

Complexity Theory

Georg Moser

Institute of Computer Science @ UIBK

Summer 2008

- Summary of Last Lecture: Logspace Computability
- Exercises
- Circuit Value Problem
- The Cook-Levin Theorem



Completeness

hardness

Definition

a set $A \subseteq \Sigma^*$ is \leq_m^{\log} -hard for a complexity class \mathcal{C} if

- $\forall B \in \mathcal{C}$ we have $B \leq_m^{\log} A$

Definition

a set $A \subseteq \Sigma^*$ is complete for \mathcal{C} with respect to \leq_m^{\log} if

completeness

- 1 A is \leq_m^{\log} -hard for \mathcal{C}
- 2 $A \in \mathcal{C}$

Definition

MAZE

- given a directed graph $G = (V, E)$ and nodes $s, t \in V$
- determine whether there is a directed path from s to t in G
- this problem is called MAZE or directed graph reachability

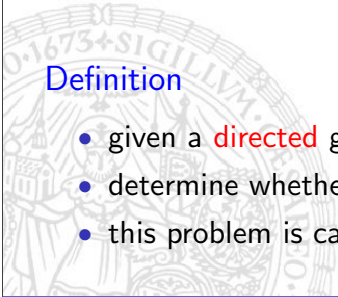
Theorem

Jones, Lien, and Lasser (1976)

MAZE is \leq_m^{\log} -complete for NLOGSPACE

Corollary

MAZE \in LOGSPACE if and only if LOGSPACE = NLOGSPACE



Homework

- 1 Homework 3.3
- 2 Miscellaneous Exercise 4
- 3 Miscellaneous Exercise 6
- 4 Miscellaneous Exercise 8
- 5 Miscellaneous Exercise 10



Proof

the proof has two parts

- ① show $CVP \in P$
- ② show CVP is \leq_m^{\log} -hard for P

Proof ①

easy ■

Comments

- we already know that the evaluation of a Boolean formula can be performed in LOGSPACE
- while Boolean formulas are labelled trees, Boolean circuits are labelled dags

The Circuit Value Problem

Definition

Boolean circuit

a **Boolean circuit** is a program that consists of finitely many assignments of form:

$$\begin{array}{lll} P_i := 1 & P_i := P_j \wedge P_k & P_i := \neg P_j \\ P_i := 0 & P_i := P_j \vee P_k & \end{array}$$

where $j, k < i$ and every P_i is defined at most once

the **value** of circuit is the value of P_n , where n is maximal

Definition

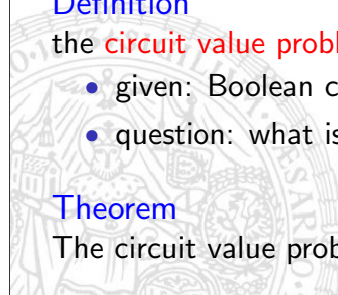
CVP

the **circuit value problem (CVP)** is defined as:

- given: Boolean circuit (with several inputs)
- question: what is the value of the circuit?

Theorem

The circuit value problem is \leq_m^{\log} -complete for P



Proof ②

- let $A \in P$ and $A = L(M)$
- M deterministic, single-tape polynomial time-bounded with time-bound n^c
- Q set of states of M
- $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$
- encode configurations of M in a finite alphabet Δ

Observations

- 1 runtime is bounded by n^c ; memory consumption is bounded by n^c
- 2 represent computation as $(n^c + 1) \times (n^c + 1)$ matrix with entries in Δ
- 3 computation can be encoded as **local consistency conditions**
- 4 encode these as Boolean circuit

Construction

variables:

$$P_{ij}^a \quad (0 \leq i, j \leq n^c, a \in \Gamma) \quad Q_{ij}^q \quad (0 \leq i, j \leq n^c, q \in Q)$$

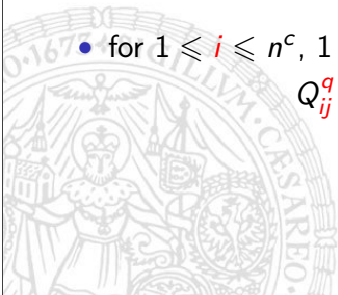
Local Consistency Conditions

- for $1 \leq i \leq n^c, 0 \leq j \leq n^c, b \in \Gamma$:

$$P_{ij}^b := \bigvee_{\delta(p,a)=(q,b,d)} (Q_{i-1,j}^p \wedge P_{i-1,j}^b) \vee \bigvee_{p \in Q} (P_{i-1,j}^b \wedge \bigwedge_{p \in Q} \neg Q_{i-1,j}^p)$$

- for $1 \leq i \leq n^c, 1 \leq j \leq n^c - 1, q \in Q$:

$$Q_{ij}^q := \bigvee_{\delta(p,a)=(q,b,R)} (Q_{i-1,j-1}^p \wedge P_{i-1,j-1}^a) \vee \bigvee_{\delta(p,a)=(q,b,L)} (Q_{i-1,j+1}^p \wedge P_{i-1,j+1}^a)$$



Local Consistency Conditions (cont'd)

- for $j = 0$:

$$Q_{i0}^q := \bigvee_{\delta(p,a)=(q,b,L)} (Q_{i-1,1}^p \wedge P_{i-1,1}^a)$$

- for $j = n^c$:

$$Q_{in^c}^q := \bigvee_{\delta(p,a)=(q,b,R)} (Q_{i-1,n^c-1}^p \wedge P_{i-1,n^c-1}^a)$$

- encoding of start configuration on $x = a_1 \dots a_n$:

$$P_{0,0}^+ := 1 \quad P_{0,j}^{a_j} := 1 \quad (1 \leq j \leq n)$$

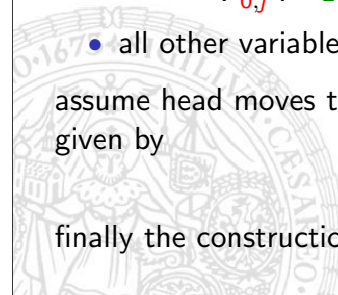
$$P_{0,j}^{\perp} := 1 \quad (n+1 \leq j \leq n^c) \quad Q_{0,0}^s := 1$$

- all other variables are set to false

assume head moves to the left before accepting, acceptance of M on x is given by

$$Q_{n^c,0}^t \vee Q_{n^c,1}^t$$

finally the construction is **logspace computable** ■



The Cook-Levin Theorem

Theorem

Boolean satisfiability is \leq_m^{\log} -complete for NP

Observations

- with SAT the input values are not given, but have to be found
- CVP is defined in terms of circuits
- SAT is defined in terms of formulas

Proof

additional assignments: $P_i := ?$

rest on blackboard ■

