COURSE: THEORY OF AUTOMATA COMPUTATION

TOPICS TO BE COVERED

- Equivalence of NTM and DTM

EQUIVALENCE OF NTM AND DTM

Theorem: For any NTM M_n, there exists a DTM M_d such that:

- if M_n halts on input α with output β, then M_d halts on input α with output β, and
- if M_n does not halt on input α, then M_d does not halt on input α.

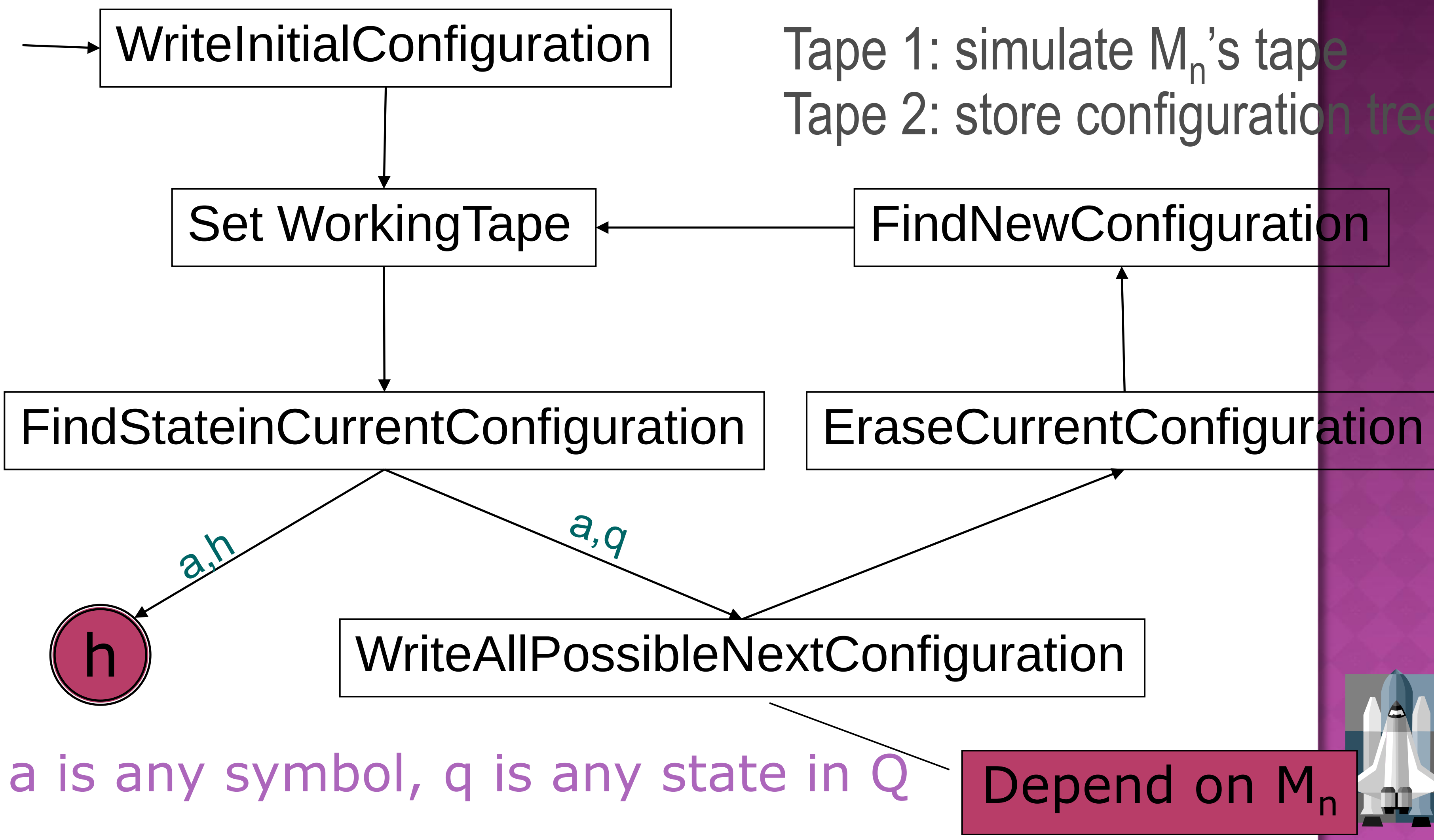
Proof:

Let $M_n = (Q, \Sigma, \Gamma, \delta, s)$ be an NTM.

We construct a 2-tape TM M_d from M_n as follows:

CONSTRUCT A DTM EQUIVALENT TO AN NTM

Tape 1: simulate M_n's tape
Tape 2: store configuration tree



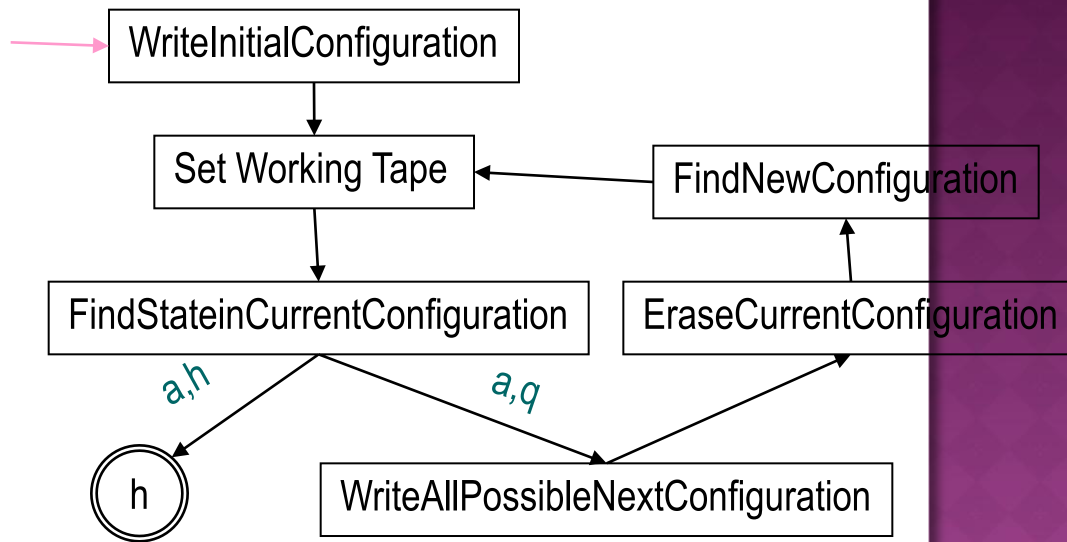
a is any symbol, q is any state in Q

HOW M_D WORKS

Tape 1

@	0	1	1	0	Δ

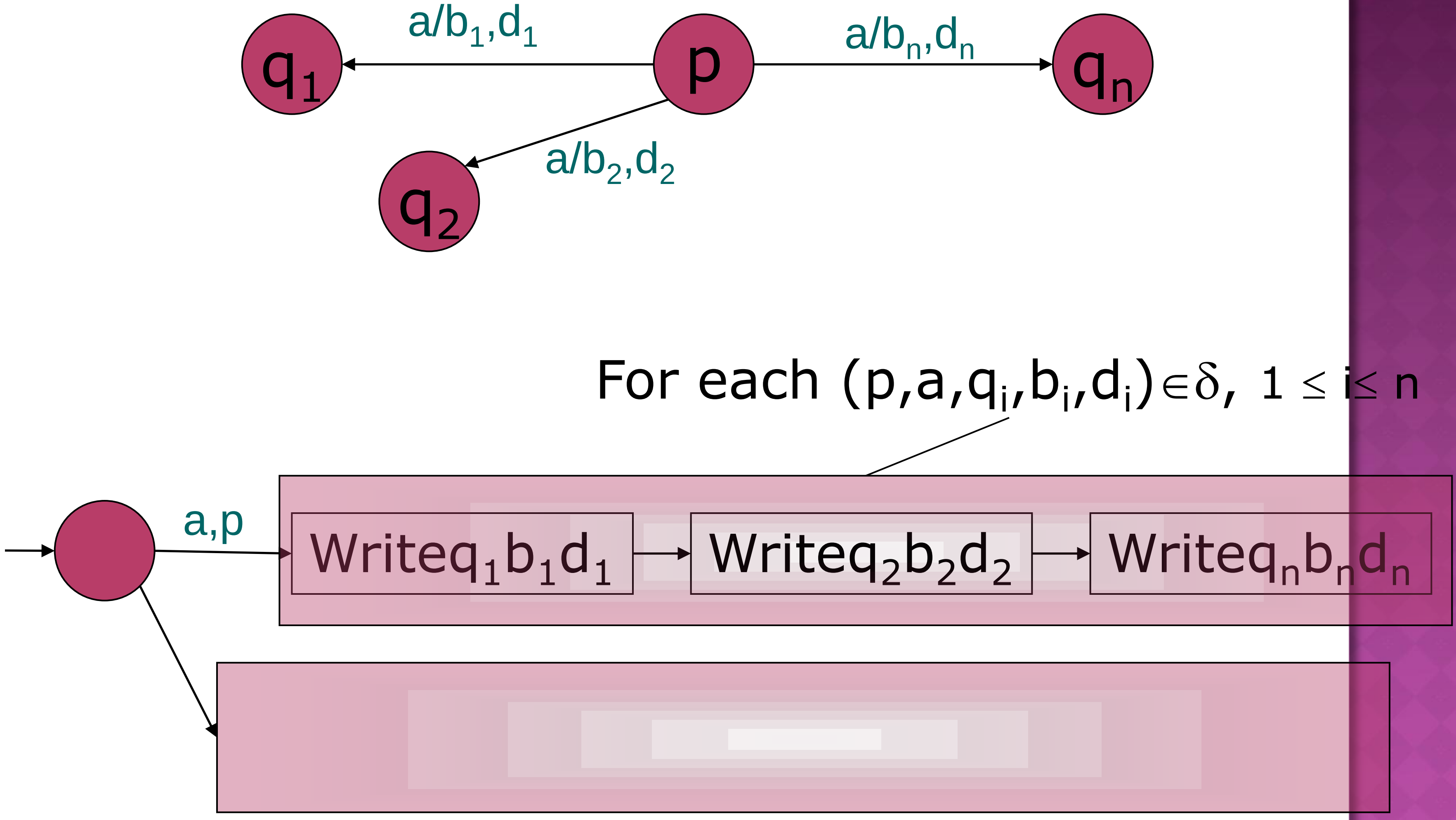
Current state: s



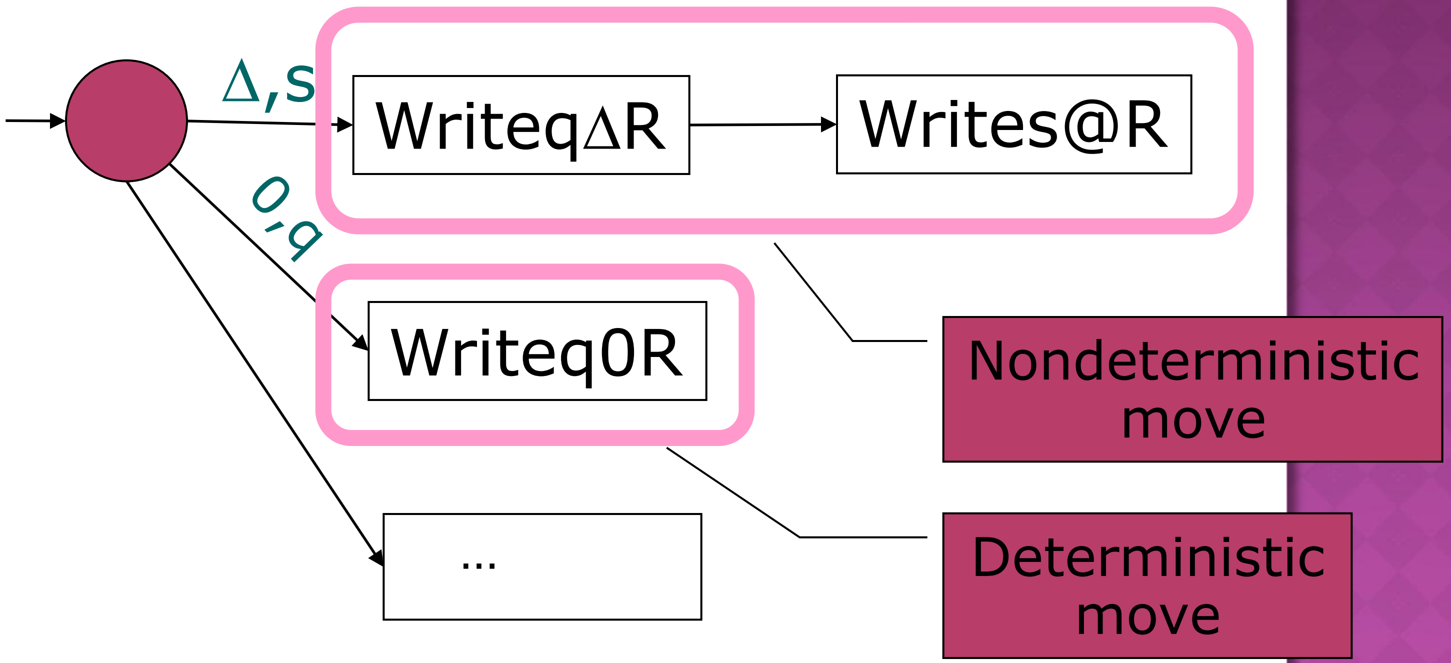
Tape 2

-	_	-	-	-	-	-	_	-	_	-	-	-	-	-	_	*
*	@	s	0	1	1	0	Δ	#	Δ	0	1	q	1	0	Δ	#

WRITEALLPOSSIBLENEXTCONFIGURATION



EXAMPLE: WRITEALLPOSSIBLENEXTCONFIGURATION



IF M_N HALTS ON INPUT α WITH OUTPUT β

- Then, there is a positive integer n such that the initial configuration $(s, \underline{\Delta}\alpha)$ of M_n yeilds a halting configuration $(h, \underline{\Delta}\beta)$ in n steps.
- From the construction of M_d, the configuration $(h, \underline{\Delta}\beta)$ must appear on tape 2 at some time.
- Then, M_d must halt with β on tape 1.

IF M_N DOES NOT HALT ON INPUT α

- Then, M_n cannot reach the halting configuration. That is, $(s, \underline{\Delta}\alpha)$ never yields a halting configuration $(h, \underline{\Delta}\beta)$.
- From the construction of M_d, the configuration $(h,\underline{\Delta}\beta)$ never appears on tape 2.
- Then, M_d never halt.

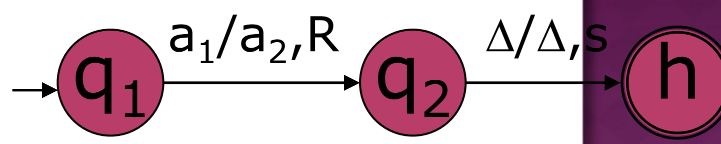
UNIVERSAL TURING MACHINE

- Given the description of a DTM T and an input string z, a universal TM simulates how T works on input z.
- What's need to be done?
  - How to describe T and z on tape
    - Use an encoding function
  - How to simulate T

ENCODING FUNCTION

- Let $T=(Q, \Sigma, \delta, s)$ be a TM. The encoding function e(T) is defined as follows:
  - e(T)=e(s)#e(δ),
  - e(δ)=e(m_1)#e(m_2)#…#e(m_n)#, where $\delta = \{m_1, m_2,\ldots, m_n\}$
  - e(m)=e(p),e(a),e(q),e(b),e(d), where $m = (p, a, q, b, d)$
  - e(z)=1e(z_1)1e(z_2)1…1e(z_m)1, where $z=z_1z_2\ldots z_m$ is a string
  - e(Δ)=0, e(a_i)=0^{i+1}, where a_i is in Σ
  - e(h)=0, e(q_i)=0^{i+1}, where q_i is in Q

EXAMPLE OF ENCODED TM



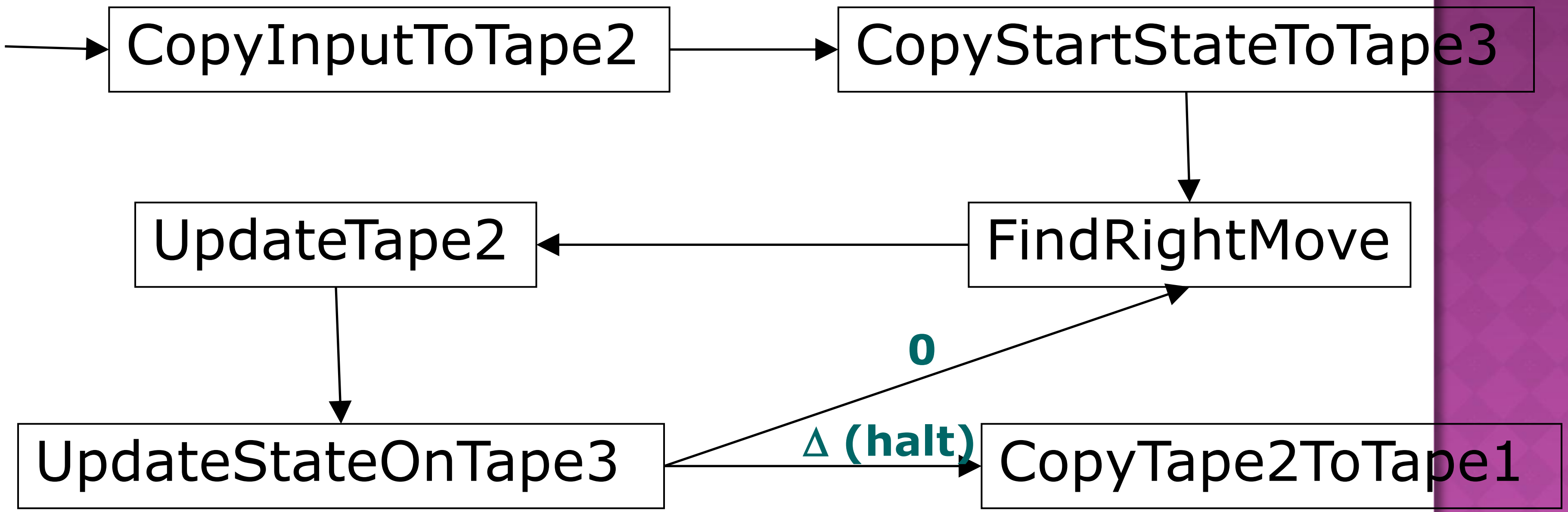
- $e(\Delta)=0$, $e(a_1)=00$, $e(a_2)=000$
- $e(h)=0$, $e(q_1)=00$, $e(q_2)=000$
- $e(S)=0$, $e(L)=00, e(R)=000$
- $e(\Delta a_1 a_1 a_2 \Delta) = 1e(\Delta)1e(a_1)1e(a_1)1e(a_2)1e(\Delta)1$
  $= 101001001000101$
- $e(m_1) = (q_1),e(a_1),e(q_2),e(a_2),e(R)$
  $= 00,00,000,000,000$
- $e(m_2) = e(q_2),e(\Delta),e(h),e(\Delta),e(S)$
  $= 000,0,0,0,0$
- $e(\delta) = e(m_1)\#e(m_2)\#\ldots\#$
  $= 00,00,000,000,000\#000,0,0,0,0\#\ldots\#$
- $e(T) = e(s)\#e(\delta)$
  $= 00\#00,00,000,000,000\#000,0,0,0,0\#\ldots\#$
- Input = $e(Z)|e(T)|$
  $= 101001001000101|00\#00,00,000,000,000\#000,0,0,0,0\#\ldots\#|$

UNIVERSAL TURING MACHINE

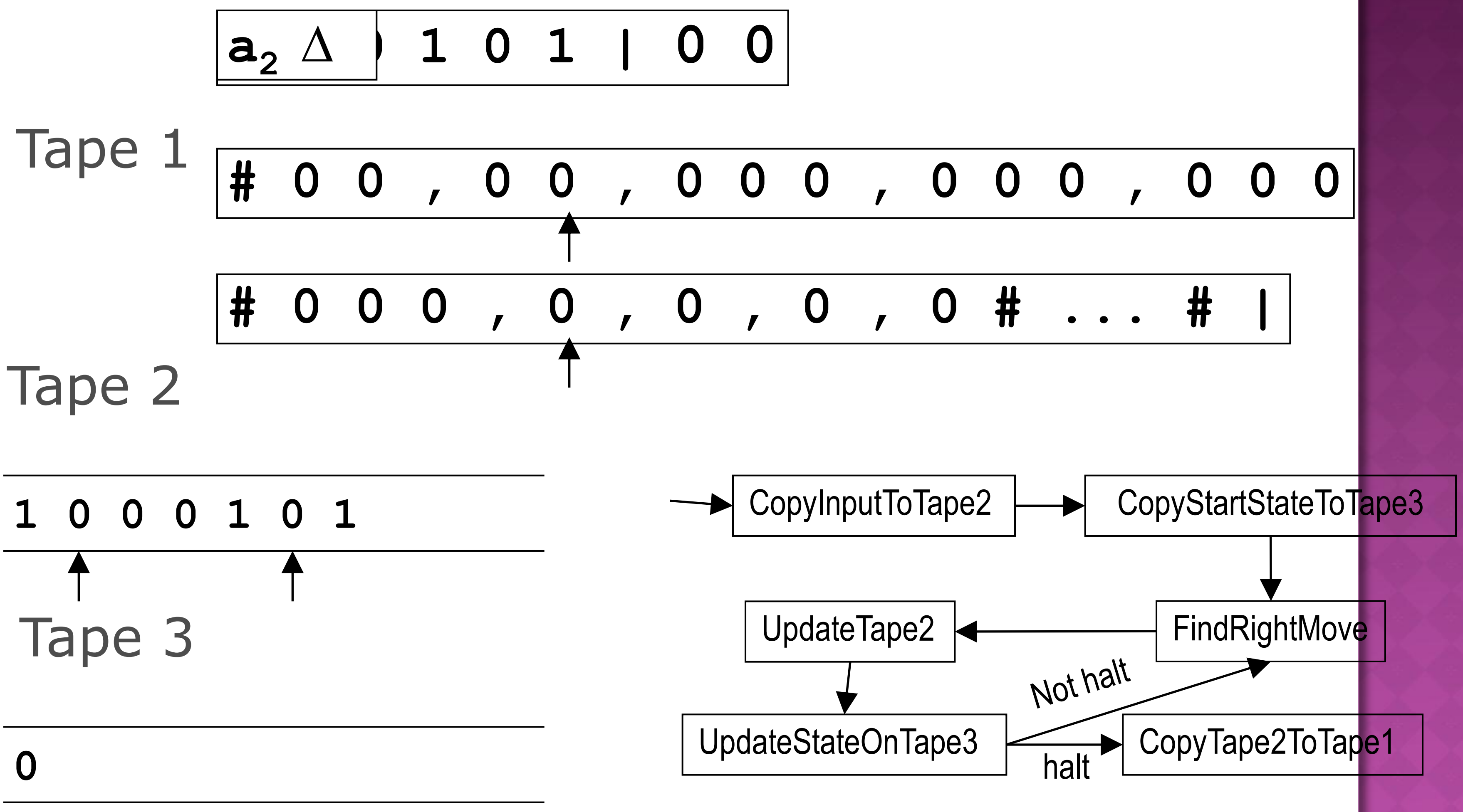
Tape 1: I/O tape, store the transition function of T and input of T

Tape 2: simulate T's tape

Tape 3: store T's state



HOW UTM WORKS



CHURCH-TURING THESIS

- Turing machines are formal versions of algorithms.
- No computational procedure will be considered an algorithm unless it can be presented as a Turing machine.

CHECKLIST

- Construct a DTM, multitape TM, NTM accepting languages or computing function
- Construct composite TM
- Prove properties of languages accepted by specific TM
- Prove the
- Describe the relationship between TM and FA
- Prove the relationship between TM and FA