

Semântica de Linguagens de Programação

Fabio Mascarenhas - 2011.2

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

Recursão - fix em FAE

- Podemos definir `fix` como função, ao invés de primitiva

Recursão - fix em FAE

- Podemos definir `fix` como função, ao invés de primitiva

```
{fun {f}
  {with
    {frec {fun {frec}
            {f {fun {arg}
                {{frec frec} arg}}}}}
    {frec frec}}}
```

Recursão - fix em FAE

- A função `sum` definida usando `fix` (cores indicam escopo)

```
{with {sum {fix {fun {sum}
                {fun {x}
                  {if0 x
                    0
                    {+ {sum {- x 1}}
                      x}}}}}}}}
  {sum 5}}
```

Recursão - fix em FAE

- Podemos escrever `sum` sem `fix` facilmente, se chamarmos ela de um jeito diferente

```
{with {sum
      {with {sum
            {fun {sum}
              {fun {x}
                {if0 x
                  0
                  {+ {{sum sum} {- x 1}}
                    x}}}}}} {sum sum}}
  {sum 5}}
```

Recursão - fix em FAE

- Podemos extrair a parte que faz sum de dentro da função recursiva, tendo, e passar a função recursiva {sum sum} como parâmetro pra ela, tomando cuidado com um loop infinito

```
{with {sum
      {{fun {f}
        {with {sum
              {fun {sum}
                {f {fun {x}
                  {{sum sum} x}}}}}}
              {sum sum}}}}
      {fun {sum}
        {fun {x}
          {if0 x 0 {+ {sum {- x 1}} x}}}}}}
      {sum 5}}
```

Recursão - fix em FAE

- Vamos chamar a `{fun {f} ...}` do slide anterior de `fix` e jogá-la pra fora do primeiro `with`

```
{with {fix {fun {f}
      {with {sum
            {fun {sum}
              {f {fun {x}
                  {{sum sum} x}}}}}
            {sum sum}}}}}
  {with {sum {fix
        {fun {sum}
          {fun {x}
            {if0 x 0 {+ {sum {- x 1}}
                        x}}}}}
      {sum 5}}}}
```

Recursão - fix em FAE

- Os nomes dentro de fix não importam, vamos renomeá-los para deixar ela mais “genérica”

```
{with {fix {fun {f}
      {with {x
            {fun {x}
              {f {fun {y}
                 {{x x} y}}}}}}
      {x x}}}}
{with {sum {fix
      {fun {sum}
        {fun {x}
          {if0 x 0 {+ {sum {- x 1}}
                     x}}}}}}
      {sum 5}}}}
```

Recursão - fix em FAE

- Por último vamos “desenrolar” o `with` dentro de `fix` usando substituição

```
{with {fix {fun {f}
      {{fun {x} {f {fun {y}
                  {{x x} y}}}}}
      {fun {x} {f {fun {y}
                  {{x x} y}}}}}}}

{with {sum {fix
      {fun {sum}
      {fun {x}
      {if0 x 0 {+ {sum {- x 1}}
                  x}}}}}}}

{sum 5}}}
```

http://en.wikipedia.org/wiki/Fixed_point_combinator

Incrementando FAE

- Funções com múltiplos parâmetros - açúcar sintático
- Mais operações numéricas - * e /, comparação
- Booleanos e i f
- Listas e programação funcional

IMP - Linguagem Imperativa

- Divisão entre *expressões* e *comandos*
- De início, vamos usar AE + identificadores (sem `with`!) como expressões
- Comandos são `if`, `while`, `set`, `print`, `skip`, e sequência
- Sem escopo, todos os identificadores são “globais”
- Mas como `set` pode funcionar?

IMP

```
IAE ::= <num> | <id>
      | {+ <AE> <AE>}
      | {- <AE> <AE>}
```

```
IMP ::= {if <IAE> <IMP> <IMP>}
      | {while <IAE> <IMP>}
      | {set <id> <IAE>}
      | {print <IAE>}
      | {skip}
      | {<IMP>...}
```