CS653: Functional Programming
2017-18 II^{nd} Semester

Type Checking and Type Inferencing

Amey Karkare

`karkare@cse.iitk.ac.in`

`http://www.cse.iitk.ac.in/~karkare/cs653`

Department of CSE, IIT Kanpur

- Performance*
  - Quiz 1: Min: 0, Mean: 51, Max: 80 (full!!)
  - Midsem: Min: 3, Mean: 42, Max: 100 (full!!)

*Before regrading.

Agenda

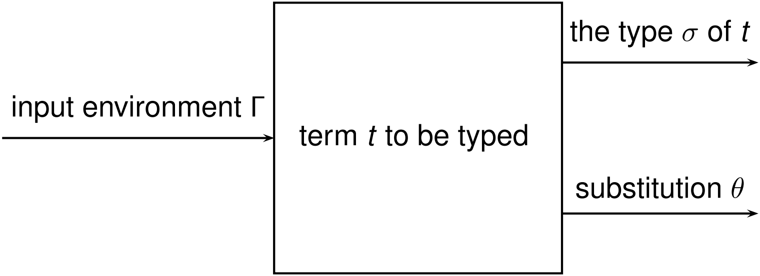
- Type Inferencing

Acknowledgements

- The slides are based on Amitabha Sanyal's notes on types.

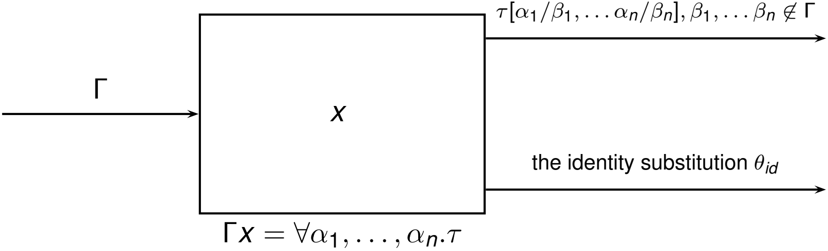
The Hindley-Milner Algorithm

By case analysis on the term *t* in the following diagram:



Hindley-Milner - Type checking variables

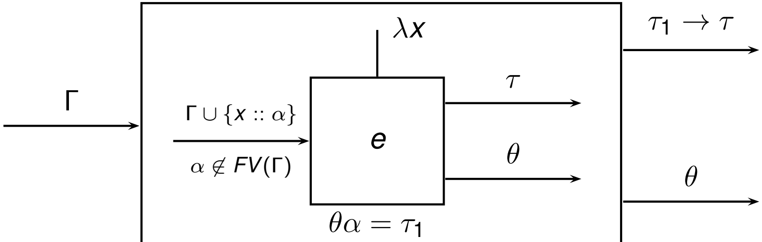
t is a variable *x*



- $\beta_1, \ldots, \beta_n$ are fresh variables.
- Reason for monomorphising the type of x: We try to find the type of a variable only in the context of an application, and our application is monomorphic.

Hindley-Milner - Type checking abstractions

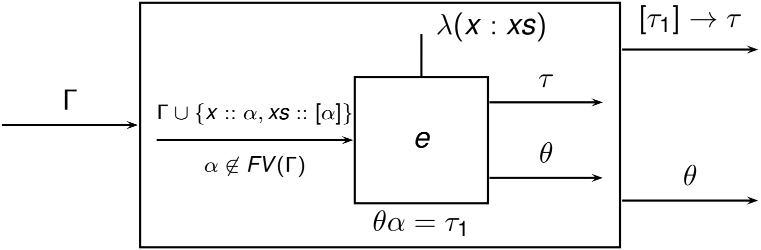
t is a lambda abstraction $\lambda x.e$



- Typecheck e in an environment Γ augmented with an assumed type α for x. Assume that result is a type τ and a substitution θ.
- Let the (possibly refined) type of α in θ be τ_1.
- The type of $\lambda x.e$ is $\tau_1 \to \tau$. The final substitution is also θ.

Hindley-Milner - Type checking abstractions

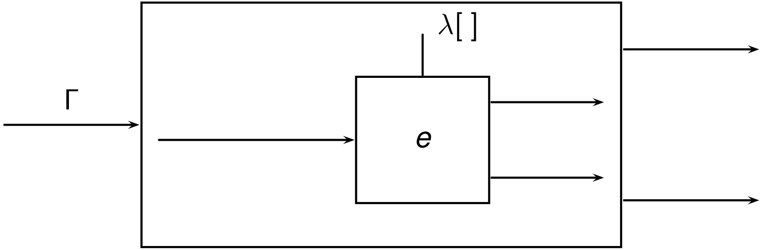
t is a pattern matching lambda abstraction $\lambda(x : xs).e$



- Typecheck e in an environment Γ augmented with assumed types α and $[\alpha]$ for x and xs. Let result type be τ and the substitution be θ.
- Let the (possibly) refined type of α in θ be τ_1.
- The type of $\lambda x.e$ is $[\tau_1] \rightarrow \tau$ and the final substitution is θ.

Hindley-Milner - Type checking abstractions

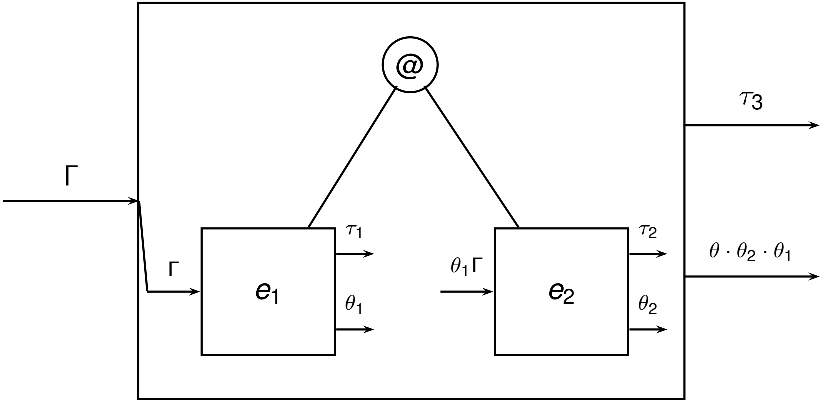
t is a pattern matching lambda abstraction $\lambda[\,].e$



- What will you do in this case?

Hindley-Milner - Type checking applications

t is an application $(e_1\ e_2)$



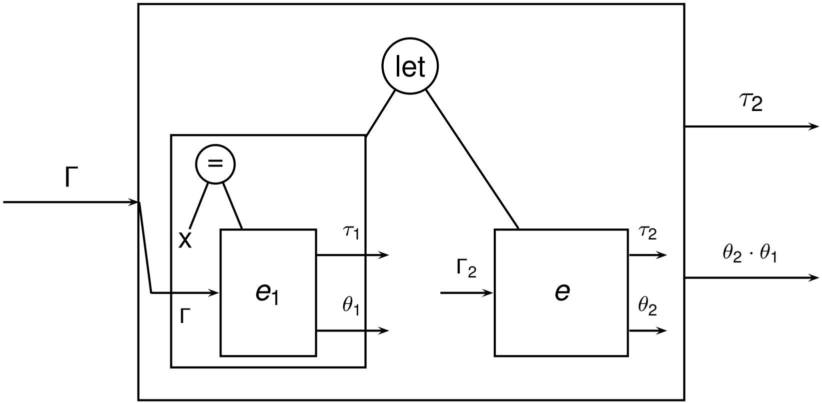
θ = unify($\theta_2\ \tau_1, \tau_2 \to \alpha$) and $\theta\alpha = \tau_3$

Hindley-Milner - Type checking applications

- Typecheck e_1 with the initial environment Γ. Result is τ_1 and θ_1.
- Typecheck e_2 with the environment $\theta_1\Gamma$. Result is τ_2 and θ_2.
- Unify $\theta_2\tau_1$ and $\tau_2 \rightarrow \alpha$. Assume that unifier is θ. And the unified term $(\theta\alpha)$ is τ_3.
- Type of the application is τ_3 and the final substitution is $\theta \cdot \theta_2 \cdot \theta_1$.

Hindley-Milner - Type checking `let`s

t is a let expression `let` $x = e_1$ `in` e



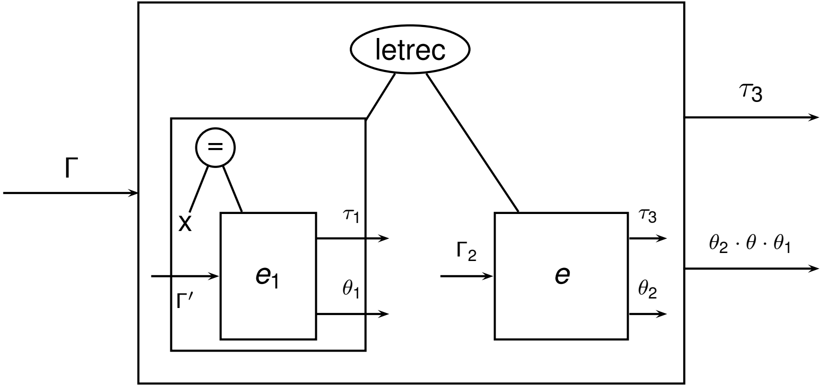
$\sigma = \forall\alpha_1 \ldots \alpha_n.\tau_1,\ \alpha_1 \ldots \alpha_n \in \tau_1,$
$\alpha_1 \ldots \alpha_n \notin FV(\Gamma),\ \Gamma_2 = \theta_1\Gamma \cup \{x :: \sigma\}$

Hindley-Milner - Type checking `let`s

- Typecheck e_1 in the environment Γ, resulting in a type τ_1 and a substitution θ_1
- Let σ be a polymorphic form of τ_1 and let $\Gamma_2 = \theta_1\Gamma_1 \cup \{x :: \sigma\}$.
- Typecheck e in environment Γ_2. Assume result is τ_2 and θ_2.
- Type of `let` is τ_2, and the final substitution is $\theta_2 \cdot \theta_1$.

Hindley-Milner - Type checking `letrec`s

t is `letrec` $x = e_1$ `in` e



$\Gamma' = \Gamma \uplus \{x :: \alpha\}$
$\tau_2 = \theta_1 \alpha, \theta = \text{unify}(\tau_1, \tau_2), \tau' = \theta \tau_1$
$\sigma = \forall \alpha_1 \ldots \alpha_n . \tau', \ \alpha_1 \ldots \alpha_n \in \tau',$
$\alpha_1 \ldots \alpha_n \notin FV(\Gamma), \ \Gamma_2 = \theta_1 \Gamma \uplus \{x :: \sigma\}$

Hindley-Milner - Type checking `letrec`s

- Typecheck e_1 in environment Γ augmented with a type assumption α for the variable x. Assume the the result is a type τ_1 and a changed environment Γ_1.
- Let τ_2 be the refined type of x in Γ_1. Unify this with the type τ_1 of e_1 . Let the unifier be θ and the unified type be τ'.
- Let σ be an appropriate polymorphic form of τ'. Also let Γ_2 be Γ_1 modified taking the unification process into account and further augmented with the type of x as σ.
- Typecheck e in the environment Γ_2 resulting in a type τ_3 and a modified environment Γ_3.
- The type of the let expression is τ_3, and the modified environment is Γ_3 with the type of x deleted.