From $H$ to $K$

Internal Seminar – 7 January 2008

Constraints over Herbrand Terms.
  Herbrand Domain.
  Term Comparisons over Herbrand Terms.
  Prolog Predicates over Terms.

From Maps to Graphs.
  Herbrand Terms as Maps.
  Map Constraints.
  Maps as Association Graphs.

Association Graphs and Constraints.
  Association Graphs.
  Constraints over Association Graphs.
Herbrand Domain.

Let $\Sigma$ be a signature (e.g. $\{f/2, g/2, h/1, a/0, b/0, \ldots \}$) and
$\forall X, \phi(X) = \{f(x_1, \ldots, x_n) \mid f/n \in \Sigma, x_1, \ldots x_n \in X\}$.

Definition (Inductive Terms)

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

Definition (Co-inductive Terms, Regular Terms)

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

Construction Rule

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

$$f/n \in \Sigma, x_1, \ldots, x_n \in T(\Sigma) \quad \frac{f(x_1, \ldots, x_n) \in T(\Sigma)}{f(x_1, \ldots, x_n) \in T(\Sigma)}$$
Definition (Lexicographic order)

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

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

Proposition

*If $\Sigma$ is countable, $=$ over terms and $<$ over $\Sigma$ is not enough to test $<_\text{lex}$.***
Prolog Predicates over Terms.

\[ X <_{\text{lex}} Y \leftarrow \text{functor}(X, F, N), \text{functor}(Y, G, M), F/N < G/M. \]

\[ X <_{\text{lex}} Y \leftarrow \text{functor}(X, F, N), \text{functor}(Y, F, N), \text{argleq}(0, N, X, Y). \]

\[ \text{argleq}(l, N, X, Y) \leftarrow l < N, \text{arg}(X, l, Xi), \text{arg}(Y, l, Yi), (Xi <_{\text{lex}} Yi; Xi = Yi, J \text{ is } l + 1, \text{argleq}(J, N, X, Y)) \]

My Wish

Equality plus two constraints:

- \textbf{functor} to go back and forth between signatures and terms.
- \textbf{arg} to access a subterm without fully specifying the parent signature.
Herbrand Terms as Maps.

Intuition.
With \texttt{arg} constraint, terms are colored arrays of terms and \texttt{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 \texttt{arg} constraints.

\[
X \in Y \,:\, \ Y[\_] = X.
\]

\texttt{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 $X \in Y := X[Y] = \emptyset$ 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\} \models \{a : \{\}; b : \{\}; c : \{\}\}.

- \langle a, b, c \rangle \models \{0 : a; 1 : b; 2 : c\}.

- f(a, b, c) \models \langle f, a, b, c \rangle.

- X[X] = X.
Association Graphs.

Definition (Composition Law)

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

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 $s(x) = \{y | x \cdot y \neq \bot\}$ is finite (\forall x).

Let $x[y] = x \cdot y$.

Proposition

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

Intuition

$X = \{z_1 : y_1; \ldots ; z_n : y_n\} \vdash X[z_i] = y_i,$

$(\forall z', z' \neq z_1 \land \cdots \land z' \neq z_n \rightarrow X[z'] = \bot).$
Constraints over Association Graphs.

Association and Enumeration.

(A) $X[Z] = Y$. (*Path follow.*)

(E) $X = \{Z_1 : Y_1; \ldots; Z_n : Y_n\}$. (*Support.*)

Support and cardinality.

- For constraint $\text{functor}(X, F, N), X = F(x_1, \ldots, x_N)$, with
  the encoding $X = \llbracket F/N, x_1, \ldots, x_N \rrbracket$, \n  $\text{functor}(X, F, N) :- X[0] = F/N$.

- For arrays $X = \llbracket X_0, \ldots, X_N \rrbracket \triangleright \{0 : X_0; \ldots; N : X_N\}$, \n  $\text{length}(X) = N + 1 :- X[N] \neq \bot \land X[N + 1] = \bot$.

- For sets $X = \{X_0, \ldots, X_N\} \triangleright \{X_0 : \{\}; \ldots; X_N : \{\}\}$, \n  $\text{card}(X)$ is not derived from $(A, E)$. With enumeration
  constraints, cardinality is syntactically fixed:

  $\text{card}(X) \leq n :- X = \underbrace{\{\_,\_,\ldots,\_\}}_{n \text{ times}}$
Merry Christmas!

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