

程序代写代做 CS编程辅导

FIT2014 Theory of Computation



Lecture 23

Recursively enumerable languages

WeChat: cstutorcs

Assignment Project Exam Help

slides by Graham Farr

Email: [tutorcs@163.com](mailto:tutorcs@163.com)

QQ: 749389476

COMMONWEALTH OF AUSTRALIA

Copyright Regulations 1969

<https://tutorcs.com>

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.

# Overview

程序代写代做 CS编程辅导



- ▶ recursively enumerable (r.e.) languages
- ▶ relationship with decidability
- ▶ enumerators
- ▶ non-r.e. languages

WeChat: cstutorcs

Assignment Project Exam Help

Email: [tutorcs@163.com](mailto:tutorcs@163.com)

QQ: 749389476

<https://tutorcs.com>

# Decidability

程序代写代做 CS编程辅导

Recall:



A language  $L$  is decidable if and only if there exists a Turing machine  $T$  such that

WeChat: cstutorcs

Accept( $T$ ) =  $L$

Reject( $T$ ) =  $\bar{L}$

Loop( $T$ ) =  $\emptyset$

Email: tutorcs@163.com

Reminder:  $\bar{L} = \Sigma^* \setminus L$ , where  $\Sigma$  is the alphabet.

<https://tutorcs.com>

# Recursively enumerable languages: definition

程序代写代做 CS编程辅导



A language  $L$  is **recursively enumerable** if there exists a Turing machine  $T$  such that

WeChat: cstutorcs  
 $\text{Accept}(T) = L$

Assignment Project Exam Help  
Strings outside  $L$  may be *rejected*, or may make  $T$  *loop forever*.

Email: [tutorcs@163.com](mailto:tutorcs@163.com)

QQ: 749389476

<https://tutorcs.com>

# Recursively enumerable: synonyms

程序代写代做 CS编程辅导

recursively enumerable (r.e.)

= computable

= partially decidable

= Turing recognisable (used in Sipser)

= type 0

Email: [tutorcs@163.com](mailto:tutorcs@163.com)

QQ: 749389476

= *computable*

<https://tutorcs.com>

...but risk of confusion, as “computable” is sometimes used for “decidable”.



WeChat: [tutorcs](https://tutorcs.com)

Assignment Project Essay Help

## Decidable versus r.e.

程序代写代做 CS编程辅导

Every decidable language is recursively enumerable.

Is every recursively enumerable language decidable?



Consider:

WeChat: cstutorcs

$\text{HALT} = \{T : T \text{ halts, if input is } T\}$

Assignment Project Exam Help

This is the language corresponding to the Halting Problem.

Email: tutorcs@163.com

We know it's not decidable.

QQ: 749389476

Is it recursively enumerable?

<https://tutorcs.com>

## Decidable versus r.e.

程序代写代做 CS编程辅导

Let  $M$  be a Turing machine which, as input, a Turing machine  $T$  and

- ▶ simulates what happens when  $T$  is run with *itself* as its input.
- ▶ If  $T$  stops (in any state),  $M$  stops.



Here,  $M$  could be obtained by modifying a UTM.

WeChat: cstutorcs

Assignment Project Exam Help

Accept( $M$ ) = HALT

Email: tutorcs@163.com

Reject( $M$ ) =  $\emptyset$

QQ: 749389476

Loop( $M$ ) =  $\overline{\text{HALT}}$

<https://tutorcs.com>

## Decidable versus r.e.

程序代写代做 CS编程辅导



So HALT is recursively enumerable

So some recursively enumerable languages are not decidable.

WeChat: cstutorcs

Consider the list of undecidable languages given in Lecture 22.

Assignment Project Exam Help

Which ones are recursively enumerable?

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



## Decidable versus r.e.

### Theorem.

A language is decidable if and only if both it and its complement are r.e.

### Proof.

(  $\implies$  )

Let  $L$  be any decidable language.

程序代写代做 CS编程辅导



WeChat: cstutorcs

We have seen that every decidable language is r.e. So  $L$  is r.e.

Now, the complement of a decidable language is also decidable.

(See Lecture 20, comment on closure properties of the class of decidable languages.)

So  $\bar{L}$  is also decidable, and therefore also r.e.

So  $L$  and  $\bar{L}$  are both r.e.

Email: [tutorcs@163.com](mailto:tutorcs@163.com)

QQ: 749389476

<https://tutorcs.com>

## Decidable versus r.e.

(  $\Leftarrow$  )

Let  $L$  be any language such that both  $L$  and  $\bar{L}$  are both r.e.

程序代写代做 CS编程辅导



Since they are each r.e., there exist Turing machines  $M_1$  and  $M_2$  such that

$$\text{Accept}(M_1) = L$$

$$\text{Accept}(M_2) = \bar{L}.$$

WeChat: estutorcs

Assignment Project Exam Help

Note, each of these TMs might *loop forever* for inputs they don't accept.

Email: [tutorcs@163.com](mailto:tutorcs@163.com)

Construct a new Turing machine  $M'$  that simulates *both*  $M_1$  and  $M_2$ :

QQ: 749389476

Input:  $x$

Repeatedly:

<https://tutorcs.com>

Do one step of  $M_1$ . If it **accepts**, then Accept.

Do one step of  $M_2$ . If it **accepts**, then Reject.

## Decidable versus r.e.

程序代写代做 CS编程辅导

$M'$  is a decider:

- ▶ every string belongs to either  $L_1$  or  $L_2$ ,
- ▶ therefore is accepted by either  $M_1$  or  $M_2$ ,
- ▶ therefore will eventually be either accepted or rejected by  $M'$ .



WeChat: cstutorcs

Furthermore,  $M'$  accepts  $x$  if and only if  $M_1$  accepts  $x$ .

Assignment Project Exam Help

So  $M'$  is a decider for  $L$ .

Email: tutorcs@163.com

So  $L$  is decidable.

QQ: 749389476

<https://tutorcs.com>



## A non-r.e. language

Is every language recursively enumerable?

程序代写代做 CS编程辅导

Consider:

$$\overline{\text{HALT}} = \{ \text{ops forever, if input is } T \}$$



Assume  $\overline{\text{HALT}}$  is r.e.

WeChat: cstutorcs

We already know that HALT is r.e.

Assignment Project Exam Help

So, both HALT and its complement are r.e.

Email: tutorcs@163.com

Therefore, by the previous theorem, HALT is decidable.

QQ: 749389476

<https://tutorcs.com>

Contradiction!

Therefore  $\overline{\text{HALT}}$  is not r.e.

# Enumerators

程序代写代做 CS编程辅导

## Definition



An **enumerator** is a Turing machine which outputs a sequence of strings.

This can be a finite or infinite sequence.

WeChat: cstutorcs

If it's infinite, then the enumerator will never halt.

Assignment Project Exam Help

Email: [tutorcs@163.com](mailto:tutorcs@163.com)

It never accepts or rejects; it just keeps outputting strings, one after another.

QQ: 749389476

- ▶ If the sequence is finite, then the enumerator may stop once it has finished outputting. But the state it enters doesn't matter.

<https://tutorcs.com>

# Enumerators

程序代写代做 CS编程辅导

## Definition



A language  $L$  is **enumerated** by an enumerator  $M$  if

$$L = \{\text{all strings in the sequence outputted by } M\}$$

Assignment Project Exam Help

Members of  $L$  may be outputted in any order by  $M$ , and repetition is allowed.

Email: [tutorcs@163.com](mailto:tutorcs@163.com)

## Theorem

A language is recursively enumerable if and only if it is enumerated by some enumerator.

QQ: 749389476

<https://tutorcs.com>

# Enumerators and r.e. languages

## Theorem

程序代写代做 CS编程辅导

A language is recursively enumerable if and only if it is enumerated by some enumerator.



## Proof.

(  $\Leftarrow$  )

Let  $L$  be a language, and let  $M$  be an enumerator for it.

WeChat: cstutorcs

Construct a Turing machine  $M'$  as follows:

Assignment Project Exam Help

Input: a string  $x$

Simulate  $M$ , and for each string  $y$  it generates:

Email: tutorcs@163.com

Test if  $x = y$ . If so, accept; otherwise, continue.

QQ: 749389476

A string  $x$  is accepted by  $M'$  if and only if it is in  $L$ .

<https://tutorcs.com>

So  $\text{Accept}(M') = L$ . So  $L$  is r.e.

# Enumerators and r.e. languages

程序代写代做 CS编程辅导

(  $\implies$  ) Let  $L$  be r.e. Then there is a TM  $M$  such that  $\text{Accept}(M) = L$ . Take all strings, in order:

$\epsilon, a, b, \dots, bb, aaa, aab, aba, \dots$



Simulate the execution of  $M$  on each of these strings, in parallel.

WeChat: cstutorcs

As soon as any of them stops and accepts its string,  
we pause our simulation, output that string, and then resume the simulation.

Email: tutorcs@163.com

**CAREFUL:**

QQ: 749389476

Infinitely many executions to simulate, but we only have finite time!  
How do we schedule all these simulations?

<https://tutorcs.com>



# Enumerators and r.e. languages

程序代写代做 CS编程辅导

Denote the strings by  $x_1, x_2, \dots$



Algorithm:

For each  $k = 1, 2, \dots$

For each  $i = 1, \dots, k$  WeChat: cstutorcs

Simulate the next step of the execution of  $M$  on  $x_i$   
(provided that execution hasn't already stopped). Assignment Project Exam Help

If this makes  $M$  accept, then Email: tutors@163.com

output  $x_i$  and skip  $i$  in all further iterations;

else if this makes  $M$  reject, then QQ: 749389476

output nothing and skip  $i$  in all further iterations. https://tutorcs.com

# Enumerators and r.e. languages

程序代写代做 CS编程辅导



This algorithm can be implemented on a Turing machine.  
Any string accepted by  $M$  will be outputted.  
So this is an enumerator for  $L$ .



WeChat: cstutorcs

Assignment Project Exam Help

This result explains the term “recursively enumerable” (and “computably enumerable”).

Email: [tutorcs@163.com](mailto:tutorcs@163.com)

It also explains why r.e. languages are sometimes called *computable*, since there is a computer that can compute all its members (i.e., can generate them all).

QQ: 749389476


<https://tutorcs.com>

# Exercises

## Theorem.

程序代写代做 CS编程辅导

A language  $L$  is r.e. if and only if  
there is a decidable two-argument predicate  $P$  such that


$$x \in L \iff \exists y : P(x, y).$$

WeChat: cstutorcs

This  $P$  is a *verifier*:

Assignment Project Exam Help

if you are given  $y$  then you can use  $P$  to *verify* that  $x$  is in  $L$  (if it is).

Email: [tutorcs@163.com](mailto:tutorcs@163.com)

But it may be hard to *find* such a  $y$ .

QQ: 749389476

## Theorem.

<https://tutorcs.com>

If  $K \leq_m L$  and  $L$  is r.e. then  $K$  is r.e.

# Recursively enumerable languages

程序代写代做 CS编程辅导

r.e.

$\overline{\text{HALT}}$



HALT

WeChat: cstutorcs

Decidable

Assignment Project Exam Help

Email: [tutorcs@163.com](mailto:tutorcs@163.com)

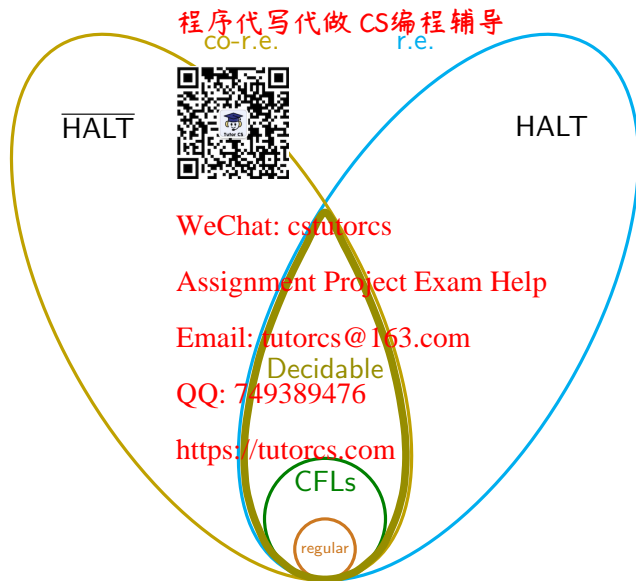
QQ: 749389476

<https://tutorcs.com>

CFLs

regular

# Recursively enumerable languages



# Revision

程序代写代做 CS编程辅导



- ▶ definition of recursively enumerable languages
- ▶ relationship between decidability and recursive enumerability
- ▶ enumerators and their relationship with r.e. languages
- ▶ a language that is r.e. but not decidable, with proof
- ▶ a language that is not r.e., with proof

WeChat: cstutorcs

Assignment Project Exam Help

Email: [tutorcs@163.com](mailto:tutorcs@163.com)

Reading: Sipser, pp. 170, 209–211.

QQ: 749389476

Preparation: Sipser, pp. 275–286.

<https://tutorcs.com>