Assignment 2 **CAS776 Fall 2002** **due on October 31**

1. As is mentioned in the book, the proof is almost identical to the proof of the linear speedup for time. The only difference is that when measuring space, we use the convention of TM with input tape, whose length we don't take into account, and that is why there is no n factor as in the linear speedup for time. The idea is still the same: encode several symbols of M by a single symbol of M'.

2. (b) Very similar to (d), the only difference is that instead of asking if $U'(M;z) = 0$, we check if $U'(M;z) =$ something, and if it is equal to something, the something must be 0, since no other output is given.

   (c) Let U' be a modified UTM, with an extra symbol $\square$, and an extra state X. On input $M;x$, U' simulates M on x, and if M halts on x, U' writes $\square$ wherever it is on the tape, and cycles through all the states, including X which it never enters otherwise, and then halts. If (c) were decidable, we could decide H as follows: check if on input $\underbrace{M;z}_{x}$, $\underbrace{U'}_{M}$ uses all the states. If U' uses all the states, then in particular it used X, so M must have halted on z. If U' does not use all the states, then it never reached the final phase, so M never halts on z.

   (d) Let U' be a modified UTM, such that on input $M;x$, U' simulates M on x, and outputs 0 if M halts in either state (accept or reject). If (d) were decidable, we could decide H as follows: check if $\underbrace{U'}_{M}(\underbrace{M;z}_{x}) = \underbrace{0}_{y}$.

3. **4.4.11** (a) just note that $\vee$ can be expressed over $\{\wedge, \neg\}$ as follows: $\alpha \vee \beta \equiv \neg(\neg\alpha \wedge \neg\beta)$. (b) We show that NOR can express $\wedge$ and $\neg$: $\mathrm{NOR}(x,x) \equiv \neg(x \vee x) \equiv \neg x \wedge \neg x \equiv \neg x$. Also, $\mathrm{NOR}(\mathrm{NOR}(x,x), \mathrm{NOR}(y,y)) \equiv \mathrm{NOR}(\neg x, \neg y) \equiv \neg(\neg x \vee \neg y) \equiv (x \wedge y)$. Same for NAND. (c) Examine all connectives one by one; we know that $\neg$ by itself cannot express functions with more than one variable, so it is no good. AND and OR ($\wedge$ and $\vee$) can only express monotone functions (by next problem), so they are also no good. Etc.

4. **4.4.13** $\Longleftarrow$ by induction on the number of gates. If there is one gate, the circuit is $x \wedge y$ or $x \vee y$, by looking at the truth table we see that it computes a monotone function.

   $\Longrightarrow$ by induction on the number of variables. Suppose f has one variable, and f is a monotone function. Then it must be one of the following functions: (i) $f(0) = 0$, $f(1) = 0$, (ii) $f(0) = 0$, $f(1) = 1$, (iii) $f(0) = f(1) = 1$. In the first case f is given by the circuit consisting of the single constant 0 gate, in the second case by the single input gate x, and in the last case by the single constant gate 1.

   Suppose now that f has $n > 1$ many variables, $f(x_1, x_2, \ldots, x_n)$, and f is a monotone function. Then:

$$f(x_1, x_2, \ldots, x_n) = f(0, x_2, \ldots, x_n) \vee (x_1 \wedge f(1, x_2, \ldots, x_n))$$

To see this, suppose that $x_1 = 0$. Then, the RHS is equal to $f(0, x_2, \ldots, x_n)$. Suppose that $x_1 = 1$. Then, the RHS is equal to $f(0, x_2, \ldots, x_n) \vee f(1, x_2, \ldots, x_n)$. If $f(1, a_2, \ldots, a_n) = 0$, so must $f(0, a_2, \ldots, a_n)$ (by monotonicity of f). On the other hand, if $f(1, a_2, \ldots, a_n) = 1$, then the whole disjunction is 1.

By induction, both $f(0, x_2, \ldots, x_n)$ and $f(1, x_2, \ldots, x_n)$ have circuits over $\{\wedge, \vee\}$, since they are monotone functions themselves; thus f has a circuit over $\{\wedge, \vee\}$.