Turing Machine variants

We extend the hardware of our Turing Machine..

\+ 2-way infinite tape (M±)
++ more than one tape (M_2, M_3..)
+++ 2-dimensional tape

and explore Church's thesis..

..if it is true, the enhanced TMs should be no more powerful than the ordinary one..

?can the same problems be solved?.. speed, space irrelevant)

..if we can demonstrate that an ordinary TM can solve any problem that a +, ++ or +++ TM can solve

→ support for Church's thesis that a TM is general enough to represent all algorithms

Equivalence of Turing Machines

Suppose M_1, M_2 are Turing Machines, possibly different kinds of TM, with the same input alphabet:

M_1 and **M_2** are **equivalent** if $\mathbf{f_{M\ 1} = f_{M\ 2}}$

that is, M_1 and M_2 compute the same function…they have the same input-output function.

We show that 2 kinds of TM are equivalent by showing that for any machine of one kind there is an equivalent machine of the other kind (and vice versa).

..that is, given the same input they compute the same output
or both Halt & Fail
or both run forever

..proving equivalence :

Given **M** a simple TM

M^+ a variant with extra features (..more expensive..?)

any function computable by M

- show there is an M+ with $f_{M+} = f_M$
- usually straightforward..

typically M's features are a subset of M+

any function computable by M+

- show there is an M with $f_M = f_{M+}$
- not so easy
- done by simulation..

M mimics Machine M^+

-copies what M^+ does at detailed level

For big differences in complexity..need to show in stages..M,M+,M++

M± a TM with 2-way infinite tape…

input word on squares 0, 1, 2.. output stored onto squares 0, 1, 2..

prove the equivalence of M and M±

a) …for each M show there is an M± such that $f_{M\pm} = f_M$

for each M = { Q, ∑, I, q_0, □,F}

define M± = {Q', ∑□ {**fail**}, I, q_0, □',F}

□': the first action of M± : move left

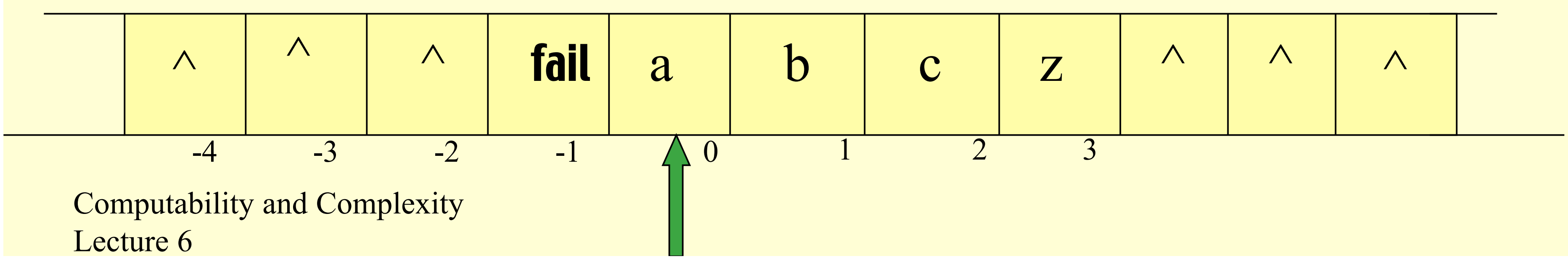
write **fail**

move right

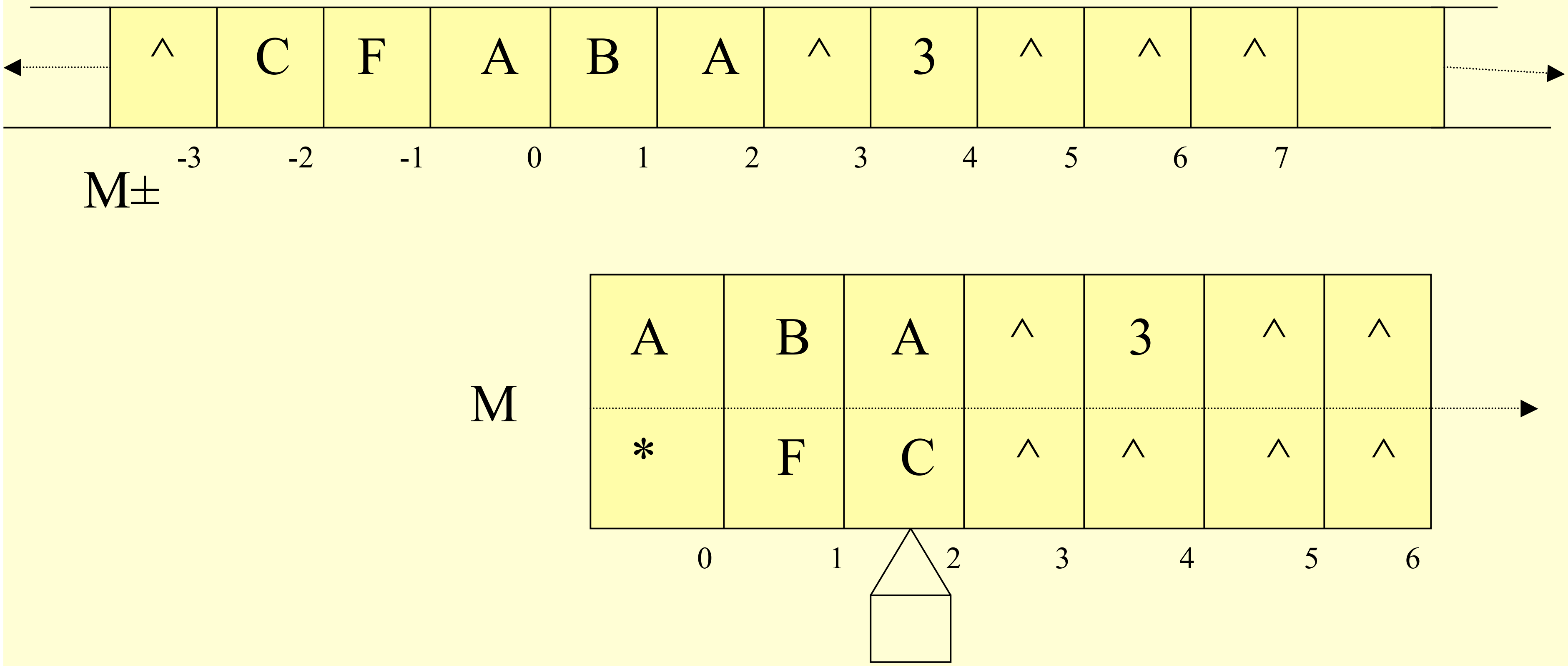
add instructions to Halt and Fail if M+ reads **fail** at any time.

otherwise all instructions are as for M.

$$f_{M\pm} = f_M$$



b) for each M± show there is an M (ordinary TM) s. t. $f_M = f_{M\pm}$



The 2-way infinite tape of M± is simulated by 2 tracks on M's tape, square 0 marked by * in track2.

It uses a variable `track` in the states to remember whether it is using track 1 of M± (right of sq 0, `track` = 1) or track 2 (left of sq 0, `track` = -1)

M and M±

M mimics M±. The output of M± is built up in track 1 of M's tape, starting in Square 0.

When M± reaches a halting state, M restores its tape to a single track before Halt & Succeed.

If M± has no applicable instruction, neither does M, it will Halt & Fail

If M± does not halt, neither will M.

We conclude that M and M± are equivalent

Multi-tape Turing machines - with a read head for each tape.
The heads move independently. Each reads a symbol from its tape
The δ-function determines the action to be taken.

e.g for a 3-tape TM this depends on the
current state and the 3 symbols read:
write 3 symbols, 3 head moves, next state.

Formal definition:

M = (Q, ∑, I, q_0, δ, F), where

δ: Qx∑x∑x∑ $\rightarrow$ Qx ∑x∑x∑x{-1,0,1}x{-1,0,1}x{-1,0,1}

In general: **δ: Qx$\sum^n$ $\rightarrow$ Qx $\sum^n$x$\{-1,0,1\}^n$** for an n-tape TM

Multi-tape Turing machines..continued

- if n=1 this is an ordinary 1-tape Turing Machine
- the number of tapes is determinable only from the δ-function
- there is one state set, Q, one input alphabet, I and one full alphabet,∑
- input and output: always on tape 1 from square 0, up to the first ^.
- state diagram notation: we need to accommodate all the tapes:

eg for 2 tapes the label on the transition from one state to another

((a, b), (a', b'), (d, d'))

symbol on tape 1 **symbol on tape 2** **write to tape 1** **write to tape 2** **move head 1** **move head 2**

n-Tape TMs are equivalent to 1-tape TMs..

a) Show that for any 1-tape TM there is an equivalent n-tape TM:

Given a 1-tape TM, $M = (Q, \Sigma, I, q_0, \delta, F)$

let $M_n = (Q, \Sigma, I, q_0, \delta', F)$ where

$\delta': Q \times \Sigma^n \rightarrow Q \times \Sigma^n \times \{-1,0,1\}^n$

$\delta'(q, a_1, a_2, \ldots a_n) = (q', b_1, \wedge, \wedge, \ldots \wedge, d_1, 0, 0 \ldots, 0)$

Where $\delta(q, a_1) = (q', b_1, d_1)$

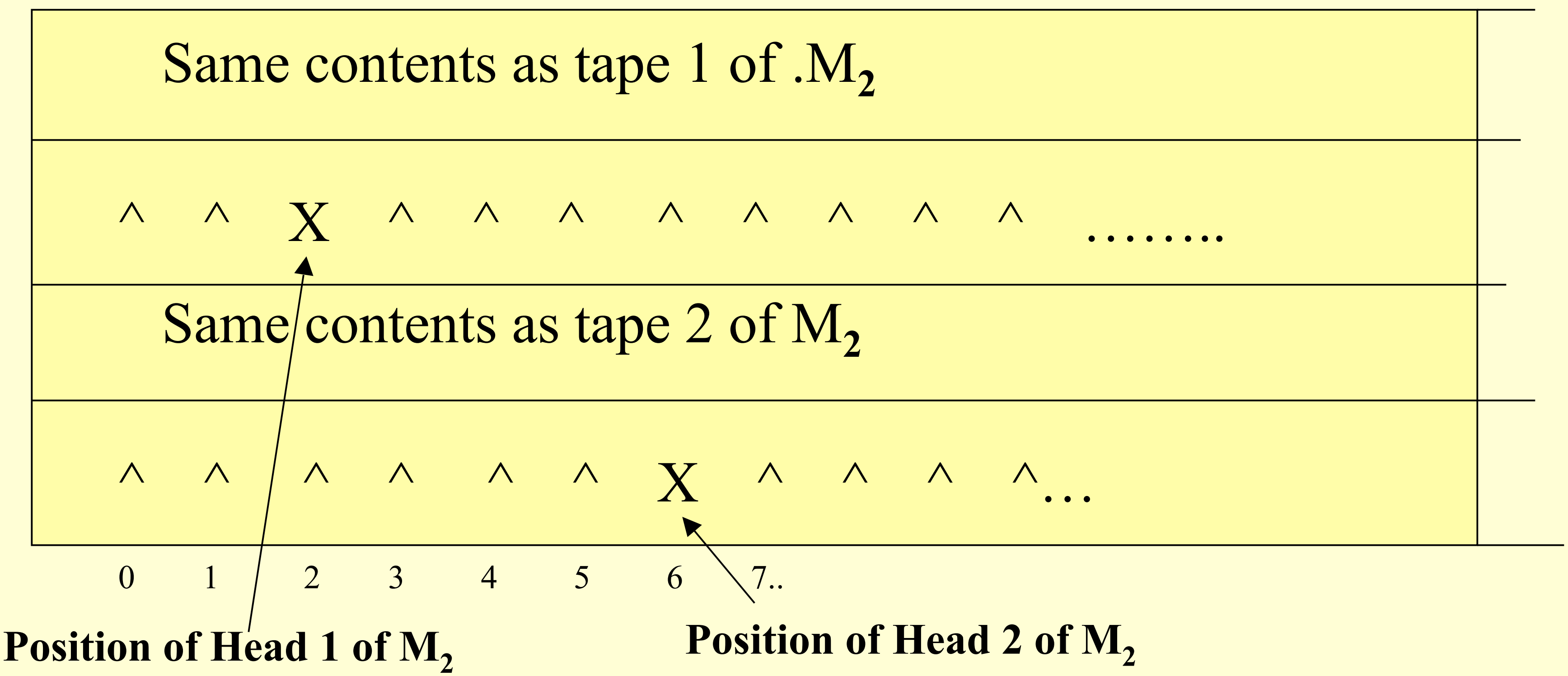
M_n computes the same function as M:
it uses only tape 1, in exactly the same way as M uses its tape, producing the same output for the same input.

b) (for n=2) there is a 1-tape TM equivalent to any 2-tape TM

Given M_2 - a 2-tape TM..construct a 1-tape simulation of M_2 which has the same input-output function.

M has 1 tape with 4 tracks:

M updates its tape to follow the actions of M_2



The TM M, equivalent to M_2..

• Initialise M: start in q_0. Set up square 0: $\delta(q_0,a) = (q',(a,x,\wedge,x),1)$
(assume square 0 marked..as if 5th track)
we use dynamic track setup to set up 4 tracks to the end of the input:
$\delta(q',a') = (q',(a', \wedge,\wedge,\wedge), 1)$ if $a' \neq \wedge$. Return M's head to square 0.

• Operation of M: we know current state q;
Find inputs by moving head right from squ 0 looking for 'x' in track 2
when found, current symbol from M_2's tape 1 is in track1
when 'x' is found in track 4, current symbol of M tape 2 is in track3.
these 2 symbols can be remembered in the state, say y and z.
Look for $\delta(q,y,z) = (q'', b_1, b_2, d_1, d_2)$
if no instruction $\delta(q,y,z)$ then Halt & Fail
otherwise M's head updates tracks 1,2 for M's tape 1
3,4 for M's tape 2.
M's head returns to sq 0; now in state q''.

Operation of M..continued

If q" is a halting state of M_2 then M

restores its tape to a single track

H&S with output the same as tape 1 of M_2

otherwise M continues to simulate M_2 until it reaches a halting state

or Halts and Fails because: no applicable instruction

or tries to move left from square 0

or if M_2 does not halt, neither does M.

a TM M++ with 2-dimensional tape

(0,3)							
	a	b	^	a	^	^	^
(0,1)	1	1	1	0	a	^	^
	^	1	^	a	1	^	^

(0,0) (1,0) (2,0) (3,0)…

δ: Q x ∑ $\to$ Q x ∑ x {L, R, U, D, 0}.

Input in squares: (0,0), (1,0), (2,0), (3,0), (4,0),…(k,0), k≥ 0;

all other squares ^.

Output also along the "x-axis". Rest of tape is workspace

≡ a 1-tape TM with an unbounded number of tracks

move≤ 1 square right, left, up, down each instruction.

..show equivalence to a 2-tape ordinary TM

(□ equivalence to a 1-tape ordinary TM)

Tape 1:

^ *	1	^	a	1	*	1	1	1	0	a	*	a	b	^	a	^	*	

'rows' of the 2-dim. Tape

- filled with blanks to length of the longest
- separated by * and terminated by **.

Tape 2: used for scratch work, especially to keep track of where in the 2-dim tape the M++ head is.

Summary - enhanced hardware TMs

with: 2-way tape
multiple tapes
2-dimensional tape
all these are equivalent to ordinary TMs..

i.e. they cannot calculate anything which cannot be done with an ordinary TM
...they are not more powerful: cannot calculate anything new

This supports the Church-Turing thesis:

that the Turing Machine is a valid formalism for algorithms:

any algorithm it can be implemented by a TM