**RWTH Aachen** **WS 2003/04**
**Lehrgebiet Theoretische Informatik** Klausur
Mölle—Richter—Rossmanith 29.03.2004

**Diplom-Vorprüfung zur Vorlesung Berechenbarkeit und Komplexität**

Lösungsvorschlag

**Aufgabe 1 (10 Punkte)**
Dies ist ein semantisch nicht-triviales Entscheidungsproblem:

1. Es ist nicht leer, denn Turingmaschinen für die leere Sprache gehören natürlich dazu.
2. Es enthält nicht die Kodierungen aller Turingmaschinen, da die Sprache $\{0,1\}^*$ nicht gut komprimierbar ist.
3. Ob die Kodierung einer Turingmaschine dazugehört, hängt nur von der von ihr akzeptierten Sprache ab (nämlich ob diese gut komprimierbar ist).

Nach dem Satz von Rice ist die gegebene Sprache daher nicht rekursiv.

**Aufgabe 2 (10 Punkte)**

a) Wahr, denn wir können Turingmaschinen für beide Sprachen abwechselnd schrittweise simulieren und akzeptieren, wenn eine der simulierten Maschinen akzeptiert.

b) Wahr, denn wir können Turingmaschinen für beide Sprachen nacheinander simulieren und akzeptieren, wenn beide Simulationen im akzeptierenden Zustand enden.

c) Falsch, denn dann wäre $\{0,1\}^* - L_H$ ja rekursiv aufzählbar.

d) Wahr, denn eine NTM kann eine Zerlegung $w = w_1 \dots w_k$ raten und anschließend $w_i \in A$ für alle $i$ verifizieren (wie in Teil b).

**Aufgabe 3 (10 Punkte)**
Wir identifizieren eine Variable $x_i$ mit jedem Knoten in $V = \{v_1, \dots, v_n\}$. Für jeden Knoten sorgt die Ungleichung $0 \le x_i \le 1$ dafür, daß der Knoten in $C$ liegt oder nicht. Ungleichungen $x_i + x_j \ge 1$ für jede Kante $\{x_i, x_j\}$ sorgen dafür, daß jede Kante abgedeckt wird. Schließlich sorgt die Ungleichung $x_1 w(v_1) + \cdots + x_n w(v_n) \le b$ dafür, daß das Maximalgewicht nicht überschritten wird. So haben wir eine Bijektion zwischen Lösungen des WVC und des zugehörigen ILP: $v_i$ gehört zu $C$ genau dann, wenn $x_i = 1$. Wir haben nur polynomiell viele Variablen und Ungleichungen verwendet, und das ILP ist leicht aus der WVC-Instanz zu erzeugen.

**Aufgabe 4 (10 Punkte)**
Da es nur polynomiell viele Konfigurationen mit logarithmisch viel Platz gibt, müssen wir nur feststellen können, ob eine $O(\log n)$ platzbeschränkte NTM innerhalb von polynomiell vielen Schritten akzeptiert. Dafür gibt es viele Möglichkeiten:
Man kann z.B. den ganzen Berechnungsgraphen konstruieren und dann feststellen, ob eine akzeptierende Konfiguration von der Startkonfiguration erreichbar ist.