Chapter 4

Context-free Grammars and Languages

4.0 Review

▷

The book uses a transition funtion

$$\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \to 2^{Q \times \Gamma^*},$$

i.e., a function into (finite) subsets of $Q \times \Gamma^*$.

My personal favourite is a (finite) transition relation

$$\delta \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \times Q \times \Gamma^*.$$

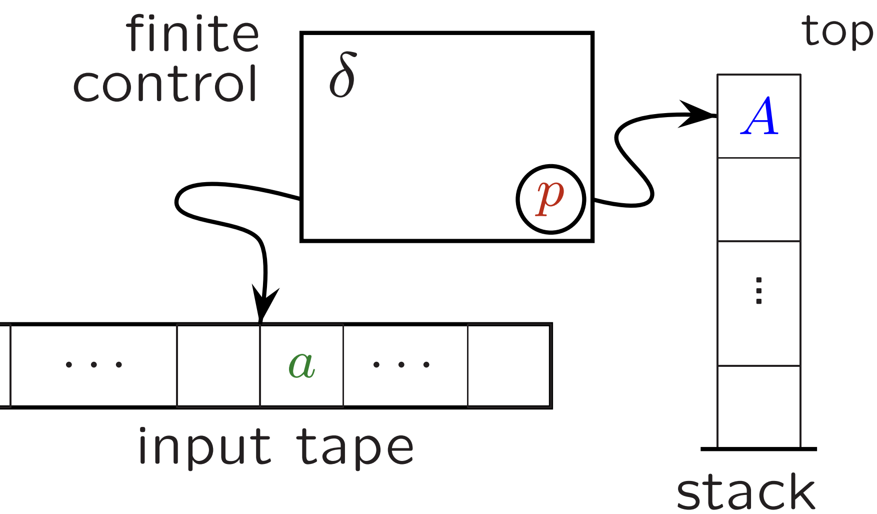
In the former one writes

$$\delta(p, a, A) \ni (q, \alpha)$$

and in the latter

$$(p, a, A, q, \alpha) \in \delta.$$

The meaning is the same. ◁



7-tuple

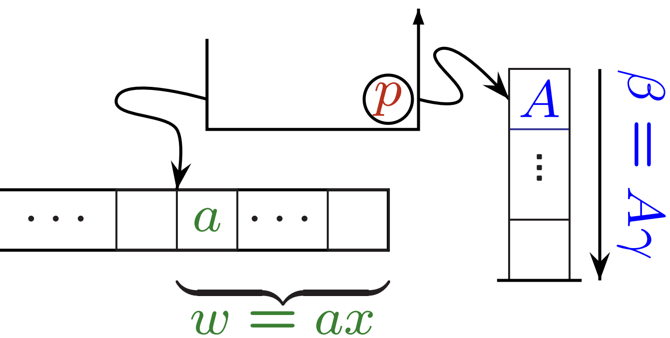
$$\mathcal{A} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$$

Q	*states*	p, q	
$q_0 \in Q$	*initial state*		
$F \subseteq Q$	*final states*		
Σ	*input alphabet*	a, b	w, x
Γ	*stack alphabet*	A, B	α
$Z_0 \in \Gamma$	*initial stack symbol*		

transition function (finite)

$$\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \to 2^{Q \times \Gamma^*}$$

$$\underbrace{(\overset{\text{from}}{p}\quad \underset{\text{read}}{a}\quad \underset{\text{pop}}{A})}_{\text{before}} \ni \underbrace{(\overset{\text{to}}{q}\quad \underset{\text{push}}{\alpha})}_{\text{after}}$$

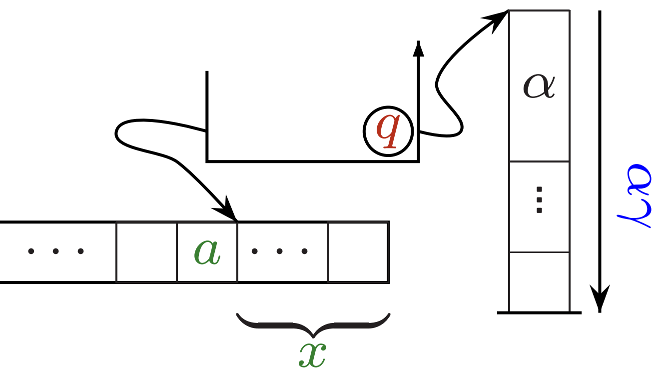


$Q \times \Sigma^* \times \Gamma^*$ configuration

(p, w, β)
- p state
- w input, unread part
- β stack, top-to-bottom

move (step) $\vdash_{\mathcal{A}}$

$(p, ax, A\gamma) \vdash_{\mathcal{A}} (q, x, \alpha\gamma)$ iff
$(p, a, A, q, \alpha) \in \delta$, $x \in \Sigma^*$ and $\gamma \in \Gamma^*$



computation $\vdash^*_{\mathcal{A}}$

$L(\mathcal{A})$ *final state language*

$\{ x \in \Sigma^* \mid (q_0, x, Z_0) \vdash^*_{\mathcal{A}} (q, \epsilon, \gamma)$
for some $q \in F$ and $\gamma \in \Gamma^* \}$

$L_e(\mathcal{A})$ *empty stack language*

$\{ x \in \Sigma^* \mid (q_0, x, Z_0) \vdash^*_{\mathcal{A}} (q, \epsilon, \epsilon)$
for some $q \in Q \}$

▷

The basic theorem of context-free languages: Theorem 1.5.6. the equivalence of cfg and pda.

It is due to

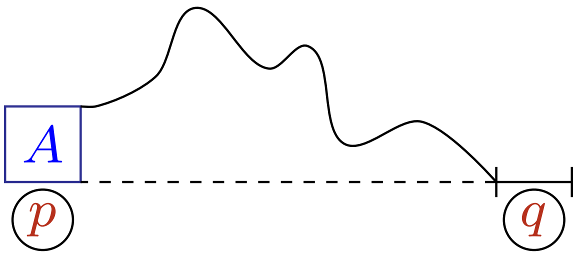
Chomsky 'Context Free Grammars and Pushdown Storage',

Evey 'Application of Pushdown Store Machines', and

Schützenberger 'On Context Free Languages and Pushdown Automata' all in 1962/3.

Starting with a pda under empty stack acceptance we construct an equivalent cfg. Its nonterminals are triplets $[p, A, q]$ representing computations of the pda. Productions result from recursively breaking down computations. A single instruction yields many productions, mainly because intermediate states of the computations have to be guessed.

◁

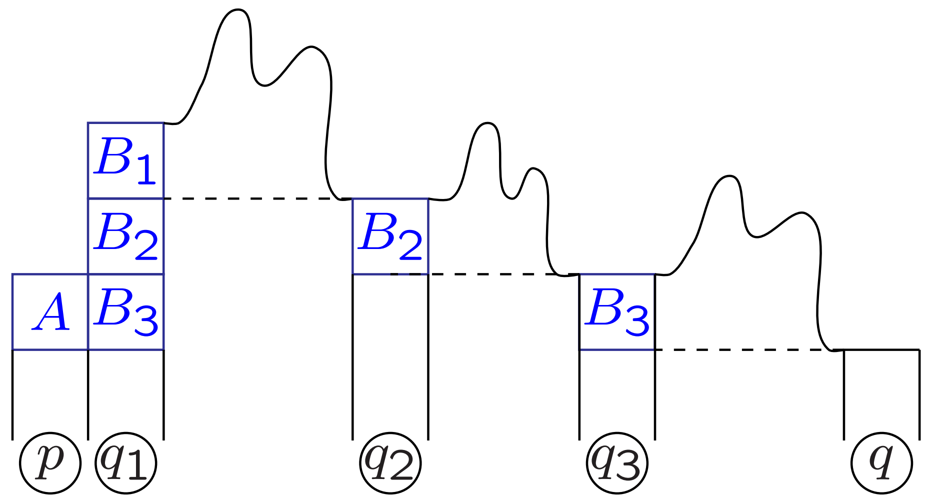


nonterminals $[\,p, A, q\,]$ $\qquad p, q \in Q,\ A \in \Gamma$

$[\,p, A, q\,] \Rightarrow_G^* w \iff (\,p, w, A\,) \vdash^* (\,q, \epsilon, \epsilon\,)$

productions

$S \to [\,q_{in}, Z_{in}, q\,]$ $\qquad$ for all $q \in Q$



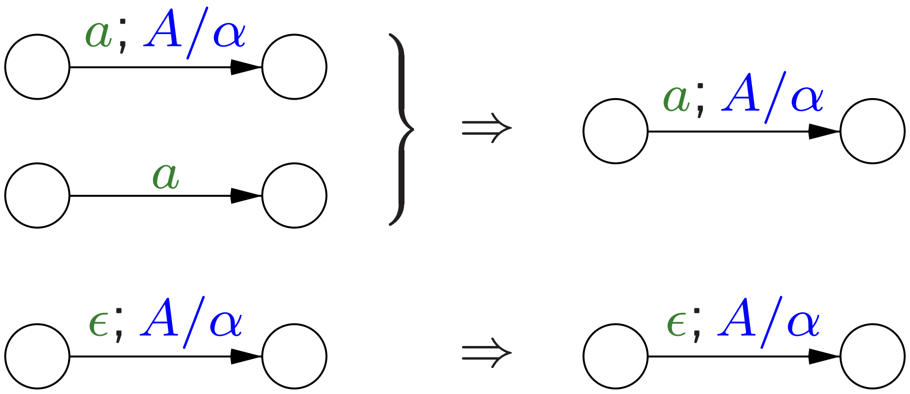
$$[\,p, A, q\,] \to a\,[\,q_1, B_1, q_2\,]\,[\,q_2, B_2, q_3\,] \cdots [\,q_n, B_n, q\,]$$

$$\delta(p, a, A) \ni (q_1, B_1 \cdots B_n)$$

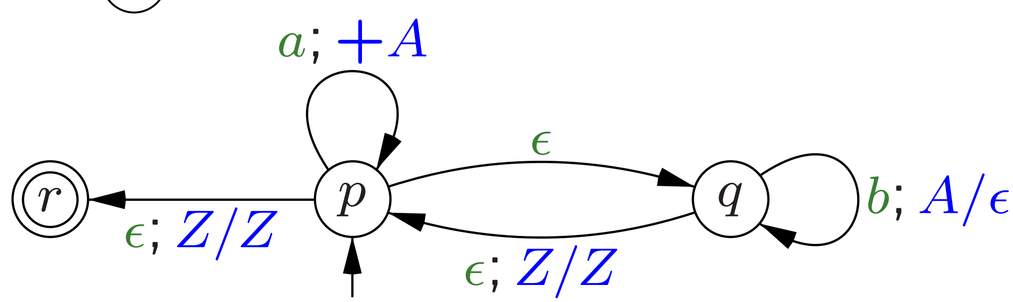
$$q, q_2, \ldots, q_n \in Q$$

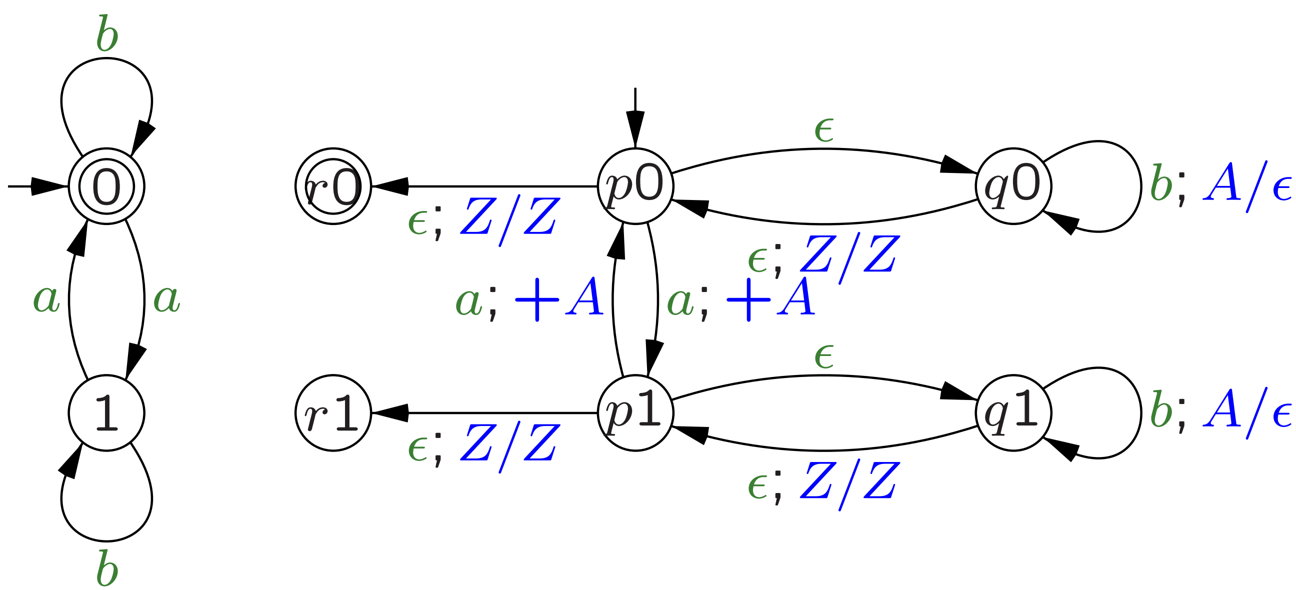
$[\,p, A, q\,] \to a$ $\qquad \delta(p, a, A) \ni (q, \epsilon)$

Theorem 1.5.6



$\{ a^n b^n \mid n \geq 1 \}^* \cap$
$\{ w \in \{a,b\}^* \mid \#_a x \text{ even} \}$





4.1 Closure properties

closed under . . .
union, concatenation, star
(using grammars)

not under intersection, complement

$L = \{ a^n b^n c^n \mid n \geq 0 \}$ not in CF

$\{a^i b^i \mid i \geq 0\} c^* \cap a^* \{b^i c^i \mid i \geq 0\}$

$\{a, b, c\}^* - L$ *is* CF (exercise)

	RLIN REG	DPDA	CF PDAe	DLBA	MON LBA	REC	TYPE0 RE
intersection	+	−	−	+	+	+	+
complement	+	+	−	+	+	+	−
union	+	−	+	+	+	+	+
concatenation	+	−	+	+	+	+	+
star, plus	+	−	+	+	+	+	+
ϵ-free morphism	+	−	+	+	+	+	+
morphism	+	−	+	−	−	−	+
inverse morphism	+	+	+	+	+	+	+
intersect reg lang	+	+	+	+	+	+	+
mirror	+	−	+	+	+	+	+
	fAFL		fAFL	AFL	AFL	AFL	fAFL

$\cap \; ^c \; \cup$ boolean operations

$\cup \; \cdot \; ^*$ regular operations

$h \; h^{-1} \; \cap R$ (full) trio operations

Next: An 'intuitive' pictorial representation of the direct product construction of a PDA and a FST, showing the image of a PDA language under a transduction is again accepted by a PDA. This proves closure of CF under several operations.

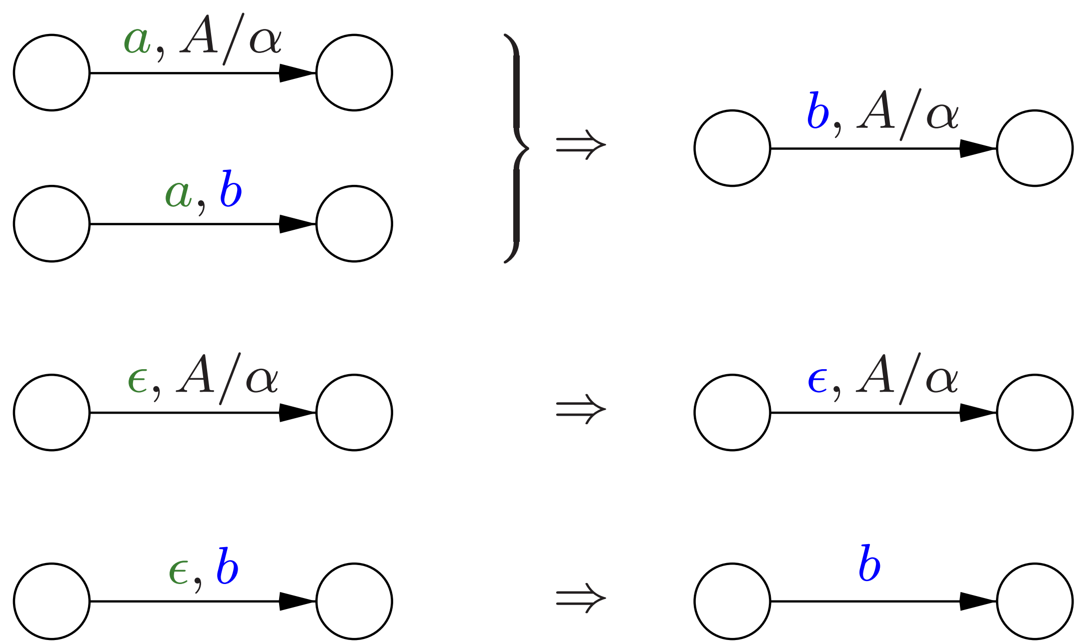
Same construction is given on the transparency after that one, but now in a more precise specification. No formal proof (induction on computations) is given.

Note! Shallit works the reverse way, from full trio operations to FST's. Recall that a family of languages is closed under FST's iff it is closed under morphisms, inverse morphisms and intersection with regular languages. The 'if'-part is Nivat's Theorem 3.5.3, the 'only-if' follows from the fact that these operations can all be performed by a suitable FST.

Thm. CF is closed under fs transductions

$L \in$ CF (given by PDA) FST $\mathcal{A} : \Sigma^* \to \Delta^*$

$$T(\mathcal{A})(L) = \{ v \in \Delta^* \mid u \in L, (u,v) \in T(\mathcal{A}) \}$$



Cor. CF is closed under morphisms, inverse morphisms; intersection, quotient & concatenation with regular languages (x3); prefix, suffix ...

$$\text{PDA } \mathcal{A} = (Q, \Delta, \Gamma, \delta, q_{in}, Z_{in}, F)$$

$$\text{FST } \mathcal{M} = (P, \Delta, \Sigma, \varepsilon, p_{in}, E)$$

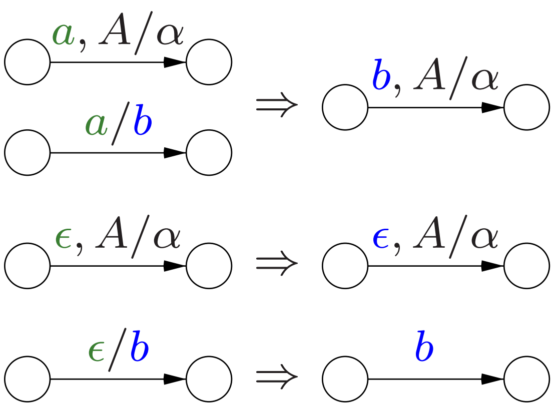
$$T(\mathcal{M})(L(\mathcal{A})) \Rightarrow \text{PDA } \mathcal{A}' = (Q', \Sigma, \Gamma, \delta', q'_{in}, Z_{in}, F')$$

formally – $Q' = Q \times P$

– $q'_{in} = \langle q_{in}, p_{in} \rangle$

– $F' = F \times E$, and

– δ' is defined by



if $\delta(q_1, a, A) \ni (q_2, \alpha)$, and $(p_1, a, b, p_2) \in \varepsilon$ (with $a \neq \epsilon$)

then $\delta'(\langle q_1, p_1 \rangle, b, A) \ni (\langle q_1, p_1 \rangle, \alpha)$

if $\delta(q_1, \epsilon, A) \ni (q_2, \alpha)$ and $p \in P$,

then $\delta'(\langle q_1, p \rangle, \epsilon, A) \ni (\langle q_1, p \rangle, \alpha)$

if $q \in Q$ and $(p_1, \epsilon, b, p_2) \in \varepsilon$,

then $\delta'(\langle q, p_1 \rangle, b, A) \ni (\langle q, p_1 \rangle, \alpha)$

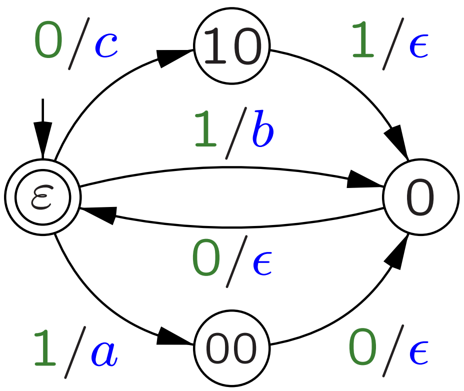
As an example of finite state transducers and the closure construction: the inverse morphism.

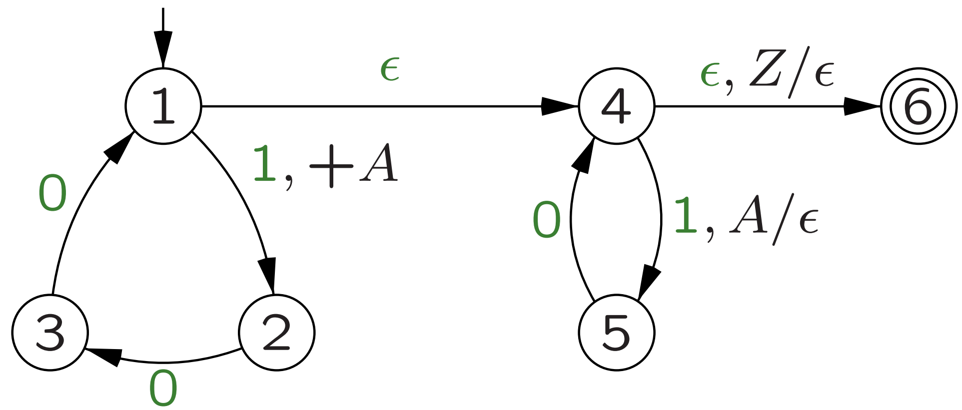
In Shallit this is Thm. 4.1.4, without explicit FST.

For a morphism h we construct a FST that realizes h^{-1}. Then for the context-free language $K = \{(100)^n(10)^n \mid n \geq 0\}$ we construct PDA for K and $h^{-1}(K)$.

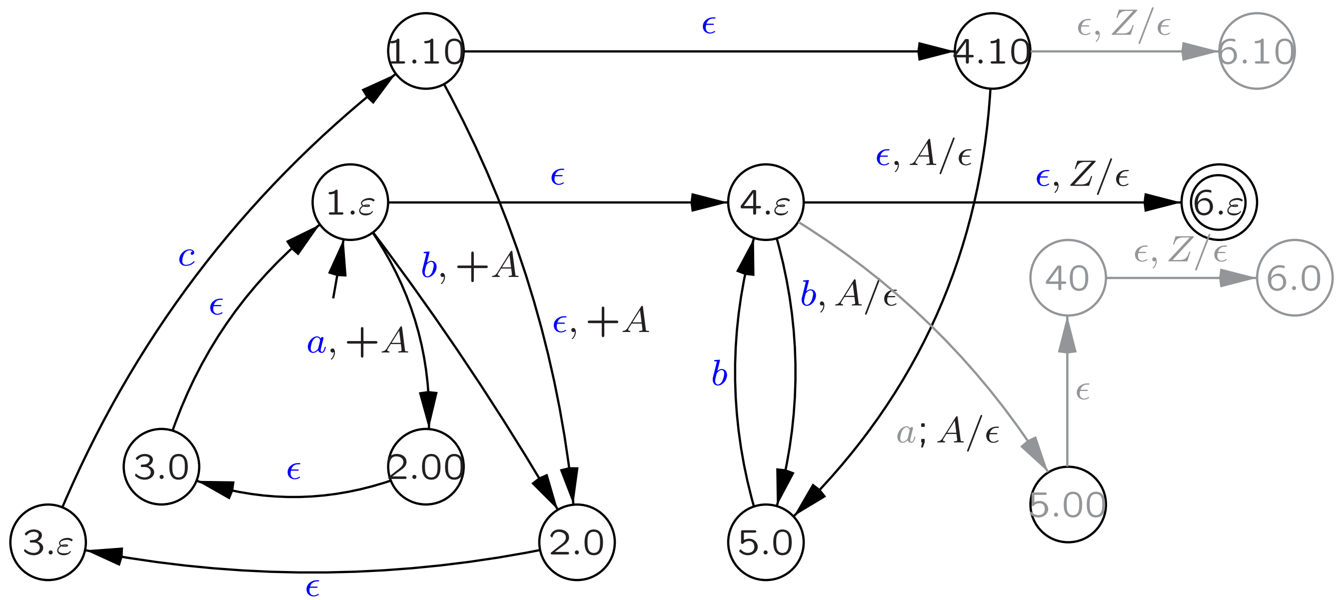
$$h : \begin{cases} a \mapsto 100 \\ b \mapsto 10 \\ c \mapsto 010 \end{cases}$$

$K = \{ \ (100)^n (10)^n \mid n \geq 0 \ \}$

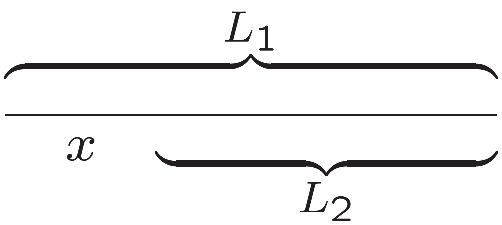




$h^{-1}(K) = \{ \ w \in \{a, b, c\}^* \mid h(w) \in K \ \}$



100	100	100	10	10	10
a	a	a	b	b	b
b	c	c	c	b	b
a	b	c	c	b	b



$L_1, L_2 \subseteq \Sigma^*$

$L_1/L_2 = \{\, x \in \Sigma^* \mid xy \in L_1 \text{ for some } y \in L_2 \,\}$

can 'hide' computations

Ex. $L_1 = \{\, a^{2n}cba^n \mid n \geq 1 \,\}\{\, ba^{2n}ba^n \mid n \geq 1 \,\}^* ba$

$L_2 = c \cdot \{\, ba^nba^n \mid n \geq 1 \,\}^*$

$L_1/L_2 = \{\, a^{2^n} \mid n \geq 1 \,\}$

Thm. CF not closed under quotient

Theorem 4.1.6

As promised, the CF languages are closed under right quotient with regular languages, since for every regular language R we can transform the FSA for R into a FST that performs the quotient by R as its function.

The next slide implements this construction. Given a PDA $\mathcal{A}$ and a FSA $\mathcal{M}$ it directly constructs the PDA for the quotient of the languages. It uses the general format for transductions from previous slides, as if the transducer for the quotient had been given. In fact, is has been implicitly derived from the FSA, by adding a single state ı, see sketch to the left for a specific example.

$L(\mathcal{A}) = L$ PDA $\mathcal{A} = (Q, \Delta, \Gamma, \delta, q_{in}, Z_{in}, F)$

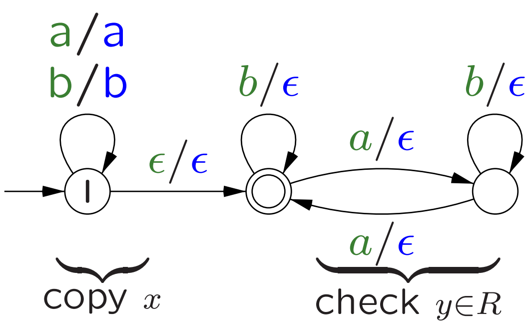
$L(\mathcal{M}) = R$ FSA $\mathcal{M} = (P, \Delta, \varepsilon, p_{in}, E)$

PDA for right quotient L/R

$\mathcal{A}' = (Q', \Delta, \Gamma, \delta', q'_{in}, Z_{in}, F')$

$Q' = Q \times (P \cup \{\mathsf{I}\})$

quotient transducer



$K/R =$
$\{ x \mid xy \in K$ and $y \in R \}$

δ' contains

$(\langle q_1, \mathsf{I} \rangle, a, A, \langle q_2, \mathsf{I} \rangle, \alpha)$ for $\delta(q_1, a, A) \ni (q_2, \alpha)$

$(\langle p, \mathsf{I} \rangle, \epsilon, A, \langle p, p_{in} \rangle)$ for $p \in P$, $A \in \Gamma$

$(\langle q_1, p \rangle, \epsilon, A, \langle q_2, p \rangle, \alpha)$
for $\delta(q_1, \epsilon, A) \ni (q_2, \alpha)$, $p \in Q$

$(\langle q_1, p_1 \rangle, \epsilon, A, \langle q_2, p_2 \rangle, \alpha)$
for $\delta(q_1, a, A) \ni (q_2, \alpha)$ & $(p_1, a, p_2) \in \varepsilon$

$q'_{in} = \langle q_{in}, \mathsf{I} \rangle$

$F' = F \times E$

Exercise 4.18

family of languages $\mathcal{L}$ is a full trio (or cone)

iff $\mathcal{L}$ is closed under morphism h,
inverse morphism h^{-1}, and
intersection with regular languages $\cap R$

iff $\mathcal{L}$ is closed under finite state transductions T

Cor. full trio closed under prefix, quotient, . . .

Thm. REG and CF are full trio's.

4.2 Unary context-free languages

$L \subseteq \{0\}^*$ $\qquad L \in \mathsf{CF}$ iff $L \in \mathsf{REG}$

pumping constant n, $m \geq n$

$z = 0^m = uvwxy$

$a_m = |uwy|$, $b_m = |vx|$

$z = 0^{a_m}\, 0^{b_m}$, $1 \leq b_m \leq n$

$M = \{m \in \mathbb{N} \mid 0^m \in L\}$

$L' = \{x \in L \mid |x| < n\}$

$L = L' \cup \bigcup_{m \in M} 0^{a_m} 0^{b_m} = L' \cup \bigcup_{m \in M} 0^{a_m} (0^{b_m})^*$

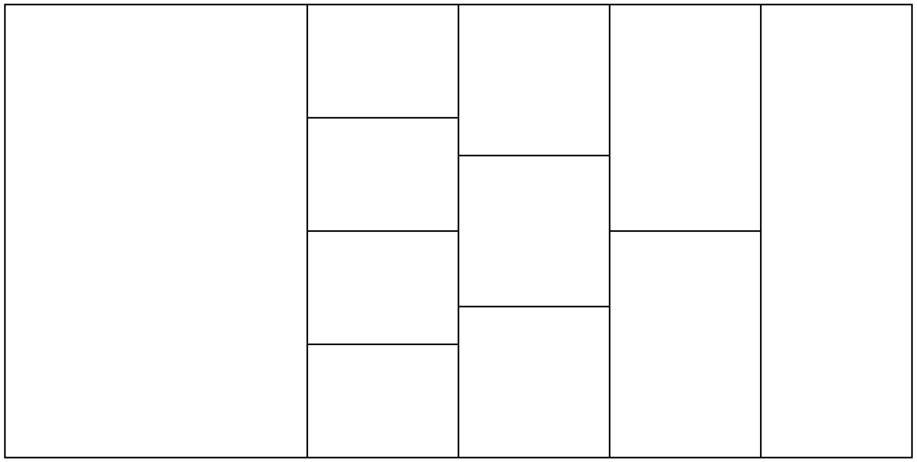
infinite union $\Rightarrow$ finite

$z = 0^{a_m} 0^{b_m}$ $\qquad b = b_m = b_{m'}$, $m < m'$, $a_m = a_{m'} \pmod b$

$z' = 0^{a'_m} 0^{b'_m}$ $\qquad 0^{a_m}(0^b)^* \supseteq 0^{a_{m'}}(0^b)^*$

$m_{ab} = \min\{\, m \in M \mid b_m = b, a_m = a \pmod b \,\}$

$L = L' \cup \bigcup_{0 \leq a < b \leq n} 0^{m_{ab}} (0^b)^*$



4.6 Parikh's theorem

$h : \Sigma \to \{0\}, \quad x \mapsto 0$

CF $\rightsquigarrow$ REG same length sets

Parikh map *commutative image*

$\psi : \Sigma^* \to \mathbb{N}^k$

$w \mapsto (|w|_{a_1}, \dots, |w|_{a_k})$

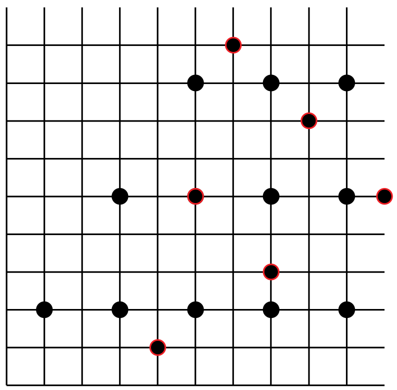
$aabaccbacca \mapsto (5, 2, 4)$

$c(ab)^*c(bc)^*c \mapsto \{ (k, k+\ell, 3+\ell) \mid k, \ell \in \mathbb{N} \} =$

$\{ (0,0,3) + k \cdot (1,1,0) + \ell \cdot (0,1,1) \mid k, \ell \in \mathbb{N} \}$

$(abc)^*$	REG
$\{ (ab)^n c^n \mid n \in \mathbb{N} \}$	LIN − REG
$\{ w \in \{ab, c\}^* \mid \#_a(w) = \#_b(w) \}$	CF − LIN
$\{ a^n b^n c^n \mid n \in \mathbb{N} \}$	CS − CF

$\mapsto \{ (n,n,n) \mid n \in \mathbb{N} \} = \{ n \cdot (1,1,1) \mid n \in \mathbb{N} \}$



linear set $\vec{u}_0, \vec{u}_1, \ldots \vec{u}_r \in \mathbb{N}^k$

$A = \{\vec{u}_0 + a_1\vec{u}_1 + \ldots + a_r\vec{u}_r \mid a_1, \ldots, a_r \in \mathbb{N}\}$

semilinear finite union

4.6.1 semilinear sets closed under
union, intersection and complement

4.6.3 X semilinear, then $X = \psi(L)$ for regular L

$\omega(\vec{u}_0) \cdot \{\, \omega(\vec{u}_1), \ldots, \omega(\vec{u}_r) \,\}^*$

$\omega : \mathbb{N}^k \to \{a_1, \ldots, a_k\}^* \qquad \psi(\omega(\vec{u})) = \vec{u}$

4.6.5 $\psi(L)$ semilinear for CFL L

Lemma 4.6.4

G Chomsky normal form
k variables $p = 2^{k+1}$
$z \in L(G)$, $|z| \geq p^j$

$S \Rightarrow^*$
$uAy \Rightarrow^*$
$uv_1Ax_1y \Rightarrow^*$
$uv_1v_2Ax_2x_1y \Rightarrow^*$
$\ldots \qquad \Rightarrow^*$
$uv_1v_2 \ldots v_jAx_j \ldots x_2x_1y \Rightarrow^*$
$uv_1v_2 \ldots v_jwx_j \ldots x_2x_1y = z$

$v_ix_i \neq \epsilon$
$|uv_1v_2 \ldots v_jx_j \ldots x_2x_1y| \leq p^j$

Theorem 4.6.5

$\psi(L)$ semilinear for CFL L

$L_U \subseteq L$
derivation with variables U
$L = \bigcup_{S \subseteq U \subseteq V} L_U$

$\ell = |U|$
$E = \{ w \in L_U \mid |w| < p^\ell \}$ $\quad S \Rightarrow^* w$
$F = \{ vx \mid 1 \leq |vx| \leq p^\ell,$
$A \Rightarrow^* vAx$ for some $A \in U \}$

$\psi(L_U) = \psi(EF^*)$

"$\subseteq$" induction on $|z|$, $z \in L_U$

"$\supseteq$" induction on t,
$z = e_0f_1 \ldots f_t \in EF^*$

Ex. $L = \{ a^i b^j \mid j \neq i^2 \}$ not in CF

$\psi(L) = \{ (i,j) \mid j \neq i^2 \}$ not semilinear
complement $\{ (i, i^2) \mid i \in \mathbb{N} \}$
corresponding regular language?
lengths $\{ i^2 + i \mid i \in \mathbb{N} \}$ cannot be pumped

4.3 Ogden's lemma
4.4 Applications of Ogden's lemma

long words can be pumped

$\forall$ for every CF language L

$\exists$ there exists a constant $n \geq 1$

such that

$\forall$ for every $z \in L$

with $|z| \geq n$

$\exists$ there exists a decomposition $z = uvwxy$

with $|vwx| \leq n$, $|vx| \geq 1$

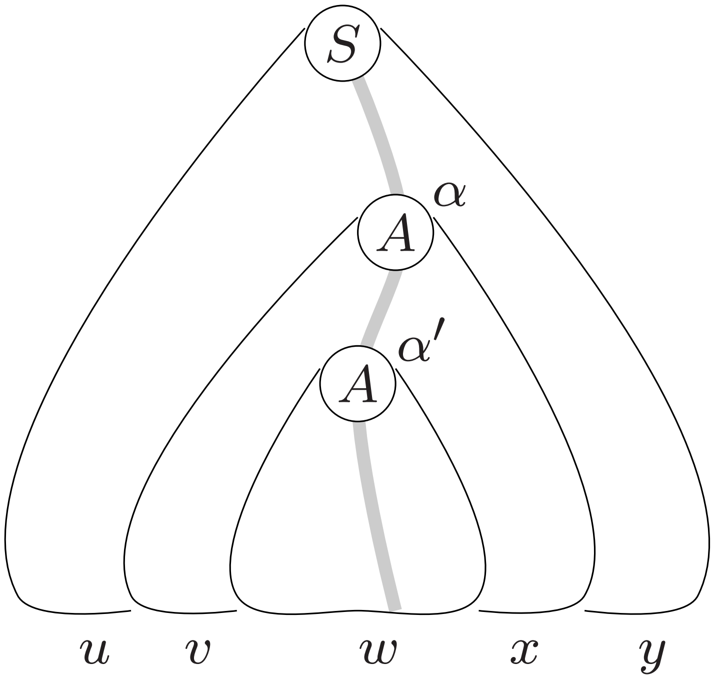
such that

$\forall$ for all $i \geq 0$, $uv^iwx^iy \in L$

$\|x\|$ marked symbols in x

$\forall$ for every CF language L
$\exists$ there exists a constant $n \geq 1$
such that
$\forall$ for every $z \in L$
with $\|z\| \geq n$
$\exists$ there exists a decomposition $z = uvwxy$
with $\|vwx\| \leq n$, $\|vx\| \geq 1$
such that
$\forall$ for all $i \geq 0$, $uv^iwx^iy \in L$

Lemma 4.3.1



$S \Rightarrow^* uAy$
$A \Rightarrow^* vAx$
$A \Rightarrow^* w$

$uv^iwx^iy \in L$

$G = (V, \Sigma, P, S)$

$k = |V|$ $d = \max\{|\alpha| \mid A \to \alpha \in P\}$

branch point: ≥ 2 children with marked descendants

if each path has $\leq \ell$ branch points, then $\leq d^\ell$ marked letters

pumping constant $n = d^{k+1} > d^k$

$\exists$ path with $> k$ branch points

take path with most branch points

α, α' same label A, as low as possible

$\|vx\| \geq 1$ α branch point
$\|vwx\| \leq n$ no repetition below α
$\|w\| \geq 1$ α' branch point

$L = \{ a^i b^j c^k \mid$
$i = j$ or $j = k$ but not both $\}$
not context-free

n as Ogden, assume ≥ 3

$z = \underline{a^n}\, b^n\, c^{n+n!}$

$z = uvwxy$

v, x each cannot have different symbols else $uv^2wx^2y \notin a^*b^*c^*$

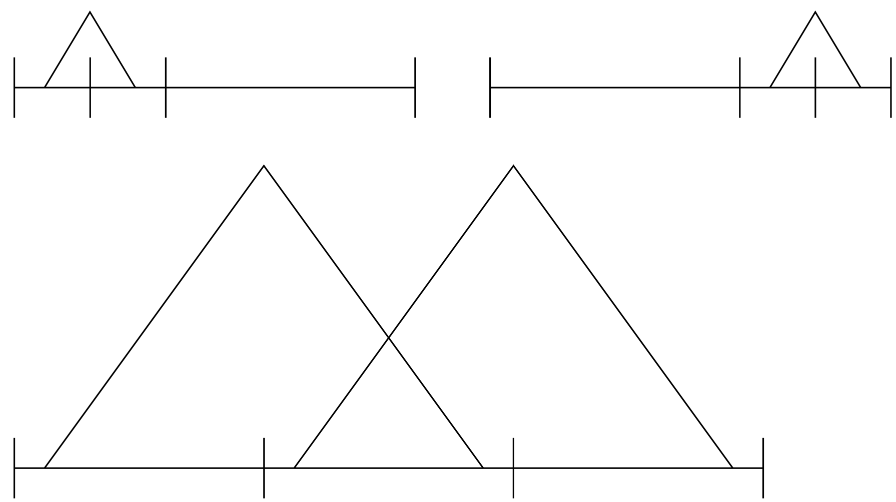
possibilities

- $vx = a^k$
  $uv^0wx^0y = a^{n-k}b^nc^{n+n!} \notin L$
- $v = a^k$, $x = b^\ell$ $(k \neq \ell)$
  $uv^0wx^0y = a^{n-k}b^{n-\ell}c^{n+n!} \notin L$
- $v = a^k$, $x = b^\ell$ $(k = \ell)$
  consider $i = \frac{n!}{\ell} + 1$
  add $i - 1$ copies of ℓ a's
  $uv^iwx^iy = a^{n+n!}b^{n+n!}c^{n+n!} \notin L$
- $v = a^k$, $x = c^\ell$
  $uv^2wx^2y = a^{n+k}b^nc^{n+n!+\ell} \notin L$

grammar ambiguous
language inherently ambiguous

$z = \underline{a^n}\, b^n\, c^{n+n!}$ $\qquad$ $z' = a^{n+n!}\, b^n\, \underline{c^n}$

$L = \{\, a^i b^j c^k \mid i = j$ or $j = k \,\}$.
is inherently ambiguous



see example 4.3.2

$$[p, A, q] \Rightarrow^*_G w \iff (p, w, A) \vdash^*_{\mathcal{M}} (q, \epsilon, \epsilon)$$

Thm. PDA $\mathcal{M}$ with n states and p stack symbols each CFG for $L_e(\mathcal{M})$ has at least $n^2 p$ variables

4.5 The interchange lemma

$\forall$ for every CF language L

$\exists$ there exists constant $c > 0$

$\forall$ such that for all $n \geq m \geq 2$,

all subsets $R \subseteq L \cap \Sigma^n$

$\exists$ there exists $Z = \{z_1, z_2, \ldots, z_k\} \subseteq R$,

with $k \geq \frac{|R|}{c(n+1)^2}$

and compositions $z_i = w_i x_i y_i$

such that

(a) $|w_1| = |w_2| = \ldots = |w_k|$

(b) $|y_1| = |y_2| = \ldots = |y_k|$

(c) $\frac{m}{2} < |x_1| = |x_2| = \ldots = |x_k| \leq m$

(d) $w_i x_j y_i \in L$ for all $1 \leq i, j \leq k$

Lem. G CFG in Chomsky normal form for L, $m \geq 2$
$z \in L$, $|z| \geq m$, then $S \Rightarrow^* wAy \Rightarrow^* wxy = z$
with $\frac{m}{2} < |x| \leq m$

$z \rightsquigarrow (n_1, A, n_2)$ where $n_1 = |w|$, $n_2 = |z|$

Chapter 2 Thue-Morse sequence

t_n number of 1's in base-2 expansion of n

or iterate $0 \mapsto 01$, $1 \mapsto 10$

$0 \cdot 1 \cdot 10 \cdot 1001 \cdot 10010110 \cdot 1001011001101001 \ldots$

overlapfree *no* $axaxa$ $(a \in \Sigma_2,\ x \in \Sigma_2^*)$

$00 \mapsto 1$, $01 \mapsto 2$, $10 \mapsto 0$, $11 \mapsto 1$ 'sliding'

$210201210120210201202101210201 2\ldots$

squarefree *no* xx $(x \in \Sigma_3^*)$

Theorem 2.5.2

$\Sigma = \{0, 1, \ldots, i-1\}$ $\quad L_i = \{\, xyyz \mid x, y, z \in \Sigma^*, y \neq \epsilon \,\}$

Thm. L_6 not in CF

[see Chapter 2] r squarefree string of length $\frac{n}{4}-1$ over $\{0, 1, 2\}$

$A_n = \{3r3r \amalg s \mid s \in \{4, 5\}^{n/2}\}$

$\amalg$ perfect shuffle (alternate strings)

$z \in A_n$ contains a square iff it *is* a square

$B_n = L_6 \cap A_n = \{3r3r \amalg ss \mid s \in \{4, 5\}^{n/4}\}$

$|B_n| = 2^{\frac{n}{4}}$ $\quad$ choose $m = n/2$

[take n large] $Z = \{z_1, z_2, \ldots, z_k\}$ $\quad k \geq \frac{2^{n/4}}{c(n+1)^2} > 2^{n/8}$

$z_i = w_i x_i y_i$, $\frac{m}{2} < |x_i| \leq m$ (etc.)

$w_i x_j y_i \in B_n$ hence $x_i = x_j$

(x_i fixed by other symbols in z_i)

hence $\frac{n}{4}$ symbols fixed for Z, $\frac{n}{8}$ in $\{4, 5\}$

at most $\frac{n}{8}$ free, $|Z| \leq 2^{n/8}$

contradiction

4.7 Deterministic context-free languages

what we learn about

deterministic context-free languages

- is an *automaton* notion

- less powerful than CF

- closed under complement (nontrivial)

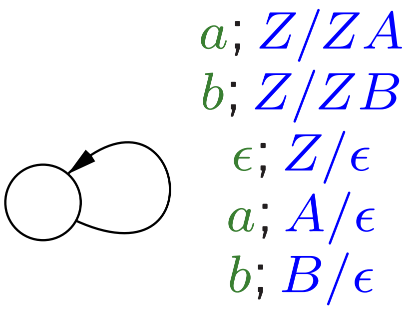
- see also Chapter 5 on parsing

	RLIN REG	DPDA	CF PDAe	DLBA	MON LBA	REC	TYPE0 RE
intersection	+	−	−	+	+	+	+
complement	+	+	−	+	+	+	−
union	+	−	+	+	+	+	+
concatenation	+	−	+	+	+	+	+
star, plus	+	−	+	+	+	+	+
ϵ-free morphism	+	−	+	+	+	+	+
morphism	+	−	+	−	−	−	+
inverse morphism	+	+	+	+	+	+	+
intersect reg lang	+	+	+	+	+	+	+
mirror	+	−	+	+	+	+	+
	fAFL		fAFL	AFL	AFL	AFL	fAFL

$\cap \; ^c \; \cup$ boolean operations

$\cup \; \cdot \; ^*$ regular operations

$h \; h^{-1} \; \cap R$ (full) trio operations



$Z \to aZA$
$Z \to bZB$
$Z \to \epsilon$
$A \to a$
$B \to b$

$P = \{\ ww^R \mid w \in \{a,b\}^*\ \}$ guessing the middle

$(aabbaa, Z) \vdash (aabbaa, \epsilon) \not\vdash$

$\top$

$(abbaa, ZA) \vdash (abbaa, A) \vdash (bbaa, \epsilon) \not\vdash$

$\top$

$(bbaa, ZAA) \vdash (bbaa, AA) \not\vdash$

$\top$

$(baa, ZBAA) \vdash (baa, BAA) \vdash (aa, AA) \vdash (a, A) \vdash (\epsilon, \epsilon)$ ok.

$\top$

$(aa, ZBBAA) \vdash (aa, BBAA) \not\vdash$

$\top$

$(a, ZABBAA) \vdash (a, ABBAA) \vdash (\epsilon, BBAA) \not\vdash$

$\top$

$(\epsilon, ZAABBAA) \vdash (\epsilon, ABBAA) \not\vdash$

also $\{\ a^n b^n \ \mid n \in \mathbb{N}\ \} \cup \{\ a^n b^\ell c^n \ \mid \ell, n \in \mathbb{N}\ \}$

Determinism means the automaton has no choice: at each moment it can take at most one step to continue its computation. To translate this intuition to a restriction on the instructions for PDA is nontrivial, as the next step is determined both by input letter and by topmost stack symbol. Additionally this is complicated by the choice between reading an input letter and following a λ-instruction.

We quote from our chapter:

The PDA $\mathcal{A} = (Q, \Delta, \Gamma, \delta, q_{in}, A_{in}, F)$ is *deterministic* if

- for each $p \in Q$, each $a \in \Delta$, and each $A \in \Gamma$, δ does not contain both an instruction $(p, \lambda, A, q, \alpha)$ and an instruction (p, a, A, q', α').

- for each $p \in Q$, each $a \in \Delta \cup \{\lambda\}$, and each $A \in \Gamma$, there is at most one instruction (p, a, A, q, α) in δ.

determinism means 'no choice'

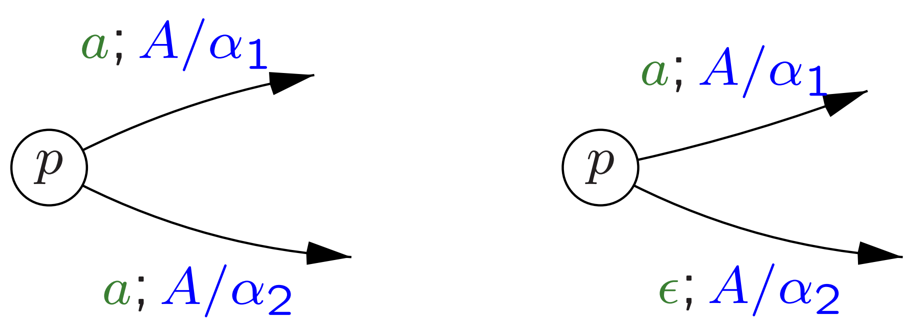
... where to start (ok)

... between two actions

with same *tape & stack* symbols

... between letter or ϵ

<u>not</u> allowed



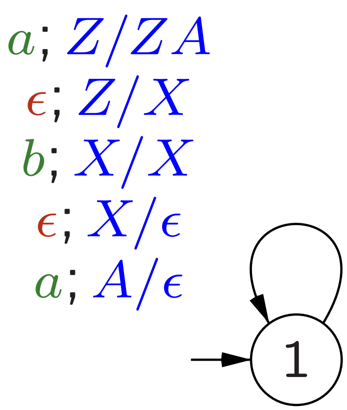
$(p, a, A) \ni (q_1, \alpha_1)$ $\quad$ $(p, a, A) \ni (q_1, \alpha_1)$

$(p, a, A) \ni (q_2, \alpha_2)$ $\quad$ $(p, \epsilon, A) \ni (q_2, \alpha_2)$

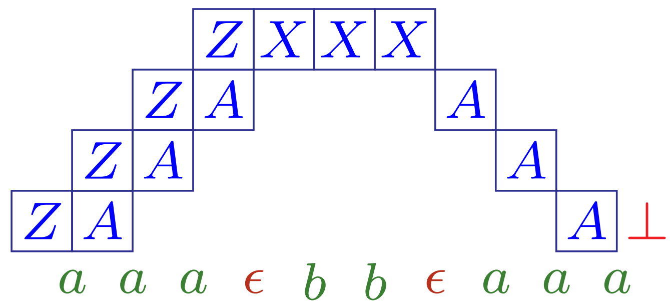
FSA	=	DFSA	=	RLIN
PDAe	=	PDA	=	CF
DPDAe	$\subset$	DPDA	$\subset$	CF

final state: deterministic CF languages

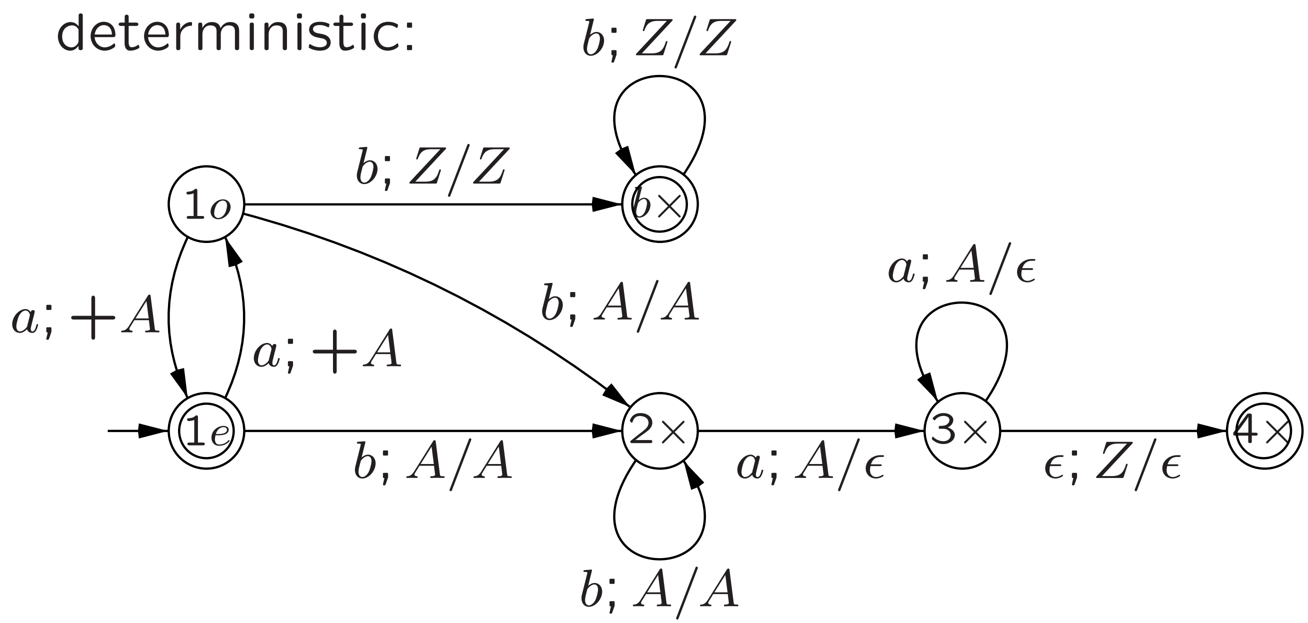
'context-free' but uses automata



$\{ a^n b^m a^n \mid m, n \in \mathbb{N} \}$

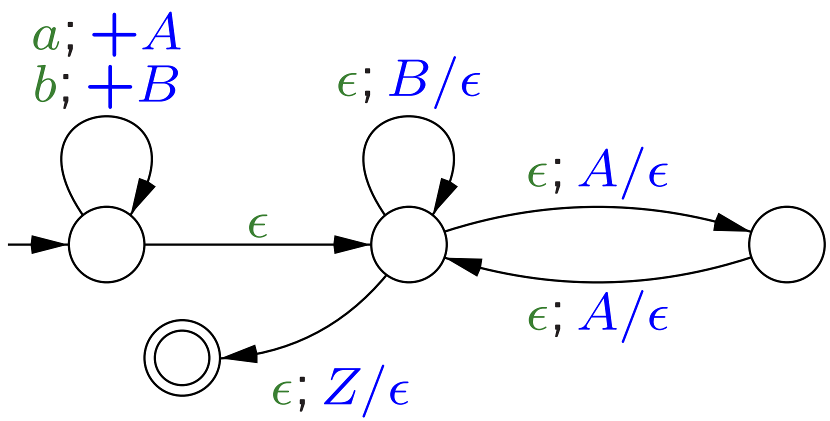


deterministic:



closure under complement $F \leftrightarrow Q - F$

- $\star$ completely read input
  - $*$ input+stack may block
  - $*$ infinite ϵ-computations!
- $\star$ computations without reading
  - $*$ accept afterwards



$\{A, B, Z\}$, initial Z

Lem. equivalent PDA that always scans entire input

$(q_0, w, Z_0) \vdash^* (q, \epsilon, \alpha) \quad q \in Q, \ \alpha \in \Gamma^*$

$\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

$Q' = Q \cup \{d, f\}, \ \Gamma' = \Gamma \cup \{X_0\}, \ F' = F \cup \{f\},$

'dead' states

$\delta'(d, a, X) = \{(d, X)\}$

$\delta'(f, a, X) = \{(d, X)\}$ for $a \in \Sigma$ and $X \in \Gamma'$

avoid empty stack

add 'bottom' X_0

$\delta'(q_0', \epsilon, X_0) = \{(q_0, Z_0 X_0)\}$

$\delta'(q, a, X_0) = \{(d, X_0)\}$ for $q \in Q$ and $a \in \Sigma$

undefined transitions

$\delta'(q, a, X_0) = \{(d, X_0)\}$

when $\delta(q, a, X) = \varnothing$ and $\delta(q, \epsilon, X) = \varnothing$

infinite loops*

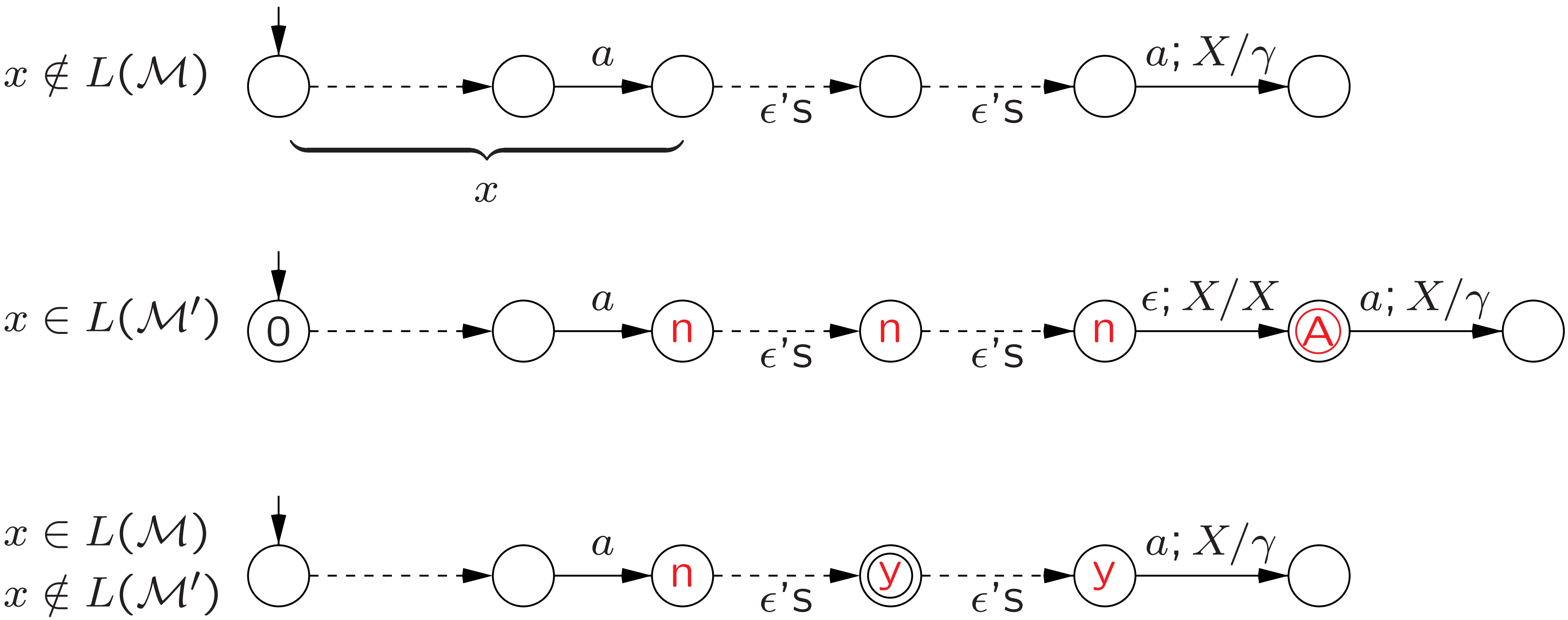
when $\mathcal{M}$ enters infinite ϵ-loop on (q, ϵ, X)

$\delta'(q, \epsilon, X) = \{(d, X)\}$ without final states

$\delta'(q, \epsilon, X) = \{(f, X)\}$ with final state

* "The actual implementation is a bit complex"

Lemma 4.7.1


x ∉ L(M)
a
ϵ's
ϵ's
a; X/γ
x
x ∈ L(M′)
0
a
n
ϵ's
n
ϵ's
n
ϵ; X/X
A
a; X/γ
x ∈ L(M)
x ∉ L(M′)
a
n
ϵ's
y
ϵ's
y
a; X/γ

Thm. DCFL (= DPDA) closed under complement

$\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

$Q' = Q \times \{n, y, A\}, \; F' = Q \times \{A\}$

$q_0' = [q_0, y]$ if $q_0 \in F$, $q_0' = [q_0, n]$ otherwise

$\delta(q, a, X) = (p, \gamma)$	$\delta'([q, y], a, X) = ([p, y], \gamma)$	$p \in F$
$(a \in \Sigma)$	$\delta'([q, y], a, X) = ([p, n], \gamma)$	$p \notin F$
	$\delta'([q, n], \epsilon, X) = ([q, A], X)$	
	$\delta'([q, A], a, X) = ([p, y], \gamma)$	$p \in F$
	$\delta'([q, A], a, X) = ([p, n], \gamma)$	$p \notin F$
$\delta(q, \epsilon, X) = (p, \gamma)$	$\delta'([q, y], \epsilon, X) = ([p, y], \gamma)$	
	$\delta'([q, n], \epsilon, X) = ([p, y], \gamma)$	$p \in F$
	$\delta'([q, n], \epsilon, X) = ([p, n], \gamma)$	$p \notin F$

Ex. $\{ w \in \{a, b\}^* \mid w \neq xx \}$ not in DCFL

Thm. L DCFL

at least one Myhill-Nerode class is infinite

$x \in \Sigma^* \rightsquigarrow x', q, A\alpha$

after processing xx' stack height $|A\alpha|$ minimal

$(q_0, xx', Z_0) \vdash^* (q, \epsilon, A\alpha)$

any continuation independent of α

infinitely many xx' end in same minimal q, A

infinitely many xx' all in L or all in $\Sigma^* - L$

have the same 'extensions'

$(q_0, xx'z, Z_0) \vdash^* (q, z, A\alpha) \vdash^* (p, \epsilon, \gamma\alpha)$ $(p \in F)$

iff $(q_0, x_1 x_1' z, Z_0) \vdash^* (q, z, A\alpha_1) \vdash^* (p, \epsilon, \gamma\alpha_1)$

Cor. PAL $= \{\, x \in \{a, b\}^* \mid x = x^R \,\}$ not in DCFL

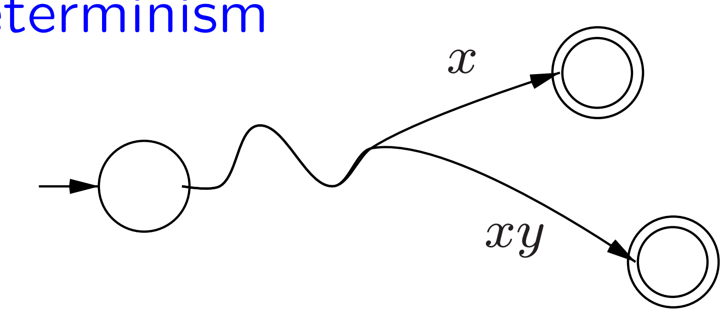
(exercise) no strings equivalent

▷

Consider a language that both includes string x and an extension xy of it. Nondeterministic automata may have quite different accepting computations on both strings. For deterministic automata we know that the computation that accepts xy must start with the accepting computation on x. ◁

language L $\quad x \in L$, $xy \in L$

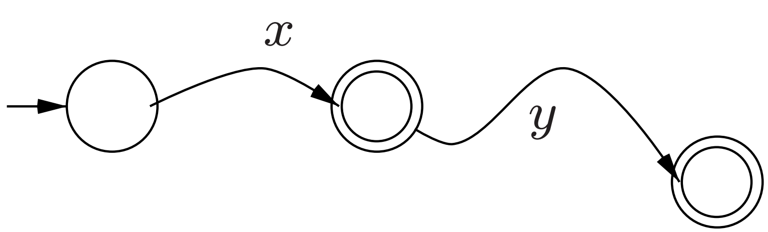
∗ nondeterminism



$\underline{a^n}\ \underline{b^n}$

$\underline{a^n}\ b^m\ \underline{c^n}$ $\quad$ different behaviour on b's

∗ determinism



computation on xy and on x must coincide!

apply this to:

haspref(L) $= \{\, xy \mid \underline{x} \in L, \underline{x}y \in L, y \neq \epsilon \,\}$

In order to rigorously show that $\mathsf{DPDA} \subset \mathsf{PDA} = \mathsf{CF}$ we define a 'strange operation' haspref. We show that DPDA and CF behave differently with respect to this operator. See properties on the slide.

This part of the slides was used for another lecture (where closure under complement was not proved).

haspref$(L) = \{ xy \mid x \in L, xy \in L, y \neq \epsilon \}$

$L_0 = \{ a^n b^n \mid n \geq 1 \} \cup \{ a^n b^m c^n \mid m, n \geq 1 \}$

haspref$(L_0) = \{ a^n b^m c^n \mid m \geq n \geq 1 \} \notin$ CF

- CF = PDA is not closed under haspref
- DPDA *is* closed under haspref

[proof follows]

consequences

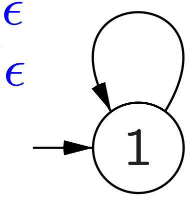
- DPDA ⊂ PDA = CF $L_0 \in$ CF − DPDA
- DPDA is not closed under union
- also $\{ ww^R \mid w \in \{a, b\}^* \} \notin$ DPDA

Geraud Senizergues (2001) proved that the equivalence problem for deterministic PDA (i.e. given two deterministic PDA A and B, is $L(A) = L(B)$?) is decidable.

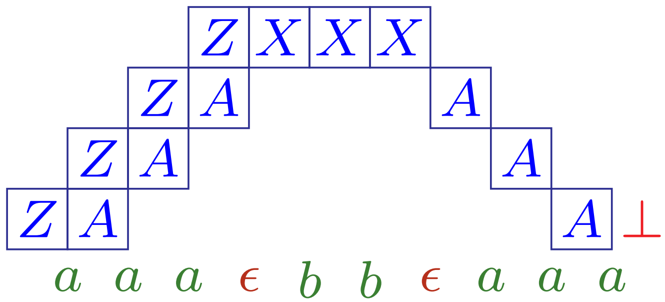
For nondeterministic PDA, equivalence is undecidable.

4.8 Linear languages

$a;\ Z/ZA$
$\epsilon;\ Z/X$
$b;\ X/X$
$\epsilon;\ X/\epsilon$
$a;\ A/\epsilon$



$\{\ a^n b^m a^n \mid m, n \in \mathbb{N}\ \}$



$Z \to aZa$

$Z \to X$

$X \to bX$

$X \to \epsilon$

linear grammar: rhs at most one variable

$A \to \alpha B \beta,\ X \to \alpha$

$A, B \in V,\ \alpha.\beta \in \Sigma^*$

$\{\ a^n b^n \mid n \in \mathbb{N}\ \}$

$\{\ a^n b^n c^m \mid m, n \in \mathbb{N}\ \}$

$\{\ a^n b^n a^m b^m \mid m, n \in \mathbb{N}\ \}$ not LIN, *why?*

long words can be pumped

$\forall$ for every LIN language L
$\exists$ there exists a constant $n \geq 1$
such that
$\forall$ for every $z \in L$
with $|z| \geq n$
$\exists$ there exists a decomposition $z = uvwxy$
with $|uvxy| \leq n$, $|vx| \geq 1$
such that
$\forall$ for all $i \geq 0$, $uv^iwx^iy \in L$

context-free	(((((())())()))(())
linear	(((((())))))
example	((()))(((())))

$L = \{\, a^i b^i c^j d^j \mid i, j \geq 0 \,\}$ in CFL − LIN

$z = a^n b^n c^n d^n$

$|uvxy| \leq n$

v and x each consist of a's or d's

$v = a^k$, $x = d^\ell$, $k + \ell \geq 1$

$uv^0wx^0y = a^{n-k} b^n c^n d^{n-\ell} \notin L$

and two other possibilities

$\{\, x \in \{a,b\}^* \mid x = x^R \,\}$ in LIN - DCF

$\{\, a^i b^i c^j d^j \mid i,j \geq 0 \,\}$ in DCF - LIN

$\{ a^i b^i c^j d^j \mid i,j \geq 0 \}$ in CFL − LIN

$= \{ a^i b^i \mid i \geq 0\} \cdot \{c^j d^j \mid j \geq 0 \}$ in LIN·LIN

not closed under concatenation

$= \{ a^i b^i \mid i \geq 0\} \cdot c^* d^* \cap a^* b^* \cdot \{c^j d^j \mid j \geq 0 \}$

not closed under intersection

closed under finite state transductions:
(inverse) morphism, intersection regular
use machine model $\longrightarrow$

not closed under star

$$T(\{ a^i b^i \mid i \geq 0\}^*) = \{ a^i b^i c^j d^j \mid i,j \geq 0 \}$$

▷

As we have seen, both the context-free and the reguar languages have characterizations using grammars as well as using automata.

Here we show the same holds for the linear languages, they are accepted by one-turn push-down automata, where the stack behaviour consists of two phases, the first one adding to the stack, the second one popping.

This cannot be directly derived from the classical PDA to CFG triplet construction, as this will not generally yield a linear grammar when one starts with a one-turn pushdown.

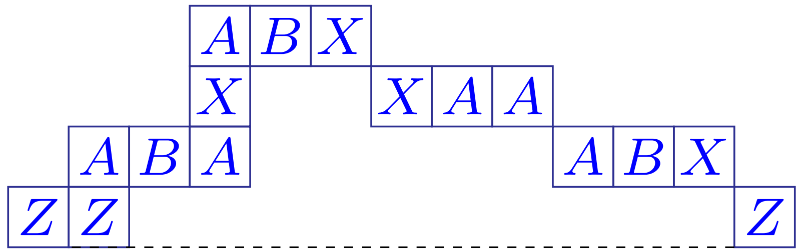
◁

RLIN = FSA

LIN = 1tPD

CF = PD

one-turn pushdown automata



$Q = Q^+ \cup Q^-$, $q_{in} \in Q^+$

$(p, a, A, q, \alpha) \in \delta$ then $\begin{cases} p, q \in Q^+ \text{ and } |\alpha| \geq 1, \text{ or} \\ p \in Q,\ q \in Q^- \text{ and } |\alpha| \leq 1 \end{cases}$

standard construction:

$(p, a, A, q, BC) \in \delta$ then

$$[p, A, r] \to a[q, B, s][s, C, r]$$

not linear

$$[p, A, q] \Rightarrow_G^* w \iff (p, w, A) \vdash^* (q, \epsilon, \epsilon)$$

here $q \in Q^-$

*

$$\delta(p, a, A) \ni (q_1, B_1 \cdots B_n)$$

$$[p, A, q] \to a\,[q_1, B_1, q_2]\ \underbrace{[q_2, B_2, q_3] \cdots [q_n, B_n, q]}_{\text{generate regular languages}}$$

$$p, q_1 \in Q, q, q_2, \ldots, q_r \in Q^-$$

$$B_1, \ldots, B_r \in \Gamma \ (1 \le r \le \text{max-rhs})$$

$p \in Q^-$ if $(q, \alpha) \in \delta(p, a, A)$ then $q \in Q^-$, $|\alpha| \le 1$

$[p, A, r] \to a[q, B, r]$ $\qquad \delta(p, a, A) \ni (q, B)$

$[p, A, q] \to a$ $\qquad \delta(p, a, A) \ni (q, \epsilon)$

include this information in $[q_1, B_1, q_2]$

generate regular language(s) to the right

backwards! (left-linear grammar)

then next step pushdown

*

LIN / LIN = RE

later perhaps, Chapter 6

LIN *not* closed under quotient

extra exercise

7. Is the class of CFLs closed under the shuffle operation shuff $\|$ (introduced in Section 3.3)? How about perfect shuffle $\amalg$?

not context-free
$\{ ww \mid w \in \Sigma^* \}$
$\{ a^n b^n c^n \mid n \geq 0 \}$
$\{ a^n b^m a^n b^m \mid n, m \geq 0 \}$

intersect shuffle with regular language

15. Let $G = (V, \Sigma, P, S)$ be a context-free grammar.

(a) Prove that the language of all sentential forms derivable from S is context-free.

(b) Prove that the language consisting of all sentential forms derivable by a leftmost derivation from S is context-free.

variables V become terminals
simulated by 'new' variables

leftmost derivations are precisely simulated when constructing PDA for CFG transparencies made for

Second Course in Formal Languages and Automata Theory

based on the book by Jeffrey Shallit
of the same title

Hendrik Jan Hoogeboom, Leiden
http://www.liacs.nl/~hoogeboo/second/