Type Systems for Programming Languages (15-814)
Lecture Notes, Fall 2006, Week One

Kyung-Ah Sohn

September 12 and 14, 2006

1 Inductive Definition

Judgement

A judgment (assertion, predicate, relation) is a statement that one or more objects have a property or stand in some relation to one another.

Example 1.1. x `nat` is a judgment.

Inference rules

The following two rules constitute an inductive definition of x `nat`.

$$\frac{}{\text{zero } \mathtt{nat}} \qquad\qquad \frac{x\ \mathtt{nat}}{\text{succ}(x)\ \mathtt{nat}}$$

x `nat` is the *strongest* judgement that is *closed under* the given rules.

- `if` $x\ J$ is closed under the rules, `then` x `nat` implies $x\ J$.
- *i.e.*, `if zero` J and $x\ J$ implies $\text{succ}(x)\ J$, `then` x `nat` implies $x\ J$

  → Principle of Mathematical Induction.

Example 1.2. Inductive definition of $s\ \text{string}_{\sum}$ ($\sum$: alphabet)

$$\frac{}{\varepsilon\ \text{string}_{\sum}} \qquad\qquad \frac{a \in \sum \qquad s\ \text{string}_{\sum}}{a.s\ \text{string}_{\sum}}$$

Rule induction

1. $\varepsilon\,\mathrm{P}$
2. $(a.y)\ P$ if $y\ P$ and $a \in \sum$

   $\Rightarrow \quad x$ `string` implies $x\ P$.

Defining Functions by rules

Example 1.3 (Addition of natural numbers)**.** $A(m,n,p)$ meaning "P is the sum of m and n" or $m+n=p$:

$$\frac{m\,\texttt{nat}}{A(m,0,m)} \qquad\qquad \frac{A(m,n,p)}{A(m,\texttt{succ}(n),\texttt{succ}(p))}$$

Claim A has mode $(\forall,\forall,\exists!)$

i.e., $\forall m\ \texttt{nat}.\ \forall n\ \texttt{nat}.\ \exists! p\,\texttt{nat}.\ A(m,n,p)$

i.e., A defines a function of m and n

Proof) $P(n) := \forall m\ \texttt{nat}.\ \exists! p\ \texttt{nat}.\ A(m,n,p)$

To Show : $n\,\texttt{nat}$ implies $P(n)$

Sufficient To Show :

(1) $P(\texttt{zero})$
take $p=m$, $A(m,\texttt{zero},m)$ by the first rule for A

(2) if $P(n)$, then $P(\texttt{succ}(n))$
Suppose $P(n)$, i.e., $\forall m\,\texttt{nat}.\ \exists! p\,\texttt{nat}.\ A(m,n,p)$
TS: $P(\texttt{succ}(n))$, i.e., $\forall m'\ \texttt{nat}.\ \exists! p'\ \texttt{nat}.\ A(m',\texttt{succ}(n),p')$

Suppose m' $\texttt{nat}$.
$A(m',\texttt{succ}(n),p')$
take $m=m'$. Choose p s.t. $A(m',n,p)$ by IH.
take $p'=\texttt{succ}(p)$ (uniquely determined).

Mode

- total function $(\forall,\exists!)$
- partial function $(\forall,\exists^{\leq 1})$
- total relation $(\forall,\exists)$

[partial function $\wedge$ total relation] $\equiv$ total function

Example 1.4. String concatenation $s\hat{\ }t=u$

$$\frac{t\,\texttt{string}}{\varepsilon\hat{\ }t=t} \qquad\qquad \frac{s\hat{\ }t=u}{(a.s)\hat{\ }t=a.u}$$

Theorem 1.5. This has mode $(\forall,\forall,\exists!)$

Proof) $P(s) := \forall t\ \texttt{string}.\ \exists! u\ \texttt{string}.\ s\hat{\ }t=u$.

1. $P(\varepsilon)$ because $\forall t\ \texttt{string}.\ \varepsilon\hat{\ }t=t$ by the first rule for string concatenation. (Since the rule to define such $\varepsilon\hat{\ }t$ is unique, $u=t$ is uniquely determined)
2. Suppose $P(s)$. Then $\forall t\texttt{string}.\ \exists! u.\ s\hat{\ }t=u$. By the second rule of string concatenation, $(a.s)\hat{\ }t=a.u$ for $a\in\sum$. Since there is only one rule to define $(a.s)\hat{\ }t$ and such u is unique by the induction hypothesis, $u'=a.u$ $\texttt{string}$ is uniquely determined, which implies $P(a.s)$.

   $\Rightarrow s\,\texttt{string}$ implies $P(s)$.

Example 1.6. Length $|s| = n$, $(\forall, \exists!)$

$$\frac{}{|\varepsilon| = \texttt{zero}} \qquad \frac{|s| = n}{|a.s| = \texttt{succ}(n)}$$

Fact: if $|s| = n$, then s `string`, n `nat`

Hypothetical Judgements (Entailment)

Definition 1.7.

1. Derivability
   $J_1, \cdots, J_n \vdash J$
   J is derivable from $J_1, \cdots, J_n$ as fresh axioms
2. Admissibility
   $J_1, \cdots, J_n \vDash J$
   If $J_1, \cdots, J_n$ are derivable from given rules, then so is J

Example 1.8. $\#\ \texttt{nat} \vdash \texttt{succ(succ(}\#\texttt{))}\ \texttt{nat}$: uniform derivation of conclusion from hypothesis

Example 1.9. $\texttt{succ}(a)\ \texttt{nat} \vDash a\ \texttt{nat}$
($\texttt{succ}(a)\ \texttt{nat} \nvdash a\ \texttt{nat}$)

Every derivable consequence is also admissible, but not conversely.

Fact: Derivability is stable under extension of the rule set. But not for admissibility!

2 Concrete Syntax

String

Representation of a language

CFGs / BNF
$E ::= N \mid E_1 \times E_2 \mid E_1 + E_2$

Re-phrased as inductive definition

$x\ E \quad x$ is a string of syntactic category E

$$\frac{n\ \texttt{nat}}{n\ E} \qquad \frac{s_1\ E \qquad s_2\ E}{s_1{}^\wedge(\times)^\wedge s_2\ E} \qquad \frac{s_1\ E \qquad s_2\ E}{s_1{}^\wedge(+)^\wedge s_2\ E}$$

Ambiguity

Example 2.1. $s = 1 + 2 \times 3$

a) $1 + 2^\wedge \times^\wedge 3$

b) $1^\wedge +^\wedge 2 \times 3$

Definition 2.2. $s\ E \Downarrow n\ \texttt{nat}$ *expression s has value n*

$$\frac{n\ \texttt{nat}}{n\ E \Downarrow n\ \texttt{nat}} \qquad \frac{s_1\ E \Downarrow n_1\ \texttt{nat} \qquad s_2\ E \Downarrow n_2\ \texttt{nat} \qquad n_1 \times n_2 = n\ \texttt{nat} \qquad s_1 \times s_2 = s\ \texttt{string}}{s\ E \Downarrow n\ \texttt{nat}}$$

Conjecture: this has mode $(\forall, \exists^{\leq 1})$

$\Rightarrow$ if $s\ E$, then $\exists^{\leq 1} n\ \texttt{nat}.\ s\ E \Downarrow n\ \texttt{nat}$

But there exist two different ways to evaluate the given expression as in Example 2.1. $\Rightarrow$ This conjecture is false!

Abstract syntax

A language is a set of trees (abstract syntax trees (`ast`)).

Example 2.3. x `ast` x is an abstract syntax tree

$$\frac{n\ \texttt{nat}}{\texttt{num}[n]\ \ \texttt{ast}} \qquad \frac{a_1\ \texttt{ast} \qquad a_2\ \texttt{ast}}{\texttt{plus}[a_1,\ a_2]\ \ \texttt{ast}} \qquad \frac{a_1\ \texttt{ast} \qquad a_2\ \texttt{ast}}{\texttt{times}[a_1,\ a_2]\ \ \texttt{ast}}$$

In ML, this becomes:

```
datatype ast = Num of int
 | Plus of ast * ast
 | Times of ast * ast;
```

Example 2.4. $a\ \texttt{ast}\ \Downarrow\ n\ \texttt{nat}$

$$\frac{n\ \texttt{nat}}{\texttt{num}[n]\texttt{ast}\ \Downarrow\ n\ \texttt{nat}} \qquad \frac{a_1\ \texttt{ast}\ \Downarrow\ n_1\ \texttt{nat} \qquad a_2\ \texttt{ast}\ \Downarrow\ n_2\ \texttt{nat} \qquad a = \texttt{plus}(a_1,\ a_2) \qquad n = n_1 + n_2\ \texttt{nat}}{a\ \texttt{ast}\ \Downarrow\ n\ \texttt{nat}}$$

$$\frac{a_1\ \texttt{ast}\ \Downarrow\ n_1\ \texttt{nat} \qquad a_2\ \texttt{ast}\ \Downarrow\ n_2\ \texttt{nat} \qquad a = \texttt{times}(a_1,\ a_2) \qquad n = n_1 \times n_2\ \texttt{nat}}{a\ \texttt{ast}\ \Downarrow\ n\ \texttt{nat}}$$

Claim: a `ast` implies $\exists! n.\ \texttt{nat}\quad a\ \texttt{ast}\ \Downarrow\ n\ \texttt{nat}$

Proof) By *structural induction* on a.

1. If $n\ \texttt{nat}$, then $\texttt{num}[n]\ \texttt{ast} \Downarrow n\ \texttt{nat}$ and this value is uniquely determined.
2. If $\exists! n_1\ \texttt{nat}.\ a_1\ \texttt{ast}\ \Downarrow\ n_1\ \texttt{nat}$ and $\exists! n_2\ \texttt{nat}.\ a_2\ \texttt{ast}\ \Downarrow\ n_2\ \texttt{nat}$, then $\texttt{plus}(a_1,\ a_2)\,\texttt{ast}\ \Downarrow\ n = n_1 + n_2\ \texttt{nat}$ where n is uniquely determined by the second induction rule
3. If $\exists! n_1\ \texttt{nat}.\ a_1\ \texttt{ast}\ \Downarrow\ n_1\ \texttt{nat}$ and $\exists! n_2\ \texttt{nat}.\ a_2\ \texttt{ast}\ \Downarrow\ n_2\ \texttt{nat}$, then $\texttt{times}(a_1,\ a_2)\,\texttt{ast}\ \Downarrow\ n = n_1 \times n_2\ \texttt{nat}$ where n is uniquely determined by the third induction rule

Structural Induction

To Show: a `ast` implies $a\ P$.
Sufficient To Show:

1. $n\ \texttt{nat}\ \ \supset \texttt{num}[n]\ P$
2. $a_1\ P$ and $a_2\ P\ \ \supset\ \ \texttt{plus}(a_1,\ a_2)\ P$
3. $a_1\ P$ and $a_2\ P\ \ \supset\ \ \texttt{times}(a_1,\ a_2)\ P$

Ambiguity

a) $1 + 2^\wedge \times^\wedge 3$
b) $1^\wedge +^\wedge 2 \times 3$

Fix:

1. Introduce precedences
2. Introduce parentheses to undo precedences

$s\ F$	factors
$s\ T$	terms
$s\ E$	expressions

$$\frac{n\ \texttt{nat}}{n\ F} \qquad \frac{s_1\ E \qquad s = ({}^\wedge s_1{}^\wedge)}{s\ F} \qquad \frac{s\ F}{s\ T} \qquad \frac{s_1\ F \qquad s_2\ T \qquad s = s_1{}^\wedge \times^\wedge s_2}{s\ T} \qquad \frac{s\ T}{s\ E}$$

$$\frac{s_1\ T \qquad s_2\ E \qquad s = s_1{}^\wedge +^\wedge s_2}{s\ E}$$

Parsing/Formatting

Simultaneous inductive definitions

$$s\ E \ \leftrightarrow\ a\ \texttt{ast}$$

$$s\ T \ \leftrightarrow\ a\ \texttt{ast}$$

$$s\ F \ \leftrightarrow\ a\ \texttt{ast}$$

$$\frac{n\ \texttt{nat}}{n\ F \ \leftrightarrow\ \texttt{num}[n]\ \texttt{ast}} \qquad \frac{s = ({}^\wedge s_1{}^\wedge) \qquad s_1\ E \ \leftrightarrow\ a_1\ \texttt{ast}}{s\ F \ \leftrightarrow\ a_1\ \texttt{ast}} \qquad \frac{s\ F \ \leftrightarrow\ a\ \texttt{ast}}{s\ T \ \leftrightarrow\ a\ \texttt{ast}}$$

$$\frac{s = s_1{}^\wedge \times^\wedge s_2 \qquad s_1\ F \ \leftrightarrow\ a_1\ \texttt{ast} \qquad s_2\ T \ \leftrightarrow\ a_2\ \texttt{ast}}{s\ T \ \leftrightarrow\ \texttt{times}(a_1,\ a_2)\ \texttt{ast}} \qquad \text{etc.}$$

Abstract Binding Trees (ABTs)

How do we account for variables?

1. binding + scope: what points where
2. substitution: variables stand for all possible instances

Definition 2.5. AST's arity
Θ operates with arity $k \geq 0$

$$\frac{a_1\ \texttt{ast}, \cdots, a_k\ \texttt{ast}}{\Theta(a_1, \cdots, a_k)\ \texttt{ast}} \qquad \text{(arity of } \Theta = k\text{)}$$

`num`$[n]$	0
`plus`	2
`times`	2

Definition 2.6. $[x \leftrightarrow y]z = z'\ \texttt{name}$.

$$\frac{}{[x \leftrightarrow y]x = y\ \texttt{name}} \qquad \frac{}{[x \leftrightarrow y]y = x\ \texttt{name}} \qquad \frac{x\#y\ \texttt{name} \qquad y\#z\ \texttt{name}}{[x \leftrightarrow y]z = z\ \texttt{name}}$$

Definition 2.7. ABT's arity: finite sequence of natural numbers $(n_1, \ldots, n_k)$
valence: natural number. # of variables bound on that P

Definition 2.8. $a\ \texttt{abt}^{\texttt{n}}$.

$$\frac{x\ \texttt{name}}{x\ \texttt{abt}^0} \qquad \frac{x\ \texttt{name} \qquad a\ \texttt{abt}^{\texttt{n}}}{x.a\ \texttt{abt}^{\texttt{n}+1}} \qquad \frac{a_1\ \texttt{abt}^{\texttt{n}_1}\ \ldots\ a_k\ \texttt{abt}^{\texttt{n}_k}}{\Theta(a_1, \ldots, a_k)\ \texttt{abt}^0}$$

Arity of $\Theta = (n_1, \ldots, n_k),\ k \geq 0$ # arguments

Example 2.9.

$\texttt{num}[n]$	()
`plus`	(0, 0)
`trees`	(0, 0)
`let`	(0, 1)

Example 2.10. $\texttt{let}\ (\ \texttt{plus}(1{,}2),\ x.\texttt{times}(x, x))$
$\sim\ \texttt{let}\ x\ \texttt{be}\ 1{+}2\ \text{in}\ x \times x$

Four judgements

$x\ \#\ a$	x lies apart from a a does not involve x
$[x \leftrightarrow y]a = b\ \texttt{abt}^{\texttt{n}}$	swap x for y and vice versa in a to obtain b
$a =_\alpha b\ \texttt{abt}^{\texttt{n}}$	α-equivalence
$[a/x]b = c\ \texttt{abt}^{\texttt{n}}$	the result of substituting a for x in b is c

Example 2.11. $[0/y]\ (\texttt{let}\ x\ \texttt{be}\ 2{+}3\ \texttt{in}\ x + y)$
$= \texttt{let}\ x\ \texttt{be}\ 2{+}3\ \texttt{in}\ x + 0$
$[x + x/y]\ (\texttt{let}\ x\ \texttt{be}\ 2{+}3\ \texttt{in}\ x + y\ )$
$\neq\ \texttt{let}\ x\ \texttt{be}\ 2 + 3\ \texttt{in}\ x + (x + x)$ (X)

Apartness

$$\frac{x\#y\ \texttt{name}}{x\#y\ \texttt{abt}^0} \qquad \frac{x\#a_1\ \texttt{abt}^{\texttt{n}_1} \qquad \ldots \qquad x\#a_k\ \texttt{abt}^{\texttt{n}_k}}{x\#\Theta(a_1, \ldots, a_k)\ \texttt{abt}^0} \qquad \frac{}{x\#x.a\ \texttt{abt}^{\texttt{n}+1}} \qquad \frac{x\#y\ \texttt{name} \qquad x\#a\ \texttt{abt}^{\texttt{n}}}{x\#y.a\ \texttt{abt}^{\texttt{n}+1}}$$

Swapping

$$\frac{[x \leftrightarrow y]z = z'\ \texttt{name}}{[x \leftrightarrow y]z = z'\ \texttt{abt}^0} \qquad \frac{[x \leftrightarrow y]a_1 = a_1'\ \texttt{abt}^{\texttt{n}_1} \qquad \ldots \qquad [x \leftrightarrow y]a_k = a_k'\ \texttt{abt}^{\texttt{n}_k}}{[x \leftrightarrow y]\Theta(a_1, \ldots, a_k) = \Theta(a_1', \ldots, a_k')\ \texttt{abt}^0}$$

$$\frac{[x \leftrightarrow y]z = z'\ \texttt{name} \qquad [x \leftrightarrow y]a = a'\ \texttt{abt}^{\texttt{n}}}{[x \leftrightarrow y]z.a = z'.a'\ \texttt{abt}^{\texttt{n}+1}}$$

α-equivalence

$a =_\alpha b\ \mathtt{abt}^n$

$$\frac{}{x =_\alpha x\ \mathtt{abt}^{\mathtt{0}}} \qquad \frac{a_1 =_\alpha b_1\ \mathtt{abt}^{\mathtt{n}_1} \qquad \ldots \qquad a_k =_\alpha b_k\ \mathtt{abt}^{\mathtt{n}_k}}{\Theta(a_1, \ldots, a_k) =_\alpha O(b_1, \ldots, b_k)\ \mathtt{abt}^{\mathtt{0}}} \qquad \frac{a =_\alpha b\ \mathtt{abt}^{\mathtt{n}}}{x.a =_\alpha x.b\ \mathtt{abt}^{\mathtt{n+1}}}$$

$$\frac{y\#a \qquad x\#y \qquad [x \leftrightarrow y]a =_\alpha b\ \mathtt{abt}^{\mathtt{n}}}{x.a =_\alpha y.b\ \mathtt{abt}^{\mathtt{n+1}}}$$

Fact: $=_\alpha$ is an equivalence relation.

Derivable Rules

1) $$\frac{x\#y\ \mathtt{name} \qquad y\#a\ \mathtt{abt}^{\mathtt{n}}}{x.a =_\alpha y.[y \leftrightarrow x]a\ \mathtt{abt}^{\mathtt{n+1}}}$$ "α-conversion"

2) $$\frac{x\#y\ \mathtt{name} \qquad z\#a\ \mathtt{abt}^{\mathtt{n}} \qquad z\#b\ \mathtt{abt}^{\mathtt{n}} \qquad [x \leftrightarrow z]a\ \Leftrightarrow_\alpha\ [y \leftrightarrow z]b\ abt^n}{x.a\#y.b\ \mathtt{abt}^{\mathtt{n+1}}}$$