Structural Iteration

- $\overline{f}(VAR\ a) = f_{VAR}(a)$
- $\overline{f}(APP\ t_1\ t_2) = f_{APP}(\overline{f}(t_1), \overline{f}(t_2))$
- $a \notin A \Rightarrow \overline{f}(LAM\ a\ t) = f_{LAM}(a, \overline{f}(t))$

But from this we get by the usual coding-trick
the structural recursion principle.

Support and Supports

- $supp\ x \stackrel{\text{def}}{=}$
  $$\{a : \alpha \mid \text{infinite}\{b \mid (a\ b)\bullet x \neq x\}\}$$
- $a \# x \stackrel{\text{def}}{=} a \notin supp(x)$
- $X\ supports\ x \stackrel{\text{def}}{=}$
  $$\forall a, b \notin X.\ \ (a\ b)\bullet x = x$$

Properties

- $(supp\ x)$ $supports\ x$, because

$$a \# x \ \wedge\ b \# x \ \Rightarrow\ (a\,b)\bullet x = x$$

- If S $supports\ x$ and S is a finite set, then

$$(supp\ x) \subseteq S$$

- If S $supports\ x$, S is a finite set and

$$\forall S'.\ (\text{finite } S' \wedge S'\ supports\ x) \Rightarrow S \subseteq S'$$

then $S = supp\ x$. (By first two props.)

Properties

- $(supp\ x)$ $supports\ x$, because

$$a \# x \ \wedge \ b \# x \ \Rightarrow \ (a\,b)\bullet x = x$$

- If S $supports\ x$ and S is a finite set, then

$$(supp\ x) \subseteq S$$

- If S $suppo$

$\forall S'.\ (\text{finite } S'$

then $S = su$

Assume the contrary, then

(1) $\exists a.\ a \in (supp\ x) \wedge a \notin S$
(2) $\forall b \notin S \ \Rightarrow \ (a\,b)\bullet x = x$
(3) $\{b \mid \ (a\,b)\bullet x \neq x\} \ \subseteq \ S$
(4) finite $\{b \mid \ (a\,b)\bullet x \neq x\}$
(5) $a \notin (supp\ x)$

Properties

- $(supp\ x)$ $supports$ x, because

$$a \# x \ \wedge \ b \# x \ \Rightarrow \ (a\,b)\bullet x = x$$

- If S $supports$ x and S is a finite set, then

$$(supp\ x) \subseteq S$$

- If S $supports$ x, S is a finite set and

$$\forall S'.\ (\text{finite } S' \wedge S'\ supports\ x) \Rightarrow S \subseteq S'$$

then $S = supp\ x$. (By first two props.)

Finitely Supported Types

- Atoms are finitely supported, so are tuples, lists finite sets provided they are build up from finitely supported types.
- Sets are in general not finitely supported:

$$(supp\ A) = \varnothing$$

because $\forall a\, b.\ (a\, b) \bullet A = A$. If we take an infinite subset A' of A whose co-set is also infinite, then

$$(supp\ A') = A'.$$

Functions

- The identity function (even if it mentions infinitely many atoms in its graph).

$$\emptyset \ \textit{supports}\ (\lambda x.x)$$

- Not every function over finitely supported domains is finitely supported. Trivial example: the set of atoms is countable; there are subjective functions $N \longrightarrow A$; every finitely supported such function must have a finite image.

Choice Function

- Any choice function for the set of finitely supported subsets of A, i.e. any function

$$choose \in (A \to_{\mathsf{fs}} bool) \to A,$$

satisfying

$$f(a) = \mathsf{true} \Rightarrow f(choose(f)) = \mathsf{true}$$

for all $f : A \to_{\mathsf{fs}} bool$ and every $a \in A$.

- $choose$ does not have a finite support.

Choice Function

- Any choice function for the set of finitely supported subsets of A, i.e. any function

satis

$f($ ... true

for all $f : A \rightarrow_{fs}$ *bool* and every $a \in A$.

The lack of a finite support for choice functions motivated Andrew Pitts and Jamie Gabbay to construct a permutation model inside the set theory by Freankel and Mostowski (as it turns out now with this work, this was a dead end).

- *choose* does not have a finite support.

Complaints about HOL

- I would like to define the type of finitely supported functions:

$$\texttt{typedef}\ \iota_1 \longrightarrow_{fs} \iota_2 = \{f \mid \exists S.\mathsf{finite}\ S \wedge S\ \mathit{supports}\ f\}$$

  this is all possible, but I cannot use Isabelle's λ to "write down" such functions.

$$f_{fs}(x) = Abs\_FS(\overline{f}(Rep\_FS\ x))$$

- I think one can make a case that there should be more research on HOL, overloading and ax-classes.