

FIT2014 Theory of Computation

Assignment Project Exam Help

Lecture 21

Mapping Reductions

<https://powcoder.com>

slides by Graham Farr

Add WeChat powcoder

COMMONWEALTH OF AUSTRALIA

Copyright Regulations 1969

Warning

This material has been reproduced and communicated to you by or on behalf of Monash University
in accordance with s113P of the Copyright Act 1968 (the Act).

The material in this communication may be subject to copyright under the Act.

Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

Do not remove this notice.

Assignment Project Exam Help

- ▶ Mapping reductions: relating one language to another
- ▶ Definition
- ▶ Properties
- ▶ Examples

<https://powcoder.com>

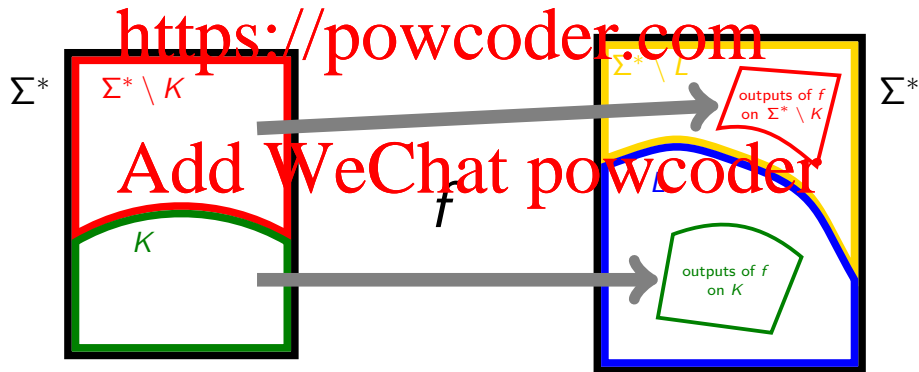
Add WeChat powcoder

Mapping reductions

Definition

A **mapping reduction** from language K to language L is a computable function $f : \Sigma^* \rightarrow \Sigma^*$ such that, for every $x \in \Sigma^*$,

$$x \in K \quad \text{if and only if} \quad f(x) \in L.$$



Mapping reductions

Notation. **Assignment Project Exam Help**
 $K \leq_m L$ means: \exists a mapping reduction from K to L .

<https://powcoder.com>
A very simple property.
Every language is mapping-reducible to itself:

Add WeChat $\forall L. L \leq_m L$ powcoder

Mapping reductions

Theorem

If there is a mapping reduction f from K to L , then:

If L is decidable, then K is decidable.

Symbolically:

$$(K \leq_m L) \wedge (L \text{ is decidable}) \implies (K \text{ is decidable})$$

Proof.

Decider for K :

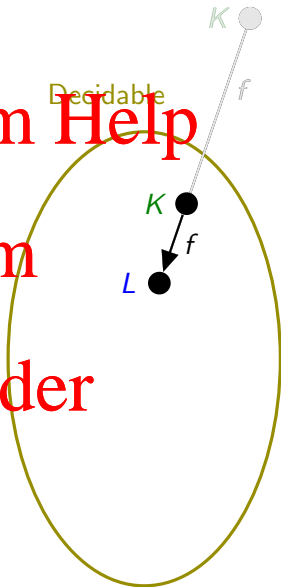
Input: x .

Compute $f(x)$.

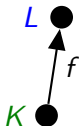
Run the Decider for L on $f(x)$.

// This L -Decider accepts $f(x)$ if and only if $x \in K$,
since f is a mapping reduction from K to L . \square

Decidable



Mapping reductions



Corollary

If there is a mapping reduction f from K to L , then:

If K is undecidable, then L is undecidable.

Symbolically:

$$(K \leq_m L) \wedge (K \text{ is } \underline{\text{undecidable}}) \implies (L \text{ is } \underline{\text{undecidable}})$$

Proof.

Contrapositive of previous Theorem. \square

Decidable

K

L

f

<https://powcoder.com>

Add WeChat powcoder

EQUAL to HALF-AND-HALF

Mapping reduction f :

Input: a word w over alphabet $\{a,b\}$

Sort w

Output the sorted word.

$w \in \text{EQUAL} \iff$ it has the same number of a's as b's

\iff after sorting, it has the same number of a's as b's

(since sorting does not affect letter frequencies)
 $\iff f(w)$ consists of some number of a's followed by

the same number of b's

$\iff f(w) \in \text{HALF-AND-HALF}$

HALF-AND-HALF to PARENTHESES

Mapping reduction:

Input: a word w

For each letter of w in turn:

If previous letter was b and current letter is a

// We have just seen ba which is impossible in HALF-AND-HALF.

Output the string $)$.

else

replace current letter as follows:

$a \rightarrow ($
 $b \rightarrow)$

Output: the string obtained from w by doing all these replacements.

EQUAL to PARENTHESES

Is there a mapping reduction from EQUAL to PARENTHESES?

Yes! Compose the two previous mapping reductions.

This is a special case of:

Theorem.

Mapping reducibility is transitive:

$$K \leq_m L \leq_m M \Rightarrow K \leq_m M.$$

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Mapping reductions: transitivity

Theorem.

Mapping reducibility is transitive:

$$K \leq_m L \leq_m M \implies K \leq_m M.$$

Proof.

Let f be a mapping reduction from K to L , and

let g be a mapping reduction from L to M .

We claim that the composition $g \circ f$, defined for all w by $g \circ f(w) = g(f(w))$, is a mapping reduction from K to M .

Since f and g are both computable, $g \circ f$ must be too.

$$\begin{aligned} w \in K &\iff f(w) \in L && \text{(since } f \text{ is a mapping reduction from } K \text{ to } L) \\ &\iff g(f(w)) \in M && \text{(since } g \text{ is a mapping reduction from } L \text{ to } M) \\ &\iff (g \circ f)(w) \in M && \text{(by definition of } g \circ f). \end{aligned}$$



FA-Empty \longrightarrow No-Digraph-Path

Assignment Project Exam Help

From previous lecture

FA-Empty $:= \{ \langle A \rangle : A \text{ is a FA and } L(A) = \emptyset \}$

Digraph-Path $:= \{ \langle G, s, t \rangle : G \text{ is a directed graph, } s, t \text{ are vertices in } G, \text{ and } \text{there exists a directed } s-t \text{ path in } G. \}$

No-Digraph-Path $:= \{ \langle G, s, t \rangle : G \text{ is a directed graph, } s, t \text{ are vertices in } G, \text{ and } \text{there does not exist a directed } s-t \text{ path in } G. \}$

Add WeChat powcoder

We give a mapping reduction from FA-Empty to No-Digraph-Path.

FA-Empty \longrightarrow No-Digraph-Path

Mapping reduction

Input: $\langle A \rangle$ where A is a Finite Automaton

1. Construct the directed graph G of A :

- ▶ initially, vertices of $G :=$ states of A
- ▶ every transition $v \xrightarrow{x} w$ in A becomes a directed edge (v, w) from v to w in G .
- ▶ then add a new vertex t
- ▶ for every Final State v of A , add a new directed edge (v, t) from v to t in G .

2. Specify s and t :

- ▶ $s :=$ vertex of Start State of A .
- ▶ t is as created above (the new vertex).

3. Output: $\langle G, s, t \rangle$

FA-Empty \longrightarrow No-Digraph-Path

Assignment Project Exam Help

$A \in \text{FA-Empty} \iff$ there is no sequence of transitions in A leading
from Start State to a Final State

\iff there is no path in G leading from s to a vertex
representing a Final State

\iff there is no path in G leading from s to t
 $\iff \langle G, s, t \rangle \in \text{No-Digraph-Path}$

Add WeChat powcoder

RegExpEquiv \longrightarrow FA-Empty

Assignment Project Exam Help

From previous lecture:

RegExpEquiv $\equiv \{ \langle A, B \rangle : A, B \text{ are regular expressions and } L(A) = L(B) \}$
FA-Empty $\equiv \{ \langle A \rangle : A \text{ is a FA and } L(A) = \emptyset \}$

We give a mapping reduction from RegExpEquiv to FA-Empty.

Add WeChat powcoder

RegExpEquiv \longrightarrow FA-Empty

Assignment Project Exam Help

Mapping reduction:

Input: $\langle A, B \rangle$ where A and B are regular expressions

1. Construct a FA C that defines the language

$$(L(A) \cap \overline{L(B)}) \cup (\overline{L(A)} \cap L(B)).$$

2. Output: C

Add WeChat powcoder

Reducing *from* a decidable language

Let's reduce from EnglishPalindromes to YearsOfTransitsOfVenus:

YearsOfTransitsOfVenus $:= \{n : \text{a Transit of Venus occurs in year } n\}$
 $:= \{\dots, 1761, 1769, 1874, 1882, 2004, 2012, 2117, \dots\}$

Mapping reduction:

Input: a string w over the English alphabet

If w is a palindrome

output 2012.

else

output 2021.

<https://powcoder.com>

Add WeChat powcoder

Reducing *from* a decidable language

Theorem.

If L_1 is decidable and L_2 is *any* language except \emptyset and Σ^* then

Assignment Project Exam Help

Proof. Let D be a decider for L_1 .

Let $x^{(\text{yes})}$ be any specific word in L_2 .

Let $x^{(\text{no})}$ be any specific word in $\overline{L_2}$.

Mapping reduction from L_1 to L_2 :

Input: a string w

1. Run D on w .
2. If D accepts w then output $x^{(\text{yes})}$
else output $x^{(\text{no})}$.

<https://powcoder.com>

Add WeChat powcoder

There's not much point
in a mapping reduction
that *decides* L_1 !

Assignment Project Exam Help

Reading: Sipser, pp. 234–238. <https://powcoder.com>

Add WeChat powcoder