Fundamentele Informatica 3

voorjaar 2014

`http://www.liacs.nl/home/rvvliet/fi3/`

Rudy van Vliet
kamer 124 Snellius, tel. 071-527 5777
rvvliet(at)liacs(dot)nl

college 10, 14 april 2014

9. Undecidable Problems
9.3. More Decision Problems Involving Turing Machines
9.4. Post's Correspondence Problem

A slide from lecture 9:

Definition 9.6. Reducing One Decision Problem to Another, and Reducing One Language 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 9:

Theorem 9.7. Suppose $L_1 \subseteq \Sigma_1^*$, $L_2 \subseteq \Sigma_2^*$, and $L_1 \leq L_2$. If L_2 is recursive, then L_1 is recursive.

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.

Proof. . .

A slide from lecture 9:

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-*$\Lambda \leq$ *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-*$\Lambda \leq P_R \ldots$

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

T_2 highly unspecified. . .

A slide from lecture 9:

Definition 9.6. Reducing One Decision Problem to Another, and Reducing One Language 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.

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 $L(T) = \mathit{NSA}$?

9.4. Post's Correspondence Problem

Instance:

10	01	0	100	1
101	100	10	0	010

Instance:

10	01	0	100	1
101	100	10	0	010

Match:

10	1	01	0	100	100	0	100
101	010	100	10	0	0	10	0

Definition 9.14. Post's Correspondence Problem

An instance of Post's correspondence problem (PCP) is a set

$$\{(\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_n, \beta_n)\}$$

of pairs, where $n \geq 1$ and the α_i's and β_i's are all nonnull strings over an alphabet Σ.

The decision problem is this:

Given an instance of this type, do there exist a positive integer k and a sequence of integers $i_1, i_2, \ldots, i_k$, with each i_j satisfying $1 \leq i_j \leq n$, satisfying

$$\alpha_{i_1}\alpha_{i_2}\ldots\alpha_{i_k} = \beta_{i_1}\beta_{i_2}\ldots\beta_{i_k} \quad ?$$

$i_1, i_2, \ldots, i_k$ need not all be distinct.

Definition 9.14. Post's Correspondence Problem (continued)

An instance of the modified Post's correspondence problem ($MPCP$) looks exactly like an instance of PCP, but now the sequence of integers is required to start with 1. The question can be formulated this way:

Do there exist a positive integer k and a sequence $i_2, i_3, \ldots, i_k$ such that

$$\alpha_1 \alpha_{i_2} \ldots \alpha_{i_k} = \beta_1 \beta_{i_2} \ldots \beta_{i_k} \quad ?$$

(Modified) correspondence system, match.

Theorem 9.15. *MPCP* $\leq$ *PCP*

Proof.

For instance

$$I = \{(\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_n, \beta_n)\}$$

of *MPCP*, construct instance $J = F(I)$ of *PCP*, such that I is yes-instance, if and only if J is yes-instance.

For $1 \leq i \leq n$, if

$$(\alpha_i, \beta_i) = (a_1 a_2 \ldots a_r,\ b_1 b_2 \ldots b_s)$$

we let

$$(\alpha'_i, \beta'_i) = (a_1 \# a_2 \# \ldots a_r \#,\ \# b_1 \# b_2 \ldots \# b_s)$$

For $1 \leq i \leq n$, if

$$(\alpha_i, \beta_i) = (a_1 a_2 \ldots a_r,\ \ b_1 b_2 \ldots b_s)$$

we let

$$(\alpha'_i, \beta'_i) = (a_1 \# a_2 \# \ldots a_r \#,\ \ \# b_1 \# b_2 \ldots \# b_s)$$

If

$$(\alpha_1, \beta_1) = (a_1 a_2 \ldots a_r,\ \ b_1 b_2 \ldots b_s)$$

add

$$(\alpha''_1, \beta''_1) = (\# a_1 \# a_2 \# \ldots a_r \#,\ \ \# b_1 \# b_2 \ldots \# b_s)$$

Finally, add

$$(\alpha'_{n+1}, \beta'_{n+1}) = (\$, \#\$)$$

Theorem 9.16. *Accepts* $\leq$ *MPCP*

The technical details of the proof of this result do not have to be known for the exam. However, one must be able to carry out the construction below.

Proof. . .

For every instance (T, w) of *Accepts*, construct instance $F(T, w)$ of *MPCP*, such that . . .

A slide from lecture 3

Notation:

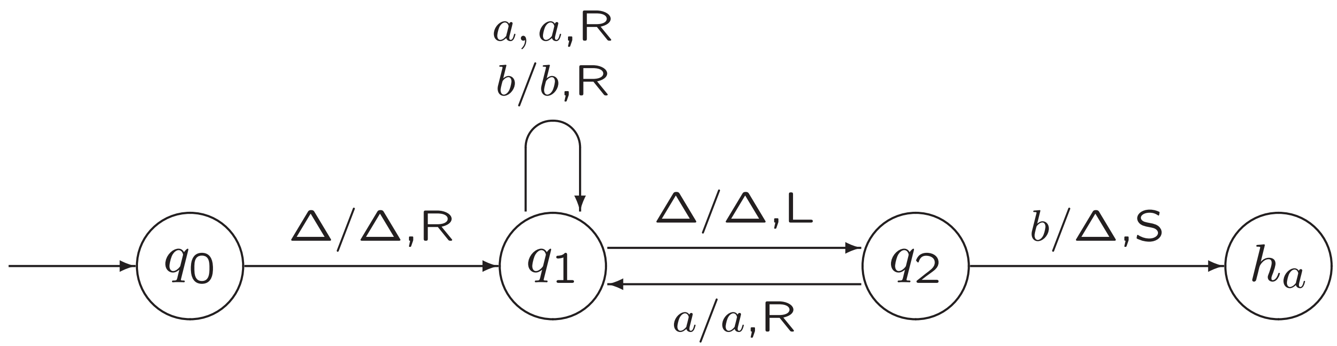
description of tape contents: $x\underline{\sigma}y$ or $x\underline{y}$

configuration $xqy = xqy\Delta = xqy\Delta\Delta$

initial configuration corresponding to input x: $q_0\Delta x$

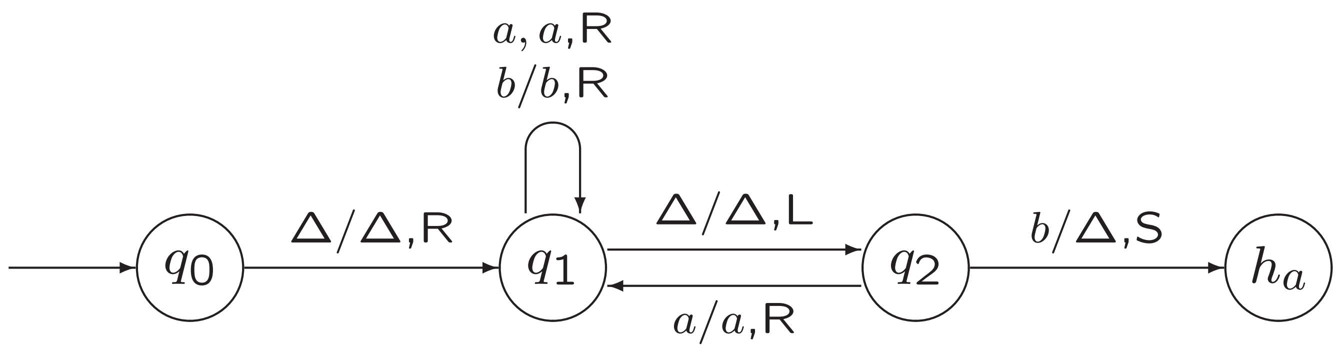
In the third edition of the book, a configuration is denoted as $(q, x\underline{y})$ or $(q, x\underline{\sigma}y)$ instead of xqy or $xq\sigma y$.
This old notation is also allowed for Fundamentele Informatica 3.

Example 9.18. A Modified Correspondence System for a TM



T accepts . . .

Example 9.18. A Modified Correspondence System for a TM



T accepts all strings in $\{a, b\}^*$ ending with b.

Proof of Theorem 9.16. (continued)

Take

$(\alpha_1, \beta_1) = (\#, \# q_0 \Delta w \#)$

Pairs of type 1: (a, a) for every $a \in \Gamma \cup \{\Delta\}$, and $(\#, \#)$

Pairs of type 2: corresponding to moves in T, e.g.,

(qa, bp), if $\delta(q, a) = (p, b, R)$

(cqa, pcb), if $\delta(q, a) = (p, b, L)$

Proof of Theorem 9.16. (continued)

Take

$(\alpha_1, \beta_1) = (\#, \#q_0\Delta w\#)$

Pairs of type 1: (a, a) for every $a \in \Gamma \cup \{\Delta\}$, and $(\#, \#)$

Pairs of type 2: corresponding to moves in T, e.g.,

(qa, bp), if $\delta(q, a) = (p, b, R)$

(cqa, pcb), if $\delta(q, a) = (p, b, L)$

$(q\#, pa\#)$, if $\delta(q, \Delta) = (p, a, S)$

Proof of Theorem 9.16. (continued)

Take

$(\alpha_1, \beta_1) = (\#, \#q_0\Delta w\#)$

Pairs of type 1: (a, a) for every $a \in \Gamma \cup \{\Delta\}$, and $(\#, \#)$

Pairs of type 2: corresponding to moves in T, e.g.,

(qa, bp), if $\delta(q, a) = (p, b, R)$

(cqa, pcb), if $\delta(q, a) = (p, b, L)$

$(q\#, pa\#)$, if $\delta(q, \Delta) = (p, a, S)$

Pairs of type 3: for every $a, b \in \Gamma \cup \{\Delta\}$, the pairs

$(h_a a, h_a), \quad (a h_a, h_a), \quad (a h_a b, h_a)$

One pair of type 4:

$(h_a\#\#, \#)$

Proof of Theorem 9.16. (continued)

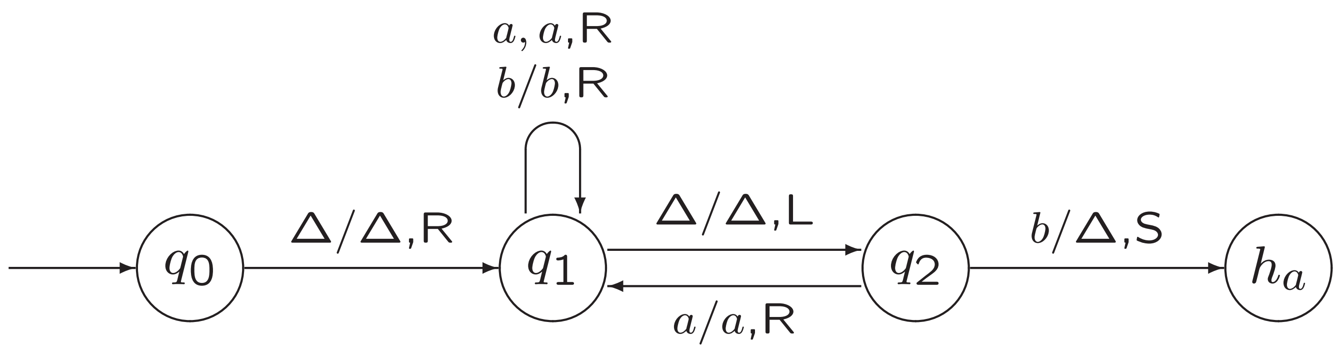
Two assumptions in book:

1. T never moves to h_r
2. $w \neq \Lambda$ (i.e., special initial pair if $w = \Lambda$)

These assumptions are not necessary. . .

Theorem 9.17.
Post's correspondence problem is undecidable.

Example 9.18. A Modified Correspondence System for a TM



T accepts all strings in $\{a, b\}^*$ ending with b.

Pairs of type 2:

$$(q_0\Delta, \Delta q_1) \quad (q_0\#, \Delta q_1\#) \quad (q_1 a, a q_1) \quad (q_1 b, b q_1)$$
$$(a q_1 \Delta, q_2 a \Delta) \quad (b q_1 \Delta, q_2 b \Delta) \quad \ldots$$

Study this example yourself.

Huiswerkopgave 3

Reducties en (on-)beslisbaarheid