

- a. Show that  $\text{NP} \cup \text{coNP} \subseteq \text{P}^{\text{SAT},1}$ .
- b. Assume  $\text{NP} \neq \text{coNP}$ . Show that  $\text{P} \cup \text{coNP} \subsetneq \text{P}^{\text{SAT},1}$ .
- 9.26 Suppose  $A$  and  $B$  are two oracles. One of them is an oracle for  $TQBF$ , but you don't know which. Give an algorithm that has access to both  $A$  and  $B$ , and which is guaranteed to solve  $TQBF$  in polynomial time.
- 9.27 Let  $E_{\text{REG}\uparrow} = \{\langle R \rangle \mid R \text{ is a regular expression with exponentiation and } L(R) = \emptyset\}$ . Show that  $E_{\text{REG}\uparrow} \in \text{P}$ .

### SELECTED SOLUTIONS

- 9.1 We have defined the time complexity classes in terms of the big- $O$  notation so constant factors have no effect. The function  $2^{n+1}$  is  $O(2^n)$  and thus  $A \in \text{TIME}(2^n)$  iff  $A \in \text{TIME}(2^{n+1})$ .
- 9.2 The containment  $\text{TIME}(2^n) \subseteq \text{TIME}(2^{2n})$  holds because  $2^n \leq 2^{n+1}$ . The containment is proper by virtue of the time hierarchy theorem. The function  $2^{2n}$  is time constructible because a TM can write the number 1 followed by  $2n$  0s in  $O(2^{2n})$  time. Hence the theorem guarantees that a language  $A$  exists that can be decided in  $O(2^{2n})$  time but not in  $O(2^{2n}/\log 2^{2n}) = O(2^{2n}/2n)$  time. Therefore  $A \in \text{TIME}(2^{2n})$  but  $A \notin \text{TIME}(2^n)$ .
- 9.3  $\text{NTIME}(n) \subseteq \text{NSPACE}(n)$  because any machine that operates in time  $t(n)$  on every computation branch can use at most  $t(n)$  tape cells on every branch. Furthermore  $\text{NSPACE}(n) \subseteq \text{SPACE}(n^2)$  due to Savitch's theorem. However,  $\text{SPACE}(n^2) \subsetneq \text{SPACE}(n^3)$  because of the space hierarchy theorem. The result follows because  $\text{SPACE}(n^3) \subseteq \text{PSPACE}$ .
- 9.7 (a)  $\Sigma^{500}$ ; (b)  $(\Sigma \cup \varepsilon)^{500}$ ; (c)  $\Sigma^{500}\Sigma^*$ ; (d)  $(\Sigma \cup \varepsilon)^{499} \cup \Sigma^{501}\Sigma^*$ .
- 9.20 (a) Let  $A$  be any language and  $k \in \mathcal{N}$ . If  $A \in \text{P}$  then  $\text{pad}(A, n^k) \in \text{P}$  because we can determine whether  $w \in \text{pad}(A, n^k)$  by writing  $w$  as  $s\#^l$  where  $s$  doesn't contain the # symbol, then testing whether  $|w| = |s|^k$ , and finally testing whether  $s \in A$ . Implementing the first test in polynomial time is straightforward. The second test runs in time  $\text{poly}(|w|)$ , and because  $|w|$  is  $\text{poly}(|s|)$ , the test runs in time  $\text{poly}(|s|)$  and hence is in polynomial time. If  $\text{pad}(A, n^k) \in \text{P}$  then  $A \in \text{P}$  because we can determine whether  $w \in A$  by padding  $w$  with # symbols until it has length  $|w|^k$  and then testing whether the result is in  $\text{pad}(A, n^k)$ . Both of these tests require only polynomial time.
- (b) Assume that  $\text{P} = \text{SPACE}(n)$ . Let  $A$  be a language in  $\text{SPACE}(n^2)$  but not in  $\text{SPACE}(n)$  as shown to exist in the space hierarchy theorem. The language  $\text{pad}(A, n^2) \in \text{SPACE}(n)$  because we have enough space to run the  $O(n^2)$  space algorithm for  $A$  using space that is linear in the padded language. Because of the assumption,  $\text{pad}(A, n^2) \in \text{P}$ , and hence  $A \in \text{P}$  by part (a), and hence  $A \in \text{SPACE}(n)$  using the assumption once again. But that is a contradiction.