Fundamentele Informatica 3

voorjaar 2019

http://www.liacs.leidenuniv.nl/~vlietrvan1/fi3/

Rudy van Vliet
kamer 140 Snellius, tel. 071-527 2876
rvvliet(at)liacs(dot)nl

college 7, 25 maart 2019

9. Undecidable Problems
9.2. Reductions and the Halting Problem
9.3. More Decision Problems Involving Turing Machines

Huiswerkopgave,
inleverdatum vandaag, 11:05 uur

A slide from lecture 6:

For general decision problem P and reasonable encoding e,

$$\begin{aligned} Y(P) &= \{e(I) \mid I \text{ is yes-instance of } P\} \\ N(P) &= \{e(I) \mid I \text{ is no-instance of } P\} \\ E(P) &= Y(P) \cup N(P) \end{aligned}$$

$E(P)$ must be recursive

A slide from lecture 6:

Definition 9.3. Decidable Problems

If P is a decision problem, and e is a reasonable encoding of instances of P over the alphabet Σ, we say that P is *decidable* if $Y(P) = \{e(I) \mid \; I \text{ is a yes-instance of } P\}$ is a recursive language.

A slide from lecture 6:

Definition 9.6. Reducing One Decision Problem to Another ...

Suppose P_1 and P_2 are decision problems. We say P_1 is reducible to P_2 ($P_1 \leq P_2$)

- if there is an algorithm
- that finds, for an arbitrary instance I of P_1, an instance $F(I)$ of P_2,
- such that

  for every I the answers for the two instances are the same,

  or I is a yes-instance of P_1

    if and only if $F(I)$ is a yes-instance of P_2.

...

A slide from lecture 6:

Theorem 9.7.

. . .

Suppose P_1 and P_2 are decision problems, and $P_1 \leq P_2$. If P_2 is decidable, then P_1 is decidable.

A slide from lecture 6:

Two more decision problems:

Accepts: Given a TM T and a string w, is $w \in L(T)$?

Halts: Given a TM T and a string w, does T halt on input w ?

A slide from lecture 6:

Theorem 9.8. Both *Accepts* and *Halts* are undecidable.

Proof.

1. Prove that *Self-Accepting* $\leq$ *Accepts* . . .

2. Prove that *Accepts* $\leq$ *Halts* . . .

In context of decidability: decision problem $P \approx$ language $Y(P)$

Question

> "is instance I of P a yes-instance ?"

is essentially the same as

> "does string x represent yes-instance of P ?",

i.e.,

> "is string $x \in Y(P)$?"

9.3. More Decision Problems Involving Turing Machines

Accepts: Given a TM T and a string x, is $x \in L(T)$?
Instances are . . .

Halts: Given a TM T and a string x, does T halt on input x ?
Instances are . . .

Self-Accepting: Given a TM T, does T accept the string $e(T)$?
Instances are . . .

Accepts: Given a TM T and a string x, is $x \in L(T)$?
Instances are . . .

Halts: Given a TM T and a string x, does T halt on input x ?
Instances are . . .

Self-Accepting: Given a TM T, does T accept the string $e(T)$?
Instances are . . .

Now fix a TM T:
T-*Accepts*: Given a string x, does T accept x ?
Instances are . . .
Decidable or undecidable ? (cf. **Exercise 9.7.**)

Exercise 9.7.

As discussed at the beginning of Section 9.3, there is at least one TM T such that the decision problem

“Given w, does T accept w ?”

is unsolvable.

Show that every TM accepting a nonrecursive language has this property.

Theorem 9.9. The following five decision problems are undecidable.

1. *Accepts*-Λ: Given a TM T, is $\Lambda \in L(T)$?

Proof.

1. Prove that *Accepts* $\leq$ *Accepts*-Λ . . .

Reduction from *Accepts* to *Accepts*-Λ.

Instance of *Accepts* is (T_1, x) for TM T_1 and string x.
Instance of *Accepts*-Λ is TM T_2.

$T_2 = F(T_1, x) =$

$$\textit{Write}(x) \to T_1$$

T_2 accepts Λ, if and only if T_1 accepts x.

If we had an algorithm/TM A_2 to solve *Accepts*-Λ,
then we would also have an algorithm/TM A_1 to solve *Accepts*,
as follows:

A_1:
Given instance (T_1, x) of *Accepts*,
1. construct $T_2 = F(T_1, x)$;
2. run A_2 on T_2.

A_1 answers 'yes' for (T_1, x),
if and only if A_2 answers 'yes' for T_2,
if and only T_2 accepts Λ,
if and only if T_1 accepts x.

Exercise 9.8.

Show that for every $x \in \Sigma^*$, the problem *Accepts* can be reduced to the problem:

Given a TM T, does T accept x?

(This shows that, just as *Accepts*-Λ is unsolvable, so is *Accepts*$-x$, for every x.)

Theorem 9.9. The following five decision problems are undecidable.

2. *AcceptsEverything*:
   Given a TM T with input alphabet Σ, is $L(T) = \Sigma^*$?

Proof.

2. Prove that *Accepts-Λ* $\leq$ *AcceptsEverything* . . .

Accepts-Λ: Given a TM T, is $\Lambda \in L(T)$?

Exercise 9.9.

Construct a reduction from *Accepts*-Λ to *Accepts*-$\{\Lambda\}$:

Given a TM T, is $L(T) = \{\Lambda\}$?

Theorem 9.9. The following five decision problems are undecidable.

3. *Subset*: Given two TMs T_1 and T_2, is $L(T_1) \subseteq L(T_2)$?

Proof.

3. Prove that *AcceptsEverything* $\leq$ *Subset* . . .

Theorem 9.9. The following five decision problems are undecidable.

4. *Equivalent*: Given two TMs T_1 and T_2, is $L(T_1) = L(T_2)$

Proof.

4. Prove that *Subset* $\leq$ *Equivalent* . . .

‘The intersection of two Turing machines’

Theorem 9.9. The following five decision problems are undecidable.

4. *Equivalent*: Given two TMs T_1 and T_2, is $L(T_1) = L(T_2)$

Proof.

4. Prove that *Subset* $\leq$ *Equivalent* . . .

Subset: Given two TMs T_1 and T_2, is $L(T_1) \subseteq L(T_2)$?

Equivalent: Given two TMs T_1 and T_2, is $L(T_1) = L(T_2)$

Exercise 9.10.

a. Given two sets A and B, find two sets C and D, defined in terms of A and B, such that $A = B$ if and only if $C \subseteq D$.

b. Show that the problem *Equivalent* can be reduced to the problem *Subset*.

AcceptsEverything:

Given a TM T with input alphabet Σ, is $L(T) = \Sigma^*$?

Equivalent: Given two TMs T_1 and T_2, is $L(T_1) = L(T_2)$

Exercise 9.11. Construct a reduction from *AcceptsEverything* to the problem *Equivalent*.

Accepts-Λ: Given a TM T, is $\Lambda \in L(T)$?

Theorem 9.9. The following five decision problems are undecidable.

5. *WritesSymbol*:
   Given a TM T and a symbol a in the tape alphabet of T, does T ever write a if it starts with an empty tape ?

Proof.

5. Prove that *Accepts-Λ* $\leq$ *WritesSymbol* . . .

AtLeast10MovesOn-Λ:
Given a TM T, does T make at least ten moves on input Λ ?

WritesNonblank: Given a TM T, does T ever write a nonblank symbol on input Λ ?

Theorem 9.10.
The decision problem *WritesNonblank* is decidable.

Proof. . .

Definition 9.11. A Language Property of TMs

A property R of Turing machines is called a *language property* if, for every Turing machine T having property R, and every other TM T_1 with $L(T_1) = L(T)$, T_1 also has property R.

A language property of TMs is *nontrivial* if there is at least one TM that has the property and at least one that doesn't.

In fact, a language property is a property *of the languages accepted by* TMs.

Theorem 9.12. Rice's Theorem
If R is a nontrivial language property of TMs, then the decision problem

P_R: Given a TM T, does T have property R ?

is undecidable.

Proof. . .

Prove that *Accepts*-$\Lambda \leq P_R$ . . .

(or that *Accepts*-$\Lambda \leq P_{\text{not}-R}$ . . .)

Examples of decision problems to which Rice's theorem can be applied:

1. *Accepts-L*: Given a TM T, is $L(T) = L$? (assuming . . .)
2. *AcceptsSomething*:
   Given a TM T, is there at least one string in $L(T)$?
3. *AcceptsTwoOrMore*:
   Given a TM T, does $L(T)$ have at least two elements ?
4. *AcceptsFinite*: Given a TM T, is $L(T)$ finite ?
5. *AcceptsRecursive*:
   Given a TM T, is $L(T)$ recursive ? (note that . . .)

All these problems are undecidable.

Rice's theorem cannot be applied (directly)

- if the decision problem does not involve just one TM

Equivalent: Given two TMs T_1 and T_2, is $L(T_1) = L(T_2)$

Rice's theorem cannot be applied (directly)

• if the decision problem does not involve just one TM
Equivalent: Given two TMs T_1 and T_2, is $L(T_1) = L(T_2)$

• if the decision problem involves the *operation* of the TM
WritesSymbol: Given a TM T and a symbol a in the tape alphabet of T, does T ever write a if it starts with an empty tape ?
WritesNonblank: Given a TM T, does T ever write a nonblank symbol on input Λ ?

• if the decision problem involves a *trivial* property
Accepts-NSA: Given a TM T, is $L(T) =$ *NSA* ?

Exercise 9.12.
For each decision problem below, determine whether it is decidable or undecidable, and prove your answer.

a. Given a TM T, does it ever reach a nonhalting state other than its initial state if it starts with a blank tape?

Exercise 9.12.
For each decision problem below, determine whether it is decidable or undecidable, and prove your answer.

b. Given a TM T and a nonhalting state q of T, does T ever enter state q when it begins with a blank tape?

e. Given a TM T, is there a string it accepts in an even number of moves?

j. Given a TM T, does T halt within ten moves on every string?

l. Given a TM T, does T eventually enter every one of its nonhalting states if it begins with a blank tape?