



Technische
Universität
Braunschweig



Theoretische Informatik 2

Arne Schmidt

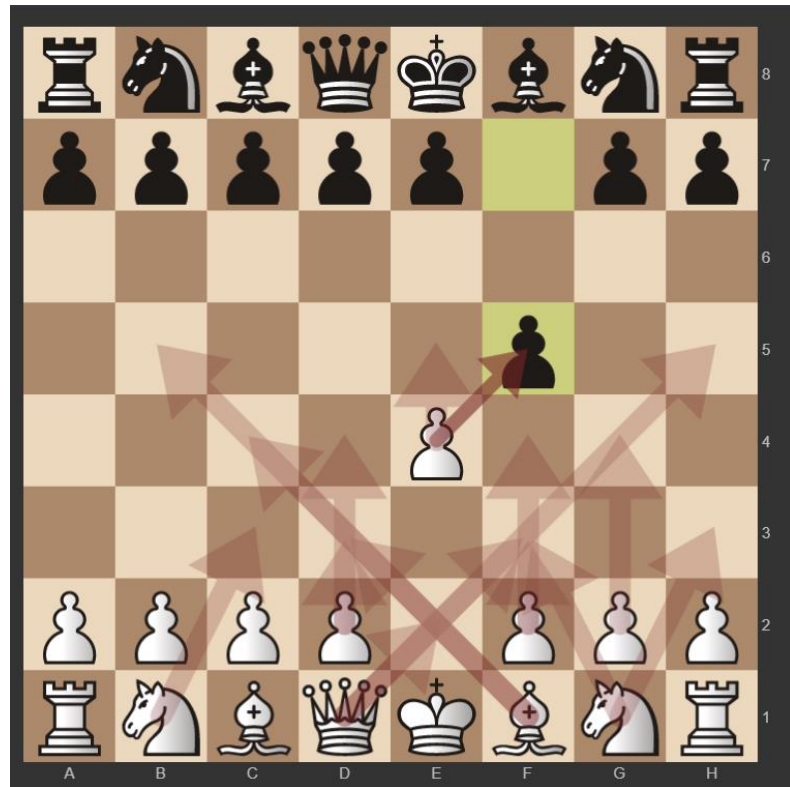
Schach

Kann Weiß in dieser Stellung immer gewinnen?

Zur Beantwortung dieser Frage, müsste man folgendes entscheiden:

- Es gibt einen Zug für Weiß, sodass
- Für alle Züge von Schwarz
- Ein Zug für Weiß existiert, sodass
- Für alle Züge von Schwarz
- Ein Zug für Weiß existiert, sodass
-
- Sodass Weiß Schwarz schachmatt setzt.

$$\exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \forall x_6 \exists x_7 \forall x_8 \dots x_\infty: \varphi(x_1, \dots, x_n) = 1$$



Heute



Kapitel 12 – PSPACE und der Satz von Savitch

Kapitel 12.1 – QBF ist PSPACE vollständig

Quantified Boolean Formula

Definition 12.1

Eine quantifizierte Boolesche Formel ist von der Form

$$F = Q_1 x_1 \dots Q_n x_n : G(x_1, \dots, x_n),$$

wobei $Q_i \in \{\forall, \exists\}$ Quantoren sind und G eine Boolesche Formel mit Variablen $x_1, \dots, x_n$ ist.

Der Präfix $Q_1 x_1 \dots Q_n x_n$ heißt auch **Quantorenblock** von F .

Der **Wahrheitswert** von F wird rekursiv bestimmt. Dazu sei

$$F_{(x_1=false)} = Q_2 x_2 \dots Q_n x_n : G(false, x_2, \dots, x_n)$$

$$F_{(x_1=true)} = Q_2 x_2 \dots Q_n x_n : G(true, x_2, \dots, x_n)$$

Dann ist F wahr wenn:

- $Q_1 = \forall$ und $F_{(x_1=false)} \wedge F_{(x_1=true)}$ ist wahr
- $Q_1 = \exists$ und $F_{(x_1=false)} \vee F_{(x_1=true)}$ ist wahr

Beispiel 12.2

Welche der folgenden Formeln haben den Wahrheitswert *true*?

$$F_1 = \forall x \exists y: (x \Leftrightarrow y)$$

$$F_2 = \exists x \forall y: (x \Leftrightarrow y)$$

F_1 ist true:

Egal wie wir x wählen, mit $y = x$ ist die Bedingung erfüllt.

F_2 ist false:

Egal wie wir x wählen, spätestens mit $y = \neg x$ scheitert die Formel.

Bemerkung 12.3

SAT ist ein Spezialfall von QBF mit nur Existenzquantoren (und in CNF).

Komplexität von QBF

Theorem 12.4

QBF ist PSPACE-vollständig.

~~Beweis später an der Tafel.~~

Skizze für PSPACE-schwere:

Transformiere eine polynomiell-platzbeschränkte NTM in eine QBF mit quadratischem blow-up.

Nicht-Determinismus



Kapitel 12.2 – Satz von Savatch

Satz von Savitch

Theorem 12.5

Sei $s: \mathbb{N} \rightarrow \mathbb{N}$ eine Platzschranke mit $s(n) \geq \log n$ für alle n . Es gilt

$$\text{NSPACE}(s) \subseteq \text{DSPACE}(s^2)$$

~~Beweis später an der Tafel.~~

Korollar 12.6

Es gilt $\text{PSPACE} = \text{NPSPACE}$ und $\text{EXPSPACE} = \text{NEXPSPACE}$.

Bemerkung 12.7

Der Satz von Savitch beweist nicht $L = NL$!

Kapitel 12.3 – Ein paar PSPACE-vollständige Probleme

Reguläre Sprachen

Universality

Gegeben: Ein endlicher Automat A über dem Alphabet Σ .

Frage: Gilt $\mathcal{L}(A) = \Sigma^*$?

Equivalence

Gegeben: Zwei endliche Automaten A und B .

Frage: Gilt $\mathcal{L}(A) = \mathcal{L}(B)$?

Inclusion

Gegeben: Zwei endliche Automaten A und B .

Frage: Gilt $\mathcal{L}(A) \subseteq \mathcal{L}(B)$?

Robot Motion Planning

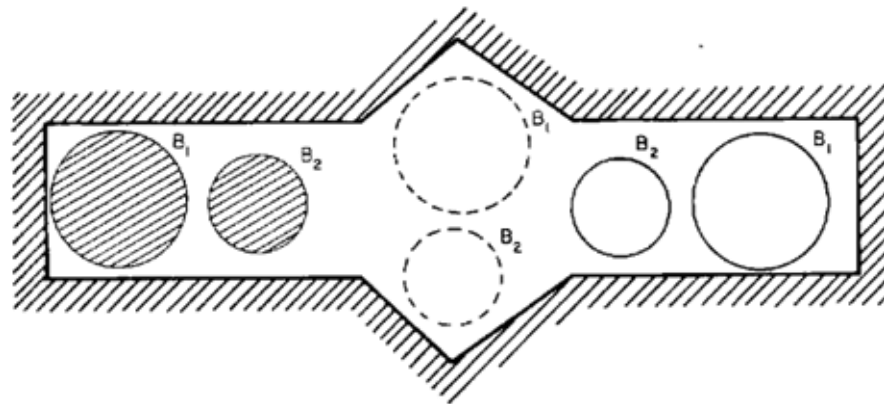
Jacob T. Schwartz

Computer Science Department
Courant Institute of Mathematical Sciences
New York University
New York, New York 10012

Micha Sharir

School of Mathematical Sciences
Tel Aviv University
Tel Aviv, Israel

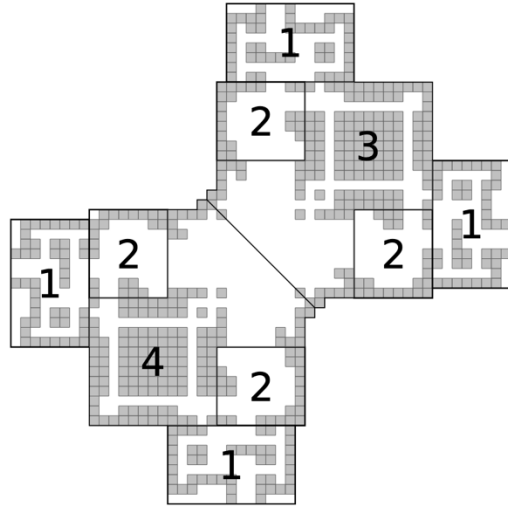
**On the Piano Movers' Problem: III.
Coordinating the Motion of Several Independent Bodies:
The Special Case of Circular Bodies Moving Amidst Polygonal Barriers**



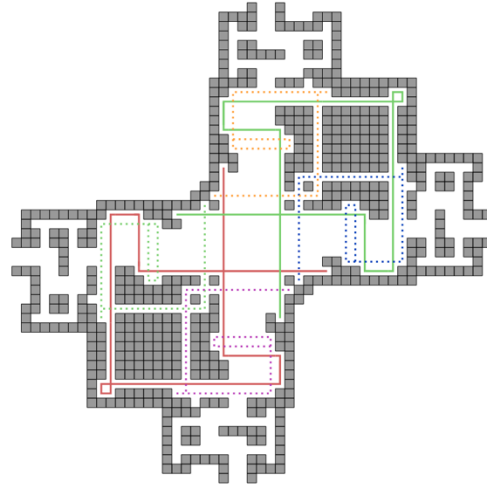
Polynomialzeit-Algorithmus bzgl. Komplexität der Umgebung, aber exponentiell in Anzahl Roboter.

TILT Rekonfiguration

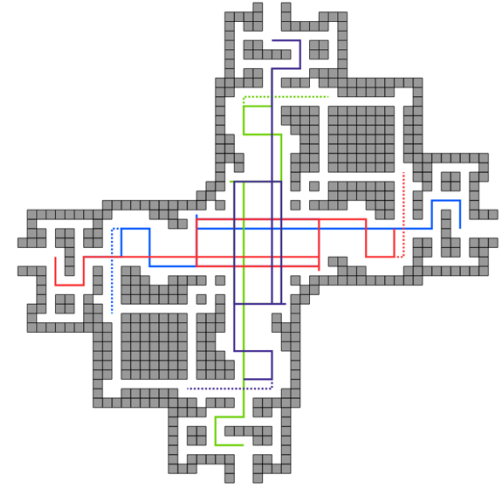
Roboter bewegen sich durch Kippbewegungen in die gleiche Richtung.



(a) Sections of Gadgets



(b) Paths of the State Tile



(c) Paths of the Robot Polyomino

Rush Hour



Classic Nintendo games are (computationally) hard

Greg Aloupis^{a,1}, Erik D. Demaine^b, Alan Guo^{b,*,2}, Giovanni Viglietta^c

^a *Chargé de Recherches du FNRS, Département d'Informatique, Université Libre de Bruxelles, Belgium*

^b *MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar St., Cambridge, MA 02139, USA*

^c *School of Electrical Engineering and Computer Science, University of Ottawa, Canada*

Darunter unter anderem:

- Legend of Zelda
- Donky Kong Country

“(Wie) gelange ich zum Ende des Levels?”

Nächstes Kapitel



Hierarchiesätze

Welche Inklusionen
sind echt?

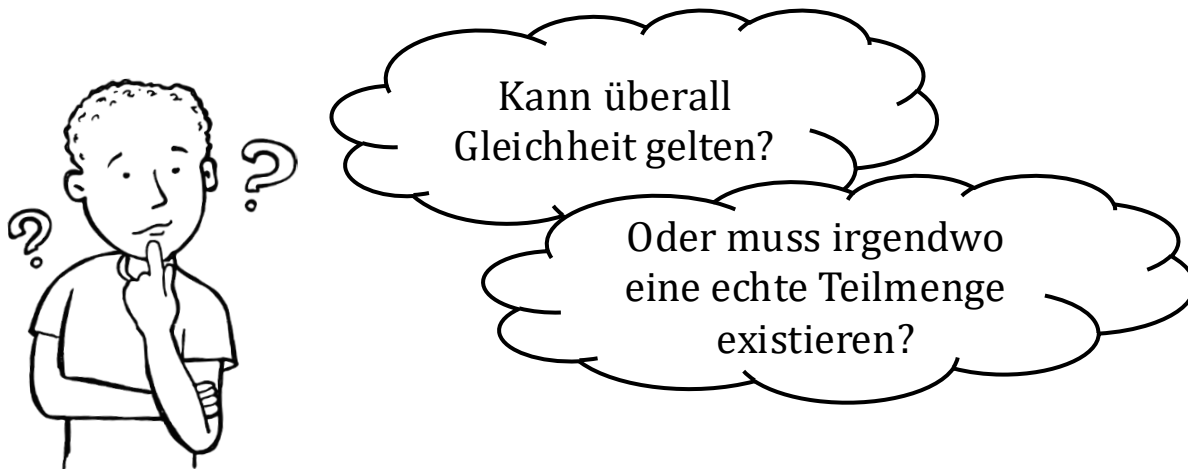
Inklusionen

Zur Erinnerung

Theorem 7.32

Die robusten Komplexitätsklassen bilden eine aufsteigende Kette,

$$L \subseteq NL \subseteq P \subseteq NP \subseteq PSPACE \subseteq NPSPACE \subseteq EXP \subseteq NEXP \subseteq EXPSPACE \subseteq \dots$$



Heute

Zur Erinnerung

Theorem 7.32

Die robusten Komplexitätsklassen bilden eine aufsteigende Kette,

$$L \subseteq NL \subseteq P \subseteq NP \subseteq PSPACE \subseteq NPSPACE \subseteq EXP \subseteq NEXP \subseteq EXPSPACE \subseteq \dots$$



Wir zeigen heute: Es kann nicht
überall Gleichheit gelten.

Wir wissen nur nicht, wo in dieser
Kette echte Teilmengen existieren.

Genauer zeigen wir:
 $NL \subsetneq PSPACE \subsetneq EXPSPACE$,
 $P \subsetneq EXP$, $NP \subsetneq NEXP$

Kapitel 13 – Hierarchiesätze

Klein-o-Notation

Definition 13.1

Zu einer Funktion $f: \mathbb{N} \rightarrow \mathbb{N}$ ist

$$o(f) = \{ g: \mathbb{N} \rightarrow \mathbb{N} \mid \forall m \in \mathbb{N}, \exists n_0 \in \mathbb{N}: \forall n \geq n_0: m \cdot g(n) < f(n) \}$$

die Klasse der Funktionen, die asymptotisch **echt kleiner als** f sind.

Beispiel 13.2:

- $n^2 \in O(n^3)$ und $n^2 \in o(n^3)$, aber $n^2 \notin o(n^2)$
- $2^n \in o(3^n)$
- $\log n \in o(n)$, und sogar $\log^a(n) \in o(n^b)$ für bel. $a, b \in \mathbb{R}^+$

Platzkonstruierbarkeit

Definition 13.3

Eine Funktion $f: \mathbb{N} \rightarrow \mathbb{N}$ heißt **platzkonstruierbar**, wenn

- $\forall n \in \mathbb{N}: f(n) \geq \log n$
- Es eine TM M mit read-only Eingabe und write-only Ausgabe gibt, die auf Eingabe y der Länge $|y|$ eine Ausgabe der Länge $f(|y|)$ zurückgibt und Platzverbrauch (auf den Arbeitsbändern) $O(f)$ hat.

Beispiel 13.4:

Die folgenden Funktionen sind platzkonstruierbar

$$\log n, n, n^2, n^3, \dots, 2^n, 2^{n^2}, 2^{n^3}, \dots$$

Platzhierarchiesatz

Satz 13.5

Es sei $f: \mathbb{N} \rightarrow \mathbb{N}$ platzkonstruierbar. Dann gilt

$$DSPACE(o(f)) \subsetneq DSPACE(O(f))$$

Beweisskizze:

- Gib eine Sprache $\mathcal{L}(M) \in DSPACE(O(f)) \setminus DSPACE(o(f))$ an.
- Konstruiere dazu Maschine M mit Platzverbrauch $O(f)$.
- Zeige dann, $\mathcal{L}(M) \notin DSPACE(o(f))$ mittels Diagonalisierung

Folgerungen

Korollar 13.6

Es gilt

- a) $L \subsetneq \text{PSPACE}$
- b) $\text{PSPACE} \subsetneq \text{EXPSPACE}$
- c) Für jedes $k \in \mathbb{N}$: $\text{DSPACE}(O(n^k)) \subsetneq \text{DSPACE}(O(n^{k+1}))$

Mit dem Satz von Savitch und dem det. Hierarchiesatz ergibt sich:

Korollar 13.7

Es sei $f: \mathbb{N} \rightarrow \mathbb{N}$ platzkonstruierbar. Dann gilt

$$\text{NSPACE}(o(f)) \subsetneq \text{NSPACE}(O(f \cdot f))$$

Korollar 13.8

Es gilt $\text{NL} \subsetneq \text{NPSPACE} = \text{PSPACE}$

Zeithierarchiesatz

Satz 13.9

Es sei $f: \mathbb{N} \rightarrow \mathbb{N}$ zeitkonstruierbar und g eine Funktion mit $g(n) \log g(n) \in o(f)$. Dann gilt

$$DTIME(g) \subsetneq DTIME(f)$$

Beweis ähnlich wie für den Platzhierarchiesatz.

Korollar 13.10

Es gilt

- a) $P \subsetneq EXP$
- b) $NP \subsetneq NEXP$
- c) Für jedes $k \in \mathbb{N}$: $DTIME(O(n^k)) \subsetneq DTIME(O(n^{k+1}))$

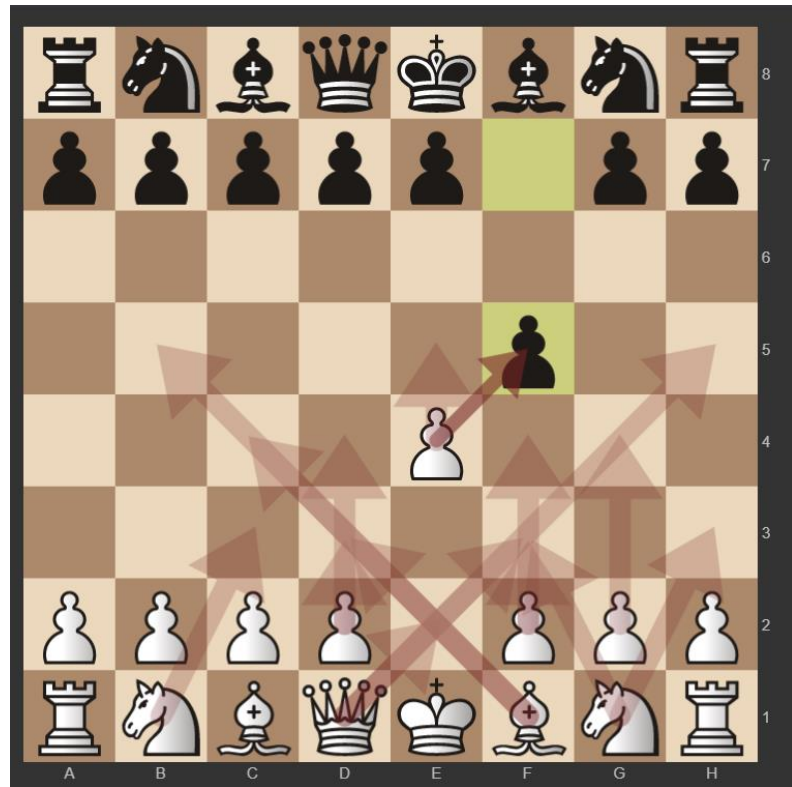
Damit gibt es bspw. keine Konstante $k \in \mathbb{N}$, sodass $P = DTIME(O(n^k))$

Probleme in EXP



Welche Probleme lassen sich nicht in Polyzeit lösen?

Jedes EXP-vollständige Problem!
Darunter:
Schach auf einem $n \times n$ Brett.



Das Ende!

Vorlesungszeit
endet



Klausurenphase
beginnt



Morgen



Zusammenfassung

Evaluation

Klausur