Syntaks og semantik

Lektion 7

4 marts 2008

Forord

1. Pushdown-automater
2. Automater med stacke
3. Grammatikker
4. Chomsky-hierarkiet

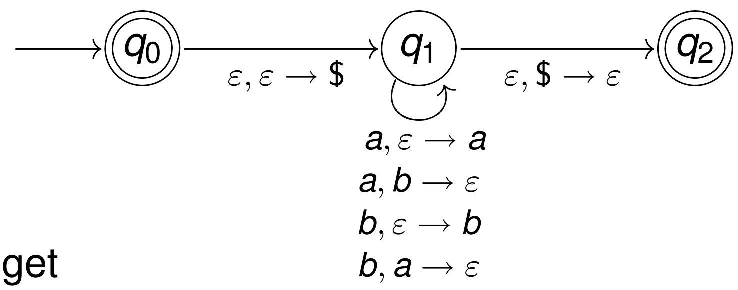
Definition 2.13: En pushdown-automat (PDA) er en 6-tupel $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$, hvor delene er

1. Q : en endelig mængde af tilstande
2. Σ : input-alfabetet
3. Γ : stack-alfabetet
4. $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to \mathcal{P}(Q \times \Gamma_\varepsilon)$: transitionsfunktionen
5. $q_0 \in Q$: starttilstanden
6. $F \subseteq Q$: mængden af accepttilstande

M siges at acceptere et ord $w \in \Sigma^*$ hvis der findes $m \in \mathbb{N}$ og $w_1, w_2, \dots, w_m \in \Sigma_\varepsilon$, $r_0, r_1, \dots, r_m \in Q$ og $s_0, s_1, \dots, s_m \in \Gamma^*$ således at $w = w_1 w_2 \dots w_m$ og

1. $r_0 = q_0$ og $s_0 = \varepsilon$,
2. for alle $i = 0, 1, \dots, m-1$ findes $a, b \in \Gamma_\varepsilon$ og $t \in \Gamma^*$ som opfylder $s_i = at$, $s_{i+1} = bt$ og $(r_{i+1}, b) \in \delta(r_i, w_{i+1}, a)$, og
3. $r_m \in F$.

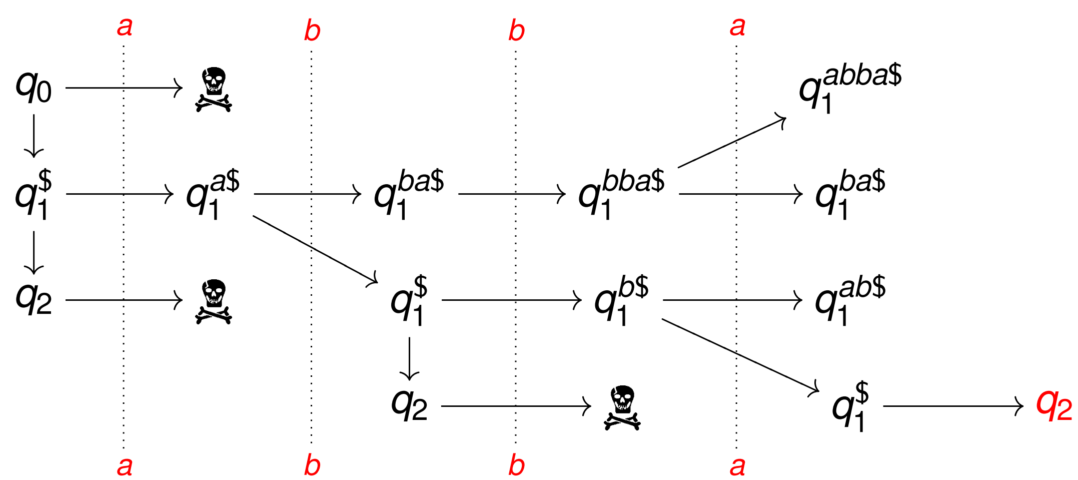
Eksempel:



Genkender sproget

$$\{w \in \{a, b\}^* \mid \text{antallet af } a \text{ i } w = \text{antallet af } b \text{ i } w\}$$

At læse strengen *abba*:



Definition: Et sprog siges at være kontekstfrit hvis der findes en CFG der genererer det.

Sætning 2.20: Et sprog er kontekstfrit hvis og kun hvis der findes en PDA der genkender det.

- at bevise CFG $\Rightarrow$ PDA:
  Lav en CFG G om til en ("generaliseret") PDA med 3 tilstande: Fra q_{start} til q_{loop} pushes startsymbolet fra G på stacken. Fra q_{loop} til sigselv er der transitioner der
  - ekspanderer en variabel i G til en af dens højresider i Gs produktioner,
  - forsøger at matche en terminal fra input med en terminal fra stacken.

  Fra q_{loop} til q_{accept} er der en transition der kun er tændt når stacken er tom.
- at bevise PDA $\Rightarrow$ CFG: Senere i dag

Definition: En automat med k stacke, for $k \in \mathbb{N}_0$, er en 6-tupel $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$, hvor delene er

1. Q : en endelig mængde af tilstande
2. Σ : input-alfabetet
3. Γ : stack-alfabetet
4. $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon^k \to \mathcal{P}(Q \times \Gamma_\varepsilon^k)$: transitionsfunktionen
5. $q_0 \in Q$: starttilstanden
6. $F \subseteq Q$: mængden af accepttilstande

- $k = 0$: NFA
- $k = 1$: PDA
- $k \geq 2$: Turing-maskine!
  – to stacke er nok!

Definition: En **grammatik** er en 4-tupel $G = (V, \Sigma, R, S)$, hvor delene er

1. V : en endelig mængde af variable
2. Σ : en endelig mængde af terminaler, med $V \cap \Sigma = \emptyset$
3. $R : (V \cup \Sigma)^+ \to \mathcal{P}\big((V \cup \Sigma)^*\big)$: produktioner
4. $S \in V$: startvariablen

- alle produktioner på formen $A \to \varepsilon$, $A \to a$ eller $A \to aB$, for $A, B \in V$ og $a \in \Sigma$: **regulær** grammatik
- alle produktioner på formen $A \to w$, for $A \in V$ og $w \in (V \cup \Sigma)^*$: **kontekstfri** grammatik
- alle produktioner på formen $uAv \to uwv$, for $A \in V$ og $u, v, w \in (V \cup \Sigma)^*$: **kontekst-sensitiv** grammatik

Eksempel på en kontekst-sensitiv grammatik:

$$S \to aBSc \mid abc \qquad Ba \to aB \qquad Bb \to bb$$

Genererer sproget $\{a^n b^n c^n \mid n \in \mathbb{N}_+\}$

	Type 3	Type 2	Type 1	Type 0
	regulære sprog	**kontekstfrie** sprog	**kontekst-sensitive** sprog	**rekursivt** enumerable sprog
	regulære grammatikker	kontekstfrie grammatikker	kontekst-sensitive grammatikker	generelle grammatikker
	endelige automater	pushdown-automater	lineært begrænsede Turing-maskiner	Turing-maskiner
determ-inisme	ingen ind-skrænkning	indskrænkning	**vides ikke**	ingen ind-skrænkning
lukning:				
$\cup, \circ, {}^*$	ja	ja	ja	ja
$\cap$	ja	nej	ja	ja
$\overline{}$	ja	nej	ja	nej

Kontekstfrie og ikke kontekstfrie sprog

5. Ethvert sprog genkendt af en PDA er kontekstfrit
6. Pumpelemmaet for kontekstfrie sprog

Lemma 2.27: Lad Σ være et alfabet og P en PDA over Σ. Da findes en CFG G over Σ med $[\![G]\!] = [\![P]\!]$.

Bevis: Lad $P = (Q, \Sigma, \Gamma, \delta, q_0, F)$. Vi konstruerer $G = (V, \Sigma, R, S)$:

1. Sørg for at P kun har én accepttilstand q_a og at stacken tømmes før P går i q_a.
2. Lad $V = \{A_{pq} \mid p, q \in Q\}$, og sørg for at A_{pq} deriverer præcist de strenge som bringer P fra p med tom stack til q med tom stack.
3. Lad $S = A_{q_0 q_a}$. Voilà!

Lemma 2.27: Lad Σ være et alfabet og P en PDA over Σ. Da findes en CFG G over Σ med $[\![G]\!] = [\![P]\!]$.

Bevis: Lad $P = (Q, \Sigma, \Gamma, \delta, q_0, F)$. Vi konstruerer $G = (V, \Sigma, R, S)$:

1. Sørg for at P kun har én accepttilstand q_a og at stacken tømmes før P går i q_a.
   Nyt stacksymbol \$. Tre nye tilstande: q_s, q_e og q_a. Nye transitioner: $q_s \xrightarrow{\varepsilon,\varepsilon\to\$} q_0$, $q \xrightarrow{\varepsilon,\varepsilon\to\varepsilon} q_e$ for alle $q \in F$, $q_e \xrightarrow{\varepsilon,a\to\varepsilon} q_e$ for alle $a \in \Sigma$, og $q_e \xrightarrow{\varepsilon,\$\to\varepsilon} q_a$.

   Sørg for at enhver transition *enten* pusher *eller* popper.
   - Erstat enhver transition $q \xrightarrow{a,b\to c} r$ med $q \xrightarrow{a,b\to\varepsilon} q_1 \xrightarrow{\varepsilon,\varepsilon\to c} r$
   - Erstat enhver transition $q \xrightarrow{a,\varepsilon\to\varepsilon} r$ med $q \xrightarrow{a,\varepsilon\to x} q_1 \xrightarrow{\varepsilon,x\to\varepsilon} r$ for et eller andet symbol $x \in \Gamma$.

Lemma 2.27: Lad Σ være et alfabet og P en PDA over Σ. Da findes en CFG G over Σ med $[\![G]\!] = [\![P]\!]$.

Bevis: Lad $P = (Q, \Sigma, \Gamma, \delta, q_0, F)$. Vi konstruerer $G = (V, \Sigma, R, S)$:

2. Lad $V = \{A_{pq} \mid p, q \in Q\}$, og sørg for at A_{pq} deriverer præcist de strenge som bringer P fra p med tom stack til q med tom stack.
   - Lav en produktion $A_{pp} \to \varepsilon$ for alle $p \in Q$ (terminering)
   - Lav en produktion $A_{pq} \to A_{pr}A_{rq}$ for alle $p, q, r \in Q$ (rekursion)
   - For alle $p, q, r, s \in Q$: Hvis $p \xrightarrow{a,\varepsilon\to t} r$ og $s \xrightarrow{b,t\to\varepsilon} q$ for nogle $a, b \in \Sigma_\varepsilon$ og et $t \in \Gamma$: Lav en produktion $A_{pq} \to aA_{rs}b$. (produktion)

   – der skal *argumenteres* for at dette giver det rigtige resultat!

Sætning 2.34: For ethvert kontekstfrit sprog A findes der et (naturligt) tal p således at ethvert ord $s \in A$ der har længde mindst p kan opsplittes i fem stykker, $s = uvxyz$, med

- $|vy| > 0$ og $|vxy| \leq p$,
- og således at ordene $uv^ixy^iz \in A$ for alle $i \in \mathbb{N}_0$.

Anvendelse: Vis a sproget X *ikke er kontekstfrit*:

Antag at X er kontekstfrit. Så må det opfylde pumpelemmaet. Lad p være pumpelængden.

Find en streng s som

- har $|s| \geq p$, dvs. *bør kunne pumpes*,
- men som *ikke kan pumpes*, ligegyldigt hvordan man opsplitter $s = uvxyz$.

Modstrid!

Bevis: Lad $G = (V, \Sigma, P, S)$ være en CFG med $[\![G]\!] = A$.

1. Lad b være længden af den længste streng på højresiden af produktionerne i G: $b = \max\{|s| \mid s \in P(A), A \in V\}$
2. Lad $p = b^{|V|+1}$. Fejl i bogen! Tag et $s \in A$ med $|s| \geq p$.
   $|V|$ er antallet af variable i G.

Bevis: Lad $G = (V, \Sigma, P, S)$ være en CFG med $[\![G]\!] = A$.

1. Lad b være længden af den længste streng på højresiden af produktionerne i G: $b = \max\{|s| \mid s \in P(A), A \in V\}$
2. Lad $p = b^{|V|+1}$. Fejl i bogen! Tag et $s \in A$ med $|s| \geq p$.
3. Lad τ være et af de parsetræer for s der har færrest punkter. τ har højde mindst $|V| + 1$.
   Lad h være højden af τ. Hvert punkt i τ har *højst b sønner*, så τ har *højst b^h blade*. Tegnene i s står i bladene, så s har længde højst b^h. Men $|s| > b^{|V|}$, så $h > |V|$.

PDA ⇒ CFG Pumpelemmaet

Bevis: Lad $G = (V, \Sigma, P, S)$ være en CFG med $[\![G]\!] = A$.

1. Lad b være længden af den længste streng på højresiden af produktionerne i G: $b = \max\{|s| \mid s \in P(A), A \in V\}$
2. Lad $p = b^{|V|+1}$. Fejl i bogen! Tag et $s \in A$ med $|s| \geq p$.
3. Lad τ være et af de parsetræer for s der har færrest punkter. τ har højde mindst $|V| + 1$.
4. Lad ℓ være en sti i τ af længde mindst $|V| + 2$.
5. ℓ indeholder mindst $|V| + 1$ variable (og én terminal), så blandt de *sidste* $|V| + 1$ variable i ℓ er der en der forekommer *to gange*. Kald den R.
6. Lad x være den delstreng af s der deriveres af den *sidste* forekomst af R. Strengen der deriveres af den *næstsidste* forekomst af R kan da skrives vxy, og $s = uvxyz$.
   Dvs. $R \overset{*}{\Rightarrow} x$, $R \overset{*}{\Rightarrow} vRy \overset{*}{\Rightarrow} vxy$, og $S \overset{*}{\Rightarrow} uRz \overset{*}{\Rightarrow} uvRyz \overset{*}{\Rightarrow} uvxyz$.

6. Lad x være den delstreng af s der deriveres af den *sidste* forekomst af R. Strengen der deriveres af den *næstsidste* forekomst af R kan da skrives vxy, og $s = uvxyz$.
7. Den næstsidste forekomst af R er blandt de sidste $|V| + 1$ variable i ℓ, så deltræet med dette R som rod har højde *højst* $|V| + 1$, så $|vxy| \leq b^{|V|+1} = p$. Fejl i bogen!
8. Ved at erstatte deltræet med det *næstsidste* R som rod, med deltræet med det *sidste* R som rod fås derivationen $S \overset{*}{\Rightarrow} uRz \overset{*}{\Rightarrow} uxz$. Dvs.
   - $uxz = uv^0xy^0z \in A$
   - $|vy| > 0$, for ellers ville $s = uxz$, og det parsetræ for uxz vi lige har lavet er mindre end det vi startede med. Modstrid til (3).
9. Ved at erstatte deltræet med det *sidste* R som rod, med deltræet med det *næstsidste* R som rod fås derivationen $S \overset{*}{\Rightarrow} uRz \overset{*}{\Rightarrow} uvRyz \overset{*}{\Rightarrow} uv^2Ry^2z \overset{*}{\Rightarrow} uv^2xy^2z$.
   Ved at gentage dette fås derivationer til uv^ixy^iz for alle $i \in \mathbb{N}$.