COMP2022: Formal Languages and Logic

2017, Semester 1, Week 13

Joseph Godbehere

June 6, 2017

THE UNIVERSITY OF SYDNEY

Outline

- Proof of undecidability of the Halting problem
- Assignment 3 results (best proofs!)
- Exam advice
  - Topics
  - Structure
  - etc.
- Revision

Proving the Halting Problem is undecidable

Suppose the Halting problem is decidable.

Then there exists some universal Turing machine H such that $H(a, b)$ accepts if and only if the TM represented by a would halt on input b.

The language L of H is:

$$\{a, b \mid a \text{ represents a TM, } b \text{ is an input string, } a \text{ halts on } b\}$$

Proving the Halting Problem is undecidable

$H(a, b)$ accepts iff TM a halts on input b

Let $X(c)$ be a Turing machine which either:

- If $H(c, c)$ accepts, then loop forever
- If $H(c, c)$ rejects, then halt

Proving the Halting Problem is undecidable

Now consider what happens if we use "X, X" as input to H. i.e. what does X do when given its *own* representation as input.

Proving the Halting Problem is undecidable

Now consider what happens if we use "X, X" as input to H. i.e. what does X do when given its *own* representation as input.

Case 1: If $H(X, X)$ accepts, then X must halt on input X, because H is a solution to the Halting problem. However, X will loop forever because $H(X, X)$ accepts.

Proving the Halting Problem is undecidable

Now consider what happens if we use "X, X" as input to H. i.e. what does X do when given its *own* representation as input.

Case 1: If $H(X, X)$ accepts, then X must halt on input X, because H is a solution to the Halting problem. However, X will loop forever because $H(X, X)$ accepts.

Case 2: If $H(X, X)$ rejects, then X cannot halt on input X, because H is a solution to the Halting problem. However, X will halt because $H(X, X)$ rejected.

Proving the Halting Problem is undecidable

Now consider what happens if we use "X, X" as input to H. i.e. what does X do when given its *own* representation as input.

Case 1: If $H(X, X)$ accepts, then X must halt on input X, because H is a solution to the Halting problem. However, X will loop forever because $H(X, X)$ accepts.

Case 2: If $H(X, X)$ rejects, then X cannot halt on input X, because H is a solution to the Halting problem. However, X will halt because $H(X, X)$ rejected.

Both cases lead to contradictions, so the assumption was incorrect. i.e. The Halting problem cannot be decidable.

Statistics

	attempts	proofs	lines	errors
Q1	493	173	6007	4540
Q2	237	173	1356	1474
Q3	403	178	2569	1969
Q4	271	152	4182	2266

Most persistent student submitted over 800 lines

Most painful inference rule

Disjunctive Syllogism (DS)

$$\frac{(A \vee B),\ \neg A}{B}$$

- 1400 errors, 603 correct uses

Loneliest inference rule

Constructive Dilemma (CD)

$$\frac{((A \to B) \land (C \to D)),\ (A \lor C)}{(B \lor D)}$$

- Used correctly only 11 times

Shortest proof: Q1

First to find a proof this short was zzha4377 (+ 14 others)

P.	L.	Formula	Justification	Refs
1	1	A	Premise	
2	2	$\neg\neg(B \leftrightarrow (A \wedge \neg B))$	Premise	
2	3	$(B \leftrightarrow (A \wedge \neg B))$	Double Negation	2
2	4	$((B \to (A \wedge \neg B)) \wedge ((A \wedge \neg B) \to B))$	Equivalence	3
2	5	$(B \to (A \wedge \neg B))$	Simplification	4
2	6	$(\neg B \vee (A \wedge \neg B))$	Def. of Implication	5
2	7	$((\neg B \vee A) \wedge (\neg B \vee \neg B))$	Distribution	6
2	8	$((\neg B \vee \neg B) \wedge (\neg B \vee A))$	Commutation	7
2	9	$(\neg B \vee \neg B)$	Simplification	8
2	10	$\neg B$	Idempotence	9
1, 2	11	$(A \wedge \neg B)$	Conjunction	1, 10
2	12	$(((A \wedge \neg B) \to B) \wedge (B \to (A \wedge \neg B)))$	Commutation	4
2	13	$((A \wedge \neg B) \to B)$	Simplification	12
1, 2	14	B	Modus Ponens	11, 13
1, 2	15	$(B \wedge \neg B)$	Conjunction	10, 14
1	16	$\neg(B \leftrightarrow (A \wedge \neg B))$	Indirect Proof	2, 15

Shortest proof: Q2

First to find a proof this short was lili5475 (+ 69 others)

P.	L.	Formula	Justification	Refs
1	1	A	Premise	
2	2	$((A \wedge \neg B) \rightarrow B)$	Premise	
2	3	$(A \rightarrow (\neg B \rightarrow B))$	Exportation	2
1, 2	4	$(\neg B \rightarrow B)$	Modus Ponens	1, 3
1, 2	5	$(\neg\neg B \vee B)$	Def. of Implication	4
1, 2	6	$(B \vee B)$	Double Negation	5
1, 2	7	B	Idempotence	6

Shortest proof: Q3

First to find a proof this short was lili5475 (+ 55 others)

P.	L.	Formula	Justification	Refs
1	1	B	Premise	
1	2	$\neg\neg B$	Double Negation	1
1	3	$(\neg\neg B \vee \neg A)$	Addition	2
1	4	$\neg(\neg B \wedge A)$	DeMorgan's Laws	3
1	5	$(\neg\neg B \wedge \neg(\neg B \wedge A))$	Conjunction	2, 4
1	6	$\neg(\neg B \vee (\neg B \wedge A))$	DeMorgan's Laws	5
1	7	$\neg(B \rightarrow (\neg B \wedge A))$	Def. of Implication	6
1	8	$\neg(B \rightarrow (A \wedge \neg B))$	Commutation	7

Shortest proof: Q4

First to find a proof this short was kste4439 (+ 3 others)

P.	L.	Formula	Justification	Refs
1	1	A	Premise	
2	2	$(B \to (A \wedge \neg B))$	Premise	
2	3	$(\neg B \vee (A \wedge \neg B))$	Def. of Implication	2
2	4	$((\neg B \vee A) \wedge (\neg B \vee \neg B))$	Distribution	3
2	5	$((\neg B \vee A) \wedge \neg B)$	Idempotence	4
2	6	$(\neg B \wedge (\neg B \vee A))$	Commutation	5
2	7	$\neg B$	Simplification	6
1, 2	8	$(A \wedge \neg B)$	Conjunction	1, 7
1, 2	9	$((A \wedge \neg B) \wedge \neg B)$	Conjunction	7, 8
1, 2	10	$(\neg\neg(A \wedge \neg B) \wedge \neg B)$	Double Negation	9
1, 2	11	$\neg(\neg(A \wedge \neg B) \vee B)$	DeMorgan's Laws	10
1, 2	12	$\neg((A \wedge \neg B) \to B)$	Def. of Implication	11
1	13	$((B \to (A \wedge \neg B)) \to \neg((A \wedge \neg B) \to B))$	Conditional Proof	2, 12
1	14	$(\neg(B \to (A \wedge \neg B)) \vee \neg((A \wedge \neg B) \to B))$	Def. of Implication	13
1	15	$\neg((B \to (A \wedge \neg B)) \wedge ((A \wedge \neg B) \to B))$	DeMorgan's Laws	14
1	16	$\neg(B \leftrightarrow (A \wedge \neg B))$	Equivalence	15

Exam Notes

You are allowed to bring a single sheet of A4 paper containing your own notes (printed or written, on both sides.)

The tables of Equivalence Laws and Inference Rules for propositional and predicate logic will be provided to you with the exam paper (the exact same sheet you can download on Ed.)

Exam structure

There are some short answer questions testing general knowledge of the course.

Most of the exam consists of practical questions, applying the methods we've used in tutorials and assignments (for example, minimising a DFA)

Possible exam topics

Possible exam topics

Weeks 1 to 12

Regular languages

- DFA
- NFA
- Regular expressions
- Transformations between representations
- Minimal DFA
- Prove that a language is regular by giving a DFA, NFA, or Regular Expression
- Prove that a language is *not* regular by using the Pumping Lemma to find a contradiction

Context-Free languages

- Context-Free Grammars
- Push Down Automata (PDA)
- LL(1) grammars
  - Identifying non-LL(1) grammars
  - Finding equivalent LL(1) grammars
  - FIRST and FOLLOW sets
  - Building parse tables
- Chomsky Normal Form

Propositional Logic

- Translating arguments from English into Propositional Logic
- Proofs in the Natural Deduction System
  - Equivalence Laws
  - Inference Rules
- Proof by Resolution
  - Conjunctive Normal Form
  - Resolution Rule (+ Indirect Proof)
- Proving logical tautologies
  - NDS proof
  - Quine's method

Predicate Logic

- Translating arguments from English into Predicate Logic
- Proofs in the Natural Deduction System
  - Equivalence Laws
    - Plus laws involving quantifiers
  - Inference Rules
    - Plus rules involving quantifiers
- Proving if a formula is valid (or invalid)
- Prenex Conjunctive Normal Form

Turing Machines

- Turing Machines
  - Designing a TM to solve a problem

- Concepts and examples of:
  - Computability
  - Decidability
  - (In)tractability

Time management

- There are a lot of topics!
- Rule of thumb: number of marks = number of minutes
  - The 2 hour exam has 100 marks total
- Don't know how to proceed with a question?
  - Skip it!
  - Do questions you are confident on first
  - Then go back to the harder questions

Working

- Show your working
  - An incorrect answer without working will get zero
  - An incorrect answer with working might get partial marks

- Some questions *require* you to show your working
  - i.e. you cannot get full marks without showing how you reached the solution

Revision sessions

When:

- Tuesday afternoon?
- Thursday afternoon?
- Exact times will be announced on Ed later this week

Where:

- SIT Lecture Theatre 123

What:

- There will be no specific content.
- Just come with questions.

Links to other units

- all IT courses, in particular:
  - COMP2007/2907: Algorithms and Complexity
  - COMP3308/3608: Introduction to AI
  - COMP3109: Programming Languages and Paradigms
- MATH3066: Algebra and Logic
  - Very interesting mathematics course. Explores predicate logic and Turing machines more rigourously than we did this semester.
- PHIL2650: Logic and Computation

That's it!

Thanks for the fun semester.

Don't forget to complete the Unit of Study Survey:

http://sydney.edu.au/itl/surveys/complete/