Automated Reasoning: Some Successes and New Challenges

Predrag Janičić
www.matf.bg.ac.rs/~janicic
Automated Reasoning GrOup (ARGO)
Faculty of Mathematics
University of Belgrade, Serbia

Central European Conference on Information and Intelligent Systems
(CECIIS 2011)
Varaždin, Croatia, September 21-23, 2011.

Faculty of Mathematics, University of Belgrade

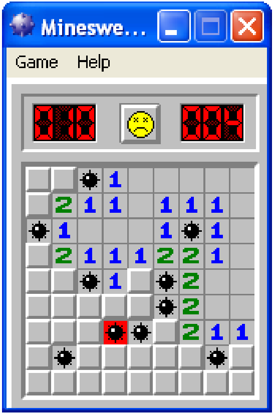
- University of Belgrade
- Faculty of Mathematics
- Automated Reasoning GrOup (ARGO)
    - Area: automated and interactive theorem proving, SAT, SMT, geometry reasoning
    - 10 members
    - More at: http://argo.matf.bg.ac.rs/

What is this talk about?

This talk is about...

This talk is about...

... how to play *minesweeper* ...



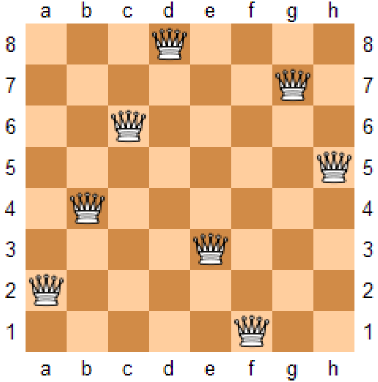
This talk is about...

... how to play *sudoku* ...

		3						
6			5					4
		4	3	7			1	8
1				9	2	6	7	
	2		8					9
4			1				2	
	6			5				1
	4	8			9	2		7
		1		8	4		3	

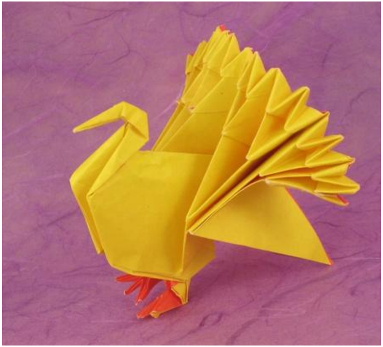
This talk is about...

... how to place 8 queens on a chessboard ...



This talk is about...

... how to explore origami ...

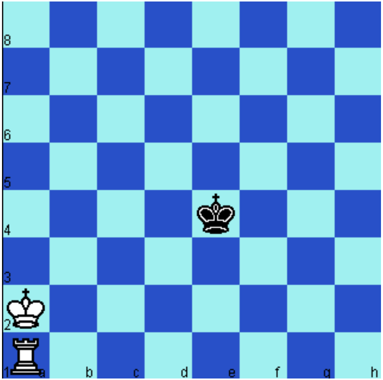


This talk is about...

... how to arrange oranges in a supermarket ...

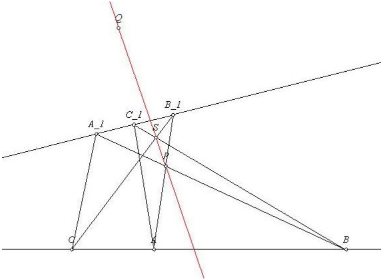
This talk is about...

... how to play chess endgames ...



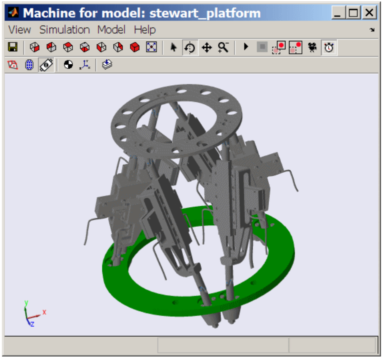
This talk is about...

... how to solve geometry puzzles ...



This talk is about...

... how to make computer-aided design even smarter ...



This talk is about...

... how to make timetables ...

This talk is about...

... how to find a seed if a 100th pseudorandom number is given ...

$$x_{n+1} \equiv 1664525 x_n + 1013904223 \pmod{2^{32}}$$

This talk is about...

... how to solve equations over finite domains ...

$$x^8 + 3x^5 + 4x^3 = 1013904223 \pmod{2^{32}}$$

This talk is about...

... how to prove mathematical conjectures too hard for humans ...
For example:

Every Robbins algebra is Boolean algebra

This talk is about...

... how to verify software...

```
Program Check_Group
  use crystallographic_symmetry, only: Space_Group_Type, set_spacegroup
  use reflections_utilities, only: Hkl_Absent
  use Symmetry_Tables, only: spgr_info, Set_Spgr_Info

  ....... ! Read reflections, apply criterion of "goodness" for checking,
          ! set indices i1,i2 for search in space group tables ...
  ....... ! omitted for simplicity
  call Set_Spgr_Info()
  m=0
  do_group: do i=i1,i2
    hms=adjustl(spgr_info(i)%HM)
    hall=spgr_info(i)%hall
    if( hms(1:1) /= "P" .and. .not. check_cent ) cycle do_group ! Skip centred groups
    call set_spacegroup(hall,Spacegroup,Force_Hall="y")
    do j=1,nhkl
       if(good(j) == 0) cycle  !Skip reflections that are not good (overlap) for checking
       absent=Hkl_Absent(hkl(:,j), Spacegroup)
       if(absent .and. intensity(j) > threshold) cycle do_group  !Group not allowed
    end do
    ! Passing here means that all reflections are allowed in the group -> Possible group!
    m=m+1
    num_group(m)=i
  end do  do_group
  write(unit=*,fmt=*) " => LIST OF POSSIBLE SPACE GROUPS, a total of ",m," groups are possible"
  write(unit=*,fmt=*) "     ------------------------------------------------------"
  write(unit=*,fmt=*) "     Number(IT)      Hermann-Mauguin Symbol     Hall Symbol"
  write(unit=*,fmt=*) "     ------------------------------------------------------"
  do i=1,m
    j=num_group(i)
    hms=adjustl(spgr_info(j)%HM)
    hall=spgr_info(j)%hall
    numg=spgr_info(j)%N
    write(unit=*,fmt="(i10,4a)")  numg,"                   ",hms,"         ",hall
  end do
  ...............
```

This talk is about...

... how to verify hardware...

This talk is about...

... how to verify safety critical systems...

This talk is about...

... Automated Reasoning

Then... what is automated reasoning?

- *...understanding different aspects of reasoning and development of algorithms and computer programs that solve problems requiring reasoning*
- Combines results and techniques of mathematical logic, theoretical computer science, algorithmics and artificial intelligence
- *The beauty of a theorem from mathematics, the preciseness of an inference rule in logic, the intrigue of a puzzle, and the challenge of a game — all are present in the field of automated reasoning.* (Wos)

History of Automated Reasoning

- Roots in ancient Greece
- Leibniz's dreams
- Modern history starts in 1950's

Automated Reasoning Today

- Several conferences and journals
- Several hundreds researchers
- Many applications

Disclaimer

- This is just a very short overview of automated reasoning
- Many subareas, systems, results, applications not covered

SAT Problem (SATisfiability)

- Problem of deciding if a given propositional formula in CNF is satisfiable
- Example: is $(p \vee q \vee \neg r) \wedge (p \vee \neg q \vee r) \wedge (p \vee \neg q \vee \neg r)$ satisfiable?
- Decidable problem
- Canonical NP-complete problem
- Can be reduced to any NP-complete problem and vice versa

Encoding Problems to SAT: Example

- Solve $x + y = 3 \pmod 4$
- Encode x as $[p, q]$
- Encode y as $[r, s]$
- Encode 3 as $[\top, \top]$
- $x + y$ is $[(p \bigoplus r) \bigoplus (q \wedge s), (q \bigoplus s)]$
- Hence, $(p \bigoplus r) \bigoplus (q \wedge s) \equiv \top$ and $(q \bigoplus s) \equiv \top$
- Transform to CNF and find a model

SAT Solvers

- *Logic Theorist* able to prove propositional theorems (Newell, Simon, Shaw, 1956)
- Improved some proofs from *Principia Mathematica*, but the authors failed to publish a paper on the system
- Early solvers DP/DPLL (Davis, Putnam, Longmann, Loveland, 1960, 1962)
- Modern solvers are DPLL-like, but much more advanced
- Can solve instance with millions of clauses

Modern SAT Solvers

- Complex, efficient, well understood, verified...
- BerkMin, grasp, MiniSAT, picoSAT, SATzilla, zChaff
- ArgoSAT, ArgoSmArT developed by the ARGO group
- URSA a system for reducing problems to SAT (ARGO group)

Applications of SAT Solvers

- Applications in many fields: software and hardware verification, timetabling, combinatorial problems, etc.
- "Swiss army knife" for a wide domain of tasks
- ... including most of the given example problems (minesweeper, sudoku, queens, timetabling, verification tasks, problems over finite domains)

Some challenges

- checking unsatisfiability proofs of huge input instances
- development of verified real-world solvers
- development of non-DPLL-based solvers
- development of non-CNF solvers

Validity/Satisfiability in FOL

- Predicates and functions, quantification of variables
- Validity/Satisfiability problem in FOL is undecidable...
- But semidecidable: for each valid formula it can be proved that it is valid
- First such procedures by Skolem and Herbrand (1920s and 1930s)

Resolution Method

- Skolem's and Herbrand's results led to the *resolution method* by Robinson (1965)
- Many variations, many provers, many successes, high expectations
- One of major successes: *all Robbins algebras are Boolean algebras* (open for fifty years, proved in 1997)
- Powerful modern provers based on the resolution method such as E, Otter/Prover9, Spass, Vampire
- Many applications

Provers for Specific FOL Theories

- Uniform proof procedures for pure FOL such as resolution method inefficient for concrete theories
- In addition, many interesting FOL theories are decidable
- First specialized prover for specific FOL theory (linear arithmetic) by Davis (1954), based on Presburger's procedure
- Example of LA formula: $\forall x \forall y.(x > y + 1 \geq x > y)$
- "...its great triumph was to prove that the sum of two even numbers is even"

SMT Solvers

- Satisfiability problem for universal fragment of specific FOL theories: *Satisfiability Modulo Theory* (SMT)
- Modern SMT solvers: Boolector, MathSAT, Yices, Z3,...
- Tremendous advances over the last years, can solve problem instances taking gigabytes of memory
- More expressive, easier problem encoding than with SAT
- Many applications, especially in verification
- URSA Major a system for reducing problems to SMT (ARGO group)

Some challenges

- Dealing with quantification
- Routine verification (*Verification Grand Challenge*)

HOL

- Even more expressive (e.g., quantification over predicate and function symbols)
- Automation of reasoning is very complex
- Used as a setting for interactive theorem proving

Interactive Theorem Proving

- *Proof assistants*) are used to check (and guide) proofs constructed by the user, by verifying each proof step with respect to the given underlying logic
- Formal proofs replace, often flawed, informal proofs
- Formal proof is typically several times longer than a corresponding informal proof
- In some systems, everything checked by extremely small kernel
- Popular proof assistants: Isabelle, Coq, HOL Light, PVS, Mizar, ACL2

Mathematical Revolutions

Wiedijk: "In mathematics there have been three main revolutions:

1. The introduction of proof by the Greeks in the fourth century BC
2. The introduction of rigor in mathematics in the nineteenth century
3. The introduction of [computer supported] formal mathematics in the late twentieth and early twenty-first centuries."

QED ("quod erat demonstrandum")

- A call for a large-scale international effort QED (1993)
- Goal: a computer-based database of all important, established mathematical knowledge, strictly formalized and checked automatically
- In the meanwhile: many QED-style projects, conferences, journals

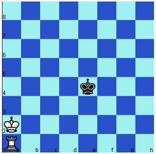
QED-style Successes

- Many of the most significant theorems already proved formally
- "Four color theorem" (Gonthier, 2005)
- The Kepler conjecture (no packing of congruent balls has density greater than that of the face-centered cubic packing)

  Hales and coauthors (from 2003, estimated 66 man-years)
- Verification of Pentium-like AMD5K86 microprocessor
- Verification of SAT solvers (ARGO group)

Other Applications

- Formal reasoning in other domains (not only math and computer science)
- For instance, formal reasoning about origami or formal reasoning in chess:
  - retrograde chess analysis
  - analysis of correctness of endgame strategies



Some challenges

- Theorem provers that are easy to use by mathematicians and more closely resemble traditional mathematics
- Automation of technical parts

Automated Reasoning in Geometry

- Solving problems in geometry: old and very challenging task
- Some geometry theories are decidable (Tarski, 1951)
- Automation (for both decidable and undecidable problems) is additional challenge
- One of the first automated provers aimed at geometry (Gelertner, 1959), able to prove some congruences
- Applications in CAD, robotics, education

Algebraic Theorem Provers — Wu's Method

- Wu's method (1977)
- Can prove hundreds of complex theorems of Euclidean geometry (e.g., those from IMOs)
- Considered by some to be "the most successful" theorem prover overall
- Selected as one of "the four new great Chinese inventions"

Algebraic Theorem Provers — Gröbner Bases method

- Gröbner bases method, one of the major theories in computer algebra
- Invented by Buchberger (1965)
- Applications in coding theory, cryptography, integer programming, ...
- Applicable to geometry theorem proving

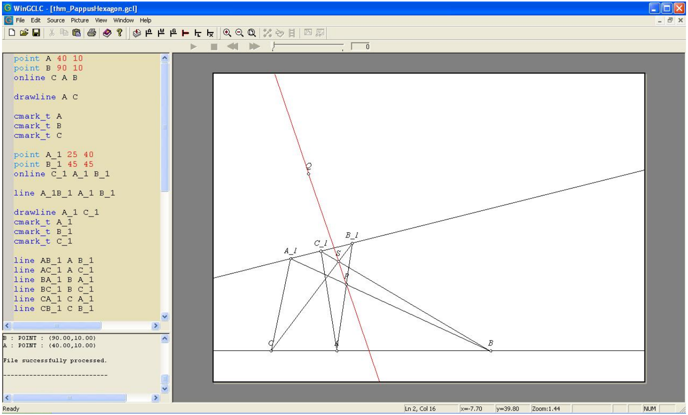
Coordinate-free Methods

- Produce (more or less) traditional, readable proofs
- Several method (by Chou, Gao, Zhang, 1990s):
  - Area method
  - Full angle method
  - Deductive database method

GCLC Tool

- Geometry software (ARGO group)
- Uses a custom "geometry programming" language
- Dynamic geometry features
- Three automated theorem provers built-in: Wu's method, Gröbner bases method, the area method

GCLC Screenshot



GCLC Example Proof Fragment (by the Area Method)

$\frac{\left(S_{APC} \cdot \left(\frac{\overrightarrow{BD}}{\overrightarrow{DC}} \cdot \frac{\overrightarrow{CE}}{\overrightarrow{AE}}\right)\right)}{S_{BPC}} = 1$	by algebraic simplifications	(5)
$\frac{\left(S_{APC} \cdot \left(\frac{\overrightarrow{BD}}{\overrightarrow{DC}} \cdot \frac{S_{CPB}}{S_{APB}}\right)\right)}{S_{BPC}} = 1$	by Lemma 8 (point E eliminated)	(6)
$\frac{\left(S_{APC} \cdot \left(\left(-1 \cdot \frac{\overrightarrow{BD}}{\overrightarrow{CD}}\right) \cdot \frac{S_{CPB}}{S_{APB}}\right)\right)}{(-1 \cdot S_{CPB})} = 1$	by geometric simplifications	(7)
$\frac{\left(S_{APC} \cdot \frac{\overrightarrow{BD}}{\overrightarrow{CD}}\right)}{S_{APB}} = 1$	by algebraic simplifications	(8)
$\frac{\left(S_{APC} \cdot \frac{S_{BPA}}{S_{CPA}}\right)}{S_{APB}} = 1$	by Lemma 8 (point D eliminated)	(9)
$\frac{\left(S_{APC} \cdot \frac{S_{BPA}}{(-1 \cdot S_{APC})}\right)}{(-1 \cdot S_{BPA})} = 1$	by geometric simplifications	(10)
$1 = 1$	by algebraic simplifications	(11)

ArgoCLP prover

- Synthetic geometry theorem prover (ARGO group)
- Based on coherent logic
- Produces both formal and readable proofs

ArgoCLP Example Proof Fragment

4. From the facts that $p \neq q$, and the point A is incident to the line p, and the point A is incident to the line q, it holds that the lines p and q intersect (by axiom ax_D5).

5. From the facts that the lines p and q intersect, and the lines p and q do not intersect we get a contradiction.

Contradiction.

6. Assume that the point A is not incident to the line q.

7. From the facts that the lines p and q do not intersect, it holds that the lines q and p do not intersect (by axiom ax_nint_l_l_21).

8. From the facts that the point A is not incident to the line q, and the point A is incident to the plane α, and the line q is incident to the plane α, and the point A is incident to the line p, and the line p is incident to the plane α, and the lines q and p do not intersect, and the point A is incident to the line r, and the line r is incident to the plane α, and the lines q and r do not intersect, it holds that $p = r$ (by axiom ax_E2).

9. From the facts that $p = r$, and $p \neq r$ we get a contradiction.

Contradiction.

Therefore, it holds that $p = r$.

This proves the conjecture.

Theorem proved in 9 steps and in 0.02 s.

Some challenges

- Development of provers that produce readable proofs efficiently
- Use in mathematical education
- More industrial applications

Conclusions

- AR has made a lot of striking successes over the last decades
- A rich scientific discipline, with strong theoretical grounds and with many applications
- A new driving force for mathematical logic
- AR tools used in everyday practice in mathematics, computer science, engineering, and education
- Many new challenges are set, more successes to come