Context-Sensitive Languages.

We know that unrestricted languages (= those generated by unrestricted grammars, not just arbitrary sets of strings over an alphabet) make up the class of r.e. sets, and therefore have no general bounds on either time or space used to accept them.

We also found that the "natural" way to look at language acceptance is to use Non-Deterministic Turing Machines, which allow us to use, at each moment of decision, all possible string replacements (left-hand-side of a grammar rule replaced by the right-hand-side in any given sentential form).

The question we want to ask is: what class of languages can be accepted in Non-Deterministic Linear Space \((NSPACE(n))\): the corresponding Turing Machines are usually known as Linear Bounded Automata)?
Context-Sensitive Languages

Why is this question interesting?

If nothing else, because a machine with unbounded parallelism would allow us to obtain a solution in linear space along its shortest successful computational path. It would also tell us that we could abandon any computational track that exceeds the linear bound without fear of rejecting a string in the language.

Def. A context-sensitive language is one generated by a context-sensitive grammar: a grammar whose rules $\alpha \rightarrow \beta$ satisfy the condition $|\beta| \geq |\alpha|$, where $\alpha$ is a non-empty string of terminals and nonterminals, possibly augmented by the rule $S \rightarrow \varepsilon$. 
Context-Sensitive Languages

Ex.: Find a context-sensitive grammar for \( L = \{a^{2^n} \mid n \geq 0\} \).

Soln.: recall the grammar

1) \( S \rightarrow [Ra] \)
2) \( S \rightarrow a \)
3) \( Ra \rightarrow aaR \)
4) \( R] \rightarrow L[ \)
5) \( R] \rightarrow L_h \)
6) \( aL \rightarrow La \)
7) \( [L \rightarrow [R \)
8) \( aL_h \rightarrow L_ha \)
9) \( [L_h \rightarrow \epsilon \)

This is an unrestricted grammar generating the language: note that rules 5 and 9 violate the context-sensitive requirements (at least if all individual symbols are tape symbols).

We will change it to a context-sensitive one (not all unrestricted grammars can be so changed) by hiding the auxiliary symbols \( L, R, [, ] \), and attaching them to their neighboring real symbols (which will eventually become terminal symbols).
Context-Sensitive Languages

We use new non-terminal symbols $x A_y$ to replace a substring $XaY$ in a sentential form of grammar $G$, where $X$ and $Y$ do not contain the symbol $a$. When possible, we attach an auxiliary symbol $X$ to $a$ on its right: new symbols $x A_y$ and $A_y$ are used only when $Y$ ends with $]$. We will use $\overline{\text{[ ]}}$ to indicate how the individual symbols are "glued together" into a composite symbol.

Rules 1) and 2) are replaced by:

1) $S \rightarrow [Ra]$
1a) $S \rightarrow \overline{[Ra]}$

2) $S \rightarrow a$
2a) $S \rightarrow a$

For which the right-hand-side has length 1 - same as the left-hand-side (we have collapsed 4 symbols into 1 in rule 1)).
Context-Sensitive Languages

Rules 3)

3) $Ra \rightarrow aaR$

is replaced by:

3a) $[R^A a \rightarrow aa [R^A]$

3b) $[R^A R^A_1 \rightarrow aa [R^A_1]$

3c) $[R^A_1 \rightarrow a [R^A]$

3d) $[R^A a \rightarrow [A^R a [R^A]$

3e) $[R^A A^R_1 \rightarrow [A^R a [R^A]$

3f) $[R^A_1 \rightarrow [A^R A^R_1]$
Context-Sensitive Languages

Replace rules 4) - 9):

4a) $A_R \rightarrow A_L$

5a) $A_R \rightarrow A_{Lh}$

6a) $a \rightarrow L^A a$

6b) $[A \rightarrow [L^A] a$

6c) $a \rightarrow L^A A$

6d) $[A \rightarrow [L^A] A$

6e) $A_L \rightarrow L^A$

6f) $[A \rightarrow [L^A] A$

8a) $A_{Lh} \rightarrow L^A$

8b) $A_{Lh} \rightarrow L^A a$

8c) $[A \rightarrow [L^A] a$

9a) $[L^A] \rightarrow a$
Context-Sensitive Languages

To understand the new rules, we work out a few derivations.

1. \( S \xrightarrow{1a)} [R^A] \xrightarrow{3f)} [A] \xrightarrow{4a)} [A] A_{R} \xrightarrow{5a)} [A] A_{L} \xrightarrow{7a)} [A] a \xrightarrow{3e)} [A] _{R}A_{L} \xrightarrow{8a)} [A] aa \xrightarrow{9a)} a \xrightarrow{8c)} [L_{h} A] aa \xrightarrow{9a)} a\)

2. \( S \xrightarrow{1a)} [R^A] \xrightarrow{3f)} [A] \xrightarrow{4a)} [A] A_{R} \xrightarrow{6a)} [A] A_{L} \xrightarrow{6d)} [L_{h} A] A_{R} \xrightarrow{5a)} [A] aa \xrightarrow{8a)} A_{L} A_{h} \xrightarrow{9a)} a\)

3. Check which other rules will be used in the derivation:

\[ S \rightarrow aaaaaaaa. \]
Context-Sensitive Languages

Theorem 6.35. The class of context-sensitive languages is exactly the class $NSPACE(n)$.

Proof. Let $G$ be a CSG. Construct a standard 1-worktape NTM $M$ that accepts all strings $w \in L(G)$ in space $O(n)$.

Intuition: $M$ keeps a sentential form $\alpha$ on its worktape. Non-deterministically it applies the rules of $G$ to derive the next sentential form until it is equal to the input. By assumption on $G$, the lengths of the successive sentential forms are non-decreasing. Thus the successful computation uses $n$ cells.
Context-Sensitive Languages

More Formal:

1. Write $S$ on the work tape, $\alpha = S$.
2. Repeat until either a rejecting or accepting condition is met:
   a) Nondeterministically select a grammar rule $u \to v$ of $G$.
   b) Find the first occurrence of $u$ in $\alpha$, and replace it by $v$. If no $u$ is found, reject. Let this new string ($\beta$) be the new string on the worktape.
   c) Compare $\beta$ with the input $w$. If $w = \beta$, accept; if $|\beta| > |w|$, reject; otherwise continue with $\beta$ replacing $\alpha$.

If $w \in L(G)$, there is a leftmost derivation for $w$:

$$S = \alpha_0 \Rightarrow \alpha_1 \Rightarrow \ldots \Rightarrow \alpha_m = w.$$ 

Corresponding to this derivation, there is a computation path of $M$ on input $w$ for which the strings written on the worktape at step 2b) are exactly the strings above. Since the sentential forms at each step never shrink, this computation path cannot lead to a rejection before accepting.
Context-Sensitive Languages

The only thing still missing is the proof that $M$ works within a space bound $n + c$, where $c$ is the max length of the r.h.s. of the rules of $G$. If the rule $u \rightarrow v$ satisfies $|u| = |v|$, the character by character substitution takes no extra space. If $|u| < |v|$, then we must move part of the sentential form $\alpha$ to the right to make space. The construction of the function

$$\text{insert}_i^k(x_1, x_2, \ldots, x_k, y) = (x_1, \ldots, x_{i-1}, y, x_i, \ldots, x_k)$$

(Ex. 4.9, p. 171, shows how it can be accomplished).
Context-Sensitive Languages

\[ L \in NSPACE(n) \] We need to construct a context-sensitive grammar that generates it.

We know, from some previous results, how to construct an unrestricted grammar, given a Deterministic Turing Machine. We assume, in this case, that \( \exists M_1 \), a one-tape NTM that accepts \( L \) along a computation path that does not read any symbol beyond the two blanks that surround the input \( w \in L \).

We can use the transitions of the NTM exactly the same way we used the transitions of the DTM: for each \( (q, a) \in Q \times \Sigma \) (this means \( a \neq B \)) we define a non-terminal symbol \( q#a \). We will use \textbf{boldface} \( (q#a) \) to distinguish these non-terminals from the regular elements of the tape alphabet \( \Gamma \).
Context-Sensitive Languages

We define the grammar $G_1$ with the rules ($c$ can be $\mathbb{B}$):

1. For each instruction $(p, b, L) \in \delta(q, a)$ of $M_1$, $G_1$ has the rules:
   $$c \quad q\#a \rightarrow p\#c \quad b \quad \forall \ c \in \Gamma$$

2. For each instruction $(p, b, R) \in \delta(q, a)$ of $M_1$, $G_1$ has the rules:
   $$q\#a \quad c \rightarrow b \quad p\#c \quad \forall \ c \in \Gamma$$

This is the grammar constructed from a general TM (Thm. 4.19), except for omitting the boundary symbols $[\, , \, ]$, where the rules in group 1 above correspond to the rules in groups 1 & 2 of Thm. 4.19, and the rules in group 2 above correspond to the rules in the previous group 3. Since $M_1$ never reads beyond the leftmost blank or the first blank to the right of the input, there is no reason to keep track of them.
Since "shrinking" is simply taken care of by allowing the machine to keep all the trailing blanks ever visited.

Reverse all the rules of $G_1$, add the rules

$$S \rightarrow B \, h\#B \, T, \quad T \rightarrow BT \mid B, \quad s#B \rightarrow L_h,$$

$$aL_h \rightarrow L_ha, \quad BL_h \rightarrow \varepsilon$$

for all $a \in \Sigma$, with $L_h$ a new non-terminal.

One can show this extended grammar generates the non-empty strings of $L$. 
We mention two further useful results:

**Thm. 6.36**: for any fully space-constructible function \( s(n) \geq \log n \),
\[
\text{NSPACE}(s(n)) = \text{co-NSPACE}(s(n)).
\]

**Corollary 6.37**: If \( A \) is a context-sensitive language, then \( \overline{A} \) is also context-sensitive.