CISC 462 Asn 4 solutions

$\boxed{1.}$ a) Suppose $A \leq_p B$ via function f and $B \leq_p C$ via function g. Function f is computed in time $p_f(n)$ and function g in time $p_g(n)$.

To show $A \leq_p C$ we consider the composition of f and g:

$g \circ f : \Sigma^* \to \Sigma^*$, $(g \circ f)(w) = g(f(w))$. Now

$(g \circ f)(w) \in C$ iff $f(w) \in B$ iff $w \in A$.

Time needed to compute $(g \circ f)(w)$; $|w| = n$:

- $f(w)$ is computed in time $p_f(n)$, and $|f(w)| \leq p_f(n)$
- $g(f(w))$ is computed in time at most $p_g(p_f(n))$.

Total time $p_f(n) + p_g(p_f(n))$ is polynomial in n.

$\boxed{1.}$ b) Since $B \neq \emptyset$, $B \neq \Sigma^*$ we can choose $w_1 \in B$ and $w_2 \notin B$. Define a function $f: \Sigma^* \to \Sigma^*$:

$$f(u) = \begin{cases} w_1 & \text{if } u \in A \\ w_2 & \text{if } u \notin A. \end{cases}$$

Since $A \in P$, f can be computed in polynomial time: the algorithm just checks whether or not $u \in A$ and then outputs a constant string w_1 or w_2.

For all $u \in \Sigma^*$:

$$u \in A \text{ iff } f(u) \in B$$

2. (i) DOUBLE-SAT $\in$ NP: Nondeterministic poly-time decider for DOUBLE-SAT:

M = " On input $\langle \phi \rangle$

1. Guess two different truth assignments for variables of ϕ
2. If both assignments make ϕ evaluate to true, accept"

(ii) We show that SAT $\leq_p$ DOUBLE-SAT via function f:

$$\begin{cases} f(\langle \phi \rangle) = \langle \phi \wedge (x_1 \vee x_2) \rangle & \text{where variables } x_1, x_2 \text{ do not appear in } \phi \\ f(w) = \varepsilon & \text{if } w \text{ does not encode a Boolean formula} \end{cases}$$

Now
$(\forall w \in \Sigma^*)$ $w \in$ SAT iff $f(w) \in$ DOUBLE-SAT

Note that if ϕ has satisfying truth assignment, $\phi \wedge (x_1 \vee x_2)$ has at least two satisfying assignments because either x_1 or x_2 can be assigned value true.

By Theorem 7.36 it follows that DOUBLE-SAT is NP-complete.

$\boxed{3.}$ We note that $w \in \{ (,) \}^*$ is in A iff

(i) the number of left and right parentheses in w are equal, and,

(ii) any prefix of w contains at least as many left parentheses as right parentheses.

A can be decided by a TM that reads the input from left-to-right and on the work tape keeps track of the number of left-paren. <u>minus</u> number of right parentheses encountered.

O(log n) space is sufficient for a binary counter.

After reading each symbol, the TM updates the value of the counter — this does not require space overhead.

If the counter becomes negative, the TM rejects.

The TM accepts if the counter has value zero at the end of the input.

$\boxed{4.}$ (i) $A_{LBA} \in PSPACE$

In polynomial space we can simulate an LBA M on input w keeping track of the number of steps. This is done as in Theorem 5.9. If the simulation has not halted in $q \cdot n \cdot g^n$ steps, we reject. (Here q is the number of states of M, $n = |w|$ and g is the size of the alphabet of M.)

(ii) Let $A \in PSPACE$ be arbitrary, where A is decided by a single-tape TM N in space n^k.

We define mapping reduction f by setting, for all $w \in \Sigma^*$, $f(w) = \langle N', w\sqcup^{|w|^k} \rangle$

Here N' operates exactly as N except that N' can use only the part of the tape where the input is given, that is, N' is an LBA.

Now (continues next page)

$\boxed{4.}$ (continued)

$w \in A$ iff N accepts w in space n^k iff N' accepts $w\sqcup^{|w|^k}$ iff $f(w) \in A_{LBA}$.

The above shows that A reduces to A_{LBA} via the function f. The function f can be computed in polynomial time because N' is a fixed LBA and in polynomial time we can append $|w|^k$ blank symbols to a string w.

$\boxed{5.}$ a) We show that $A_{LBA} \notin NL$.

By the space hierarchy theorem there exists $B \in SPACE(n)$ such that $B \notin SPACE((\log n)^2)$. By Savitch's theorem it follows that $B \notin NL$. Linear space TM's can be converted to LBA's, so B is decided by some LBA N. The reduction $w \mapsto \langle N, w \rangle$ is a logspace reduction from B to A_{LBA}. So if $A_{LBA} \in NL$, this would imply $B \in NL$ which is a contradiction.

b) We do not know whether $A_{LBA} \in P$.

If $P = PSPACE$, then the answer is "yes".

On the other hand, A_{LBA} is PSPACE-complete, so $P \neq PSPACE$ implies $A_{LBA} \notin P$.

$\boxed{6.}$ a) $NSPACE(n \cdot \log n) \underset{\text{Savitch}}{\subseteq} SPACE(n^2(\log n)^2)$

$\underset{\text{space hierarchy}}{\subsetneq} SPACE(n^3)$ because $n^2(\log n)^2 = o(n^3)$

b) $TIME(n^3 \cdot \log n) \overset{\text{time hierarchy}}{\subsetneq} TIME(n^3\sqrt{n})$

Justification: $t_2(n) = n^3\sqrt{n}$, $t_1(n) = n^3 \cdot \log n$

$$\frac{t_2(n)}{\log t_2(n)} = \frac{n^3\sqrt{n}}{3\frac{1}{2} \cdot \log n} \quad \text{Now}$$

$$t_1(n) \Big/ \frac{t_2(n)}{\log t_2(n)} = \frac{n^3 \cdot \log n \cdot 3\frac{1}{2} \cdot \log n}{n^3 \cdot \sqrt{n}} = \frac{3\frac{1}{2}(\log n)^2}{\sqrt{n}} \xrightarrow[n \to \infty]{} 0$$

This means that $t_1(n) = o\left(\frac{t_2(n)}{\log t_2(n)}\right)$

c) $TIME(2^n) \underset{\text{time hierarchy}}{\subsetneq} TIME(3^n)$

Justification:

$$2^n \Big/ \frac{3^n}{\log 3^n} = \left(\frac{2}{3}\right)^n \cdot \log 3 \cdot n \longrightarrow 0 \text{ when } n \to \infty$$

Thus $2^n = o\left(\frac{3^n}{\log 3^n}\right)$

$\boxed{6.}$ d) $NTIME(n\cdot\sqrt{n}) \subseteq NSPACE(n\sqrt{n})$

$\subseteq SPACE(n^3) \subsetneq SPACE(n^3 \cdot \log n)$

Savitch

space hierarchy

e) $f(n)$ is $O(n^3 \cdot \log n)$ because we can ignore constant number of small values. This means that $TIME(n^3) \subseteq TIME(f(n))$ (non-strict inclusion)

The time hierarchy theorem does not apply for $t_1(n) = n^3$ and $t_2(n) = n^3 \cdot \log n$.