Lecture 7 - Algebraic computation, Uniform and Non-uniform

Rafael Oliveira
rafael.oliveira.teaching@gmail.com
University of Waterloo

CS 860 - Graduate Complexity Theory
Fall 2022

Overview

- Uniform Computation over a Ring (Field)
- Algebraic Circuits

Problems of interest

Let R be a ring (commutative with unit)

- System of polynomial equations — decision
- Semi-algebraic systems of equations — decision
- Root finding — search
-

Finite BSS model

- Let R be a ring
  Apart from ring operations, have (in unit cost):
  - if R is ordered (like $\mathbb{R}$), then we have access to ≥ 0
  - else, only able to test $= 0$
  - if R field, then can divide

Finite BSS model

- Let R be a ring
  Apart from ring operations, have (in unit cost):
  - if R is ordered (like $\mathbb{R}$), then we have access to ≥ 0
  - else, only able to test $= 0$
  - if R field, then can divide
- Finite machine M over R:
  1. Three spaces:
     - Input space: $\mathcal{I}_M = R^n$
     - State space: $\mathcal{S}_M = R^m$
     - Output space: $\mathcal{O}_M = R^\ell$

Finite BSS model

- Let R be a ring
  Apart from ring operations, have (in unit cost):
  - if R is ordered (like $\mathbb{R}$), then we have access to ≥ 0
  - else, only able to test $= 0$
  - if R field, then can divide
- Finite machine M over R:
  1. Three spaces:
     - Input space: $\mathcal{I}_M = R^n$
     - State space: $\mathcal{S}_M = R^m$
     - Output space: $\mathcal{O}_M = R^\ell$
  2. directed graph G with 4 types of nodes:
     - input node — in-degree 0, outdegree 1
     - output nodes — out-degree 0
     - computation nodes — outdegree 1
     - branch nodes — outdegree 2

Finite BSS model

- Let R be a ring
  Apart from ring operations, have (in unit cost):
  - if R is ordered (like $\mathbb{R}$), then we have access to ≥ 0
  - else, only able to test $= 0$
  - if R field, then can divide
- Finite machine M over R:
  1. Three spaces:
     - Input space: $\mathcal{I}_M = R^n$
     - State space: $\mathcal{S}_M = R^m$
     - Output space: $\mathcal{O}_M = R^\ell$
  2. directed graph G with 4 types of nodes:
     - input node — in-degree 0, outdegree 1
     - output nodes — out-degree 0
     - computation nodes — outdegree 1
     - branch nodes — outdegree 2
  3. Each node performs a computation over R
     - Input node: $I : \mathcal{I}_M \to \mathcal{S}_M$ — linear map
     - Computation: $g : \mathcal{S}_M \to \mathcal{S}_M$ — polynomial (rational) map
     - Branch: $h : \mathcal{S}_M \to R$ — testing $= 0$ or ≥ 0
     - Output: $O_M : \mathcal{S}_M \to \mathcal{O}_M$ — linear map

Computation over Finite Machines

- Suppose we have the following problem (with $R = \mathbb{C}$):
  - Input: $f \in \mathbb{C}[x]$
  - Output: find approximation z to a root of f

Computation over Finite Machines

- Suppose we have the following problem (with $R = \mathbb{C}$):
  - Input: $f \in \mathbb{C}[x]$
  - Output: find approximation z to a root of f
- Algorithm: Newton's method

Infinite Tape BSS Model

- $R^\infty := \bigsqcup_{n\geq 0} R^n$
- $R_\infty :=$ bi-infinite direct sum space

$$(\ldots, x_{-2}, x_{-1}, x_0.x_1, x_2, \ldots)$$

  with $x_k = 0$ for $|k|$ sufficiently large

Infinite Tape BSS Model

- R_∞ := bi-infinite direct sum space

$$(\dots, x_{-2}, x_{-1}, x_0.x_1, x_2, \dots)$$

  with $x_k = 0$ for $|k|$ sufficiently large
- Polynomials and rational functions $h : R^m \to R$ on R_∞ defined by evaluation at coordinates $[m] := \{1, 2, \dots, m\}$

Infinite Tape BSS Model

- $R_\infty :=$ bi-infinite direct sum space
$$(\dots, x_{-2}, x_{-1}, x_0.x_1, x_2, \dots)$$
with $x_k = 0$ for $|k|$ sufficiently large
- Polynomials and rational functions $h : R^m \to R$ on R_∞ defined by evaluation at coordinates $[m] := \{1, 2, \dots, m\}$
- Infinite tape model has in addition to the finite model an extra node called shift nodes σ, where $\sigma_l(x)_i = x_{i+1}$ and $\sigma_r(x)_i = x_{i-1}$

Shifts the distinguished marker

Infinite Tape BSS Model

- $R_\infty :=$ bi-infinite direct sum space

$$(\ldots, x_{-2}, x_{-1}, x_0.x_1, x_2, \ldots)$$

  with $x_k = 0$ for $|k|$ sufficiently large
- Polynomials and rational functions $h : R^m \to R$ on R_∞ defined by evaluation at coordinates $[m] := \{1, 2, \ldots, m\}$
- Infinite tape model has in addition to the finite model an extra node called shift nodes σ, where $\sigma_l(x)_i = x_{i+1}$ and $\sigma_r(x)_i = x_{i-1}$
- Input/output maps $I_\infty : R^\infty \to R_\infty$ and $O_\infty : R_\infty \to R^\infty$:
  - $I_\infty(x) = (\ldots, 0, \hat{n}.x_1, \ldots, x_n, 0, 0, \ldots) \qquad x \in R^n$
  - $O_\infty(\ldots, x_0.x_1, \ldots, x_\ell, \ldots) = \begin{cases} 0 \in R^0, \text{ if } \ell = 0 \\ (x_1, \ldots, x_\ell) \in R^\ell \text{ otherwise} \end{cases}$

    where $\ell = \min_{i \geq 0}\{x_{-i} = 0\}$

Languages and decision problems

- A language $L \subseteq \mathbb{R}^\infty$
- L is computable/decidable if its characteristic function χ_L is decidable by R-machine

Languages and decision problems

- A language $L \subseteq \mathbb{R}^\infty$
- L is computable/decidable if its characteristic function χ_L is decidable by R-machine
- Complexity of computing a problem: number of nodes traversed in computation[1]

[1] Can define different costs for handling different elements of R, which yield different complexity measures. See **[BCSS]**, Chapter 4.

Languages and decision problems

- A language $L \subseteq \mathbb{R}^\infty$
- L is computable/decidable if its characteristic function χ_L is decidable by R-machine
- Complexity of computing a problem: number of nodes traversed in computation
- $\mathsf{P}_R :=$ class of languages decided by poly-time R-machines

Languages and decision problems

- A language $L \subseteq \mathbb{R}^\infty$
- L is computable/decidable if its characteristic function χ_L is decidable by R-machine
- Complexity of computing a problem: number of nodes traversed in computation
- $\mathsf{P}_R :=$ class of languages decided by poly-time R-machines
- Can now define reductions in similar way to boolean setting.

Languages and decision problems

- A language $L \subseteq \mathbb{R}^\infty$
- L is computable/decidable if its characteristic function χ_L is decidable by R-machine
- Complexity of computing a problem: number of nodes traversed in computation
- $\mathsf{P}_R :=$ class of languages decided by poly-time R-machines
- Can now define reductions in similar way to boolean setting.
- Projections: given language L, let

$$\Pi(L) := \{x \in R^\infty \mid \exists y \in R^\infty \text{ s.t. } (x,y) \in L\}$$

Languages and decision problems

- A language $L \subseteq \mathbb{R}^\infty$
- L is computable/decidable if its characteristic function χ_L is decidable by R-machine
- Complexity of computing a problem: number of nodes traversed in computation
- $\mathsf{P}_R :=$ class of languages decided by poly-time R-machines
- Can now define reductions in similar way to boolean setting.
- Projections: given language L, let

$$\Pi(L) := \{x \in R^\infty \mid \exists y \in R^\infty \text{ s.t. } (x, y) \in L\}$$

-
$$\mathsf{NP}_R := \{\Pi(L) \mid L \in \mathsf{P}_R\}$$

Languages and decision problems

- A language $L \subseteq \mathbb{R}^\infty$
- L is computable/decidable if its characteristic function χ_L is decidable by R-machine
- Complexity of computing a problem: number of nodes traversed in computation
- $\mathsf{P}_R :=$ class of languages decided by poly-time R-machines
- Can now define reductions in similar way to boolean setting.
- Projections: given language L, let

$$\Pi(L) := \{x \in R^\infty \ \mid \ \exists y \in R^\infty \text{ s.t. } (x,y) \in L\}$$

-

$$\mathsf{NP}_R := \{\Pi(L) \ \mid \ L \in \mathsf{P}_R\}$$

- Boolean parts: given complexity class $\mathcal{C}$

$$0/1 - \mathcal{C} := \{L \cap \{0,1\}^* \ \mid \ L \in \mathcal{C}\}$$

Complete Problems

- Hilbert Nullstellensatz (HN):
  - Input: polynomials $p_1, \ldots, p_r \in R[x_1, \ldots, x_n]$
  - Output: YES, if there is $\alpha \in R^n$ such that $p_i(\alpha) = 0$ for all $i \in [r]$

Complete Problems

- Hilbert Nullstellensatz (HN):
  - Input: polynomials $p_1, \ldots, p_r \in R[x_1, \ldots, x_n]$
  - Output: YES, if there is $\alpha \in R^n$ such that $p_i(\alpha) = 0$ for all $i \in [r]$
- $HN_{\mathbb{C}}$ is $\mathsf{NP}_{\mathbb{C}}$-complete.
- $0/1 - HN_{\mathbb{C}}$ is $0/1 - \mathsf{NP}_{\mathbb{C}}$-complete.

Complete Problems

- Hilbert Nullstellensatz (HN):
  - Input: polynomials $p_1, \ldots, p_r \in R[x_1, \ldots, x_n]$
  - Output: YES, if there is $\alpha \in R^n$ such that $p_i(\alpha) = 0$ for all $i \in [r]$
- $HN_{\mathbb{C}}$ is $\mathrm{NP}_{\mathbb{C}}$-complete.
- $0/1 - HN_{\mathbb{C}}$ is $0/1 - \mathrm{NP}_{\mathbb{C}}$-complete.
- Semi-algebraic fesibility (SA-FEAS):
  - Input: polynomials $p_1, \ldots, p_r, q_1, \ldots, q_s \in R[x_1, \ldots, x_n]$ where R is ordered ring
  - Output: YES, if there is $\alpha \in R^n$ such that $p_i(\alpha) \geq 0$ and $q_i > 0$ for all $i \in [r]$

Complete Problems

- Hilbert Nullstellensatz (HN):
  - Input: polynomials $p_1, \ldots, p_r \in R[x_1, \ldots, x_n]$
  - Output: YES, if there is $\alpha \in R^n$ such that $p_i(\alpha) = 0$ for all $i \in [r]$
- $HN_{\mathbb{C}}$ is $\mathrm{NP}_{\mathbb{C}}$-complete.
- $0/1 - HN_{\mathbb{C}}$ is $0/1 - \mathrm{NP}_{\mathbb{C}}$-complete.
- Semi-algebraic fesibility (SA-FEAS):
  - Input: polynomials $p_1, \ldots, p_r, q_1, \ldots, q_s \in R[x_1, \ldots, x_n]$ where R is ordered ring
  - Output: YES, if there is $\alpha \in R^n$ such that $p_i(\alpha) \geq 0$ and $q_i > 0$ for all $i \in [r]$
- $SA - FEAS$ is $\mathrm{NP}_{\mathbb{R}}$-hard

Relation to other models

- If $R = \mathbb{Z}/2\mathbb{Z}$ then infinite tape BSS is a classical Turing Machine

Relation to other models

- If $R = \mathbb{Z}/2\mathbb{Z}$ then infinite tape BSS is a classical Turing Machine
- If $R = \mathbb{R}$ then $\mathsf{P}_{/poly} \subset P_{\mathbb{R}}$

  Can encode the advice in one entry of the tape - access its bits using ≥ 0 branch nodes

Relation to other models

- If $R = \mathbb{Z}/2\mathbb{Z}$ then infinite tape BSS is a classical Turing Machine
- If $R = \mathbb{R}$ then $\mathsf{P}_{/poly} \subset P_{\mathbb{R}}$

  Can encode the advice in one entry of the tape - access its bits using ≥ 0 branch nodes
- $0/1 - \mathsf{P}_{\mathbb{C}} \subseteq \mathsf{BPP}$

Relation to other models

- If $R = \mathbb{Z}/2\mathbb{Z}$ then infinite tape BSS is a classical Turing Machine
- If $R = \mathbb{R}$ then $\mathsf{P}_{/poly} \subset P_{\mathbb{R}}$

  Can encode the advice in one entry of the tape - access its bits using ≥ 0 branch nodes
- $0/1 - \mathsf{P}_{\mathbb{C}} \subseteq \mathsf{BPP}$
- $0/1 - \mathsf{NP}_{\mathbb{C}} \subseteq \mathsf{PSPACE}$

Relation to other models

- If $R = \mathbb{Z}/2\mathbb{Z}$ then infinite tape BSS is a classical Turing Machine
- If $R = \mathbb{R}$ then $\mathsf{P}_{/poly} \subset P_{\mathbb{R}}$

  Can encode the advice in one entry of the tape - access its bits using ≥ 0 branch nodes
- $0/1 - \mathsf{P}_{\mathbb{C}} \subseteq \mathsf{BPP}$
- $0/1 - \mathsf{NP}_{\mathbb{C}} \subseteq \mathsf{PSPACE}$
- Under GRH

$$0/1 - \mathsf{NP}_{\mathbb{C}} \subseteq \mathsf{PH}$$

Relation to other models

- If $R = \mathbb{Z}/2\mathbb{Z}$ then infinite tape BSS is a classical Turing Machine
- If $R = \mathbb{R}$ then $\mathsf{P}_{/poly} \subset P_{\mathbb{R}}$

  Can encode the advice in one entry of the tape - access its bits using ≥ 0 branch nodes
- $0/1 - \mathsf{P}_{\mathbb{C}} \subseteq \mathsf{BPP}$
- $0/1 - \mathsf{NP}_{\mathbb{C}} \subseteq \mathsf{PSPACE}$
- Under GRH

  $$0/1 - \mathsf{NP}_{\mathbb{C}} \subseteq \mathsf{PH}$$
- If K, L are algebraically closed fields of characteristic zero, then

  $$\mathsf{NP}_K = \mathsf{P}_k \Leftrightarrow \mathsf{NP}_L = \mathsf{P}_L$$

- Uniform Computation over a Ring (Field)
- Algebraic Circuits

Definition & Reductions

Complexity Classes

- VP

$$\{F_n\}_n \in \mathsf{VP} \Leftrightarrow \exists c \in \mathbb{N} \text{ and } \{C_n\}_n \text{ circuit s.t.}$$

$$S(C_n) \leq n^c,\ \deg(C_n) \leq n^c,\ \text{and } C_n(x) = F_n(x)$$

Complexity Classes

- VP
- VNP

$$\{F_n\}_n \in \mathsf{VNP} \Leftrightarrow \exists c \in \mathbb{N} \text{ and } \{C_n\}_n \in \mathsf{VP}, t(n) \leq n^c \text{ s.t.}$$

$$F_n(x) = \sum_{b \in \{0,1\}^m} C_{t(n)}(x, b)$$

Complete polynomial: $\mathsf{Per}_n(X) = \sum_{\sigma \in S_n} \prod_{i=1}^n X_{i\sigma(i)}$

Complexity Classes

- VP
- VNP

  Complete polynomial: $\mathrm{Per}_n(X) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} X_{i\sigma(i)}$
- VBP

  Complete polynomial: $\mathrm{Det}_n(X) = \sum_{\sigma \in S_n} (-1)^{\sigma} \prod_{i=1}^{n} X_{i\sigma(i)}$

Complexity Classes

- VP
- VNP

  Complete polynomial: $\mathrm{Per}_n(X) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} X_{i\sigma(i)}$
- VBP

  Complete polynomial: $\mathrm{Det}_n(X) = \sum_{\sigma \in S_n} (-1)^{\sigma} \prod_{i=1}^{n} X_{i\sigma(i)}$
- VNC

Theorem 1

$$VP = VNC = VNC^2$$

Polynomial Identity Testing

References I

Arora, Sanjeev and Barak, Boaz (2009)
Computational Complexity, A Modern Approach — Chapter 16
Cambridge University Press

Blum, L. and Cucker, F and Shub, M. and Smale, S. (1998)
Complexity and real computation — Chapters 1-5
Springer Science & Business Media