Proving Undecidability via Reductions

Alice Gao
Lecture 24

Based on work by J. Buss, L. Kari, A. Lubiw, B. Bonakdarpour, D. Maftuleac, C. Roberts, R. Trefler, and P. Van Beek
Outline

Proving Undecidability via Reductions
Learning Goals
Revisiting the Learning Goals
Learning Goals

By the end of this lecture, you should be able to:
Proving that a problem is undecidable by a reduction from the halting problem

▶ Define reduction.
▶ Describe at a high level how we can use reduction to prove that a decision problem is undecidable.
▶ Prove that a decision problem is undecidable by using a reduction from the halting problem.
We proved that the Halting problem is undecidable. How do we prove that another problem is undecidable?

» We could prove it from scratch, or
» We could prove that it is as difficult as the halting problem. Hence, it must be undecidable.
We will prove undecidability via reductions. Problem A is reducible to problem B.

- Given an algorithm for solving $B$, we could use it to solve $A$.
- If $B$ is decidable, then $A$ is decidable.
- If $A$ is undecidable, then $B$ is undecidable.
Theorem: Problem X is undecidable.

**Proof by Contradiction.**

Assume that there is an algorithm $B$, which solves problem $X$.

We will construct algorithm $A$, which uses algorithm $B$ to solve the halting problem. (Describe algorithm $A$.)

Since algorithm $B$ solves problem $X$, algorithm $A$ solves the halting problem.

This contradicts with the fact that the halting problem is undecidable. Therefore, algorithm $B$ does not exist. $\square$
Example 1 of reduction proofs

The halting-no-input problem:

*Given a program P which takes no input, does P halt?*

Theorem: The halting-no-input problem is undecidable.
Example 2 of reduction proofs

The both-halt problem:

*Given two programs* $P_1$ and $P_2$ *which take no input,*

*do both programs halt?*

Theorem: The both-halt problem is undecidable.
CQ 1: To prove that the both-halt problem is undecidable, does the following reduction work?

Let $P_1$ run $P$ with input $I$. Let $P_2$ do nothing and terminate immediately.

(A) Yes
(B) No
(C) I don’t know.
CQ 2: To prove that the both-halt problem is undecidable, does the following reduction work?

Let $P_1$ run $P$ with input $I$. Let $P_2$ run an infinite loop and never terminate.

(A) Yes
(B) No
(C) I don’t know.
Example 3 of reduction proofs

The exists-halting-input problem

*Given a program $P$, does there exist an input $I$ such that $P$ halts with input $I$?*

Theorem The exists-halting-input problem is undecidable.
The partial-correctness problem

Given a Hoare triple \( \{ P \} C \{ Q \} \),

is the triple satisfied under partial correctness?

Theorem: The partial-correctness problem is undecidable.
Revisiting the learning goals

By the end of this lecture, you should be able to:
Proving that a problem is undecidable by a reduction from the halting problem

▶ Define reduction.
▶ Describe at a high level how we can use reduction to prove that a decision problem is undecidable.
▶ Prove that a decision problem is undecidable by using a reduction from the halting problem.