Algorithmen & Komplexität

Johannes Lengler
Institut für Theoretische Informatik

Komplexitätstheorie

P = *effizient entscheidbare Probleme*

NP = *(einseitig) effizient verifizierbare Probleme*

$P \stackrel{?}{=} NP$ → *1 Million US-$ (Clay-Foundation)*

[eines von sieben Millennium-Problemen]

Definition:

Seien $f, g : \{0,1\}^* \to \{0,1\}$ Entscheidungsprobleme. Wir sagen, dass *f* *polynomiell reduzierbar* auf *g* ist, $f \leq_p g$, falls es eine Funktion $\kappa : \{0,1\}^* \to \{0,1\}^*$ gibt, die in polynomieller Zeit berechnet werden kann, mit:

$$\forall x \in \{0,1\}^* : \ f(x) = g(\kappa(x)).$$

Definition:

Ein Entscheidungsproblem *g* heisst *NP-schwer* genau dann, wenn $f \leq_p g$ für alle Probleme f in NP gilt. Ist zusätzlich g in NP, so heisst g *NP-vollständig*.

SAT ist NP-vollständig

Theorem (Cook-Levin, 1971):

- SAT ist NP-vollständig.

Satz: $\mathrm{SAT} \leq_p 3\text{-}\mathrm{SAT}$

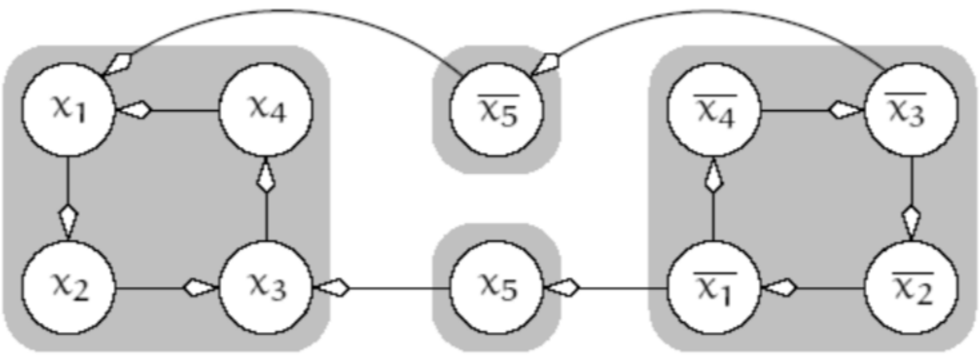
Korollar: 3-SAT ist NP-vollständig.

Satz: 2-SAT ist in P.

Beweis:
Konstruiere einen gerichteten Graphen G_F = (V,A) mit:

- $V := X \cup \bar{X} = \{x_1, \ldots, x_n, \overline{x_1}, \ldots, \overline{x_n}\}.$
- $A := \bigcup_{(x \vee y) \text{ Klausel in } F} \{(\bar{x}, y), (\bar{y}, x)\}.$

Beispiel:



$$F = (\bar{x_1} \vee x_2) \wedge (\bar{x_2} \vee x_3) \wedge (\bar{x_3} \vee x_4) \wedge (\bar{x_4} \vee x_1) \wedge (\bar{x_5} \vee x_3) \wedge (x_5 \vee x_1)$$

Behauptung: F ist genau dann *nicht* erfüllbar, wenn es eine Variable x gibt, sodass x und $\overline{x}$ auf einem (gerichteten) Zyklus liegen.

Theorem (Cook-Levin, 1971):

- SAT ist NP-vollständig.

Satz: $\mathrm{SAT} \leq_p 3\text{-}\mathrm{SAT}$

Korollar: 3-SAT ist NP-vollständig.

Satz: $3\text{-}\mathrm{SAT} \leq_p 3\text{-}\mathrm{COL}$

Korollar: 3-COL ist NP-vollständig.

Nullstelle Mod N ist NP-vollständig

Theorem: Die Sprache NST = **„**Nullstelle mod n“ ist NP-vollständig.

Input: Ein Polynom P mit Variablen $x_1,...,x_s$ und ein $n \in \mathbb{N}$.
Output: 1, falls P mod n eine Nullstelle hat. 0 sonst.

Nullstelle Mod n ist NP-vollständig

Theorem: Die Sprache NST = „Nullstelle mod n“ ist NP-vollständig.

Beweis:

- Das Problem ist in NP. (!!)
- Reduktion $3\text{-SAT} \leq_p \text{NST}$: Sei $F = D_1 \wedge \ldots \wedge D_m$ eine Boolesche Formel in 3-KNF.
- Zu einem Literal L setzen wir $P_L := \begin{cases} x_i, & \text{falls } L = x_i, \\ (1 - x_i), & \text{falls } L = \overline{x}_i. \end{cases}$
- Für die i-te Klausel $D_i = (L_1 \vee L_2 \vee L_3)$ definieren wir
$$Q_i := 1 - (1 - P_{L_1})(1 - P_{L_2})(1 - P_{L_3}).$$
- Wir setzen $\kappa(F) := \big(-1 + \prod_{i=1}^m Q_i \ , \ 2 \big).$
- Eine Belegung $f : \{x_1, \ldots, x_s\} \to \{0, 1\}$ erfüllt F genau dann wenn für diese Werte $\prod_{i=1}^m Q_i \equiv 1 \bmod 2$ gilt.
- Also $3\text{-}SAT(F) = NST(\kappa(F))$.

Reduzierbarkeit

Definition:

Seien $f, g : \{0,1\}^* \to \{0,1\}$ Entscheidungsprobleme. Wir sagen, dass *f* *polynomiell reduzierbar* auf *g* ist, $f \leq_p g$, falls es eine Funktion $\kappa : \{0,1\}^* \to \{0,1\}^*$ gibt, die in polynomieller Zeit berechnet werden kann, mit:

$$\forall x \in \{0,1\}^* : \; f(x) = g(\kappa(x)).$$

Alternative Definition (truth table reduction):

Seien $f, g : \{0,1\}^* \to \{0,1\}$ Entscheidungsprobleme. Eine *m-(truth table) reduction* von f auf *g* ist ein Tupel

$$\kappa_1, \ldots, \kappa_m : \{0,1\}^* \to \{0,1\}^*$$

und ein $T : \{0,1\}^m \to \{0,1\}$, alles in polynomieller Zeit berechenbar, sodass

$$\forall x \in \{0,1\}^* : \; f(x) = 1 \iff T(g(\kappa_1(x)), \ldots, g(\kappa_m(x))) = 1.$$

Manchmal erlaubt man sogar, dass m von |x| abhängt.

Weitere Verallgemeinerungen möglich

Hamiltonkreis und Hamiltonpfad

Theorem:
Es gibt in beide Richtungen eine polynomielle n^2-Reduktion zwischen den Sprachen *Hamiltonkreis* und *Hamiltonpfad*, wobei n die Grösse des Input-Graphen ist.

Korollar: *Hamiltonkreis* ist in P $\iff$ *Hamiltonpfad* ist in P.

Theorem (ohne Beweis): *Hamiltonkreis* ist NP-vollständig.

Korollar: *Hamiltonpfad* ist in P $\iff$ P = NP.

(Tatsächlich ist auch *Hamiltonpfad* NP-vollständig.)

Probleme in NP

Probleme in NP:

NP-vollständig

- alle Probleme in P
- GRAPH-ISOMORHPHISMUS
  $L = \{(G, H) \mid G \text{ und } H \text{ sind isomorphe Graphen}\}$
- 3-COL
  $L = \{G \mid \ G \text{ ist mit 3 Farben färbbar}\}$
- CLIQUE
  $L = \{(G, m) \mid \text{Es gibt } m \text{ paarweise adjazente Knoten in } G\}$
- HAMILTONKREIS
  $L = \{G \mid G \text{ hat einen Kreis, der jeden Knoten genau einmal besucht}\}$
- NULLSTELLE-MOD-N
  $L = \{(F, n) \mid F \text{ ist ein Polynom, das über } \mathbb{Z}/n\mathbb{Z} \text{ eine Nullstelle hat}\}$
  z.B.: $(x^2 + 1,\ 41) \in L, \qquad (x^2 + y^2 + z^2 - 7,\ 8) \notin L$
- SAT (SATISFIABILITY)
  $L = \{(F, n) \mid F \text{ ist erfüllbare aussagenlogische Formel in K-Normalform}\}$
  z.B.: $(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \overline{x}_3) \wedge (\overline{x}_1) \wedge (\overline{x}_2) \notin L$

NP-vollständig

Theorem (Cook-Levin, 1971):

- SAT ist NP-vollständig

Theorem (Karp 1972):

Die folgenden 21 Probleme sind NP-vollständig:

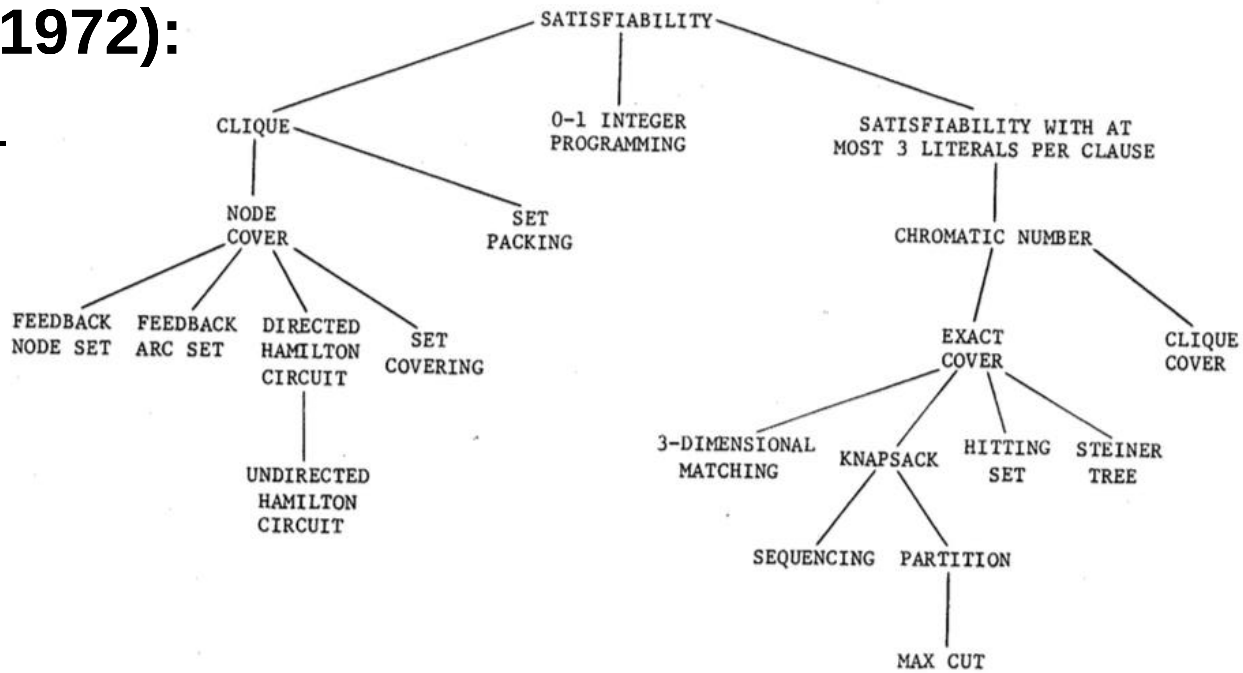


FIGURE 1 - Complete Problems

96

RICHARD M. KARP

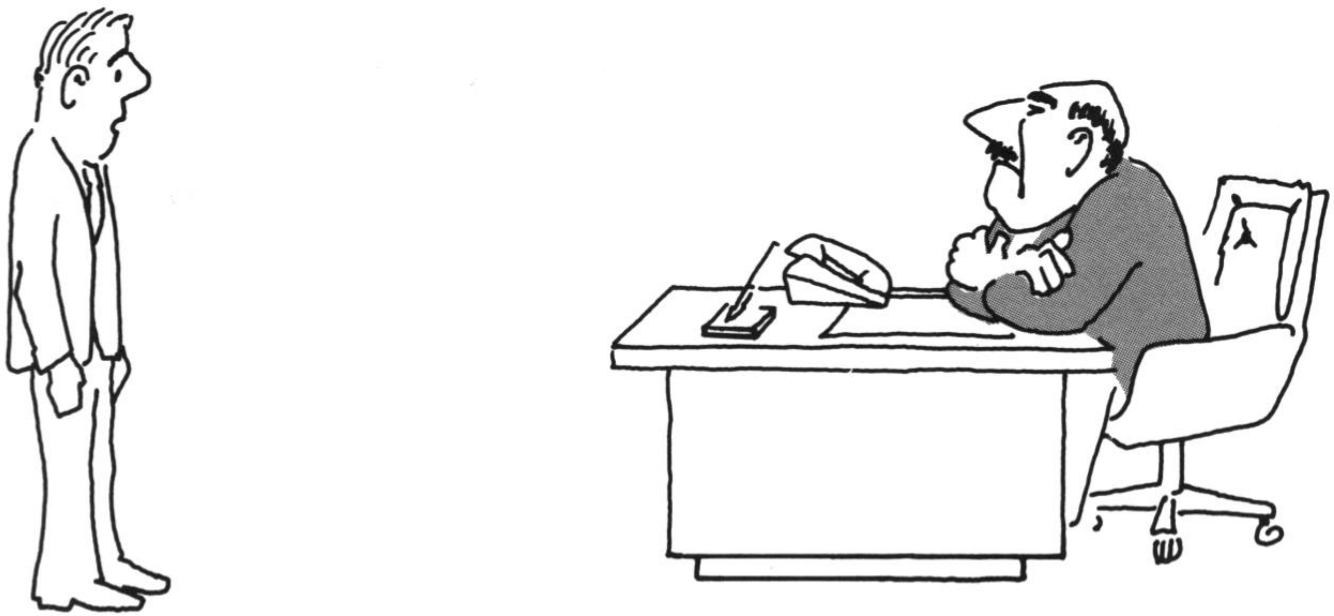
Classic Nintendo Games are (Computationally) Hard

Greg Aloupis* Erik D. Demaine† Alan Guo†‡ Giovanni Viglietta§

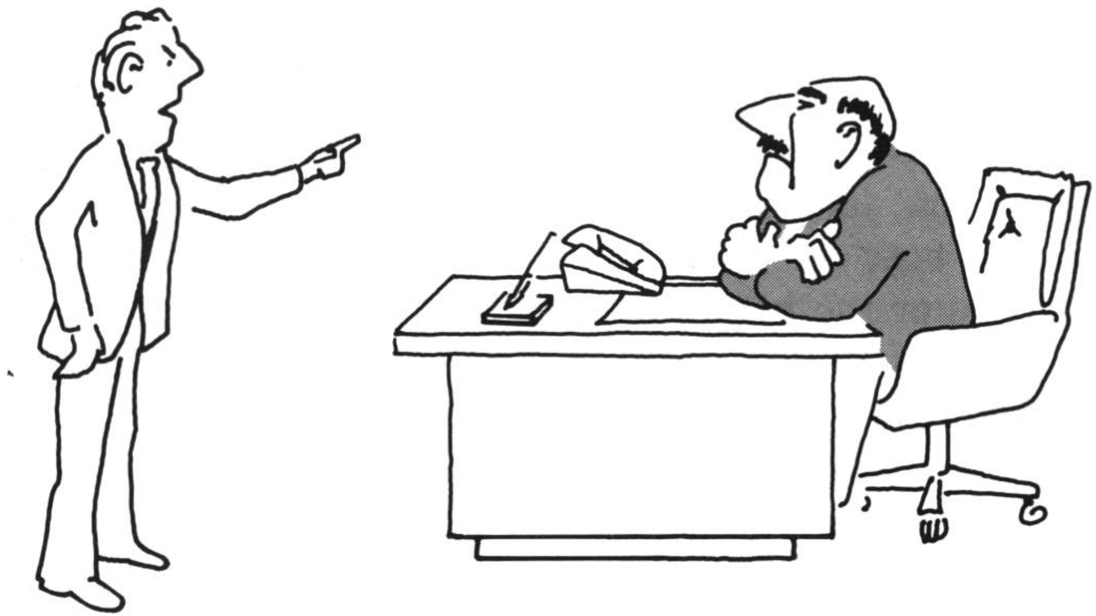
February 10, 2015

Abstract

We prove NP-hardness results for five of Nintendo's largest video game franchises: Mario, Donkey Kong, Legend of Zelda, Metroid, and Pokémon. Our results apply to generalized versions of Super Mario Bros. 1–3, The Lost Levels, and Super Mario World; Donkey Kong Country 1–3; all Legend of Zelda games; all Metroid games; and all Pokémon role-playing games. In addition, we prove PSPACE-completeness of the Donkey Kong Country games and several Legend of Zelda games.



“I can’t find an efficient algorithm, I guess I’m just too dumb.”



"I can't find an efficient algorithm, because no such algorithm is possible!"

"I can't find an efficient algorithm, but neither can all these famous people."