Kontextfria grammatiker

Dagens föreläsning

- Motivation och bakgrund
- Beståndsdelar
- Formell definition
- Konstruktion av CFG:er
- Flertydighet
- Chomsky-normalform

Motivation och bakgrund

Finita automater är smidiga, men klart begränsade.

Exempel: kan en finit automat skapas som accepterar språket $L = \{a^n \# b^n \mid n \geq 0\}$?

Motivation och bakgrund, forts.

Nej, det gick visst inte!

Idag skall vi se en kraftfullare modell, som exempelvis kan användas för att sätta upp syntaxregler i programmeringsspråk.

Javas grammatik kan exempelvis hittas på Internet:

`http://java.sun.com/docs/books/jls/second_edition/html/syntax.doc.html`

Beståndsdelar

Kontextfria grammatiker består av *produktioner* (regler) på formen

$$\begin{aligned} S &\rightarrow aSb \\ S &\rightarrow T \\ T &\rightarrow \# \end{aligned}$$

Vänsterledet består av en *icke-terminal* (variabel) och högerledet av en sekvens som får bestå av såväl andra icke-terminaler som *terminaler*.

Vi genererar strängar med hjälp av grammatiken genom att genomföra ersättning av icke-terminalerna, tills vi bara har terminaler kvar.

Derivering

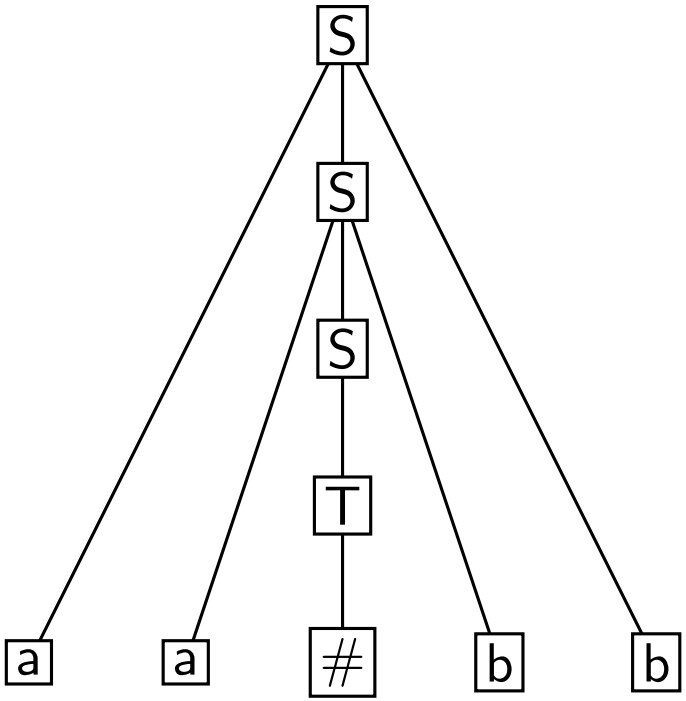
Sekvensen av ersättningar (deriveringssteg) för att generera en sträng kallas för *derivering*.

Deriveringar kan visas antingen textuellt...

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaTbb \Rightarrow aa\#bb$$

Derivering, forts.

...eller grafiskt via ett *parsningsträd*.



Mer informellt

Alla strängar som kan genereras av en kontextfri grammatik är dess *språk*. Ett språk som kan genereras av *någon* kontextfri grammatik ingår i mängden av *kontextfria språk*.

Vi brukar ofta förkorta skrivsättet avseende produktioner med samma vänsterled – högerleden radas upp och separeras med ett $|$, alltså:

$$\begin{aligned} S &\rightarrow aSb \mid T \\ T &\rightarrow \# \end{aligned}$$

Fråga 1

Konstruera en grammatik G_1 där
$L(G_1) = \{w \mid w$ börjar och slutar med samma symbol $s, s \in \{0, 1\}\}$.
Terminalerna är $\{0, 1\}$.

Fråga 1, svar

Reglerna i G_1 kan exempelvis vara följande:

$$\begin{aligned} S &\rightarrow 1 \mid 0 \mid 1T1 \mid 0T0 \\ T &\rightarrow 1T \mid 0T \mid \epsilon \end{aligned}$$

Fråga 2

Konstruera en grammatik G_2 där
$L(G_2) = \{w \mid \text{längden av strängen är udda}\}$. Terminalerna är $\{0, 1\}$.

Fråga 2, svar

Reglerna i G_2 kan exempelvis vara följande:

$$
\begin{aligned}
S &\rightarrow 0T \mid 1T \\
T &\rightarrow 0T0 \mid 0T1 \mid 1T0 \mid 1T1 \mid \epsilon
\end{aligned}
$$

Formell definition

En *kontextfri grammatik* är ett system (V, Σ, R, S) där

- V är en ändlig mängd icke-terminaler,
- Σ är en ändlig mängd, disjunkt från V av terminaler,
- R är en ändlig mängd produktioner, där varje produktion består av en icke-terminal och en sekvens av icke-terminaler och terminaler och
- $S \in V$ är startvariabeln.

Formell definition, forts.

Om u, v, w är sekvenser av variabler och terminaler, och $A \to w$ är en regel i grammatiken, säger vi att uAv genererar (*yields*) uwv, vilket skrivs $uAv \Rightarrow uwv$.

Vi säger att u härleder (*derives*) v, skrivet $u \overset{*}{\Rightarrow} v$ om $u = v$ eller om det finns en sekvens $u_1, u_2, \ldots u_k$ för $k \geq 0$ och

$$u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \ldots \Rightarrow u_k$$

Grammatikens språk är $\{w \in \Sigma^* \mid S \overset{*}{\Rightarrow} w\}$.

Fråga 3

Konstruera grammatiken G_3 som genererar strängar med perfekt balanserade parenteser, alltså exempelvis strängen $(()()((())))$?

Fråga 3, svar

Grammatiken G_3 kan exempelvis ha följande regler:

$$S \rightarrow (S) \mid SS \mid \epsilon$$

Exempel 1, begränsad aritmetik

Grammatiken $G_{ex1} = (V, \Sigma, R, \langle \text{EXPR} \rangle)$ genererar strängar innehållande aritmetiska uttryck med addition och multiplikation på korrekt form.
$\Sigma = \{a, +, \times, (,)\}$, $V = \{\langle \text{EXPR} \rangle, \langle \text{TERM} \rangle, \langle \text{FACTOR} \rangle\}$ och reglerna är:

$$
\begin{aligned}
\langle \text{EXPR} \rangle &\rightarrow \langle \text{EXPR} \rangle + \langle \text{TERM} \rangle \mid \langle \text{TERM} \rangle \\
\langle \text{TERM} \rangle &\rightarrow \langle \text{TERM} \rangle \times \langle \text{FACTOR} \rangle \mid \langle \text{FACTOR} \rangle \\
\langle \text{FACTOR} \rangle &\rightarrow (\langle \text{EXPR} \rangle) \mid a
\end{aligned}
$$

Att konstruera CFG:er

Boken ger fyra vettiga tips kring hur man konstruerar kontextfria grammatiker.

1. Grammatiker går utmärkt att kombinera (ny startregel som går till beståndsdelarnas startregler), så uppdelning kan förenkla konstruktionen!
2. Finns redan en DFA för språket är det lätt att skapa en CFG som genererar samma språk (visar detta nästa bild).
3. Kräver språket att två delsträngar är beroende av varandras struktur, placera "tillväxtzonen" i mitten enligt idén i fråga 2 med parenteserna!
4. Rekursiva strukturer som i Exempel 1 kan enkelt skapas genom att införa en regel som leder tillbaka till en "tidigare" icke-terminal.

DFA $\longrightarrow$ CFG-konvertering

- Gör en variabel R_i för varje tillstånd q_i i DFA:n
- Lägg till regeln $R_i \longrightarrow aR_j$ till CFG:n om $\delta(q_i, a) = q_j$ är en transition i DFA:n
- Lägg till regeln $R_i \longrightarrow \epsilon$ om q_i är ett accepterande tillstånd i DFA:n
- Sätt R_0 som startvariabel, där q_0 är starttillståndet i DFA:n

Vilken intressant slutsats om språkens förhållande till varandra kan vi dra av detta?

Fråga 4

Konstruera en grammatik G_4, där $L(G_4) = \{w \mid w \text{ är ett palindrom}\}$.
Terminalerna är $\{a, b, c, d\}$.

Fråga 4, svar

Reglerna i G_4 kan exempelvis vara:

$$S \quad \rightarrow \quad aSa \mid bSb \mid cSc \mid dSd \mid \epsilon$$

Flertydighet

Flertydighet dyker upp då vi med hjälp av en grammatik kan generera en sträng på fler än ett sätt. I exempelvis programmeringsspråk är detta ett problem.

Vi säger att en sträng har genererats på ett flertydigt sätt om det finns flera olika deriveringar som kan generera den. Om grammatiken kan generera någon sträng på ett flertydigt sätt är den *flertydig*.

Flertydigt exempel

Låt följande vara reglerna i grammatiken G_{ojsan}, med i övrigt samma beståndsdelar som G_{ex1}:

$$\langle \text{EXPR} \rangle \rightarrow \langle \text{EXPR} \rangle + \langle \text{EXPR} \rangle \mid \langle \text{EXPR} \rangle \times \langle \text{EXPR} \rangle \mid (\langle \text{EXPR} \rangle) \mid a$$

Hur har vi genererat strängen $a + a \times a$?

Flertydighet formellt

Vi säger att vi har gjort en vänsterderivering om vi i varje steg har ersatt den vänstraste icke-terminalen. Med hjälp av detta kan vi definiera flertydighet formellt.

En sträng w deriveras på ett flertydig sätt i en kontextfri grammatik G om den har två eller fler olika vänsterderiveringar. Grammatiken G är flertydig om den genererar någon sträng på ett flertydigt sätt.

Fråga 5

Vissa språk är ofrånkomligt flertydiga. Ett exempel är språket $L_5 = \{a^i b^j c^k \mid i = j \text{ eller } j = k\}$. Hur ser en grammatik G_5 som genererar språket L_5 ut och varför (inget formellt bevis krävs) är grammatiken flertydig?

Fråga 5, svar

Vi kan lätt styra att vi får lika många a som b och lägga till ett gäng c efteråt, eller hålla koll på antalen b och c och stoppa några a framför. Men vi har mycket svårt att säga hur strängen $aaabbbccc$ har genererats!

Ovanstående är i regelform exempelvis:

$$
\begin{aligned}
S &\rightarrow \langle AB\rangle\langle C\rangle \mid \langle A\rangle\langle BC\rangle \\
\langle A\rangle &\rightarrow a\langle A\rangle \mid \epsilon \\
\langle C\rangle &\rightarrow c\langle C\rangle \mid \epsilon \\
\langle AB\rangle &\rightarrow a\langle AB\rangle b \mid \epsilon \\
\langle BC\rangle &\rightarrow b\langle BC\rangle c \mid \epsilon
\end{aligned}
$$

Chomsky-normalform

Chomsky-normalform är en förenklad form av kontextfria grammatiker, som är mycket praktiska att använda i algoritmer. En kontextfri grammatik är på Chomsky-normalform om varje regel är på formen:

$$\begin{aligned} A &\rightarrow BC \\ A &\rightarrow a \end{aligned}$$

a är en terminal och A, B, C är icke-terminaler, men B, C får inte vara startvariabeln. Dessutom tillåts att startvariabeln har en produktion som leder till ϵ.

Bokens **teorem 2.9** säger att alla kontextfria språk kan genereras av grammatiker på Chomsky-normalform. Detta bevisas genom konstruktion, vilket den intresserade kan titta på.

Fråga till nästa gång

Med kontextfria grammatiker har vi sett att det är möjligt att avgöra om en sträng tillhör språket $\{a^n b^n\}, n \geq 0$, men kan vi skapa en grammatik som gör detsamma för språket $\{a^n b^n c^n\}, n \geq 0$? Hur ser den i så fall ut?