ADT: WARM UP

RAUL H.C.LOPES

1. Introdução

Esta lista[1] contém exercícios de construção e correção de algoritmos recursivos e propriedades de seqüências e árvores. Uma prova só será considerada válida se cada um de seus passos (repito, cada um de seus passos) for claramente justificado por:

- um axioma ou regra de inferência da lógica;
- um teorema ou lema previamente provado e apresentado nesta prova.

Cada passo de uma prova deve ser devidamente anotado com o axioma, regra de inferência ou teorema que o justifica.

Todas as suas definições e provas serão, em princípio, apresentadas em lógica de primeira ordem, usando *equational reasoning*, como feito em sala de aula. Você é livre para usar outra lógica e/ou formalismo de prova desde que devidamente justificado: ou seja, anuncie claramente o formalismo que usará e identifique explicitamente cada regra de inferência usada nas provas. Por exemplo, definições em Haskell, deverão ser acompanhadas por axiomas definindo a semântica dos operadores da linguagem usados, o que provavelmente você não conhece.

A notação de pontos de Curry, apresentada em sala e notas de aula e na página 34 de [1], é utilizada para reduzir o uso de parênteses.

2. Quantificadores

Quantificadores representam papel importante em propriedade e provas sobre algoritmos e estruras de dados. Estas notas introduzem axiomas fundamentais para trabalhar com quantificadores, mas não substituem fontes clássicas como [3] e [2].

Nesta seção, assuma que:

- $R.x, Q.X, P.x$ são funções booleanas de x (onde x ocorre livre);
- $f.x$ é uma função aritmética de x.
- $S.x, T.x$ são uma funções quaisquer de x.

Os quantificadores são apresentados neste curso com a seguinte notação:

[1] Veja seção 5 sobre alterações desta lista feitas desde a última versão.

Notação 1. $(\star x : R.x : P.x)$
onde:

- *x é o parâmetro sobre o qual a sentença é quantificada (*bound variable*);*
- *$R.x$ é o universo da quantificação;*
- *$P.x$ é o corpo da quantificação.*

Alguns exemplos de quantificadores comumente usados e sua notação usual em textos de matemática:

$$(\wedge x : R.x : P.x) = (\forall x : R.x : P.x)$$
$$(\vee x : R.x : P.x) = (\exists x : R.x : P.x)$$
$$(+x : R.x : f.x) = (\sum x : R.x : f.x) = \sum_{R.x} f.x$$
$$(.x : R.x : f.x) = (\prod x : R.x : f.x) = \prod_{R.x} f.x$$

Os seguintes operadores definem máximo e mínimo de inteiros.

$$a \leq b \Rightarrow .\ a \downarrow b = a \tag{1}$$
$$a \geq b \Rightarrow .\ a \downarrow b = b \tag{2}$$
$$a \leq b \Rightarrow .\ a \uparrow b = b \tag{3}$$
$$a \geq b \Rightarrow .\ a \uparrow b = a \tag{4}$$

Suas respectivas identidades são:

$$a \downarrow \infty = a = \infty \downarrow a \tag{5}$$
$$a \uparrow -\infty = a = -\infty \uparrow a \tag{6}$$

Eles também podem ser usados como quantificadores. Por exemplo em:

$$(\downarrow i : i \in \mathbb{N} : 10 \leq i < 100) f.i$$

significando o mínimo dos $f.i$ com i inteiro maior ou igual que 10 e menor do que 100.

Uma variável x tem ocorrência livre em uma expressão P se tem uma ocorrência que não é parâmetro de (amarrada por) nenhum quantificador. A notação $P[x := E]$ será usada para indicar o resultado de substituir todas as ocorrências livres de x (em P) por E.

Assuma que $\star$ seja um operador binário com as seguintes propriedade:

Simetria: $b \star c = c \star b$

Associatividade: $a \star (b \star c) = (a \star b) \star c$

Identidade $u_\star$: $u_\star \star b = b = b \star u_\star$

Os seguintes axiomas permitem trabalhar com quantificadores:

(7) $(\star x : false : T.x) = u_\star$

(8) $(\star x : x = E : T.x) = T[x := E]$

(9) $(\star x : R.x : S.x) \star (\star x : R.x : T.x) = (\star x : R.x : S.x \star T.x)$

(10) $(\star x : P.x \vee R.x : T.x) \star (\star x : P.x \wedge R.x : T.x) \quad =$
$(\star x : P.x : T.x) \star (\star x : R.x : T.x)$

(11) $(\star x : P.x \vee R.x : T.x) = (\star x : P.x : T.x) \star (\star x : R.x : T.x)$

Exercício 1. *Prove:*

(1) $(\star i \in \mathbb{N} \ : \ 0 \leq i < n + 1 \ : \ T.i) \ = \ (\star i \in \mathbb{N} \ : \ 0 \leq i < n \ : \ T.i) \star T[i := n]$

Solução.

$$
\begin{array}{ll}
 & (\star i \in \mathbb{N} : 0 \leq i < n + 1 : T.i) \\
= & \langle \textit{arimética: } 0 \leq i < n + 1 = 0 \leq i \leq n \rangle \\
 & (\star i \in \mathbb{N} : 0 \leq i \leq n : T.i) \\
= & \langle \textit{aritmética: } 0 \leq i \leq n = (0 \leq i < n \vee i = n) \rangle \\
 & (\star i \in \mathbb{N} : 0 \leq i < n \vee i = n : T.i) \\
= & \langle \textit{axioma } 11 \rangle \\
 & (\star i \in \mathbb{N} : 0 \leq i < n : T.i) \star (\star i \in \mathbb{N} : i = n : T.i) \\
= & \langle \textit{axioma } 8 \rangle \\
 & (\star i \in \mathbb{N} : 0 \leq i < n : T.i) \star T[i := n]
\end{array}
$$

□

(2) $(\star i \in \mathbb{N} : 0 \leq i < n + 1 : T.i) = T[i := 0] \star (\star i \in \mathbb{N} : 0 < i < n + 1 : T.i)$

Os seguintes axiomas valem para os quantificadores lógicos $(\forall, \exists)$.

(12) $(\forall x : R.x : P.x) = \forall x : R.x \Rightarrow P.x$

(13) $(\exists x : R.x : P.x) = \exists x : R.x \wedge P.x$

(14) $(\exists x : R.x : P.x) = \neg(\forall x : R.x : \neg P.x)$

(15) $(\forall x : R.x : P.x) \vdash R[x := E] \Rightarrow P[x := E]$

A seguinte regra de inferência é usada para provar sentenças universalmente quantificadas, assumindo que $\hat{x}$ é parâmetro que não ocorre livre em Γ ou P.

$$\frac{\Gamma \vdash P[x := \hat{x}]}{\Gamma \vdash \forall x : P.x} \tag{16}$$

3. Seqüências

Uma seqüência de elementos de um tipo A é uma seqüência vazia (denotada por $\perp$) ou resultado de adicionar um elemento de A a uma seqüência de elementos de A. Os axiomas a seguir definem seqüências de elementos de A, denotada $seq.A$:

$$\perp \in seq.A \tag{17}$$

$$(a \lhd x) \in seq.A \Leftarrow x \in seq.A \wedge a \in A \tag{18}$$

(19) fecho universal.

Assumindo que $x, y \in seq.A$ e que $a, b \in A$, os seguintes axiomas tratam a igualdade sobre seqüências.

$$\perp \neq (a \lhd x) \tag{20}$$

$$(a \lhd x) = (b \lhd y) \ .=. \ a = b \wedge x = y \tag{21}$$

O princípio da indução sobre seqüências estabelece condições para que uma propriedade P seja válida para qualquer seqüência, sendo denotado $WPI(P)$.

$$WPS(P) \stackrel{\triangle}{=} ((\forall x : x \in seq.A : (\forall a : a \in A : P.x) \Rightarrow P.a \lhd x)) \tag{22}$$

$$WPI(P) \stackrel{\triangle}{=} (P.\perp \wedge WPS(P)) \Rightarrow ((\forall x : x \in seq.A : P.x)) \tag{23}$$

O axioma 25, a seguir, define o princípio de indução sobre inteiros não negativos, onde $P.i$ é uma propriedade qualquer parametrizada em relação a i.

$$SPS(P) \stackrel{\triangle}{=} (\forall n : n \in \mathbb{N} : (\forall i \in \mathbb{N} : 0 \leq i < n : P.i) \Rightarrow P.n) \tag{24}$$

$$SPI(P) \stackrel{\triangle}{=} SPS(P) \Rightarrow (\forall n : n \in \mathbb{N} : P.n) \tag{25}$$

Exercício 2. *Prove que*

$$SPS(P) = (\forall n : n \in \mathbb{N} : P.n)$$

Note que o princípio de indução sobre seqüências pode ser obtido do princípio de indução sobre inteiros. Note também que o princípio 25 permite induzir um princípio de indução forte sobre seqüências.

As equações a seguir definem o operador *snoc* (denotado $\triangleright$, em oposição, logicamente ao operador *cons*, $\triangleleft$).

$$\bot \triangleright a \mathrel{.=.} a \triangleleft \bot \tag{26}$$

$$(a \triangleleft x) \triangleright b \mathrel{.=.} a \triangleleft (x \triangleright b) \tag{27}$$

As equações a seguir definem um operador *cat* (denotado $\hat{\ }$) para concatenar duas seqüências.

$$\bot \hat{\ } y = y \tag{28}$$

$$(a \triangleleft x) \hat{\ } y = .a \triangleleft (x \hat{\ } y) \tag{29}$$

Pertinência em uma seqüência é definida a seguir.

$$a \notin \bot \tag{30}$$

$$a \in (b \triangleleft x) \mathrel{=.} a = b \vee a \in x \tag{31}$$

(32)

O número de elementos de uma seqüência é dado pelos axiomas a seguir.

$$\|\bot\| = 0 \tag{33}$$

$$\|a \triangleleft x\| = 1 + \|x\| \tag{34}$$

Uma seqüência é não decrescente quando a atende aos axiomas 35 e 36.

$$\bot . \nearrow \tag{35}$$

$$a \triangleleft x . \nearrow = x . \nearrow \wedge (\forall b : b \in x : a \leq b) \tag{36}$$

Exercício 3. *Dados os axiomas*

$$min \bot = +\infty \tag{37}$$

$$min . a \triangleleft x = a . \downarrow min\, x \tag{38}$$

prove que

(1) $(\forall x : x \in seq.A : min\, x = (\downarrow a : a \in x : a))$
(2) $(\forall x : x \in seq.A : (\forall a : a \in x : min\, x \leq a))$

De agora em diante, e assumindo os resultados do exercício 3, quando x for uma seqüência, valerá a seguinte notação:

$$\downarrow x = min\, x$$

Exercício 4. *Dados os axiomas*

(39) $$del\, a \perp = \perp$$

(40) $$a = b \Rightarrow .\ del\, a\, (b \lhd x) = x$$

(41) $$a \neq b \Rightarrow .\ del\, a\, (b \lhd x) = b \lhd (del\, a\, x)$$

Prove

(1)
$$(\forall a, b : a, b \in A \wedge b \in del\, a\, x : \downarrow x \leq b)$$

(2)
$$(\forall x \in seq.A : \|x\| > 0 : \|del\ \downarrow x\, x\| < \|x\|)$$

Solução.

A prova pode é facilmente construída por indução (axioma 23), sendo a propriedade trivialmente verificada para seqüências de tamanho zero e um. Para uma seqüência $b \lhd x$, prova-se, considerando casos $b = \downarrow(b \lhd x)$ e $b \neq \downarrow(b \lhd x)$:

$$\begin{array}{ll} & \|del\ \downarrow(b \lhd x)\, (b \lhd x)\| \\ = & \langle \textit{caso } b = \downarrow(b \lhd x) \textit{e axioma 40} \rangle \\ & \|x\| \\ < & \langle \textit{aritmética} \rangle \\ & 1 + \|x\| \\ = & \langle \textit{axioma 34} \rangle \\ & \|b \lhd x\| \\ = & \langle \textit{caso } b \neq \downarrow . b \lhd x \textit{ e axioma 41} \rangle \\ & \|b \lhd del\ \downarrow(b \lhd x) x\, \| \\ = & \langle \textit{axioma 34} \rangle \\ & 1 + \|del\ \downarrow x\, x\| \\ < & \langle \textit{Hipótese de indução} \rangle \\ & 1 + \|x\| \\ = & \langle \textit{axioma 34} \rangle \\ & \|b \lhd x\| \end{array}$$

□

Exercício 5. *Dados os axiomas*

(42) $$selSort \perp = \perp$$

(43) $$x \neq \perp \Rightarrow .\ selSort\, x = (\downarrow x) \lhd . selSort.del\, (\downarrow x)\, x$$

Prove que

$$(selSort\, x)\nearrow$$

Solução.

O teorema vale trivialmente quando $x = \bot$. *Usando indução forte, define-se:*

$$R.i \stackrel{\triangle}{=} (\forall x \in seq.A : \|x\| = i : (selSort\, x)\nearrow)$$

Usando o princípio de indução sobre $\mathbb{N}$, *axioma 25, assume-se*

$$HI \stackrel{\triangle}{=} (\forall i \in \mathbb{N} : i < n : R.i)$$

prova-se, a seguir, que $R.n$.

$$\begin{array}{ll}
& R.n \\
=\langle & \textit{assumiundo que } \|x\| = n\rangle \\
& (selSort\, x)\nearrow \\
=\langle & \textit{axioma 43}\rangle \\
& (\downarrow x \triangleleft selSort.del\ \downarrow x\, x)\nearrow \\
=\langle & \textit{axioma 36}\rangle \\
& (selSort.del\ \downarrow x\, x)\nearrow \wedge (\forall a : a \in (selSort.del\ \downarrow x\, x) : \downarrow x \leq a) \\
=\langle & \textit{HI e lema 4}\rangle \\
& true \wedge (\forall a : a \in (selSort.del\ \downarrow x\, x) : \downarrow x \leq a) \\
=\langle & \textbf{identidade}(\wedge)\rangle \\
& (\forall a : a \in (selSort.xdel \downarrow x x) : \downarrow x \leq a) \\
\Leftarrow\langle & \textit{lema a provar: } (\forall a : a \in x : a \in selSort x)\rangle \\
& (\forall a : a \in del\ \downarrow x\, x : \downarrow x \leq a) \\
=\langle & \textit{exercício 4}\rangle \\
& true
\end{array}$$

Exercício 6. *Dados os axiomas*

$$rev \bot = \bot \tag{44}$$

$$rev\, (a \triangleleft x) = (rev\, x) \triangleright a \tag{45}$$

prove que

(1) $rev\, .x \tilde{} y = (rev\, y) \tilde{} (rev\, x)$
(2) $rev\, .x \tilde{} y \tilde{} z = rev\, z .\tilde{} .rev\, y \tilde{} .rev\, x$
(3) $rev\, (x \triangleright a) =. a \triangleleft .rev\, x$

Exercício 7. *Considere os axiomas:*

(46) $ispref(\bot, x)$

(47) $\neg ispref(a \lhd x, \bot)$

(48) $ispref(a \lhd x, b \lhd y) =. \ a = b \wedge ispref(x, y)$

Prove que

$$ispref(x, y) = (\exists z : z \in seq.a : y = x \^{} z)$$

4. Árvore binária de pesquisa

Os axiomas a seguir definem uma árvore binária sobre um conjunto (tipo de dados) A:

(49) $\bot \in bint.A$

(50) $x, y \in bint.A \wedge a \in A \Rightarrow \langle x, a, y\rangle \in bint.A$

(51) Fecho universal sobre $49, 50$

A altura de uma árvore é dada pelos axiomas a seguir:

(52) $h\bot = 0$

(53) $h\langle x, a, y\rangle = 1 + .(hx) \uparrow (hy)$

Os axiomas a seguir estabelecem o conceito de pertinência em árvore binária.

(54) $a \notin \bot$

(55) $a \in \langle x, b, y\rangle = a \in x \vee a = b \vee a \in y$

O princípio de indução para árvores binárias estabelece uma indução sobre altura da árvore. Sua definição baseia-se no princípio de indução sobre $\mathbb{N}$, o conjunto dos inteiros não negativos. O axioma de indução sobre árvores binárias é obtido de 25 substituindo $P.i$ por uma propriedade R que se objetiva provar sobre árvores de altura i.

(56)

$$BINTP(R, i) \stackrel{\triangle}{=} (\forall x \in bint.A : hx = i : R.x)$$

(57)

$$(\forall x : x \in bint.A : R.x) \quad =$$
$$(\forall n : n \in \mathbb{N} : (\forall i \in \mathbb{N} : 0 \leq i < n : BINTP(R, i)) \Rightarrow BINTP(R, n))$$

Exercício 8. *Prove que o axioma 57 estabelece que, para provar que uma propriedade R vale para qualquer árvore binária, basta provar que:*

- *R vale para a árvore vazia;*

- *o fato de que R vale para qualquer árvore de altura menor do que k, implica que R vale para qualquer árvore de altura k, onde k é qualquer inteiro positivo.*

As condições de ordenação de uma árvore binária são definidas a seguir.

$$(\bot)\nearrow \tag{58}$$

$$\begin{aligned}\langle x, a, y\rangle\nearrow = & x\nearrow \\ & \wedge\, y\nearrow \\ & \wedge\, (\forall b : b \in x : b < a) \\ & \wedge\, (\forall b : b \in y : b > a)\end{aligned} \tag{59}$$

Exercício 9. *Assumindo que x e y são árvores binárias, e dados os axiomas*

$$a \rightsquigarrow \bot = \langle \bot, a, \bot\rangle \tag{60}$$

$$a < b \Rightarrow .\ a \rightsquigarrow \langle x, b, y\rangle = \langle a \rightsquigarrow x, b, y\rangle \tag{61}$$

$$a = b \Rightarrow .\ a \rightsquigarrow \langle x, b, y\rangle = \langle x, b, y\rangle \tag{62}$$

$$a > b \Rightarrow .\ a \rightsquigarrow \langle x, b, y\rangle = \langle x, b, a \rightsquigarrow y\rangle \tag{63}$$

Prove:

(1) $(x)\nearrow \Rightarrow .\ b \in a \rightsquigarrow x =. \ a = b \vee b \in x$
(2) $x.\nearrow \Rightarrow (a \rightsquigarrow x)\nearrow$

Exercício 10. *Assumindo que x e y são árvores binárias, e dados os axiomas*

$$\downarrow\langle \bot, a, y\rangle = a \tag{64}$$

$$\downarrow\langle x, a, y\rangle = \downarrow x \tag{65}$$

$$del\, a\, \bot = \bot \tag{66}$$

$$a = b \Rightarrow .\ del\, a\, \langle \bot, b, y\rangle = y \tag{67}$$

$$a = b \wedge x \neq \bot \Rightarrow .\ del\, a\, \langle x, b, y\rangle = \langle del\, (\uparrow x)\, x, \uparrow x, y\rangle \tag{68}$$

$$a < b \Rightarrow .\ del\, a\, \langle x, b, y\rangle = \langle del\, a\, x, b, y\rangle \tag{69}$$

$$a > b \Rightarrow .\ del\, a\, \langle x, b, y\rangle = \langle x, b, del\, a\, y\rangle \tag{70}$$

(71)

Prove

(1) $(x)\nearrow \Rightarrow (del\, a\, x)\nearrow$
(2) $a \in del\, b\, x =. \ a \neq b \wedge a \in x$

5. Alterações

Esta lista sobreu as seguintes alterções desde a versão do dia 29/11/04:

- Princípio de indução sobre $\mathbb{N}$ foi transferido para a seção 3 para facilitar algumas provas sobre seqüências.
- Antigo exercício 7 foi movido para também com indução sobre naturais: é agora exercício 2.
- Exercício sobre relação entre min e $(\Downarrow)$, antigo exercício 2, foi corrigido.
- Os seguintes exercícios estão solucionados: primeiro exercício sobre quantificadores e prova de correção da ordenação produzida pelo $selSort$.
- Corrigido o exercício 4.

Referências

1. Haskell B. Curry, *Foundations of mathematical logic*, Dover Publications, Inc., 1977.
2. David Gries and Fred B.Schneider, *A logical approach to discrete mathematics*, Springer-Verlag, 1993.
3. Stephen Cole Kleene, *Introduction to metamathematics*, North-Holland Publishing CO., 1964.