Grundzüge der Theoretischen Informatik: Präsenzblatt 10

Beachten Sie: Die Präsenzblätter und -lösungen wurden von den Tutoren erstellt und können Fehler enthalten. Diese Blätter sind nicht für das Bestehen notwendig und dienen allein ihrer selbstständigen Übung.

Aufgabe 1 (Inklusionen).

(a) Sortieren Sie die folgenden Mengen bezüglich Inklusion so gut es Ihnen möglich ist:

- P
- $\mathsf{DTime}(\log n)$
- $\mathsf{EXP} := \bigcup_{k \in \mathbb{N}} \mathsf{DTime}(2^{(n^k)})$
- REC
- $\mathsf{DSpace}(n)$
- $\mathsf{co-NP} := \{L \mid \overline{L} \in \mathsf{NP}\}$
- $\mathsf{NSpace}(n^3)$
- $\mathsf{co-RE} := \{L \mid \overline{L} \in \mathsf{RE}\}$
- $\mathsf{DSpace}(\log n)$
- $\mathsf{PSPACE} := \bigcup_{k \in \mathbb{N}} \mathsf{DSpace}(n^k)$
- NP
- RE

Hinweis: Sie werden nicht alle Klassen bzgl. Inklusion miteinander vergleichen können.

(b) Von welchen Inklusionen können Sie Striktheit zeigen?

(c) Was ändert sich an der Ordnung, wenn $\mathsf{P} = \mathsf{NP}$?

(d) **Bonus:** Zeigen Sie: $\mathsf{co-NP} \neq \mathsf{NP} \Rightarrow \mathsf{P} \neq \mathsf{NP}$

Aufgabe 2 (NP-vollständigkeit für Anfänger).

Sei DNF die Menge aller aussagenlogischen Formeln in disjunktiver Normalform.
Betrachten Sie das folgende Problem:

$$L = \{F \in \mathsf{DNF} \mid F \text{ ist keine Tautologie}\}$$

Zeigen Sie: L ist NP-vollständig.

Aufgabe 3 (Vorsicht bei DNF)**.**
Betrachten Sie nochmals die Sprache L aus Aufgabe 2. Wir erhalten das folgende Problem, wenn wir das Prädikat ändern:

$$L' = \{F \in \mathsf{DNF} \mid F \text{ ist erfüllbar}\}$$

(a) Zeigen Sie: $L' \in \mathsf{P}$.

(b) Warum folgt hieraus **nicht**, dass $\mathsf{P} = \mathsf{NP}$?

Aufgabe 4 (NP-vollständigkeit für Fortgeschrittene)**.**
In dieser Aufgabe betrachten wir die Entscheidungsversion des Problems des Handlungsreisenden. Gegeben sind:

- Ein ungerichteter Graph $G = (V, E)$ (z.B. ein Straßennetz aus dem ein Theoretiker alle Einbahnstraßen entfernt hat)
- Eine Kostenfunktion für die Kanten $c : E \to \mathbb{N}$ (z.B. Benzinkosten, Länge oder Fahrzeit pro Straße)
- Eine natürliche Zahl $b \in \mathbb{N}$.

Das Ziel des Handlungsreisenden ist es, herauszufinden, ob es die Möglichkeit einer Rundreise R durch den Graphen gibt, wobei jeder Knoten genau einmal besucht wird, und der Start- und der Endknoten identisch sind. Außerdem dürfen die Gesamtkosten $c(R)$ der Rundreise R die Grenze b nicht übersteigen. Dabei berechnet sich $c(R)$ wie folgt:

$$c(R) = \textstyle\sum_{e \in R} c(e)$$

Nun soll die NP-vollständigkeit dieses Problems gezeigt werden. Gehen Sie dabei wie folgt vor:

(a) Definieren Sie das Problem des Handlungsreisenden TSP formal als Sprache.

(b) Zeigen Sie, dass $\mathsf{TSP} \in \mathsf{NP}$ indem Sie eine nichtdeterministische Turingmaschine konstruieren, die das Problem in polynomieller Zeit löst.

(c) Zeigen Sie, dass TSP NP-schwer ist, in dem Sie von einem Ihnen bereits bekannten NP-schweren Problem many-one reduzieren.
Tipp: Möglicherweise hilft es, wenn Sie sich das letzte Übungsblatt noch einmal anschauen.

Aufgabe 5 (NP-vollständigkeit für Spielkinder).
In dieser **Bonusaufgabe** geht es darum zu zeigen, dass Super Mario Bros ein durchaus schwieriges Spiel/Problem ist. Genauer gesagt, werden Sie zeigen, dass Super Mario Bros tatsächlich NP-schwer ist. Dazu muss das Spiel erstmal als mathematisches Problem formuliert werden. Wir verwenden die folgende Notation:

- W ist die Menge aller möglichen "röhrenfreien" Welten/Level in Super Mario Bros.
- Für einen bestimmten Punkt p innerhalb eines Levels $L \in \mathsf{W}$ schreiben wir $p \in L$.
- Für $L \in \mathsf{W}$ definieren wir die Relation $\rightarrow_L \subseteq L \times L$, so dass

$$p \rightarrow_L q :\Leftrightarrow \text{Gestartet von } p \text{ kann der } \textbf{kleine} \text{ Mario } q \text{ erreichen.}$$

  Beachten Sie, dass $\rightarrow_L$ in der Regel nicht symmetrisch ist.

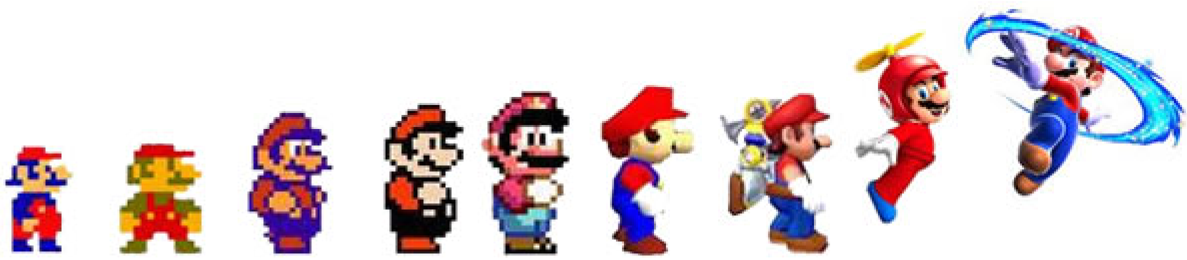
Nun können wir Super Mario als Entscheidungsproblem definieren:

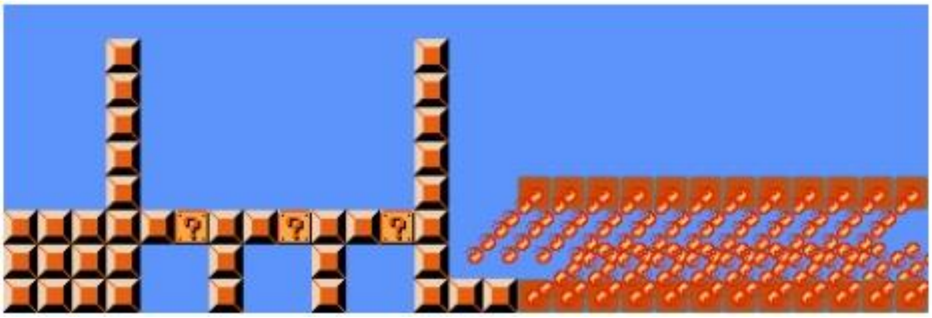
$$\mathsf{SuperMario} := \{(L, s, t) \mid L \in \mathsf{W}, s, t \in L \wedge s \rightarrow_L t\}$$

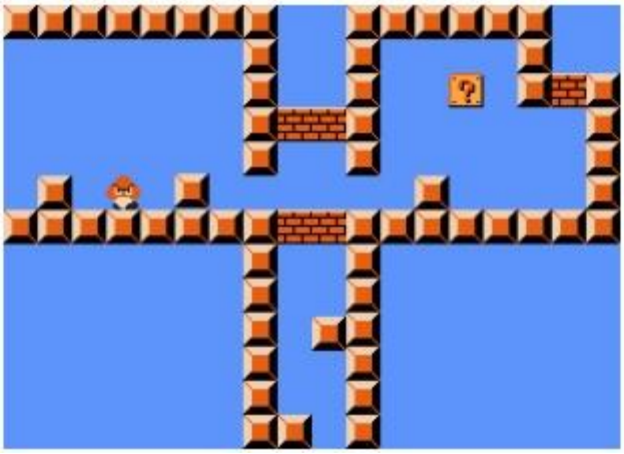
Um zu zeigen, dass SuperMario NP-schwer ist, empfiehlt es sich, von dem Problem 3SAT zu reduzieren, wobei 3SAT die Menge aller erfüllbaren aussagenlogischen Formeln in "3CNF", d.h. in konjunktiver Normalform beschreibt, bei denen jede Klausel **genau** 3 Literale enthält.
Ihre Aufgabe ist es nun, die Gadgets auf der nächsten Seite zu benutzen um - gegeben eine Formel F in 3CNF - ein Level L_F in Super Mario Bros, samt Start- und Endpunkt s und t zu konstruieren, so dass folgendes Äquivalenz gilt:

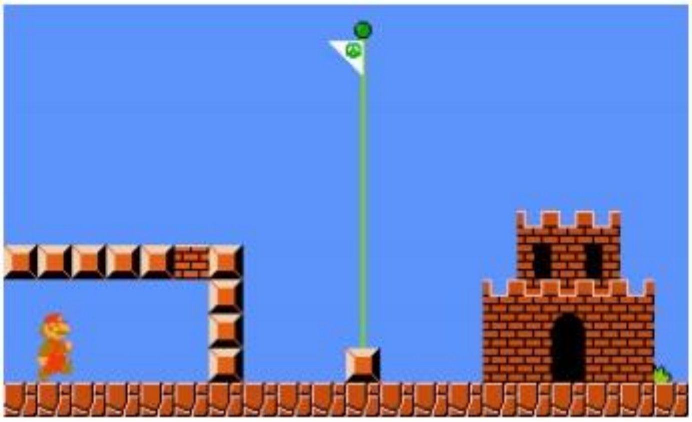
$$(L_F, s, t) \in \mathsf{SuperMario} \Leftrightarrow F \in \mathsf{3SAT}$$

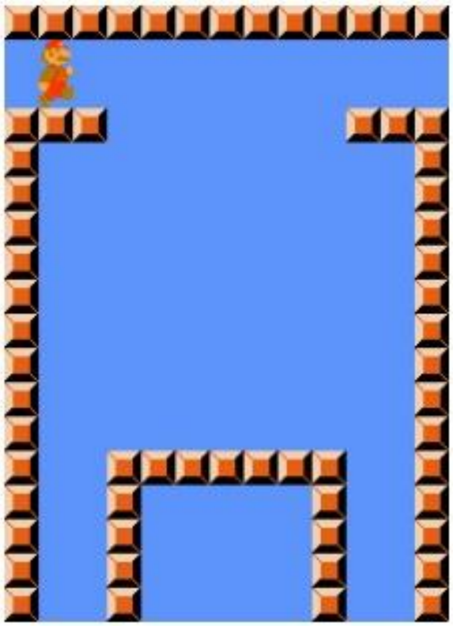
Hinweis: Wir haben aus Gründen der Einfachheit auf die Formalisierung der "Größe" eines Levels verzichtet. Trotzdem müssen Sie darauf achten, dass Ihre Konstruktion nicht "explodiert". Vielleicht schaffen Sie es auch zu beweisen, dass die Größe ihrer Konstruktion nur polynomiel - bei geeigneter Formalisierung der Größenfunktion - in der Eingabegröße wächst.











Referenzen:

- Aloupis, Greg, et al. "Classic Nintendo games are (NP-) hard." *arXiv preprint arXiv:1203.1895* (2012).
- http://4.bp.blogspot.com/_xB8Z7B62t6M/TUj0YK1qGiI/AAAAAAAAAPU/MiT5W1p4Fz8/s1600/super_mario_timeline.jpg