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:

$$c_4 = \lambda s.\ \lambda z.\ s\ (s\ (s\ (s\ z)))$$

Church encoding of lists

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

Briefly, though, here's the intuition:

$$\texttt{c}_4 \texttt{ = } \lambda\texttt{s. } \lambda\texttt{z. s (s (s (s z)))}$$

$$\texttt{[v}_1\texttt{;v}_2\texttt{;v}_3\texttt{;v}_4\texttt{] = } \lambda\texttt{s. } \lambda\texttt{z. s v}_1 \texttt{ (s v}_2 \texttt{ (s v}_3 \texttt{ (s v}_4 \texttt{ z)))}$$

Typing derivations

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

1. $\texttt{f:Bool}\to\texttt{Bool} \vdash \texttt{f (if false then true else false)} : \texttt{Bool}$
2. $\texttt{f:Bool}\to\texttt{Bool} \vdash \lambda\texttt{x:Bool. f (if x then false else x)} : \texttt{Bool}\to\texttt{Bool}$

The two typing relations

Question: What is the relation between these two statements?

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

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 `Nat` and `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 `t` is a term of language 1 (involving only booleans, conditions, numbers, and numeric operators) and `T` is a type of language 1 (either `Bool` or `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 $\texttt{x:R} \in \Gamma$.
5. If $\Gamma \vdash \lambda\texttt{x:T}_1\texttt{.t}_2 : \texttt{R}$, 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 $\mathtt{v}$ is a value of type $\mathtt{Bool}$, then $\mathtt{v}$ is either $\mathtt{true}$ or $\mathtt{false}$.
2. If $\mathtt{v}$ is a value of type $\mathtt{T_1}{\rightarrow}\mathtt{T_2}$, then $\mathtt{v}$ has the form $\lambda\mathtt{x}{:}\mathtt{T_1}.\mathtt{t_2}$.

Progress

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

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 $\mathtt{x} \notin dom(\Gamma)$, 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 $\Gamma \vdash \mathtt{t} : \mathtt{T}$ and $\mathtt{t} \longrightarrow \mathtt{t}'$, then $\Gamma \vdash \mathtt{t}' : \mathtt{T}$.

Proof: By induction on typing derivations.

Case T-APP: Given $\mathtt{t} = \mathtt{t}_1\ \mathtt{t}_2$

$\Gamma \vdash \mathtt{t}_1 : \mathtt{T}_{11} \rightarrow \mathtt{T}_{12}$

$\Gamma \vdash \mathtt{t}_2 : \mathtt{T}_{11}$

$\mathtt{T} = \mathtt{T}_{12}$

Show $\Gamma \vdash \mathtt{t}' : \mathtt{T}_{12}$

Preservation

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

Proof: By induction on typing derivations.

Case T-App: Given $\texttt{t} = \texttt{t}_1 \ \texttt{t}_2$

$\Gamma \vdash \texttt{t}_1 : \texttt{T}_{11} \rightarrow \texttt{T}_{12}$

$\Gamma \vdash \texttt{t}_2 : \texttt{T}_{11}$

$\texttt{T} = \texttt{T}_{12}$

Show $\Gamma \vdash \texttt{t}' : \texttt{T}_{12}$

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 $t = t_1\ t_2$

$\Gamma \vdash t_1 : T_{11} \rightarrow T_{12}$

$\Gamma \vdash t_2 : T_{11}$

$T = T_{12}$

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

Proof: By induction on typing derivations.

Case T-App: Given

$$\mathtt{t} = \mathtt{t}_1\ \mathtt{t}_2$$
$$\Gamma \vdash \mathtt{t}_1 : \mathtt{T}_{11} \to \mathtt{T}_{12}$$
$$\Gamma \vdash \mathtt{t}_2 : \mathtt{T}_{11}$$
$$\mathtt{T} = \mathtt{T}_{12}$$

Show $\Gamma \vdash \mathtt{t}' : \mathtt{T}_{12}$

By the inversion lemma for evaluation, there are three subcases...

Subcase: $\mathtt{t}_1 = \lambda \mathtt{x}{:}\mathtt{T}_{11}.\ \mathtt{t}_{12}$

$\mathtt{t}_2$ a value $\mathtt{v}_2$

$\mathtt{t}' = [\mathtt{x} \mapsto \mathtt{v}_2]\mathtt{t}_{12}$

Uh oh.

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.

Case T-App: Given $\mathtt{t} = \mathtt{t}_1\ \mathtt{t}_2$

$\Gamma \vdash \mathtt{t}_1 : \mathtt{T}_{11} \to \mathtt{T}_{12}$

$\Gamma \vdash \mathtt{t}_2 : \mathtt{T}_{11}$

$\mathtt{T} = \mathtt{T}_{12}$

Show $\Gamma \vdash \mathtt{t}' : \mathtt{T}_{12}$

By the inversion lemma for evaluation, there are three subcases...

Subcase: $\mathtt{t}_1 = \lambda \mathtt{x}{:}\mathtt{T}_{11}.\ \mathtt{t}_{12}$

$\mathtt{t}_2$ a value $\mathtt{v}_2$

$\mathtt{t}' = [\mathtt{x} \mapsto \mathtt{v}_2]\mathtt{t}_{12}$

Uh oh. What do we need to know to make this case go through??

The "Substitution Lemma"

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

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

The "Substitution Lemma"

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

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

The "Substitution Lemma"

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

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

The “Substitution Lemma”

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

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

Case T-APP:

$$\begin{aligned} &\mathtt{t} = \mathtt{t_1}\ \mathtt{t_2} \\ &\Gamma, \mathtt{x{:}S} \vdash \mathtt{t_1} : \mathtt{T_2 \to T_1} \\ &\Gamma, \mathtt{x{:}S} \vdash \mathtt{t_2} : \mathtt{T_2} \\ &\mathtt{T} = \mathtt{T_1} \end{aligned}$$

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

The "Substitution Lemma"

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

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

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

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

The "Substitution Lemma"

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

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

Case T-Abs:

$$\mathtt{t} = \lambda\mathtt{y}{:}\mathtt{T}_2.\mathtt{t}_1 \qquad \mathtt{T} = \mathtt{T}_2 \rightarrow \mathtt{T}_1$$
$$\Gamma, \mathtt{x}{:}\mathtt{S}, \mathtt{y}{:}\mathtt{T}_2 \vdash \mathtt{t}_1 : \mathtt{T}_1$$

By our conventions on choice of bound variable names, we may assume $\mathtt{x} \neq \mathtt{y}$ and $\mathtt{y} \notin FV(\mathtt{s})$. Using *permutation* on the given subderivation, we obtain $\Gamma, \mathtt{y}{:}\mathtt{T}_2, \mathtt{x}{:}\mathtt{S} \vdash \mathtt{t}_1 : \mathtt{T}_1$. Using *weakening* on the other given derivation ($\Gamma \vdash \mathtt{s} : \mathtt{S}$), we obtain $\Gamma, \mathtt{y}{:}\mathtt{T}_2 \vdash \mathtt{s} : \mathtt{S}$. Now, by the induction hypothesis, $\Gamma, \mathtt{y}{:}\mathtt{T}_2 \vdash [\mathtt{x} \mapsto \mathtt{s}]\mathtt{t}_1 : \mathtt{T}_1$. By T-Abs, $\Gamma \vdash \lambda\mathtt{y}{:}\mathtt{T}_2.\ [\mathtt{x} \mapsto \mathtt{s}]\mathtt{t}_1 : \mathtt{T}_2 \rightarrow \mathtt{T}_1$, i.e. (by the definition of substitution), $\Gamma \vdash [\mathtt{x} \mapsto \mathtt{s}]\lambda\mathtt{y}{:}\mathtt{T}_2.\ \mathtt{t}_1 : \mathtt{T}_2 \rightarrow \mathtt{T}_1$.