Diskrečioji matematika, IT 1 kursas

Dr. Robertas Petuchovas

Vilniaus universitetas

2016 - 2017 m.

III skyrius. Konstravimo technikos

3.1 Indukciškai apibrėžiamos aibės

Norint apibrėžti aibę S indukciškai, reikia atlikti tris žingsnius:

1. Bazė: Pateikti vieną ar daugiau elementų iš S.
2. Indukcija: Aprašyti vieną ar daugiau taisyklių sukonstruoti S elementams iš jau esančių elementų.
3. Uždarumas: Nurodyti, kad jokie kiti elementai nepatenka į aibę S (visada tarsim pagal nutylėjimą).

Pastaba. Bazės elementai ir indukcijos taisyklės yra vadinami konstruktoriais.

1 pavyzdys. Rasime induktyvų aibės

$$S = \{3, 16, 29, 42, \ldots\}$$

apibrėžimą.

Sprendimas: *Bazė*: $3 \in S$.
Indukcija: Jeigu $x \in S$, tai $x + 13 \in S$.

---

2 pavyzdys. Apibūdinsime aibę S, indukciškai apibrėžtą taip:

Bazė: $2 \in S$.

Indukcija: Jeigu $x \in S$, tai $x \pm 3 \in S$.

Ats.: $S = \{\ldots, -7, -4, -1,\ 2,\ 5,\ 8, \ldots\}$.

---

3 pavyzdys. Rasime induktyvų

$$S = \{3, 4, 5, 8, 9, 12, 16, 17, 20, 24, 33, \ldots\}$$

apibrėžimą.

Sprendimas: Panaudosime „skaldyk ir valdyk” techniką. Turime

$$S = \{3, 5, 9, 17, 33, \ldots\} \cup \{4, 8, 12, 16, 20, 24, \ldots\}.$$

Todėl

Bazė: $3, 4 \in S$.

Indukcija: Jeigu $x \in S$, kai x yra nelyginis, tada $2x - 1 \in S$, kitu atveju $x + 4 \in S$.

4 pavyzdys. Rasime induktyvų

$$S = \{\Lambda, ac, aacc, aaaccc, \ldots\} = \{a^n c^n \mid n \in \mathbb{N}\}$$

apibrėžimą.

Sprendimas: *Bazė:* $\Lambda \in S$.

Indukcija: Jeigu $x \in S$, tai $axc \in S$.

---

5 pavyzdys. Rasime induktyvų $S = \{a^{n+1} bc^n \mid n \in \mathbb{N}\}$ apibrėžimą.

Sprendimas: *Bazė:* $ab \in S$.

Indukcija: Jeigu $x \in S$, tai $axc \in S$.

6 pavyzdys. Apibūdinsime aibę S, kurios apibrėžimas toks:
Bazė: $a, b \in S$.
Indukcija: Jeigu $x \in S$, tai $f(x) \in S$.

Sprendimas:

$$S = \{a, f(a), f(f(a)), \ldots\} \cup \{b, f(b), f(f(b)), \ldots\}$$

arba, kitaip užrašius,

$$\begin{aligned} S &= \{f^n(a) \mid n \in \mathbb{N}\} \cup \{f^n(b) \mid n \in \mathbb{N}\} \\ &= \{f^n(x) \mid x \in \{a, b\} \text{ ir } n \in \mathbb{N}\}. \end{aligned}$$

Toliau prisiminkime, $\mathsf{cons}(\mathsf{x}, \mathsf{t}) = \mathsf{x} :: \mathsf{t}$. Operacija $::$ nėra asociatyvi, turime tik $x :: y :: z = x :: (y :: z)$.

7 pavyzdys. Apibūdinsime aibę S apibrėžtą taip:
Bazė: $\langle 0 \rangle \in S$.
Indukcija: Jeigu $x \in S$, tai $1 :: x \in S$.
Ats.: $S = \{\langle 0 \rangle, \langle 1, 0 \rangle, \langle 1, 1, 0 \rangle, \ldots\}$.

---

8 pavyzdys. Rasime induktyvų $S = \{\langle\rangle, \langle a, b \rangle, \langle a, b, a, b \rangle, \ldots\}$ apibrėžimą.
Sprendimas: *Bazė:* $\langle\rangle \in S$.
Indukcija: Jeigu $x \in S$, tai $a :: b :: x \in S$.

---

9 pavyzdys. Rasime induktyvų $S = \{\langle\rangle, \langle\langle\rangle\rangle, \langle\langle\langle\rangle\rangle\rangle, \ldots\}$ apibrėžimą.
Sprendimas: Bazė $\langle\rangle \in S$.
Indukcija: Jeigu $x \in S$, tai $x :: \langle\rangle \in S$.

Binariųjų medžių užrašymas

Binarųjį medį žymėsime

$$t(L, x, R);$$

čia x yra medžio šaknis, L yra kairysis x pografis, o R - dešinysis x pografis. Tuščią medį žymėsime $\langle\rangle$. Jeigu

$$T = t(L, x, R),$$

tai:

$\text{root(T)} = \text{x},$

$\text{left(T)} = \text{L},$

$\text{right(T)} = \text{R}.$

10 pavyzdys. Apibūdinsime aibę S, kurios indukcinis apibrėžimas toks:

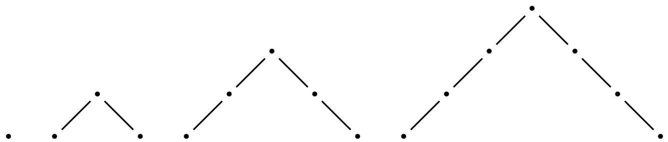
Bazė: $t(\langle\rangle, \bullet, \langle\rangle) \in S$.

Indukcija: Jeigu $T \in S$, tai

$$t\,(T, \bullet, t(\langle\rangle, \bullet, \langle\rangle)) \in S.$$

---

11 pavyzdys. Rasime induktyvų aibės S apibrėžimą, kai žinome, kad ją sudaro tokie medžiai:



ir t.t.

Atsakymas:

Bazė: $t(\langle\rangle, \bullet, \langle\rangle) \in S$.

Indukcija: Jei $T \in S$, tai

$$t(t(left(T), \bullet, \langle\rangle),\ \bullet\ ,\ t(\langle\rangle, \bullet, right(T))) \in S.$$

---

12 pavyzdys. Rasime induktyvų $S = \{a\}^* \times \mathbb{N}$ apibrėžimą.

Sprendimas:

Bazė: $(\Lambda, 0) \in S$.

Indukcija: Jeigu $(s, n) \in S$, tai

$$(as, n),\ (s, n+1) \in S.$$

---

13 pavyzdys. Rasime induktyvų aibės

$$S = \{(x, -y) \mid x, y \in \mathbb{N} \text{ ir } x \geq y\}$$

apibrėžimą. Vienas iš sprendimų:

Bazė: $(0, 0) \in S$.

Indukcija: Jeigu $(x, y) \in S$, tai

$$(x + 1, y),\ (x + 1, y - 1) \in S.$$

Užduotis. Suraskite sprendimą, kuris nekonstruotų tų pačių elementų keletą kartų.

Sprendimas:

Bazė: $(0, 0) \in S$.

Indukcija: 1. Jeigu $(x, y) \in S$, tai $(x + 1, y) \in S$.

2. Jeigu $(x, -x) \in S$, tai $(x + 1, x - 1) \in S$.

Rekurentiškai apibrėžiamos funkcijos ir procedūros

Funkcija f yra apibrėžta rekurentiškai, jeigu bent viena reikšmė $f(x)$ yra apibrėžta per kitą reikšmę $f(y)$, $x \neq y$.

Taip pat, sakome, kad procedūra P yra apibrėžta rekurentiškai, jeigu veiksmas $P(x)$ yra apibrėžtas per kitą veiksmą $P(y)$, $x \neq y$.

Rekurentinio apibrėžimo technika (kai argumento x reikšmių sritis S yra indukciškai apibrėžiama):

1. Nustatome $f(x)$ reikšmę (veiksmą $P(x)$) su kiekvienu bazės elementu $x \in S$.

2. Nurodome taisykles, kiekvienam indukciškai apibrėžtam elementui $x \in S$ nustatančias $f(x)$ reikšmę (veiksmą $P(x)$) per jau nustatytas f reikšmes (veiksmus P).

1 pavyzdys. Rasime rekurentinę funkcijos f formulę, kai žinome, kad $f : \mathbb{N} \to \mathbb{N}$ ir

$$f(n) = 0 + 3 + 6 + \ldots + 3n.$$

Sprendimas: Pastebime, kad $\mathbb{N}$ yra induktyviai apibrėžiama aibė: $0 \in \mathbb{N}$; jeigu $n \in \mathbb{N}$, tai $n + 1 \in \mathbb{N}$. Todėl turime nustatyti $f(0)$ reikšmę ir apibrėžti $f(n+1)$ per $f(n)$. Matome, kad $f(0) = 0$. Toliau pastebime,

$$\begin{aligned} f(n+1) &= (0 + 3 + 6 + \ldots + 3n) + 3(n+1) \\ &= f(n) + 3(n+1). \end{aligned}$$

Taigi, funkciją f galime užrašyti tokia rekurentine formule:

$f(0) = 0, \quad f(n+1) = f(n) + 3(n+1)$, kai $n \geq 0$.

Du alternatyvūs apibrėžimai:

- $f(0) = 0, \quad f(n) = f(n-1) + 3n \quad (n \geq 1)$.
- $f(n) = \textbf{if } n = 0 \textbf{ then } 0 \textbf{ else } f(n-1) + 3n$.

2 pavyzdys. Rasime f-jos cat rekurentinį apibrėžimą, kai

$$\mathsf{cat} : \mathsf{A}^* \times \mathsf{A}^* \to \mathsf{A}^*$$

ir

$$\mathsf{cat}(\mathsf{s}, \mathsf{t}) = \mathsf{st}.$$

Sprendimas: Turime, kad A^* induktyviai apibrėžiama: $\Lambda \in \mathsf{A}^*$; jeigu $\mathsf{a} \in \mathsf{A}$ ir $\mathsf{x} \in \mathsf{A}^*$, tai $\mathsf{ax} \in \mathsf{A}^*$. Funkciją cat apibrėšime pagal pirmą argumentą. Pastebime,

$$\mathsf{cat}(\Lambda, \mathsf{t}) = \Lambda\mathsf{t} = \mathsf{t}$$

ir

$$\mathsf{cat}(\mathsf{ax}, \mathsf{t}) = \mathsf{axt} = \mathsf{a}(\mathsf{xt}) = \mathsf{a}\,\mathsf{cat}(\mathsf{x}, \mathsf{t}).$$

Taigi, rekurentinis apibrėžimas toks:

$\mathsf{cat}(\Lambda, \mathsf{t}) = \mathsf{t}$,

$\mathsf{cat}(\mathsf{ax}, \mathsf{t}) = \mathsf{a}\,\mathsf{cat}(\mathsf{x}, \mathsf{t})$.

if-then-else forma:

Pastaba. Operacijas cons, head, tail galime naudoti ir žodžiams; $\mathsf{ax} = \mathsf{cons}(\mathsf{a}, \mathsf{x})$. Todėl paskutinio pavyzdžio atsakymą galime parašyti ir taip.

$\mathsf{cat}(\mathsf{s}, \mathsf{t}) = \mathbf{if}\ \mathsf{s} = \Lambda\ \mathbf{then}\ \mathsf{t}\ \mathbf{else}$

$$\mathsf{head}(\mathsf{s})\,\mathsf{cat}(\mathsf{tail}(\mathsf{s}), \mathsf{t}).$$

3 pavyzdys. Tegul

$$f : lists(\mathbb{Q}) \to \mathbb{Q}$$

ir

$$f(\langle x_1, \ldots, x_n \rangle) = x_1 + \ldots + x_n.$$

Apibrėšime funkciją f rekurentiškai.

Sprendimas: Aibė $lists(\mathbb{Q})$ yra induktyviai apibrėžiama: $\langle\rangle \in lists(\mathbb{Q})$; jeigu $h \in \mathbb{Q}$ ir $t \in lists(\mathbb{Q})$, tai $h :: t \in lists(\mathbb{Q})$. Pastebėkime, kad f galime užrašyti kitu būdu:

$$\begin{aligned} f(\langle x_1, \ldots, x_n \rangle) &= x_1 + f(\langle x_2, \ldots, x_n \rangle) \\ &= head(\langle x_1, \ldots, x_n \rangle) + f(tail(\langle x_1, \ldots, x_n \rangle)). \end{aligned}$$

Paimkime $f(<>) = 0$. Priėjome prie atsakymo:

$f(<>) = 0$, $f(h :: t) = h + f(t)$.

if-then-else forma:

$f(L) = \textbf{if } L = \langle\rangle \textbf{ then } 0 \textbf{ else } head(L) + f(tail(L))$.

4 pavyzdys. Funkcija $f : \mathbb{N} \to \mathbb{N}$ apibrėžiama rekurentiškai:

$$f(0) = 0, \quad f(1) = 0, \quad f(x+2) = 1 + f(x).$$

Jos *if-then-else* forma:

$$f(x) = \textbf{if } x = 0 \textbf{ or } x = 1 \textbf{ then } 0 \textbf{ else } 1 + f(x-2).$$

Ką daro funkcija f?

Sprendimas: Kad išsiaiškintume, išrašykime keletą f reikšmių.

$$map(f, \langle 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 \rangle) = \langle 0, 0, 1, 1, 2, 2, 3, 3, 4, 4 \rangle.$$

Taigi, $f(x)$ reikšmė yra $\lfloor x/2 \rfloor$, t.y. $f(x) = \lfloor x/2 \rfloor$.

5 pavyzdys. Tegul

$$f : lists(\mathbb{Q}) \to \mathbb{Q}$$

ir

$$f(\langle x_1, \ldots, x_n \rangle) = x_1 x_2 + x_2 x_3 + \ldots + x_{n-1} x_n.$$

Rasime f rekurentinį apibrėžimą.

Sprendimas: Tegul $f(\langle\rangle) = 0$ ir $f(\langle x \rangle) = 0$. Kai $n \geq 2$, galime parašyti

$$\begin{aligned} f(\langle x_1, \ldots, x_n \rangle) &= x_1 x_2 + (x_2 x_3 + \ldots + x_{n-1} x_n) \\ &= x_1 x_2 + f(\langle x_2, \ldots, x_n \rangle). \end{aligned}$$

Todėl f rekurentinė formulė tokia: $f(\langle\rangle) = 0$, $f(\langle x \rangle) = 0$,

$$f(h :: t) = h \cdot head(t) + f(t).$$

Per *if-then-else*:

$$\begin{aligned} f(L) = \ &\textbf{if } L = \langle\rangle \textbf{ or } tail(L) = \langle\rangle \textbf{ then } 0 \\ &\textbf{else } head(L) \cdot head(tail(L)) + f(tail(L)). \end{aligned}$$

6 pavyzdys. Funkcija

$$\mathit{isin} : A \times \mathit{lists}(A) \to (\mathit{true}, \mathit{false})$$

tikrina, ar x yra sąraše L. Rasime jos rekurentinę formulę.

Sprendimas:

$\mathit{isin}(x, \langle\rangle) = \mathit{false}$,

$\mathit{isin}(x, x :: t) = \mathit{true}$,

$\mathit{isin}(x, h :: t) = \mathit{isin}(x, t)$.

if-then-else pavidalu:

$$\begin{aligned}\mathit{isin}(x, L) = {} & \textbf{if } L = \langle\rangle \textbf{ then } \mathit{false} \textbf{ else} \\ & \textbf{if } x = \mathit{head}(L) \textbf{ then } \mathit{true} \\ & \textbf{else } \mathit{isin}(x, \mathit{tail}(L)).\end{aligned}$$

7 pavyzdys. Rasime rekurentinį funkcijos

$$\mathit{sub} : \mathit{lists}(A) \times \mathit{lists}(A) \to \{\mathit{true}, \mathit{false}\}$$

apibrėžimą, kuris parodo, ar L elementai yra M elementai.

Sprendimas: $\mathit{sub}(\langle\rangle, M) = \mathit{true}$ ir

$$\mathit{sub}(h :: t, M) = \textbf{if } \mathit{isin}(h, M) \textbf{ then } \mathit{sub}(t, M) \textbf{ else } \mathit{false}.$$

if-then-else pavidalu:

$$\begin{aligned} \mathit{sub}(L, M) = {} & \textbf{if } L = \langle\rangle \textbf{ then } \mathit{true} \textbf{ else} \\ & \textbf{if } \mathit{isin}(\mathit{head}(L), M) \textbf{ then } \mathit{sub}(\mathit{tail}(L), M) \\ & \textbf{else } \mathit{false}. \end{aligned}$$

8 pavyzdys. Rasime rekurentinę funkcijos

$$\mathit{intree} : \mathbb{Q} \times \mathit{binSearchTrees}(\mathbb{Q}) \to \{\mathit{true}, \mathit{false}\}$$

formulę, kuri patikrina, ar x yra binariajame paieškos medyje T.

Sprendimas:

$\mathit{intree}(x, \langle\rangle) = \mathit{false}$,

$\mathit{intree}(x, \mathit{tree}(L, x, R)) = \mathit{true}$,

$$\begin{aligned}\mathit{intree}(x, \mathit{tree}(L, y, R)) = {} & \textbf{if } x < y \textbf{ then } \mathit{intree}(x, L) \\ & \textbf{else } \mathit{intree}(x, R).\end{aligned}$$

if-then-else forma:

$$\begin{aligned}\mathit{intree}(x, T) = \textbf{if } T = \langle\rangle \textbf{ then } & \mathit{false} \\ & \textbf{else if } x = \mathit{root}(T) \textbf{ then } \mathit{true} \\ & \textbf{else if } x < \mathit{root}(T) \textbf{ then} \\ & \qquad \mathit{intree}(x, \mathit{left}(T)) \\ & \textbf{else } \mathit{intree}(x, \mathit{right}(T)).\end{aligned}$$

Binariųjų medžių apkeliavimas

Trys standartinės procedūros binariajam medžiui apkeliauti, apibrėžiamos rekurentiškai, yra tokios:

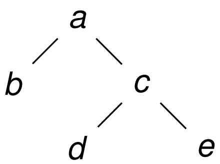
$\mathit{preorder}(T)$: $\textbf{if } T \neq \langle\rangle \textbf{ then}$
$\mathit{visit\ root}(T)$; $\mathit{preorder}(\mathit{left}(T))$; $\mathit{preorder}(\mathit{right}(T))$ $\textbf{fi}$.

$\mathit{inorder}(T)$: $\textbf{if } T \neq \langle\rangle \textbf{ then}$
$\mathit{inorder}(\mathit{left}(T))$; $\mathit{visit\ root}(T)$; $\mathit{inorder}(\mathit{right}(T))$ $\textbf{fi}$.

$\mathit{postorder}(T)$: $\textbf{if } T \neq \langle\rangle \textbf{ then}$
$\mathit{postorder}(\mathit{left}(T))$; $\mathit{postorder}(\mathit{right}(T))$; $\mathit{visit\ root}(T)$ $\textbf{fi}$.

---

9 pavyzdys. Apkeliausime medį



kiekvienu iš trijų būdų.

Sprendimas: Preorder: abcde
Inorder: badce
Postorder: bdeca

Kitame pavyzdyje prireiks funkcijos cat, kuri apjungia du sąrašus. Jos apibrėžimas toks:

$$cat(\langle\rangle,\ L) = L,$$

$$cat(h :: t, L) = h :: cat(t, L).$$

10 pavyzdys. Sugalvosime rekurentinį funkcijos

$$post : binaryTrees(A) \to lists(A)$$

apibrėžimą, kai $post(T)$ yra medžio T viršūnių sąrašas, kurios jame surašytos postorder medžio T apkeliavimo tvarka.

Sprendimas:

$post(\langle\rangle) = \langle\rangle$,

$post(tree(L, x, R)) = cat(post(L),\ cat(post(R), \langle x\rangle))$.

---

11 pavyzdys. Rasime rekurentinį funkcijos

$$f : binaryTrees(\mathbb{Q}) \to \mathbb{Q}$$

apibrėžimą, kai $f(T)$ yra medžio T viršūnių suma.

Sprendimas:

$f(\langle\rangle) = 0, \quad f(tree(L, x, R)) = x + f(L) + f(R)$.

Begalinės sekos

Tegul $f(x)$ apibrėžia begalinio ilgio seką (begalinio ilgio sąrašą). Galime sukonstruoti $f(x)$ rekurentinį apibrėžimą, reikšmę $f(x)$ apibrėžę per x ir $f(y)$; $y \neq x$.

12 pavyzdys. Apibrėšime

$$f(x) = \langle x, x^2, x^4, x^8, \ldots \rangle$$

rekurentiškai.

Sprendimas:

$$\begin{aligned} f(x) &= \langle x, x^2, x^4, x^8, \ldots \rangle \\ &= x :: \langle x^2, x^4, x^8, \ldots \rangle \\ &= x :: f(x^2). \end{aligned}$$

13 pavyzdys. Kokia yra seka $g(x,k) = x^k :: g(x,k+1)$?
Atsakymas:

$$\begin{aligned} g(x,k) &= x^k :: g(x,k+1) \\ &= x^k :: x^{k+1} :: g(x,k+2) \\ &= \dots \\ &= \langle x^k, x^{k+1}, x^{k+2}, \dots \rangle. \end{aligned}$$

---

14 pavyzdys. Aprašysime seką $l(x) = \langle x, x^3, x^5, x^7, \dots \rangle$.
Sprendimas:
Tegul $h(x,k) = x^k :: h(x,k+2)$, tada $l(x) = h(x,1)$.

---

15 pavyzdys. Aprašysime seką $s(x) = \langle 1, x^2, x^4, x^6, \dots \rangle$.
Sprendimas:
$s(x) = h(x,0)$ (žr. 14 pavyzdį).

3.3 Gramatikos

Ap. Gramatika vadiname baigtinę aibę taisyklių, kurios vadinamos produkcijomis ir naudojamos žodžiams iš kalbos apibūdinti.

Žymėjimai. Produkcijos yra pavidalo

$$\alpha \to \beta.$$

Čia α ir β yra žodžiai virš abėcėlės, kurią sudaro terminalai ir neterminalai. Užrašą $\alpha \to \beta$ skaitome kaip „α galime pakeisti β", „α yra β", „α paimame β" ir pan.

Gramatikos, kurią sudaro keturios produkcijos, pavyzdys:

$$\{S \to aSB, \quad S \to \Lambda, \quad B \to bB, \quad B \to b\}$$

arba, sutrumpinta forma,

$$S \to aSB \mid \Lambda, \quad B \to bB \mid b.$$

- *Terminalai* – kalbos abėcėlės elementai. Šiuo atveju $\{a, b\}$.
- *Neterminalai* – gramatikos simboliai (didžiosios raidės) skirtingi nuo terminalų. Šiuo atveju aibės $\{S, B\}$ elementai.
- *Pradžios simbolis* – S; išskirtinis neterminalas, nuo kurio pradedame sudarinėti kalbos žodžius.

- *Sakinio forma* – bet kuris žodis, sudarytas iš terminalų ir neterminalų.
- *Derivacija* – sakinio formos transformacija naudojant produkcijas. Pavyzdžiui, jeigu

  $$x\alpha y$$

  yra sakinio forma ir $\alpha \to \beta$ yra produkcija, tai α pakeitimas β šiame sakinyje vadinamas *derivacijos žingsniu*, kurį užrašome tokiu būdu

  $$x\alpha y \Rightarrow x\beta y.$$

Derivacijos pavyzdys:

$$S \Rightarrow aSB \Rightarrow aaSBB \Rightarrow aaBB \Rightarrow aabBB \Rightarrow aabbB \Rightarrow aabbb.$$

Tai yra *kairinė derivacija*, kuri kiekvienu žingsniu pakeičia kairiausią neterminalą. Simbolis

$$\Rightarrow^+$$

reiškia vieną ar daugiau žingsnių, o

$$\Rightarrow^*$$

reiškia nulį ar daugiau žingsnių. Galime parašyti

$$S \Rightarrow^+ aabbb,$$

$$S \Rightarrow^* aabbb,$$

$$aSB \Rightarrow^* aSB.$$

Gramatikos kalba

Ap. Gramatikos kalba yra aibė visų terminalų, gaunamų iš pradžios simbolio.

Pavyzdys. Rasime gramatikos $S \to aSB \mid \Lambda,\ B \to bB \mid b$ kalbą.

Sprendimas: kalbos žodžių struktūrai suprasti išnagrinėsime keletą derivacijų.

$S \Rightarrow \Lambda,$

$S \Rightarrow aSB \Rightarrow aB \Rightarrow ab,$

$S \Rightarrow aSB \Rightarrow aB \Rightarrow abB \Rightarrow abbB \Rightarrow abbb,$

$S \Rightarrow aSB \Rightarrow aaSBB \Rightarrow aaBB \Rightarrow aabB \Rightarrow aabb,$

$S \Rightarrow aSB \Rightarrow aaSBB \Rightarrow aaBB \Rightarrow aabBB \Rightarrow^+ aabbbb.$

Taigi, suprantame, gramatikos kalba yra

$$\{a^n b^{n+k} \mid n, k \in \mathbb{N}\}.$$

Užduotis. Apibūdinkite gramatikos

$$S \to a \mid bcS$$

kalbą.

Atsakymas: $\{(bc)^n a \mid n \in \mathbb{N}\}$.

Gramatikų konstravimas

Pavyzdys. Rasime kalbos $\{a^n b \mid n \in \mathbb{N}\}$ gramatiką.

Atsakymas: $S \to aS \mid b$.

1 užduotis. Sugalvokite kalbos $\{ba^n \mid n \in \mathbb{N}\}$ gramatiką.

Atsakymas: $S \to Sa \mid b$.

2 užduotis. Sugalvokite kalbos $\{(ab)^n \mid n \in \mathbb{N}\}$ gramatiką.

Atsakymas: $S \to Sab \mid \Lambda$ arba $S \to abS \mid \Lambda$.

Gramatikų junginiai

Tegul L ir M yra gramatikos su pradžios simboliais atitinkamai A ir B. Taip pat, tarkime, kad šių gramatikų neterminalai irgi yra skirtingi. Tokiu atveju, galime sukonstruoti tokias gramatikas:

- $L \cup M$ gramatika, kurios pradžia $S \to A \mid B$.
- LM gramatika, kurios pradžia $S \to AB$.
- L^* gramatika, kurios pradžia $S \to AS \mid \Lambda$.

Pavyzdys. Rasime kalbos $\{a^m b^m c^n \mid m, n \in \mathbb{N}\}$ gramatiką.

Sprendimas: Pastebime, kad kalba yra kalbų

$$L = \{a^m b^m \mid m \in \mathbb{N}\} \quad \text{ir} \quad M = \{c^n \mid n \in \mathbb{N}\}$$

sandauga LM. Todėl ją galime aprašyti per gramatikų L ir M produkcijas:

$$S \to AB, \quad A \to aAb \mid \Lambda, \quad B \to cB \mid \Lambda.$$

Pavyzdys. Tegul

$$Odd = \{1, 3, 5, 7, 9, \ldots\}.$$

Rasime Odd aibės gramatiką.

Spr.: Tegul $O = \{1, 3, 5, 7, 9\}$, $P = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ ir $D = \{0\} \cup P$. Pastebime, kad

$$Odd = (PD^*)^* O.$$

Pažymime O, P ir D pradinius simbolius A, B ir C. Tada

$$A \to 1 \mid 3 \mid 5 \mid 7 \mid 9, \quad B \to A \mid 2 \mid 4 \mid 6 \mid 8 \quad \text{ir} \quad C \to B \mid 0.$$

Pažymime D^*, PD^* ir $(PD^*)^*$ pradinius simbolius E, F ir G;

$$E \to CE \mid \Lambda, \quad F \to BE \quad \text{ir} \quad G \to FG \mid \Lambda.$$

Gauname, kad kalbos Odd su pradžios simboliu S gramatika

$$S \to GA.$$

Pavyzdys. Rasime kalbos L gramatiką, kai turime jos indukcinį apibrėžimą:

Bazė: $a, b, c \in L$.
Indukcija: Jei $x, y \in L$, tai $f(x),\ g(x, y) \in L$.

Sprendimas: Kalbos L pirmųjų žodžių išrašymas gali padėti suprasti, kokios produkcijos sudaro jos gramatiką. Iš tikrųjų, kalbos L gramatika yra tokia

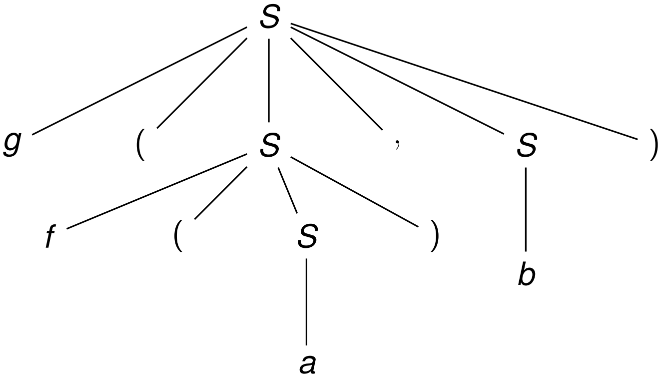
$$S \to a \mid b \mid c \mid f(S) \mid g(S, S).$$

Pavyzdžiui, žodžio $g\,(f(a), g(b, f(c)))$ derivacija gali būti tokia

$$\begin{aligned} S &\Rightarrow g(S, S) \Rightarrow g(f(S), S) \Rightarrow g(f(a), S) \Rightarrow g(f(a), g(S, S)) \\ &\Rightarrow g(f(a), g(b, S)) \Rightarrow g(f(a), g(b,\ f(S))) \Rightarrow g(f(a), g(b, f(c))). \end{aligned}$$

Sintaksės medis (A Parse Tree) – medis, kuris vaizduoja derivaciją. Jo šaknis yra pradžios simbolis, o viršūnių, kurios yra ne lapai, vaikai yra simboliai (terminalai, neterminalai ir Λ), einantys produkcijų dešinėse pusėse atitinkamuose derivacijų žingsniuose.

Pavyzdys.



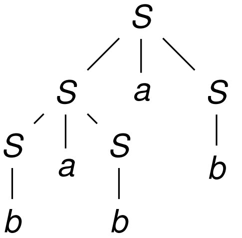
Nevienareikšmė gramatika – gramatika, kurios kalboje $\exists$ nors vienas žodis, kuris turi du skirtingus jo sintaksės medžius arba dvi skirtingas kairines (arba dešinines) derivacijas.

Klausimas. Ar gramatika $S \to SaS \mid b$ yra nevienareikšmė?

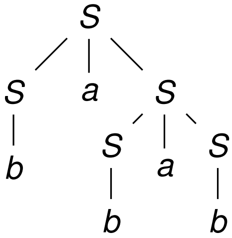
Atsakymas: Taip. Nes, pavyzdžiui, žodis $babab$ turi dvi skirtingas kairines derivacijas:

$S \Rightarrow SaS \Rightarrow SaSaS \Rightarrow baSaS \Rightarrow babaS \Rightarrow babab$,
$S \Rightarrow SaS \Rightarrow baS \Rightarrow baSaS \Rightarrow babaS \Rightarrow babab$.

Šių derivacijų sintaksės medžiai, atitinkamai, yra



ir



Klausimas. Ar gramatika $S \to abS \mid Sab \mid c$ yra nevienareikšmė?

Atsakymas: Taip, nes žodis $abcab$ turi dvi skirtingas derivacijas:

$$S \Rightarrow abS \Rightarrow abSab \Rightarrow abcab$$

ir

$$S \Rightarrow Sab \Rightarrow abSab \Rightarrow abcab.$$

Vienareikšmė gramatika – ne nevienareikšmė gramatika.

Kartais galima sugalvoti vienareikšmę gramatiką kalbai, kuriai priskirta nevienareikšmė gramatika.

Pavyzdys. Praeitame pavyzdyje parodėme, kad gramatika

$$S \to SaS \mid b$$

yra nevienareikšmė. Šios gramatikos kalba yra $\{b, bab, babab, \ldots\}$. Kita šios kalbos gramatika yra

$$S \to baS \mid b.$$

Ši yra vienareikšmė, nes S yra baS arba b ir tokiu būdu mes negalime dvejopai sukonstruoti to paties žodžio.

Pavyzdys. Paskutinėje užduotyje išsiaiškinome, kad gramatika

$$S \to c \mid abS \mid Sab$$

yra nevienareikšmė. Jos kalba

$$\{(ab)^m c(ab)^n \mid m, n \in \mathbb{N}\}.$$

Yra ir kitokia šios kalbos gramatika

$$S \to abS \mid cT \quad \text{ir} \quad T \to abT \mid \Lambda.$$

Ši yra vienareikšmė, nes S yra abS arba cT, todėl nepavyksta parašyti tos pačios sakinio formos dvejopai ir, kadangi T yra abT arba Λ, nėra dviejų skirtingų to paties žodžio derivacijų.

Rekomenduojamos literatūros sąrašas

- James L. Hein, "Discrete Structures, Logic, and Computability"
- + StudentStudyGuide.pdf