Programming Languages and Compilers (CS 421)

Reza Zamani Nasab

http://www.cs.illinois.edu/class/cs421/

Based in part on slides by Mattox Beckman, as updated by Vikram Adve, Gul Agha and Elsa Gunter

Type Declarations

- *Type declarations*: explicit assignment of types to variables (signatures to functions) in the code of a program
  - Must be checked in a strongly typed language
  - Often not necessary for strong typing or even static typing (depends on the type system)

Type Inference

- *Type inference*: A program analysis to assign a type to an expression from the program context of the expression
  - Fully static type inference first introduced by Robin Miller in ML
  - Haskle, OCAML, SML all use type inference
    - Records are a problem for type inference

Format of Type Judgments

- A *type judgement* has the form

$$\Gamma \text{ |- exp} : \tau$$

- Γ is a typing environment
  - Supplies the types of variables and functions
  - Γ is a list of the form [$x : \sigma$, . . .]
- exp is a program expression
- τ is a type to be assigned to exp
- |- pronounced "turnstyle", or "entails" (or "satisfies")

Example Valid Type Judgments

- [ ] |- true or false : bool
- [x : int] |- x + 3 : int
- [p : int -> string] |- p(5) : string

Format of Typing Rules

Assumptions

$$\frac{\Gamma_1 \mid\!- \text{exp}_1 : \tau_1 \quad . \; . \; . \; \Gamma_n \mid\!- \text{exp}_n : \tau_n}{\Gamma \mid\!- \text{exp} : \tau}$$

Conclusion

- Idea: Type of expression determined by type of components
- Rule without assumptions is called an *axiom*
- Γ may be omitted when not needed

Format of Typing Rules

Assumptions

$$\frac{\Gamma_1 \mid\!- \mathrm{exp}_1 : \tau_1 \quad . \; . \; . \quad \Gamma_n \mid\!- \mathrm{exp}_n : \tau_n}{\Gamma \mid\!- \mathrm{exp} : \tau}$$

Conclusion

- Γ, exp, τ are *parameterized* environments, expressions and types - *i.e.* may contain *meta-variables*

Axioms - Constants

$$\frac{}{\vdash n : \text{int}}$$ (assuming n is an integer constant)

$$\frac{}{\vdash \text{true} : \text{bool}} \qquad \frac{}{\vdash \text{false} : \text{bool}}$$

- These rules are true with any typing environment
- n is a meta-variable

Axioms - Variables

Notation: Let $\Gamma(x) = \sigma$ if $x : \sigma \in \Gamma$ and there is no $x : \tau$ to the left of $x : \sigma$ in Γ

Variable axiom:

$$\frac{}{\Gamma \mid\!- \, x : \sigma} \quad \text{if } \Gamma(x) = \sigma$$

Simple Rules - Arithmetic

Primitive operators ($\oplus \in$ { +, -, *, ...}):

$$\frac{\Gamma \mid\!- \; e_1 : \text{int} \qquad \Gamma \mid\!- \; e_2 : \text{int}}{\Gamma \mid\!- \; e_1 \oplus e_2 : \text{int}}$$

Relations ($\sim \; \in$ { < , > , =, <=, >= }):

$$\frac{\Gamma \mid\!- \; e_1 : \text{int} \qquad \Gamma \mid\!- \; e_2 : \text{int}}{\Gamma \mid\!- \; e_1 \sim e_2 : \text{bool}}$$

Simple Rules - Booleans

Connectives

$$\frac{\Gamma \mid\!- \; e_1 : \text{bool} \qquad \Gamma \mid\!- \; e_2 : \text{bool}}{\Gamma \mid\!- \; e_1 \;\&\&\; e_2 : \text{bool}}$$

$$\frac{\Gamma \mid\!- \; e_1 : \text{bool} \qquad \Gamma \mid\!- \; e_2 : \text{bool}}{\Gamma \mid\!- \; e_1 \;||\; e_2 : \text{bool}}$$

Simple Example

- Let Γ = [x:int ; y:bool]
- Show Γ |- y || (x + 3 > 6) : bool
- Start building the proof tree from the bottom up

$$\frac{?}{\Gamma \vdash y \mid\mid (x + 3 > 6) : bool}$$

Simple Example

- Let Γ = [x:int ; y:bool]
- Show Γ |- y || (x + 3 > 6) : bool
- Which rule has this as a conclusion?

$$\frac{?}{\Gamma \mid\!\!- \; y \,||\, (x + 3 > 6) : bool}$$

Simple Example

- Let Γ = [x:int ; y:bool]
- Show Γ |- y || (x + 3 > 6) : bool
- Booleans: ||

$$\frac{\Gamma \mid\!- \ y : \text{bool} \qquad \Gamma \mid\!- \ x + 3 > 6 : \text{bool}}{\Gamma \mid\!- \ y \mid\mid (x + 3 > 6) : \text{bool}}$$

Simple Example

- Let Γ = [x:int ; y:bool]
- Show Γ |- y || (x + 3 > 6) : bool
- Pick an assumption to prove

$$\frac{\dfrac{?}{\Gamma \text{ |- } y : \text{bool}} \qquad \Gamma \text{ |- } x + 3 > 6 : \text{bool}}{\Gamma \text{ |- } y \text{ || } (x + 3 > 6) : \text{bool}}$$

Simple Example

- Let Γ = [x:int ; y:bool]
- Show Γ |- y || (x + 3 > 6) : bool
- Which rule has this as a conclusion?

$$
\dfrac{\dfrac{?}{\Gamma \;|\!-\; y : \text{bool}} \qquad \Gamma \;|\!-\; x + 3 > 6 : \text{bool}}{\Gamma \;|\!-\; y \;||\; (x + 3 > 6) : \text{bool}}
$$

Simple Example

- Let Γ = [x:int ; y:bool]
- Show Γ |- y || (x + 3 > 6) : bool
- Axiom for variables

$$\dfrac{\dfrac{}{\Gamma \mid\!- \; y : \mathrm{bool}} \qquad \Gamma \mid\!- \; x + 3 > 6 : \mathrm{bool}}{\Gamma \mid\!- \; y \mid\mid (x + 3 > 6) : \mathrm{bool}}$$

Simple Example

- Let Γ = [x:int ; y:bool]
- Show Γ |- y || (x + 3 > 6) : bool
- Pick an assumption to prove

$$\dfrac{\dfrac{}{\Gamma \vdash y : \mathrm{bool}} \qquad \dfrac{?}{\Gamma \vdash x + 3 > 6 : \mathrm{bool}}}{\Gamma \vdash y \mid\mid (x + 3 > 6) : \mathrm{bool}}$$

Simple Example

- Let Γ = [x:int ; y:bool]
- Show Γ |- y || (x + 3 > 6) : bool
- Which rule has this as a conclusion?

$$\frac{\overline{\Gamma \text{ |- } y : bool} \qquad \overset{?}{\overline{\Gamma \text{ |- } x + 3 > 6 : bool}}}{\Gamma \text{ |- } y \text{ || } (x + 3 > 6) : bool}$$

Simple Example

- Let Γ = [x:int ; y:bool]
- Show Γ |- y || (x + 3 > 6) : bool
- Arithmetic relations

$$\dfrac{\dfrac{}{\Gamma \text{ |- } y : \text{bool}} \qquad \dfrac{\Gamma \text{ |- } x + 3 : \text{int} \quad \Gamma \text{ |- } 6 : \text{int}}{\Gamma \text{ |- } x + 3 > 6 : \text{bool}}}{\Gamma \text{ |- } y \text{ || } (x + 3 > 6) : \text{bool}}$$

Simple Example

- Let Γ = [x:int ; y:bool]
- Show Γ |- y || (x + 3 > 6) : bool
- Pick an assumption to prove

$$\dfrac{\dfrac{}{\Gamma \mid\!\!- \ y : \text{bool}} \qquad \dfrac{\Gamma \mid\!\!- \ x + 3 : \text{int} \qquad \dfrac{?}{\Gamma \mid\!\!- \ 6 : \text{int}}}{\Gamma \mid\!\!- \ x + 3 > 6 : \text{bool}}}{\Gamma \mid\!\!- \ y \mid\mid (x + 3 > 6) : \text{bool}}$$

Simple Example

- Let Γ = [x:int ; y:bool]
- Show Γ |- y || (x + 3 > 6) : bool
- Which rule has this as a conclusion?

$$\dfrac{\dfrac{}{\Gamma\ |\!-\ y : \text{bool}} \qquad \dfrac{\Gamma\ |\!-\ x + 3 : \text{int} \qquad \dfrac{?}{\Gamma\ |\!-\ 6 : \text{int}}}{\Gamma\ |\!-\ x + 3 > 6 : \text{bool}}}{\Gamma\ |\!-\ y\ ||\ (x + 3 > 6) : \text{bool}}$$

Simple Example

- Let Γ = [x:int ; y:bool]
- Show Γ |- y || (x + 3 > 6) : bool
- Axiom for constants

$$\dfrac{\dfrac{}{\Gamma \;|\!\!-\; y : \text{bool}} \qquad \dfrac{\Gamma \;|\!\!-\; x + 3 : \text{int} \qquad \dfrac{}{\Gamma \;|\!\!-\; 6 : \text{int}}}{\Gamma \;|\!\!-\; x + 3 > 6 : \text{bool}}}{\Gamma \;|\!\!-\; y \;||\; (x + 3 > 6) : \text{bool}}$$

Simple Example

- Let Γ = [x:int ; y:bool]
- Show Γ |- y || (x + 3 > 6) : bool
- Pick an assumption to prove

$$
\dfrac{\dfrac{}{\Gamma \;|\!\!-\; y : \text{bool}} \qquad \dfrac{\dfrac{?}{\Gamma \;|\!\!-\; x + 3 : \text{int}} \quad \dfrac{}{\Gamma \;|\!\!-\; 6 : \text{int}}}{\Gamma \;|\!\!-\; x + 3 > 6 : \text{bool}}}{\Gamma \;|\!\!-\; y \;||\; (x + 3 > 6) : \text{bool}}
$$

Simple Example

- Let Γ = [x:int ; y:bool]
- Show Γ |- y || (x + 3 > 6) : bool
- Which rule has this as a conclusion?

$$\dfrac{\dfrac{}{\Gamma \;|\!\!-\; y : \mathrm{bool}} \qquad \dfrac{\dfrac{?}{\Gamma \;|\!\!-\; x + 3 : \mathrm{int}} \quad \dfrac{}{\Gamma \;|\!\!-\; 6 : \mathrm{int}}}{\Gamma \;|\!\!-\; x + 3 > 6 : \mathrm{bool}}}{\Gamma \;|\!\!-\; y \;||\; (x + 3 > 6) : \mathrm{bool}}$$

Simple Example

- Let Γ = [x:int ; y:bool]
- Show Γ |- y || (x + 3 > 6) : bool
- Arithmetic operations

$$
\dfrac{\dfrac{}{\Gamma \mid\!- \ y : \mathrm{bool}} \qquad \dfrac{\dfrac{\Gamma \mid\!- \ x : \mathrm{int} \quad \Gamma \mid\!- \ 3 : \mathrm{int}}{\Gamma \mid\!- \ x + 3 : \mathrm{int}} \quad \dfrac{}{\Gamma \mid\!- \ 6 : \mathrm{int}}}{\Gamma \mid\!- \ x + 3 > 6 : \mathrm{bool}}}{\Gamma \mid\!- \ y \ || \ (x + 3 > 6) : \mathrm{bool}}
$$

Simple Example

- Let Γ = [x:int ; y:bool]
- Show Γ |- y || (x + 3 > 6) : bool
- Pick an assumption to prove

$$\dfrac{\dfrac{}{\Gamma \;|\!\!-\; y : \text{bool}} \qquad \dfrac{\dfrac{\Gamma \;|\!\!-\; x : \text{int} \quad \dfrac{?}{\Gamma \;|\!\!-\; 3 : \text{int}}}{\Gamma \;|\!\!-\; x + 3 : \text{int}} \qquad \dfrac{}{\Gamma \;|\!\!-\; 6 : \text{int}}}{\Gamma \;|\!\!-\; x + 3 > 6 : \text{bool}}}{\Gamma \;|\!\!-\; y \;||\; (x + 3 > 6) : \text{bool}}$$

Simple Example

- Let Γ = [x:int ; y:bool]
- Show Γ |- y || (x + 3 > 6) : bool
- Which rule has this as a conclusion?

$$\dfrac{\dfrac{}{\Gamma \mid\!- \; y : \text{bool}} \qquad \dfrac{\dfrac{\Gamma \mid\!- \; x : \text{int} \quad \dfrac{?}{\Gamma \mid\!- \; 3 : \text{int}}}{\Gamma \mid\!- \; x + 3 : \text{int}} \qquad \dfrac{}{\Gamma \mid\!- \; 6 : \text{int}}}{\Gamma \mid\!- \; x + 3 > 6 : \text{bool}}}{\Gamma \mid\!- \; y \mid\mid (x + 3 > 6) : \text{bool}}$$

Simple Example

- Let Γ = [x:int ; y:bool]
- Show Γ |- y || (x + 3 > 6) : bool
- Axiom for constants

$$\dfrac{\dfrac{}{\Gamma\ |\!-\ y : \text{bool}} \qquad \dfrac{\dfrac{\Gamma\ |\!-\ x : \text{int} \quad \dfrac{}{\Gamma\ |\!-\ 3 : \text{int}}}{\Gamma\ |\!-\ x + 3 : \text{int}} \quad \dfrac{}{\Gamma\ |\!-\ 6 : \text{int}}}{\Gamma\ |\!-\ x + 3 > 6 : \text{bool}}}{\Gamma\ |\!-\ y\ ||\ (x + 3 > 6) : \text{bool}}$$

Simple Example

- Let Γ = [x:int ; y:bool]
- Show Γ |- y || (x + 3 > 6) : bool
- Pick an assumption to prove

$$\dfrac{\dfrac{}{\Gamma \mid\!- \ y : \text{bool}} \qquad \dfrac{\dfrac{\dfrac{?}{\Gamma \mid\!- \ x : \text{int}} \quad \dfrac{}{\Gamma \mid\!- \ 3 : \text{int}}}{\Gamma \mid\!- \ x + 3 : \text{int}} \quad \dfrac{}{\Gamma \mid\!- \ 6 : \text{int}}}{\Gamma \mid\!- \ x + 3 > 6 : \text{bool}}}{\Gamma \mid\!- \ y \mathrel{||} (x + 3 > 6) : \text{bool}}$$

Simple Example

- Let Γ = [x:int ; y:bool]
- Show Γ |- y || (x + 3 > 6) : bool
- Which rule has this as a conclusion?

$$\dfrac{\dfrac{}{\Gamma \;|\!-\; y : \text{bool}} \qquad \dfrac{\dfrac{\dfrac{?}{\Gamma \;|\!-\; x : \text{int}} \quad \dfrac{}{\Gamma \;|\!-\; 3 : \text{int}}}{\Gamma \;|\!-\; x + 3 : \text{int}} \quad \dfrac{}{\Gamma \;|\!-\; 6 : \text{int}}}{\Gamma \;|\!-\; x + 3 > 6 : \text{bool}}}{\Gamma \;|\!-\; y \,||\, (x + 3 > 6) : \text{bool}}$$

Simple Example

- Let Γ = [x:int ; y:bool]
- Show Γ |- y || (x + 3 > 6) : bool
- Axiom for variables

$$
\dfrac{\dfrac{}{\Gamma \mid\!\!- \; y : \text{bool}} \qquad \dfrac{\dfrac{\dfrac{}{\Gamma \mid\!\!- \; x : \text{int}} \quad \dfrac{}{\Gamma \mid\!\!- \; 3 : \text{int}}}{\Gamma \mid\!\!- \; x + 3 : \text{int}} \qquad \dfrac{}{\Gamma \mid\!\!- \; 6 : \text{int}}}{\Gamma \mid\!\!- \; x + 3 > 6 : \text{bool}}}{\Gamma \mid\!\!- \; y \mid\mid (x + 3 > 6) : \text{bool}}
$$

Simple Example

- Let Γ = [x:int ; y:bool]
- Show Γ |- y || (x + 3 > 6) : bool
- No more assumptions! DONE!

$$\dfrac{\dfrac{}{\Gamma \;|\text{-}\; y : \text{bool}} \qquad \dfrac{\dfrac{\dfrac{}{\Gamma \;|\text{-}\; x : \text{int}} \quad \dfrac{}{\Gamma \;|\text{-}\; 3 : \text{int}}}{\Gamma \;|\text{-}\; x + 3 : \text{int}} \quad \dfrac{}{\Gamma \;|\text{-}\; 6 : \text{int}}}{\Gamma \;|\text{-}\; x + 3 > 6 : \text{bool}}}{\Gamma \;|\text{-}\; y \;||\; (x + 3 > 6) : \text{bool}}$$

Type Variables in Rules

- If_then_else rule:

$$\frac{\Gamma \mid\!- \; e_1 : \text{bool} \quad \Gamma \mid\!- \; e_2 : \tau \quad \Gamma \mid\!- \; e_3 : \tau}{\Gamma \mid\!- \; (\text{if } e_1 \text{ then } e_2 \text{ else } e_3) : \tau}$$

- τ is a type variable (meta-variable)
- Can take any type at all
- All instances in a rule application must get same type
- Then branch, else branch and if_then_else must all have same type

Function Application

- Application rule:

$$\frac{\Gamma \mid\!- \ e_1 : \tau_1 \rightarrow \tau_2 \quad \Gamma \mid\!- \ e_2 : \tau_1}{\Gamma \mid\!- \ (e_1 \ e_2) : \tau_2}$$

- If you have a function expression e_1 of type $\tau_1 \rightarrow \tau_2$ applied to an argument of type τ_1, the resulting expression has type τ_2

Application Examples

$$\frac{\Gamma \mid\!- \text{print\_int} : \text{int} \rightarrow \text{unit} \qquad \Gamma \mid\!- 5 : \text{int}}{\Gamma \mid\!- (\text{print\_int } 5) : \text{unit}}$$

- e_1 = print_int, e_2 = 5,
- τ_1 = int, τ_2 = unit

$$\frac{\Gamma \mid\!- \text{map print\_int} : \text{int list} \rightarrow \text{unit list} \qquad \Gamma \mid\!- [3;7] : \text{int list}}{\Gamma \mid\!- (\text{map print\_int } [3;\ 7]) : \text{unit list}}$$

- e_1 = map print_int, e_2 = [3; 7],
- τ_1 = int list, τ_2 = unit list

Fun Rule

- Rules describe types, but also how the environment Γ may change
- Can only do what rule allows!
- fun rule:

$$\frac{[x : \tau_1] + \Gamma \text{ |- } e : \tau_2}{\Gamma \text{ |- fun } x \text{ -> } e : \tau_1 \rightarrow \tau_2}$$

Fun Examples

$$\frac{[y : int\ ] + \Gamma\ |\!-\ y + 3\ :\ int}{\Gamma\ |\!-\ fun\ y\ \text{->}\ y + 3\ :\ int \rightarrow int}$$

$$\frac{[f : int \rightarrow bool] + \Gamma\ |\!-\ f\ 2\ ::\ [true]\ :\ bool\ list}{\Gamma\ |\!-\ (fun\ f\ \text{->}\ f\ 2\ ::\ [true])\ :\ (int \rightarrow bool) \rightarrow bool\ list}$$

Let and Let Rec

- let rule:

$$\frac{\Gamma \mid\!- \ e_1 : \tau_1 \qquad [x : \tau_1] + \Gamma \mid\!- \ e_2 : \tau_2}{\Gamma \mid\!- \ (\text{let x} = e_1 \text{ in } e_2) : \tau_2}$$

- let rec rule:

$$\frac{[x : \tau_1] + \Gamma \mid\!- \ e_1 : \tau_1 \quad [x : \tau_1] + \Gamma \mid\!- \ e_2 : \tau_2}{\Gamma \mid\!- \ (\text{let rec x} = e_1 \text{ in } e_2) : \tau_2}$$

Mia Copa

- The above system can't handle polymorphism as in OCAML
- No type variables in type language (only meta-variable in the logic)
- Would need:
  - Object level type variables and some kind of type quantification
  - **let** and **let rec** rules to introduce polymorphism
  - Explicit rule to eliminate (instantiate) polymorphism

Example

- Which rule do we apply?

$$\frac{?}{\text{|- (let rec one = 1 :: one in let x = 2 in fun y -> (x :: y :: one)) : int} \rightarrow \text{int list}}$$

|- (let rec one = 1 :: one in

let x = 2 in

fun y -> (x :: y :: one)) : int → int list

Example

- Let rec rule:

1

[one : int list] |-

(1 :: one) : int list

2

[one : int list] |-

(let x = 2 in

fun y -> (x :: y :: one))

: int $\rightarrow$ int list

|- (let rec one = 1 :: one in

let x = 2 in

fun y -> (x :: y :: one)) : int $\rightarrow$ int list

Proof of 1

- Which rule?

[one : int list] |- (1 :: one) : int list

Proof of 1

- Application

$$\frac{\overset{③}{[\text{one} : \text{int list}] \mid\!- ((::)\ 1): \text{int list} \to \text{int list}} \qquad \overset{④}{[\text{one} : \text{int list}] \mid\!- \text{one} : \text{int list}}}{[\text{one} : \text{int list}] \mid\!- (1 :: \text{one}) : \text{int list}}$$

Proof of 3

$$\dfrac{\dfrac{\text{Constants Rule}}{[\text{one} : \text{int list}] \mid\!-\ (::) : \text{int} \rightarrow \text{int list} \rightarrow \text{int list}} \qquad \dfrac{\text{Constants Rule}}{[\text{one} : \text{int list}] \mid\!-\ 1 : \text{int}}}{[\text{one} : \text{int list}] \mid\!-\ ((::)\ 1) : \text{int list} \rightarrow \text{int list}}$$

Proof of 4

- Rule for variables

$$\frac{}{[\text{one} : \text{int list}] \mid\!- \text{one}:\text{int list}}$$

Proof of 2

- Constant

5 [x:int; one : int list] |- fun y -> (x :: y :: one)) : int → int list

$$\frac{\dfrac{}{\text{[one : int list] |- 2:int}} \qquad \text{[x:int; one : int list] |- fun y -> (x :: y :: one)) : int} \rightarrow \text{int list}}{\text{[one : int list] |- (let x = 2 in fun y -> (x :: y :: one)) : int} \rightarrow \text{int list}}$$

Proof of 5

$$\frac{?}{[x:int;\ one : int\ list] \mid\!- \ fun\ y \text{ -> } (x :: y :: one)) \\ : int \rightarrow int\ list}$$

Proof of 5

$$\frac{\dfrac{?}{\text{[y:int; x:int; one : int list] |- (x :: y :: one) : int list}}}{\text{[x:int; one : int list] |- fun y -> (x :: y :: one)) : int} \rightarrow \text{int list}}$$

Proof of 5

6

[y:int; x:int; one : int list] |-
((::) x):int list$\rightarrow$ int list

7

[y:int; x:int; one : int list] |-
(y :: one) : int list

[y:int; x:int; one : int list] |- (x :: y :: one) : int list

[x:int; one : int list] |- fun y -> (x :: y :: one))

: int $\rightarrow$ int list

Proof of 6

Constant   Variable

$$\dfrac{\dfrac{}{[...] \mid\!- (::) : int \to int\ list \to int\ list} \quad \dfrac{}{[...; x:int;...] \mid\!- x:int}}{[y:int;\ x:int;\ one : int\ list] \mid\!- ((::)\ x) : int\ list \to int\ list}$$

Proof of 7

Pf of 6 [y/x] Variable

⋮

[y:int; ...] |- ((::) y) :int list$\rightarrow$ int list   [...; one: int list] |- one: int list

[y:int; x:int; one : int list] |- (y :: one) : int list

Curry - Howard Isomorphism

- Type Systems are logics; logics are type systems
- Types are propositions; propositions are types
- Terms are proofs; proofs are terms

- Functions space arrow corresponds to implication; application corresponds to modus ponens

Curry - Howard Isomorphism

- Modus Ponens

$$\frac{A \Rightarrow B \quad A}{B}$$

- Application

$$\frac{\Gamma \mid\!\!- \; e_1 : \alpha \rightarrow \beta \quad \Gamma \mid\!\!- \; e_2 : \alpha}{\Gamma \mid\!\!- \; (e_1 \; e_2) : \beta}$$