Theoretische Grundlagen der Informatik II
Blatt 5

Markus Vieth, David Klopp, Christian Stricker

23. November 2015

Aufgabe 1

Aufgabe 2

Aufgabe 3

Vorüberlegung:

$$P = NP \Rightarrow SAT \in NP = P \Rightarrow SAT \in P$$

zu Zeigen:

Es existiert ein Algorithmus, welcher für eine erfüllbare aussagenlogische Formel Φ in polynomieller Zeit eine erfüllende Belegung ausgibt.

Beweis:

Sei die Laufzeit des polynomiellen SAT-Algorithmus = p Sei $X := \{x_i | \forall x_i \in \Phi\}$ die Liste aller Literale in Φ, |X| ist im worst-case = n.
if (SAT($\Phi(X)$){ //Teste ob Φ erfüllbar ist; (Laufzeit ist p) for(int i=0; i < |X|; i++) { //Setze jedes x_i auf einen festen Wert, (im worst-case n Durchläufe) X[i]=0; //Teste ob Φ erfüllbar, wenn $x_i = 0$ if(!(SAT($\Phi(X)$))) { //Wenn nicht: (Laufzeit = p) X[i]=1; //, dann muss x_i gleich 1 sein } //Wenn SAT mit $x_i = 0$ erfüllbar ist, bleibt x_i gleich 0 } //Wenn dies für alle x_i erfolgreich durchgeführt wurde, besitzt X eine erfüllende Belegung String result ="“; for (int i = 0; i < |X| ; i++) result = result+X[i]+"“; return result; } else return "nicht erfüllbar“;
Die erste Überprüfung dauert p lange, die folgende Schleife hat im worst-case n Durchläufe und in jedem Durchlauf einen SAT-Aufruf mit der Laufzeit p, somit beträgt die Laufzeit $T(n) = p + p \cdot n \Rightarrow$ die Laufzeit des oben genannten Algorithmus ist selbst wieder polynomiell, da die Laufzeit höchstens ein Polynom vom Grad $grad(p) + 1$ ist. Das die Ausgabe ein legitimes Ergebnis ist, ist trivial und geht aus dem Algorithmus hervor.