University of Victoria

CSC 320 - Spring 2023

Foundations of Computer Science

Tutorial 09

Teaching Team

Learning Outcomes:

- Use reduction to prove a language is undecidable.
- Become familiar with reduction.

Interesting Article:

"Decidable and Undecidable Problems about Quantum Automata" [1]

March 21st, 2023

Consider the language:

$$A_{TM} = \{ < M, w > \mid M \text{ accepts } w \}$$

It is easy to show the A_{TM} is Turing Recognizable. Given an input $< M, w >$, simulate M on w. If M accepts w then ***accept*** and if M rejects w then ***reject***.

Theorem A_{TM} is undecidable.

Proof

Suppose A_{TM} is decidable. Then there exists a halting TM H that on input $\langle M, \omega \rangle$, accepts if M accepts ω and rejects if M does not accept ω.

Using H, construct another machine N as follows.

Description of Turing Machine N

Input : $\langle M \rangle$ where M is a TM

A description of H is hardcore into the machine N.

1. Simulate H on $\langle M, \langle M \rangle \rangle$.
2. If H rejects then accept else reject.

By definition!

Note that the machine H is a halting TM, hence N is also a halting TM.
Now consider what happens when the machine N is provided with the input $\langle N \rangle$.
the constructed machine

$$\boxed{N \text{ accepts } \langle N \rangle \iff H \text{ rejects } \langle N, \langle N \rangle \rangle \iff N \text{ does not accept } \langle N \rangle}$$

This is a contradiction. Hence A_{TM} is undecidable. $\square$

Why is this important?

We can now show that a language is undecidable by "reducing" a known undecidable language to the given language.

Reducibility : Informally, converting one problem to another.

if I can read a map I can go around a city

Example
- going around in a city reduces to ability to read a map
- getting good grades reduces to solving problems
- multiplication reduces to addition (doing it several times)

Decidability and Reductions

Consider we have two problems: A and B. Furthermore, we show that we can reduce A to B. That is, $A \rightarrow B$.

Remember that a reduction means we can solve A using B, If we have the answer to B, then we have the answer to A as well!

Now consider what if we know that A is really difficult to solve? Then B must also be really difficult to solve, since otherwise we just solved it using the reduction.

So, what if we know that B is really easy to solve? Then A is also easy to solve since we can use the answer to B to solve A.

As shown in the reduction.

Proving a Language is Undecidable using a Reduction

We will use proof by contradiction to construct our proof.

How? If we are trying to prove that a problem X is undecidable, we will first assume for contradiction that X is decidable and therefore there exists some decider R which decides X. Next we will reduce a known undecidable problem (e.g., A_{TM} to X).

reduction is $A_{TM} \rightarrow X$

More specifically, we will build a decider for A_{TM} that uses R. ← Remember our goal, thus we will attempt to build since we want a contradiction.

So, by creating a decider for A_{TM}, we have made a contradiction since we know, by theorem, that A_{TM} is undecidable. Thus, we can also conclude th the language X must also not be decidable.

General Steps to Prove a Language is Undecidability using a Reduction

Step 1 Assume for contradiction that X is decidable. ← We must state this before continuing with our proof! We cannot simply create our TM.
Then, by definition, there exists some Turing Machine (TM) R which is a decider for X.

Step 2 Write a skeleton reduction from a known undecidable language to X. ← We will use A_{TM}! But think about how we could use another undecidable language to prove th same language. Does the structure change?

Step 3 Create a third TM C whose language is carefully created such that by running R on input ⟨C⟩, R will give us the answer to the problem we are reducing from (e.g., A_{TM}).

R is our decider for X! So, we want R on input ⟨C⟩ to provide an answer for A_{TM}.

Step 4 Finish writing the reduction!
Explain that we wrote a decider for an undecidable problem which is a contradiction!

Step 5 Conclude that X is undecidable.

The proof is not complete unless we make this statement!

Note: We can simply state that "X is undecidable" after writing the contents of Step 4, but we can also include a more detailed final statement.

Observe that 0^n1^n is non-regular and Σ^* is regular.

$- L(C) = \Sigma^*$ if $w \in L(M)$

Question 01

Prove that the following language is undecidable by reduction from A_{TM}.

$\overline{A_{TM}} \leq \text{Regular}_{TM}$

$$A_{TM} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ accepts } w\}$$

$$\text{Regular}_{TM} = \{\langle M\rangle \mid M \text{ is a TM and } L(M) \text{ is a regular language}\}$$

Goal Prove that Regular_{TM} is undecidable.

Step 1 Assume for contradiction that Regular_{TM} is decidable.
Then, by definition, there exists some decider R which decides Regular_{TM}.

Step 2 We will now build a decider S for A_{TM} using R. ← We know this isn't possible! However, we will try to create our TM C which satisfies A_{TM}.

S = "On input $\langle M, w\rangle$: where M is a TM and w is a string

Step 3

1. Construct TM C. ← Description of M and w hardcoded into its description!

   C = "On input x:
   1. If x is of the form 0^n1^n, accept.
   2. If x does not have this form:
      – Run M on input w and
        i) if M accepts w, accept.
        Otherwise, reject."

2. Run R on input $\langle C\rangle$. ← Observe that we never actually run C, we use R to decide a property of C.
3. If R accepts, accept.
   If R rejects, reject."

Step 4 Note that $L(C)$ is Σ^* (which is regular) if M accepts w. ← R is a decider since each step will halt.
$L(C)$ is 0^n1^n (non-regular) if M does not accept w.
Thus, $L(C)$ is regular if and only if (iff) M accepts w. ← if R decides that $L(C)$ is regular then...
So R accepts iff M accepts w.
Therefore, S is a decider for A_{TM}, but this is a contradiction since A_{TM} is undecidable.

Step 5 We can then conclude that Regular_{TM} is also undecidable. ∎ ← Remember we cannot stop at step 4 with the statement "a contradiction" – It does not bring us to the "goal" of our proof. We must be explicit and formal!

Question 02

Prove that the following language is undecidable by reduction from A_{TM}.

General Idea

If M accepts 0011 then M accepts 1100

$$A_{TM} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ accepts } w\}$$

$$S_{TM} = \{\langle M\rangle \mid M \text{ is a TM that accepts } w^r \text{ whenever it accepts } w\}$$

Goal Prove that S_{TM} is undecidable

Step 1 Assume for contradiction that S_{TM} is decidable.
Then, by definition, there exists some decider R which decides S_{TM}.

Step 2 We will now build a decider S for A_{TM} using R. ← We know this isn't possible! However, we will try to create our TM C which satisfies A_{TM}.

S = "On input $\langle M, w\rangle$: Where M is a TM and w is a string

Step 3 1. Construct TM C. ← Description of M and w hardcoded into its description!

C = "On input x :

1. If x is of the form 0^+1^+, accept. ← (i.e., 00^*11^*)
2. If x does not have this form :
   - Run M on input w and
     i) if M accepts w, accept.
     Otherwise, reject."

2. Run R on input $\langle C\rangle$. ← Observe that we never actually run C, we use R to decide a property of C.
3. If R accepts, accept.
   If R rejects, reject."

Step 4 Note the reason this works is similar to the correctness proof of $Regular_{TM}$. ← R is a decider since each step will halt.

So, S is a decider for A_{TM}.

Step 5 We can then conclude that S_{TM} is also undecidable. $\square$ ← Remember we cannot stop at step 4 with the statement "a contradiction" – It does not bring us to the "goal" of our proof. We must be explicit and formal!

Question 04

Prove that the following language is undecidable by reduction from A_{TM}.

$$A_{TM} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ accepts } w\}$$

$$E_{TM} = \{\langle M\rangle \mid M \text{ is a TM and } L(M) = \emptyset\}$$

$$= \{\langle M_1\rangle, \langle M_2\rangle, \langle M_3\rangle, \dots \mid L(M_i) = \emptyset,\ i \geq 0\}$$

Goal Prove that E_{TM} is undecidable.

Step 1 Assume for contradiction that E_{TM} is decidable.
Then, by definition, there exists some decider R which decides E_{TM}.

Step 2 We will now build a decider S for A_{TM} using R. ← We know this isn't possible! However, we will try to create our TM C which satisfies A_{TM}.

S = "On input $\langle M, w\rangle$:

Step 3

1. Construct TM C. ← Description of M and w hardcoded into its description!

   C = "On input x:
   1. If M accepts w, accept.
   2. If M does not accept w, reject."

2. Run R on input $\langle C\rangle$.

3. If R accepts, reject.
   If R rejects, accept."

If we accept w then $L(C) \neq \emptyset$.

Step 4 Note that the language of C is empty if and only if (iff) M does not accept w.
Therefore, R accepts iff M does not accept w, so S will reject.

Thus, S is a decider for A_{TM}, but A_{TM} is undecidable. So we obtain a contradiction.

Step 5 We can then conclude that E_{TM} is also undecidable. $\square$ ← Remember we cannot stop at step 4 with the statement "a contradiction"
– It does not bring us to the "goal" of our proof. We must be explicit and formal!

Question 05

Prove that the following language is undecidable by reduction from ALL_{CFG}.

$$ALL_{CFG} = \{\langle G \rangle \mid G \text{ is a CFG and } L(G) = \Sigma^*\}$$

$$EQ_{CFG} = \{\langle G, H \rangle \mid G \text{ and } H \text{ are CFGs and } L(G) = L(H)\}$$

Goal Prove that EQ_{CFG} is undecidable.

Step 1 Assume for contradiction that EQ_{CFG} is decidable.
Then, by definition, there exists some decider R which decides EQ_{CFG}.

Step 2 We will now build a decider S for ALL_{CFG} using R.

S = " On input $\langle G \rangle$:

Step 3

1. Construct a Context Free Grammar (CFG) H
   — where $L(H) = \Sigma^*$
2. Run R on input $\langle G, H \rangle$.
3. If R accepts, accept.
   If R rejects, reject."

Step 4 Note that this is a decider since every step halts.
We know this since if R decides accept, we know that $L(G) = L(H) = \Sigma^*$
And if R decides reject, we know that $L(G) \neq \Sigma^*$.

Therefore, S is a decider for ALL_{CFG}. We know that this is a contradiction since ALL_{CFG} is undecidable.

Step 5 We can then conclude that EQ_{CFG} is also undecidable. ◻

Remember we cannot stop at step 4 with the statement "a contradiction"
– It does not bring us to the "goal" of our proof. We must be explicit and formal!

Resources

[1] V. D. Blondel, E. Jeandel, P. Koiran, and N. Portier, "Decidable and undecidable problems about quantum automata," *SIAM Journal on Computing*, vol. 34, no. 6, pp. 1464–1473, 2005. DOI: 10.1137/S0097539703425861. [Online]. Available: https://doi.org/10.1137/S0097539703425861.