Aufgabe 1

SPEHAPF $\preccurlyeq_p$ ALHAPF

Wir haben also eine Instanz von SPEHAPF gegeben, also einen Graphen G und zwei Knoten a und b. Wir fügen zwei neue Knoten v_1 und v_2 hinzu. Wir fügen eine Kante von v_1 zu a hinzu und eine weitere von b zu v_2. Wir nennen den neuen Graphen G'

Jeder hamiltonsche Pfad in G' muss mit v_1 beginnen, da v_1 von keinem anderen Knoten erreichbar ist und muss mit v_2 enden, da kein anderer Knoten von v_2 erreichbar ist. Jede Lösung von ALHAPF auf G' entspricht genau einem hamiltonschen Pfad von a nach b in G, denn jeder hamiltonsche Pfad in G' muss mit v_1 beginnen und der einzig mögliche Nachfolger, nach Konstruktion, ist a und der Pfad mit v_2 enden muss und der einzig mögliche Vorgänger b ist. Jeder hamiltonsche Pfad in G' ist also ein hamiltonscher Pfad zwischen a und b in G, wenn man den ersten und letzten Knoten weglässt. Damit gilt natürlich auch, dass wenn ALHAPF keinen Pfad in G' findet, es auch keinen in G geben kann.

ALHAPF $\preccurlyeq_p$ SPEHAPF

Wir haben eine Instanz von ALHAPF gegeben, die wir mit SPEHAPF lösen wollen. Wir fügen zwei neue Knoten v_1 und v_2 in den Graph ein, so dass es eine Kante von v_1 zu jedem Knoten des ursprünglichen Graphen gibt und eine Kante von jedem ursprünglichen Knoten zu v_2. Nun lassen wir SPEHAPF laufen mit v_1 als a und v_2 als b. Offensichtlich gilt wenn es vor der Transformation einen Hamiltonschen Pfad gab, dann gibt es nun immer noch einen solchen Pfad, nur dass er mit v_1 beginnt und mit v_2 endet. Wenn es vorher keinen hamiltonschen Pfad gab, dann kann es jetzt immer noch keinen geben, da jeder hamiltonsche Pfad im transformierten Graphen mit v_1 beginnen muss und mit v_2 enden muss.

Aufgabe 2

(a)

$$
\begin{aligned}
GP := \{(x_1,\dots,x_n,U_1,\dots,U_m)\ |\exists c_1,\dots c_n \in \mathbb{Z},\ &\text{die die Ungleichungen } U_1,\dots,U_m \\
&\text{in den Variablen } x_1,\dots,x_n \text{ erfüllen}\}
\end{aligned}
$$

(b) Zeige $GP \in$ NP:
Sei $(x_1,\dots,x_n,U_1,\dots,U_m)$ gegeben. Wähle nichtdeterministisch $c_1,\dots c_n \in \mathbb{Z}$. Überprüfe, ob diese Werte, die Ungleichungen $U_1,\dots,U_m$ erfüllen.

Zeige nun 3-SAT $\preccurlyeq_p GP$:
Die Idee hierbei ist eine Formel in eine Menge von Ungleichungen zu überführen, sodass eine erfüllende Belegung in kanonischer Weise eine Lösung liefert und umgekehrt. Sei F also eine Formel in 3-KNF. Bilde folgende Ungleichungen für jede vorkommende Variable x in F:

$$x \geq 0 \tag{1}$$
$$-x \geq -1 \tag{2}$$
$$x + \overline{x} \geq 1 \tag{3}$$
$$-x - \overline{x} \geq -1 \tag{4}$$

Diese Ungleichungen fordern, dass jedes Literal als 'wahr' (1) oder 'falsch' (0) interpretiert wird. Weiterhin muss entweder ein Literal l oder dessen Negation $\bar{l}$ 'wahr' sein. Nun werden noch alle Klauseln $c = l_1 \vee l_2 \vee l_3$ aus F umgewandelt in

$$l_1 + l_2 + l_3 \geq 1. \tag{5}$$

Die Reduktion lautet demnach: $F \mapsto (x_1, \overline{x_1}, \dots x_n, \overline{x_n}, U_1, \dots, U_m)$. Diese Umwandlung ist in polynomieller Zeit möglich.

- $F \in$ 3- SAT:
  Sei β eine erfüllende Belegung für F. Definiere $c_1, \ldots, c_{2n} \in \mathbb{Z}$, mit

  $$c_{2i} := \beta(\overline{x_i}) \text{ und } c_{2i-1} := 1 - c_{2i} \; \forall i = 1, \ldots, n.$$

  $c_1, \ldots, c_{2n}$ erfüllen trivialerweise die Ungleichungen 1, 2, 3, 4 für jede Variable, da β wohldefiniert ist. Außerdem wird mindestens ein Literal l jeder Klausel von β auf 'wahr' (1) gesetzt. Damit ist wegen Ungleichung 1 auch Ungleichung 5 für jede Klausel erfüllt. Insgesamt gilt also $c_1, \ldots, c_{2n}$ erfüllen die Ungleichungen $U_1, \ldots, U_m$.
- $(x_1, \overline{x_1}, \ldots x_n, \overline{x_n}, U_1, \ldots, U_m) \in GP$:
  Sei $c_1, \ldots, c_{2n}$ eine Lösung für $U_1, \ldots, U_m$ und F die ursprüngliche Formel in 3-KNF. Definiere eine Belegung β, sodass

  $$\beta(\overline{x_i}) := c_{2i} \; \forall i = 1, \ldots, n.$$

  Wegen den Ungleichungen 1, 2, 3, 4 für jede Variable, ist $\beta(l) \geq 0 \; \forall$ Literale l. Da $c_1 \ldots, c_{2n}$ Ungleichung 5 für jede Klausel in F erfüllen, gilt dass mindestens ein Literal l pro Klausel auf 'wahr' (1) gesetzt wird. Damit ist β eine erfüllende Belegung für F.

Aufgabe 3

Betrachte

$$\text{P-UNIV} := \{(\langle M \rangle, x, 1^m) | M \text{ ist NTM und } M \text{ akzeptiert } x \text{ in } m \text{ Schritten}\}.$$

Laut Vorlesung ist P-UNIV NP-vollständig und bei Betrachtung des Beweises hierfür, sieht man, dass insbesondere P-UNIV $\in$ NTime(n) gilt. Wegen der Annahme $P = NP$ gibt es ein $b \in \mathbb{N}$ mit P-UNIV $\in$ DTime(n^b). Sei $M_{\text{P-UNIV}}$ die DTM die P-UNIV entscheidet.

Sei $L \in \text{NTime}(n^k)$. Wir konstruieren DTM M' auf Eingabe x:

1. Schreibe $(\langle M \rangle, x, 1^{|x|^k})$ auf das Arbeitsband.
2. Simuliere $M_{\text{P-UNIV}}$ mit dem Arbeitsband als Eingabe.

M' entscheidet genau L, da $L \in \text{NTime}(n^k)$. Der erste Schritt ist in $\mathcal{O}(n^k)$. Der zweite ist in $\mathcal{O}((n^k)^b \log((n^k)^b)) = \mathcal{O}(n^{k \cdot b} \log(n^{k \cdot b}))$, da die Simulation einer Turingmaschine in $\mathcal{O}(n \log n)$ liegt, $M_{\text{P-UNIV}}$ Laufzeit in $\mathcal{O}(n^k)$ hat und die Eingabe für die Simulation in $\mathcal{O}(n^k)$ liegt. Weiterhin gilt

$$\text{DTime}(\mathcal{O}(n^{k \cdot b} \log(n^{k \cdot b}))) \subseteq \text{DTime}(n^{2bk}).$$

Also $L \in \text{DTime}(n^{2bk})$. Damit ist die Aussage gezeigt.