Solvers Principles and Architecture (SPA)

General Introduction

Master Sciences Informatique (Sif)
September 18th, 2017
Rennes

Khalil Ghorbal
khalil.ghorbal@inria.fr

Before understanding Solvers

Before understanding Solvers

We need to talk about Problems

What makes a problem important ?

Abstraction of problems arising from important applications.

Abstraction of problems arising from important applications.

When asked how old she was, Suzie replied, "In 2 years I will be twice as old as I was 5 years ago." How old is she?

Abstraction of problems arising from important applications.

When asked how old she was, Suzie replied, "In 2 years I will be twice as old as I was 5 years ago." How old is she?

$$x + 2 = 2(x - 5)$$
$$\rightarrow x = 12$$

Abstraction of problems arising from important applications.

When asked how old she was, Suzie replied, "In 2 years I will be twice as old as I was 5 years ago." How old is she?

$$x + 2 = 2(x - 5)$$
$$\to x = 12$$

Travelling Salesman Problem

Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?

Reduction of other problems.

$$\begin{pmatrix} Problem1 \\ \vdots \\ ProblemN \end{pmatrix} \rightsquigarrow ProblemA$$

- The transformation (reduction) may be non-trivial to find
- Needs to be **"simpler"** than solving the new problem

Reduction of other problems.

$$\begin{pmatrix} Problem1 \\ \vdots \\ ProblemN \end{pmatrix} \leadsto ProblemA$$

- The transformation (reduction) may be non-trivial to find
- Needs to be **"simpler"** than solving the new problem

Reduction of other problems.

$$\begin{pmatrix} Problem1 \\ \vdots \\ ProblemN \end{pmatrix} \rightsquigarrow ProblemA$$

- The transformation (reduction) may be non-trivial to find
- Needs to be **"simpler"** than solving the new problem

Satisfiability

Given a set V of Boolean variables and a collection C of clauses over V, is there a satisfying truth assignment for C?

Quadratic Diophantine Equations

Given positive integers a, b, and c, are there positive integers x and y such that $ax^2 + by^2 = c$? (Transformation from 3SAT [Manders and Adleman 1978].)

Satisfiability

Given a set V of Boolean variables and a collection C of clauses over V, is there a satisfying truth assignment for C?

Quadratic Diophantine Equations

Given positive integers a, b, and c, are there positive integers x and y such that $ax^2 + by^2 = c$? (Transformation from 3SAT [Manders and Adleman 1978].)

Poincaré said so!

- Its **long resistance** (beyond the current state-of-the art methods)
- Requires **new insights** (connections, perspectives) to get solved
- Example: Hilbert's famous list pf problems (1900)
- Example: Millennium Prize Problems

Riemann Hypothesis

All the non-trivial zeros of the Riemann zeta function have their real part equal to $\frac{1}{2}$.

Poincaré said so!

- Its **long resistance** (beyond the current state-of-the art methods)
- Requires **new insights** (connections, perspectives) to get solved
- Example: Hilbert's famous list pf problems (1900)
- Example: Millennium Prize Problems

Riemann Hypothesis

All the non-trivial zeros of the Riemann zeta function have their real part equal to $\frac{1}{2}$.

Poincaré said so!

- Its **long resistance** (beyond the current state-of-the art methods)
- Requires **new insights** (connections, perspectives) to get solved
- Example: Hilbert's famous list pf problems (1900)
- Example: Millennium Prize Problems

Riemann Hypothesis

All the non-trivial zeros of the Riemann zeta function have their real part equal to $\frac{1}{2}$.

Poincaré said so!

- Its **long resistance** (beyond the current state-of-the art methods)
- Requires **new insights** (connections, perspectives) to get solved
- Example: Hilbert's famous list pf problems (1900)
- Example: Millennium Prize Problems

Riemann Hypothesis

All the non-trivial zeros of the Riemann zeta function have their real part equal to $\frac{1}{2}$.

Poincaré said so!

- Its **long resistance** (beyond the current state-of-the art methods)
- Requires **new insights** (connections, perspectives) to get solved
- Example: Hilbert's famous list pf problems (1900)
- Example: Millennium Prize Problems

Riemann Hypothesis

All the non-trivial zeros of the Riemann zeta function have their real part equal to $\frac{1}{2}$.

Outline

Satisfiability (DPLL algorithm)

Is there a Boolean assignment that satisfies

$$(v_1 \vee \bar{v}_2) \wedge (\bar{v}_1 \vee v_2)$$

Quantifier Elimination (Cylindrical Algebraic Decomposition)

Is the following sentence true over the reals

$$\forall a, b. \exists x. \quad x^2 + ax + b = 0$$

Important Problems You Must Be Aware Of

Satisfiability (DPLL algorithm)

Is there a Boolean assignment that satisfies

$$(v_1 \vee \bar{v}_2) \wedge (\bar{v}_1 \vee v_2)$$

Quantifier Elimination (Cylindrical Algebraic Decomposition)

Is the following sentence true over the reals

$$\forall a, b. \exists x. \quad x^2 + ax + b = 0$$

Convex Optimization (SemiDefinite Programming)

$$\begin{aligned} \text{Min/Max} \quad & C \bullet X \\ \text{Subject to} \quad & A_i \bullet X = b_i, \quad i = 1, \ldots, m \\ & X \succeq 0 \end{aligned}$$

Differential Equations (Numerical Algorithms)

$$i\hbar \frac{\partial}{\partial t} \Psi(\mathbf{r}, t) = H \, \Psi(\mathbf{r}, t)$$

Convex Optimization (SemiDefinite Programming)

$$\begin{array}{rl} \text{Min/Max} & C \bullet X \\ \text{Subject to} & A_i \bullet X = b_i, \quad i = 1, \ldots, m \\ & X \succeq 0 \end{array}$$

Differential Equations (Numerical Algorithms)

$$i\hbar \frac{\partial}{\partial t} \Psi(\mathbf{r}, t) = H\, \Psi(\mathbf{r}, t)$$

Object of study in mathematics is the set of solutions of equations

$$f(x) = 0$$

Nature and operators in f

- Linear (vector of)
- Polynomial (vector of)
- With special operators: derivations
- ...

Solution Space

- Finite fields ($\mathbb{Z}/p\mathbb{Z}$)
- Reals
- Differential functions
- Probability densities
- ...

Object of study in mathematics is the set of solutions of equations

$$f(x) = 0$$

Nature and operators in f

- Linear (vector of)
- Polynomial (vector of)
- With special operators: derivations
- ...

Solution Space

- Finite fields ($\mathbb{Z}/p\mathbb{Z}$)
- Reals
- Differential functions
- Probability densities
- ...

Object of study in mathematics is the set of solutions of equations

$$f(x) = 0$$

Nature and operators in f

- Linear (vector of)
- Polynomial (vector of)
- With special operators: derivations
- ...

Solution Space

- Finite fields ($\mathbb{Z}/p\mathbb{Z}$)
- Reals
- Differential functions
- Probability densities
- ...

- Existence?
- Unicity?
- Exact Solving (exact general form of the solution)
- Properties of the set of solutions (finiteness, bounded, structure, symmetries etc.)
- Generalizations
- Approximations (numerical methods, relaxation)

- Existence?
- Unicity?
- Exact Solving (exact general form of the solution)
- Properties of the set of solutions (finiteness, bounded, structure, symmetries etc.)
- Generalizations
- Approximations (numerical methods, relaxation)

- Existence?
- Unicity?
- Exact Solving (exact general form of the solution)
- Properties of the set of solutions (finiteness, bounded, structure, symmetries etc.)
- Generalizations
- Approximations (numerical methods, relaxation)

- Existence?
- Unicity?
- Exact Solving (exact general form of the solution)
- Properties of the set of solutions (finiteness, bounded, structure, symmetries etc.)
- Generalizations
- Approximations (numerical methods, relaxation)

- Existence?
- Unicity?
- Exact Solving (exact general form of the solution)
- Properties of the set of solutions (finiteness, bounded, structure, symmetries etc.)
- Generalizations
- Approximations (numerical methods, relaxation)

- Existence?
- Unicity?
- Exact Solving (exact general form of the solution)
- Properties of the set of solutions (finiteness, bounded, structure, symmetries etc.)
- Generalizations
- Approximations (numerical methods, relaxation)

- Existence?
- Unicity?
- Exact Solving (exact general form of the solution)
- Properties of the set of solutions (finiteness, bounded, structure, symmetries etc.)
- Generalizations
- Approximations (numerical methods, relaxation)

Purpose: **Classification**

Outline

Definition

- Description of the parameters
- Statement of what an answer, or solution, is required to satisfy

Example: Traveling Salesman Problem

The problem consists of a finite set of locations/cities $C = \{c_1, \ldots, c_m\}$ and for each pair of cities c_i, c_i in C, the distance $d(c_i, c_j)$ between them. A solution is an ordering $c_{\pi(1)}, c_{\pi(2)}, \ldots, c_{\pi(m)}$ such that minimizes

$$\left(\sum_{i=1}^{m-1} d\left(c_{\pi(i)}, c_{\pi(i+1)}\right)\right) + d\left(c_{\pi(m)}, c_{\pi(1)}\right)$$

Definition

- Description of the parameters
- Statement of what an answer, or solution, is required to satisfy

Example: Traveling Salesman Problem

The problem consists of a finite set of locations/cities $C = \{c_1, \ldots, c_m\}$ and for each pair of cities c_i, c_i in C, the distance $d(c_i, c_j)$ between them. A solution is an ordering $c_{\pi(1)}, c_{\pi(2)}, \ldots, c_{\pi(m)}$ such that minimizes

$$\left(\sum_{i=1}^{m-1} d\left(c_{\pi(i)}, c_{\pi(i+1)}\right)\right) + d\left(c_{\pi(m)}, c_{\pi(1)}\right)$$

Step-by-step procedure to solve **any** instance of a given problem.

Efficiency

- Time complexity (is not the only important parameter)
- How does the time needed to solve the problem evolves when the input length increases?
- Number of symbols in the description of the instance with respect to the encoding scheme for the problem.

Example of input length

- Alphabet $\{c, [,], /, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- "c[1]c[2]c[3]c[4]//10/5/9//6/9//3" (32 symbols)

Step-by-step procedure to solve **any** instance of a given problem.

Efficiency

- Time complexity (is not the only important parameter)
- How does the time needed to solve the problem evolves when the input length increases?
- Number of symbols in the description of the instance with respect to the encoding scheme for the problem.

Example of input length

- Alphabet $\{c, [,], /, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- "c[1]c[2]c[3]c[4]//10/5/9//6/9//3" (32 symbols)

Polynomial Time Complexity

Time complexity is $O(p(n))$ for some polynomial p with input length n.

Exponential Time Complexity

Time complexity cannot be bounded by a polynomial time complexity (including $n^{\log n}$).

	$n = 10$	$n = 30$	$n = 60$
n^3	0.001 s	0.027 s	0.216 s
3^n	0.059 s	6.5 years	1.3×10^{13} centuries

Polynomial Time Complexity

Time complexity is $O(p(n))$ for some polynomial p with input length n.

Exponential Time Complexity

Time complexity cannot be bounded by a polynomial time complexity (including $n^{\log n}$).

	$n = 10$	$n = 30$	$n = 60$
n^3	0.001 s	0.027 s	0.216 s
3^n	0.059 s	6.5 years	1.3×10^{13} centuries

- Undecidable (not solvable by any algorithm)
- Intractable? (no polynomial time algorithm can possibly solve it)
- Complexity (worst case with respect to the input length)
- Target special cases
- Approximations (relaxations)

- Undecidable (not solvable by any algorithm)
- Intractable? (no polynomial time algorithm can possibly solve it)
- Complexity (worst case with respect to the input length)
- Target special cases
- Approximations (relaxations)

- Undecidable (not solvable by any algorithm)
- Intractable? (no polynomial time algorithm can possibly solve it)
- Complexity (worst case with respect to the input length)
- Target special cases
- Approximations (relaxations)

- Undecidable (not solvable by any algorithm)
- Intractable? (no polynomial time algorithm can possibly solve it)
- Complexity (worst case with respect to the input length)
- Target special cases
- Approximations (relaxations)

- Undecidable (not solvable by any algorithm)
- Intractable? (no polynomial time algorithm can possibly solve it)
- Complexity (worst case with respect to the input length)
- Target special cases
- Approximations (relaxations)

- Undecidable (not solvable by any algorithm)
- Intractable? (no polynomial time algorithm can possibly solve it)
- Complexity (worst case with respect to the input length)
- Target special cases
- Approximations (relaxations)

Purpose: **Computational**

- Dive into three different problems: **SMT**, **CAD**, **SDP**
- Study their related solvers: **Algorithms** and **Data Structure**
- Learn how to **Deconstruct** a problem

- Dive into three different problems: **SMT**, **CAD**, **SDP**
- Study their related solvers: **Algorithms** and **Data Structure**
- Learn how to **Deconstruct** a problem

- Dive into three different problems: **SMT**, **CAD**, **SDP**
- Study their related solvers: **Algorithms** and **Data Structure**
- Learn how to **Deconstruct** a problem