

Rozwiązanie testu

25 czerwca 2015

1. $\text{SAT} \in \text{NC}$. nie wiadomo
Nie wiemy nawet, czy SAT nie należy do NC^1 .
2. $\text{BPP} \cap \text{coBPP} = \text{ZPP}$. nie wiadomo
Klasa BPP jest zamknięta na dopełnienie, więc oczywiście $\text{BPP} \cap \text{coBPP} = \text{BPP}$, podczas gdy $\text{ZPP} = \text{RP} \cap \text{coRP}$. Stąd oczywiście $\text{ZPP} \subseteq \text{BPP}$, ale przeciwna inkluzja nie jest znana.
3. Dla każdego problemu NP -zupełnego istnieje PTAS. nie wiadomo
Wprawdzie wiadomo, że dla *problemu komiwojażera* nie istnieje algorytm aproksymacyjny (a więc tym bardziej PTAS), ale **przy założeniu**, że $\text{P} \neq \text{NP}$. Gdyby $\text{P} = \text{NP}$, to każdy optymalizacyjny problem w NP miałby algorytm deterministyczny, który w trywialny sposób spełnia założenia PTAS (błąd = 0).
4. $\text{NPSpace} \cap \text{coNPSpace} = \text{PSPACE}$. tak
Deterministyczna klasa PSPACE jest oczywiście zamknięta na dopełnienie, a z Tw. Savitcha $\text{NPSpace} = \text{PSPACE}$. Zatem wszystkie występujące tu klasy są równe.
5. $\text{NP} \cap \text{coNP} = \text{P}$. nie wiadomo
6. $\text{EXPSPACE} = \text{coEXPSPACE}$. tak
Z definicji złożoności pamięciowej.
7. $\text{P} = \text{P/poly}$. nie
Klasa P/poly jest klasą złożoności niejednorodnej i zawiera nieprzeliczalnie wiele problemów, w szczególności problemy nieobliczalne.
8. $\text{AC} = \text{NC}$. tak
Z definicji obu klas $\text{AC} = \bigcup \text{AC}^n$, $\text{NC} = \bigcup \text{NC}^n$ oraz inkluzji $\text{NC}^n \subseteq \text{AC}^n \subseteq \text{NC}^{n+1}$.
9. $\text{BPP} \subseteq \text{NP}$. nie wiadomo
Wiemy, że $\text{BPP} \subseteq \text{PSPACE}$, więc brak inkluzji implikowałby $\text{NP} \neq \text{PSPACE}$ (co jest znanym problemem otwartym). Z drugiej strony, gdyby inkluzja zachodziła, możliwe byłyby dwa przypadki. (1) Pewien problem NP -zupełny L jest w BPP . Ale ta klasa jest zamknięta na dopełnienia, więc również $\bar{L}$ jest w BPP , co pociąga za sobą $\text{coNP} \subseteq \text{NP}$ (inny problem otwarty). (2) W przeciwnym razie inkluzja jest ostra, a ponieważ $\text{P} \subseteq \text{BPP}$, mielibyśmy $\text{P} \neq \text{NP}$.
10. $\text{PCP}(\mathcal{O}(1), \mathcal{O}(1)) = \text{P}$. tak
Wielomianowa maszyna używająca stałej liczby bitów losowych i pytająca o stałą liczbę bitów dowodu może być łatwo zasymulowana przez maszynę deterministyczną.
11. $\text{IP} = \text{coIP}$. tak
Z Tw. Shamira $\text{IP} = \text{PSPACE}$, a ta ostatnia klasa jest, jak już zauważyliśmy, zamknięta na dopełnienie.

12. Jeśli $P = NP$, to $EXPTIME = NEXPTIME$. tak
 Technika tzw. podkładania poduszek (ang. *padding*). Przypuśćmy, że L jest rozpoznawany przez niedeterministyczną maszynę w czasie c^{n^k} ; wtedy język $\{w\perp^{c|w|^k} : w \in L\}$ jest rozpoznawany przez niedeterministyczną maszynę w czasie $\mathcal{O}(n)$ (gdzie $\perp$ jest nowym symbolem). Hipotetyczny *deterministyczny* algorytm wielomianowy dla tego ostatniego języka dałby się łatwo przełożyć na deterministyczny algorytm wykładniczy dla języka L , w którym pierwsza faza polegałaby na dopisaniu odpowiedniej liczby $\perp$.
13. $AC^0 \subseteq Reg$, gdzie Reg to klasa języków regularnych. nie
 Łatwo jest skonstruować obwód o stałej głębokości rozpoznający palindromy.
14. $Reg \subseteq AC^0$. nie
 Wiadomo z wykładu, że język $PARITY$, czyli język regularny $0^*(10^*10^*)^*$ nie należy do AC^0 .
15. Klasa $DTIME(2^n)$ jest zamknięta na redukcje w $LOGSPACE$. nie
 Znowu *padding*. Z Tw. o hierarchii istnieje język L w $DTIME(2^{n^2}) - DTIME(2^n)$. Wtedy język $\{w\perp^{|w|^2} : w \in L\}$ jest w $DTIME(2^n)$. Jednak nietrudno jest (w $LOGSPACE$) zredukować język L do tego ostatniego języka.