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"?

Announcements

Assignment #5 has been posted.
Due Friday Aug. 3 at the beginning of class.
We will have a tutorial next week.

Final exam tutorial:

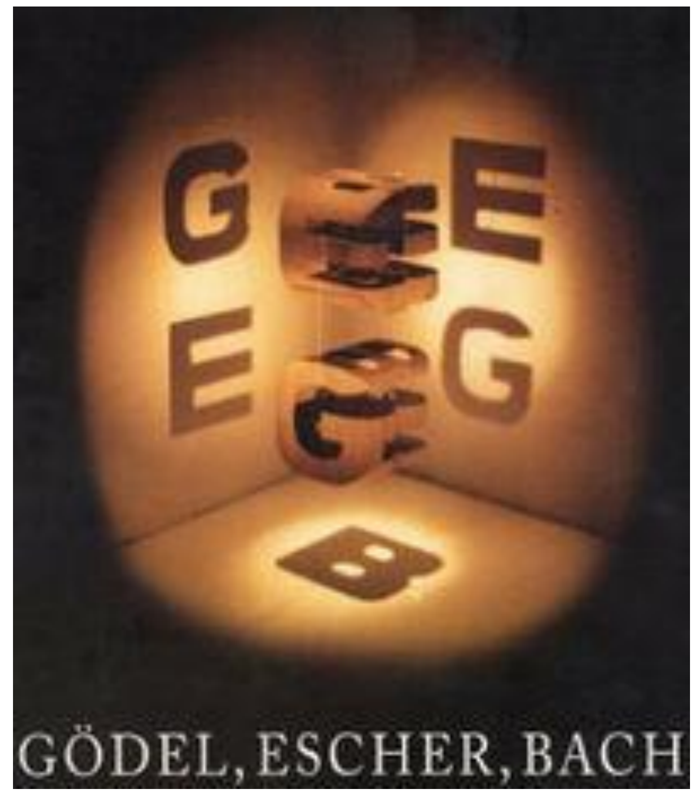
Monday Aug. 6, 10am, ECS 116.

If the building is locked, I will prop open the back door to ECS (the one that opens on to the campus).

Old finals are available from the course web page.

Gödel, Escher, Bach: an Eternal Golden Braid is a Pulitzer Prize-winning book by Douglas Hofstadter, described by the author as "a metaphorical fugue on minds and machines in the spirit of Lewis Carroll". On its surface, it examines logician Kurt Gödel, artist M. C. Escher and composer Johann Sebastian Bach, discussing common themes in their work and lives. At a deeper level, the book is a detailed and subtle exposition of concepts fundamental to mathematics, symmetry, and intelligence.

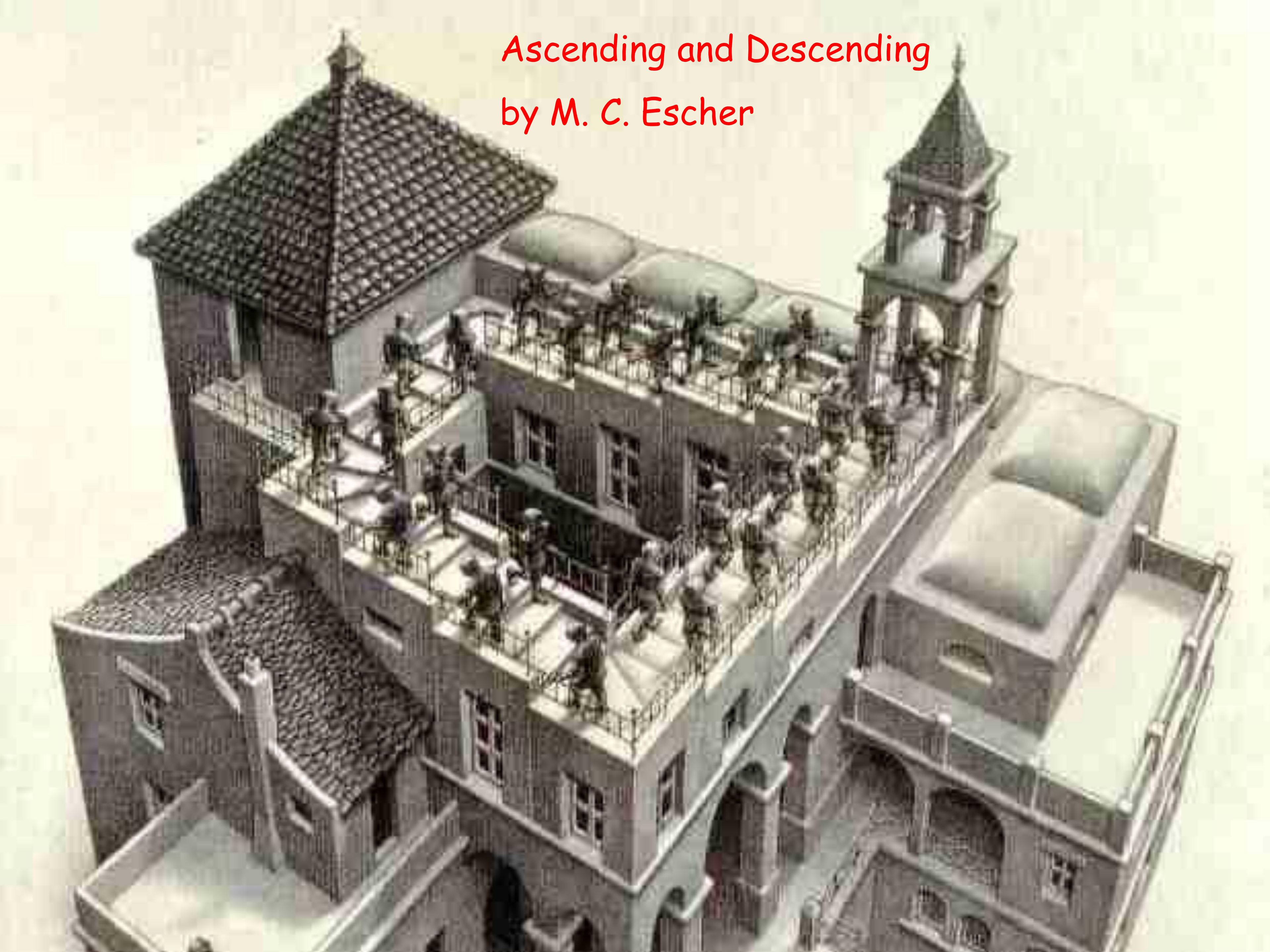
Through illustration and analysis, the book discusses how self-reference and formal rules allow systems to acquire meaning despite being made of "meaningless" elements. It also discusses what it means to communicate, how knowledge can be represented and stored, the methods and limitations of symbolic representation, and even the fundamental notion of "meaning" itself.

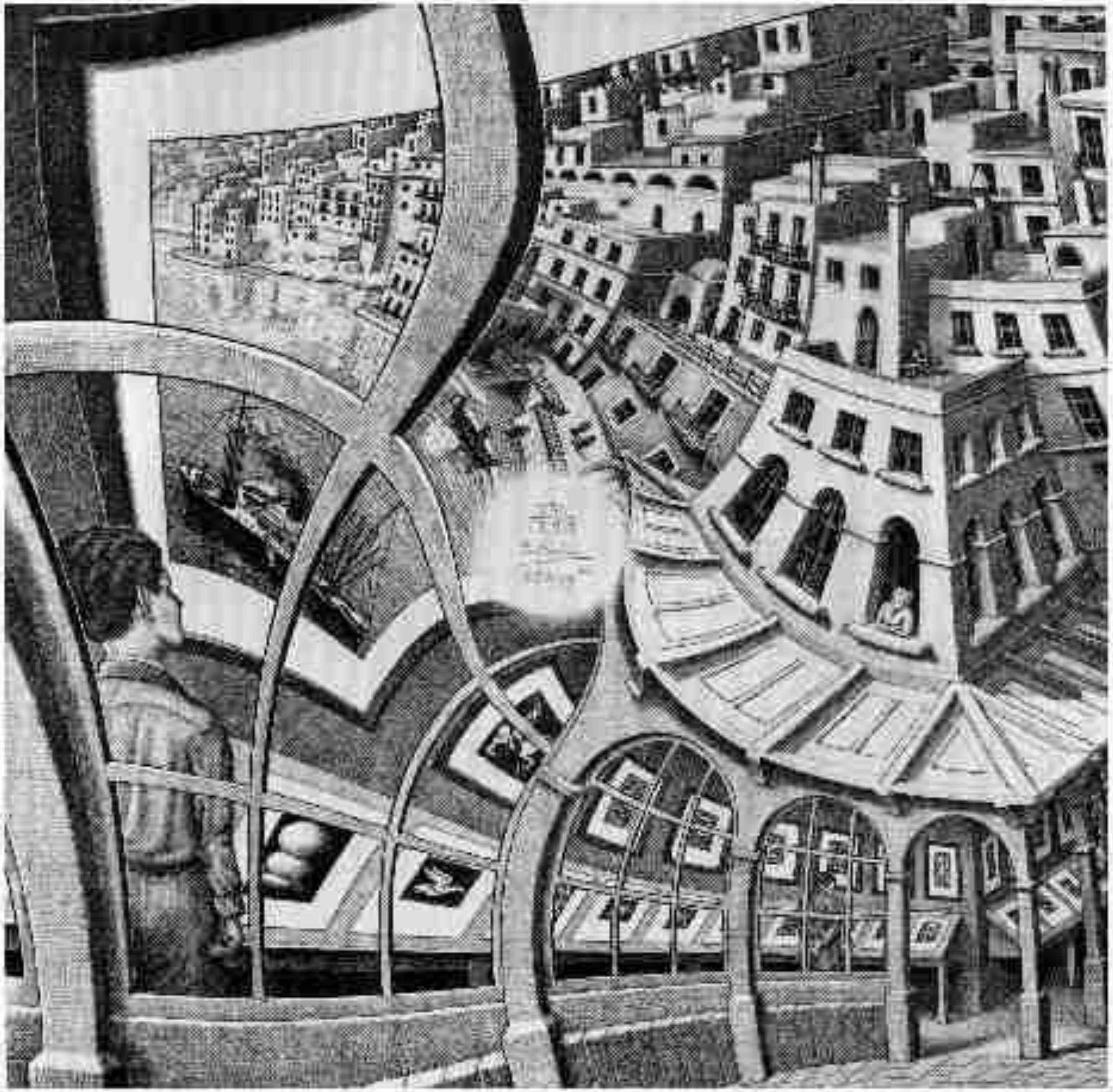


From Wikipedia, the free encyclopedia.

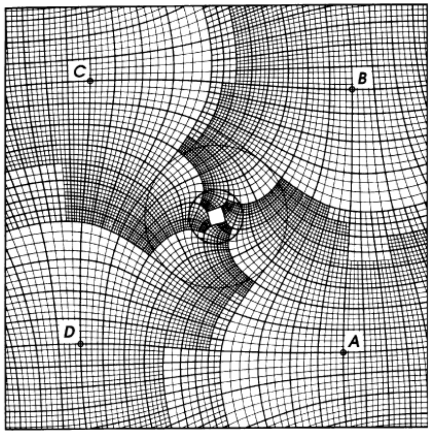
Ascending and Descending

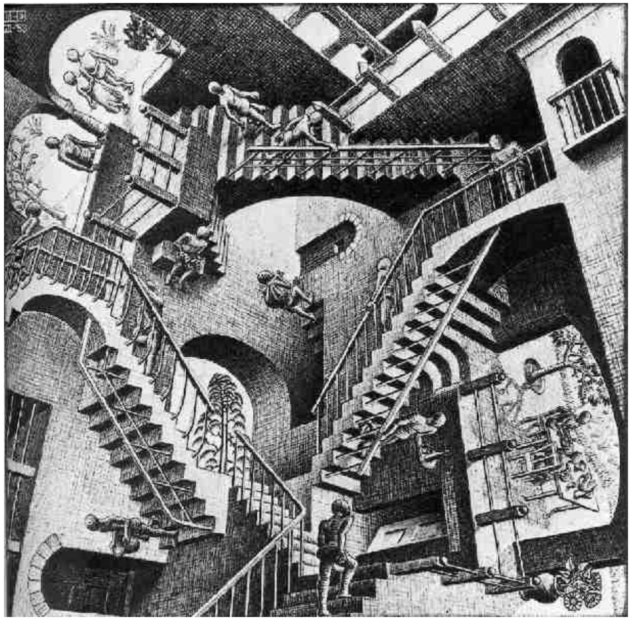
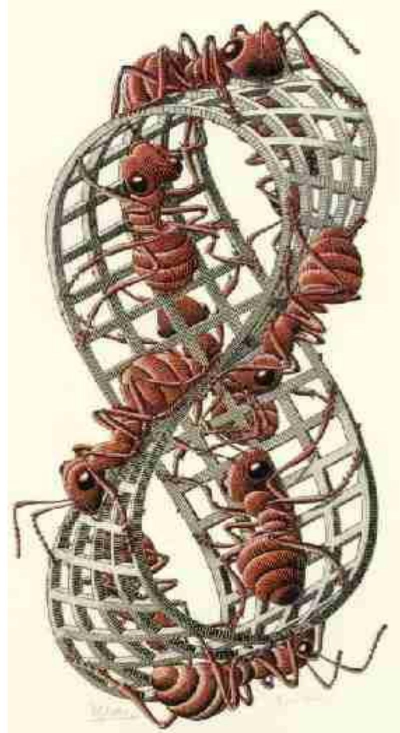
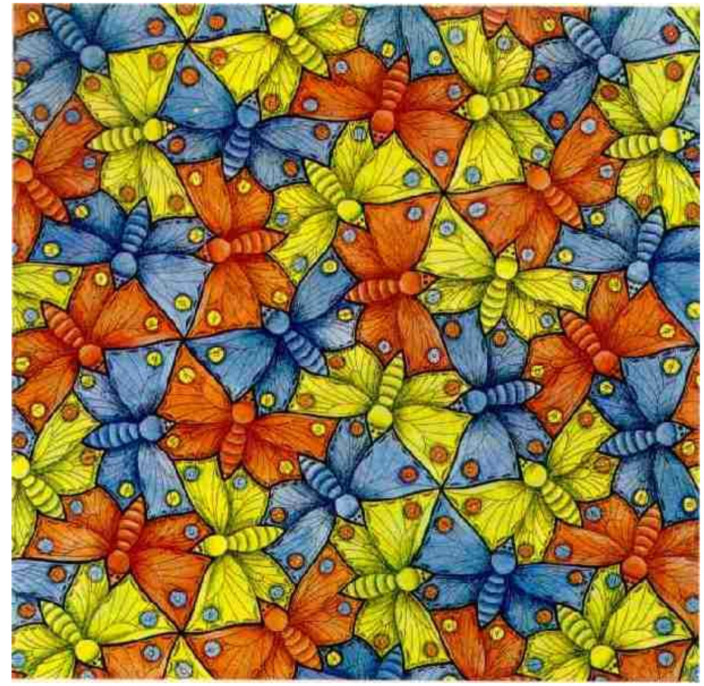
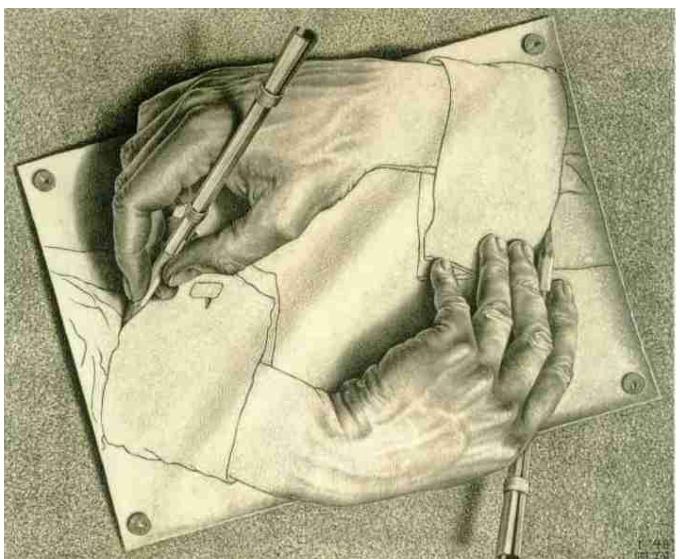
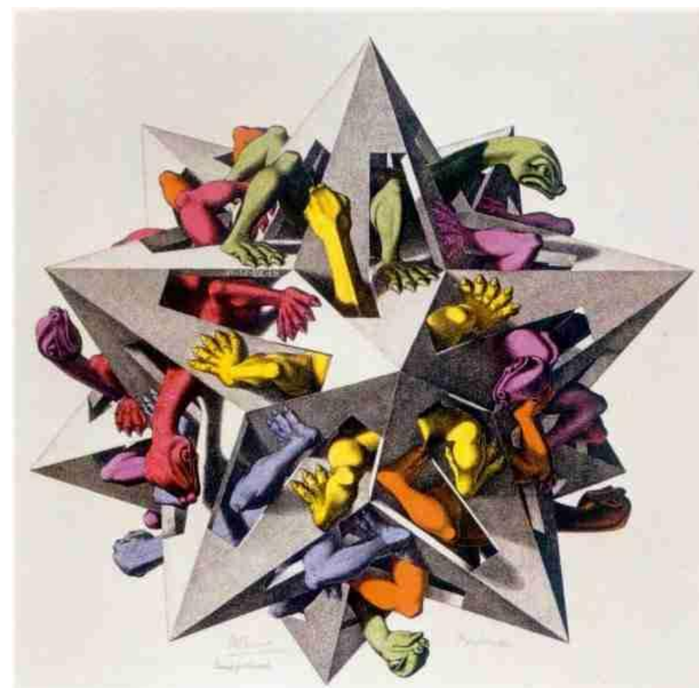
by M. C. Escher





Print Gallery by M. C. Escher



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.

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"}

∪ { 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"}

U { 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.

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?

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.