

Tópicos em LP

Fabio Mascarenhas – 2018.1

<http://www.dcc.ufrj.br/~fabiom/comp2>

Inferência de tipos

- A unificação consegue achar uma associação entre variáveis e tipos que torna dois tipos iguais, caso ela exista
- Caso os dois tipos não tenham nenhuma variável em aberto, a unificação faz o mesmo trabalho da igualdade
- Podemos trocar todas as comparações de tipos que fazemos em nosso verificador por unificações, e permitir que o programador omita declarações de tipos
- Uma declaração omitida é substituída por uma variável em aberto, e a unificação vai dizer qual tipo essa declaração deveria ter

Inferência de tipos - generalização

- Quando generalizamos um tipo, uma variável em aberto é como um parâmetro de tipo:

```
function bar(x)
  local baz = function (y)
    return x
  end
  return baz(1)
end
```

- Quando generalizamos bar, o tipo dela tem uma variável em aberto, que foi associado à variável x e por unificação foi parar no tipo de retorno
- A generalização transforma essas variáveis de tipo em parâmetros de tipo genéricos

Inferência de tipos - especialização

- A especialização de tipos agora pode deixar variáveis de tipo em aberto, que vão seguir no processo de inferência, permitindo inferir tipos como o da função abaixo:

```
function primeiro(s)
    return seq_byte(s, 1)
end
```

- Também podemos inferir tipos mais específicos:

```
function foo(s)
    return seq_byte(s, 1) + 0
end
```

Sobrecarga

- A operação de concatenação em SmallLua é *sobrecarregada*: ela funciona tanto para strings quanto para sequências
- Podemos fazer a verificação de tipos da concatenação tentando primeiro unificar seus argumentos com strings, para depois unificar com tipos sequência
- Mas isso nos impede de escrever funções genéricas usando concatenação:

```
function concat(s)
  if #s == 1 then
    return seq_byte(s, 1)
  else
    return seq_byte(s, 1) ..
           concat(seq_sub(s, 2, #s))
  end
end
```

Tags

- A ideia para permitir funções genéricas sobrecarregadas é ter *tags* associadas a variáveis de tipo em aberto e a parâmetros de tipo
- Essas tags limitam a variável ou parâmetro a tipos que suportam dado conjunto de operações sobrecarregadas
- Em SmallLua vamos usar duas tags: *eq* para tipos que podem ser comparados com `==` e *concat* para tipos que podem ser concatenados
- Quando unificamos duas variáveis em aberto unimos seus conjuntos de tags, e quando unificamos uma variável em aberto com outro tipo qualquer verificamos se esse tipo aceita as tags da variável