TRANSITIONS

$$\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R\}$$

THUS $\delta(q, X) = (p, Y, L)$ MEANS THAT IF IN STATE q AND TAPE HEAD IS SCANNING THE SYMBOL X, THEN MOVE TO STATE p, REPLACE X BY Y ON TAPE CELL, AND MOVE TAPE HEAD 1 CELL LEFT

DETERMINISTIC TM (DTM), ABOVE DEFINES A DTM — FOR EACH $\delta(q, X)$ WE HAVE AT MOST ONE POSSIBLE MOVE — ALTHOUGH $\delta(q, X)$ COULD BE UNDEFINED

ACCEPTANCE? IF WHEN DTM IS STARTED WITH w ON TAPE IT EVENTUALLY ENTERS A FINAL STATE

THUS WE MAY AS WELL ASSUME THAT ALL FINAL/ACCEPTING STATES ARE "HALTING STATES" — IN THAT NO TRANSITIONS ARE DEFINED OUT OF THEM.

REJECTION? —
- HALT IN NON-FINAL STATE
- NEVER HALT (INFINITE LOOP)

RECALL IN NFA/PDA WE WOULD HALT WHEN END OF INPUT IS REACHED — THEN WE CHECK IF STATE IS FINAL

IN T.M. INPUT IS NOT STREAMING BY BUT IS INSTEAD GIVEN ON TAPE — SO WE NEED EXPLICIT NOTION OF HALTING.

HALTING WHEN IN STATE q AND HEAD SEES X, SUCH THAT $\delta(q, X)$ IS UNDEFINED.

FINAL STATES HAVE ABSOLUTELY NO TRANSITIONS — ALWAYS HALT

EXAMPLE $L = \{0^n 1^n \mid n \geq 1\}$

INITIALLY

$\cdots BBB\,0000\,1111\,BBB \cdots$

$\uparrow$ HEAD

T.M. IDEA MATCH LEFTMOST 0 WITH LEFTMOST 1, REPLACING THEM BY X AND Y (RESPECTIVELY) AND REPEAT.

T.M. M $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

$Q = \{q_0, q_1, q_2, q_3, q_4\}$ $F = \{q_4\}$

$\Sigma = \{0, 1\}$ $\Gamma = \{0, 1, X, Y, B\}$

TRANSITIONS

$\delta(q_0, 0) = (q_1, X, R)$
$\delta(q_0, Y) = (q_3, Y, R)$

REPLACE 0 BY X AND LOOK FOR A MATCHING 1 — BUT IF Y IS SEEN, GO FOR ENDGAME

$\delta(q_1, 0) = (q_1, 0, R)$
$\delta(q_1, Y) = (q_1, Y, R)$
$\delta(q_1, 1) = (q_2, Y, L)$

SKIP OVER 0'S, Y'S TILL 1 IS FOUND — REPLACE IT BY Y AND START HEADING BACK TO LEFT

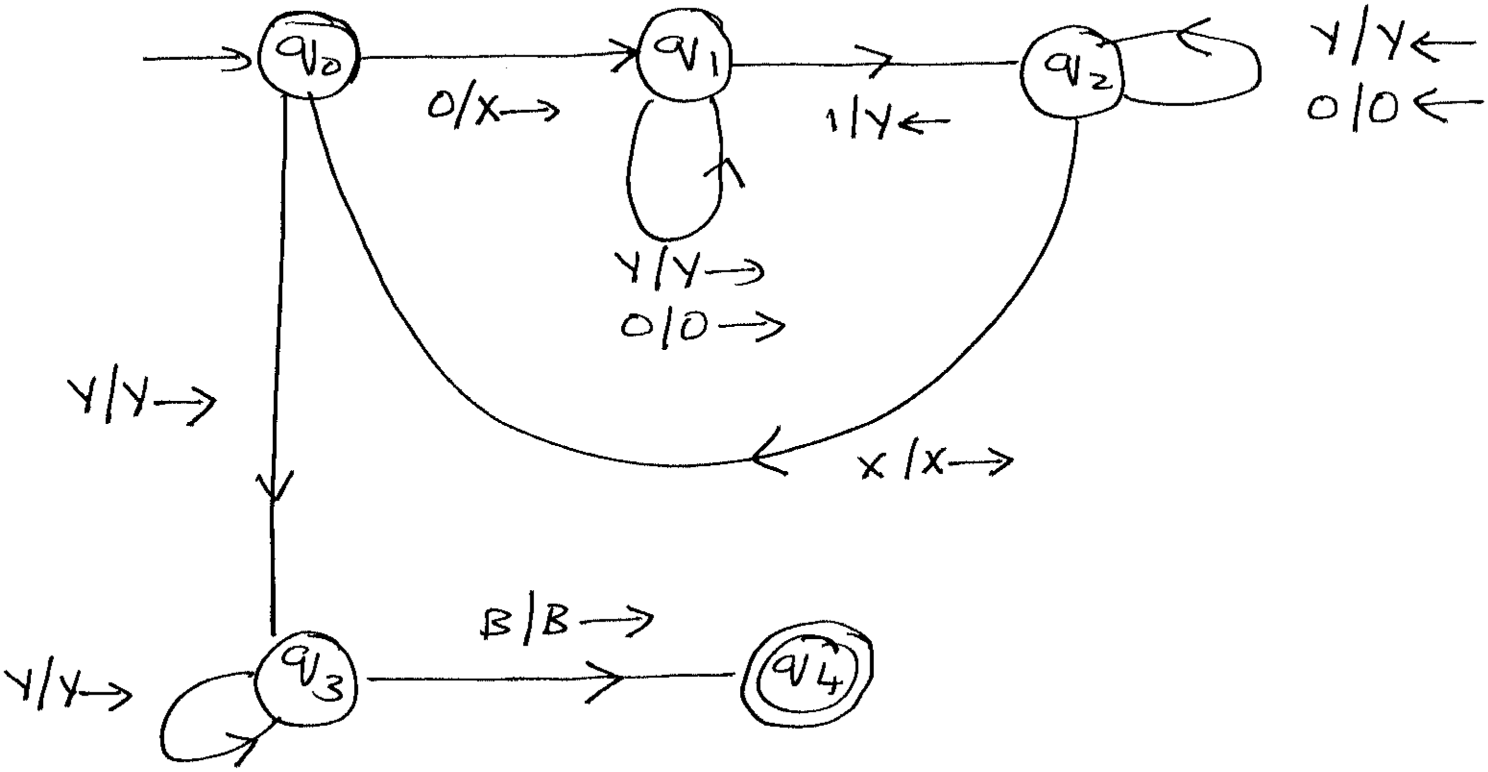
$\delta(q_2, Y) = (q_2, Y, L)$
$\delta(q_2, 0) = (q_2, 0, L)$
$\delta(q_2, X) = (q_0, X, R)$

MOVE LEFT SKIPPING 0/Y TILL FIRST X IS FOUND — MOVE RIGHT TO LOOK FOR LEFTMOST 0

$\delta(q_3, Y) = (q_3, Y, R)$
$\delta(q_3, B) = (q_4, B, R)$

ENDGAME — MAKE SURE NO EXTRA 1'S LEFT OVER.

TRANSITION DIAGRAM



NOTATION A|B→ MEANS REPLACE A BY B ON TAPE CELL BEING SCANNED, MOVE RIGHT

REMARK LOTS OF TRANSITIONS UNDEFINED — IF INPUT DOES NOT MEET DESIRED FORMAT, THE T.M. WILL GET "STUCK" AND HALT IN NON-FINAL

I.D. USED TO SHOW EXECUTION LIKE IN PDAS

$$\text{I.D.} = \alpha_1 \, q \, \alpha_2 \quad \underline{\text{WITH}} \quad q \in Q$$
$$\alpha_1, \alpha_2 \in \Gamma^*$$

MEANS NON-BLANK PORTION OF TAPE HAS $\alpha_1\alpha_2$ WITH HEAD AT LEFTMOST SYMBOL

THUS $ID = \alpha_1 q \alpha_2$ CORRESPONDS TO

BLANKS	α_1	α_2	BLANKS

$\uparrow$ STATE q.

AS BEFORE WE USE $\vdash, \overset{*}{\vdash}$ TO SHOW ID'S CHANGING.

EXAMPLE

$$
\begin{aligned}
q_0 0011 &\vdash X q_1 011 \vdash X0 q_1 11 \\
&\vdash X q_2 0Y1 \vdash q_2 X0Y1 \\
&\vdash X q_0 0Y1 \vdash XX q_1 Y1 \\
&\vdash XXY q_1 1 \vdash XX q_2 YY \\
&\vdash X q_2 XYY \vdash XX q_0 YY \\
&\vdash XXY q_3 Y \vdash XXYY q_3 \\
&\vdash XXYYB q_4 .
\end{aligned}
$$

LANGUAGE GIVEN DTM M

$$L(M) = \left\{ \omega \;\middle|\; q_0 w \overset{*}{\vdash} \alpha_1 p \alpha_2, \text{ WHERE } p \in F \text{ AND } \alpha_1, \alpha_2 \in \Gamma^* \right\}$$

REMARK WE USE LANGUAGE RECOGNITION AS A CONVENIENT NOTION OF PROBLEM-SOLVING ABILITY

HOWEVER T.M. CAN EASILY COMPUTE FUNCTIONS AND PRODUCE OUTPUT BY LEAVING IT ON THE TAPE

RECURSIVELY ENUMERABLE LANGUAGES

CLASS OF LANGUAGES ACCEPTED BY T.M.

PROGRAMMING TRICKS.

IDEA WE PRESENT SOME NOTATIONAL CONVENIENCES WHICH MAKE IT EASIER TO "PROGRAM" T.M. AND ALSO SERVE TO HIGHLIGHT THEIR GENERALITY AND POWER

BASICALLY WE IMPOSE NOTATIONAL STRUCTURE ON STATES AND TAPE SYMBOLS

TRICK 1 [CPU REGISTERS — USING STATES AS MEMORY STORE]

IDEA ALLOW STATE NAMES TO BE OF THE TYPE

$$[q, X_1, \ldots, X_k]$$

WHERE X_i ACTS AS MEMORIZED SYMBOLS.

EXAMPLE $L = \{ ww^R \mid w \in \{0,1\}^* \}$

DEFINE M

$$Q = \{ [q,-], [q,0], [q,1], [p,0], [p,1], s, \theta \}$$

$$\Sigma = \{0,1\} \qquad \Gamma = \{0,1,B\}$$

$$q_0 = [q,-] \qquad F = \{\theta\}$$

IDEA GIVEN ww^R, MATCH LEFTMOST SYMBOL WITH RIGHTMOST, ERASING BOTH.

USE CPU REGISTER TO STORE LEFTMOST SYMBOL WHILE HEADING RIGHT TO FIND RIGHTMOST

TRANSITIONS

STEP 1

$$\left.\begin{aligned} \delta([q,-],0) &= ([q,0],B,R) \\ \delta([q,-],1) &= ([q,1],B,R) \\ \delta([q,-],B) &= (f,B,R) \end{aligned}\right\}$$

REMEMBER LEFTMOST SYMBOL & REPLACE BY BLANK, OR ACCEPT ϵ

STEP 2

$$\left.\begin{aligned} \delta([q,i],j) &= ([q,i],j,R) \quad \forall\, i,j \in \{0,1\} \\ \delta([q,i],B) &= ([p,i],B,L) \end{aligned}\right\}$$

SKIP TO RIGHTMOST SYMBOL

STEP 3

$$\left.\delta([p,i],i) = (s,B,L) \quad \forall i \in \{0,1\}\right\}$$

MATCH RIGHTMOST WITH REGISTER

STEP 4

$$\left.\begin{aligned} \delta(s,i) &= (s,i,L),\ \forall i \\ \delta(s,B) &= ([q,-],B,R) \end{aligned}\right\}$$

SKIP OVER TO LEFTMOST SYMBOL

TRICK 2 [MULTIPLE TRACKS]

IDEA VIEW TAPE AS HAVING MULTIPLE TRACK AND Γ AS HAVING COMPOSITE SYMBOLS

0	Y	0
1	1	Z
X	1	0

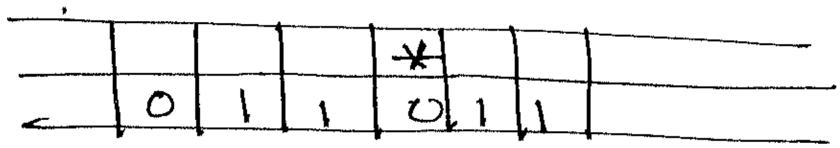
THUS Γ NOW CONTAINS SYMBOLS SUCH AS

$$\begin{bmatrix} 0 \\ 1 \\ X \end{bmatrix}, \begin{bmatrix} Y \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ Z \\ 0 \end{bmatrix}$$

CONSIDER $L = \{ww \mid w \in \{0,1\}^+\}$

NOTE FIRST NEEDS TO FIND MID-POINT, AND THEN WE CAN USE MATCHING PROCESS AS IN $w.w^R$

TO FIND MID-POINT WE VIEW TAPE AS 2 TRACKS



WHERE WE USE THE TOP TRACK TO PUT MARKERS OVER SYMBOLS.

IDEA PUT MARKERS ON LEFTMOST | RIGHTMOST SYMBOLS AND SLOWLY MOVE THEM IN TILL THEY MEET AT THE MID-POINT.

TAPE SYMBOLS $\begin{bmatrix}B\\B\end{bmatrix}, \begin{bmatrix}B\\0\end{bmatrix}, \begin{bmatrix}B\\1\end{bmatrix}, \begin{bmatrix}*\\0\end{bmatrix}, \begin{bmatrix}*\\1\end{bmatrix}$

COULD CALL THESE B, 0, 1, A, B BUT IT WOULD BE LESS INSIGHTFUL IN GENERAL.

ASSUME INITIALLY IN STATE q_0, SCANNING LEFTMOST

$$\delta\left(q_0, \begin{bmatrix}B\\i\end{bmatrix}\right) = \left(q_1, \begin{bmatrix}*\\i\end{bmatrix}, R\right)$$

$$\delta\left(q_1, \begin{bmatrix}B\\i\end{bmatrix}\right) = \left(q_1, \begin{bmatrix}B\\i\end{bmatrix}, R\right)$$

$$\delta\left(q_1, \begin{bmatrix}B\\B\end{bmatrix}\right) = \left(q_2, \begin{bmatrix}B\\B\end{bmatrix}, L\right)$$

$$\delta(q_1, [{* \atop i}]) = (q_2, [{B \atop i}], L)$$

$$\delta(q_2, [{B \atop i}]) = (q_3, [{* \atop i}], L)$$

$$\delta(q_3, [{B \atop i}]) = (q_3, [{B \atop i}], L)$$

$$\delta(q_3, [{* \atop i}]) = (q_0, [{B \atop i}], R)$$

NOTE ONE EACH OF ABOVE TRANSITIONS FOR $i \in \{0,1\}$

AT END WE HAVE HEAD POINTING TO FIRST SYMBOL OF SECOND w WITH A $*$ ABOVE IT, IN STATE q_0

TRICK 3 [SUBROUTINES | PROCEDURE CALLS]

EXAMPLE SHIFTING OVER

$$\left.\begin{array}{ll} \text{GIVEN} & ID_1 = \alpha\, q_i\, X \beta \\ \text{WANT} & ID_2 = \alpha \,\square\, q_i\, X \beta \end{array}\right\} \begin{array}{l} X \in \Gamma \\ \alpha, \beta \in \Gamma^* \\ \square \in \Gamma \end{array}$$

SUBROUTINE CAN BE USED REPEATEDLY TO CREATE SPACE IN MIDDLE OF THE TAPE

FOR EXAMPLE USEFUL FOR IMPLEMENTING COUNTERS AS A PART OF COMPLEX PROCES

$\$0\$ \rightarrow \$1\$ \rightarrow \$\square 1\$ \rightarrow \$01\$$
$\rightarrow \$10\$ \rightarrow \$11\$ \rightarrow \$\square 11\$$
$\rightarrow \$011\$ \rightarrow \$100\$ \rightarrow \ldots$

PROCEDURE CALL $\delta(q_i, X) = \left([p, X], \begin{bmatrix} j \\ \square \end{bmatrix}, R\right), \forall X \in \Gamma$

- MEMORIZE RETURN STATE p_j, ERASED SYMBOL X
- STATE p INVOKES PROCEDURE

PROCEDURE p

1) SHIFT 1 CELL TO THE RIGHT

$$\delta([p, X], Y) = ([p, Y], X, R)$$

$\forall X, Y \in \Gamma$ WITH $Y \neq B$

2) TILL REACHED END OF β

$$\delta([p, Y], B) = (t, Y, L) \qquad \forall Y \in \Gamma$$

3) RETURN TO CALLING POINT

$$\delta(t, Y) = (t, Y, L) \qquad \forall Y \neq \begin{bmatrix} j \\ \square \end{bmatrix}$$

4) EXIT RETURN TO STATE p_j

$$\delta\left(t, \begin{bmatrix} j \\ \square \end{bmatrix}\right) = (q_j, \square, R)$$

NOTE WE CAN IMPLEMENT ANY KIND OF PROCEDURE / FUNCTION CALL WITH ANY KIND OF PARAMETER PASSING — WITH ENOUGH WORK.

TEXTBOOK SEE SIMPLER NOTION OF SUBROUTINES

ENHANCING T.M.

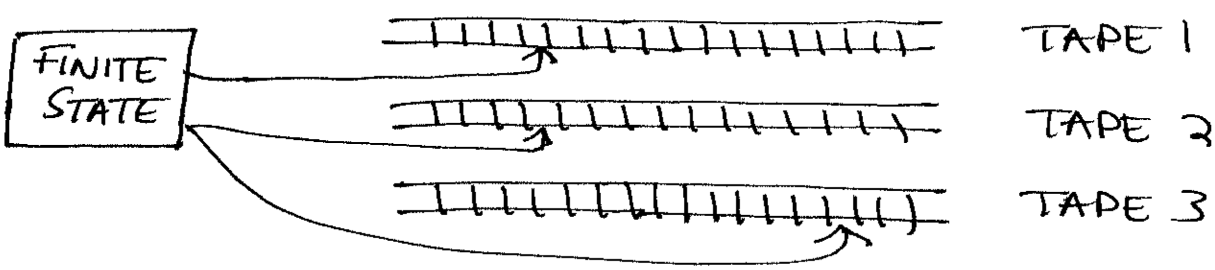
OBSERVE IF T.M. SEEN SO FAR IS ABLE TO CAPTURE ALL THAT WE CAN COMPUTE, THEN ADDING FEATURES TO IT SHOULD NOT ENHANCE ITS POWER

WE SHOW ADDING FOLLOWING FEATURES GIVES T.M. WHICH CAN BE EASILY SIMULATED BY OUR STANDARD T.M.

- MULTIPLE TAPES/HEADS
- NON-DETERMINISM

EXERCISE TRY TO THINK OF OTHER WAYS TO ENHANCE T.M. AND WHETHER STANDARD T.M. CAN SIMULATE THESE FEATURES.

MULTI-TAPE T.M.



INITIALLY INITIALLY, INPUT w IS ON TAPE 1 WITH TAPE-HEAD AT LEFTMOST SYMBOL — REMAINING TAPES ARE BLANK.

EACH HEAD INDEPENDENT & PART OF TRANSITION

$$\delta(q, h_1, h_2, \ldots, h_k) = (p, (w_1, m_1), (w_2, m_2), \ldots)$$

h_i : SYMBOL UNDER HEAD i

w_i : WRITE IN h_i

CONSIDER SIMULATING k-TAPE T.M. M_k BY 1-TAPE T.M. M_1

IDEA USE 2k TRACKS IN M_1 — FOR EACH TAPE OF M_k USE ONE TRACK OF M_1 TO STORE <u>TAPE CONTENTS</u> AND ANOTHER TRACK TO MARK <u>HEAD POSITION</u> WITH *

TAPE 1	A	B	X	Y	Z
HEAD 1			*		
TAPE 2	0	1	0	1	1
HEAD 2	*				
TAPE 3	a	b	c	d	e
HEAD 3					*

IN M_1 EACH TRANSITION OF M_k IS SIMULATED BY A WHOLE SERIES OF TRANSITIONS

STEP 0 START AT LEFTMOST CELL WHERE SOME TRACK CONTAINS A NON-BLANK SYMBOL

STEP 1 SWEEP RIGHT — PICKING UP EACH h_i BY NOTING SYMBOL MARKED BY EACH $*_j$ STORING h_i'S IN "CPU REGISTER"

STEP 2 SWEEP LEFT — WRITING w_i AND MOVING *'S AS PER M_k'S TRANSITION.

VERIFY CAN CONSTRUCT M_1 TO SIMULATE M_k WITHOUT AFFECTING THE LANGUAGE.

REMARK FROM NOW ON, I WILL PROVIDE ONLY SUCH A HIGH-LEVEL VIEW OF T.M. CONSTRUCTIONS — LOW-LEVEL PROGRAMMING DETAILS ARE VERY CUMBERSOME AND OMITTED — JUST VERIFY T.M CAN BE CONSTRUCTED

SIMULATION SPEED.

OBSERVE WHILE ENHANCEMENTS DO NOT ~~AFFECT~~ AFFECT THE POWER OF T.M., THEY DO IMPACT ITS EFFICIENCY.

RUNNING TIME T.M. M IS SAID TO HAVE RUNNING TIME $T(n)$ IF IT HALTS WITHIN $T(n)$ STEPS ON ALL INPUTS OF LENGTH n (NOTE $T(n)$ COULD BE INFINITE)

THM IF M_k HAS RUNNING TIME $T(n)$, THEN M_1 WILL SIMULATE IT WITH RUNNING TIME $O(T(n)^2)$.

PROOF SUPPOSE INPUT ω HAS LENGTH n.

THEN M_k CAN USE $T(n)$ TIME ON IT

CLAIM THE k HEADS OF M_k CANNOT BE MORE THAN $2T(n)$ APART AND M_k USES $\leq 2T(n)$ TAPE CELLS ON EACH TAPE.

WHY? AT EACH STEP, LEFTMOST AND RIGHTMOST TAPE HEADS CAN DRIFT APART BY AT MOST 2 ADDITIONAL CELLS

CONSIDER M_1

- MAKES 2 SWEEPS FOR TRANSITION OF M_k
- TIME PER SWEEP IS $O(T(n))$
- NUMBER OF TRANSITIONS OF M_k IS $\leq T(n)$

THUS TOTAL TIME $= O(T(n)^2)$.