From $\mathcal{H}$ to $\mathcal{K}$

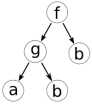
Internal Seminar – 7 January 2008

Herbrand Domain.

Let Σ be a signature (*e.g.* $\{f/2, g/2, h/1, a/0, b/0, \dots\}$) and $\forall X, \phi(X) = \left\{ f(x_1, \dots, x_n) \middle| f/n \in \Sigma, x_1, \dots x_n \in X \right\}$.

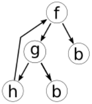
Definition (Inductive Terms)

$T_f(\Sigma)$ is the least fixpoint of ϕ.



Definition (Co-inductive Terms, Regular Terms)

$T_r(\Sigma)$ is the greatest fixpoint of ϕ.



Construction Rule

$T_f(\Sigma)$ and $T_r(\Sigma)$ are defined inductively (*resp.* coinductively) with respect to the following rule:

$$\frac{f/n \in \Sigma, x_1, \dots, x_n \in T(\Sigma)}{f(x_1, \dots, x_n) \in T(\Sigma)}$$

Term Comparisons over Herbrand Terms.

Definition (Lexicographic order)

Let $<$ be an order over Σ. There exists a unique order $<_{\text{lex}}$ over $T_f(\Sigma)$ or $T_r(\Sigma)$ which is a lexicographic extension of $<$:

$$\frac{f/n < g/m}{f(x_1, \ldots, x_n) <_{\text{lex}} g(y_1, \ldots, y_m)}$$

$$\frac{x_1 = y_1 \land \cdots \land x_i = y_i \land x_{i+1} <_{\text{lex}} y_{i+1}}{f(x_1, \ldots, x_n) <_{\text{lex}} f(y_1, \ldots, y_n)}$$

Proposition

If Σ is countable, $=$ over terms and $<$ over Σ is not enough to test $<_{\text{lex}}$.

Prolog Predicates over Terms.

$$X <_{\text{lex}} Y \text{ :- } \mathbf{functor}(X, F, N), \mathbf{functor}(Y, G, M), F/N < G/M.$$

$$X <_{\text{lex}} Y \text{ :- } \mathbf{functor}(X, F, N), \mathbf{functor}(Y, F, N), \mathrm{argleq}(0, N, X, Y).$$

$$\mathrm{argleq}(I, N, X, Y) \text{ :- } I < N, \mathbf{arg}(X, I, Xi), \mathbf{arg}(Y, I, Yi), (Xi <_{\text{lex}} Yi; Xi = Yi, J \textbf{ is } I + 1, \mathrm{argleq}(J, N, X, Y))$$

My Wish

Equality plus two constraints:

- **functor** to go back and forth between signatures and terms.
- **arg** to access a subterm without fully specifying the parent signature.

Herbrand Terms as Maps.

Intuition.

With **arg** constraint, terms are colored arrays of terms and **arg**(X, I, Y) should be written $X[I] = Y$.

Expressiveness Note.

Sets with equality and membership constraints can be reduced to terms with equality and **arg** contraints.

$X \in Y$:– $Y[\_] = X$.

functor gives the cardinality constraint.

Just One Step From Maps.

Should $X[N] = Y$ imply that N is an integer?

Map Constraints.

Association (A).

Ternary constraint: $X[Z] = Y$ where Z is the *key* and Y is the *value*.

Connection to Type Theory: Open Rows and Projection.

Enumeration (E).

$X = \{Z_1 : Y_1; Z_2 : Y_2; \ldots; Z_n : Y_n\}$. $(Z_1, \ldots, Z_n)$ is called the *map support*.

Connection to Type Theory: Closed Rows.

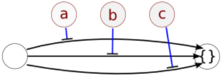
Some Sets Constraints.

Using the encoding $X \in Y :\!-\, X[Y] = \{\}$ as intuition, one can define $\cup$, $\cap$ and $\setminus$ as operations over map supports (*e.g.* with value to value equality for common keys).

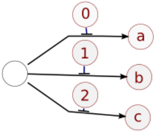
Connection to Type Theory: Row Concatenation, Absent and Private Rows

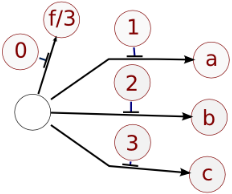
Maps as Association Graphs.

- $\{a, b, c\} \triangleq \{a : \{\}; b : \{\}; c : \{\}\}$.



- $\llbracket a, b, c \rrbracket \triangleq \{0 : a; 1 : b; 2 : c\}$.





- $f(a, b, c) \triangleq \llbracket f, a, b, c \rrbracket$.
- $X[X] = X$.

Association Graphs.

Definition (Composition Law)

A *magma* is a pair $G = (E, \cdot)$ where $\cdot$ is an intern composition law of E, $\cdot : E \times E \to E$.

Definition (Association Graph)

An *association graph* $G = (E, \cdot, \bot)$ is a magma such that:

- $\bot \cdot x = x \cdot \bot = \bot$ $(\forall x)$.
- the support $s(x) = \left\{ y \middle| x \cdot y \neq \bot \right\}$ is finite $(\forall x)$.

Let $x[y] \stackrel{\wedge}{=} x \cdot y$.

Proposition

Interpreted on formulae upon $\langle \exists, =, \neq, A \rangle$ theory, magma and association graph domains are equivalent.

Intuition

$$X = \{z_1 : y_1; \dots ; z_n : y_n\} :- \; X[z_i] = y_i,$$
$$(\forall z', z' \neq z_1 \wedge \cdots \wedge z' \neq z_n \to X[z'] = \bot).$$

Constraints over Association Graphs.

Association and Enumeration.

$$\begin{array}{ll} \text{(A)} & X[Z] = Y.\ (\textit{Path follow.}) \\ \text{(E)} & X = \{Z_1 : Y_1; \ldots; Z_n : Y_n\}.\ (\textit{Support.}) \end{array}$$

Support and cardinality.

- For constraint **functor**(X, F, N), $X = F(x_1, \ldots, x_N)$, with the encoding $X = [\![F/N, x_1, \ldots, x_N]\!]$,
  functor(X, F, N) :– $X[0] = F/N$.
- For arrays $X = [\![X_0, \ldots, X_N]\!] \stackrel{\wedge}{=} \{0 : X_0; \ldots; N : X_N\}$,
  length$(X) = N + 1$:– $X[N] \neq \bot \wedge X[N+1] = \bot$.
- For sets $X = \{X_0, \ldots, X_N\} \stackrel{\wedge}{=} \{X_0 : \{\}; \ldots; X_N : \{\}\}$,
  card(X) is not derived from (A, E). With enumeration constraints, cardinality is syntactically fixed:

$$\mathbf{card}(X) \leq n \text{ :– } X = \left\{ \underbrace{\_, \_, \ldots, \_}_{n \text{ times}} \right\}$$

Merry Christmas!

- Prolog-like term processing with constraints.
- Data structures:
  1. Associative tables from terms to terms.
  2. Arrays.
  3. Sets.
  4. Records.
  5. Automatas.
- How results over tree data structures are generalizable to maps?