

Final Exam
CAS705
December 5, 2005
Examiner: Michael Soltys
Duration: 3hrs

1. Recall that CIRCUIT VALUE is the language of $\langle C, x \rangle$ such that x is an input that satisfies the circuit C .
 - (a) Show that CIRCUIT VALUE is **P**-complete.
 - (b) Show that CIRCUIT VALUE remains **P**-complete even if we make the following two restrictions: (1) all the gates are OR's and AND's (i.e., C is monotone), and (2) all the gates are arranged in alternating layers of AND's and OR's.
2. Show that the class of languages decidable with alternating Turing machines that use space $O(\log(n))$ and time $O(\log^i(n))$ is precisely (log-space uniform) **NCⁱ**.
3. Let 2SAT be the language of satisfiable formulas in conjunctive normal form, where each clause has exactly 2 literals. Use resolution to show that 2SAT is in **P**.
4. Show that if **P** = **NP** then **P** = **BPP**.

(**Hint.** First show that **BPP** \subseteq Σ_2 **P**; recall that the idea of the proof was to use “amplification” to bring the error down to $1/2^n$, and then to define $A(x)$ to be the set of paths in the computation tree that are accepting (not *correct*). Bound the size of $A(x)$ for when x is in the language and for when it is not. Finally, use “translations” of $A(x)$, i.e., $t \oplus A(x) = \{t \oplus w \mid w \in A(x)\} \dots$. For the second part, show that if **P** = **NP**, then the entire polytime hierarchy **PH** collapses to **P**.)
5. A language is *regular* if it is described by a regular expression, or by a finite automaton. Alternatively, a language is regular if it is recognized by the following model of a Turing machine: it scans the input left to right changing states, and has two sets of states, accepting and rejecting. If after scanning the last symbol it finds itself in an accepting state it accepts, and rejects otherwise. Note that it never writes anything on the tape.

Show that the set of regular languages is contained in **NC¹**.

(**Hint.** Here is an idea: assume $\Sigma = \{0, 1\}$, and define M_a , $a \in \Sigma$, to be the Boolean matrix with a 1 (true) in position (i, j) iff on symbol a , the machine moves from state q_i to state q_j . Given $a \in \{0, 1\}^*$, $a = a_1 a_2 \dots a_n$, consider $M_{a_1} M_{a_2} \dots M_{a_n}$; how can

you tell if a is in the language from this *iterated Boolean matrix product*? (Boolean product means that $+$ is replaced by \vee and \times is replaced by \wedge .)

6. Show that if all languages in \mathbf{NP} have polysize circuits, then the polytime hierarchy collapses to its second level, that is, show that if $\mathbf{NP} \subseteq \mathbf{P/poly}$ then $\mathbf{PH} = \Sigma_2\mathbf{P}$.

(**Hint.** Note that it is enough to show that if $\mathbf{NP} \subseteq \mathbf{P/poly}$, then $\Pi_2\mathbf{P} \subseteq \Sigma_2\mathbf{P}$. Why does this imply the collapse? To show that $\mathbf{NP} \subseteq \mathbf{P/poly} \implies \Pi_2\mathbf{P} \subseteq \Sigma_2\mathbf{P}$ argue as follows: assume L is in $\Pi_2\mathbf{P}$, so $L = \{x \mid \forall y \exists z R(x, y, z)\}$. Consider the language $L' = \{\langle x, y \rangle \mid \exists z R(x, y, z)\}$, which is in \mathbf{NP} by definition. Argue that there exists a polytime function f such that $L = \{x \mid \forall y f(\langle x, y \rangle) \in \text{SAT}\}$.)

7. A family of circuits $\{C_n\}$ is a *randomized circuit family for f* if in addition to the n inputs x_1, x_2, \dots, x_n , it takes m random bits r_1, r_2, \dots, r_m , and furthermore C_n satisfies two properties:

- (a) If $f_n(x_1, \dots, x_n) = 0$ then $C_n(x_1, \dots, x_n, r_1, \dots, r_m) = 0$ *regardless* of the value of the r_i 's (i.e., no possibility of error).
- (b) If $f_n(x_1, \dots, x_n) = 1$, then $C_n(x_1, \dots, x_n, r_1, \dots, r_m) = 1$ with probability at least $1/2$.

Show that if f has a randomized polysize circuit family, then it has a polysize circuit family (i.e., it can be *de-randomized*).

(**Hint.** Form a matrix M with 2^n rows, corresponding to each input, and 2^m columns, corresponding to each random input. Let M_{jk} be 1 if the random input corresponding to the k -th column is a *witness* to the input corresponding to the j -th input (i.e., this choice of random bits sets the circuit on that input to be 1). Eliminate all rows for which f_n is 0. At least half the entries in each surviving row are 1, so there must be a column where at least half the entries are 1)

End of Exam