

# EEL670 Trabalho 08

Erica Ferreira

Abril 2021

## 1.

Considere uma linguagem qualquer com tipos inteiro e real na qual as seguintes expressões são legais:

1.  $1+2$
2.  $1.0+2$
3.  $1+2.0$
4.  $1.0+2.0$

### **$1+2$**

Como os dois têm o mesmo tipo, a resposta será 3 para todos os casos.

### **$1.0+2$**

#### **a) Coercion sem Overloading**

Como não tem overloading, vamos considerar que a função  $+$  só funciona com inteiros, o Coercion vai mudar o tipo de 1.0 que antes era float para int, tendo como o resultado 3.

#### **b) Overloading sem Coercion**

Com o overloading, uma função pode receber vários tipos porém para fazer a soma, os dois têm que ser iguais, logo daria erro.

#### **c) Coercion e Overloading**

Apesar do Overloading ter precedência em relação ao Coercion, a função não pode operar com tipos diferentes, por isso o Coercion vem primeiro, logo o Coercion vai transformar o 1(float) em int e a função  $+$  agora com Overloading vai aceitar operar floats, logo temos como resultado 3.

#### **d) Subtipos sem Coercion e sem Overloading**

Erro, o 1.0 não é subtipo do tipo padrão da função de soma (inteiros) e como não temos Coercion, eles não serão convertidos nem a função aceitará outros tipos já que não tem Overloading.

#### **e) Em ML**

Como em ML não faz nenhum tipo de Coercion, teremos erro.

**1+2.0**

#### **a) Coercion sem Overloading**

Como não tem Overloading, vamos considerar que a função + só funciona com inteiros, o Coercion vai mudar o tipo de 2.0 que antes era float para int, tendo como o resultado 3.

#### **b) Overloading sem Coercion**

Com o Overloading, uma função pode receber vários tipos porém para fazer a soma, os dois têm que ser iguais, logo daria erro

#### **c) Coercion e Overloading**

Apesar do Overloading ter precedência em relação ao Coercion, a função não pode operar com tipos diferentes, por isso o Coercion vem primeiro, logo o Coercion vai transformar o 2(int) em float e a função + agora com Overloading vai aceitar operar floats, logo temos como resultado 3.0.

#### **d) Subtipos sem Coercion e sem Overloading**

Erro, o 2.0 não é subtipo do tipo padrão da função de soma (inteiros) e como não temos Coercion, eles não serão convertidos nem a função aceitará outros tipos já que não tem Overloading.

#### **e) Em ML**

Como em ML não faz nenhum tipo de Coercion, teremos erro, uma vez que o operador + não opera com tipos diferentes ao mesmo tempo.

**1.0 + 2.0**

**a) Coercion sem Overloading**

Resultado 3, sem o Overloading o operador + só aceitaria inteiros, logo o Coercion transformaria os números em int, uma vez que no Coercion, a função que define o tipo.

**b) Overloading sem Coercion**

Resultado 3.0, como os tipos são iguais, não há a necessidade de conversão (Coercion).

**c) Coercion e Overloading**

Resultado 3.0, mesmo tipo e com função aceitando vários tipos (Overloading).

**d) Subtipos sem Coercion e sem Overloading**

Erro, considerando que a função + tem int como tipo padrão, mesmo sendo do mesmo tipo, sem Overloading, dará erro.

**e) Em ML**

Os tipos são iguais e como ML suporta Overloading, logo temos que o resultado é igual a 3.0.