

UNIVERSIDADE FEDERAL DE SANTA MARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA - PPGI

Subtipos e Metateoria dos Subtipos

LINGUAGENS DE PROGRAMAÇÃO – ELC921
PROF^A DR^A JULIANA KAISER VIZZOTTO

ALUNOS: Alberto Kummer, Daniel Di Domenico, Fernando
Campagnolo, Jéssica Lasch de Moura e José Puiati

15 Subtyping

- Também chamado de *subtype polymorphism*;
- Característica presente nas linguagens orientadas a objetos;
- Cálculo Lambda simplesmente tipado com subtipos: $\lambda_{<}$:

15.1 Subsumption

- Sem subtipos:

- ✓ Regras de tipos bastante rígidas;
- ✓ Rejeição de expressões que, aos olhos do programador, são bem tipadas.

- Exemplo:

$$\frac{\Gamma \vdash t_1 : T_{11} \rightarrow T_{12} \quad \Gamma \vdash t_2 : T_{11}}{\Gamma \vdash t_1 t_2 : T_{11}} \quad (\text{T-APP})$$

$(\lambda r : \{x : \text{Nat}\}. \quad r.x) \quad \{x = 0, y = 1\}$ Inválido?

15.1 Subsumption

- Sem subtipos:

- ✓ Regras de tipos bastante rígidas;
- ✓ Rejeição de expressões que, aos olhos do programador, são bem tipadas.

- Exemplo:

$$\frac{\Gamma \vdash t_1 : T_{11} \rightarrow T_{12} \quad \Gamma \vdash t_2 : T_{11}}{\Gamma \vdash t_1 t_2 : T_{11}} \quad (\text{T-APP})$$

$(\lambda r : \{x : \text{Nat}\}. \quad r.x) \quad \{x = 0, y = 1\}$ Válido

15.1 Subsumption

■ Objetivo dos subtipos:

- ✓ Refinamento das regras de tipos;
- ✓ Se S é subtipo de T ($S <: T$), qualquer termo do tipo S pode ser utilizado no contexto onde T é esperado;
- ✓ Princípio da substituição segura (*safe substitution*).

Regra *Subsumption*

$$\frac{\Gamma \vdash t : S \quad S <: T}{\Gamma \vdash t : T} \quad (\text{T-SUB})$$

Adaptado de (Pierce, 2002).

15.1 Subsumption

- Exemplo: $(\lambda r: \{x : \text{Nat}\}. r.x) \quad \{x = 0, y = 1\}$
 - ✓ Considerando que: $\{x : \text{Nat}, y : \text{Nat}\} <: \{x : \text{Nat}\}$
 - ✓ A regra T-SUB permite a aplicação pois são tipos válidos.

15.4 The Top and Bottom Types

Formas de subtipos - Top e Bot

Formas sintáticas:

$$\begin{array}{l} T ::= \dots \\ \quad \textit{Top} \\ \quad \textit{Bot} \end{array}$$

Regras de subtipos:

$$\begin{array}{ll} S <: \textit{Top} & (\text{S-Top}) \\ \textit{Bot} <: T & (\text{S-Bot}) \end{array}$$

Adaptado de (Pierce, 2002).

15.4 The Top and Bottom Types

■ Top:

- ✓ Elemento **máximo** da relação de subtipos;
- ✓ Equivale ao tipo *Object* das linguagens orientadas a objetos;
- ✓ Dispositivo técnico sofisticado em sistemas que combinam subtipos com poliformismo.

■ Bot:

- ✓ Elemento **mínimo** da relação de subtipos;
- ✓ Tipo vazio (não existem valores do tipo Bot);
- ✓ Muito útil para expressar algumas operações que não visam retorno de valores, como exceções, pois:
 - Permite ao programador definir expressões sem retorno com o tipo Bot;
 - Indica ao *typechecker* que a expressão pode ser utilizada com segurança em qualquer contexto.

15.4 The Top and Bottom Types

- Exemplo:

$\lambda x : T.$

if $\langle \text{valor apropriado para } x \rangle$ then

$\langle \text{calcula o resultado} \rangle$

else

 error

15.4 The Top and Bottom Types

- Tipo Bot dificulta a implementação;
- Mudança da regra de tipo da aplicação:
 - ✓ $t_1 \ t_2 :$
 - ✓ t_1 pode ser tanto do tipo seta ($T_1 \rightarrow T_2$) ou do tipo Bot.

SLIDES KADICO

B.C. Pierce. *Types and Programming Languages*. MIT Press, 2002. ISBN 9780262162098.