4.3. General Phrase-Structure Grammars

A phrase-structure grammar is a 4-tuple $G = (N, T, S, P)$, where $P \subseteq (N \cup T)^* \times (N \cup T)^*$ is a finite semi-Thue system on $N \cup T$ such that $|\ell|_N \geq 1$ for all $(\ell \to r) \in P$ (see Section 2).

Theorem 4.20

For each phrase-structure grammar G, the language $L(G)$ is recursively enumerable.

Proof.

For $G = (N, T, S, P)$ we describe a 2-tape-NTM M that accepts the language $L(G)$: The input w is stored on tape 1, and on tape 2, M guesses a derivation of G, starting from S.
As soon as a terminal word z has been generated on tape 2, M checks whether $z = w$ holds. In the affirmative, M halts, while in the negative, M enters an infinite loop. Hence, $L(M) = L(G)$. □

Theorem 4.21

If L is a recursively enumerable language, then there exists a phrase-structure grammar G such that $L(G) = L$.

Proof.

Let $L \subseteq \Sigma^*$ be r.e., and let $M = (Q, \Sigma, \Gamma, \Box, \delta, q_0, q_1)$ be a 1-TM s.t. $L(M) = L$. We construct a grammer $G = (N, \Sigma, A_1, P)$ by taking

$$N := ((\Sigma \cup \{\varepsilon\}) \times \Gamma) \cup Q \cup \{A_1, A_2, A_3\},$$

and defining P as follows:

(1) $A_1 \to q_0 A_2$,

(2) $A_2 \to [a, a]A_2$ for all $a \in \Sigma$,

(3) $A_2 \to A_3$,

(4) $A_3 \to [\varepsilon, \Box]A_3$,

(5) $A_3 \to \varepsilon$.

Using these productions G generates a sentential form

$$q_0[a_1, a_1][a_2, a_2] \cdots [a_n, a_n][\varepsilon, \Box] \cdots [\varepsilon, \Box]$$

for a word $w = a_1 a_2 \cdots a_n \in \Sigma^*$. This sentential form encodes the initial configuration of the TM M on input w.

Proof of Theorem 4.21 (cont.)

Further, P contains the following productions for simulating M:

(6) $q[a, X] \to [a, Y]p$, if $\delta(q, X) = (p, Y, R)$,

(7) $[b, Z]q[a, X] \to p[b, Z][a, Y]$ for all $b \in \Sigma \cup \{\varepsilon\}$ and $Z \in \Gamma$, if $\delta(q, X) = (p, Y, L)$,

(8) $q[a, X] \to p[a, Y]$, if $\delta(q, X) = (p, Y, 0)$.

If $q_0 w \Box^m \vdash_M^* upv$ holds for some $u, v \in \Gamma^*$, $p \in Q$, and $m \geq 0$, then

$$q_0[a_1, a_1][a_2, a_2] \cdots [a_n, a_n][\varepsilon, \Box]^m \to_P^*$$

$$[a_1, u_1] \cdots [a_k, u_k]p[a_{k+1}, v_1] \cdots [a_n, v_{n-k}][\varepsilon, v_{n-k+1}] \cdots [\varepsilon, v_{n+m-k}],$$

where $u = u_1 u_2 \ldots u_k$ and $v = v_1 v_2 \ldots v_{n+m-k}$.

Proof of Theorem 4.21 (cont.)

Finally, P also contains some productions for treating halting configurations:

$$(9)\qquad \left.\begin{array}{lcl} [a,X]q_1 & \to & q_1 a q_1 \\ q_1[a,X] & \to & q_1 a q_1 \\ q_1 & \to & \varepsilon \end{array}\right\} \text{ for all } a \in \Sigma \cup \{\varepsilon\} \text{ and } X \in \Gamma.$$

If $q_0 w \vdash_M^* u q_1 v$, that is, if $w \in L(M)$, then:

$$\begin{array}{lcl} A_1 & \to^* & q_0[a_1,a_1][a_2,a_2]\cdots[a_n,a_n][\varepsilon,\Box]^m \\ & \to^* & [a_1,u_1]\cdots[a_k,u_k]q_1[a_{k+1},v_1]\cdots[a_n,v_{n-k}]\cdots[\varepsilon,v_{n+m-k}] \\ & \to^* & a_1 a_2 \cdots a_n = w, \end{array}$$

which shows that $L(M) \subseteq L(G)$.

Conversely, if $w \in L(G)$, then we see from the construction above that M halts on input w, which implies that $L(G) = L(M)$. $\square$

Corollary 4.22

A language L is recursively enumerable if and only if it is generated by a phrase-structure grammar.

From the various characterizations of the class of r.e. languages, we obtain the following closure properties.

Corollary 4.23

(a) *The class RE is closed under union, intersection, product, Kleene star, reversal, and inverse morphisms.*

(b) *The class RE is not closed under complementation.*

4.4. Context-Sensitive Languages

A grammar $G = (N, T, S, P)$ is context-sensitive if each production $(\ell \to r) \in P$ has the form $\alpha X \beta \to \alpha u \beta$, where $X \in N$, $\alpha, \beta, u \in (N \cup T)^*$, and $u \neq \varepsilon$. In addition, G may contain the production $(S \to \varepsilon)$, if S does not occur on the right-hand side of any production.

A language L is called context-sensitive if there exists a context-sensitive grammar G such that $L = L(G)$.

By CSL(Σ) we denote the set of all context-sensitive languages on Σ, and CSL denotes the class of all context-sensitive languages.

A grammar $G = (N, T, S, P)$ is called monotone if $|\ell| \leq |r|$ holds for each production $(\ell \to r) \in P$. Also a monotone grammar may contain the production $(S \to \varepsilon)$ if S does not occur on the right-hand side of any production.

Theorem 4.24

(a) *For each context-sensitive grammar G, there is a monotone grammar G' such that $L(G') = L(G)$*

(b) *For each monotone grammar G', there is a context-sensitive grammar G such that $L(G) = L(G')$.*

Proof.

(a) $\Rightarrow$ (b): By definition each context-sensitive grammar is monotone.

(b) $\Rightarrow$ (a): Let $G' = (N', T, S', P')$ be a monotone grammar.
From G' we construct $G'' = (N'', T, S', P'')$ by taking

$$N'' := N' \cup \{ A_a \mid a \in T \} \text{ and } P'' := h(P') \cup \{ A_a \to a \mid a \in T \},$$

where h is defined through $A \mapsto A$ $(A \in N')$ and $a \mapsto A_a$ $(a \in T)$.
G'' is monotone, and all the new productions are context-sensitive.
It remains to replace the productions in $h(P')$ by context-sensitive productions.

Proof of Theorem 4.24 (cont.)

Let $A_1 \cdots A_m \to B_1 \cdots B_n$ $(2 \leq m \leq n)$ be a production from $h(P')$, where $A_i, B_j \in N''$. We introduce new nonterminals $Z_1, Z_2, \ldots, Z_m$ and replace the above production by the following ones:

$$\begin{array}{rcl}
A_1 \cdots A_m & \to & Z_1 A_2 \cdots A_m \\
Z_1 A_2 \cdots A_m & \to & Z_1 Z_2 A_3 \cdots A_m \\
 & \vdots & \\
Z_1 Z_2 \cdots Z_{m-1} A_m & \to & Z_1 \cdots Z_m B_{m+1} \cdots B_n \\
Z_1 \cdots Z_m B_{m+1} \cdots B_n & \to & B_1 Z_2 \cdots Z_m B_{m+1} \cdots B_n \\
 & \vdots & \\
B_1 \cdots B_{m-1} Z_m B_{m+1} \cdots B_n & \to & B_1 \cdots B_{m-1} B_m B_{m+1} \cdots B_n.
\end{array}$$

The new productions are context-sensitive. It's obvious that they simulate the old production. On the other hand, the new productions can only be used for this purpose, as the new nonterminals $Z_1, Z_2, \ldots, Z_m$ do not occur in any other production. By repeating the process above for each production from $h(P')$, we obtain a context-sensitive grammar G such that $L(G) = L(G')$ □

Example.

Let G be the following monotone grammar:

$$G = (\{S, B\}, \{a, b, c\}, S, \{S \to aSBc, S \to abc, cB \to Bc, bB \to bb\}).$$

We claim that $L(G) = L := \{\, a^n b^n c^n \mid n \geq 1 \,\}$.

Claim 1:

$L \subseteq L(G)$.

Proof of Claim 1.

We show that $a^n b^n c^n \in L(G)$ by induction on n.
For $n = 1$, we have $S \to abc$.
Assume that $S \to^* a^n b^n c^n$ has been shown for some $n \geq 1$.
We consider the following derivation:

$$S \to aSBc \to^* a \cdot a^n b^n c^n \cdot Bc \to^* a^{n+1} b^n Bc^{n+1} \to a^{n+1} b^{n+1} c^{n+1}.$$

□

Example (cont.)

Claim 2:

$L(G) \subseteq L$.

Proof of Claim 2.

By applying production 1 repeatedly, we obtain a sentential form $a^n S(Bc)^n$, which is rewritten into a sentential form $a^n S\alpha$ by production 3, where $\alpha \in \{B, c\}^+$ and $|\alpha|_B = |\alpha|_c = n$.
In order to get rid of S, production 2 must be applied, that is, we obtain $a^n S\alpha \to a^{n+1} bc\alpha$.
To get rid of all nonterminals B in α, all occurrences of c must be moved to the right and then all B are rewritten into b by production 4, that is, $a^{n+1} bc\alpha \to^* a^{n+1} bB^n c^{n+1} \to^* a^{n+1} b^{n+1} c^{n+1}$.
Hence, $L(G) \subseteq L$. □

Together Claims 1 and 2 show that $L(G) = L$. □

Each context-free grammar in CNF (Theorem 3.9) is context-sensitive.

On the other hand, $\{ a^n b^n c^n \mid n \geq 1 \} \notin$ CFL (see the first example after Theorem 3.14).

Corollary 4.25

CFL $\subsetneq$ CSL.

A grammar $G = (N, T, S, P)$ is in Kuroda Normal Form, if it only contains productions of the following forms:

$$(A \to a), (A \to BC), (AB \to CD), \text{ where } a \in T \text{ and } A, B, C, D \in N.$$

With respect to the production $(S \to \varepsilon)$, we have the same restriction as before.

Theorem 4.26

Given a context-sensitive grammar G, one can effectively construct an equivalent context-sensitive grammar G′ that is in Kuroda Normal Form.

Proof of Theorem 4.26.

As in the proof of Theorem 3.9 we can first revise G in such a way that terminals only occur on the right-hand side of productions of the form $(A \to a)$.

Next we replace productions of the form $(A \to B_1 B_2 \cdots B_n)$ $(n > 2)$ by new productions

$$(A \to B_1 Z_2), (Z_2 \to B_2 Z_3), \ldots, (Z_{n-1} \to B_{n-1} B_n),$$

where $Z_2, Z_3, \ldots, Z_{n-1}$ are new nonterminals.

Finally, let $(A_1 A_2 \cdots A_m \to B_1 \cdots B_n)$ be a productions s.t. $m > 1$ and $m + n > 4$. We choose new nonterminals $Z_2, Z_3, \ldots, Z_{n-1}$ and replace the production above by the following productions in Kuroda form:

Proof of Theorem 4.26 (cont.)

$$
\begin{array}{rcl}
(A_1A_2 & \to & B_1Z_2), \\
(Z_2A_3 & \to & B_2Z_3), \\
 & \vdots & \\
(Z_{m-1}A_m & \to & B_{m-1}Z_m), \\
(Z_m & \to & B_mZ_{m+1}), \\
(Z_{m+1} & \to & B_{m+1}Z_{m+2}), \\
 & \vdots & \\
(Z_{n-1} & \to & B_{n-1}B_n).
\end{array}
$$

The new productions can simulate the old one. On the other hand, they cannot be used in any other way, as the new nonterminals do not occur in any other productions.

By repeating this process for all productions of the form above, we obtain a grammar G' in Kuroda Normal Form s.t. $L(G') = L(G)$. □

A linear-bounded automaton (LBA) M is a 1-NTM

$$M = (Q, \Sigma, \Gamma, \Box, \delta, q_0, q_1)$$

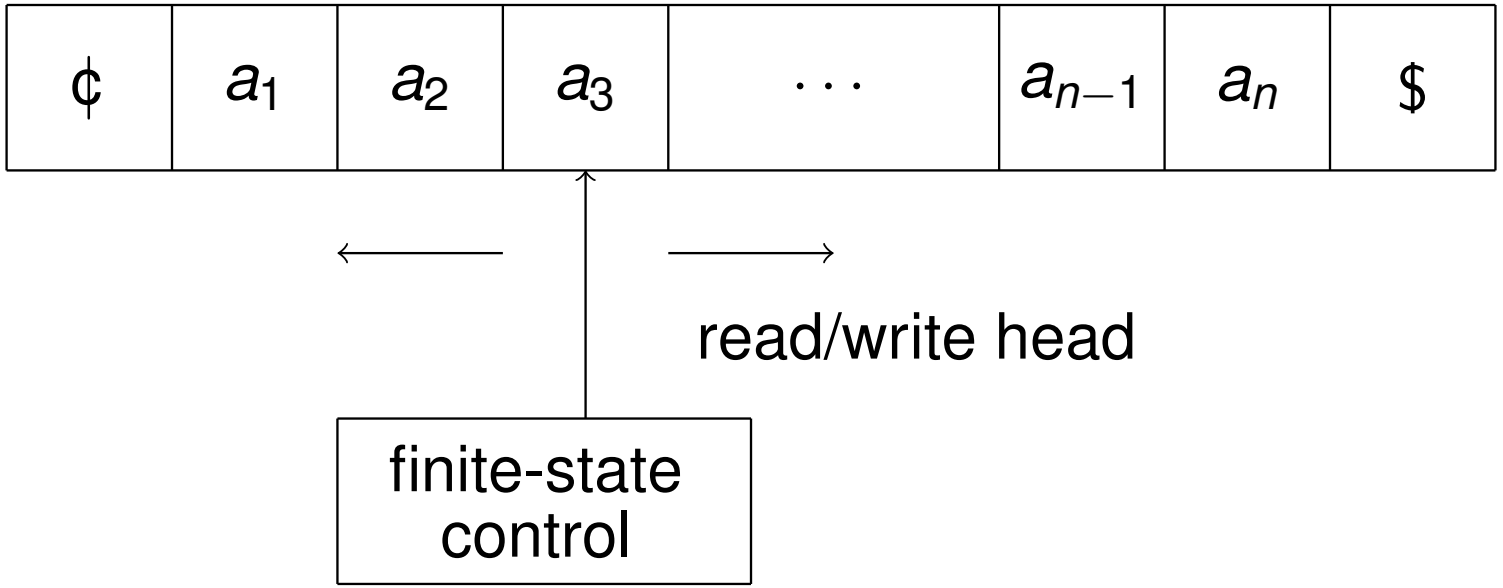
with two special symbols $¢, \$ \in \Gamma$.

For $w \in \Sigma^*$, $q_0 ¢ w \$$ is the initial configuration of M on input w, and

$$L(M) := \{ w \in \Sigma^* \mid \exists \alpha, \beta \in \Gamma^* : q_0 ¢ w \$ \vdash_M^* ¢ \alpha q_1 \beta \$ \}$$

is the language accepted by M.

An LBA can be depicted as follows:



Theorem 4.27

For each $L \in$ CSL, there exists an LBA M such that $L = L(M)$.

Proof.

We proceed as in the proof of Theorem 4.20, only that here our NTM M has a tape with two tracks instead of two tapes:

On track 1, the input word w is stored, and on track 2, a derivation of the context-sensitive grammar G is simulated nondeterministically. For doing so, we can assume that G is in Kuroda Normal Form.

If the simulated derivation generates the word w, then M accepts. If another terminal word is obtained, or if the space provided by the length of the input w does not allow for another step, then M enters an infinite loop.

Thus, M is an LBA such that $L(M) = L$. □

Theorem 4.28

For each LBA M, there exists a context-sensitive grammar G such that $L(G) = L(M)$.

Proof.

Here we proceed as in the proof of Theorem 4.21, that is, from a given LBA M, we construct a grammar G that simulates the computations of M.

Given a word $w = a_1 a_2 \cdots a_n \in \Sigma^*$ as input, the corresponding initial configuration $q_0 ¢ w \$$ of M is encoded by the word

$$\binom{¢\, a_1}{¢\, q_0 a_1}\binom{a_2}{a_2}\cdots\binom{a_{n-1}}{a_{n-1}}\binom{a_n \$}{a_n \$},$$

which is derived from the start symbol S of G by applying some context-free productions.

Proof of Theorem 4.28 (cont.)

Then a computation of the LBA M is simulated on the lower track only using productions $(\ell \to r)$ satisfying $|\ell| = |r|$.

If $w \in L(M)$, then a word of the following form can be derived:

$$\binom{¢\,a_1}{¢\,b_1}\binom{a_2}{b_2}\cdots\binom{a_i}{q_1\,b_i}\cdots\binom{a_{n-1}}{b_{n-1}}\binom{a_n\$}{b_n\$},$$

which can be rewritten into the word $a_1 a_2 \cdots a_{n-1} a_n = w$ by using monotone productions.

Hence, G is a context-sensitive grammar satisfying $L(G) = L(M)$. □

Corollary 4.29

A language L is context-sensitive iff there exists an LBA M such that $L(M) = L$.

A language L is called deterministic context-sensitive if it is accepted by a deterministic LBA. We denote the corresponding class of languages by DCSL.

Obviously, CFL $\subsetneq$ DCSL $\subseteq$ CSL, but it is still open whether the inclusion DCSL $\subseteq$ CSL is proper.
This is the famous LBA Problem (see [Hartmanis, Hunt 1974]).

Corollary 4.30

CSL $\subseteq$ REC, *that is, each context-sensitive language has decidable membership problem.*

Proof.

Let G be a context-sensitive grammar. In order to decide whether $w \in L(G)$ it suffices to generate all sentential forms α of G that satisfy the condition $|\alpha| \leq |w|$ and that can be derived from the start symbol S. If w is found in this way, then $w \in L$; otherwise, $w \notin L$. □

Corollary 4.31

For each recursively enumerable language L on Σ, there exists a context-sensitive language L' on $\Gamma \supsetneq \Sigma$ such that $\Pi_\Sigma(L') = L$. Here $\Pi_\Sigma : \Gamma^* \to \Sigma^*$ is the morphism that is defined by

$$a \mapsto a \,(a \in \Sigma) \text{ and } b \mapsto \varepsilon \,(b \in \Gamma \setminus \Sigma).$$

Proof.

Let L be a r.e. language on Σ. Then there exists a 1-TM $M = (Q, \Sigma, \Delta, \Box, \delta, q_0, q_1)$ s.t. $L(M) = L$.

We take $\Gamma := \Sigma \cup \{\$\}$ and

$$L' := \{\, w\$^n \mid w \in L \text{ and } M \text{ accepts } w \text{ in space } |w| + n \,\}.$$

Proof of Corollary 4.31 (cont.)

For $w \in \Sigma^*$ and $m \in \mathbb{N}$, M is given the input $w\$^m$.
Now M runs until it either accepts, which implies that $w\$^m \in L'$,
or until it needs more space than $|w| + m$, or until it gets into a loop.
In the latter two cases, $w\$^m \notin L'$.

From M we easily obtain an LBA that accepts L'.
Hence, $L' \in \text{CSL}$ and $L = \Pi_\Sigma(L')$. □

Because of Corollary 4.31, CSL is called a basis for the class RE.

Corollary 4.32

The class CSL *is not closed under morphisms.*

Proof.

This follows from the inclusion $\text{CSL} \subseteq \text{REC} \subsetneq \text{RE}$ and from Corollary 4.31. □

Theorem 4.33

The class CSL *is closed under union, product, Kleene star, and* ε*-free morphisms.*

Theorem 4.34

The class CSL *is closed under intersection and inverse morphisms.*

Theorem 4.35 (Immerman 1987, Szelepczenyi 1987)

The class CSL *is closed under complementation.*

Corollary 4.36

CSL $\subsetneq$ REC.

Actually, the membership problem for a context-sensitive language is decidable in exponential time.

On the other hand, Corollaries 4.14 and 4.31 imply that the following problems are undecidable for CSL:

- finiteness,
- emptiness,
- regularity,
- context-freeness,
- inclusion, and
- equality.

Chapter 5:

Summary

Summary on Characterizations:

Language classes	Grammars	Automata
Typ 3 (regular)	regular	DFA NFA
det. context-free		DPDA
Typ 2 (context-free)	context-free	PDA
Typ 1 (context-sensitive)	monotone	LBA
Typ 0 (recursively enumerable)	general	TM NTM

Summary on Closure Properties:

Operation	REG	DCFL	CFL	CSL	RE
Union	+	-	+	+	+
Intersection	+	-	-	+	+
Intersection with REG	+	+	+	+	+
Complementation	+	+	-	+	-
Product	+	-	+	+	+
Kleene star	+	-	+	+	+
Morphism	+	-	+	-	+
Inverse Morphism	+	+	+	+	+

Summary on Decision Problems:

Decision problem	REG	DCFL	CFL	CSL	RE
Membership	+	+	+	+	−
Emptiness	+	+	+	−	−
Finiteness	+	+	+	−	−
Equality	+	+	−	−	−
Inclusion	+	−	−	−	−
Regularity	+	+	−	−	−

\+ decidable
− undecidable