CSE 105
THEORY OF COMPUTATION

Spring 2018

http://cseweb.ucsd.edu/classes/sp18/cse105-ab/

Today's learning goals

Sipser Ch 5.1, 5.3

- Define and explain core examples of computational problems, include A_{**}, E_{**}, EQ_{**}, $HALT_{TM}$ (for ** either DFA or TM)
- Explain what it means for one problem to reduce to another
- Define computable functions, and use them to give mapping reductions between computational problems.

Mapping reduction

Sipser p. 235, 237

Problem A is **mapping reducible** to problem B means there is a computable function f: $\Sigma^* \rightarrow \Sigma^*$ such that for all strings x in Σ^*

x is in A iff f(x) is in B

Warm up: Prove that if A mapping reduces to B then the complement of A mapping reduces to the complement of B.

Mapping reduction

Sipser p. 235, 237

Problem A is **mapping reducible** to problem B means there is a computable function f: $\Sigma^* \rightarrow \Sigma^*$ such that for all strings x in Σ^*

x is in A iff f(x) is in B

Theorems 5.22, 5.28: If A is mapping reducible to B…

… and B is decidable, then A is decidable.

… and A is undecidable, then B is undecidable.

… and B is recognizable, then A is recognizable.

… and A is unrecognizable, then B is unrecognizable.

Decidable	Undecidable but recognizable	Undecidable and unrecognizable
A_{DFA}	A_{TM}	A_{TM}^{C}
E_{DFA}	$HALT_{TM}$	$HALT_{TM}^{C}$
EQ_{DFA}		

Give algorithm!

Know A_{TM} mapping reduces to $HALT_{TM}$ and vice versa.

What about E_{TM} and EQ_{TM} ?

E_{TM} = { <M> | M is TM, L(M) is empty}

A. Decidable
B. Undecidable, recognizable, with recognizable complement
C. Undecidable, recognizable, with unrecognizable complement
D. Undecidable, unrecognizable, with recognizable complement
E. Undecidable, unrecognizable, with unrecognizable complement

Give an example of a string in E_{TM} , and a string not in E_{TM}

Claim: A_{TM} is no harder than E_{TM}^{C}

In other words: we want to mapping reduce A_{TM} to E_{TM}^{C}

Define computable $F : \Sigma^* \rightarrow \Sigma^*$

Input string	Output string

Claim: A_{TM} is no harder than E_{TM}^{C}

In other words: we want to mapping reduce A_{TM} to E_{TM}^{C}

Define computable $F : \Sigma^* \rightarrow \Sigma^*$

Input string	Output string
<M,w> where M accepts w	<M'> where L(M') nonempty
<M,w> where M does not accept w	<M'> where L(M') empty
x not encoding any <M,w>	$const_{out}$

Claim: A_{TM} is no harder than E_{TM}^{C}

In other words: we want to mapping reduce A_{TM} to E_{TM}^{C}

Define computable $F : \Sigma^* \rightarrow \Sigma^*$

F = "On input x:

1. Type-check whether x= <M,w> for some TM M, and string w. If not, output $const_{out}$.
2. Construct the following machine M'

3. Output <M'>"

Claim: A_{TM} is no harder than E_{TM}^{C}

In other words: we want to mapping reduce A_{TM} to E_{TM}^{C}

Define computable $F : \Sigma^* \rightarrow \Sigma^*$

Input string	Output string
<M,w> where M accepts w	<M'> where L(M') = {w}
<M,w> where M does not accept w	<M'> where L(M') = {}
x not encoding any <M,w>	$const_{out}$

Thus, x is in A_{TM} iff F(x) is in E_{TM}^{C}.

Since A_{TM} is undecidable, so is E_{TM}^{C} and hence also E_{TM}.

$EQ_{TM} = \{ \langle M_1, M_2 \rangle \mid M_1, M_2 \text{ TMs}, L(M_1)=L(M_2)\}$

A. Decidable
B. Undecidable, recognizable, with recognizable complement
C. Undecidable, recognizable, with unrecognizable complement
D. Undecidable, unrecognizable, with recognizable complement
E. Undecidable, unrecognizable, with unrecognizable complement

Give an example of a string in EQ_{TM} , and a string not in EQ_{TM}

Claim: $HALT_{TM}$ is no harder than EQ_{TM}

In other words: we want to mapping reduce $HALT_{TM}$ to EQ_{TM}

Define computable $F : \Sigma^* \rightarrow \Sigma^*$

Input string	Output string

Claim: $HALT_{TM}$ is no harder than EQ_{TM}

In other words: we want to mapping reduce $HALT_{TM}$ to EQ_{TM}

Define computable $F : \Sigma^* \rightarrow \Sigma^*$

Input string	Output string
<M,w> where M halts on w	$<M_1,M_2>$ where $L(M_1)=L(M_2)$
<M,w> where M loops on w	$<M_1,M_2>$ where $L(M_1)\neq L(M_2)$
x not encoding any <M,w>	$const_{out}$

Claim: $HALT_{TM}$ is no harder than EQ_{TM}

In other words: we want to mapping reduce $HALT_{TM}$ to EQ_{TM}

Define computable $F : \Sigma^* \rightarrow \Sigma^*$

F = "On input x:

1. Type-check whether x= <M,w> for some TM M, and string w. If not, output $const_{out}$.
2. Construct the following machine M_1

3. Output $<M_1, M_2>$"

Claim: $HALT_{TM}$ is no harder than EQ_{TM}

In other words: we want to mapping reduce $HALT_{TM}$ to EQ_{TM}

Define computable $F : \Sigma^* \rightarrow \Sigma^*$

Input string	Output string
<M,w> where M halts on w	$<M_1, M_2>$ where $L(M_1) = \Sigma^* = L(M_2)$
<M,w> where M loops on w	$<M_1, M_2>$ with $L(M_1) = \Sigma^*$, $L(M_2) = \Sigma^* - \{w\}$
x not encoding any <M,w>	$const_{out}$

Thus, x is in $HALT_{TM}$ iff F(x) is in EQ_{TM}.

Since $HALT_{TM}$ is undecidable, so is EQ_{TM}

EQ_{TM}

- $HALT_{TM}$ mapping reduces to EQ_{TM}
- $HALT_{TM}$ is undecidable
- $HALT_{TM}$ is recognizable, so $HALT_{TM}^{C}$ is unrecognizable

- Therefore: EQ_{TM} is undecidable and its complement is unrecognizable.

Is EQ_{TM} itself recognizable?

Prove $HALT_{TM}^{C}$ mapping reduces to EQ_{TM}

Claim: $HALT_{TM}^{C}$ is no harder than EQ_{TM}

In other words: we want to mapping reduce $HALT_{TM}^{C}$ to EQ_{TM}

Define computable $F : \Sigma^* \rightarrow \Sigma^*$

Input string	Output string
<M,w> where M loops on w	$<M_1,M_2>$ where $L(M_1)=L(M_2)=\{w\}$
<M,w> where M halts on w	$<M_1,M_2>$ where $L(M_1)=\{w\}\neq L(M_2)=\{\}$
x not encoding any <M,w>	$Const'_{out}$

Next time

Pre-class reading.