Computational Logic

# Complexity Theory

Georg Moser

Institute of Computer Science @ UIBK

Summer 2008

## Outline

- Summary of Last Lecture: Parallel Complexity
- Exercises
- Relation to Time-Space Classes (Proof)

## Definition

a family of Boolean circuits $C_0, C_1, C_2, \ldots$ is a logspace-uniform family of Boolean circuits of polylog depth and polynomial size if

1. $C_n$ has $n$ inputs and is composed of $\wedge$, $\vee$ and $\neg$-gates
2. $C_n$ is of depth at most $(\log n)^{O(1)}$

   depth is the length of the longest path from input to output
3. $C_n$ has no more than $n^{O(1)}$ gates
4. the $(C_i)_{i \in \mathbb{N}}$ is logspace-uniform:
   $\exists$ a logspace transducer that produces the circuits $C_n$ on input $0^n$

## Definition NC

a set $A \subseteq \{0,1\}^*$ is in NC if $\exists$ a logspace-uniform family of Boolean circuits of polylog depth and polynomial size, where each $C_n$ has one output and $\forall\, x \in \{0,1\}^*$

$$x \in A \Leftrightarrow C_{|x|}(x) = 1$$





## Definition STA

the class STA$(S(n), T(n), A(n))$ is the class of sets accepted by ATMs that

1. are $S(n)$-space bounded,
2. $T(n)$-time bounded, and
3. consist of at most $A(n)$ alternating sections

## Theorem

$\mathrm{NC} = \mathrm{STA}(\log n, *, (\log n)^{O(1)})$

# Homework

1. Homework 4.1.
2. Homework 4.2.
3. Homework 5.3.
4. Miscellaneous Exercises 29.
5. Miscellaneous Exercises 31.





## Theorem

$\mathrm{NC} = \mathrm{STA}(\log n, *, (\log n)^{\mathrm{O}(1)})$

## Definition

logspace-uniform means $\exists$ a logspace transducer $M$ that produces the circuits $C_n$ on input $0^n$

more precisely

1. $M$ enumerates all names of the gates in $C_n$
2. $\forall$ gates $c$ in $C_n$ indicates type of $c$
3. defines connections between gates
4. indicates for one gate that it is an output gate

## $NC \subseteq STA(\log n, *, (\log n)^{O(1)})$

- $\exists$ a logspace-uniform family of Boolean circuits $C_n$ of polylog depth and polynomial size
- $\exists$ logspace-uniform transducer $M$
- construct ATM $N$ that simulates the family $C_n$:
  on input $x$ ($|x| = n$), $N$ runs $M$ to produce $C_n$ and evaluate $C_n(x)$:
  1. first $N$ finds the output gate and writes in on its tape
  2. in the following assume $d$ and type of $d$ are written on the tape
  3. if $d$ is $\wedge$ or $\vee$ find two input to these gates then branch accordingly
  4. if $d$ is input, accept if $d = 1$ and reject if $d = 0$
  5. if $d$ is $\neg$-gate, find the unique input gate that inputs to $d$, accept/reject accordingly





## $STA(\log n, *, (\log n)^{O(1)}) \subseteq NC$

- $\exists$ alternating logspace machine $N$, making at most $(\log n)^c$ alternations on input $x$ ($|x| = n$)
- represent the next-configuration relation as $n^c \times n^c$ Boolean matrix $R_x$
- construction of the circuit $C_n$
  1. compute entries of $R_x$, depth of $R_x$ is 1
  2. construct matrices:

$$S_x = \{(\alpha, \beta) \mid R_x(\alpha, \beta) = 1 \text{ and } \text{type}(\alpha) = \text{type}(\beta)\}$$
$$T_x = \{(\alpha, \beta) \mid R_x(\alpha, \beta) = 1 \text{ and } \text{type}(\alpha) \neq \text{type}(\beta)\}$$
$$S_x^* = \text{reflexive, transitive closure of } S_x$$

  3. an existential configuration $\alpha$ @ $i + 1$ is accepting iff $\exists$ (universal) $\beta$ @ $i$ with $S_x^* T_x(\alpha, \beta) = 1$, $\beta$ is accepting
  4. a universal configuration $\alpha$ @ $i + 1$ is accepting iff $\forall$ (existential) $\beta$ @ $i$ with $S_x^* T_x(\alpha, \beta) = 1$, $\beta$ is accepting

## $\mathrm{STA}(\log n, *, (\log n)^{O(1)}) \subseteq \mathrm{NC}$ (cont'd)

- represent as circuit calculations via Boolean vectors $b_i$ of length $n^c$ such that $b_i(\alpha) = 1$, if $\alpha$ @ $i$ accepts , and:

$$\vee \qquad b_{i+1} := S_x^* T_x b_i \qquad \wedge \qquad b_{i+1} := \neg(S_x^* T_x(\neg b_i))$$

- initially $b_0$ is zero
- output $b_{(\log n)^c}(\texttt{start})$