Find a partner for this game.
Player A: Writes down a computation, one configuration at a time for some TM which has K={s,t} and Σ={#,a} starting with (s, #aaa[#]). Keep your TM secret from player B.
Player B: When you know the answer to this question for player A's TM, tell them to stop and tell them the answer.
Question: Does Player A's TM ever move its head to the left when started with (s, #aaa[#])?

Player A gets one point for each configuration written down before player B knows the answer. If Player B says stop and claims an answer that Player A can contradict with a counterexample, keep playing.
How many points can Player A get? Switch roles and try some other cases with more states and/or more symbols in the alphabet.

Recall that TM's are deterministic so if the TM is in a state q and has the symbol σ on the current tape square and it transfers to a state r with a given head instruction, then it must do the same thing every time the machine has current state q with σ on the current tape square.

How does Player A's score depend on the number of states and number of symbols?

Computer Science Lab Evaluations

Please login to connex to evaluate your tutorial instructor this week.

Due to database corruption problems, anyone who completed a lab evaluation prior to November 21, 2011 will have to re-do that lab evaluation.

Each lab evaluation response is anonymous. All information and comments will be given to selected Computer Science personnel only after the final grades have been sent to Records Services.

Assignment #5: Available from class web page. Due Friday Dec. 2 at the beginning of class.

Final exam tutorial:
Sunday Dec. 4 at 12:00pm in ECS 116.

There will be tutorials next week (the last week of classes).

Proving Problems Undecidable

To prove problem Q is not decidable:

1. Select a problem P known to not be decidable.

2. Prove that if you can solve Q you can solve P.

3. This is a contradiction since P is not decidable and therefore Q is not decidable.

M is a TM with start state s defined over the alphabet {#,(,),0,1,a,q,,}:

State	Sym.	Next State	Head
s	#	s	L
s	)	t	L
t	0	h	R
t	1	t	1

Num	State	Head
0	h	L
1	s	R
2	t	#
3		(
4		)
5		0
6		1
7		a
8		q
9		,

1. What is "M"? Use the table given to number the states and head instructions.

2. Does M halt on input "M"?

Let H_1 = {"M": M halts on input "M"}.

A string "M":

(q01,a0010,q01,a0000),(q01,a0100,q10,a0000),
(q10,a0101,q00,a0001),(q10,a0110,q10,a0110) $\in H_1$

State	Sym.	Next State	Head
s	#	s	L
s	)	t	L
t	0	h	R
t	1	t	1

Let H_1 = {"M": M halts on input "M"}.

Change one transition so it does not halt:

(q01,a0010,q01,a0000),(q01,a0100,q10,a0000),
(q10,a0101,q10,a0101),(q10,a0110,q10,a0110) $\notin H_1$

State	Sym.	Next State	Head
s	#	s	L
s	)	t	L
t	0	t	0
t	1	t	1

q00a())01aaq01 $\notin H_1$

$\varepsilon \notin H_1$

H_1 = {"M": M halts on input "M"}

The complement of H_1 = L = $\overline{H_1}$

= {"M" : M does not halt on input "M"}

U { w: w ≠ "M" for any TM M}

Theorem: L is not Turing-acceptable.

Proof: Assume that TM M′ accepts L.

What does TM M′ do on input "M′"?

We are assuming M′ accepts L.

L = {“M” : M does not halt on input “M”}

U { w: w ≠ “M” for any TM M}

Case 1: M′ halts on “M′”.

But then “M′” is not in L so M′ should not halt- contradiction.

Case 2: M′ does not halt on “M′”.
But then “M′” is in L so M′ should halt- contradiction.

H_1 = {"M": M halts on input "M"}

L = {"M" : M does not halt on input "M"}

$\cup$ { w: w ≠ "M" for any TM M}

Implications:

1. L is not Turing-acceptable.
2. L is not Turing-decidable.
3. H_1 is not Turing-decidable.

Suppose I construct a TM M_f which:

1. Preserves its input u.

2. Simulates a machine M_b on input ε.

3. If M_b hangs on input ε, force an infinite loop.

4. If M_b halts on input ε, then run a UTM on the original input u which forces an infinite loop if u is not "M" for any TM M, and otherwise it runs M (u= "M") on input ε. If M hangs on input ε, the simulating UTM also hangs.

What language does this machine M_f accept in each of two cases:

Case 1: When machine M_b does not halt on input ε.

Case 2: When machine M_b halts on input ε.

Assignment #5: Available from class web page. Due Friday Dec. 2 at the beginning of class.

Final exam tutorial:
Sunday Dec. 4 at 12:00pm in ECS 116.

There will be tutorials this week.

Known: H_1 = {"M": M halts on input "M"} is not Turing decidable.

Theorem 5.4.2 (p. 255): The following problems are not Turing-decidable:

(a) Given M, w: Does M halt on input w?

(b) Given M: Does M halt on input ε?

(c) Given M: Is there any string on which M halts?

(d) Given M: Does M halt on every string?

(e) Given two TM's M_1 and M_2: Do they halt on the same input strings?

The following languages are Turing-decidable:

(a) {“M” “w” : M halts on w after at most 700 steps}

(b) {“M” “w”: M halts on w without using more than the first 100 tape squares}

For part (b): we only have to simulate M for a finite number of steps and in that time frame it will either halt, hang, use more than the first 100 tape squares, or it will never halt. How many steps must we run M for?