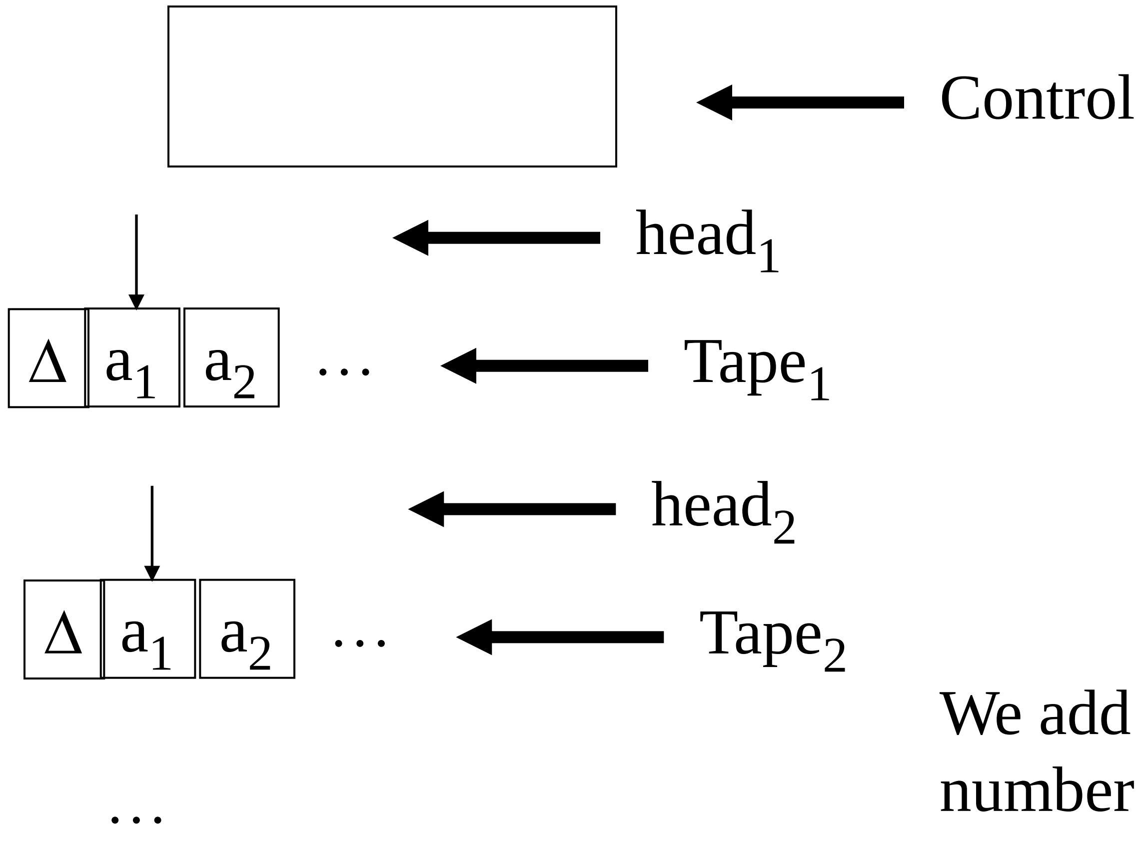
Multi-tape Turing Machines: Informal Description



We add a finite number of tapes

Multi-tape Turing Machines: Informal Description (II)

- Each tape is bounded to the left by a cell containing the symbol Δ

- Each tape has a unique header

- Transitions have the form (for a 2-tape Turing machine):

$$((p,(x_1, x_2)), (q, (y_1, y_2)))$$

Such that each x_i is in Σ and each y_i is in Σ or is $\rightarrow$ or $\leftarrow$.
and if $x_i = \Delta$ then $y_i = \rightarrow$ or $y_i = \Delta$

Multi-tape Turing Machines

Construct a 2-tape Turing machine that recognizes the language:

$$L = \{a^n b^n : n = 0, 1, 2, \ldots\}$$

Input:

Tape1: $\underline{\Delta}w$

Tape2: $\underline{\Delta}$

Hints:

- use the second tape as an stack
- Use the machines M1 and M0

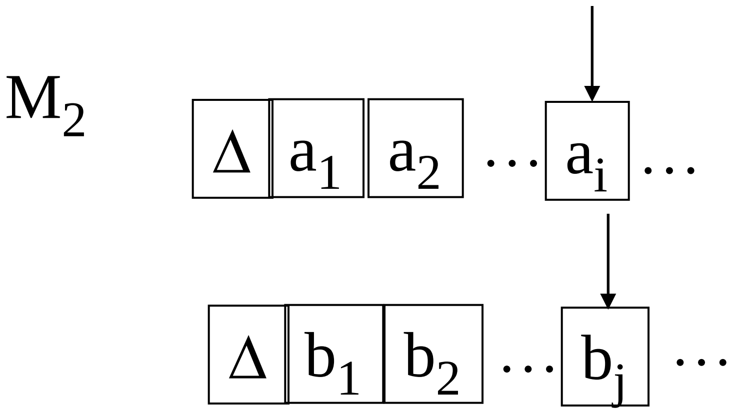
Output:

Tape1: $\Delta 1\ldots$ if $w \in L$

or

Tape1: $\Delta 0\ldots$ if $w \notin L$

Multi-tape Turing Machines vs Turing Machines



We can simulate a 2-tape Turing machine M2 in a Turing machine M:

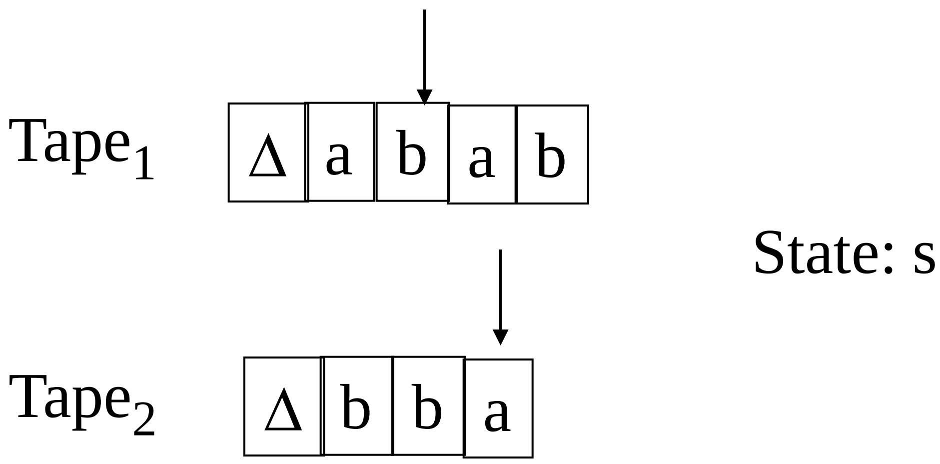
- we can represent the contents of the 2 tapes in the single tape by using special symbols

- We can simulate one transition from the M2 by constructing multiple transitions on M

- We introduce several (finite) new states into M

Using States to "Remember" Information

Configuration in a 2-tape Turing Machine M2:



State in the Turing machine M: "s+b+1+a+2"

Which represents:

- M2 is in state s
- Cell pointed by first header in M2 contains b
- Cell pointed by second header in M2 contains an a

Using States to "Remember" Information (2)

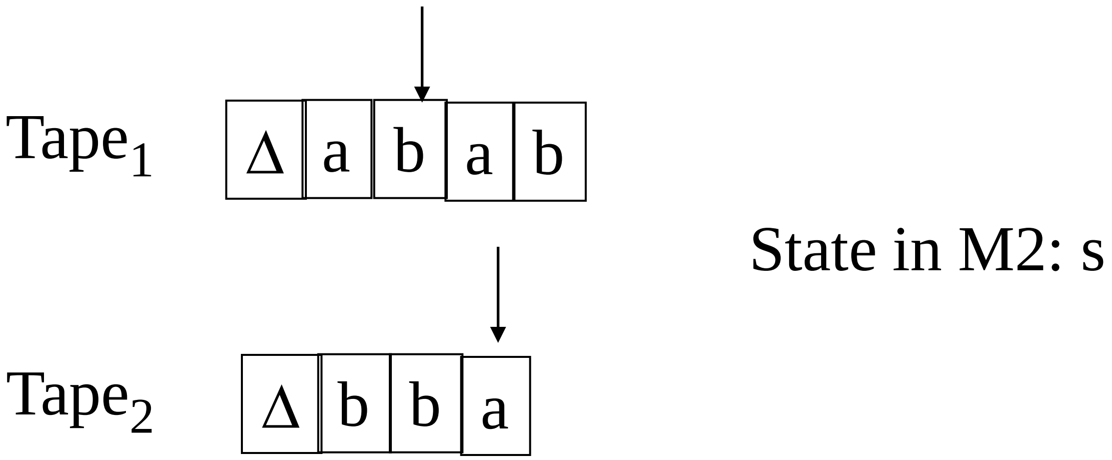
State in the Turing machine M: "s+b+1+a+2"

How many states are there in M?

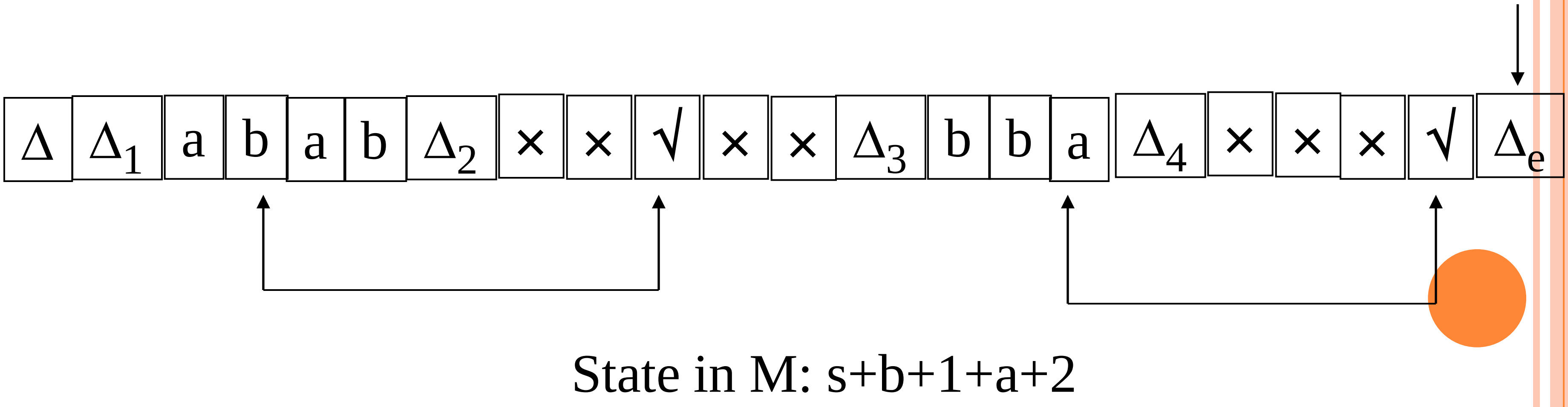
(# states in M2) * |Σ or → or ←| * |Σ or → or ←|

Yes, we need large number of states for M but it is finite!

Configuration in a 2-tape Turing Machine M2:



Equivalent configuration in a Turing Machine M:



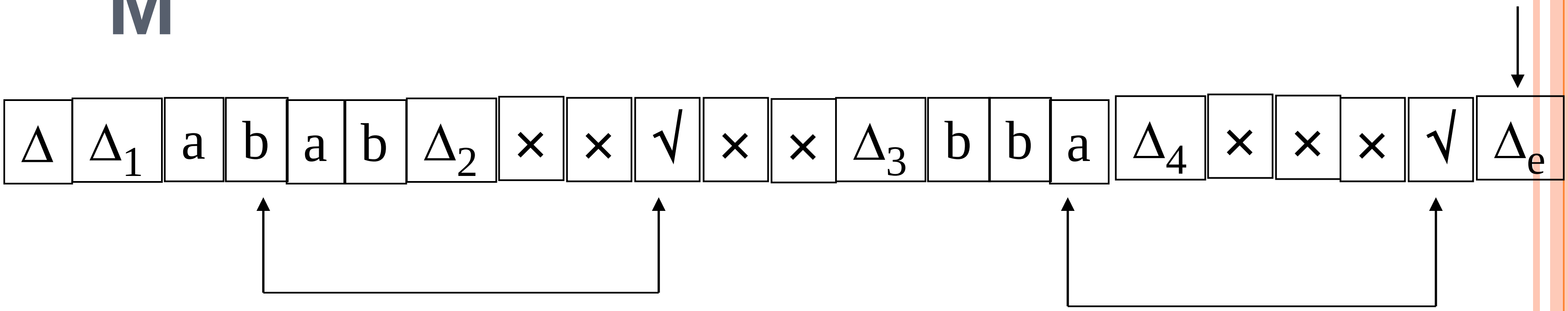
SIMULATING M2 WITH M

- The alphabet Σ of the Turing machine M extends the alphabet Σ_2 from the M_2 by adding the separator symbols: Δ_1, Δ_2, Δ_3 , Δ_4 and Δ_e, and adding the mark symbols: √ and ×

- We introduce more states for M, one for each 5-tuple p+α+1+ β+2 where p in an state in M_2 and α+1+ β+2 indicates that the head of the first tape points to α and the second one to β

- We also need states of the form p+←+1+→+2 for control purposes

SIMULATING TRANSITIONS IN M2 WITH M



State in M: s+b+1+a+2

- At the beginning of each iteration of M2, the head starts at Δ_e and both M and M2 are in an state s
- We traverse the whole tape do determine the state $p+\alpha+1+\beta+2$,

Thus, the transition in M2 that is applicable must have the form:

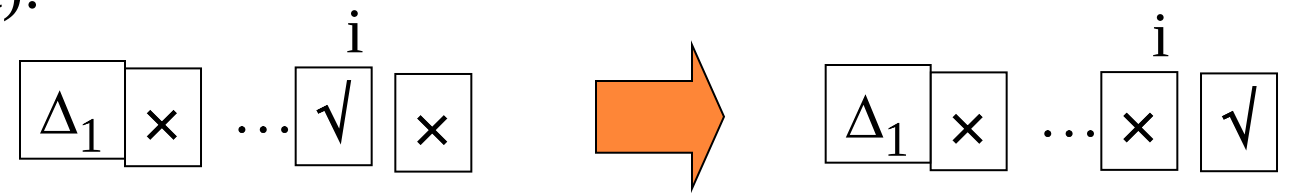
$$((p,(\alpha, \beta)), (q,(\gamma,\psi))) \text{ in } M_2$$

$$p+\alpha+1+\beta+2 \Rightarrow q+\gamma+1+\psi+2 \quad \text{in M}$$

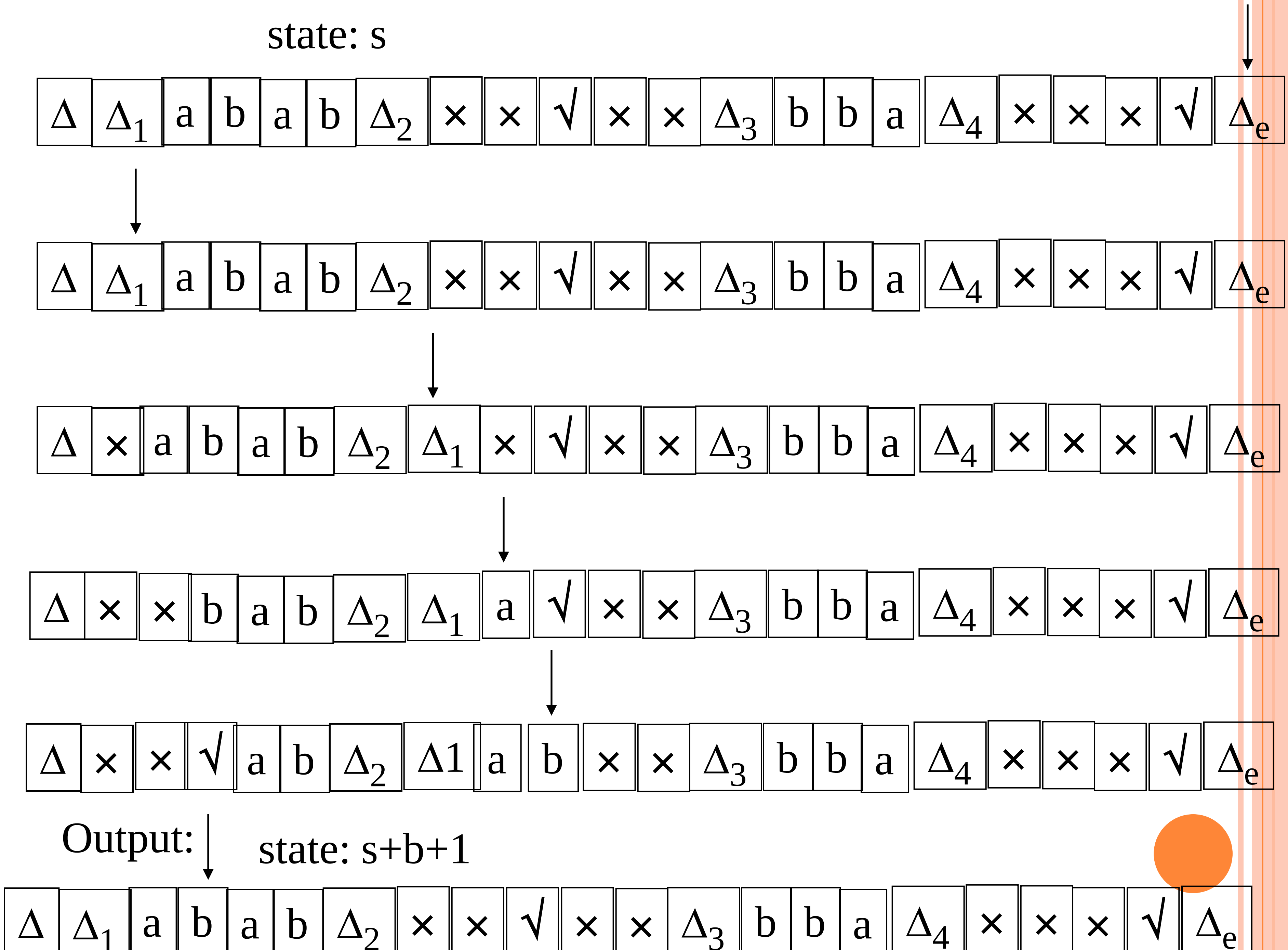
Simulating transitions in M2 with M (2)

- To apply the transformation (q,(γ,ψ)), we go forwards from the first cell.

- If the γ (or ψ) is → (or ←) we move the marker to the right (left):



- If the γ (or ψ) is a character, we first determine the correct position and then overwrite



state: s
Δ Δ1 a b a b Δ2 × × √ × × Δ3 b b a Δ4 × × × √ Δe
Δ Δ1 a b a b Δ2 × × √ × × Δ3 b b a Δ4 × × × √ Δe
Δ × a b a b Δ2 Δ1 × √ × × Δ3 b b a Δ4 × × × √ Δe
Δ × × b a b Δ2 Δ1 a √ × × Δ3 b b a Δ4 × × × √ Δe
Δ × × √ a b Δ2 Δ1 a b × × Δ3 b b a Δ4 × × × √ Δe
Output:
state: s+b+1
Δ Δ1 a b a b Δ2 × × √ × × Δ3 b b a Δ4 × × × √ Δe

Multi-tape Turing Machines vs Turing Machines (6)

- We conclude that 2-tape Turing machines can be simulated by Turing machines. Thus, they don't add computational power!

- Using a similar construction we can show that 3-tape Turing machines can be simulated by 2-tape Turing machines (and thus, by Turing machines).

- Thus, k-tape Turing machines can be simulated by Turing machines

IMPLICATIONS

•If we show that a function can be computed by a k-tape Turing machine, then the function is Turing-computable

•In particular, if a language can be decided by a k-tape Turing machine, then the language is decidable

Example: Since we constructed a 2-tape TM that decides $L = \{a^n b^n : n = 0, 1, 2, \ldots\}$, then L is Turing-computable.

Implications (2)

Example: Show that if L1 and L2 are decidable then L1 $\cup$ L2 is also decidable

Proof. …

HOMEWORK

1. Prove that (ab)* is Turing-enumerable (**Hint**: use a 2-tape Turing machine.)

2. Exercise 4.24 a) and b) (**Hint**: use a 3-tape Turing machine.)

3. For proving that Σ* is Turing-enumerable, we needed to construct a Turing machine that computes the successor of a word. Here are some examples of what the machine will produce (w → w' indicates that when the machine receives w as input, it produces w' as output)

   a → b → aa → ab → ba → bb → aaa