UDESC UNIVERSIDADE DO ESTADO DE SANTA CATARINA

Exercícios Unificação

Ciência da Computação

1) Considere a seguinte gramática, representando uma versão simplificada do sistema de tipos da linguagem Haskell:

τ	::=	int	Números inteiros
	|	a	Variáveis
	|	$\tau \to \tau$	Funções

...isto é, um tipo pode ser representado por **int**, o número dos inteiros, por uma variável de tipos (aqui representada por *letras minúsculas*), ou por funções, representadas por uma seta entre dois subtipos.

Para as seguintes expressões, indique se é possível unificar os dois tipos ou não, e, se for, qual o unificador mais geral capaz de tornar as expressões iguais:

a) a ~ **int**
b) **int** ~ b
c) **int** ~ **int**
d) a→b ~ c→c
e) a→b ~ (a→c)→d
f) **int** ~ a→b
g) a→b→**int** ~ **int**→c
h) (a→b)→a ~ d→e
i) (a→a)→b ~ b→**int**→**int**
j) (a→b)→c→d ~ x→y

Lembrando que a seta associa para a esquerda, isto é, a→b→c = a→(b→c).

2) Ainda considerando a gramática acima, considere as seguintes expressões Haskell:

```
fmap :: (a -> b) -> [a] -> [b]
(+) :: Int -> Int -> Int
```

Ao informarmos a seguinte expressão, `fmap (+)`, o inferidor de tipos de Haskell irá tentar unificar o primeiro argumento da função fmap com o tipo do parâmetro fornecido. Qual será então o tipo inferido para a expressão?

3) Considere a seguinte gramática para fórmulas de primeira ordem:

e	::=	x	Átomos
	|	a	Variáveis
	|	$x(e_1, ..., e_n)$	Predicados

...isto é, uma fórmula pode ser um átomo (fixos, representados por *letras minúsculas*), uma variável (aqui representadas por *letras maiúsculas*), ou predicados, que contém um átomo à sua esquerda e N subfórmulas dentro de parênteses.

Para as seguintes expressões, indique se é possível unificar as duas fórmulas ou não, e, se for, qual o unificador mais geral capaz de tornar as expressões iguais:

a) X ~ foo
b) bar ~ Y
c) foo() ~ foo()
d) foo() ~ bar()
e) color(apple, red) ~ color(apple, C)
f) list(a, b, C) ~ list(X)
g) X ~ cons(10, X)
h) append(nil, cons(10, nil)) ~ append(X, Y)
i) append(nil, cons(10, nil)) ~ append(X, cons(H, T))

Note que as regras para unificação de tipos e predicados, conforme vistas acima, estão listadas nos slides fornecidos.