Puzzle: switch positions of dark and white knights



Commentary: Puzzle borrowed from twitter.



CS 433 Automated Reasoning 2025

Instructor: Ashutosh Gupta

A useful example: verification of AI



Slightly altered images give wrong output in Google cloud vision

Image source: https://arxiv.org/pdf/1905.03333.pdf



CS 433 Automated Reasoning 2025

Lecture 1: Introduction and background

Instructor: Ashutosh Gupta

IITB India

Compile date: 2025-01-10



Topic 1.1

What is automated reasoning?



Automated reasoning (logic)

We will use reasoning and logic synonymously.

Have you ever said to someone, "be reasonable"?

whatever your intuition was that is reasoning

Why we care? Logic is the calculus of computer science

Commentary: 'Automated reasoning' consists of two words. Let us start with an explanation of the word 'reasoning'. The question is like asking what is time? We can not explain time. We can only experience time. Similarly, if someone says something nonsensical. You may say, "be reasonable." That is the experience of reasoning. We can not explain it. As computer scientists, we care about logic because it is the calculus of computer science. What do we mean by that? Recall from your high school studies, we used differential equations to analyzed electrical circuits. Differential equations are calculus of electrical engineering.

0		<u></u>		
9	U	9	ື	

Example: applying logic

Logic is inferring conclusions from given premises

Example 1.1

- 1. Humans are mortal
- 2. Socrates is a human

Socrates is mortal



Commentary: Let us consider the examples of reasoning here. We think that the first reasoning is correct. However, the second reasoning is clearly faulty. What is wrong even though both the reasoning follow the same pattern? In the second reasoning, the first premise is talking about the size of the set of apostles instead of the properties of each of them. The English language made the matters ambiguous. We methods to separate good and bad reasonings.

6	6	9	0	
W	U	S	ື	

Instructor: Ashutosh Gupta

Automated reasoning aims to

enable machines to

identify the valid reasoning!!

Commentary: Now let us focus on the word 'automated'. Like many other aspects of human life, we use machines to help us in identifying the bad reasoning. Automated reasoning is the field that consists of the formalism, algorithms, and tools that enables the identification. In the course, we will study them.

\sim	0	0	\sim
(cc)	(ii)	\sim	വ
9	U	9	ື

CS 433 Automated Reasoning 2025

Instructor: Ashutosh Gupta

Automated reasoning is a backbone technology!!

Applications in verification, synthesis, solving NP-hard problems, and so on.

Automated reasoning for verification tools is like engines for the cars.

Commentary: Why do we need automated reasoning tools? There are many practical problems, where we need automated reasoning tools as an oracle to solve sub-problems. For example, software verification where we want to check if the software has indeed implemented the intended specification. The check boils down to automated reasoning problem.



Topic 1.2

Why do we need need automated reasoning? - Solving hard problems



Computers solve problems





► Weather prediction

Example 1.2

What have computers done for you?



Problems are





What is computational complexity?

Time used by best known algorithm to solve a problem in terms of input size.

Example 1.3 How much time it takes to sort an array?



Hard problems: $O(2^n)$

- Scheduling
- Sudoku solving
- Optimal circuits
- Playing Chess
- Password cracking
- Protein folding
- Traveling salesman
- SAT problem



A subclass of hard problems: non-deterministic polynomial(NP)

Hard to find solution, but checking the solution is easy!!

- Scheduling
- Sudoku solving
- Password cracking
- SAT problem

Open question: Are NP problems really hard?

We do not have a proof that there are no easy algorithm for the problems.



${\sf Outside} \ {\sf of} \ {\sf NP}$

Hard to find solution and hard to check the solution !!

- Playing Chess
- Protein folding
- Traveling salesman

Even when we have the folded structure of the protein, we cannot easily prove that this is the minimum energy configuration.



Hard problems

Everywhere and important

Needs solving



Impossible problems

There are problems for which we have a proof that there is no algorithm for them.

- Program correctness
- Synthesis



How do we solve?

Saam: Clever algorithms

► Daam: Brute force

► *Dand*: Heuristics

► *Bhed*: AI based approximate solving (e.g. AlphaFold)



▶ How much computation per second? 10^9

\blacktriangleright How much time? 10^5

\blacktriangleright How many computers? 10^5

$pprox 10^{19}$ is a hard limit.



Can you solve?

► 2-body problem



► 10²³-body problem

Medium sized problems are unsolved, e.g., biology.



Topic 1.3

Spectra of logic



Spectra of logic

Logic has been divided into increasing complexity of classes.

- 1. Propositional logic (PL)
- 2. First-order logic (FOL)
- 3. First-order logic with theories $(FOL(\mathcal{T}))$
- 4. Higher-order logic (HOL)



Commentary: Logic is a broad subject. We will study logic in the four classes of reasoning with increasing complexity. This structure allows us to study logic in convenient way. Each class introduces an additional level of complexity.



CS 433 Automated Reasoning 2025

Instructor: Ashutosh Gupta

Propositional logic (PL)

- Propositional logic
 - deals with propositions,
 - only infers from the structure over the propositions, and
 - does not look inside the propositions.

Commentary: Propositional logic is also called Boolean logic. It is the most basic form of reasoning. One needs this class to solve the puzzles such as Sudoku.

23

Example : propositional logic

Example 1.4

Is the following argument valid?

If the seed catalog is correct then if seeds are planted in April then the flowers bloom in July. The flowers do not bloom in July. Therefore, if seeds are planted in April then the seed catalog is not correct.

Let us symbolize our problem

If c then if s then f. not f. Therefore, if s then not c.

- c = the seed catalogue is correct
- s = seeds are planted in April
- f = the flowers bloom in July

PL reasons over propositional symbols and logical connectives

Commentary: The symbolic form of an argument is obtained after replacing propositions by their symbols. Writing an argument in symbolic form allows us to focus on the reasoning involved in the argument. One can easily see that the argument holds.

First-order logic (FOL)

First-order logic

- looks inside the propositions,
- much more expressive,
- includes parameterized propositions and quantifiers over individuals, and
- can express lots of interesting math.

Example 1.5

Is the following argument valid? Humans are mortal. Socrates is a human. Therefore, Socrates is mortal.

In the symbolic form,

For all x if H(x) then M(x). H(s). Therefore, M(s).

- H(x) = x is a human
- M(x) = x is mortal
- ▶ s = Socrates

FOL is not the most general logic. Many arguments can not be expressed in FOL

FOL+logical theories

In a theory, we study validity of FOL arguments under specialized assumptions (called axioms). Example 1.6

The number theory uses symbols $0, 1, ..., <, +, \cdot$ with specialized meanings

The following sentence has no sense until we assign the meanings to > and \cdot

$$orall x \; \exists p \; (p > x \land (orall v_1 orall v2 \; (v_1 > 1 \land v_2 > 1 \Rightarrow p
eq v_1 \cdot v_2)))$$

Under the meanings it says that there are arbitrarily large prime numbers.

In the earlier example, we had no interpretation of predicate 'x is human'. Here we precisely know what is predicate 'x < y'.

Commentary:	The specialized meaning are defined using axioms.	For example, the sentence $\forall x. 0 + x = x$ describe	es one of the properties of 0 and $+$.
©()\$0	CS 433 Automated Reasoning 202	5 Instructor: Ashutosh Gupta	IITB India

[Not in the course] Higher-order logic (HOL)

Higher-order logic

- includes quantifiers over "anything",
- consists of hierarchy first order, second order, third order and so on,
- most expressive logic.

Example 1.7

 $\forall \mathbf{P} \forall x. (P(x) \lor \neg P(x))$

The quantifier is over proposition P. Therefore, the formula belongs to the second-order logic.

Commentary: The first quantifier is not allowed in the first-order logic. the first-order logic quantifies over individuals, the second-order logic quantifies over sets, the third-order logic quantifies over set of sets, and so on.

0	0	0	\sim
(cc)	())	ഭാ	(9)
<u> </u>	Ś	-	9

Topic 1.4

Satisfiability problem



Satisfiability for reasoning

How to covert a reasoning problem into a computational question?

Answer: satisfiability problem

Commentary: Computers can compute. We need to pose a computational question to the computers to solve logical problems. Therefore, we are changing track to understand the satisfiability problem. We will see that we can translate all logical problems into satisfiability problem, which is a computational question.

29

© () \$0	CS 433 Automated Reasoning 2025	Instructor: Ashutosh Gupta	IITB India
-----------------	---------------------------------	----------------------------	------------

Example: satisfaction

Let x, y be rational variables.

Choose a value of x and y such that the following formula holds true.

x + y = 3

We say

$$\{x \mapsto 1, y \mapsto 2\} \models x + y = 3.$$

Commentary: x and y are not rational numbers. They are variables, i.e. symbols that can hold numbers. A model is an assignment to the variables of the formula. $m \models F$ is read as 'm satisfies F'.



Evaluation

For a given model m and formula F,

$$m \models F?$$

Example 1.8

$$\{x\mapsto 1, y\mapsto 2\}\models x+y=3.$$

Exercise 1.1

Exercise 1.2

Can we say something more about the last formula?

Satisfiablity problem

Satisfiability == Is there any model?

Is F SAT?

Harder problem!

Exercise 1.3

Are the following formulas sat?

 $\blacktriangleright x + y = 3 \land x > 0$

$$> x + y = 3 \land x > 0 \land y > 10$$

• $x > 0 \lor x < 1$ disjunction

Exercise 1.4

Can we say something more about the last formula?

Commentary: If one does not know that a formula has a model or not. Then one may ask if a given formula is satisfiable or not. A formula can be valid, i.e., all models satisfy the formula. A formula can be satisfiable, unsatisfiable, or valid.

0	0	0)
(cc)	(†)	(\$)	(ອ)

Reasoning == Satisfiability problem

All reasoning problems can be reduced to satisfiability problems.

Often abbreviated to SAT problem

Exercise 1.5

- a. How to convert checking a valid argument into a SAT problem?
- b. Convert argument "if $x \ge 2$ then $x \ge 1$." into a SAT problem.

Commentary: A typical argument looks like $Premise_1 \land ... \land Premise_n \Rightarrow Conclusion$. If we negate the logical statement, we obtain $Premise_1 \land ... \land Premise_2 \land \neg Conclusion$ after some simplifications. If we show that the negation is unsatisfiable, then only the argument is valid. Therefore, the validity check of an argument can be reduced to a satisfiability problem.

Example: SAT problem(contd.)

Let x, y be rational variables.

Choose a value of x and y such that the following formula holds true.



classes are polynomial, NP-complete, and Undecidable.

@**()**\$0

CS 433 Automated Reasoning 2025

Instructor: Ashutosh Gupta

Quantifier-free formulas

Quantifier-free formulas consists of

- theory atoms and
- Boolean structure

Example 1.9

$$(Theory atoms) \begin{array}{c} x + y = 3 \land y > 10 \land (x > 0 \lor x < -4) \\ \land \\ Boolean \ Structure \end{array}$$

Commentary:	We typically assume $\wedge(\mbox{conjunction})$ occurs in any formula.	We say a formula has Boolean structure if it has	\vee (disjunction).
@ () (\$)(0)	CS 433 Automated Reasoning 2025	Instructor: Ashutosh Gupta	IITB India

Propositional formulas

Propositional formulas are a special case, where the theory atoms are Boolean variables.

Example 1.10 Let p_1, p_2, p_3 be Boolean variables

 $p_1 \wedge \neg p_2 \wedge (p_3 \vee p_2)$

A satisfying model:

$$\{p_1\mapsto 1, p_2\mapsto 0, p_3\mapsto 1\}\models p_1\wedge \neg p_2\wedge (p_3\vee p_2)$$



A bit of jargon

Solvers for quantifier-free propositional formulas are called

SAT solvers.

Solvers for quantifier-free formulas with the other theories are called

SMT solvers.

SMT = satisfiability modulo theory



Theory solvers

SMT solvers are divided into two components.

- ► SAT solver: it solves the Boolean structure
- Theory solver: it solves the theory constraints

Example 1.11

Let x,y be rational variables.

$$x + y = 3 \land y > 10 \land x > 0$$

Since the formula has no \lor (disjunction), a solver of linear rational arithmetic can find satisfiable model using simplex algorithm.

Commentary: Si	implex can not directly handle the formulas that have disjunctions.			
@ () \$0	CS 433 Automated Reasoning 2025	Instructor: Ashutosh Gupta	IITB India	38

Exercise

Exercise 1.6 Give satisfying assignments of the following formulas

 $\blacktriangleright \neg p_1 \land (p_1 \lor \neg p_2)$

►
$$x < 3 \land y < 1 \land (x + y > 5 \lor x - y < 3)$$



Topic 1.5

Course contents and logistics



We will study the following topics

- Background: propositional and first-order logic (FOL) basics
- ► SAT solvers: satisfiability solvers for propositional logic
- SMT solvers: satisfiability modulo theory solvers
- Decision procedures: algorithms for solving theory constraints
- AI based approximate reasoning

Evaluation and website

Light evaluation

- Programming assignments: 41% (6% 10% 15% 10%)
- Online simple quizzes : 24% (2% each) only testing participation
- ▶ Midterm presentation: 10% (15 min) (Read a paper from the recent conference)
- ▶ Final: 25%

For the further information

https://www.cse.iitb.ac.in/~akg/courses/2025-ar/

All the assignments and slides will be posted on the website.



Topic 1.6

Problems



SMT solvers

Exercise 1.7

Which of the following are true about SMT solvers?

- 1. SMT solvers can make tea for us
- 2. SMT solvers can natively handle quantified formulas
- 3. Z3 is an SMT solver
- 4. The theory solvers in SMT solvers can only check satisfiability of conjunction of literals



Propagating equality

Exercise 1.8

In which of the following formulas, we can propagate equality x = t in F.

1.
$$\forall x.(x = t \Rightarrow F)$$

2. $\exists x.(x = t \Rightarrow F)$
3. $\forall x.(x = t \land F)$

4. $\exists x.(x = t \land F)$



Topic 1.7

Extra slides: understanding quantified formulas



Quantified formulas

Quantified formulas also include quantifiers.

Example 1.12

The following formulas says: give x such that for each y the body holds true.

$$\forall y. \ \underbrace{(x+y=3 \Rightarrow y > 10 \land (x > 0 \lor x < -4))}_{Body}$$

A satisfying model:

$$\{x \rightarrow -8\} \models \forall y. (x + y = 3 \Rightarrow y > 10 \land (x > 0 \lor x < -4))$$

Exercise 1.9

Can we eliminate y in the above formula to obtain constraints just over x?

Commentary: Sin	ce y is quantified, the model does not assign value to y .		
ତ୍ତାର୍ତ୍ତ	CS 433 Automated Reasoning 2025	Instructor: Ashutosh Gupta	IITB India

Example: understanding quatified formula

Example 1.13

Consider the following formula and the model again.

$$\{x \rightarrow -8\} \models \forall y. \ (x + y = 3 \Rightarrow y > 10 \land (x > 0 \lor x < -4))$$

Substitute the value of x in the formula

$$\forall y. \ (-8+y=3 \Rightarrow y > 10 \land (-8 > 0 \lor -8 < -4))$$

After simplification, we obtain

$$\forall y. (y = 11 \Rightarrow y > 10).$$

The above is clearly true.(Why?)

Exercise 1.10 *Is* $\forall y$. ($y = 11 \land y > 10$) *true*?



Exercise

Exercise 1.11

Give satisfying assignments of the following formula

$$\blacktriangleright \quad \forall x \ (x > y \Rightarrow \exists z \ (2z = x))$$



Topic 1.8

Extra slides: derived problems in logic



Problems in logic

A logical problem may come in several forms.

- Validity
- Implication
- Quantifier elimination
- Induction
- Maximum satisfiability
- Interpolation
- Abduction

A typical solver needs to be able to handle the above.

Commentary:	The last three logical problems will also be covered in the course.	However, we will not discuss them in the lecture.	
ାର୍ଚ୍ଚ	CS 433 Automated Reasoning 2025	Instructor: Ashutosh Gupta	IITB India

Validity

Is the formula true for all models?

$\models F?$

Is it harder problem than the SAT problem?

We can simply check satisfiability of $\neg F$.

Example 1.14

 $x > 0 \lor x < 1$ is valid because $x \le 0 \land x \ge 1$ is unsatisfiable.

Commentary: Any validity problem can be converted into a sat problem. Therefore, we do not need another solver for checking validity.

ାର୍ଭ	CS 433 Automated Reasoning	2025	Instructor: Ashutosh Gupta	IITB India
	· · · · · · · · · · · · · · · · · · ·			

Implication

$F \Rightarrow G?$

We need to check $F \Rightarrow G$ is a valid formula. We check if $\not\models \neg(F \Rightarrow G)$, which is equivalent to checking if $\not\models F \land \neg G$.

Example 1.15

Consider implication $(x = y + 1 \land y \ge z + 3) \Rightarrow x \ge z$.

After negating the implication, we obtain $x = y + 1 \land y \ge z + 3 \land x < z$.

After simplification, we obtain $x - z \ge 4 \land x - z < 0$.

Therefore, the negation is unsatisfiable and the implication is valid.

Commentary:	Many solvers provide interfaces to check implications,	which is an important API. Under the hood, t	he above procedure is followed.
ତାର୍ତ୍ତ	CS 433 Automated Reasoning 2025	Instructor: Ashutosh Gupta	IITB India

Quantifier elimination

given F, find G such that $G(y) \equiv \exists x. F(x, y)$

Is this a harder problem? yes(Why?)

@**()**\$0

Commentary: Quantifier elimination asks us to remove a given variable from a formula. It is projecting an object in lower dimension space. In many calculations, we need to eliminate variables. There are algorithms for quantifier elimination for some classes for formulas.

Instructor: Ashutosh Gupta

Example: quantifier elimination

Example 1.16

- *Consider formula* $\exists x. x > 0 \land x' = x + 1$
- After substituting x by x' 1, $\exists x. x' 1 > 0$.

Since x is not in the formula, we drop the quantifier and obtain x' > 1.

Exercise 1.12

- a. Eliminate quantifiers: $\exists x, y. \ x > 2 \land y > 3 \land y' = x + y$
- b. What do we do when \lor is in the formula?
- c. How to eliminate universal quantifiers?



Induction principle

 $F(0) \land \forall n. F(n) \Rightarrow F(n+1)$ $\forall n.F(n)$

Very difficult to automate.

Because, we need to prove $\forall n : G(n)$, but the induction proof technique fails on G.

Then, we need to find F(n) such that $\forall n : F(n) \Rightarrow G(n)$ and induction works on F(n).

Commentary: G(n) sare often poorly posed questions. We need to make a guess to find F(n) such that the induction method can be applied. A machine does not know how to make an informed guess.

Example : induction principle

Example 1.17

We prove $F(n) = (\sum_{i=0}^{n} i = n(n+1)/2)$ by induction principle as follows $F(0) = (\sum_{i=0}^{0} i = 0(0+1)/2)$

• We show that implication $F(n) \Rightarrow F(n+1)$ is valid, which is

$$(\sum_{i=0}^{n} i = n(n+1)/2) \Rightarrow (\sum_{i=0}^{n+1} i = (n+1)(n+2)/2).$$

Exercise 1.13

Show the above implication holds using a satisfiability checker.



End of Lecture 1

