Time Space Tradeoffs for SAT (Part I): SAT $\notin$ TISP($n^{1.1}$,$n^{0.1}$)

Complexity Seminar

Presented by Peter Fontana

Wednesday, September 30, 2009

"In a world that values simplicity, we study Complexity."

Introduction

Background/ Definitions

Proving the Result

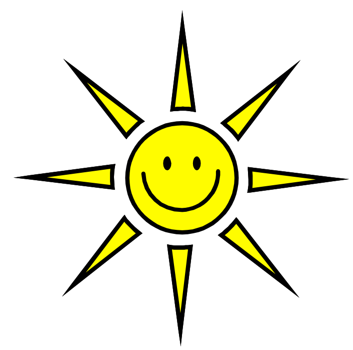
Proving Claim 1

Proving Claim 2

Conclusions

Question of the Day: Decision vs. Search

- Most Complexity Class use Decision Problems
  - Can we reduce the search problem where we only solve the decision problem *once*?
- TALK-SAT: Does there exist a talk I can give that will make all of you *"happy"* ?
  - What about actually giving that talk?
- So, are you **satisfiable**?



"In a world that values simplicity, we study Complexity."

Introduction

Background/ Definitions

Proving the Result

Proving Claim 1

Proving Claim 2

Conclusions

Main Result

- **SAT $\notin$ TISP($n^{1.1}$, $n^{0.1}$)**
  - Presented in Computational Complexity: A Modern Approach by Sanjeev Arora and Boaz Barak (2009).
- Thus, for ***any* problem in NP** we cannot solve it in **both** linear time and logarithmic space.
- Stronger results are known.

Introduction

Background/ Definitions

Proving the Result

Proving Claim 1

Proving Claim 2

Conclusions

SAT

- **Definition (SAT):** *SAT* = { ϕ | ϕ is a propositional formula (in CNF) that has some truth assignment that makes it true. }

- Here SAT means CNF-SAT,
  - Formula is in *Conjunctive Normal Form (CNF)*
  - CNF: an AND of OR clauses

Introduction

Background/ Definitions

Proving the Result

Proving Claim 1

Proving Claim 2

Conclusions

SAT - An Example

- Consider the Formula:

$$\phi = (x_1 \lor \neg x_2) \land (\neg x_1 \lor x_2 \lor x_3)$$

$$x_1 = \mathrm{T}, x_2 = \mathrm{T}, x_3 = \mathrm{T} \text{ satisfies } \phi$$

Introduction

Background/ Definitions

Proving the Result

Proving Claim 1

Proving Claim 2

Conclusions

TISP(T(n), S(n))

- **Definition (TISP(T(n),S(n))):** A Language $L \in$ *TISP(T(n), S(n))* iff it can be solved by a TM M in at most O(T(n)) time and at most O(S(n)) space.

- **Alternate Definition (TISP(T(n), S(n))):** *TISP(T(n), S(n))* = DTIME(T(n)) $\cap$ DSPACE(S(n))

- So TISP($n^{1.1}$, $n^{0.1}$) means that a TM can solve this with O($n^{1.1}$) time and O($n^{0.1}$) space

Turing Machine (TM)

State Machine
(Deterministic or Nondeterministic)

Input Tape (Read Only)

Work Tape (Read-Write)

Turning Machine TM (NDTM) M

- Takes T(n) Steps total (execution depth)
- Work Tape uses S(n) Space
- Non-deterministic TM abbreviated NDTM

Introduction

Background/
Definitions

Proving the Result

Proving Claim 1

Proving Claim 2

Conclusions

Turing Machine Properties

- Alphabet assumed to be {0,1}
- Tapes are semi-infinite with sequential access.
- Non-Determinism - Equivalent Views:
  - Can view as a giving a non-deterministic state machine
  - Can view as a deterministic state machine with a certificate (What we will do)

Introduction

Background/ Definitions

Proving the Result

Proving Claim 1

Proving Claim 2

Conclusions

Turing Machine

- **Theorem:** *There exists a Universal Turing Machine U that if M takes T(n) steps, U takes CT(n)(log (T(n)) steps.*

- This Includes:
  - A Non-binary Alphabet
  - Multiple Work Tapes
  - Two-Way Infinite Tapes

Introduction

Background/ Definitions

Proving the Result

Proving Claim 1

Proving Claim 2

Conclusions

More Time Notations

- **Definition (DTIME(T(n))):** $L \in DTIME(T(n))$ iff there exists a TM M that takes O(T(n)) time to decide L (with input length n).
- **Definition (NTIME(T(n))):** $L \in NTIME(T(n))$ iff there exists a Non-Deterministic TM (NDTM) M that takes O(T(n)) time to decide L (with input length n).
- **Example:** DTIME(n) - can solve in linear time.

Introduction

Background/ Definitions

Proving the Result

Proving Claim 1

Proving Claim 2

Conclusions

Space Complexity Notation

- **Definition (DSPACE(S(n))):** $L \in DSPACE(S(n))$ iff there exists a TM M that takes O(S(n)) space to decide L (with input length n).

- **Definition (NSPACE(S(n))):** $L \in NSPACE(S(n))$ iff there exists a NDTM M that takes O(S(n)) time to decide L (with input length n).

- **Example:** DSPACE(log(n)) - can solve in logarithmic space.

Introduction

Background/ Definitions

Proving the Result

Proving Claim 1

Proving Claim 2

Conclusions

Key Claims to Prove

- **Claim 1:** $TISP(n^{12}, n^2) \subseteq \Sigma_2 TIME(n^8)$
- **Claim 2:** *If* $NTIME(n) \subseteq DTIME(n^{1.2})$, *Then* $\Sigma_2 TIME(n^8) \subseteq NTIME(n^{9.6})$

Introduction

Background/ Definitions

Proving the Result

Proving Claim 1

Proving Claim 2

Conclusions

Claims we Accept Without Proof

- **NP Reduction Claim:** *The Cook-Levin Reduction adds a factor of polylog(n) time and polylog(n) space.*
  - Thus, If SAT $\in$ TISP($n^{1.1}$, $n^{0.1}$), then NTIME(n) $\in$ TISP($n^{1.2}$, $n^{0.2}$)

- **Theorem (Non-Deterministic Time Hierarchy):** *For any functions f, g s.t $f(n+1) = o(g(n))$, then NTIME(f(n)) $\subset$ NTIME(g(n))*
  - $\subset$ = Proper Subset

Introduction

Background/ Definitions

Proving the Result

Proving Claim 1

Proving Claim 2

Conclusions

Defining Σ_2TIME(n^8)

- **Definition:** $L \in \Sigma_2 TIME(n^8)$ iff there is a TM M such that:

  $$x \in L \Leftrightarrow \exists u \in \{0,1\}^{c|x|^8} : \forall v \in \{0,1\}^{d|x|^8} : M(x,u,v) = 1$$

  for constants c, d, where M runs in O(n^8) time. (Recall |x| = n).

- This is equivalent to a definition that uses *Alternating Turing Machines* (they will not be discussed here)

Introduction

Background/ Definitions

Proving the Result

Proving Claim 1

Proving Claim 2

Conclusions

Proving Our Result [1/5]:

- **Theorem (Main Result):**
  $NTIME(n) \not\subseteq TISP(n^{1.2}, n^{0.2})$

- *Proof of Theorem:* (Proof by Contradiction)
  - Suppose that $NTIME(n) \subseteq TISP(n^{1.2}, n^{0.2})$
    - Hence, $NTIME(n) \subseteq DTIME(n^{1.2})$ **(by Def of TISP)**
  - Then **By Lemma** (*via Padding,* in a future slide), $NTIME(n^{10}) \subseteq TISP(n^{12}, n^{2})$

Proving Our Result [2/5]

- From **Claim 1** and **Claim 2**,
  - $TISP(n^{12}, n^2) \subseteq \Sigma_2 TIME(n^8) \subseteq NTIME(n^{9.6})$
- Putting This together, we have:
  - $NTIME(n^{10}) \subseteq NTIME(n^{9.6})$
- But this contradicts the **Non-Deterministic Time Hierarchy Theorem**
  - Thus, $NTIME(n) \not\subseteq TISP(n^{1.2}, n^{0.2})$

Introduction

Background/ Definitions

Proving the Result

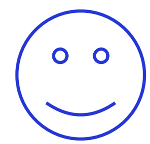
Proving Claim 1

Proving Claim 2

Conclusions

Proving Our Result [3/5]

- And from our **NP Reduction Claim**,
  - $\mathrm{NTIME}(n) \subseteq \mathrm{SAT}(n(\mathrm{polylog}(n))) \not\subseteq \mathrm{TISP}(n^{1.2}, n^{0.2})$
  - Hence $\mathbf{SAT(n) \not\subseteq TISP(n^{1.1}, n^{0.1})}$



Proving our Result [4/5] (Padding Time)

- **Lemma:** *$NTIME(n) \subseteq TISP(n^{1.2}, n^{0.2}) \Rightarrow NTIME(n^{10}) \subseteq TISP(n^{12}, n^{2})$*

- *Proof of Lemma:* Let $L \in NTIME(n^{10})$

  Consider $L' = \{x \circ 1^{|x|^{10}} \mid x \in L\}$

- The string is x concatenated with $|x|^{10}$ 1's
  - Thus, **$L' \in$ NTIME(n)** (The $NTIME(n^{10})$ algorithm only works on a fraction of the input)

Introduction

Background/ Definitions

Proving the Result

Proving Claim 1

Proving Claim 2

Conclusions

Proving our Result [5/5] (Padding Time)

- Using our Premise, L’ $\in$ TISP($n^{1.2}$, $n^{0.2}$)
- We then can use this algorithm of padding to get an TISP(n^{12}, n^{2}) algorithm for L (Since the padded input is longer)
- Thus, **L $\in$ TISP(n^{12}, n^{2})**

Proving Key Claim 1 [1/4]

- **Claim 1:** *$TISP(n^{12}, n^2) \subseteq \Sigma_2 TIME(n^8)$*
- *Proof of Claim 1:* Let $L \in TISP(n^{12}, n^2)$
- Look at the configuration graph of L
  - Each configuration takes $O(n^2)$ space
  - There is a path from C_{start} to C_{accept} in $O(n^{12})$ time
- Guessing the *one* midpoint configuration yields too many alternations (alternating quantifiers).

Configuration Graphs (An Aside) [1/4]

- **Definition (Configuration):** A *configuration* of a TM M for input x is a state (or status) that the TM can be in (during its execution on x). This consists of:
  - The current state the TM M is in
  - The position of the Input Tape Head
  - The position of the Work Tape Head
  - The contents of the Work Tape

Introduction

Background/ Definitions

Proving the Result

Proving Claim 1

Proving Claim 2

Conclusions

Configuration Graphs (An Aside) [2/4]

- **Definition (Configuration Graph CG = (V,E)):** The configurations of TM M for input x form the *Configuration Graph CG = (V,E)* [Directed Graph]
  - Each Configuration C_v is a vertex
  - We draw a directed edge from C_u to C_v iff the TM M can go from C_u to C_v in **one step**.
  - C_{start} = An initial configuration of M
  - C_{accept} = any configuration where M outputs 1

Introduction

Background/ Definitions

Proving the Result

Proving Claim 1

Proving Claim 2

Conclusions

Configuration Graphs (An Aside) [3/4]

- **Example Configuration:** (TM heads at character surrounded by *)
  - At state q_1,
  - Input Tape = 100***1***0□. (□ = blank),
  - Work Tape = □1*□*00□.

Introduction

Background/ Definitions

Proving the Result

Proving Claim 1

Proving Claim 2

Conclusions

Configuration Graphs (An Aside) [4/4]

- **Example Configuration Step:**
  - Write 1 at current position of work tape,
  - Move input head 1 position to the left,
  - Keep the state the same,
  - Keep the work tape head at the same position.

Proving Key Claim 1 [1/4] (Returning to the Proof)

- **Claim 1:** *$TISP(n^{12}, n^2) \subseteq \Sigma_2 TIME(n^8)$*
- *Proof of Claim 1:* Let $L \in TISP(n^{12}, n^2)$
- Look at the configuration graph of L
  - Each configuration takes $O(n^2)$ space
  - There is a path from C_{start} to C_{accept} in $O(n^{12})$ time
- Guessing the *one* midpoint configuration yields too many alternations (alternating quantifiers).

Proving Key Claim 1 [2/4]

- So we Guess n^6 "midpoint" configurations.
  - Divide the path into n^6 equal length segments
  - *Guess all n^6* of those nodes C_i
- For all i, we require that there is at most a path of length n^6 between C_i and C_{i+1}
- This only has one alternation (two quantifiers) in it.

Introduction

Background/ Definitions

Proving the Result

Proving Claim 1

Proving Claim 2

Conclusions

Proving Key Claim 1 [3/4]

- Formally, L can be expressed as:

$$\exists C_1, \ldots C_{n^6} : \forall 0 \leq i \leq n^6 : \text{ we can reach } C_{i+1} \text{from } C_i$$
$$\text{in at most } n^6 \text{ steps } \wedge$$
$$C_0 = C_{start} \wedge C_{n^6+1} = C_{accept}$$

- Since only n^6 C_i and one i to quantify, L $\in \Sigma_2$TIME(--)

- We *want* L $\in \Sigma_2$TIME(**n^8**) (So we analyze the time taken)

Introduction
Background/ Definitions
Proving the Result
Proving Claim 1
Proving Claim 2
Conclusions

Proving Key Claim 1 [4/4]

- We *non-deterministically* pick each C_i
- For each pair (C_i, C_{i+1}), we we can in $O(n^7)$ verify that there is a path of at most n^6 steps between them.
- Specifying the n^6 Configurations C_i can take $O(n^8)$, bits, and thus $O(n^8)$ time to read.
- Therefore, **$L \in \Sigma_2 TIME(n^8)$**

Mid-Way Recap

- From **Claim 1** and **Claim 2**,
  - $\text{TISP}(n^{12}, n^2) \underbrace{\subseteq \Sigma_2\text{TIME}(n^8)}_{\textbf{Claim 1}} \underbrace{\subseteq \text{NTIME}(n^{9.6})}_{\textbf{Claim 2}}$

- Proved **Claim 1**, now prove **Claim 2**

- From this, we concluded $\text{NTIME}(n) \not\subseteq \text{TISP}(n^{1.2}, n^{0.2})$ and thus $\text{SAT}(n) \not\subseteq \text{TISP}(n^{1.1}, n^{0.1})$

Proving Key Claim 2 [1/3]

- **Claim 2:** *If NTIME(n) $\subseteq$ DTIME($n^{1.2}$), Then Σ_2TIME(n^8) $\subseteq$ NTIME($n^{9.6}$)*

- *Proof of Claim 2:* Let L $\in \Sigma_2$TIME(n^8)

- Recall, L $\in \Sigma_2$TIME(n^8) iff there is a TM M where

$$x \in L \Leftrightarrow$$
$$\exists u \in \{0,1\}^{c|x|^8} : \forall v \in \{0,1\}^{d|x|^8} : M(x,u,v) = 1$$

for constants c, d, where M runs in O(n^8) time.

Introduction

Background/ Definitions

Proving the Result

Proving Claim 1

Proving Claim 2

Conclusions

Proving Key Claim 2 [2/3]

- Let L' be defined as such (it takes in 2 inputs) with constant d:

$$L' = \{(x, u) \mid \forall v \in \{0,1\}^{d|x|^8} : M(x, u, v) = 1\}$$

- Now Consider L^p to be (this is L' rephrased, where M^c is the complement of TM M):

$$L^p = \{(x, u) \mid \exists v \in \{0,1\}^{d|x|^8} : M^c(x, u, v) = 1\}$$

- If we let n' = (d + 1)$|x|^8$, $L^p \in$ NTIME(n')

Introduction

Background/ Definitions

Proving the Result

Proving Claim 1

Proving Claim 2

Conclusions

Proving Key Claim 2 [3/3]

- Thus $L^p \in DTIME((n')^{1.2}) = DTIME(n^{8*1.2}) = DTIME(n^{9.6})$ (**By Premise**)
  - Therefore, $L' \in DTIME(n^{9.6})$

- Recasting the definition of L in terms of L': (c is a constant)

$$L = \{x \mid \exists u \in \{0,1\}^{c|x|^8} : (x,u) \in L'\}$$

- Hence, $\mathbf{L \in NTIME(n^{9.6})}$ (**Definition of NTIME**) ☺

Introduction

Background/ Definitions

Proving the Result

Proving Claim 1

Proving Claim 2

Conclusions

Conclusions [1/2]

- **SAT $\notin$ TISP($n^{1.1}$, $n^{0.1}$)**
  - Thus, to solve any problem in NP, we know we need:
    - **either** more than linear time
    - **or** more than logarithmic space
- Stronger Results are known
  - … Some will be presented in future talks

Conclusions [2/2]

- Complexity Techniques Used:
  - Padding the Input (Padding Time)
  - Language Reduction (and Complementation)
  - Using the Configuration Graph and non-deterministically selecting “midpoints”
- This result also holds for Random Access TMs (Can access memory like a Computer accesses RAM).