CS152: Programming Languages

Lecture 10 — Type-Safety Proof

Dan Grossman
Spring 2011

Outline

- Type-safety proof
  - Also posted in non-slide form
- Discuss the proof
  - Consider lemma dependencies and what they represent
  - Consider elegance of inverting static and dynamic derivations
- Next lecture: Add more constructs to our typed language
  - Pairs, records, sums, recursion, ...
  - For each, sketch proof additions (follow a general approach)
- Further ahead: More flexible typing via polymorphism

Review: Lambda-Calculus with Constants

$$e ::= \lambda x.\ e \mid x \mid e\ e \mid c \qquad v ::= \lambda x.\ e \mid c$$

$$\frac{}{(\lambda x.\ e)\ v \to e[v/x]} \qquad \frac{e_1 \to e_1'}{e_1\ e_2 \to e_1'\ e_2} \qquad \frac{e_2 \to e_2'}{v\ e_2 \to v\ e_2'}$$

$$\frac{}{x[e/x] = e} \qquad \frac{y \neq x}{y[e/x] = y} \qquad \frac{}{c[e/x] = c}$$

$$\frac{e_1[e/x] = e_1' \quad y \neq x \quad y \notin FV(e)}{(\lambda y.\ e_1)[e/x] = \lambda y.\ e_1'} \qquad \frac{e_1[e/x] = e_1' \qquad e_2[e/x] = e_2'}{(e_1\ e_2)[e/x] = e_1'\ e_2'}$$

Stuck states: not values and no step applies...

Avoid stuck states to:

- Catch bugs (why would you want to get to such a state?)
- Ease implementation (no need to check for being stuck)

Review: Typing Judgment

Defined a type system to classify λ-terms

Some terms have types; some don't

$$\tau ::= \textbf{int} \mid \tau \to \tau \qquad \Gamma ::= \cdot \mid \Gamma, x : \tau$$

$$\frac{}{\Gamma \vdash c : \textbf{int}} \qquad \frac{}{\Gamma \vdash x : \Gamma(x)} \qquad \frac{\Gamma, x : \tau_1 \vdash e : \tau_2}{\Gamma \vdash \lambda x.\ e : \tau_1 \to \tau_2}$$

$$\frac{\Gamma \vdash e_1 : \tau_2 \to \tau_1 \qquad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash e_1\ e_2 : \tau_1}$$

Theorem: A program that typechecks under $\cdot$ won't get stuck, i.e., If $\cdot \vdash e : \tau$ then e diverges or $\exists v, n$ such that $e \to^n v$

Preservation and Progress

Theorem (slightly restated): If $\cdot \vdash e : \tau$ and $e \to^n e'$, then e' is a value or there exists an e'' such that $e' \to e''$

Follows from two key lemmas:

- Lemma (Preservation): If $\cdot \vdash e : \tau$ and $e \to e'$, then $\cdot \vdash e' : \tau$.
- Lemma (Progress): If $\cdot \vdash e : \tau$, then e is a value or there exists an e' such that $e \to e'$.

Proof of theorem given lemmas:

- "Preservation*": If $\cdot \vdash e : \tau$ and $e \to^n e'$, then $\cdot \vdash e' : \tau$
  - Trivial induction on n given Preservation
- So Progress ensures e' is not stuck

Progress

Lemma: If $\cdot \vdash e : \tau$, then e is a value or there exists an e' such that $e \to e'$

Proof: We first prove this lemma:

Lemma (Canonical Forms): If $\cdot \vdash v : \tau$, then:

- if τ is **int**, then v is some c
- if τ has the form $\tau_1 \to \tau_2$ then v has the form $\lambda x.\ e$

Proof: By inspection of the form of values and typing rules

- That is, by inversion, only one typing rule applies if $\tau =$ **int** and in that rule v is a constant, and similarly for $\tau_1 \to \tau_2$

Now prove Progress by induction on the derivation of $\cdot \vdash e : \tau$...

Progress: Induction on derivation of $\cdot \vdash e : \tau$

Derivation must end with one of four rules:

- $\cdot \vdash x : \tau$ — impossible because $\cdot \vdash e : \tau$
- $\cdot \vdash c : \textbf{int}$ — then e is a value
- $\cdot \vdash \lambda x.\ e : \tau$ — then e is a value
- $\cdot \vdash e_1\ e_2 : \tau$ where $\exists \tau'.\ \cdot \vdash e_1 : \tau' \to \tau$ and $\cdot \vdash e_2 : \tau'$
  By induction e_1 is some v_1 or can become some e_1'.
  If it can become e_1', then $e_1\ e_2 \to e_1'\ e_2$.
  Else by induction e_2 is some v_2 or can become some e_2'.
  If it becomes e_2', then $v_1\ e_2 \to v_1\ e_2'$.
  Else e is $v_1\ v_2$.
  $\cdot \vdash v_1 : \tau' \to \tau$ and Canonical Forms ensures v_1 has the form $\lambda x.\ e'$.
  So $v_1\ v_2 \to e'[v_2/x]$.

Note: If we add $+$, we need the other part of Canonical Forms.

Preservation

Lemma: If $\cdot \vdash e : \tau$ and $e \to e'$, then $\cdot \vdash e' : \tau$

Proof: By induction on the derivation of $\cdot \vdash e : \tau$. Bottom rule could conclude:

- $\cdot \vdash x : \tau$ — actually, it can't; $\cdot(x)$ doesn't exist
- $\cdot \vdash c : \textbf{int}$ — then $e \to e'$ is impossible, so lemma holds *vacuously*
- $\cdot \vdash \lambda x.\ e : \tau$ — then $e \to e'$ is impossible, so lemma holds *vacuously*
- $\cdot \vdash e_1\ e_2 : \tau$ where $\exists \tau'.\ \cdot \vdash e_1 : \tau' \to \tau$ and $\cdot \vdash e_2 : \tau'$
  There are 3 ways $e_1\ e_2 \to e'$ could be derived.
  Subcase for each ...

Preservation, $e = e_1\ e_2$ case

We have: $\cdot \vdash e_1 : \tau' \to \tau$, $\cdot \vdash e_2 : \tau'$, and $e_1\ e_2 \to e'$.
We need: $\cdot \vdash e' : \tau$.
The derivation of $e_1\ e_2 \to e'$ ensures 1 of these:

- e' is $e_1'\ e_2$ and $e_1 \to e_1'$:
  So with $\cdot \vdash e_1 : \tau' \to \tau$ and induction, $\cdot \vdash e_1' : \tau' \to \tau$.
  So with $\cdot \vdash e_2 : \tau'$ we can derive $\cdot \vdash e_1'\ e_2 : \tau$.
- e' is $e_1\ e_2'$ and $e_2 \to e_2'$:
  So with $\cdot \vdash e_2 : \tau'$ and induction, $\cdot \vdash e_2' : \tau'$.
  So with $\cdot \vdash e_1 : \tau' \to \tau$ we can derive $\cdot \vdash e_1\ e_2' : \tau$.
- e_1 is some $\lambda x.\ e_3$ and e_2 is some v and e' is $e_3[v/x]$
  Inverting $\cdot \vdash \lambda x.\ e_3 : \tau' \to \tau$ gives $\cdot, x{:}\tau' \vdash e_3 : \tau$.
  So it would suffice to know: If $\cdot, x{:}\tau' \vdash e_3 : \tau$ and $\cdot \vdash e_2 : \tau'$, then $\cdot \vdash e_3[e_2/x] : \tau$.
  That's true but we need to prove it via a Substitution Lemma.

Lemma (Substitution): If $\Gamma, x{:}\tau' \vdash e_1 : \tau$ and $\Gamma \vdash e_2 : \tau'$, then $\Gamma \vdash e_1[e_2/x] : \tau$.

Where are we

Almost done with Preservation, but in the case where $(\lambda x.\ e_3)\ e_2 \to e_3[e_2/x]$, need:

If $\cdot, x{:}\tau' \vdash e_3 : \tau$ and $\cdot \vdash e_2 : \tau'$, then $\cdot \vdash e_3[e_2/x] : \tau$.

- Intuitive: Replace assumption that $x : \tau'$ with an expression that has type τ'
- But we need an inductive proof because e_3 can be arbitrarily big and substitution is a subtle thing

Prove this lemma: If $\Gamma, x{:}\tau' \vdash e_1 : \tau$ and $\Gamma \vdash e_2 : \tau'$, then $\Gamma \vdash e_1[e_2/x] : \tau$.

- "Renaming" e_3 to e_1 in our "helper lemma" (no big deal)
- Strengthened induction hypothesis to work for any Γ
  - Else the proof will fail

Proving the Substitution Lemma

If $\Gamma, x{:}\tau' \vdash e_1 : \tau$ and $\Gamma \vdash e_2 : \tau'$, then $\Gamma \vdash e_1[e_2/x] : \tau$
Proof: By induction on derivation of $\Gamma, x{:}\tau' \vdash e_1 : \tau$

- $\Gamma, x{:}\tau' \vdash c : \textbf{int}$. Then $c[e_2/x] = c$ and $\Gamma \vdash c : \textbf{int}$
- $\Gamma, x{:}\tau' \vdash y : (\Gamma, x{:}\tau')(y)$.
  Either $y = x$ or $y \neq x$.
  If $y = x$, then $(\Gamma, x{:}\tau')(x)$ is τ' (i.e., $\tau = \tau'$) and $x[e_2/x]$ is e_2. So $\Gamma \vdash e_2 : \tau'$ satisfies the lemma.
  If $y \neq x$, then $(\Gamma, x{:}\tau')(y)$ is $\Gamma(y)$ and $y[e_2/x]$ is y.
  So we can derive $\Gamma \vdash y : \Gamma(y)$.
- $\Gamma, x{:}\tau' \vdash e_a\ e_b : \tau$.
  Then $\exists \tau_a, \tau_b$ where $\Gamma, x{:}\tau' \vdash e_a : \tau_a$ and $\Gamma, x{:}\tau' \vdash e_b : \tau_b$.
  So by induction $\Gamma \vdash e_a[e_2/x] : \tau_a$ and $\Gamma \vdash e_b[e_2/x] : \tau_b$.
  So we can derive $\Gamma \vdash e_a[e_2/x]\ e_b[e_2/x] : \tau$.
  And $(e_a\ e_b)[e_2/x]$ is $e_a[e_2/x]\ e_b[e_2/x]$.
- Only the (nested) function case is left ...

Still Proving Substitution

If $\Gamma, x{:}\tau' \vdash e_1 : \tau$ and $\Gamma \vdash e_2 : \tau'$, then $\Gamma \vdash e_1[e_2/x] : \tau$
Proof: By induction on derivation of $\Gamma, x{:}\tau' \vdash e_1 : \tau$

The last case uses these two technical lemmas (easy inductions):

- Exchange: If $\Gamma, x{:}\tau_1, y{:}\tau_2 \vdash e : \tau$, then $\Gamma, y{:}\tau_2, x{:}\tau_1 \vdash e : \tau$
- Weakening: If $\Gamma \vdash e : \tau$ and $x \notin \textbf{Dom}(\Gamma)$, then $\Gamma, x{:}\tau' \vdash e : \tau$.

The last case:

- $\Gamma, x{:}\tau' \vdash \lambda y.\ e_a : \tau$. (can assume $y \neq x$ and $y \notin \textbf{Dom}(\Gamma)$)
  Then $\exists \tau_a, \tau_b$ where $\Gamma, x{:}\tau', y{:}\tau_a \vdash e_a : \tau_b$ and $\tau = \tau_a \to \tau_b$.
  By *Exchange* $\Gamma, y{:}\tau_a, x{:}\tau' \vdash e_a : \tau_b$.
  By *Weakening* and $\Gamma \vdash e_2 : \tau'$, we know $\Gamma, y{:}\tau_a \vdash e_2 : \tau'$.
  So by induction (using $\Gamma, y{:}\tau_a$ for Γ), $\Gamma, y{:}\tau_a \vdash e_a[e_2/x] : \tau_b$.
  - (This is where we needed the stronger induction hypothesis)

  So we can derive $\Gamma \vdash \lambda y.\ e_a[e_2/x] : \tau_a \to \tau_b$.
  And $(\lambda y.\ e_a)[e_2/x]$ is $\lambda y.\ (e_a[e_2/x])$.

Lemma dependencies

Safety (evaluation never gets stuck)

- Preservation (to stay well-typed)
  - Substitution (β-reduction stays well-typed)
    - Weakening (substituting under nested λs well-typed)
    - Exchange (technical point)
- Progress (well-typed not stuck yet)
  - Canonical Forms (primitive reductions apply where needed)

Comments:

- Substitution strengthened to open terms for the proof
- When we add heaps, Preservation will use Weakening directly

Summary

What may seem a weird lemma pile is a powerful recipe:

Soundness: We don't get stuck because our induction hypothesis (typing) holds (Preservation) and stuck terms aren't well typed (contrapositive of Progress)

Preservation holds by induction on typing (replace subterms with same type) and Substitution (for β-reduction). Substitution must work over open terms and requires Weakening and Exchange.

Progress holds by induction on expressions (or typing) because either a subexpression progresses or we can make a *primitive reduction* (using Canonical Forms).

Induction on derivations – Another Look

Application cases ($e = e_1\ e_2$) are elegant and worth mastering

- Other constructs with eager evaluation of subexpressions would work analogously (e.g., $e_1 + e_2$ or (e_1, e_2))

For Preservation, lemma assumes $\cdot \vdash e_1\ e_2 : \tau$.

Inverting the typing derivation ensures it has the form:

$$\dfrac{\dfrac{\mathcal{D}_1}{\cdot \vdash e_1 : \tau' \to \tau} \qquad \dfrac{\mathcal{D}_2}{\cdot \vdash e_2 : \tau'}}{\cdot \vdash e_1\ e_2 : \tau}$$

One Preservation subcase: If $e_1\ e_2 \to e_1'\ e_2$, inverting that derivation means:

$$\dfrac{\dfrac{\mathcal{D}}{e_1 \to e_1'}}{e_1\ e_2 \to e_1'\ e_2}$$

continued...

The inductive hypothesis means there is a derivation of this form:

$$\dfrac{\mathcal{D}_3}{\cdot \vdash e_1' : \tau' \to \tau}$$

So a derivation of this form exists:

$$\dfrac{\dfrac{\mathcal{D}_3}{\cdot \vdash e_1' : \tau' \to \tau} \qquad \dfrac{\mathcal{D}_2}{\cdot \vdash e_2 : \tau'}}{\cdot \vdash e_1'\ e_2 : \tau}$$

(The app case of the Substitution Lemma is similar but we use induction twice to get the new derivation)