

$T ::= \text{NAT} \mid \text{TOP} \mid T \times T \mid T \rightarrow T \mid x \mid \mu x.T \mid \{l_i: T_i\} \mid \langle l_i: T_i \rangle$

$V ::= \dots \mid \text{FOLD}[T]v$

$\text{NATLIST} = u.x \langle \text{NIL}: \text{UNIT}, \text{CONS}: \{\text{NAT}, X\} \rangle$

$\text{UNFOLD}[ux.T] \Rightarrow [x \rightarrow ux.T]T$

FOLD does opposite

example of unfolding NATLIST once...

$\langle \text{NIL}: \text{UNIT}, \text{CONS}: \{\text{NAT}, \langle \text{NIL}: \text{UNIT}, \text{CONS}: \{\text{NAT}, X\} \rangle \} \rangle$

Example of type rule for a pair...

$\Gamma \vdash t_1: T_1$

$\Gamma \vdash t_2: T_2$

— — —

$\Gamma \vdash \{t_1, t_2\}: T_1 \times T_2$

Type rules involving FOLD and UNFOLD:

$t \rightarrow t'$

—————

$\text{UNFOLD}[T]t \rightarrow \text{UNFOLD}[T]t'$

$t \rightarrow t'$

—————

$\text{FOLD}[T]t \rightarrow \text{FOLD}[T]t'$

—————

$\text{UNFOLD}[T](\text{FOLD}[S]v) \rightarrow v$

$\Gamma \vdash t: \mu x.T \quad U = [x \mapsto \mu x.T]T$

—————

$\Gamma \vdash \text{UNFOLD}[\mu x.T]t: U$

$\Gamma \vdash t: U \quad U = [x \mapsto \mu x.T]T$

—————

$\Gamma \vdash \text{FOLD}[T]t: \mu x.T$

A suggestion for cons:

$\text{mk_nil} = \lambda_:\text{Unit}.\text{FOLD}[\text{NATLIST}] \langle \text{nil} = \text{unit} \rangle$

$\text{mk_nil} : \text{Unit} \rightarrow \text{Natlist}$

$\text{mk_cons} : \text{Nat} \times \text{Natlist} \rightarrow \text{Natlist}$

Greatest fixed point:

Fixed point is anywhere $\text{T}(\text{R})=\text{R}..$ now, R is a set.. greatest fixed point is largest set R for which this is true.. smallest fixed point is smallest set R for which this is true..

To get least fixed point, start out with empty set and add..

To get greatest fixed point, start with all sets and trim..

Q: is there any infinite tree in the least fixed point $\Gamma \dashrightarrow$ at the “limit” $T^n(\emptyset)$, which is the last set in the fixed point.. if this has an infinite tree, then all the above also have that infinite tree.. so the set cannot be finite.. so it cannot be a least fixed point..

The greatest fixed point is very different from the least fixed point.. makes point that Prolog looks for the greatest fixed point..

Q: is do I want to start at null and say “yes“, or start at all and say “no“

$F: P(\text{type} \times \text{type}) \rightarrow P(\text{type} \times \text{type})$

$F(S) = \{(U, \text{TOP}), (\text{Nat}, \text{Nat})\} U \neq x$

$U\{(U_1 \times U_2, T_1 \times T_2) \mid (U_1, T_1) \in S, (U_2, T_2) \in S\}$,

$U\{(U_1 \rightarrow U_2, T_1 \rightarrow T_2) \mid (U_1, T_1) \in S, (U_2, T_2) \in S\}$

$U\{(U, \mu x.T) \mid (U, [x \rightarrow \mu x.T]) \in S\}$

$U\{(\mu x.U, T) \mid ([x \rightarrow \mu x.U]U, T) \in S\}$

U

Conjecture from the board.. $\mu X.X, \mu X.X \in \text{LFP}(F)\Gamma$

follows our algorithm to conclude that even this simple type will not check with least fixed point

$(\mu X.\text{Nat} \times X, \text{Nat} \times \mu X.\text{Nat} \times X) \in F^{n-1}(\emptyset)\Gamma$

$(\text{Nat} \times \mu X.\text{Nat} \times X, \dots) \in F^{n-2}(\emptyset)\Gamma$

$(\mu X.\text{Nat} \times X, \dots) \in F^{n-3}(\emptyset)\Gamma$

but this has the same form as the original, which means no fixed point is ever reached, so this type cannot be in the LFP..

We have F from above.. saying it's a function from sets of pairs of finite types and its greatest fixed point is our subtyping algorithm.. why all this complexity..

“because have to have a way of showing something is in the fixed point without actually computing the greatest fixed point..”

How do we compute the greatest fixed point of FT

.. have to do it lazily, at the very least.. There's this thing where if asked a question been asked before, answer is “yes”.. where is that on the board..

-- One thing is that F must be syntax directed.. if it is, then get exactly one proof.. can assume a counter example, then if that path diverges, will be certain there is no other path so can say “yes” (or something like that..)