CIS 500
Software Foundations
Fall 2006

October 9

Review

Church encoding of lists

Church encoding of lists

... will not be on the exam. :-)

Briefly, though, here's the intuition:

c_4 = λs. λz. s (s (s (s z)))

Church encoding of lists

... will not be on the exam. :-)

Briefly, though, here's the intuition:

c_4 = λs. λz. s (s (s (s z)))

[v_1;v_2;v_3;v_4] = λs. λz. s v_1 (s v_2 (s v_3 (s v_4 z)))

Typing derivations

Exercise 9.2.2: Show (by drawing derivation trees) that the following terms have the indicated types:

1. f:Bool→Bool ⊢ f (if false then true else false) : Bool
2. f:Bool→Bool ⊢ λx:Bool. f (if x then false else x) : Bool→Bool

The two typing relations

Question: What is the relation between these two statements?

1. $\mathtt{t} : \mathtt{T}$
2. $\vdash \mathtt{t} : \mathtt{T}$

The two typing relations

Question: What is the relation between these two statements?

1. $\mathtt{t} : \mathtt{T}$
2. $\vdash \mathtt{t} : \mathtt{T}$

First answer: These two relations are completely different things.

- We are dealing with several different small programming languages, *each with its own typing relation* (between terms in that language and types in that language)
- For the simple language of numbers and booleans, typing is a *binary* relation between terms and types ($\mathtt{t} : \mathtt{T}$).
- For $\lambda_{\rightarrow}$, typing is a *ternary* relation between contexts, terms, and types ($\Gamma \vdash \mathtt{t} : \mathtt{T}$).

  (When the context is empty — because the term has no free variables — we often write $\vdash \mathtt{t} : \mathtt{T}$ to mean $\emptyset \vdash \mathtt{t} : \mathtt{T}$.)

Conservative extension

Second answer: The typing relation for $\lambda_{\rightarrow}$ *conservatively extends* the one for the simple language of numbers and booleans.

- Write "language 1" for the language of numbers and booleans and "language 2" for the simply typed lambda-calculus with base types $\mathtt{Nat}$ and $\mathtt{Bool}$.
- The terms of language 2 include all the terms of language 1; similarly typing rules.
- Write $\mathtt{t} :_1 \mathtt{T}$ for the typing relation of language 1.
- Write $\Gamma \vdash \mathtt{t} :_2 \mathtt{T}$ for the typing relation of language 2.
- *Theorem:* Language 2 conservatively extends language 1: If $\mathtt{t}$ is a term of language 1 (involving only booleans, conditions, numbers, and numeric operators) and $\mathtt{T}$ is a type of language 1 (either $\mathtt{Bool}$ or $\mathtt{Nat}$), then $\mathtt{t} :_1 \mathtt{T}$ iff $\emptyset \vdash \mathtt{t} :_2 \mathtt{T}$.

Preservation (and Weaking, Permutation, Substitution)

Review: Proving progress

Let's quickly review the steps in the proof of the progress theorem:

- inversion lemma for typing relation
- canonical forms lemma
- progress theorem

Inversion

Lemma:

1. If $\Gamma \vdash \mathtt{true} : \mathtt{R}$, then $\mathtt{R} = \mathtt{Bool}$.
2. If $\Gamma \vdash \mathtt{false} : \mathtt{R}$, then $\mathtt{R} = \mathtt{Bool}$.
3. If $\Gamma \vdash \mathtt{if}\ \mathtt{t}_1\ \mathtt{then}\ \mathtt{t}_2\ \mathtt{else}\ \mathtt{t}_3 : \mathtt{R}$, then $\Gamma \vdash \mathtt{t}_1 : \mathtt{Bool}$ and $\Gamma \vdash \mathtt{t}_2, \mathtt{t}_3 : \mathtt{R}$.
4. If $\Gamma \vdash \mathtt{x} : \mathtt{R}$, then

Inversion

Lemma:

1. If $\Gamma \vdash \texttt{true} : \texttt{R}$, then $\texttt{R} = \texttt{Bool}$.
2. If $\Gamma \vdash \texttt{false} : \texttt{R}$, then $\texttt{R} = \texttt{Bool}$.
3. If $\Gamma \vdash \texttt{if } \texttt{t}_1 \texttt{ then } \texttt{t}_2 \texttt{ else } \texttt{t}_3 : \texttt{R}$, then $\Gamma \vdash \texttt{t}_1 : \texttt{Bool}$ and $\Gamma \vdash \texttt{t}_2, \texttt{t}_3 : \texttt{R}$.
4. If $\Gamma \vdash \texttt{x} : \texttt{R}$, then $\texttt{x}{:}\texttt{R} \in \Gamma$.
5. If $\Gamma \vdash \lambda\texttt{x}{:}\texttt{T}_1.\texttt{t}_2 : \texttt{R}$, then

Inversion

Lemma:

1. If $\Gamma \vdash \texttt{true} : \texttt{R}$, then $\texttt{R} = \texttt{Bool}$.
2. If $\Gamma \vdash \texttt{false} : \texttt{R}$, then $\texttt{R} = \texttt{Bool}$.
3. If $\Gamma \vdash \texttt{if } \texttt{t}_1 \texttt{ then } \texttt{t}_2 \texttt{ else } \texttt{t}_3 : \texttt{R}$, then $\Gamma \vdash \texttt{t}_1 : \texttt{Bool}$ and $\Gamma \vdash \texttt{t}_2, \texttt{t}_3 : \texttt{R}$.
4. If $\Gamma \vdash \texttt{x} : \texttt{R}$, then $\texttt{x}{:}\texttt{R} \in \Gamma$.
5. If $\Gamma \vdash \lambda\texttt{x}{:}\texttt{T}_1.\texttt{t}_2 : \texttt{R}$, then $\texttt{R} = \texttt{T}_1 \to \texttt{R}_2$ for some $\texttt{R}_2$ with $\Gamma, \texttt{x}{:}\texttt{T}_1 \vdash \texttt{t}_2 : \texttt{R}_2$.
6. If $\Gamma \vdash \texttt{t}_1\ \texttt{t}_2 : \texttt{R}$, then

Inversion

Lemma:

1. If $\Gamma \vdash \texttt{true} : \texttt{R}$, then $\texttt{R} = \texttt{Bool}$.
2. If $\Gamma \vdash \texttt{false} : \texttt{R}$, then $\texttt{R} = \texttt{Bool}$.
3. If $\Gamma \vdash \texttt{if } \texttt{t}_1 \texttt{ then } \texttt{t}_2 \texttt{ else } \texttt{t}_3 : \texttt{R}$, then $\Gamma \vdash \texttt{t}_1 : \texttt{Bool}$ and $\Gamma \vdash \texttt{t}_2, \texttt{t}_3 : \texttt{R}$.
4. If $\Gamma \vdash \texttt{x} : \texttt{R}$, then $\texttt{x}{:}\texttt{R} \in \Gamma$.
5. If $\Gamma \vdash \lambda\texttt{x}{:}\texttt{T}_1.\texttt{t}_2 : \texttt{R}$, then $\texttt{R} = \texttt{T}_1 \to \texttt{R}_2$ for some $\texttt{R}_2$ with $\Gamma, \texttt{x}{:}\texttt{T}_1 \vdash \texttt{t}_2 : \texttt{R}_2$.
6. If $\Gamma \vdash \texttt{t}_1\ \texttt{t}_2 : \texttt{R}$, then there is some type $\texttt{T}_{11}$ such that $\Gamma \vdash \texttt{t}_1 : \texttt{T}_{11} \to \texttt{R}$ and $\Gamma \vdash \texttt{t}_2 : \texttt{T}_{11}$.

Canonical Forms

Lemma:

Canonical Forms

Lemma:

1. If $\texttt{v}$ is a value of type $\texttt{Bool}$, then $\texttt{v}$ is either $\texttt{true}$ or $\texttt{false}$.
2. If $\texttt{v}$ is a value of type $\texttt{T}_1 \to \texttt{T}_2$, then $\texttt{v}$ has the form $\lambda\texttt{x}{:}\texttt{T}_1.\texttt{t}_2$.

Progress

Theorem: Suppose $\texttt{t}$ is a closed, well-typed term (that is, $\vdash \texttt{t} : \texttt{T}$ for some $\texttt{T}$). Then either $\texttt{t}$ is a value or else there is some $\texttt{t}'$ with $\texttt{t} \longrightarrow \texttt{t}'$.

Preservation

Theorem: If $\Gamma \vdash \mathtt{t} : \mathtt{T}$ and $\mathtt{t} \longrightarrow \mathtt{t}'$, then $\Gamma \vdash \mathtt{t}' : \mathtt{T}$.

Steps of proof:

- Weakening
- Permutation
- Substitution preserves types
- Reduction preserves types (i.e., preservation)

Weakening and Permutation

Weakening tells us that we can *add assumptions* to the context without losing any true typing statements.

Lemma: If $\Gamma \vdash \mathtt{t} : \mathtt{T}$ and $\mathtt{x} \notin \mathit{dom}(\Gamma)$, then $\Gamma, \mathtt{x}{:}\mathtt{S} \vdash \mathtt{t} : \mathtt{T}$.

Weakening and Permutation

Weakening tells us that we can *add assumptions* to the context without losing any true typing statements.

Lemma: If $\Gamma \vdash \mathtt{t} : \mathtt{T}$ and $\mathtt{x} \notin \mathit{dom}(\Gamma)$, then $\Gamma, \mathtt{x}{:}\mathtt{S} \vdash \mathtt{t} : \mathtt{T}$.

Permutation tells us that the order of assumptions in (the list) Γ does not matter.

Lemma: If $\Gamma \vdash \mathtt{t} : \mathtt{T}$ and Δ is a permutation of Γ, then $\Delta \vdash \mathtt{t} : \mathtt{T}$.

Weakening and Permutation

Weakening tells us that we can *add assumptions* to the context without losing any true typing statements.

Lemma: If $\Gamma \vdash \mathtt{t} : \mathtt{T}$ and $\mathtt{x} \notin \mathit{dom}(\Gamma)$, then $\Gamma, \mathtt{x}{:}\mathtt{S} \vdash \mathtt{t} : \mathtt{T}$.

Moreover, the latter derivation has the same depth as the former.

Permutation tells us that the order of assumptions in (the list) Γ does not matter.

Lemma: If $\Gamma \vdash \mathtt{t} : \mathtt{T}$ and Δ is a permutation of Γ, then $\Delta \vdash \mathtt{t} : \mathtt{T}$.

Moreover, the latter derivation has the same depth as the former.

Preservation

Theorem: If $\Gamma \vdash \mathtt{t} : \mathtt{T}$ and $\mathtt{t} \longrightarrow \mathtt{t}'$, then $\Gamma \vdash \mathtt{t}' : \mathtt{T}$.

Proof: By induction

Preservation

Theorem: If $\Gamma \vdash \mathtt{t} : \mathtt{T}$ and $\mathtt{t} \longrightarrow \mathtt{t}'$, then $\Gamma \vdash \mathtt{t}' : \mathtt{T}$.

Proof: By induction on typing derivations.

Which case is the hard one??

Preservation

Theorem: If $\Gamma \vdash t : T$ and $t \longrightarrow t'$, then $\Gamma \vdash t' : T$.

Proof: By induction on typing derivations.
Case T-App: Given $t = t_1\ t_2$
$\Gamma \vdash t_1 : T_{11} \to T_{12}$
$\Gamma \vdash t_2 : T_{11}$
$T = T_{12}$
Show $\Gamma \vdash t' : T_{12}$

Preservation

Theorem: If $\Gamma \vdash t : T$ and $t \longrightarrow t'$, then $\Gamma \vdash t' : T$.

Proof: By induction on typing derivations.
Case T-App: Given $t = t_1\ t_2$
$\Gamma \vdash t_1 : T_{11} \to T_{12}$
$\Gamma \vdash t_2 : T_{11}$
$T = T_{12}$
Show $\Gamma \vdash t' : T_{12}$
By the inversion lemma for evaluation, there are three subcases...

Preservation

Theorem: If $\Gamma \vdash t : T$ and $t \longrightarrow t'$, then $\Gamma \vdash t' : T$.

Proof: By induction on typing derivations.
Case T-App: Given $t = t_1\ t_2$
$\Gamma \vdash t_1 : T_{11} \to T_{12}$
$\Gamma \vdash t_2 : T_{11}$
$T = T_{12}$
Show $\Gamma \vdash t' : T_{12}$
By the inversion lemma for evaluation, there are three subcases...
Subcase: $t_1 = \lambda x{:}T_{11}.\ t_{12}$
t_2 a value v_2
$t' = [x \mapsto v_2]t_{12}$

Preservation

Theorem: If $\Gamma \vdash t : T$ and $t \longrightarrow t'$, then $\Gamma \vdash t' : T$.

Proof: By induction on typing derivations.
Case T-App: Given $t = t_1\ t_2$
$\Gamma \vdash t_1 : T_{11} \to T_{12}$
$\Gamma \vdash t_2 : T_{11}$
$T = T_{12}$
Show $\Gamma \vdash t' : T_{12}$
By the inversion lemma for evaluation, there are three subcases...
Subcase: $t_1 = \lambda x{:}T_{11}.\ t_{12}$
t_2 a value v_2
$t' = [x \mapsto v_2]t_{12}$
Uh oh.

Preservation

Theorem: If $\Gamma \vdash t : T$ and $t \longrightarrow t'$, then $\Gamma \vdash t' : T$.

Proof: By induction on typing derivations.
Case T-App: Given $t = t_1\ t_2$
$\Gamma \vdash t_1 : T_{11} \to T_{12}$
$\Gamma \vdash t_2 : T_{11}$
$T = T_{12}$
Show $\Gamma \vdash t' : T_{12}$
By the inversion lemma for evaluation, there are three subcases...
Subcase: $t_1 = \lambda x{:}T_{11}.\ t_{12}$
t_2 a value v_2
$t' = [x \mapsto v_2]t_{12}$
Uh oh. What do we need to know to make this case go through??

The "Substitution Lemma"

Lemma: If $\Gamma, x{:}S \vdash t : T$ and $\Gamma \vdash s : S$, then $\Gamma \vdash [x \mapsto s]t : T$.

I.e., "Types are preserved under substitition."

The "Substitution Lemma"

Lemma: If $\Gamma, x{:}S \vdash t : T$ and $\Gamma \vdash s : S$, then $\Gamma \vdash [x \mapsto s]t : T$.

Proof: By induction on the *depth* of a derivation of $\Gamma, x{:}S \vdash t : T$. Proceed by cases on the final typing rule used in the derivation.

The "Substitution Lemma"

Lemma: If $\Gamma, x{:}S \vdash t : T$ and $\Gamma \vdash s : S$, then $\Gamma \vdash [x \mapsto s]t : T$.

Proof: By induction on the *depth* of a derivation of $\Gamma, x{:}S \vdash t : T$. Proceed by cases on the final typing rule used in the derivation.

The "Substitution Lemma"

Lemma: If $\Gamma, x{:}S \vdash t : T$ and $\Gamma \vdash s : S$, then $\Gamma \vdash [x \mapsto s]t : T$.

Proof: By induction on the *depth* of a derivation of $\Gamma, x{:}S \vdash t : T$. Proceed by cases on the final typing rule used in the derivation.

Case T-APP:
$$t = t_1\ t_2$$
$$\Gamma, x{:}S \vdash t_1 : T_2 \to T_1$$
$$\Gamma, x{:}S \vdash t_2 : T_2$$
$$T = T_1$$

By the induction hypothesis, $\Gamma \vdash [x \mapsto s]t_1 : T_2 \to T_1$ and $\Gamma \vdash [x \mapsto s]t_2 : T_2$. By T-APP, $\Gamma \vdash [x \mapsto s]t_1\ [x \mapsto s]t_2 : T$, i.e., $\Gamma \vdash [x \mapsto s](t_1\ t_2) : T$.

The "Substitution Lemma"

Lemma: If $\Gamma, x{:}S \vdash t : T$ and $\Gamma \vdash s : S$, then $\Gamma \vdash [x \mapsto s]t : T$.

Proof: By induction on the *depth* of a derivation of $\Gamma, x{:}S \vdash t : T$. Proceed by cases on the final typing rule used in the derivation.

Case T-VAR:
$$t = z$$
with $z{:}T \in (\Gamma, x{:}S)$

There are two sub-cases to consider, depending on whether z is x or another variable. If $z = x$, then $[x \mapsto s]z = s$. The required result is then $\Gamma \vdash s : S$, which is among the assumptions of the lemma. Otherwise, $[x \mapsto s]z = z$, and the desired result is immediate.

The "Substitution Lemma"

Lemma: If $\Gamma, x{:}S \vdash t : T$ and $\Gamma \vdash s : S$, then $\Gamma \vdash [x \mapsto s]t : T$.

Proof: By induction on the *depth* of a derivation of $\Gamma, x{:}S \vdash t : T$. Proceed by cases on the final typing rule used in the derivation.

Case T-ABS:
$$t = \lambda y{:}T_2.t_1 \qquad T = T_2 \to T_1$$
$$\Gamma, x{:}S, y{:}T_2 \vdash t_1 : T_1$$

By our conventions on choice of bound variable names, we may assume $x \neq y$ and $y \notin FV(s)$. Using *permutation* on the given subderivation, we obtain $\Gamma, y{:}T_2, x{:}S \vdash t_1 : T_1$. Using *weakening* on the other given derivation ($\Gamma \vdash s : S$), we obtain $\Gamma, y{:}T_2 \vdash s : S$. Now, by the induction hypothesis, $\Gamma, y{:}T_2 \vdash [x \mapsto s]t_1 : T_1$. By T-ABS, $\Gamma \vdash \lambda y{:}T_2.\ [x \mapsto s]t_1 : T_2 \to T_1$, i.e. (by the definition of substitution), $\Gamma \vdash [x \mapsto s]\lambda y{:}T_2.\ t_1 : T_2 \to T_1$.