a) Savitch's Theorem

If \( \log^k n \leq \log n \), then \( \text{NSpace}(\log n) \subseteq \text{Space}(n^k) \)

Hence \( \text{NSpace} = \text{Pspace}, \) since the square of a polynomial is a polynomial.

Hence

\[ \text{NSpace} = \text{Pspace} = \text{Co-Pspace} = \text{Co-NSpace} \]

Whence

\[ \text{NSpace} = \text{Co-NSpace} \]

since the complement of the complement of \( L \) is \( L \).

b) A CFG is a quadruple \( G = (S, N, V, P) \) where

\( N \) is a set of 'non-terminals', \( S \in N \), \( V \) a set of 'terminals' (the alphabet) \( \neq \emptyset \) and \( P \) a set of 'productions' of the form \( A \to \beta \) where \( A \in N, \beta \in (N \cup V)^* \).

A CSG is like a CFG except that \( P \) is a set of productions \( \alpha \to \beta \), \( \alpha, \beta \in (N \cup V)^* \), with \( |\alpha| \leq |\beta| \)

If \( \alpha, \omega, \omega_2 \) is a string and \( \alpha \to \beta \) a production in \( G \), write \( \alpha, \omega, \omega_2 \to \gamma, \alpha, \beta, \omega, \omega_2 \). Let

\( \Rightarrow_0 \) be the reflexive transitive closure of \( \Rightarrow \).

The language recognized by \( G \) is the set

\( \{ \omega \in V^* \mid S \Rightarrow_0 \omega \Rightarrow_0 \} \).
Suppose \( L \in \text{NTIME}(n) \). Let \( M \) be a non-deterministic TM recognizing \( L \) and running in time \( f(n) = n \). We assume \( M \) has no memory tape, and operates directly on input \( x \), with \( |x| = n \). We build \( G = \langle S, \Sigma, \delta, S \rangle \) as follows. The set \( \Sigma \) is simply the alphabet of \( M \).

Let \( N \) (non-terminals) be the set of pairs \( \langle h, a \rangle \) where \( a \) is a symbol in alphabet \( \Sigma \) of \( M \) and \( h \) is a state of \( M \) or the symbol \( ! \). We also allow \( N \) to contain our additional start symbol \( S \).

A run of \( M \) may be depicted as follows, as a sequence of strings over \( N \):

<table>
<thead>
<tr>
<th>( \langle s, t \rangle )</th>
<th>( \langle t, u \rangle )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \langle s, t \rangle )</td>
<td>( \langle t, ! \rangle )</td>
</tr>
<tr>
<td>( \langle !, a \rangle )</td>
<td>( \langle !, a' \rangle )</td>
</tr>
</tbody>
</table>

Here, \( \langle s, a \rangle \) means "tape square containing an \( a \) with head reading that square, and \( s \) state \( s \); \( \langle !, a \rangle \) means "tape square containing a with head somewhere else ". We now define the set \( P \) of productions.

For a transition, \( \langle s, a \rangle \rightarrow \langle t, b, \text{right} \rangle \), we adopt the grammar rules (productions):

\[
\langle s, !, a \rangle \rightarrow \langle s, a \rangle \langle !, a' \rangle
\]

For all \( a' \) in alphabet of \( M \). Notice how the
production is a 'backwards' version of the transition. (Similarly for \((s,a) \rightarrow (t,b,\text{left})\).)

We also adopt the productions

\[
S \rightarrow \langle s^*, \gamma \rangle \langle \gamma \rangle
\]

\[
\langle \gamma \rangle \rightarrow \langle \gamma \rangle \langle \gamma \rangle
\]

where \(S\) is the special start symbol and \(s^*\) is the accepting state of \(M\).

Finally, we adopt the productions

\[
\langle s_0, a \rangle \langle \gamma, a' \rangle \rightarrow a \langle s_0, a' \rangle
\]

for all \(a, a'\) in alphabet of \(M\), as well as

\[
\langle s_0 a \rangle \rightarrow a
\]

for all \(a\) in alphabet of \(M\). By inspection of diagram:

\[
(x = a_1 \ldots a_n \in L) \iff
\]

\[
(M \text{ has an accepting run on } a_1 \ldots a_n) \iff
\]

\[
(S \Rightarrow_g \langle s^*, \gamma \rangle \langle \gamma \rangle \ldots \langle \gamma \rangle
\]

\[
\Rightarrow_g \langle s_0 a_1 \rangle \langle \gamma \rangle \langle \gamma \rangle \ldots \langle \gamma \rangle
\]

\[
\Rightarrow_g \langle \gamma \rangle \langle \gamma \rangle \langle \gamma \rangle
\]

Hence \(L\) is a context sensitive language.
e) Suppose \( L \) is context-sensitive. Let \( G = (S, N, V, P) \) recognize \( L \). To determine whether \( x \in L \), use the following nondeterministic algorithm:

\[
\begin{align*}
\text{begin} & \quad \text{test-} L (x) \\
\text{until } x = \varepsilon & \quad \text{do} \\
\quad \text{move head to some position in } x \\
\quad \text{choose some production } \alpha \to \beta \in P \\
\quad \text{if } x \text{ matches } \beta \text{ at chosen position,} \\
\quad \quad \text{replace occurrence of } \beta \text{ by } \alpha \text{ and} \\
\quad \quad \text{move remaining symbols in } x \text{ up to} \\
\quad \quad \text{close up gaps (if } |\alpha| < |\beta|). \\
\quad \text{else} \\
\quad \quad \text{return no} \\
\text{end until} \\
\text{exit return} \quad \text{yes} \\
\text{end test-} L
\end{align*}
\]

This certainly requires no more space than \( x \), and has a terminating run iff \( S \Rightarrow_{G} x \), i.e., iff \( x \in L \).

c) Suppose \( L \) is a CSL.

By e) \( L \in \text{NSpace}(n) \)

By b) \( L \in \text{NSpace}(n) \)

By d) \( L \) is a CSL.