Halting Problem

Jidong Yuan
Beijing Jiaotong University
yuanjd@bjtu.edu.cn

SCC 120: Fundamentals of Computer Science

Halting problem

A decision problem

- $\mathbb{P}$ is the set of programs in the universe
- $\mathbb{I}$ is the set of inputs in the universe
- Consider any program p and input i such that $(p, i) \in \mathbb{P} \times \mathbb{I}$
- Question: Does p halt for i?

Some example programs

Do they halt?

- "Hello world"

```
main {
    print(''Hello world'')
}
```

  The following program

```
main {
    while(true) {
        read(input)
        print(input + ''5'') }
    }
```

- Algorithms we studied for searching and sorting

Halting problem

Solutions?

- Suppose you came up with a clever algorithm `halts(p, i)` that returns true if and only if p halts for i
  - `halts(p, i)` returns true if p halts for i, and
  - `halts(p, i)` returns false, otherwise

Halting problem

Undecidable

- No such algorithm can exist!
  - Specifically, no such algorithm that gives the correct answer within a finite amount of time can exist

Proof by Contradiction

By Turing

- Assume you had an algorithm `halts(p, i)` that returned true iff p halted for i
- Consider the following code

```
prog(z) {
        if(halts(z,z))
            loop forever
        else stop
}
```

- Run `prog` with `prog` as its argument. What happens?

Proof by Contradiction (cont.)

By Turing

```
prog(z) {
        if(halts(z,z))
            loop forever
         else stop
}
```

- Run `prog` with `prog` as its argument. What happens?
- If `halts(prog,prog)` returns true, meaning that `prog` halts on input `prog`, then it loops forever—a contradiction!
- If `halts(prog,prog)` returns false, meaning that `prog` does not halt on input `prog`, then it halts—again, a contradiction!
- Therefore, our initial assumption about $\mathrm{halts}(p,i)$ must be false

Application of the Halting Problem (HP)

Showing some other problem X to be undecidable

- *Reduce* HP to X
  - An algorithm for deciding X means an algorithm for deciding HP
  - However, since HP is undecidable, it must be the case that X is undecidable
- Example: *Dead Code* problem
  - For any program p, any input i, and any line n in p, does p on i execute n?

Dead Code Solution

Proof by contradiction

1. Assume that Dead Code (D) is decidable, meaning that it is possible to tell whether or not a program p reaches line k on input i.
2. Consider any program p. Let line k be the end of p. If p reaches k on input i, then we know that p halts.
3. Replace every occurrence of `stop` in the program with `goto line k`. Notice that this program behaves exactly the same as before, except that it halts iff it gets to line k.
4. Because D is decidable, we can tell whether or not p reaches k on i. This means that we can tell whether or not p halts on i.
5. However, we know that H is undecidable. This means that our assumption that D is decidable must be false.
6. Hence D is undecidable.