Sistemas puro e aplicado

- No Cálculo Lambda *“puro”* não existe representação para números inteiros, operações aritméticas, valores ou operadores lógicos, entre outros tipos de dados e operações usualmente encontradas em linguagens de programação de alto-nível;
- No Cálculo Lambda *“aplicado”* admite-se o uso explícito dos mesmos:

$$\lambda x.x + 1$$

$$(\lambda x.x + 1)(3) \rhd_{1\beta} [3/x](x + 1) \equiv 3 + 1 \equiv 4$$

- É possível, no entanto, representar tipos de dados e operadores quaisquer usando o sistema puro, como demonstram os casos apresentados a seguir.

Numerais de Church

Definição

Para todo $n \in \mathbb{N}$, o "Numeral de Church" de n, denotado $\overline{n}$, é um termo-λ que representa n:

$$\overline{n} := \lambda xy.x^n y$$

onde:

$$x^n y$$

é definido como:

$$\begin{cases} x^n y \equiv \underbrace{x(x(...(x}_{n \text{ vezes}}\, y)...)) \text{ se } n \geq 1 \\ x^0 y \equiv y \end{cases}$$

Numerais de Church

Exemplos

$$
\begin{aligned}
\overline{0} &:= \lambda xy.y \\
\overline{1} &:= \lambda xy.xy \\
\overline{2} &:= \lambda xy.x(xy) \\
\overline{3} &:= \lambda xy.x(x(xy)) \\
\overline{4} &:= \lambda xy.x(x(x(xy))) \\
&\ldots
\end{aligned}
$$

Numerais de Church

Propriedade

Os Numerais de Church tem a propriedade de que, para quaisquer termos F e X,

$$\overline{n} F X \rhd_\beta F^n X.$$

Em outras palavras, o numeral de Church inserido na frente de uma aplicação de uma função ao seu argumento representa a aplicação repetida dessa função o mesmo número de vezes.

Numerais de Church

Propriedade

Exemplo:

$$
\begin{array}{rcl}
\overline{2}FX & \equiv & (\lambda xy.x(xy))FX \\
& \equiv & ((\lambda xy.x(xy))F)X \\
& \rhd_{1\beta} & [F/x](\lambda y.x(xy))X \\
& \equiv & (\lambda y.F(Fy))X \\
& \rhd_{1\beta} & [X/y]F(Fy) \\
& \equiv & F(FX) \\
& \equiv & F^2X
\end{array}
$$

Numerais de Church

Sucessor

O sucessor de um Numeral de Church pode ser obtido pela aplicação da expressão:

$$\overline{succ} := \lambda uxy.x(uxy)$$

ao respectivo numeral. É fácil provar que:

$$\overline{succ}\ \overline{n} \rhd_\beta \overline{n+1}.$$

De fato, basta obervar que:

$$(\lambda uxy.x(uxy))\overline{n} \rhd_\beta \lambda xy.x(\overline{n}xy) \equiv \lambda x.\lambda y.xx^n y \equiv \lambda x.\lambda y.x^{n+1}y \equiv \overline{n+1}.$$

Numerais de Church

Sucessor

Exemplo:

$$
\begin{array}{rcl}
\overline{succ}\ \overline{0} & \equiv & (\lambda uxy.x(uxy))(\lambda xy.y) \\
 & \rhd_{1\beta} & [(\lambda xy.y)/u](\lambda xy.x(uxy)) \\
 & \equiv & (\lambda xy.x((\lambda xy.y)xy) \\
 & \rhd_{\beta} & (\lambda xy.xy) \\
 & \equiv & \overline{1}
\end{array}
$$

Numerais de Church

Adição

A adição de dois Numerais de Church pode ser obtida pela aplicação da expressão:

$$\overline{add} := \lambda uvxy.ux(vxy)$$

aos respectivos operandos. Nesse caso, temos que:

$$\overline{add}\;\overline{m}\;\overline{n} \rhd_\beta \overline{m+n}.$$

De fato, basta observar que:

$$(\lambda uvxy.ux(vxy))\overline{m}\;\overline{n} \rhd_\beta \lambda xy.\overline{m}x(\overline{n}xy) \equiv \lambda xy.\overline{m}x(x^n y)$$

$$\equiv \lambda xy.x^m(x^n y) \equiv \lambda xy.x^{m+n}y \equiv \overline{m+n}$$

Numerais de Church

Adição

Exemplo:

$$
\begin{array}{rcl}
\overline{add}\ \overline{1}\ \overline{2} & \equiv & (\lambda uvxy.ux(vxy))(\lambda xy.xy)(\lambda xy.x(xy)) \\
& \rhd_{1\beta} & ([(\lambda xy.xy)/u](\lambda vxy.ux(vxy)))(\lambda xy.x(xy)) \\
& \equiv & (\lambda vxy.(\lambda xy.xy)x(vxy))(\lambda xy.x(xy)) \\
& \rhd_{\beta} & (\lambda vxy.x(vxy))(\lambda xy.x(xy)) \\
& \rhd_{1\beta} & [(\lambda xy.x(xy))/v](\lambda xy.x(vxy)) \\
& \equiv & (\lambda xy.x((\lambda xy.x(xy))xy)) \\
& \rhd_{\beta} & (\lambda xy.x(x(xy))) \\
& \equiv \overline{3} &
\end{array}
$$

Numerais de Church

Multiplicação

A multiplicação de dois Numerais de Church pode ser obtida pela aplicação da expressão:

$$\overline{mult} := \lambda uvx.u(vx)$$

aos respectivos operandos. Nesse caso, temos que:

$$\overline{mult}\ \overline{m}\ \overline{n} \rhd_{\beta} \overline{m * n}$$

.

Numerais de Church

Multiplicação

De fato, temos que:

$$
\begin{array}{rcl}
(\lambda uvx.u(vx))\ \overline{m}\ \overline{n} & \rhd_\beta & \lambda x.\overline{m}(\overline{n}x) \\
& \equiv & \lambda x.\overline{m}((\lambda y.\lambda z.y^n z)x) \\
& \rhd_\beta & \lambda x.\overline{m}(\lambda z.x^n z) \\
& \equiv & \lambda x.(\lambda u.\lambda v.u^m v)(\lambda z.x^n z) \\
& \rhd_\beta & \lambda x.[\lambda z.x^n z/u](\lambda v.u^m v) \\
& \equiv & \lambda x.\lambda v.(\lambda z.x^n z)^m v \\
& \equiv & \lambda x.\lambda v.(\lambda z.x^n z)^{m-1}((\lambda z.x^n z)v)
\end{array}
$$

Numerais de Church

Multiplicação

Continuação:

$$
\begin{array}{rcl}
\lambda x.\lambda v.(\lambda z.x^n z)^{m-1}((\lambda z.x^n z)v) & \triangleright_\beta & \lambda x.\lambda v.(\lambda z.x^n z)^{m-1}(x^n v) \\
& \equiv & \lambda x.\lambda v.(\lambda z.x^n z)^{m-2}((\lambda z.x^n z)x^n v) \\
& \triangleright_\beta & \lambda x.\lambda v.(\lambda z.x^n z)^{m-2}(x^n(x^n v)) \\
& \equiv & \lambda x.\lambda v.(\lambda z.x^n z)^{m-2}(x^{2*n} v)) \\
& \triangleright_\beta & \lambda x.\lambda v.x^{m*n} v \\
& \equiv & \overline{m*n}
\end{array}
$$

Numerais de Church

Multiplicação

Exemplo:

$$
\begin{array}{rcl}
\overline{mult}\ \overline{2}\ \overline{2} & \equiv & (\lambda uvx.u(vx))\overline{2}\ \overline{2} \\
& \rhd_{1\beta} & ([\overline{2}/u](\lambda vx.u(vx)))\overline{2} \\
& \equiv & (\lambda vx.\overline{2}(vx))\overline{2} \\
& \rhd_{1\beta} & [\overline{2}/v](\lambda x.\overline{2}(vx)) \\
& \equiv & \lambda x.\overline{2}(\overline{2}x) \\
& \rhd_{1\beta} & \lambda x.\overline{2}(\lambda y.x(xy)) \\
& \rhd_{1\beta} & \lambda x.\lambda y.(\lambda y.x(xy))((\lambda y.x(xy))y \\
& \rhd_{1\beta} & \lambda x.\lambda y.(\lambda y.x(xy))(x(xy)) \\
& \rhd_{1\beta} & \lambda x.\lambda y.x(x(x(xy))) \\
& \equiv \overline{4} &
\end{array}
$$

Numerais de Church

Exponenciação

A exponenciação de dois Numerais de Church pode ser obtida pela aplicação da expressão:

$$\overline{exp} := \lambda uv.vu$$

aos respectivos operandos. Nesse caso, temos que:

$$\overline{exp}\ \overline{m}\ \overline{n} \rhd_\beta \overline{m^n}.$$

Numerais de Church

Exponenciação

De fato, temos que:

$$
\begin{array}{rcl}
(\lambda uv.vu)\ \overline{m}\ \overline{n} & \triangleright_\beta & \overline{n}\ \overline{m} \\
& \equiv & (\lambda x.\lambda y.x^n y)\ \overline{m} \\
& \triangleright_\beta & \lambda y.(\overline{m})^n y \\
& \equiv & \lambda y.(\overline{m}(\overline{m}(...(\overline{m}(\overline{m}y))))) \\
& \triangleright_\beta & \lambda y.(\overline{m}(\overline{m}(...(\overline{m}(\lambda w.y^m w))))) \\
& \triangleright_\beta & \lambda y.(\overline{m}(\overline{m}(...(\lambda w.y^{m^2} w)))) \\
& \triangleright_\beta & \lambda y.(\lambda w.y^{m^n} w) \\
& \equiv & \overline{m^n}
\end{array}
$$

Numerais de Church

Exponenciação

Exemplo:

$$
\begin{array}{rcl}
\overline{exp}\ \overline{2}\ \overline{2} & \equiv & (\lambda u.\lambda v.vu)\overline{2}\ \overline{2} \\
& \rhd_\beta & (\lambda v.v\overline{2})\overline{2} \\
& \rhd_\beta & \overline{2}\ \overline{2} \\
& \equiv & (\lambda x.\lambda y.x(xy))\overline{2} \\
& \rhd_\beta & \lambda y.\overline{2}(\overline{2}y) \\
& \equiv & \lambda y.\overline{2}((\lambda x.\lambda z.x.(xz))y) \\
& \rhd_\beta & \lambda y.\overline{2}(\lambda z.y(yz)) \\
& \equiv & \lambda y.(\lambda x.\lambda w.x(xw))(\lambda z.y(yz))
\end{array}
$$

Numerais de Church

Exponenciação

Exemplo (continuação):

$$
\begin{array}{lcl}
\lambda y.(\lambda x.\lambda w.x(xw))(\lambda z.y(yz)) & \triangleright_\beta & \lambda y.[\lambda z.y(yz)/x](\lambda w.x(xw)) \\
 & \equiv & \lambda y.\lambda w.[\lambda z.y(yz)/x](x(xw)) \\
 & \equiv & \lambda y.\lambda w.(\lambda z.y(yz))((\lambda z.y(yz))w) \\
 & \triangleright_\beta & \lambda y.\lambda w.(\lambda z.y(yz))(y(yw)) \\
 & \triangleright_\beta & \lambda y.\lambda w.[y(yw)/z](y(yz)) \\
 & \equiv & \lambda y.\lambda w.(y(y(y(yw)))) \\
 & \equiv & \overline{4}
\end{array}
$$

Numerais de Church

Outras operações

- Predecessor ($n - 1$ se $n > 0$ ou 0 caso contrário):

$$\overline{pred} := \lambda n.\lambda f.\lambda x.n(\lambda g.\lambda h.h(gf))(\lambda u.x)(\lambda u.u)$$

- Subtração ($m - n$ se $m \geq n$ ou 0 caso contrário):

$$\overline{sub} := \lambda m.\lambda n.(n\ \overline{pred})m$$

Numerais de Church

Expressões compostas

Através da combinação das expressões lambda anteriores, é possível representar expressões aritméticas mais complexas, como é o caso de:

$$(2^3 + 4) * 5$$

que é denotada:

$$\overline{mult} \ (\ \overline{add} \ (\ \overline{exp} \ \overline{2} \ \overline{3} \ ) \ \overline{4} \ ) \ \overline{5} \equiv$$

$$\underbrace{\underbrace{(\lambda uvx.u(vx))}_{\overline{mult}} \ \underbrace{(\underbrace{(\lambda uvxy.ux(vxy))}_{\overline{add}} \ \underbrace{(\underbrace{(\lambda uv.vu)}_{\overline{exp}} \ \overline{2} \ \overline{3})}_{2^3} \ \overline{4})}_{2^3+4} \ \overline{5}}_{(2^3+4)*5} \triangleright_\beta \overline{60}.$$

Numerais de Church

Exercícios

Determinar:

- $\overline{succ}\ \overline{1}$;
- $\overline{add}\ \overline{2}\ \overline{3}$;
- $\overline{mult}\ \overline{0}\ \overline{3}$;
- $\overline{exp}\ \overline{3}\ \overline{2}$;
- $\overline{pred}\ \overline{2}$;
- $\overline{sub}\ \overline{3}\ \overline{1}$;
- $\overline{add}\ (\overline{add}\ \overline{1}\ \overline{2})(\overline{mult}\ \overline{2}\ \overline{3})$.

Booleanos de Church

Definição

O Cálculo Lambda puro também permite a representação de valores e operações lógicas:

- $\overline{true} := \lambda x.\lambda y.x$
  (projeção do primeiro argumento);
- $\overline{false} := \lambda x.\lambda y.y$
  (projeção do segundo argumento);
  Observar que $\overline{false} \equiv \overline{0}$.

Booleanos de Church

AND

$$\overline{and} := \lambda x.\lambda y.xyx$$

É possível provar que:

$$\overline{and}\ \overline{m}\ \overline{n} \rhd_\beta \overline{m\ and\ n}$$

De fato:

$$(\lambda x.\lambda y.xyx)\ \overline{m}\ \overline{n} \rhd_\beta \overline{m}\ \overline{n}\ \overline{m}$$

- Se $m = TRUE$, então projeta como resultado o valor de n;
- Se $m = FALSE$, projeta como resultado o próprio valor de m.

Booleanos de Church

AND

Exemplos:

$$
\begin{aligned}
(\lambda x.\lambda y.xyx)\ \overline{true}\ \overline{true} &\triangleright_\beta \overline{true}\ \underbrace{\overline{true}}\ \overline{true} \\
&\triangleright_\beta \overline{true}
\end{aligned}
$$

$$
\begin{aligned}
(\lambda x.\lambda y.xyx)\ \overline{false}\ \overline{true} &\triangleright_\beta \overline{false}\ \overline{true}\ \underbrace{\overline{false}} \\
&\triangleright_\beta \overline{false}
\end{aligned}
$$

Booleanos de Church
OR

$$\overline{or} := \lambda x.\lambda y.xxy$$

É possível provar que:

$$\overline{or}\ \overline{m}\ \overline{n} \rhd_\beta \overline{m\ or\ n}$$

De fato:

$$(\lambda x.\lambda y.xxy)\ \overline{m}\ \overline{n} \rhd_\beta \overline{m}\ \overline{m}\ \overline{n}$$

- Se $m = TRUE$, então projeta como resultado o próprio valor de m;
- Se $m = FALSE$, projeta como resultado o valor de n.

Booleanos de Church

OR

Exemplos:

$$
\begin{aligned}
(\lambda x.\lambda y.xxy)\ \overline{true}\ \overline{false} &\rhd_\beta \overline{true}\ \underbrace{\overline{true}}\ \overline{false} \\
&\rhd_\beta \overline{true}
\end{aligned}
$$

$$
\begin{aligned}
(\lambda x.\lambda y.xxy)\ \overline{false}\ \overline{true} &\rhd_\beta \overline{false}\ \overline{false}\ \underbrace{\overline{true}} \\
&\rhd_\beta \overline{true}
\end{aligned}
$$

Booleanos de Church

NOT

$$\overline{not} := \lambda x.\lambda y.\lambda z.xzy$$

É possível provar que:

$$\overline{not}\ \overline{m} \rhd_\beta \overline{not(m)}$$

De fato:

$$(\lambda x.\lambda y.\lambda z.xzy)\ \overline{m} \rhd_\beta \lambda y.\lambda z.\overline{m}zy$$

- Se $m = TRUE$, então $\lambda y.\lambda z.\overline{m}zy \rhd_\beta \lambda y.\lambda z.z \equiv \overline{false}$;
- Se $m = FALSE$, então $\lambda y.\lambda z.\overline{m}zy \rhd_\beta \lambda y.\lambda z.y \equiv \overline{true}$.

Booleanos de Church

NOT

Exemplos:

$$\begin{array}{rcl} (\lambda x.\lambda y.\lambda z.xzy)\ \overline{true} & \triangleright_\beta & \lambda y.\lambda z.(\overline{true})zy \\ & \triangleright_\beta & \lambda y.\lambda z.z \\ & \equiv & \overline{false} \end{array}$$

$$\begin{array}{rcl} (\lambda x.\lambda y.\lambda z.xzy)\ \overline{false} & \triangleright_\beta & \lambda y.\lambda z.(\overline{false})zy \\ & \triangleright_\beta & \lambda y.\lambda z.y \\ & \equiv & \overline{true} \end{array}$$

Booleanos de Church

XOR

$$\overline{xor} := \lambda x.\lambda y.\lambda z.\lambda w.x(ywz)(yzw)$$

É possível provar que:

$$\overline{xor}\ \overline{m}\ \overline{n} \triangleright_\beta \overline{m\ xor\ n}$$

Booleanos de Church

XOR

De fato:

$$(\lambda x.\lambda y.\lambda z.\lambda w.x(ywz)(yzw))\ \overline{m}\ \overline{n} \rhd_\beta \lambda z.\lambda w.\overline{m}(\overline{n}wz)(\overline{n}zw)$$

- Se $m = TRUE$, então $\lambda z.\lambda w.\overline{m}(\overline{n}wz)(\overline{n}zw) \rhd_\beta \lambda z.\lambda w.\overline{n}wz$;
  - Se $n = TRUE$, então $\lambda z.\lambda w.\overline{n}wz \rhd_\beta \lambda z.\lambda w.w \equiv \overline{false}$;
  - Se $n = FALSE$, então $\lambda z.\lambda w.\overline{n}wz \rhd_\beta \lambda z.\lambda w.z \equiv \overline{true}$.
- Se $m = FALSE$, então $\lambda z.\lambda w.\overline{m}(\overline{n}wz)(\overline{n}zw) \rhd_\beta \lambda z.\lambda w.\overline{n}zw$;
  - Se $n = TRUE$, então $\lambda z.\lambda w.\overline{n}zw \rhd_\beta \lambda z.\lambda w.z \equiv \overline{true}$;
  - Se $n = FALSE$, então $\lambda z.\lambda w.\overline{n}zw \rhd_\beta \lambda z.\lambda w.w \equiv \overline{false}$.

Booleanos de Church

XOR

Exemplos:

$$
\begin{array}{rcl}
(\lambda x.\lambda y.\lambda z.\lambda w.x(ywz)(yzw))\ \overline{true}\ \overline{false} & \rhd_\beta & \\
\lambda z.\lambda w.\overline{true}((\overline{false})wz)((\overline{false}))zw & \rhd_\beta & \\
\lambda z.\lambda w.(\overline{false})wz & \rhd_\beta & \\
\lambda z.\lambda w.z & \equiv & \overline{true}
\end{array}
$$

$$
\begin{array}{rcl}
(\lambda x.\lambda y.\lambda z.\lambda w.x(ywz)(yzw))\ \overline{false}\ \overline{false} & \rhd_\beta & \\
\lambda z.\lambda w.\overline{false}((\overline{false})wz)((\overline{false}))zw & \rhd_\beta & \\
\lambda z.\lambda w.(\overline{false})zw & \rhd_\beta & \\
\lambda z.\lambda w.w & \equiv & \overline{false}
\end{array}
$$

Booleanos de Church

IF

$$\overline{if} := \lambda x.\lambda y.\lambda z.xyz$$

É possível provar que:

$$\overline{if}\ \overline{e}\ \overline{m}\ \overline{n} \rhd_\beta \overline{m} \text{ se } e = TRUE$$

$$\overline{if}\ \overline{e}\ \overline{m}\ \overline{n} \rhd_\beta \overline{n} \text{ se } e = FALSE$$

De fato:

$$(\lambda x.\lambda y.\lambda z.xyz)\ \overline{e}\ \overline{m}\ \overline{n} \rhd_\beta \overline{e}\ \overline{m}\ \overline{n}$$

- Se $e = TRUE$, então projeta m como resultado;
- Se $e = FALSE$, projeta n como resultado.

Booleanos de Church
IF

Exemplos:

$$\begin{aligned}(\lambda x.\lambda y.\lambda z.xyz)\ \overline{true}\ \overline{m}\ \overline{n} &\rhd_\beta \quad \overline{true}\ \overline{m}\ \overline{n}\\ &\rhd_\beta \quad \overline{m}\end{aligned}$$

$$\begin{aligned}(\lambda x.\lambda y.\lambda z.xyz)\ \overline{false}\ \overline{m}\ \overline{n} &\rhd_\beta \quad \overline{false}\ \overline{m}\ \overline{n}\\ &\rhd_\beta \quad \overline{n}\end{aligned}$$

Booleanos de Church

Exercícios

Avaliar:

- $\overline{and}\ \overline{true}\ \overline{false}$
- $\overline{or}\ \overline{false}\ \overline{false}$
- $\overline{xor}\ \overline{false}\ \overline{true}$
- $\overline{or}\ (\ \overline{and}\ \overline{true}\ \overline{false}\ )\ \overline{true}$;
- $\overline{not}\ (\ \overline{xor}\ \overline{true}\ (\ \overline{not}\ \overline{true}\ ))$.

Construir expressões lambda para os operadores:

- Implicação ($\Rightarrow$) (dica: usar $\overline{not}$ e $\overline{or}$);
- Bi-implicação ($\Leftrightarrow$) (dica: usar $\overline{and}$).

Booleanos de Church

Expressões compostas - ZERO

Através da combinação das expressões anteriores, é possível representar funções mais complexas, que fazem uso de valores e operadores lógicos e aritméticos simultâneamente, como é o caso da função que testa se o argumento é zero e retorna $\overline{true}$ ou $\overline{false}$:

$$\overline{zero} := \lambda x.x(\lambda y.\overline{false})\ \overline{true}.$$

Booleanos de Church

Expressões compostas - ZERO

De fato, para $n = 0$:

$$\begin{array}{rl} (\lambda x.x(\lambda y.\overline{false})\ \overline{true})\ \overline{0} & \rhd_\beta \\ \overline{0}\ (\lambda y.\overline{false})\ \overline{true} & \equiv \\ \overline{false}\ (\lambda y.\overline{false})\ \overline{true} & \rhd_\beta \\ \overline{true} & \end{array}$$

Booleanos de Church

Expressões compostas - ZERO

E para $n > 0$:

$$
\begin{aligned}
(\lambda x.x(\lambda y.\overline{false})\ \overline{true})\ \overline{n} & \quad \triangleright_\beta \\
\overline{n}\ (\lambda y.\overline{false})\ \overline{true} & \quad \equiv \\
(\lambda z.\lambda w.z^n w)(\lambda y.\overline{false})\ \overline{true} & \quad \triangleright_\beta \\
(\lambda w.(\lambda y.\overline{false})^n w)\ \overline{true} & \quad \triangleright_\beta \\
(\lambda w.(\lambda y.\overline{false})^{n-1}((\lambda y.\overline{false})w))\ \overline{true} & \quad \triangleright_\beta \\
(\lambda w.(\lambda y.\overline{false})^{n-1}\ \overline{false})\ \overline{true} & \quad \triangleright_\beta \\
& \quad \ldots \\
(\lambda w.(\lambda y.\overline{false})\ \overline{false})\ \overline{true} & \quad \triangleright_\beta \\
(\lambda w.\overline{false})\ \overline{true} & \quad \triangleright_\beta \\
\overline{false} &
\end{aligned}
$$

Booleanos de Church

Expressões compostas - LEQ

Função que testa se o primeiro argumento é menor ou igual que o segundo:

$$\overline{leq} := \lambda x.\lambda y.\overline{zero}\ (\overline{sub}\ \overline{x}\ \overline{y}).$$

De fato:

$$\begin{aligned}(\lambda x.\lambda y.\overline{zero}\ (\overline{sub}\ \overline{x}\ \overline{y}))\ \overline{m}\ \overline{n} &\quad \triangleright_\beta \\ \overline{zero}\ (\overline{sub}\ \overline{m}\ \overline{n}) &\quad \triangleright_\beta \\ \overline{zero}\ (\overline{m-n}) &\end{aligned}$$

Booleanos de Church

Exercícios

Desenvolver expressões lambda para os seguintes operadores relacionais:

- $\overline{gt}$ (maior que);
- $\overline{equal}$ (igualdade);
- $\overline{notequal}$ (desigualdade).