in \mathbb{T}_1(S, BS)$,

$$
\begin{array}{lll}
F_s(A) = A_s, & F_s(h) = h_s, & \text{(projection functor)} \\
F_X(A) = X, & F_X(h) = id_X, & \text{(constant functor)} \\
F_{e_1+\dots+e_n}(A) = F_{e_1}(A) + \dots + F_{e_n}(A), & F_{e_1+\dots+e_n}(h) = F_{e_1}(h) + \dots + F_{e_n}(h), & \\
F_{e_1\times\dots\times e_n}(A) = F_{e_1}(A) \times \ldots \times F_{e_n}(A), & F_{e_1\times\dots\times e_n}(h) = F_{e_1}(h) \times \ldots \times F_{e_n}(h), & \\
F_{word(e)}(A) = F_e(A)^*, & F_{word(e)}(h) = F_e(h)^*, & \\
F_{bag(e)}(A) = \mathcal{B}_{fin}(F_e(A)), & F_{bag(e)}(h) = \mathcal{B}_{fin}(F_e(h)), & \\
F_{set(e)}(A) = \mathcal{P}_{fin}(F_e(A)), & F_{set(e)}(h) = \mathcal{P}_{fin}(F_e(h)), & \\
F_{e^X}(A) = F_e(A)^X, & F_{e^X}(h) = F_e(h)^X. &
\end{array}
$$

We mostly write A_e instead of $F_e(A)$.

$[0] =_{def} \emptyset$

For all $n > 0$, $[n] =_{def} \{1, \ldots, n\}$.

$$(a_1, \ldots, a_n) =_{word} (b_1, \ldots, b_n) \quad \Leftrightarrow_{def} \quad (a_1, \ldots, a_m) = (b_1, \ldots, b_n)$$

$$(a_1, \ldots, a_n) =_{bag} (b_1, \ldots, b_n) \quad \Leftrightarrow_{def} \quad \exists\, f : [n] \hookrightarrow [n] : (a_{f(1)}, \ldots, a_{f(n)}) = (b_1, \ldots, b_n)$$

$$(a_1, \ldots, a_n) =_{set} (b_1, \ldots, b_n) \quad \Leftrightarrow_{def} \quad \{a_1, \ldots, a_m\} = \{b_1, \ldots, b_n\}$$

$\mathcal{B}_{fin}(A) = \{[(a_1, \ldots, a_n)]_{=_{set}} \mid a_1, \ldots, a_n \in A,\ n \in \mathbb{N}\}$

$\mathcal{B}_{fin}(h) : \mathcal{B}_{fin}(A) \to \mathcal{B}_{fin}(B)$ maps $[(a_1, \ldots, a_n)]_{=_{bag}}$ to $[(h(a_1), \ldots, h(a_n))]_{=_{bag}}$.

$\mathcal{P}_{fin}(A) = \{\{a_1, \ldots, a_n\} \mid a_1, \ldots, a_n \in A,\ n \in \mathbb{N}\}$

$\mathcal{P}_{fin}(h) : \mathcal{P}_{fin}(A) \to \mathcal{P}_{fin}(B)$ maps $\{a_1, \ldots, a_n\}$ to $\{h(a_1), \ldots, h(a_n)\}$.

Relation lifting

Given an S-sorted relation $R \subseteq A \times B$, R is extended to a $\mathbb{T}_1(S, BS)$-sorted relation inductively as follows:

Let $s \in S$, $e_1, \dots, e_n, e \in \mathbb{T}_1(S, BS)$ and $X \in BS$.

$$\begin{array}{rcl}
R_X & = & \Delta_X, \\
R_{e_1 + \dots + e_n} & = & \{((a,i),(b,i)) \mid (a,b) \in R_{e_i},\ i \in [n]\}, \\
R_{e_1 \times \dots \times e_n} & = & \{((a_1,\dots,a_n),(b_1,\dots,b_n)) \mid \forall\, i \in [n] : (a_i,b_i) \in R_{e_i})\}, \\
R_{word(e)} & = & \bigcup_{n \in \mathbb{N}} \{((a_1,\dots,a_n),(b_1,\dots,b_n)) \mid \forall\, i \in [n] : (a_i,b_i) \in R_e)\}, \\
R_{bag(e)} & = & \bigcup_{n \in \mathbb{N}} \{((a_{f(1)},\dots,a_{f(n)}),(b_1,\dots,b_n)) \mid f : [n] \hookrightarrow [n], \\
 & & \qquad \forall\, i \in [n] : (a_i,b_i) \in R_e\}, \\
R_{set(e)} & = & \bigcup_{n \in \mathbb{N}} \{((a_1,\dots,a_m),(b_1,\dots,b_n)) \mid \forall\, i \in [m]\ \exists\, j \in [n] : (a_i,b_j) \in R_e, \\
 & & \qquad \forall\, j \in [n]\ \exists\, i \in [m] : (a_i,b_j) \in R_e\}, \\
R_{e^X} & = & \{(f,g) \mid \forall\, x \in X : (f(x),g(x)) \in R_e\}.
\end{array}$$

A **Σ-algebra** A consists of

- an S-sorted set, usually also denoted by A,
- for each $f : e \to e' \in F$, a function $f^A : A_e \to A_{e'}$,
- for each $p : e \in P$, a subset p^A of A_e.

Suppose that all function and relation symbols of Σ have first-order domains and ranges. Let A, B be Σ-algebras.

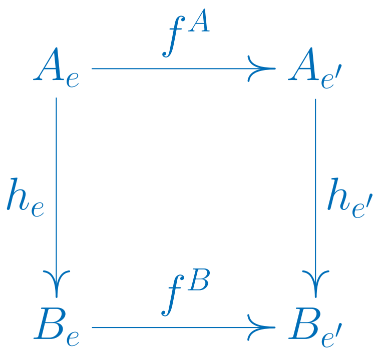
An S-sorted function $h : A \to B$ is a **Σ-homomorphism** if for all $f : e \to e' \in F$, $h_{e'} \circ f^A = f^B \circ h_e$, and for all $p : e \in P$, $h_e(p^A) \subseteq p^B$.

Alg_Σ denotes the category of Σ-algebras and Σ-homomorphisms.

⇝ A Σ-homomorphism h is iso in Alg_Σ iff h is bijective and for all $p : e \in P$, $p^B \subseteq h_e(p^A)$.

Let U_S be the forgetful functor from Alg_Σ to Set^S.

For all $f : e \to e' \in F$, $\overline{f} : F_e U_S \to F_{e'} U_S$ with $\overline{f}(A) =_{def} f^A$ for all $A \in Alg_\Sigma$ is a natural transformation:



Given a category $\mathcal{K}$ and an endofunctor F on $\mathcal{K}$,

- an F**-algebra** or F**-dynamics** is a $\mathcal{K}$-morphism $\alpha : F(A) \to A$,
- an F**-coalgebra** or F**-codynamics** is a $\mathcal{K}$-morphism $\alpha : A \to F(A)$.

Alg_F and $coAlg_F$ denote the categories of F-algebras resp. F-coalgebras where

- an Alg_F**-morphism** from $\alpha : F(A) \to A$ to $\beta : F(B) \to B$ is a $\mathcal{K}$-morphism $h : A \to B$ with $h \circ \alpha = \beta \circ F(h)$,
- a $coAlg_F$**-morphism** from $\alpha : A : F(A)$ to $\beta : B \to F(B)$ is a $\mathcal{K}$-morphism $h : A \to B$ with $F(h) \circ \alpha = \beta \circ h$.

A constructive signature $\Sigma = (S, BS, F, P)$ induces a functor $H_\Sigma : Set^S \to Set^S$:

Let $s \in S$ and $\{f_1 : e_1 \to s, \ldots, f_n : e_n \to s\} = \{f \in F \mid ran(f) = s\}$.

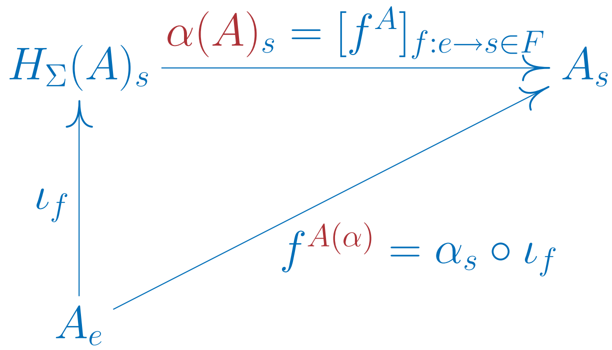
$$H_\Sigma(A)_s \;=_{def}\; A_{e_1+\cdots+e_n}.$$

Alg_Σ and Alg_{H_Σ} are equivalent categories:

Let $A \in Alg_\Sigma$ and $\alpha : A \to H_\Sigma(A) \in Alg_{H_\Sigma}$.

$\alpha(A) : A \to H_\Sigma(A)$ and the Σ-algebra $A(\alpha)$ are defined as follows:

For all $s \in S$ and $f : e \to s \in F$,



Example *Reg(CS)* $H_{Reg(CS)}(A)_{reg} = 1 + 1 + CS + A_{reg}^2 + A_{reg}^2 + A_{reg}.$ ❏

➾ $h : A \to B$ is a Σ-homomorphism $\Leftrightarrow$ h is an Alg_{H_Σ}-morphism from $\alpha(A)$ to $\alpha(B)$:

$$\begin{array}{ccc} A_e & \xrightarrow{f^A} & A_s \\ {\scriptstyle h_e}\downarrow & & \downarrow{\scriptstyle h_s} \\ B_e & \xrightarrow[f^B]{} & B_s \end{array} \quad\Longleftrightarrow\quad \begin{array}{ccc} H_\Sigma(A)_s & \xrightarrow{\alpha(A)_s} & A_s \\ {\scriptstyle H_\Sigma(h)_s}\downarrow & & \downarrow{\scriptstyle h_s} \\ H_\Sigma(B)_s & \xrightarrow[\alpha(B)_s]{} & B_s \end{array}$$

➾ $h : \alpha \to \beta$ is an Alg_{H_Σ}-morphism $\Leftrightarrow$ h is a Σ-homomorphism from $A(\alpha)$ to $A(\beta)$:

$$\begin{array}{ccc} H_\Sigma(A)_s & \xrightarrow{\alpha_s} & A_s \\ {\scriptstyle H_\Sigma(h)_s}\downarrow & & \downarrow{\scriptstyle h_s} \\ H_\Sigma(B)_s & \xrightarrow[\beta_s]{} & B_s \end{array} \quad\Longleftrightarrow\quad \begin{array}{ccc} A_e & \xrightarrow{f^{A(\alpha)}} & A_s \\ {\scriptstyle h_e}\downarrow & & \downarrow{\scriptstyle h_s} \\ B_e & \xrightarrow[f^{A(\beta)}]{} & B_s \end{array}$$

A destructive signature $\Sigma = (S, BS, F, P)$ induces a functor $H_\Sigma : Set^S \rightarrow Set^S$:

Let $s \in S$ and $\{f_1 : s \rightarrow e_1, \ldots, f_n : s \rightarrow e_n\} = \{f \in F \mid dom(f) = s\}$.

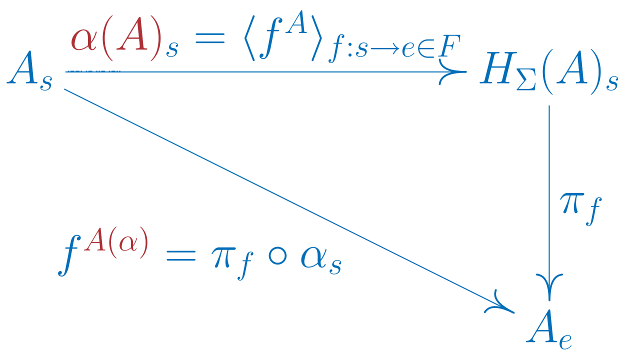
$$H_\Sigma(A)_s \;=_{def}\; A_{e_1 \times \cdots \times e_n}.$$

Alg_Σ and $coAlg_{H_\Sigma}$ are equivalent categories:

Let $A \in Alg_\Sigma$ and $\alpha : H_\Sigma(A) \rightarrow A \in coAlg_{H_\Sigma}$.

$\alpha(A) : H_\Sigma(A) \rightarrow A$ and the Σ-algebra $A(\alpha)$ are defined as follows:

For all $s \in S$ and $f : s \rightarrow e \in F$,



Example $DAut(X,Y)$ $H_{DAut(X,Y)}(A)_{state} = A^X_{state} \times Y$. ❏

Haskell implementation of Alg_Σ (without predicates)

Let $\Sigma = (S, BS, F)$ be a signature, $S = \{s_1, \dots, s_m\}$ and $F = \{f_i : e_i \to e_i' \mid 1 \leq i \leq n\}$.

```
data Sigma s_1 ... s_m = Sigma {f_1 :: e_1 -> e_1', ...,
                                f_n :: e_n -> e_n'}
```

Let V be an $\mathbb{T}(S,BS)$-sorted set of variables, A be an S-sorted set and A^V be the set of **valuations of** V **in** A, i.e., $\mathbb{T}(S,BS)$-sorted functions from V to A.

For all $g \in A^V$, $e \in \mathbb{T}(S,BS)$, $a \in A_e$, $x \in V_e$ and $z \in V$.

$$g[a/x](z) \ =_{def} \ \begin{cases} a & \text{if } z = x, \\ g(z) & \text{otherwise.} \end{cases}$$

Evaluation of terms and formulas

The $\mathbb{T}(S, BS)$-sorted extension $g^* : T_\Sigma(V) \to A$ of g is defined as follows:

- For all $x \in V$, $g^*(x) = g(x)$.
- For all $x \in X \in \cup BS$, $g^*(x) = x$.
- For all $n > 1$, $e_1, \dots, e_n \in \mathbb{T}(S, BS)$, $t = (t_1, \dots, t_n) \in T_\Sigma(V)_{e_1 \times \dots \times e_n}$ and $i \in [n]$, $g^*(\pi_i t) = g^*(t_i)$.
- For all $n > 1$, $e_1, \dots, e_n \in \mathbb{T}(S, BS)$, $i \in [n]$ and $t \in T_\Sigma(V)_{e_i}$, $g^*(\iota_i t) = (g^*(t), i)$.
- For all $n \in \mathbb{N}$ and $t_1, \dots, t_n \in T_\Sigma(V)$, $g^*(t_1, \dots, t_n) = (g^*(t_1), \dots, g^*(t_n))$.
- For all $c \in \{word, bag, set\}$, $c(t) \in T_\Sigma(V)_{c(e)}$, $g^*(c(t)) = [g^*(t)]_{=_c}$.
- For all $f : e \to e' \in F$ and $t \in T_\Sigma(V)_e$, $g^*(f(t)) = f^A(g^*(t))$.
- For all $n > 0$, $e_i, e \in \mathbb{T}(S, BS)$, $x_i \in V_{e_i}$, $t_i \in T_\Sigma(V)_e$, $i \in [n]$, and $(a, i) \in A_{e_1 + \dots + e_n}$,

$$g^*(\lambda x_1.t_1 | \dots | x_n.t_n)(a, i) = g[a/x_i]^*(t_i).$$

- For all $e, e' \in \mathbb{T}(S, BS)$, $t \in T_\Sigma(V)_{e^{e'}}$ and $u \in T_\Sigma(V)_{e'}$,

$$g^*(t(u)) = g^*(t)(g^*(u)).$$

- For all $e \in \mathbb{T}(S, BS)$, $t \in T_\Sigma(V)_2$ and $u, v \in T_\Sigma(V)_e$,

$$g^*(ite(t,u,v)) = \begin{cases} g^*(u) & \text{if } g^*(t) = 1, \\ g^*(v) & \text{otherwise.} \end{cases}$$

For all $e \in \mathbb{T}(S, BS)$ and first-order Σ-terms t, we define:

$$\begin{array}{rcl} t^A : A^V & \longrightarrow & A_e \\ g & \longmapsto & g^*(t) \end{array}$$

$\bar{t} : \_^V \rightarrow F_e U_S$ with $\bar{t}_A =_{def} t^A$ for all $A \in Alg_\Sigma$ is a natural transformation:

$$\begin{array}{ccc} A^V & \xrightarrow{\ t^A\ } & A_e \\ {\scriptstyle h^V}\downarrow & (1) & \downarrow{\scriptstyle h_e} \\ B^V & \xrightarrow[\ t^B\ ]{} & B_e \end{array}$$

(1) is equivalent to the *Substitution Lemma*:

For all $g \in A^V$, Σ-homomorphisms $h : A \to B$ and first-order Σ-terms t,

$$(h \circ g)^*(t) = (h \circ g^*)(t). \tag{2}$$

A interprets a Σ-formula φ over V by the set φ^A of valuations that satisfy φ:

For all $e \in \mathbb{T}(S, BS)$, $p : e \in P$, $t, u \in T_\Sigma(V)_e$, $\varphi, \psi \in Fo_\Sigma(V)$, $s \in S \cup BS$ and $x \in V_s$,

$$\begin{aligned}
True^A &= A^V, \\
False^A &= \emptyset, \\
p(t)^A &= \{g \in A^V \mid g^*(t) \in p^A\}, \\
(\neg\varphi)^A &= A^V \setminus \varphi^A, \\
(\varphi \wedge \psi)^A &= \varphi^A \cap \psi^A, \\
(\varphi \vee \psi)^A &= \varphi^A \cup \psi^A, \\
(\forall x\varphi)^A &= \{g \in A^V \mid \forall\, a \in A_s : g[a/x] \in \varphi^A\}, \\
(\exists x\varphi)^A &= \{g \in A^V \mid \exists\, a \in A_s : g[a/x] \in \varphi^A\}.
\end{aligned}$$

A **satisfies** φ ($A \models \varphi$) if $\varphi^A = A^V$.

The *Substitution Lemma* implies:

For all $g \in A^V$, Σ-homomorphisms $h : A \to B$ and **negation-free** Σ-formulas φ,

$$g \in \varphi^A \quad \Rightarrow \quad h \circ g \in \varphi^B.$$

Initial and final algebras

An S-sorted binary relation R on A is a **Σ-congruence on A** if for all $f : e \to e' \in F$ and $(a, b) \in R_e$, $(f^A(a), f^A(b)) \in R_{e'}$.

If Σ is destructive, then Σ-congruences are also called **Σ-bisimulations**.

An S-sorted subset B of A is a **Σ-invariant** (or **Σ-subalgebra of A**) if for all $f : e \to e' \in F$ andl $a \in A_e$, $f^A(a) \in A_{e'}$.

A Σ-algebra A **satisfies the induction principle** if for all S-sorted subsets B of A, $A \subseteq B$ iff B contains a Σ-invariant.

A is initial in $Alg_\Sigma \iff A$ satisfies the induction principle and for all Σ-algebras B there is a Σ-homomorphism from A to B.

A Σ-algebra A **satisfies the coinduction principle** if for all S-sorted binary relations R on A, $R \subseteq \Delta_A$ iff R is contained in a Σ-congruence.

A is final in $Alg_\Sigma \iff A$ satisfies the coinduction principle and for all Σ-algebras B there is a Σ-homomorphism from B to A.

Let $\Sigma = (S, BS, F)$ be a constructive signature.

T_Σ is a Σ-algebra: For all $f : e \to s \in F$ and $t \in T_{\Sigma,e}$, $f^{T_\Sigma}(t) = ft$.

Let $\sim$ be the least $\mathbb{FT}(S, BS)$-sorted equivalence relation on T_Σ such that

- for all $n > 1$, $e_1, \dots, e_n \in \mathbb{FT}(S, BS)$ and $t_i, t_i' \in T_{\Sigma,e_i}$, $i \in [n]$,

$$t_1 \sim_{e_1} t_1' \wedge \dots \wedge t_n \sim_{e_n} t_n' \text{ implies } (t_1, \dots, t_n) \sim_{e_1 \times \dots \times e_n} (t_1', \dots, t_n'),$$

- for all $n > 1$, $e \in \mathbb{FT}(S, BS)$ and $t_i, t_i' \in T_{\Sigma,e}$, $i \in [n]$,

$$t_1 \sim_e t_1' \wedge \dots \wedge t_n \sim_e t_n' \text{ implies } word(t_1, \dots, t_n) \sim_{word(s)} word(t_1', \dots, t_n'),$$

- for all $n > 1$, $e \in \mathbb{FT}(S, BS)$, $f : [n] \hookrightarrow [n]$ and $t_i, t_i' \in T_{\Sigma,e}$, $i \in [n]$,

$$t_1 \sim_e t_1' \wedge \dots \wedge t_n \sim_e t_n' \text{ implies } bag(f(t_1), \dots, f(t_n)) \sim_{bag(s)} bag(t_1', \dots, t_n'),$$

- for all $m, n > 0$, $e \in \mathbb{FT}(S, BS)$, $t_i \in T_{\Sigma,e}$, $i \in [m]$, and $t_i' \in T_{\Sigma,e}$, $i \in [n]$,

$$\forall\, i \in [m]\ \exists\, j \in [n] : t_i \sim_e t_j' \wedge \forall\, j \in [n]\ \exists\, i \in [m] : t_i \sim_e t_j'$$
$$\text{implies } set(t_1, \dots, t_m) \sim_{set(s)} set(t_1', \dots, t_n'),$$

- for all $s \in S$, $f : e \to s \in F$ and $t, t' \in T_{\Sigma,e}$, $t \sim_e t'$ implies $ft \sim_s ft'$,
- for all $X \in BS$, $\sim_X = \Delta_X$.

For simplicity, we identify T_Σ with $T_\Sigma/\!\sim$.

T_Σ is initial in Alg_Σ.

For all Σ-algebras A, the unique Σ-homomorphism

$$fold^A : T_\Sigma \to A$$

is defined inductively as follows: For all $s \in S$, $X \in BS$, $x \in X$, $c \in \{word, bag, set\}$, $e \in S \cup BS$, $t \in T^*_{\Sigma,e}$, $f : e' \to s' \in F$ and $t' \in T_{\Sigma,e'}$,

$$\begin{array}{lll} fold^A_{c(e)}(c(t)) & = & [fold^A_e(t)]_{=_c}, \\ fold^A_s(ft') & = & f^A(fold^A_{e'}(t')). \end{array}$$

Haskell implementation of T_Σ

Let $S = \{s_1, \ldots, s_m\}$ and $F = \{c_{ij} : e_{ij} \to s_i \mid 1 \le i \le m,\ 1 \le j \le n_i\}$.

```
data Ts1 = C11 e11 | ... | C1n_1 e1n_1
...
data Tsm = Cm1 em1 | ... | Cmn_m emn_m
```

Example *Reg(CS)*

```
data RegT cs = Eps | Mt | Con cs | Par (RegT cs) (RegT cs) |
               Seq (RegT cs) (RegT cs) | Iter (RegT cs)      Reg(CS)-terms

regT :: cs -> Reg cs (RegT cs)                               term algebra
regT cs = Reg Eps Mt Con Var Par Seq Iter

foldReg :: Reg cs reg -> RegT cs -> reg
foldReg alg Eps       = eps alg
foldReg alg Mt        = mt alg
foldReg alg (Con c)   = con alg c
foldReg alg (Var x)   = var alg x
foldReg alg (Par t u) = par alg (foldReg alg t) (foldReg alg u)
foldReg alg (Seq t u) = seq_ alg (foldReg alg t) (foldReg alg u)
foldReg alg (Iter t)  = iter alg (foldReg alg t)
```

❏

Algebra makes compilers generic

Given a CF grammar G and a target language (= $\Sigma(G)$-algebra A), a compiler composes a parser $parse_G : X^* \to T_{\Sigma(G)}$ with $fold^A : T_{\Sigma(G)} \to A$.

Hence the compiler *algorithm* is completely determined by the parser algorithm and thus the compiler can be made a *generic* function that transforms input from X^* directly – without constructing and traversing syntax trees (= $\Sigma(G)$-terms) – into output:

$$compile_G : Alg_{\Sigma(G)} \times X^* \to A$$

Let $\Sigma = (S, BS, F)$ be a destructive signature and

$$\begin{aligned} D &= \{d \in F \mid ran(d) \text{ is not a power type}\} \cup \\ &\quad \{d_x : s \to e \mid d : s \to e^X \in F,\ x \in X\}. \end{aligned}$$

For all $d : s \to e^X$, $a \in A_s$ and $x \in X$, $d_x^A(a) =_{def} d^A(a)(x)$.

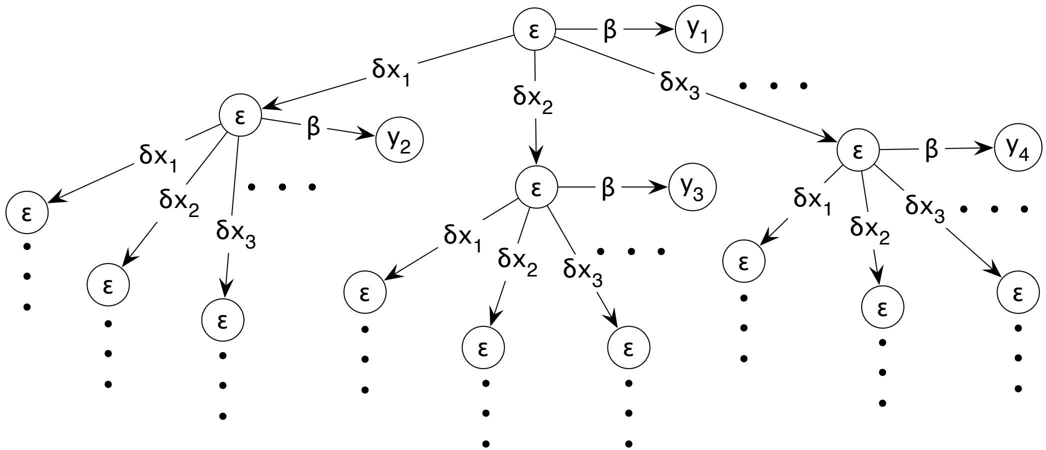
coT_Σ denotes the greatest $\mathbb{FT}(S, BS)$-sorted set of prefix closed partial functions

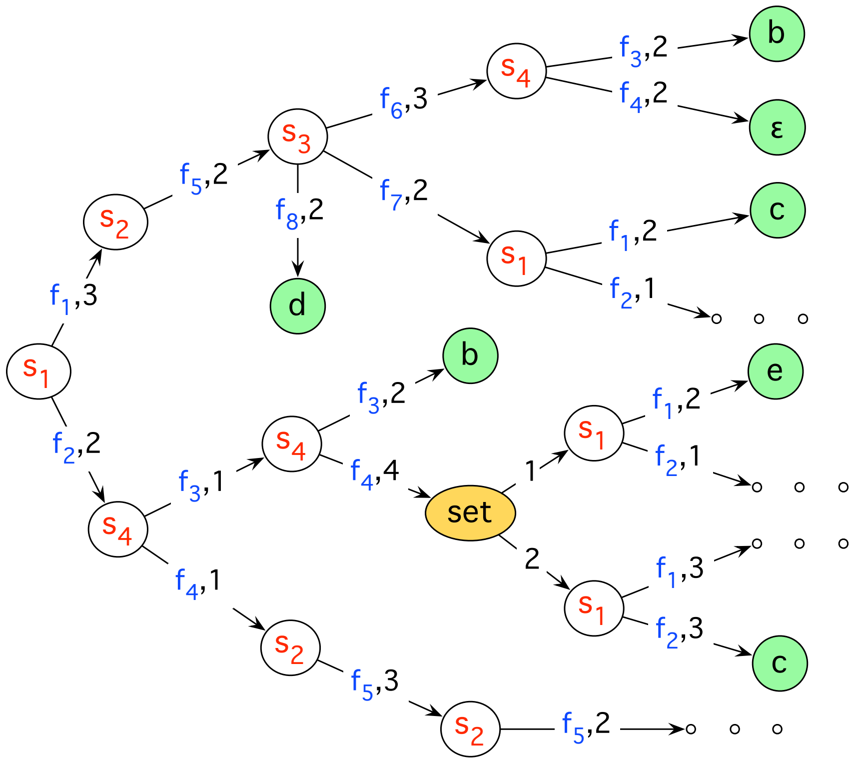
$$t : ((D \times \mathbb{N}) \cup \mathbb{N})^* \multimap 1 + \{word, bag, set\} + \cup BS$$

such that the following conditions hold true:

- For all $s \in S$, $t \in coT_{\Sigma,s}$ and $d : s \to e_1 + \cdots + e_n \in D$, $t(\epsilon) = \epsilon$ and there is $i_d \in [n]$ such that $\lambda w.t((d, i_d)w) \in coT_{\Sigma,e_i}$ and $def(t) \cap ((D \times \mathbb{N}) \cup \mathbb{N}) = \{(d, i_d) \mid dom(d) = s\}$.
- For all $c \in \{word, bag, set\}$, $s \in S \cup BS$ and $t \in coT_{\Sigma,c(s)}$, $t(\epsilon) = c$ and there is $n_t \in \mathbb{N}$ such that for all $i \in n_t$, $\lambda w.t(iw) \in coT_{\Sigma,s}$, and $def(t) \cap ((D \times \mathbb{N}) \cup \mathbb{N}) = [n_t]$.
- For all $X \in BS$, $coT_{\Sigma,X} = X$ (here identified with the set $1 \to X$ of functions).

Example A $DAut(X, Y)$-coterm of sort *state*:





A Σ*-coterm with destructors* $f_1, \ldots, f_8$ *mapping into sum types.*

Green-colored nodes contain base elements $(a, b, c, d, e, \epsilon)$*.*

Each red label s *is the sort of the subcoterm whose root is labelled with* s*.*

Let $\sim$ be the greatest $\mathbb{FT}(S,BS)$-sorted equivalence relation on coT_Σ such that

- for all $s\in S$, $t\sim_s t'$ and $d\in D\times\mathbb{N}$, $\lambda w.t(dw)\sim\lambda w.t'(dw)$,
- for all $s\in S\cup BS$ and $t\sim_{word(s)} t'$, $n_t=n_{t'}$ and for all $i\in[n_t]$, $\lambda w.t(iw)\sim_s\lambda w.t'(iw)$,
- for all $s\in S\cup BS$, $t\sim_{bag(s)} t'$ and $f:[n_t]\hookrightarrow[n_t]$, $n_t=n_{t'}$ and for all $i\in[n_t]$, $\lambda w.t(f(i)w)\sim_s\lambda w.t'(iw)$,
- for all $s\in S\cup BS$, $t\sim_{set(s)} t'$, $i\in[n_t]$ and $j\in[n_{t'}]$ there are $k\in[n_{t'}]$ and $l\in[n_t]$ such that $\lambda w.t(iw)\sim_s\lambda w.t'(kw)$ and $\lambda w.t(lw)\sim_s\lambda w.t'(jw)$,
- for all $X\in BS$, $\sim_X=\Delta_X$.

For simplicity, we identify coT_Σ with $coT_\Sigma/\sim$.

coT_Σ is a Σ-algebra: Let $s\in S$, $t\in coT_{\Sigma,s}$, $e_1,\dots,e_n\in\mathbb{FT}(S,BS)$ and $w\in D^*$.

- For all $d:s\to e_1+\dots+e_n\in F$, $d^{coT_\Sigma}(t)(w)=t((d,i_d)w)$.
- For all $X\in BS$, $d:s\to(e_1+\dots+e_n)^X\in F$ and $x\in X$,

$$d^{coT_\Sigma}(t)(x)(w)=t((d_x,i_{d_x})w).$$

coT_Σ is final in Alg_Σ.

For all Σ-algebras A, the unique Σ-homomorphism $unfold^A : A \to coT_\Sigma$ is defined as follows: For all $s \in S$, $a \in A_s$, $d \in D$, $w \in D^*$ and $i, k \in \mathbb{N}$,

$$\begin{aligned}
unfold^A_s(a)(\epsilon) &= \epsilon, \\
unfold^A_s(a)((d,i)w) &= \begin{cases} unfold^A_{e_i}(b)(w) & \text{if } dom(d) = s \wedge \\ & \quad \exists\, e_1,\dots,e_n \in \mathbb{FT}(S,BS): \\ & \quad ran(d) = e_1 + \cdots + e_n \wedge d^A(a) = (b,i), \\ \text{undefined} & \text{otherwise,} \end{cases} \\
unfold^A_s(a)(kw) &= \begin{cases} unfold^A_s(a_k)(w) & \text{if } \exists\, c \in \{word, bag, set\}, e \in S \cup BS: \\ & \quad s = c(e) \wedge a = [(a_1,\dots,a_n)]_{=_c} \wedge k \in [n], \\ \text{undefined} & \text{otherwise.} \end{cases}
\end{aligned}$$

Example $DAut(X,Y)$ $coT_{DAut(X,Y)}$ is $DAut(X,Y)$-isomorphic to the $DAut(X,Y)$-Algebra $Beh(X,Y)$ of **behavior functions**:

$$Beh(X,Y)_{state} \;=_{def}\; Y^{X^*}.$$

For all $f : X^* \to Y$, $x \in X$ und $w \in X^*$,

$$\delta^{Beh}(f)(x)(w) =_{def} f(xw) \quad \text{and} \quad \beta^{Beh}(f) =_{def} f(\epsilon).$$

For all $a \in A_{state}$, $x \in X$ and $w \in X^*$,

$$\begin{aligned} unfold^A(a)(\epsilon) &= \beta^A(a), \\ unfold^A(a)(xw) &= unfold^A(\delta^A(a)(x))(w). \end{aligned}$$

Given $f : X^* \to Y$, a $DAut(X,Y)$-algebra A and an **initial state** $s \in A_{state}$, the **initial automaton** (A, s) **realizes** $f \;\Leftrightarrow_{def}\; unfold^A(s) = f$. ❏

Haskell implementation of coT_Σ

Let $S = \{s_1, \dots, s_m\}$ and $F = \{d_{ij} : s_i \to e_{ij} \mid 1 \le i \le m,\ 1 \le j \le n_i\}$.

```
data Ts1 = C1 {attr_11 :: e11, ..., attr_1n_1 :: e1n_1}
...
data Tsm = Cm {attr_m1 :: em1, ..., attr_mn_m :: emn_m)}
```

Example $DAut(X,Y)$

```
data DAutT x y = State {next :: x -> DAutT x y, out :: y}
```

DAut(X,Y)-coterms

```
dAutT :: DAut x y (DAutT x y)
dAutT = DAut {delta = next, beta = out}
```

coterm algebra

```
dAutB :: DAut x y ([x] -> y)
dAutB = DAut {delta = \f x w -> f $ x:w, beta = \f -> f []}
```

algebra of behavior functions

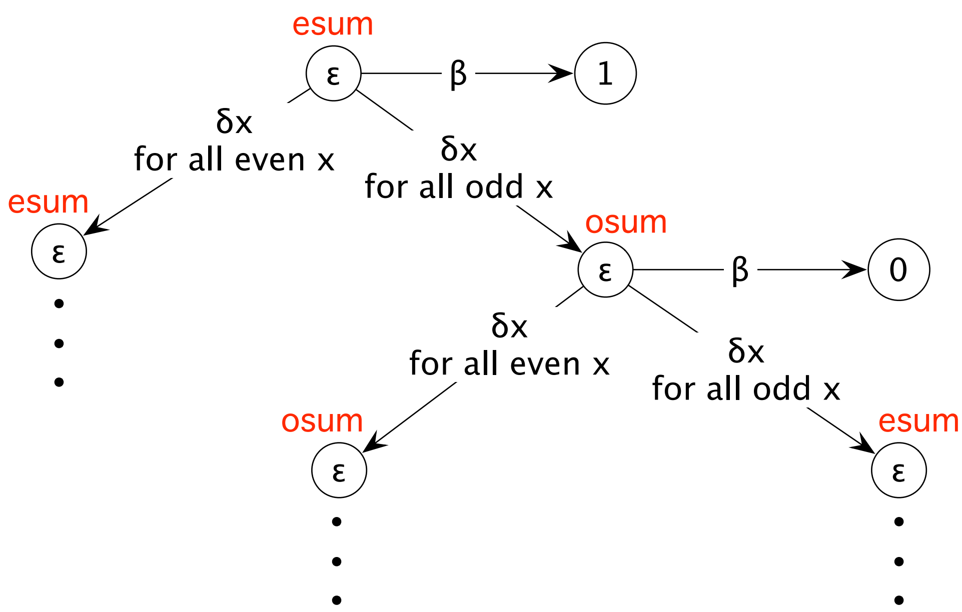
```
unfoldDAut :: DAut x y state -> state -> DAutT x y
unfoldDAut alg s = State {next = unfoldDAut alg . delta alg s,
                          out = beta alg s}
```

unfold into coterms

```
unfoldDAutF :: DAut x y state -> state -> [x] -> y
unfoldDAutF alg s []    = beta alg s
unfoldDAutF alg s (x:w) = unfoldDAutF alg (delta alg s x) w
```

unfold into behavior functions

```
esum,osum :: DAutT Int Bool
esum = State {next = \x -> if even x then esum else osum, out = True}
osum = State {next = \x -> if even x then osum else esum, out = False}
```



Let A be the $Acc(\mathbb{Z})$-subalgebra of $coT_{Acc(\mathbb{Z})}$ with $A_{state} = \{esum, osum\}$ and

$$\begin{array}{rcl} f : \mathbb{Z}^* & \longrightarrow & 2 \\ (x_1, \ldots, x_n) & \longmapsto & \sum_{i=1}^n x_i \text{ is even} \end{array} \qquad \begin{array}{rcl} g : \mathbb{Z}^* & \longrightarrow & 2 \\ (x_1, \ldots, x_n) & \longmapsto & \sum_{i=1}^n x_i \text{ is odd} \end{array}$$

$$\begin{array}{rcl} h : A & \longrightarrow & Beh(\mathbb{Z}, 2) \\ esum & \longmapsto & f \\ osum & \longmapsto & g \end{array}$$

Since h is $Acc(\mathbb{Z})$-homomorphic and $Beh(\mathbb{Z}, 2)$ is final in $Alg_{Acc(\mathbb{Z})}$, $(A, esum)$ realizes f and $(A, osum)$ realizes g:

```
unfoldDAutF dAutT esum = even . sum
unfoldDAutF dAutT osum = odd . sum
```

❏

Recursive equations

Given a constructive signature $C\Sigma = (S, BS, C)$, a destructive signature $D\Sigma = (S, BS', D)$ and a signature $B\Sigma = (\emptyset, BS \cup BS', B)$ of base operations, $\Psi = (S, BS \cup BS', B, C, D)$ is called a **bisignature**.

A set

$$E = \{d(c(x_1, \dots, x_{n_c})) = t_{d,c} \mid d : s \to e \in D,\ c : s_1 \times \cdots \times s_{n_c} \to s \in C\}$$

is a **recursive system of Ψ-equations** if the following conditions hold true:

- For all $d \in D$ and $c \in C$, $\mathit{freeVars}(t_{d,c}) \subseteq \{x_1, \dots, x_{n_c}\}$.
- C is the union of disjoint sets C_1 and C_2.
- For all $d \in D$, $c \in C_1$ and subterms du of $t_{d,c}$, u is a variable and $t_{d,c}$ is a term without elements of C_2.

  $\Rightarrow$ no nesting of destructors, but possible nestings of constructors from C_1
- For all $d \in D$, $c \in C_2$, subterms du of $t_{d,c}$ and paths p of (the tree representation of) $t_{d,c}$, u consists of destructors and a variable and p contains at most one occurrence of an element of C_2.

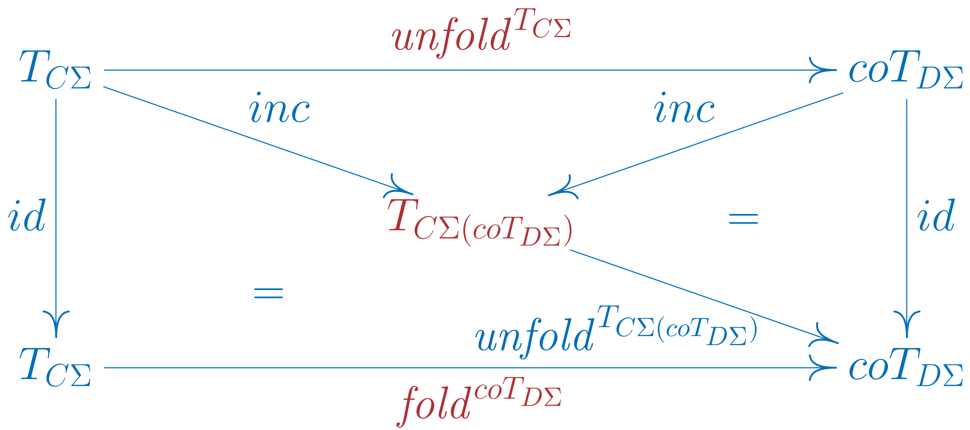
  $\Rightarrow$ no nesting of constructors from C_2, but possible nestings of destructors

Let $\Sigma = (S, BS, B \cup C \cup D)$. A **Ψ-algebra** is a Σ-algebra.

Let E be a recursive system of Ψ-equations and A, A' be Ψ-algebras that satisfy E.

$A|_{C\Sigma} = A'|_{C\Sigma}$ is initial in $Alg_{C\Sigma}$, $A|_{B\Sigma} = A'|_{B\Sigma}$ and C_2 is empty
$\Rightarrow$ $A|_{D\Sigma} = A'|_{D\Sigma}$, i.e., all $f : s \to e \in C\Sigma$ have unique interpretations in $A|_{C\Sigma}$:
f^A is **inductively defined** on $A|_{C\Sigma}$. (1)

$A|_{D\Sigma} = A'|_{D\Sigma}$ is final in $Alg_{D\Sigma}$ and $A|_{B\Sigma} = A'|_{B\Sigma}$
$\Rightarrow$ $A|_{C\Sigma} = A'|_{C\Sigma}$, i.e., all $f : e \to s \in C\Sigma$ have unique interpretations in $A|_{D\Sigma}$:
f^A is **coinductively defined** on $A|_{D\Sigma}$.
Moreover, $T_{C\Sigma} \in Alg_{D\Sigma}$, $coT_{D\Sigma} \in Alg_{C\Sigma}$ and $fold^{coT_{D\Sigma}} = unfold^{T_{C\Sigma}}$. (2)



Example 1 $D\Sigma = Stream(X)$

$$\begin{array}{rcll}\Psi & = & (& \{stream\}, \{X, 1\}, \emptyset, \\ & & & \{evens, odds, exchange, exchange' : stream \to stream\}, \\ & & & \{head : stream \to X,\ tail : stream \to stream\}\ ).\end{array}$$

The equations

$$\begin{array}{rclrcl} head(evens(s)) & = & head(s), & tail(evens(s)) & = & evens(tail(tail(s))), \\ head(odds(s)) & = & head(tail(s)), & tail(odds(s)) & = & odds(tail(tail(s))), \\ head(exchange(s)) & = & head(tail(s)), & tail(exchange(s)) & = & exchange'(s), \\ head(exchange'(s)) & = & head(s), & tail(exchange'(s)) & = & exchange(tail(tail(s))) \end{array}$$

form a recursive system of Ψ-equations. (Klin's coGSOS?)

$evens(s)$ und $odds(s)$ list the elements of s at even resp. odd positions.
$exchange(s)$ exchanges the elements at even positions with those at odd positions.

(2) $\Rightarrow$ $evens, odds, exchange, exchange'$ have unique interpretations in the final $D\Sigma$-algebra.

Example 2 $Reg(CS) + Acc(X)$

where the sort reg is substituted for $state$ and $X = \bigcup CS$

$$\begin{aligned}\Psi \;=\; (\;& \{reg\}, \{CS, X, 2\}, \\ & \{\in : X \times CS \to 2,\; max : 2 \times 2 \to 2,\; * : 2 \times 2 \to 2\}, \\ & \{eps, mt : 1 \to reg,\; con : CS \to reg,\; par, seq : reg \times reg \to reg, \\ & \;\; star : reg \to reg\}, \\ & \{\delta : reg \to reg^X,\; \beta : reg \to 2\} \;).\end{aligned}$$

The equations

$$\begin{aligned}
\delta(eps) &= \lambda x.mt, \\
\delta(mt) &= \lambda x.mt, \\
\delta(con(C)) &= \lambda x.ite(x \in C, eps, mt), \\
\delta(par(t,u)) &= \lambda x.par(\delta(t)(x), \delta(u)(x)), \\
\delta(seq(t,u)) &= \lambda x.par(seq(\delta(t)(x), u), ite(\beta(t), \delta(u)(x), mt)), \\
\delta(iter(t)) &= \lambda x.seq(\delta(t)(x), iter(t)), \\
\beta(eps) &= 1, \\
\beta(mt) &= 0,
\end{aligned}$$

$$\begin{array}{rcl} \beta(con(C)) & = & 0, \\ \beta(par(t,u)) & = & max\{\beta(t), \beta(u)\}, \\ \beta(seq(t,u)) & = & \beta(t) * \beta(u), \\ \beta(iter(t)) & = & 1 \end{array}$$

form the recursive system *BRE* of Ψ-equations, called **Brzozowski equations**.

(1) ⇒ δ, β have unique interpretations in the initial $Reg(CS)$-algebra $T_{Reg(CS)}$.
They form the $Acc(X)$-Algebra $Bro(CS)$, called the **Brzozowski automaton**.

(2) ⇒ $eps, mt, con, par, seq, star$ have unique interpretations in the final $Acc(X)$-algebra $Beh(X, 2)$.
They form the $Reg(CS)$-algebra $Lang(X)$ of (characteristic functions of) languages over X that is defined as follows:

For all $C \in CS$ and $L, L' \subseteq X^*$,

$$\begin{array}{lcl}
Lang(X)_{reg} & = & 2^{X^*}, \\
eps^{Lang(X)} & = & \{\epsilon\}, \\
mt^{Lang(X)} & = & \emptyset, \\
con^{Lang(X)}(C) & = & C, \\
par^{Lang(X)}(\chi(L), \chi(L')) & = & \chi(L \cup L'), \\
seq^{Lang(X)}(\chi(L), \chi(L')) & = & \chi(\{vw \mid v \in L, w \in L'\}), \\
iter^{Lang(X)}(\chi(L)) & = & \chi(\{w_1 \dots w_n \mid w_1, \dots, w_n \in L,\ n > 0\} \cup \{\epsilon\}).
\end{array}$$

(2) $\Rightarrow$ $fold^{Lang(X)} = unfold^{Bro(CS)}$
$\Rightarrow$ For all $t \in T_{Reg(CS)}$, $(Bro(CS), t)$ accepts the **language of** t, $fold^{Lang}(t)^{-1}(1)$.

The Brzozowski automaton becomes more efficient if its states ($= Reg(CS)$-terms) are normalized between transitions with respect to the semiring axioms, which $Lang(X)$ satisfies. δ is interpreted accordingly:

For all $t \in T_{Reg(CS)}$, $\delta^{Norm(CS)}(t) = reduce \circ \delta^{Bro(CS)}(t)$. ❑

Let $\Psi = (S, BS, B, C, D)$ be a bisignature, $D\Sigma = (S, BS, D)$ and A be a $D\Sigma$-algebra.

Given an S-sorted relation $\sim$ on A, the C**-equivalence closure** $\sim_C$ is the least S-sorted equivalence relation on A that contains $\sim$ and satisfies the following condition: For all $c : e \to s \in C$ and $a, b \in A_e$,

$$a \sim_C b \quad \text{implies} \quad c^A(a) \sim_C c^A(b).$$

An S-sorted relation $\sim$ on A is a $D\Sigma$**-congruence up to** C if for all $d : s \to e \in D$ and $a, b \in A_s$,

$$a \sim b \quad \text{implies} \quad d^A(a) \sim_C d^A(b).$$

$\sim$ is a $D\Sigma$-congruence up to C, A is final in $Alg_{D\Sigma}$ and there is a recursive system of Ψ-equations
$\Rightarrow$ $\sim_C$ is a $D\Sigma$-congruence. (3)

Example

Let Ψ be as in Example 2, $V = \{x, y, z\}$,

$t = seq(x, par(y, z))$ and $t' = par(seq(x, y), seq(x, z))$.

$$\sim \;\; = \;\; \{(g^*(t), g^*(t')) \mid g : T_{Reg(CS)}(V) \to Beh(X, 2)\}$$

is an $Acc(X)$-congruence up to C.

$\Rightarrow$ Since $Beh(X, 2)$ is final in $Alg_{Acc(X)}$, (3) implies that $\sim_C$ is $Acc(X)$-congruence.

$\Rightarrow$ Since $Beh(X, 2)$ satisfies the coinduction principle, $\sim \subseteq \Delta_{Beh(X,2)}$ and thus

$$Beh(X, 2) \;\; \models \;\; t = t'.$$

❏

As a recursive system E of Ψ-equations defines both

- destructors on constructors inductively and
- constructors on destructors coinductively,

so do the rules of a **transition system specification** or **structural operational semantics** (SOS) or the natural transformations λ that are called **distributive laws**: They all provide both

- an inductive definition of a semantics (destructors) of the syntax (constructors) of some language and
- a coinductive definition of the constructors on the language's behavioral model.

Ψ-algebras satisfying E correspond to **λ-bialgebras**.

The types of inductively or coinductively defined functions that come as unique solutions of recursive systems of Ψ-equations are those of destructors resp. constructors.

(Co)Recursion schemas that define functions with other types have been studied mainly in category-theoretical settings like distributive laws or adjunctions. For instance, in *Adjoint Folds and Unfolds*, Hinze obtains the desired types by characterizing the functions as (co)extensions of initial resp. final morphisms with respect to suitable adjunctions.

Future work: We think that most examples investigated in category-theoretical settings can be presented as recursive systems of Ψ-equations. For some of them, it might be necessary to generalize the schema, others will match the schema already because of the power of our term language that involves polynomial and even non-polynomial types.

Some application areas where (co)inductive definability has been studied in detail:

- basic process algebra
  ➾ Rutten, *Processes as Terms: Non-well-founded Models for Bisimulation*
- stream expressions and infinite sequences
  ➾ Rutten, *A Coinductive Calculus of Streams*
- tree expressions and infinite trees
  ➾ Silva, Rutten, *A Coinductive Calculus of Binary Trees*
- arithmetic expressions and valuations, CCS and transition trees
  ➾ Hutton, *Fold and Unfold for Program Semantics*
- stream function expressions and causal stream functions
  ➾ Hansen, Rutten, *Symbolic Synthesis of Mealy Machines from Arithmetic Bitstream Functions*

Iterative equations

Let $\Sigma = (S, BS, F)$ be a constructive signature and V be an S-sorted set.

An S-sorted function $E : V \to T_\Sigma(V)$ with $img(E) \cap V = \emptyset$ is called an **iterative system of Σ-equations**.

Let V be an S-sorted set, A be a Σ-algebra and A^V be the set of S-sorted functions

$g \in A^V$ **solves** E **in** A if $g^* \circ E = g$.

Iterative equations have unique solutions in the set CT_Σ of (continuous) **Σ-trees**:

CT_Σ denotes the greatest $\mathbb{FT}(S, BS)$-sorted set of prefix closed partial functions

$$t : \mathbb{N}^* \multimap F + \{word, bag, set\} + \cup BS$$

such that

- for all $s \in S$ and $t \in CT_{\Sigma,s}$, $def(t) = \emptyset$ or there are $n > 0$ and $e_1, \dots, e_n \in \mathbb{FT}(S, BS)$ with $t(\epsilon) : e_1 \times \dots \times e_n \to s \in F$, $def(t) \cap \mathbb{N} \subseteq [n]$ and $\lambda w.t(iw) \in CT_{\Sigma,e_i}$ for all $1 \le i \le n$,

- for all $c \in \{word, bag, set\}$, $s \in S \cup BS$ and $t \in CT_{\Sigma,c(s)}$ there is $n_t \in \mathbb{N}$ with $t(\epsilon) = c$, $def(t) \cap \mathbb{N} = [n_t]$ and $\lambda w.t(iw) \in CT_{\Sigma,s}$ for all $1 \leq i \leq n_t$,
- for all $X \in BS$, $CT_{\Sigma,X} = X$ (again identified with the set $1 \to X$ of functions).

Let $\sim$ be the greatest $\mathbb{FT}(S, BS)$-sorted equivalence relation on CT_Σ such that

- for all $s \in S$ and $t \sim_s t'$, $t(\epsilon) = t'(\epsilon)$ and for all $i \in \mathbb{N}$, $\lambda w.t(iw) \sim \lambda w.t'(iw)$,
- for all $s \in S \cup BS$ and $t \sim_{word(s)} t'$, $n_t = n_{t'}$ and for all $i \in [n_t]$, $\lambda w.t(iw) \sim_s \lambda w.t'(iw)$,
- for all $s \in S \cup BS$, $t \sim_{bag(s)} t'$ and $f : [n_t] \hookrightarrow [n_t]$, $n_t = n_{t'}$ and for all $i \in [n_t]$, $\lambda w.t(f(i)w) \sim_s \lambda w.t'(iw)$,
- for all $s \in S \cup BS$, $t \sim_{set(s)} t'$, $i \in [n_t]$ and $j \in [n_{t'}]$ there are $k \in [n_{t'}]$ and $l \in [n_t]$ such that $\lambda w.t(iw) \sim_s \lambda w.t'(kw)$ and $\lambda w.t(lw) \sim_s \lambda w.t'(jw)$,
- for all $X \in BS$, $\sim_X = \Delta_X$.

For simplicity, we identify CT_Σ with $CT_\Sigma/\sim$.

$\boxed{CT_\Sigma \text{ is a } \Sigma\text{-algebra:}}$ For all $f : e \to s \in F$, $t = (t_1, \dots, t_n) \in CT_{\Sigma,e}$ and $w \in \mathbb{N}^*$,

$$f^{CT_\Sigma}(t)(w) \;=_{def} \begin{cases} f & \text{if } w = \epsilon, \\ t_i(v) & \text{if } \exists\, i \in \mathbb{N} : iv = w. \end{cases}$$

Usually, $f^{CT_\Sigma}(t)$ is written as ft.

$\boxed{CT_\Sigma \text{ is initial in } CAlg_\Sigma,}$ the category of ω-continuous Σ-algebras as objects and strict and ω-continuous Σ-homomorphisms.

➾ Goguen et al., *Initial Algebra Semantics and Continuous Algebras*, 1978

The constructive signature Σ induces a destructive signature $co\Sigma$ such that $H_\Sigma = H_{co\Sigma}$

$$\begin{aligned} co\Sigma \;=\; (S,\; & \{d_s : s \to \textstyle\coprod_{f:e\to s\in F} e \mid s \in S\} \cup \\ & \{\pi_i : e_1 \times \dots \times e_n \to e_i \mid n > 1,\; e_1, \dots, e_n \in \mathbb{FT}(S, BS),\; 1 \le i \le n\}) \end{aligned}$$

$\boxed{CT_\Sigma \text{ is a } co\Sigma\text{-algebra:}}$ For all $s \in S$ and $t \in CT_{\Sigma,s}$,

$$d_s^{CT_\Sigma}(t) \;=_{def}\; ((\lambda w.t(1w), \dots, \lambda w.t(nw)), t(\epsilon))$$

where $n = |dom(t(\epsilon))|$.

CT_Σ is final in $Alg_{co\Sigma}$.

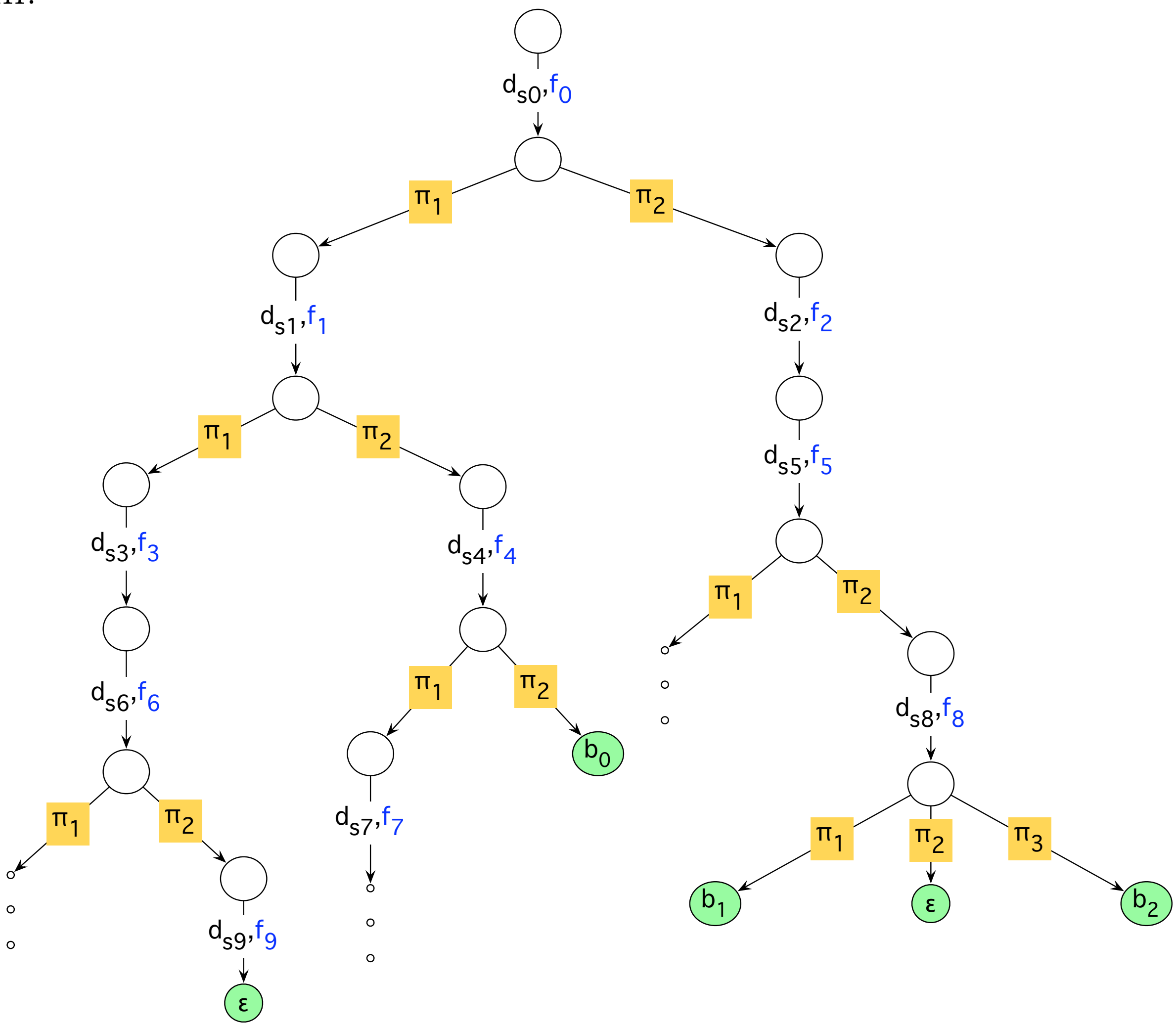
For all $co\Sigma$-algebras A, the unique Σ-homomorphism $unfold^A : A \to CT_\Sigma$ is defined as follows: For all $s \in S$, $a \in A_s$, $i \in \mathbb{N}$ and $w \in \mathbb{N}^*$,

$$\begin{array}{rcl} unfold^A(a)(\epsilon) & = & f, \\ unfold^A(a)(iw) & = & \begin{cases} unfold^A(a_i)(w) & \text{if } i \in [n], \\ \text{undefined} & \text{otherwise,} \end{cases} \end{array}$$

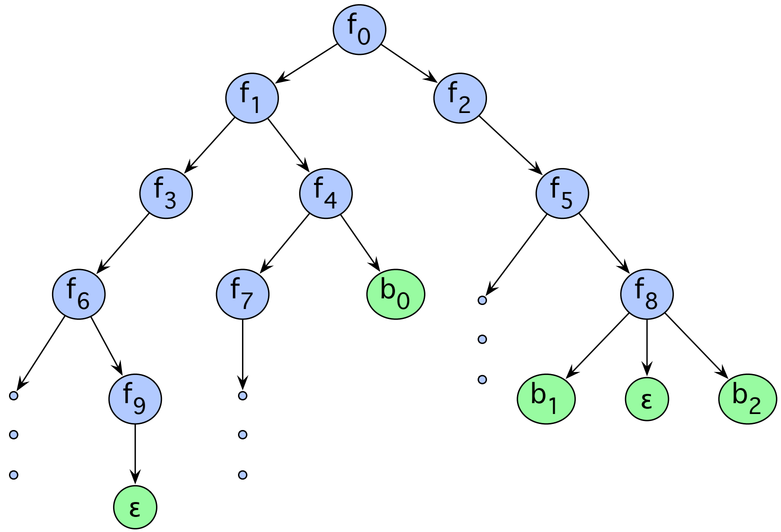
where $d_s^A(a) = ((a_1, \ldots, a_n), f)$.

Indeed, $CT_\Sigma \cong coT_{co\Sigma}$.

A $co\Sigma$-coterm:



... and the corresponding Σ-tree:



The $co\Sigma$-algebra $T_\Sigma^E(V)$ is defined as follows:

- For all $s \in S$, $T_\Sigma^E(V)_s = T_\Sigma(V)_s$.
- For all $f : e \to s \in F$ and $t \in T_\Sigma(V)_e$, $d_s^{T_\Sigma^E(V)}(ft) = (t, f)$.
- For all $x \in V$, $f : e \to s \in F$ and $t \in T_\Sigma(V)_e$, $E(x) = ft$ implies $d_s^{T_\Sigma^E(V)}(x) = (t, f)$.

$$unfold^{T_\Sigma^E(V)} \circ inc_V : V \to CT_\Sigma \text{ solves } E \text{ in } CT_\Sigma. \qquad (4)$$

Context-free grammars with base sets

A **context-free grammar** (CFG) $G = (S, BS, Z, R)$ (with base sets) consists of

- a set S of **nonterminals**,
- a set BS of **base sets**,
- a set Z of **terminals**,
- a set R of **rules** $s \to w$ with $s \in S$ and $w \in (S \cup Z \cup BS)^*$.

The **abstract syntax of** G is the constructive signature $\Sigma(G) = (S, BS, F)$ with

$$\begin{aligned} F \;=\; \{f_r : e_{i_1} \times \ldots \times e_{i_k} \to s \mid \; & r = (s \to e_1 \ldots e_n) \in R,\; s \in S, \\ & \{i_1, \ldots, i_k\} = \{1 \le i \le n \mid e_i \in S \cup BS\}\}. \end{aligned}$$

Let $CS = BS \cup \{\{z\} \mid z \in Z\}$ and $X = \bigcup CS$.

The $\Sigma(G)$**-word algebra** $Word(G)$ recovers the concrete from the abstract syntax:

For all $s \in S$, $Word(G)_s = X^*$. For all $r = (s \to e_1 \dots e_n) \in R$,

$$\begin{aligned} f_r^{Word(G)} : (X^*)^k &\to X^* \\ (w_{i_1}, \dots, w_{i_k}) &\mapsto v_1 \dots v_n \end{aligned}$$

where for all $1 \leq i \leq n$,

$$v_i =_{def} \begin{cases} w_i & \text{if } i \in \{i_1, \dots, i_k\} = \{1 \leq i \leq n \mid e_i \in S \cup BS\}, \\ e_i & \text{otherwise.} \end{cases}$$

The **language** $L(G)$ **of** G is the image of $T_{\Sigma(G)}$ under $fold^{Word(G)}$:

$$L(G) =_{def} img(fold^{Word(G)}).$$

For all $t_1, \ldots, t_n \in T_{Reg(CS)}(S)$,

$$\begin{array}{rcl} \sum_{i=1}^{n} t_i & =_{def} & par(t_1, par(t_2, \ldots, t_n)), \\ \prod_{i=1}^{n} t_i & =_{def} & seq(t_1, seq(t_2, \ldots, t_n)). \end{array}$$

G induces an iterative system of $Reg(CS)$-equations:

$$\begin{array}{rcl} E_G : S & \rightarrow & T_{Reg(CS)}(S) \\ s & \mapsto & \sum_{i=1}^{k} \overline{w_i} \end{array}$$

where

$$\{w_1, \ldots, w_k\} = \{w \in (S \cup CS)^* \mid s \rightarrow w \in R\},$$

and for all $n > 1$, $e_1, \ldots, e_n \in S \cup CS$, $s \in S$ and $C \in CS$,

$$\begin{array}{rcl} \overline{e_1 \ldots e_n} & = & \sum_{i=1}^{k} \overline{e_i}, \\ \overline{s} & = & s, \\ \overline{C} & = & con(C). \end{array}$$

$$\begin{array}{rcl} lang(G) : S & \rightarrow & 2^{X^*} \\ s & \mapsto & \chi(L(G)_s) \end{array}$$

is the least solution of E_G in $Lang(X)$.

How to translate and add iterative to recursive equations

Let $C\Sigma = (S, BS, C)$, $D\Sigma = (S, BS', D)$, $\Psi = (S, BS \cup BS', B, C, D)$ be a bisignature, $E : V \to T_{C\Sigma}(V)$ be an iterative system of $C\Sigma$-equations,

$$\begin{aligned} C_V &= \{var_s : V_s \to s \mid s \in S\}, \\ \Psi_V &= (S, BS \cup BS' \cup V, B, C \cup C_V, D), \end{aligned}$$

V' be an S-sorted set of variables, E' be a recursive system of Ψ_V-equations over V' and

$$E'_V = \{d(c(x)) = t \in E' \mid d \in D,\ c \in C_V,\ x \in V\}$$

such that $E' \setminus E'_V$ is a recursive system of Ψ-equations.

There is at most one solution of E in every Ψ-algebra A that is final in $Alg_{D\Sigma}$ and satisfies E' whenever var^A solves E in A. (5)

Proof. Let A be a Ψ-algebra such that $A|_{D\Sigma}$ is final in $Alg_{D\Sigma}$. Suppose that $g, h : V \to A$ solve E in A. We extend A to Ψ_V-algebras A_1, A_2 by defining $var^{A_1} = g$ and $var^{A_2} = h$. By assumption, both A_1 and A_2 satisfy E'_V. By (3), $A_1 = A_2$. Hence $g = var^{A_1} = var^{A_2} = h$. ❑

Example 3 $\Sigma + co\Sigma$

Let $\Sigma = (S, BS, C)$ be a constructive signature, $co\Sigma = (S, BS, D)$, $E : V \to T_\Sigma(V)$ be an iterative system of Σ-equations, C_V and Ψ_V be defined as above,

$$\begin{aligned} E' &= \{d_s(fx) = \iota_f(x) \mid s \in S,\ f : e \to s \in C,\ x \in V_e\}, \\ E'_V &= \{d_s(var_s(x)) = \iota_f(\sigma^*(t)) \mid s \in S,\ x \in V_s,\ E(x) = ft\} \end{aligned}$$

where for all $s \in S$ and $x \in V_s$, the substitution σ assigns the term $var_s(x)$ to x.

Let A be a Ψ_V-algebra with $A|_{\Sigma+co\Sigma} = CT_\Sigma$ such that var^A solves E in A.

$\boxed{A \text{ satisfies } E'\text{:}}$ For all S-sorted functions $g : V' \to CT_\Sigma$,

$$g^*(d_s(fx)) = d_s^A(f^A(g^*(x))) = d_s^A(f(g^*(x))) = (g^*(x), f) = g^*(\iota_f(x)).$$

$\boxed{A \text{ satisfies } E'_V\text{:}}$ Let $s \in S$, $x \in V_s$ and $E(x) = ft$. Since var^A solves E in A,

$$var_s^A(x) = (var_s^A)^*(E(x)) = f^A((var_s^A)^*(t)). \tag{6}$$

Moreover,

$$g^*(\sigma(x)) = g^*(var(x)) = var^A(g^*(x)) = var^A(x) \tag{7}$$

because, within E'_V, x is a base element and not a variable!

By (6) and (7), for all S-sorted functions $g : V \to CT_\Sigma$,

$$g^*(d_s(var_s(x))) = d_s^A(var_s^A(x)) \stackrel{(6)}{=} d_s^A(f^A((var_s^A)^*(t))) = d_s^A(f((var_s^A)^*(t)))$$
$$= ((var_s^A)^*(t), f) \stackrel{(7)}{=} (g^* \circ \sigma)^*(t), f) = (g^*(\sigma^*(t)), f) = g^*(\iota_f(\sigma^*(t))).$$

Hence A satisfies $E' \cup E'_V$ and thus by (4) and (5), E has a unique solution in A. ❑

Example 4 $Reg(CS) + Acc(X)$ (with reg for $state$) + G + acceptor of $L(G)$

Let $G = (S, BS, Z, R)$ be a **non-left-recursive** CFG (excludes $s \stackrel{+}{\longrightarrow}_G sw$) and $reduce$ be a function that simplifies regular expressions by applying semiring axioms.

Let $CS = BS \cup \{\{z\} \mid z \in Z\}$. For all $s \in S$ there are $k_s, n_s > 0$, $C_{s,i} \in CS$, $t_{s,i} \in T_{Reg(CS)}(S)$, $1 \leq i \leq n_s$, such that

$$reduce((E_G^*)^{k_s}(s)) \;\in\; \{t_s, par(t_s, eps)\}$$

where $t_s = \sum_{i=1}^{n_s} seq(con(C_{s,i}), t_{s,i})$.

Let $X = \bigcup CS$, $\Psi = (S, BS, B, C, D)$ be as in Example 2,

$$C_S \;=\; \{var : S \to reg \mid s \in S\},$$
$$\Psi_S \;=\; (S, BS \cup \{S\}, B, C \cup C_S, D),$$

$$E'_G = \{\delta(var(s)) = \lambda x. \sum_{i=1}^{n_s} ite(x \in C_{s,i}, \sigma^*(t_{s,i}), mt) \mid s \in S\} \cup \{\beta(var(s)) = u_s \mid s \in S\}$$

where the substitution σ replaces each $s \in S$ by the term $var(s)$ and

$$u_s = \begin{cases} 0 & \text{if } reduce((E_G^*)^{k_s}) = t_s, \\ 1 & \text{if } reduce((E_G^*)^{k_s}) = par(t_s, eps). \end{cases}$$

Let A be a Ψ_S-algebra with $A|_{Reg(CS)+Acc(X)} = Lang(X)$ such that var^A solves E_G in A. In Example 2, we have seen that A satisfies BRE.

A satisfies E'_G: Let $s \in S$. Since var^A solves E_G in A,

$$var^A(s) = (var^A)^*(s) = (var^A)^*((E_G^*)^{k_s}(s)) = (var^A)^*(reduce((E_G^*)^{k_s}(s)))$$
$$= \{(var^A)^*(\textstyle\sum_{i=1}^{n_s} seq(con(C_{s,i}), t_{s,i})),\ (var^A)^*(par(\textstyle\sum_{i=1}^{n_s} seq(con(C_{s,i}), t_{s,i}), eps))\}$$

**** If $var^{Lang} : S \to Lang(X)$ is a solution of E_G in $Lang(X)$, then the unique coinductive solution of BRE in $Beh(X,2)$ is extended to a coinductive solution of $BRE \cup E'(G)$.

If G is non-left-recursive, then $Lang(X)$ contains $lang(G)$ is the only solution of E_G in $Beh(X,2)$.

$\Rightarrow$ As BRE provided an acceptor for every regular language, $BRE \cup E'(G)$ yields an acceptor of every context-free language, given by a non-left-recursive CFG.

(Co-)Horn clauses

Let $\Sigma = (S, BS, F, P)$ and $\Sigma' = (S, BS, F, P \cup P')$ be signatures and C be a Σ-algebra.

$Alg_{\Sigma',C}$ denotes the full subcategory of Alg_Σ consisting of all Σ'-algebras A with $A|_\Sigma = C$.

$Alg_{\Sigma',C}$ is a complete lattice: For all $A, B \in Alg_{\Sigma',C}$,

$$A \leq B \;\Leftrightarrow_{def}\; \forall\, p \in P' : p^A \subseteq p^B.$$

For all $\mathcal{A} \subseteq Alg_{\Sigma',C}$ and $p : e \in P'$,

$$p^\perp = \emptyset, \;\; p^\top = A_e, \;\; p^{\sqcup\mathcal{A}} = \bigcup_{A\in\mathcal{A}} p^A \;\text{ and }\; p^{\sqcap\mathcal{A}} = \bigcap_{A\in\mathcal{A}} p^A.$$

A Σ'-formula φ is **negation-free w.r.t.** Σ if φ does not contain $\Rightarrow$, $\Leftarrow$ or $\Leftrightarrow$ and all subformulas of φ with a leading negation symbol belong to $Fo_\Sigma(V)$.

A **Horn clause for** P' is a Σ'-formula $p(t) \Leftarrow \varphi$ such that $p \in P'$ and φ is negation-free w.r.t. Σ.

Let AX be a set of Horn clauses for P'.

The AX**-step function** $\Phi : Alg_{\Sigma',C} \to Alg_{\Sigma',C}$ is defined as follows:

For all $A \in Alg_{\Sigma',C}$ and $p \in P'$,

$$p^{\Phi(A)} =_{def} \{g^*(t) \mid p(t) \Leftarrow \varphi \in AX,\ g \in \varphi^A\}.$$

Φ is monotone and thus by the Fixpoint Theorem of Knaster and Tarski, Φ has the least fixpoint

$$lfp(\Phi) = \sqcap \{A \in Alg_{\Sigma',C} \mid \Phi(A) \leq A\}.$$

Consequently,

$$lfp(\phi) \models p(x) \Leftrightarrow \bigvee_{p(t) \Leftarrow \varphi \in AX} \exists\, var(t,\varphi) : (x = t \wedge \varphi).$$

A **co-Horn clause for** P' is a Σ'-formula $p(t) \Rightarrow \varphi$ such that $p \in P'$ and φ is negation-free w.r.t. Σ.

Let AX be a set of co-Horn clauses for P'.

The AX**-step function** $\Phi : Alg_{\Sigma',C} \rightarrow Alg_{\Sigma',C}$ is defined as follows:

For all $A \in Alg_{\Sigma',C}$ and $p : e \in P'$,

$$p^{\Phi(A)} \;=_{def}\; C_e \backslash \{g^*(t) \mid pt \Rightarrow \varphi \in AX,\; g \in C^V \backslash \varphi^A\}.$$

Φ is monotone and thus by the Fixpoint Theorem of Knaster and Tarski, Φ has the greatest fixpoint

$$gfp(\Phi) \;=\; \sqcup \{A \in Alg_{\Sigma',C} \mid A \leq \Phi(A)\}.$$

Consequently,

$$gfp(\phi) \quad \models \quad p(x) \Leftrightarrow \bigwedge_{p(t) \Rightarrow \varphi \in AX} \forall\, var(t, \varphi) : (x \neq t \vee \varphi).$$

*** to be continued ***