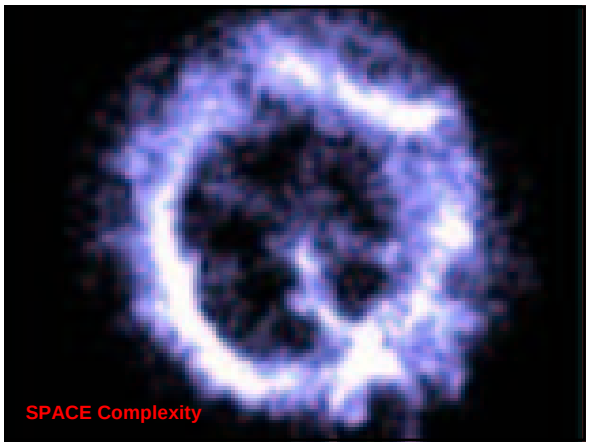


Platzklassen - Erinnerung

DSPACE(f(n))
NSPACE(f(n))

L = LOG = DSPACE(logn)
NL = NLOG = NSPACE(logn)
PSPACE = ∪ DSPACE(n^k), k > 1
NPSPACE = ∪ NSPACE(n^k), k > 1

3-Band Maschinen

Für kleine Platzklassen:

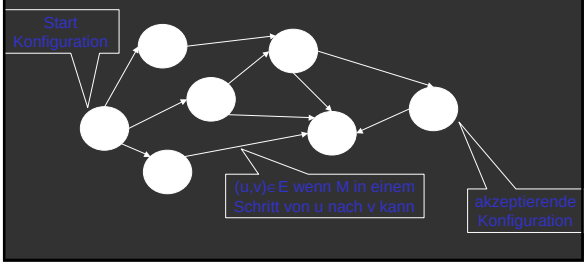
- Arbeitsband: Platzbeschränkung
- Eingabeband
- Ausgabeband

Konfigurationen

Eine Maschine mit logarithmischem (polynomiellem) Arbeitsband

hat eine polynomielle (exponentielle) Anzahl von moeglichen Konfigurationen.

Konfigurationsgraph



Savitch' Theorem

$NSPACE(S(n)) \subseteq SPACE(S(n)^2)$

Korollar:
PSPACE = NPSPACE

Idee: Platz kann wiederverwendet werden !

REACH

Problem REACH

- **Instance**: Gerichteter Graph G=(V,E) mit zwei Knoten s,t∈V
- **Query**: Gibt es einen Pfad von s nach t ?

REACH ist in NL

REACH(s,t)
for i = 1, .., |V|
  If s = t then accept
  Guess s'
  if E(s,s') then s:=s'
Reject

Logspace Reduktionen

Restriktiver als polynomielle Reduktion

A ist logspace reduzierbar auf B ($A \leq_L B$)
wenn es eine Reduktion f gibt sodass
1) $w \in A$ iff $f(w) \in B$
2) f in LOG berechenbar ist.

Logspace Reduktionen

Transitiv ? $A \leq_L B$ und $B \leq_L C$ impliziert $A \leq_L C$?

Alle Reduktionen in dieser Vorlesung sind auch logspace-Reduktionen.

Vollstaendigkeit

B ist NL-vollstaendig wenn
1. B∈ NL
2. f.a. A∈ NL, $A \leq_L B$.

B ist PTIME-vollstaendig wenn
1. B∈ PTIME
2. f.a. A∈ PTIME, $A \leq_L B$.

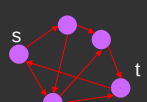
REACH ist NL-vollstaendig

Generische Reduktion
Beweisidee:

Turing Maschine M

"Akzeptiert M Eingabe x?"

$\leq_L$



"Fuehrt ein Weg von s nach t?"

UN-REACH

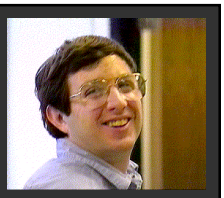
Problem UN-REACH

- Instance: Gerichteter Graph G=(V,E) mit zwei Knoten s,t∈V
- Query: Gibt es keinen Pfad von s nach t ?

coNL-vollstaendig

NL = coNL

Immerman/Szelepcsenyi 1987



(1) UN-REACH ist coNL-vollstaendig
(2) UN-REACH ∈ NL
Daher folgt NL=coNL. ■

Immerman Algorithmus

"Induktives Zaehlen"

```
s(0) := 1;
for k := 1 to n-1 do
"compute s(k) from s(k-1)"
     l := 0;
     for u:= 1 to n do if "u ∈ S(k)" then l:=l+1;
s(k) := l;
Accept and output s(n-1)

"u ∈ S(k)"
m:=0; reply := false;
for v:= 1 to n do if "v ∈ S(k-1)" then
                          m:= m+1;
                          if E(u,v) then reply:= true;
if m < s(k-1) then REJECT else return reply;

"v ∈ S(k-1)"
o := x;
for p := 1 to k-1 do
   guess v'; if not E(o,v') then REJECT
   o:= v'
   if v' = v then return true
return false;
```

Korollar

∀s(n)≥log(n),
NSPACE(s(n))=coNSPACE(s(n))

Wegen CSL = NSPACE(n) sind die kontextsensitiven Sprachen unter Komplement abgeschlossen.

Gödel Preis

Gemeinsam an
Neil Immerman und Robert Szelepcsenyi

Erstes nichttriviales Resultat zu den grossen klassischen Fragen der Komplexitaetstheorie.
Leider keine Auswirkungen auf andere Fragen.

PTIME-Vollstaendigkeit

CIRCUIT Evaluation
Instance: Schaltkreis mit fixierten Eingaben.
Query: Ist die Ausgabe 1 ?

Monotone CIRCUIT Evaluation
Instance: Schaltkreis mit fixierten Eingebaben aus UND, ODER Gattern
Query: Ist die Ausgabe 1 ?

Beide Probleme sind PTIME-vollstaendig.
Warum ?

Circuit Evaluation (Circuit Value)

Uniforme Schaltkreise

Theorem

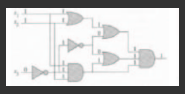
Zu jedem polynomialen Algorithmus (PTIME Algorithmus) A mit binären Eingaben und Ausgabe 0 oder 1 gibt es eine Familie von Schaltkreisen $C_1, C_2, \ldots$, sodass

(1) C_i in **Platz log i** konstruiert werden kann.

(2) Für alle Eingaben der Länge n, der Schaltkreis C_n dasselbe Ergebnis wie A liefert.

PTIME
Turing Maschine

$\leq_L$



"Akzeptiert M Eingabe x?"

"Evaluiert C mit Eingabe x zu 1 ? "

LIN = DTIME(k.n) ?

CIRCUIT Evaluation ist in LIN.

Vollstaendig ? Ja, aber

- LIN ist nicht abgeschlossen unter polynomiellen / logarithmischen Reduktionen:

  d.h. A $\leq_L$ B, B in LIN **impliziert NICHT** A in LIN.

- LIN ist nicht stabil auf verschiedenen zwischen Maschinenmodellen

LOGSPACE-Vollstaendigkeit ?

Deterministic Reachability

Variante von REACH:
Graph ohne Verzweigungen

Deterministic REACH ist LOGSPACE-vollstaendig

Warum ? Bzgl. welcher Reduktionen ?

Klassen unter LOGSPACE ?

Alle in FO (Praedikatenlogik 1. Stufe) definierbaren Grapheigenschaften koennen in LOGSPACE berechnet werden – aber nicht umgekehrt.

- ➲ FO ist echt in LOGSPACE enthalten.
- ➲ FO Reduktionen → spaeter.

Reduktionen

- $\leq_T$ Turing / Cook Reduktionen
- $\leq_{tt}$ Truth Table Reduktionen
- $\leq_P$ Polynomielle many-one Reduktionen
- $\leq_L$ Logspace many-one Reduktionen
- $\leq_{FO}$ First order many-one Reduktionen
- $\leq_{qf}$ quantorenfreie first order many-one Reduktionen

Reduktionen

- $\leq_T$ Turing / Cook Reduktionen
- $\leq_{tt}$ Truth Table Reduktionen
- $\leq_P$ Polynomielle many-one Reduktionen
- $\leq_L$ Logspace many-one Reduktionen
- $\leq_{FO}$ First order many-one Reduktionen
- $\leq_{qf}$ quantorenfreie first order many-one Reduktionen

- $\leq_T$ Turing / Cook Reduktionen

$A \leq_T B$

Ein polynomieller Algorithmus mit Orakel B kann A loesen.

Reduktionen

- $\leq_T$ Turing / Cook Reduktionen
- $\leq_{tt}$ Truth Table Reduktionen
- $\leq_P$ Polynomielle many-one Reduktionen
- $\leq_L$ Logspace many-one Reduktionen
- $\leq_{FO}$ First order many-one Reduktionen
- $\leq_{qf}$ quantorenfreie first order many-one Reduktionen

- $\leq_{tt}$ Truth Table Reduktionen

$A \leq_{tt} B$

Pol. Algorithmus bereitet zuerst mehrere Queries an B vor, stellt danach die Queries und verwendet die Resultate, um A zu loesen.

Reduktionen

- $\leq_T$ Turing / Cook Reduktionen
- $\leq_{tt}$ Truth Table Reduktionen
- $\leq_P$ Polynomielle many-one Reduktionen
- $\leq_L$ Logspace many-one Reduktionen
- $\leq_{FO}$ First order many-one Reduktionen
- $\leq_{qf}$ quantorenfreie first order many-one Reduktionen

- $\leq_P$ Polynomielle many-one Reduktionen

$A \leq_P B$

Spezialfall von truth table Reduktion, wo eine Query auf B das Endergebnis liefert.

Reduktionen

- $\leq_T$ Turing / Cook Reduktionen
- $\leq_{tt}$ Truth Table Reduktionen
- $\leq_P$ Polynomielle many-one Reduktionen
- $\leq_L$ Logspace many-one Reduktionen
- $\leq_{FO}$ First order many-one Reduktionen
- $\leq_{qf}$ quantorenfreie first order many-one Reduktionen

- $\leq_L$ logspace many-one Reduktionen

$A \leq_L B$

Spezialfall von polynomieller Reduktion, in logspace berechenbar.

GAMES

Alternating Reachability

Spieler O will von s nach t gelangen.
Spieler A sabotiert.

Jeder Spieler hat eine Teilmenge der Knoten, auf denen er ziehen darf.

Frage: *hat O eine Gewinnstrategie ?*

PTIME-vollstaendig.

Alternating Turing Machines

Verallgemeinerte Sicht des Non-Determinismus:

Alternating TMs haben existentielle und universelle Zustaende.

Akzeptanzbedingung

Alternating Reachability am Konfigurationsgraphen

O ←→ existentielle Konfigurationen.

A ←→ universelle Konfigurationen.

Alternating Turing Machines

ATIME(f(n))
ASPACE(f(n))

AL = alternating logspace = ASPACE(log n)

Alternating Reachability ist AL-vollstaendig.

Korollar: AL = PTIME.

Korollar: AL = PTIME

X ist in AL
↔
X $\leq_L$ Alt. REACH
↔
X ist in PTIME

→ Vollstaendigkeit
→ AL und PTIME sind unter polynomiellen / logarithmischen Reduktionen abgeschlossen.

Alternating Turing Machines

- Polynomielle Zeit mit nur existentiellen Zustaenden: NP
- Polynomielle Zeit mit nur universellen Zustaenden: coNP
- Logarithmischer Platz mit nur existentiellen oder universellen Zustaenden: NL
- Logarithmischer Platz mit beliebigen Zustaenden: AL = PTIME

- Polynomielle Zeit mit beliebigen Zustaenden ?

QBF

QBF als Spiel ?
APTIME vollstaendig.

Korollar:
APTIME = PSPACE

QBF

- Instance: quantifizierte Boolesche Formel ϕ
- Query: Ist ϕ wahr ?

QBF ist PSPACE-vollstaendig

Beweisidee:

"Akzeptiert PSPACE TM M Eingabe x?" $\leq_P$ $\forall x_1 \exists x_2 \forall x_3 \ldots [\ldots]$

"Ist die QBF Formel wahr ?"

QBF ist PSPACE-hart

TM M
Sprache L$\in$ PSPACE
Input x
Konfigurationen u,v:

Konstruiere Boolesche Funktion
$f_{M,x}(u, v) = 1$ gdw
M auf Eingabe x einen Schritt
von u nach v geht (vgl. Cook/Levin)

Konfigurationsgraph ist exponentiell

Binaere Suche ?

Gibt es Pfad mit Laenge $\leq d$ von u nach v ?

$\phi(u,v,1) \equiv f_{M,x}(u, v) \vee u=v$

$\phi(u,v,d) \equiv \exists w \forall x \forall y[((x=u \wedge y=w) \vee (x=w \wedge y=v)) \rightarrow \phi(x,y,d/2)]$

Quantifizierung erlaubt
Wiederverwendung von Variablen !

QBF Varianten

1 Auf jeden Allquantor folgt ein Existenzquantor, und umgekehrt.
2 Matrix ist in CNF, DNF

... alle PSPACE-vollstaendig.
Namensvariante: QSAT

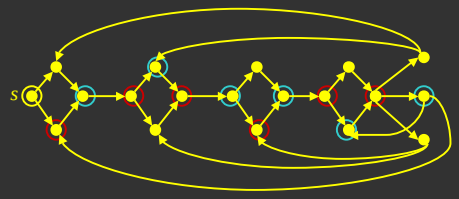
Geography

Muenchen
Novosibirsk
Kapstadt
Togliatti
Ingolstadt
Tiflis
Salzburg etc.

Geography

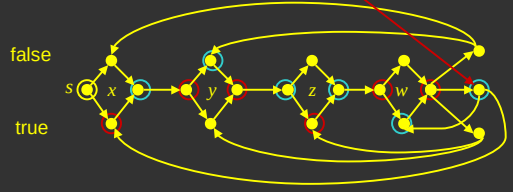
Spieler ziehen alternierend auf einem Graphen. Ein Spieler verliert, wenn alle Optionen zu einem bereits besuchten Knoten fuehren.

Geography



QBF $\leq_L$ Geography

$\exists x \forall y \exists z \forall w[(x \vee y) \wedge (\neg x \vee \neg w) \wedge (\neg y \vee \neg z)]$



Klassen & Probleme

PSPACE	QBF, Geography
NP	SAT, 3COL, CLIQUE, Hamiltonian Path, Rucksack, …
PTIME	CIRCUIT EVAL, ALT. REACH
NL	REACH
L	DET. REACH