

Konečné automaty (Finite State Machines) III

Zásobníkový automat, Úvod k Turingovu Stroji

Assoc. prof. Roman Senkerik

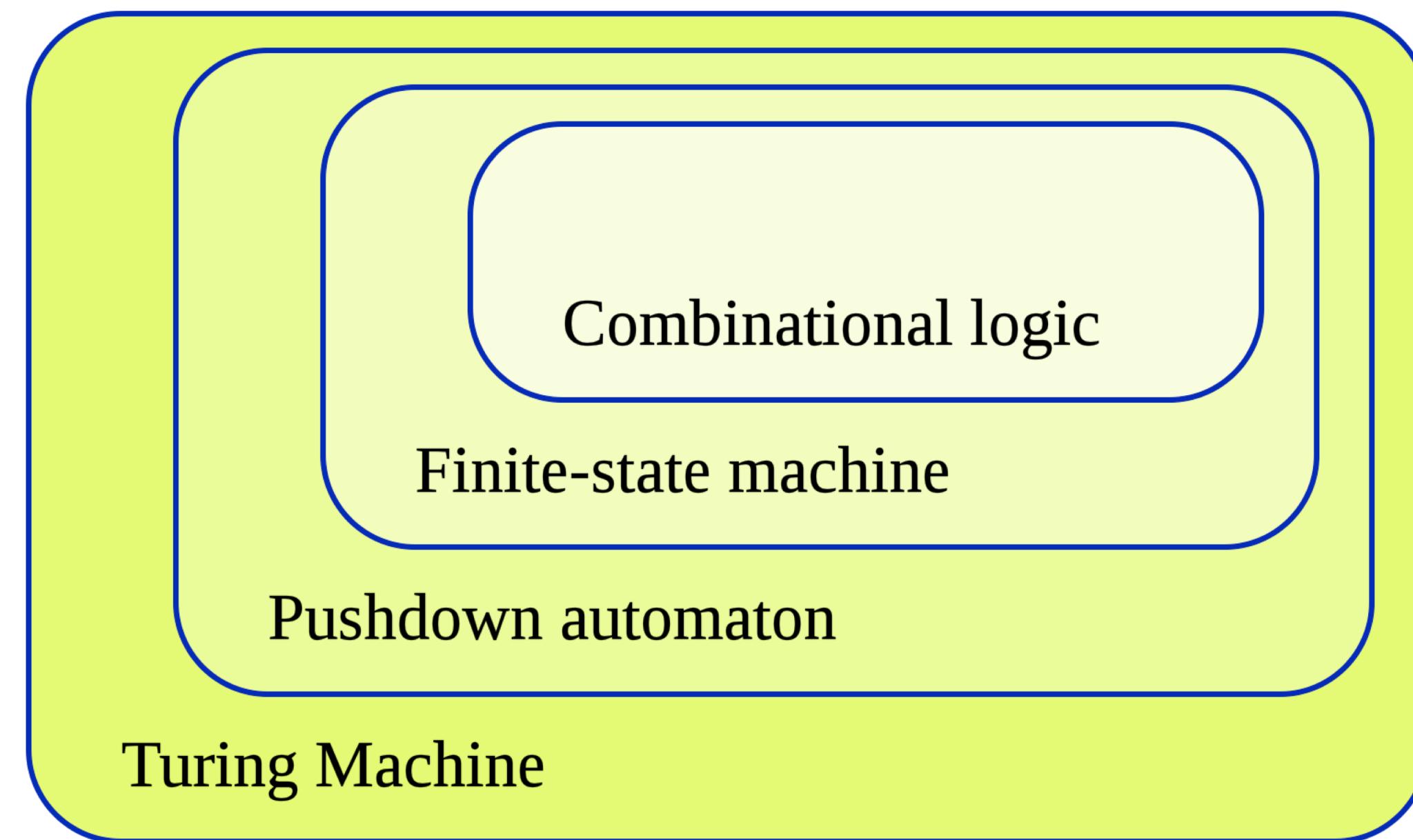


Obsah Prezentace

- Zásobníkový automat
- Turingův stroj - teorie
- Turingův stroj - příklad

1. Zásobníkový automat (Pushdown Automaton)

Automata theory

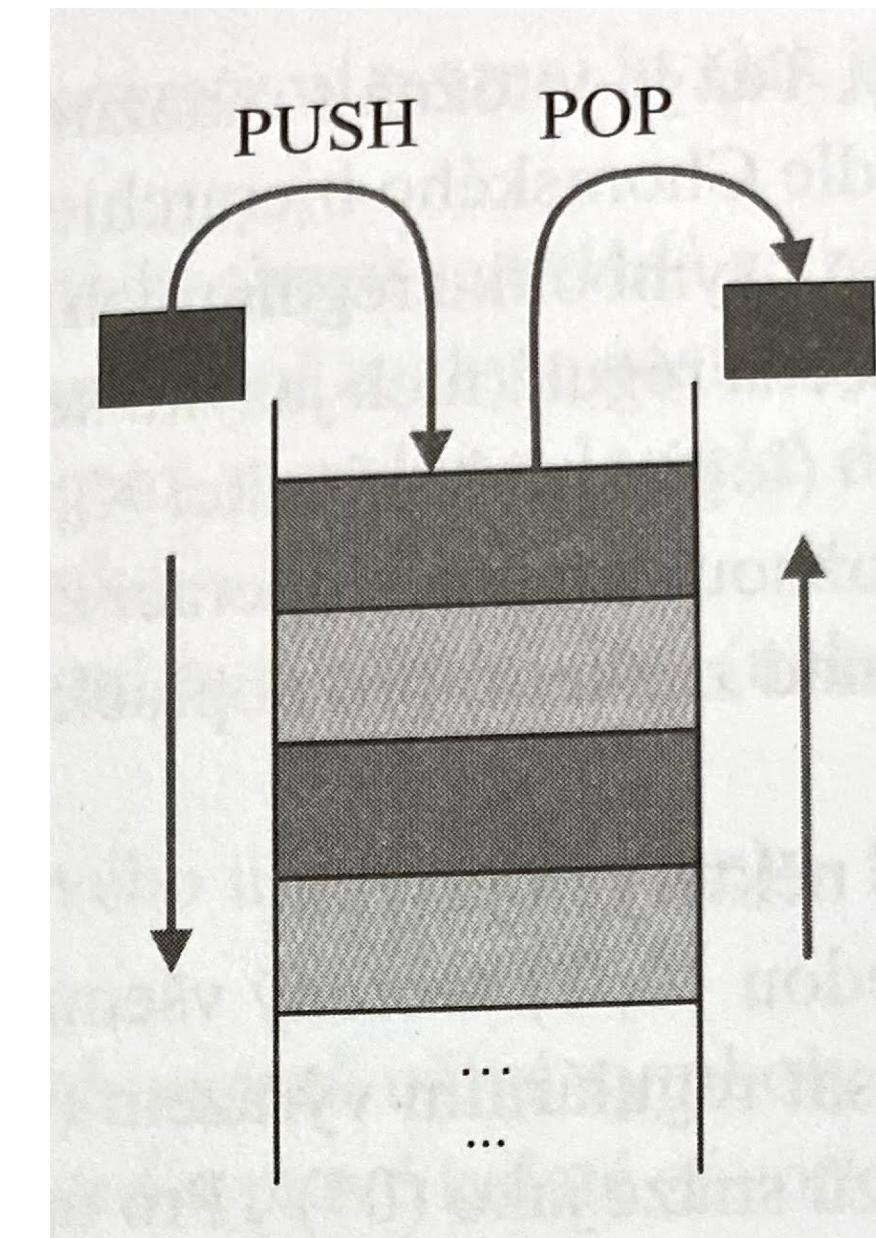


Zásobníkový automat

- Hlavní slabinou konečného automatu byla existence pouze konečné a předem omezené paměti (dané množinou stavů).
- Zásobníkový automat (neplést se “stack automaton”) je rozšířením pojmu konečný automat.
- Deterministické zásobníkové automaty dokážou rozpoznat všechny deterministické bezkontextové jazyky, zatímco nedeterministické dokážou rozpoznat všechny bezkontextové jazyky, přičemž první z nich se často používá při návrhu parseru.

Zásobníkový automat - Pushdown Automaton (PDA)

- Zásobníkový automat obsahuje jednu potenciálně neomezenou paměť - zásobník, do kterého lze ukládat (vstupní) hodnoty nebo mezivýsledky.
- Pro čtení nebo zápis ze zásobníku je vždy přístupný pouze horní symbol, ostatní jsou nepřístupné.
- Paměťová jednotka (zásobník) pracuje podle schématu LIFO (poslední dovnitř první ven).



Zásobníkový automat - Pushdown Automaton (PDA)

- Zásobníkový automat pracuje tak, že současně zkoumá symbol vstupního slova a vrchol zásobníku.
- Na základě tří dat, tedy stavu, ve kterém se nachází, přečteného symbolu vstupního slova a vrcholu zásobníku, může automat změnit svůj stav a uložit symbol nebo několik symbolů na vrchol zásobníku.
- Pokud je zásobník prázdný, automat svou práci ukončí.

Zásobníkový automat - Pushdown Automaton (PDA)

- Po přečtení posledního symbolu vstupního slova se automat rozhodne, zda slovo přijme. To lze provést dvěma způsoby:
 1. Podobně jako u konečného automatu, podle toho, zda stav, ve kterém se zastavil po přečtení celého slova, byl definován jako konečný (přijímající) stav.
 2. Přijme slovo, pokud po přečtení jeho posledního symbolu zůstane zásobník prázdný, pokud zásobník prázdný není, slovo odmítнěte.

Zásobníkový automat - Pushdown Automaton (PDA)

- I takto funkčně velmi omezená paměť výrazně zvýší možnosti stroje.
- Představte si problém, kdy na vstupu máme posloupnost znaků 0 a 1. Po libovolně dlouhé sekvenci následuje oddělovací znak c, za ním další nuly a jedničky až do konce vstupního slova (zadání). Úkolem je zjistit, zda je druhá část sekvence zrcadlovým obrazem první části nebo ne.
- S pomocí konečného automatu je tato úloha zjevně neřešitelná. Konečný automat nemá žádný prostor, ve kterém by si pamatoval, „jak vypadala první část sekvence“.

Zásobníkový automat - Pushdown Automaton (PDA)

- Tento úkol může vyřešit zásobníkový automat.
- Může uložit první část sekvence do zásobníku. V okamžiku, kdy přečte znak c, přejde do jiného vnitřního stavu a změní své chování. Začne číst znaky ze vstupu a zároveň odebírá znaky ze zásobníku. Kontroluje jejich dodržování. Pokud se znaky shodují, kontrola pokračuje, dokud není zásobník prázdný a současně se nenarazí na konec vstupu. V takovém případě může automat potvrdit symetrii posloupnosti.
- Pokud během porovnávání dojde k neshodě mezi znakem v horní části zásobníku a znakem na vstupu, nebo pokud je zásobník vyprázdněn před ukončením vstupu, nebo pokud vstup skončí dříve, než je zásobník vyprázdněn, asymetrie sekvence je potvrzena.

Zásobníkový automat - Pushdown Automaton (PDA)

- Existují obě varianty zásobníkových automatů – deterministická i nedeterministická.
- Nedeterministický zásobníkový automat přijímá dané slovo do koncovým (finite) stavem, když existuje alespoň jeden možný způsob jeho práce, ve kterém po přečtení celého slova skončí v jednom z koncových stavů.
- Přijímá slovo s prázdným zásobníkem, pokud existuje alespoň jeden možný způsob jeho práce, ve kterém bude zásobník prázdný po přečtení celého slova.

2. Turingův stroj (Turing Machine)

Turing Machine – idea

- Ani zásobníkový automat formálně dobře nepopisuje všechny možnosti, které by počítače teoreticky mohly využít.
- Paměť „zásobníkového počítače“ je konečná a zásobníkový přístup značně omezuje výpočetní možnosti.
- Představte si situaci podobnou té, která je popsána na předchozích slajdech. Musíme sestrojit automat, který rozpozná, zda posloupnost symbolů vypadá takto: Nejprve obsahuje posloupnost nul a jedniček, pak oddělovač c a pak znovu stejnou posloupnost nul a jedniček, tentokrát však ve stejném pořadí jako v první sekvenci.
- Tato nová úloha se tedy od vzorové úlohy používané pro zásobníkové automaty liší pouze tím, že dvě sekvence nul a jedniček by měly být ve stejném pořadí.

Turing Machine – idea

- Z pohledu možnosti řešení tohoto problému pomocí automatu s jedním zásobníkem je najednou vše jinak.
- Samozřejmě při čtení první části sekvence můžeme vše uložit do zásobníku, dokud nepřečteme znak c.
- Nyní následuje první znak druhé části sekvence. Měl by se to porovnat s prvním symbolem první sekvence. Leží však až na samém dně zásobníku. Nelze jej načíst a porovnat se vstupním symbolem (právě přečteném), aniž bychom jej postupně „načetli“ ze zásobníku, a tím zapomněli všechny ostatní znaky (nad ním).
- Úloha je tedy s pomocí zásobníkového automatu (s jedním zásobníkem) neřešitelná. Jazyk takových „dvojitých“ slov není bezkontextový.

Turing Machine – idea

- Ale kdybychom měli dva zásobníky, byli bychom schopni náš problém vyřešit.
- Při čtení prvního dílu uložíme znaky do zásobníku-1.
- Přečteme znak c a poté přesuneme celý obsah zásobníku-1 symbol po symbolu do zásobníku-2. Tím se obrátí pořadí znaků v zásobníku. Co bylo dole, bude nahoře a naopak.
- Při čtení druhé části vstupního slova stačí porovnat symbol po symbolu s obsahem zásobníku-2.
- Vidíme tedy, že pokud povolíme použití dvou zásobníků jako paměti místo jednoho, možnosti automatu se významně zvýší.

Turing Machine – idea

- Poněkud složitější je tvrdit, že žádné další přidávání paměti (stacků) kvalitu automatu nezvýší.
- Jinými slovy, třída úloh, které může vyřešit automat se dvěma zásobníky, je stejná jako třída úloh, které může vyřešit automat s jinak organizovanou pamětí.
- Pravděpodobně však někdy s výrazně menší účinností, ale pokud zkoumáme pouze to, že problém lze vyřešit, není pro nás účinnost výpočtu podstatná.
- Právě zmíněný výrok se v poněkud pozměněné podobě nazývá Churchův teorém.

Turing Machine – idea

- Zásobníkový automat se dvěma zásobníky by se tedy dal považovat za vhodný formální model počítače, pokud chceme prozkoumat jeho výpočetní možnosti.
- Obvykle se však používá jiný, ekvivalentní (tj. stejně výkonný) model, zvaný Turingův stroj. Autorem tohoto modelu je anglický matematik a formalista Alan Turing (* 1912, Londýn, † 1954).
- *Turingův stroj je konečný automat doplněný o paměť ve formě pásky. Na pásku jsou zapisovány symboly tzv. páskové abecedy. Páska je potenciálně nekonečná na obou stranách, i když samozřejmě popisovaná a použitá část pásky je vždy pouze konečná (po konečném počtu kroků). Turingův stroj čte pásku a zapisuje na pásku pomocí čtecí a zapisovací hlavy, která se může po pásmu pohybovat libovolně v každém směru o jednu pozici, buď doprava nebo doleva, nebo zůstat na místě.*

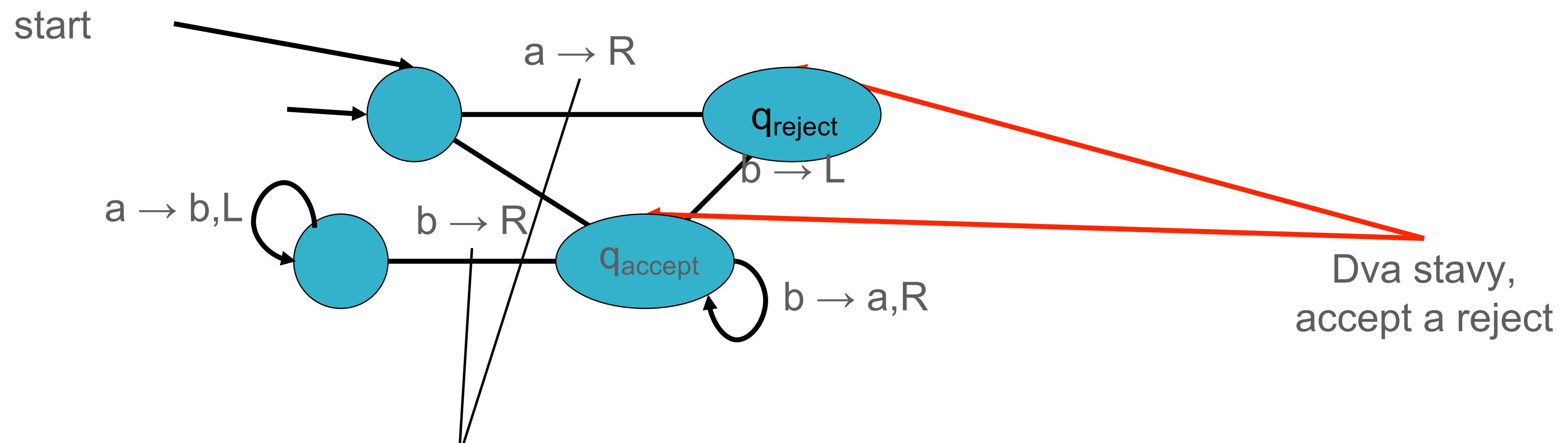
Turing Machine – idea

- Co je vlastně Turingův stroj?
- V počátečním nástinu (automatického) a-stroje (dále Turingův stroj nebo prostě jen stroj) Turing porovnává počítajícího člověka se strojem, který může mít konečný počet stavů. Turing tyto stavy označuje jako m-konfigurace (nicméně pojednání m-konfigurace je komplexnější).
- V Turingově podání (a zůstalo to dodnes) je stroj zařízení, které má čtecí hlavu, nekonečnou pásku a tabulku chování, která určuje chování stroje při čtení aktuálního symbolu z pásky a m-konfiguraci (trojici: stavu, obsahu pásky a pozici hlavy), ve které se stroj právě nachází. Symbol načtený z pásky a aktuální m-konfigurace se nazývá konfigurace Turingova stroje. V závislosti na konfiguraci pak stroj přepíše pole nebo posune pásku do stran. Turing také zavádí vyjádření kompletní konfigurace dané fáze, která je dána číslem čteného pole, m-konfigurací a posloupností všech symbolů zapsaných na pásku. Změny stroje a pásky určené po sobě jdoucími kompletními konfiguracemi se nazývají kroky stroje.

Turing Machine – idea

- „A Turing machine is a finite-state machine associated with an external storage or memory medium.“
- „A Turing machine is essentially a finite-state sequential machine that has the ability to communicate with an external store of information“.

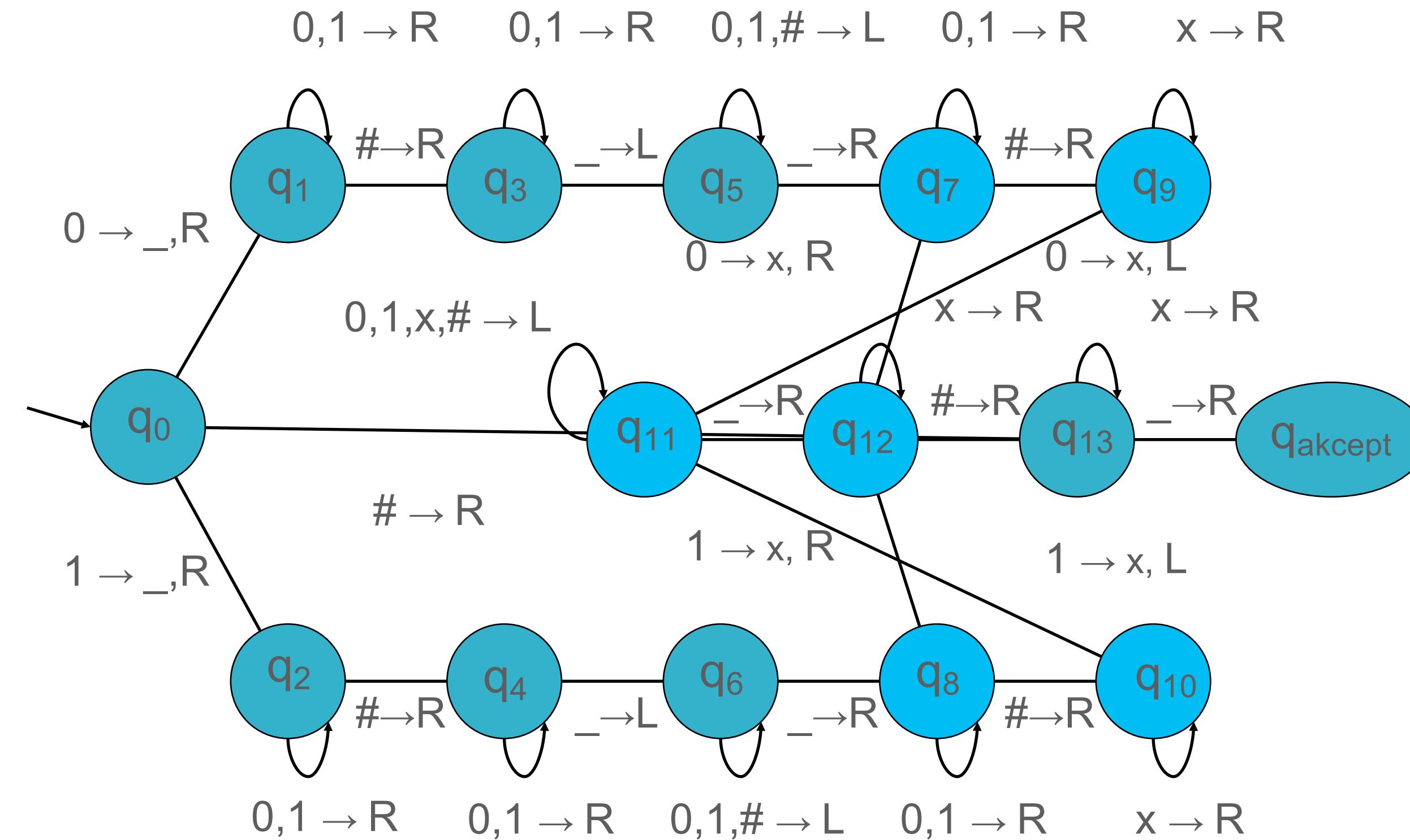
Turing Machine – idea



Přechody mezi stavy: (čtení z pásky → zápis na pásku, posun)

- $a \rightarrow R$ představuje přečti **a**, a posuň se doprava
 - $a \rightarrow L$ představuje přečti **a**, a posuň se doleva
 - $a \rightarrow b$, R představuje přečti **a**, zapiš **b**, a posuň se doprava

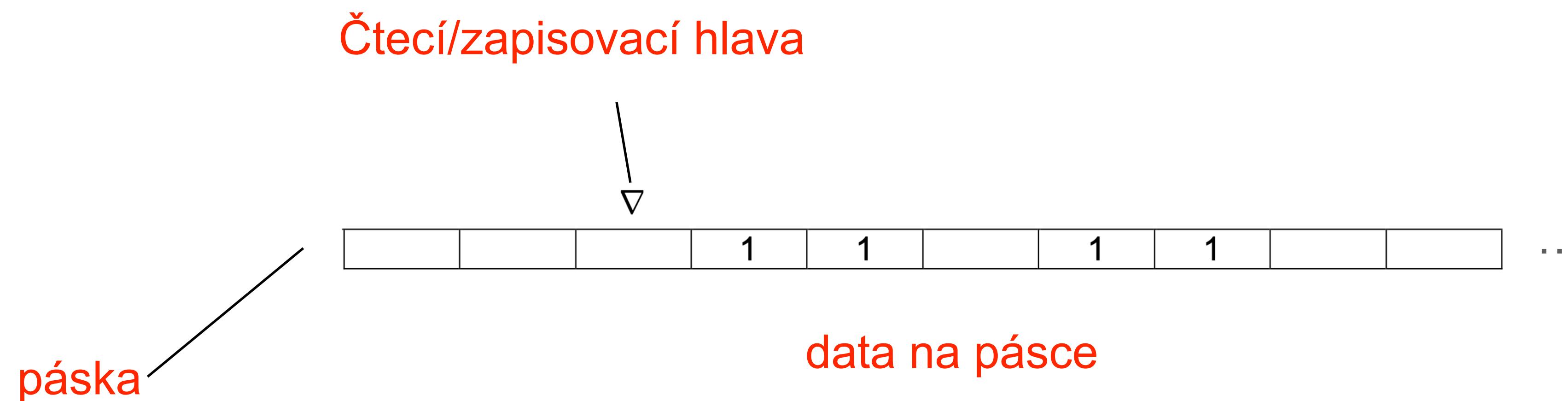
Turing Machine – idea (trošku složitější)



2. Turing Machine – Příklad

Turing Machine – příklad

- Než přejdeme k rigorózní definici stroje, ukažme si triviální příklad toho, jak to funguje.
- Cílem stroje je sečíst dvě malá čísla, řekněme 2 plus 2. Sčítání bude probíhat v „unárním systému“ (2 se bude psát jako 11, 3 jako 111 atd.).
- Na pásce stroje budou zapsána čísla, jak je znázorněno na obrázku níže.
- Tabulka jeho chování je uvedena na dalším slajdu.



Turing Machine – příklad

				1	1		1	1		
	m-konfigurace	symbol		Přechod (akce)		Nová m-konfigurace				
1	A	-		Posuň pásku o jednu pozici doprava.		A				
2	A	1		Posuň pásku o jednu pozici doprava.		B				
3	B	1		Posuň pásku o jednu pozici doprava.		B				
4	B	-		PRINT 1 a posuň pásku o jednu pozici doprava.		C				
5	C	1		Posuň pásku o jednu pozici doprava.		C				
6	C	-		Posuň pásku o jednu pozici doleva.		D				
7	D	1		ERASE 1 a žádný posun		D				
8	D	-		STOP		D				

Princip fungování stroje je jednoduchý. Jak je znázorněno, čtecí hlava s konfigurací A začíná na prázdné pozici (rádek 1). Páska se posune o jedno pole doprava. Stroj je ve stavu A a čte 1. To odpovídá akci na řádku 2 tabulky - páška se opět posune o jedno pole doprava a přečtením vstupního symbolu 1 se stroj dostal do stavu B. Přečtením 1 (rádek 3) se čtecí hlava se opět posune o jednu pozici doprava - stroj zůstane ve stavu B. V dalším kroku je aktuálně načtené pole prázdné a je tedy přepsáno o jedničku. A tak dále, dokud stroj nevymaže poslední nadbytečný symbol 1 a nezastaví se.

Turing Machine – příklad

				1	1		1	1		
	m-konfigurace	symbol			Přechod (akce)		Nová m-konfigurace			
1	A	-			Posuň pásku o jednu pozici doprava.		A			
2	A	1			Posuň pásku o jednu pozici doprava.		B			
3	B	1			Posuň pásku o jednu pozici doprava.		B			
4	B	-			PRINT 1 a posuň pásku o jednu pozici doprava.		C			
5	C	1			Posuň pásku o jednu pozici doprava.		C			
6	C	-			Posuň pásku o jednu pozici doleva.		D			
7	D	1			ERASE 1 a žádný posun		D			
8	D	-			STOP		D			

Princip fungování stroje je jednoduchý. Jak je znázorněno, čtecí hlava s konfigurací A začíná na prázdné pozici (rádek 1). Páska se posune o jedno pole doprava. Stroj je ve stavu A a čte 1. To odpovídá akci na řádku 2 tabulky - páška se opět posune o jedno pole doprava a přečtením vstupního symbolu 1 se stroj dostal do stavu B. Přečtením 1 (rádek 3) se čtecí hlava se opět posune o jednu pozici doprava - stroj zůstane ve stavu B. V dalším kroku je aktuálně načtené pole prázdné a je tedy přepsáno o jedničku. A tak dále, dokud stroj nevymaže poslední nadbytečný symbol 1 a nezastaví se.

Turing Machine – příklad

				1	1		1	1		
	m-konfigurace	symbol		Přechod (akce)					Nová m-konfigurace	
1	A	-		Posuň pásku o jednu pozici doprava.					A	
2	A	1		Posuň pásku o jednu pozici doprava.					B	
3	B	1		Posuň pásku o jednu pozici doprava.					B	
4	B	-		PRINT 1 a posuň pásku o jednu pozici doprava.					C	
5	C	1		Posuň pásku o jednu pozici doprava.					C	
6	C	-		Posuň pásku o jednu pozici doleva.					D	
7	D	1		ERASE 1 a žádný posun					D	
8	D	-		STOP					D	

Princip fungování stroje je jednoduchý. Jak je znázorněno, čtecí hlava s konfigurací A začíná na prázdné pozici (rádek 1). Páska se posune o jedno pole doprava. Stroj je ve stavu A a čte 1. To odpovídá akci na řádku 2 tabulky - páška se opět posune o jedno pole doprava a přečtením vstupního symbolu 1 se stroj dostal do stavu B. Přečtením 1 (rádek 3) se čtecí hlava se opět posune o jednu pozici doprava - stroj zůstane ve stavu B. V dalším kroku je aktuálně načtené pole prázdné a je tedy přepsáno o jedničku. A tak dále, dokud stroj nevymaže poslední nadbytečný symbol 1 a nezastaví se.

Turing Machine – příklad

			1	1		1	1		
--	--	--	---	---	--	---	---	--	--

	m-konfigurace	symbol	Přechod (akce)	Nová m-konfigurace
1	A	-	Posuň pásku o jednu pozici doprava.	A
2	A	1	Posuň pásku o jednu pozici doprava.	B
3	B	1	Posuň pásku o jednu pozici doprava.	B
4	B	-	PRINT 1 a posuň pásku o jednu pozici doprava.	C
5	C	1	Posuň pásku o jednu pozici doprava.	C
6	C	-	Posuň pásku o jednu pozici doleva.	D
7	D	1	ERASE 1 a žádný posun	D
8	D	-	STOP	D

Princip fungování stroje je jednoduchý. Jak je znázorněno, čtecí hlava s konfigurací A začíná na prázdné pozici (rádek 1). Páska se posune o jedno pole doprava. Stroj je ve stavu A a čte 1. To odpovídá akci na řádku 2 tabulky - páška se opět posune o jedno pole doprava a přečtením vstupního symbolu 1 se stroj dostal do stavu B. Přečtením 1 (rádek 3) se čtecí hlava se opět posune o jednu pozici doprava - stroj zůstane ve stavu B. V dalším kroku je aktuálně načtené pole prázdné a je tedy přepsáno o jedničku. A tak dále, dokud stroj nevymaže poslední nadbytečný symbol 1 a nezastaví se.

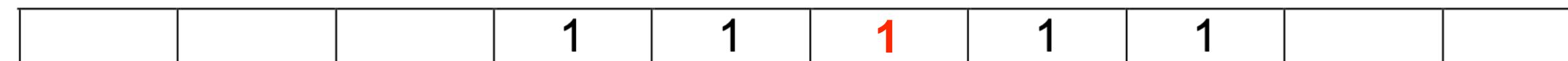
Turing Machine – příklad

			1	1	1	1	1		
--	--	--	---	---	---	---	---	--	--

	m-konfigurace	symbol	Přechod (akce)	Nová m-konfigurace
1	A	-	Posuň pásku o jednu pozici doprava.	A
2	A	1	Posuň pásku o jednu pozici doprava.	B
3	B	1	Posuň pásku o jednu pozici doprava.	B
4	B	-	PRINT 1 a posuň pásku o jednu pozici doprava.	C
5	C	1	Posuň pásku o jednu pozici doprava.	C
6	C	-	Posuň pásku o jednu pozici doleva.	D
7	D	1	ERASE 1 a žádný posun	D
8	D	-	STOP	D

Princip fungování stroje je jednoduchý. Jak je znázorněno, čtecí hlava s konfigurací A začíná na prázdné pozici (rádek 1). Páska se posune o jedno pole doprava. Stroj je ve stavu A a čte 1. To odpovídá akci na řádku 2 tabulky - páška se opět posune o jedno pole doprava a přečtením vstupního symbolu 1 se stroj dostal do stavu B. Přečtením 1 (rádek 3) se čtecí hlava se opět posune o jednu pozici doprava - stroj zůstane ve stavu B. V dalším kroku je aktuálně načtené pole prázdné a je tedy přepsáno o jedničku. A tak dále, dokud stroj nevymaže poslední nadbytečný symbol 1 a nezastaví se.

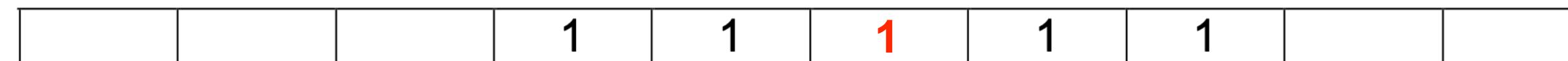
Turing Machine – příklad



	m-konfigurace	symbol	Přechod (akce)	Nová m-konfigurace
1	A	-	Posuň pásku o jednu pozici doprava.	A
2	A	1	Posuň pásku o jednu pozici doprava.	B
3	B	1	Posuň pásku o jednu pozici doprava.	B
4	B	-	PRINT 1 a posuň pásku o jednu pozici doprava.	C
5	C	1	Posuň pásku o jednu pozici doprava.	C
6	C	-	Posuň pásku o jednu pozici doleva.	D
7	D	1	ERASE 1 a žádný posun	D
8	D	-	STOP	D

Princip fungování stroje je jednoduchý. Jak je znázorněno, čtecí hlava s konfigurací A začíná na prázdné pozici (rádek 1). Páska se posune o jedno pole doprava. Stroj je ve stavu A a čte 1. To odpovídá akci na řádku 2 tabulky - páška se opět posune o jedno pole doprava a přečtením vstupního symbolu 1 se stroj dostal do stavu B. Přečtením 1 (rádek 3) se čtecí hlava se opět posune o jednu pozici doprava - stroj zůstane ve stavu B. V dalším kroku je aktuálně načtené pole prázdné a je tedy přepsáno o jedničku. A tak dále, dokud stroj nevymaže poslední nadbytečný symbol 1 a nezastaví se.

Turing Machine – příklad



	m-konfigurace	symbol	Přechod (akce)	Nová m-konfigurace
1	A	-	Posuň pásku o jednu pozici doprava.	A
2	A	1	Posuň pásku o jednu pozici doprava.	B
3	B	1	Posuň pásku o jednu pozici doprava.	B
4	B	-	PRINT 1 a posuň pásku o jednu pozici doprava.	C
5	C	1	Posuň pásku o jednu pozici doprava.	C
6	C	-	Posuň pásku o jednu pozici doleva.	D
7	D	1	ERASE 1 a žádný posun	D
8	D	-	STOP	D

Princip fungování stroje je jednoduchý. Jak je znázorněno, čtecí hlava s konfigurací A začíná na prázdné pozici (rádek 1). Páska se posune o jedno pole doprava. Stroj je ve stavu A a čte 1. To odpovídá akci na řádku 2 tabulky - páska se opět posune o jedno pole doprava a přečtením vstupního symbolu 1 se stroj dostal do stavu B. Přečtením 1 (rádek 3) se čtecí hlava se opět posune o jednu pozici doprava - stroj zůstane ve stavu B. V dalším kroku je aktuálně načtené pole prázdné a je tedy přepsáno o jedničku. A tak dále, dokud stroj nevymaže poslední nadbytečný symbol 1 a nezastaví se.

Turing Machine – příklad



	m-konfigurace	symbol	Přechod (akce)	Nová m-konfigurace
1	A	-	Posuň pásku o jednu pozici doprava.	A
2	A	1	Posuň pásku o jednu pozici doprava.	B
3	B	1	Posuň pásku o jednu pozici doprava.	B
4	B	-	PRINT 1 a posuň pásku o jednu pozici doprava.	C
5	C	1	Posuň pásku o jednu pozici doprava.	C
6	C	-	Posuň pásku o jednu pozici doleva.	D
7	D	1	ERASE 1 a žádný posun	D
8	D	-	STOP	D

Princip fungování stroje je jednoduchý. Jak je znázorněno, čtecí hlava s konfigurací A začíná na prázdné pozici (rádek 1). Páska se posune o jedno pole doprava. Stroj je ve stavu A a čte 1. To odpovídá akci na řádku 2 tabulky - páska se opět posune o jedno pole doprava a přečtením vstupního symbolu 1 se stroj dostal do stavu B. Přečtením 1 (rádek 3) se čtecí hlava se opět posune o jednu pozici doprava - stroj zůstane ve stavu B. V dalším kroku je aktuálně načtené pole prázdné a je tedy přepsáno o jedničku. A tak dále, dokud stroj nevymaže poslední nadbytečný symbol 1 a nezastaví se.

Turing Machine – příklad



	m-konfigurace	symbol	Přechod (akce)	Nová m-konfigurace
1	A	-	Posuň pásku o jednu pozici doprava.	A
2	A	1	Posuň pásku o jednu pozici doprava.	B
3	B	1	Posuň pásku o jednu pozici doprava.	B
4	B	-	PRINT 1 a posuň pásku o jednu pozici doprava.	C
5	C	1	Posuň pásku o jednu pozici doprava.	C
6	C	-	Posuň pásku o jednu pozici doleva.	D
7	D	1	ERASE 1 a žádný posun	D
8	D	-	STOP	D

Princip fungování stroje je jednoduchý. Jak je znázorněno, čtecí hlava s konfigurací A začíná na prázdné pozici (rádek 1). Páska se posune o jedno pole doprava. Stroj je ve stavu A a čte 1. To odpovídá akci na řádku 2 tabulky - páška se opět posune o jedno pole doprava a přečtením vstupního symbolu 1 se stroj dostal do stavu B. Přečtením 1 (rádek 3) se čtecí hlava se opět posune o jednu pozici doprava - stroj zůstane ve stavu B. V dalším kroku je aktuálně načtené pole prázdné a je tedy přepsáno o jedničku. A tak dále, dokud stroj nevymaže poslední nadbytečný symbol 1 a nezastaví se.

Turing Machine – příklad

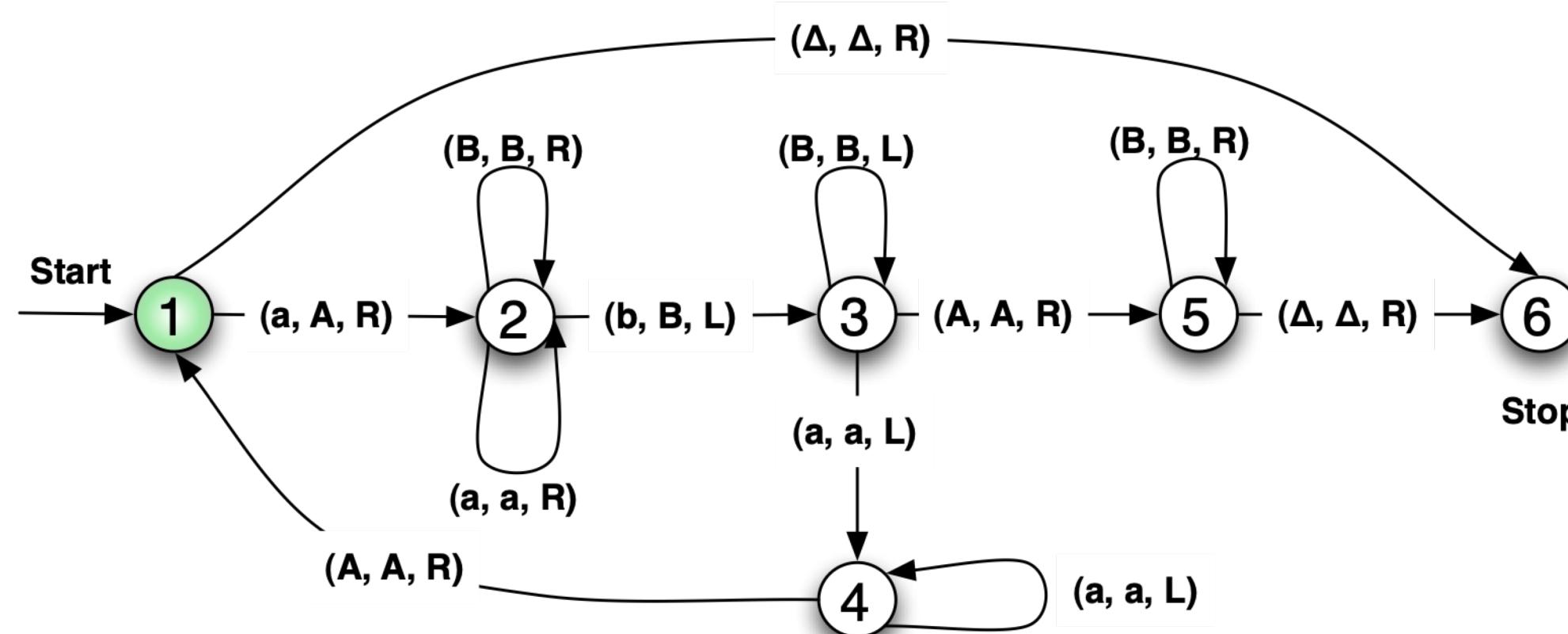


	m-konfigurace	symbol	Přechod (akce)	Nová m-konfigurace
1	A	-	Posuň pásku o jednu pozici doprava.	A
2	A	1	Posuň pásku o jednu pozici doprava.	B
3	B	1	Posuň pásku o jednu pozici doprava.	B
4	B	-	PRINT 1 a posuň pásku o jednu pozici doprava.	C
5	C	1	Posuň pásku o jednu pozici doprava.	C
6	C	-	Posuň pásku o jednu pozici doleva.	D
7	D	1	ERASE 1 a žádný posun	D
8	D	-	STOP	D

Princip fungování stroje je jednoduchý. Jak je znázorněno, čtecí hlava s konfigurací A začíná na prázdné pozici (rádek 1). Páska se posune o jedno pole doprava. Stroj je ve stavu A a čte 1. To odpovídá akci na řádku 2 tabulky - páška se opět posune o jedno pole doprava a přečtením vstupního symbolu 1 se stroj dostal do stavu B. Přečtením 1 (rádek 3) se čtecí hlava se opět posune o jednu pozici doprava - stroj zůstane ve stavu B. V dalším kroku je aktuálně načtené pole prázdné a je tedy přepsáno o jedničku. A tak dále, dokud stroj nevymaže poslední nadbytečný symbol 1 a nezastaví se.

Turing Machine – příklad II

TS akceptuje $a^n b^n$ pro $n \geq 0$

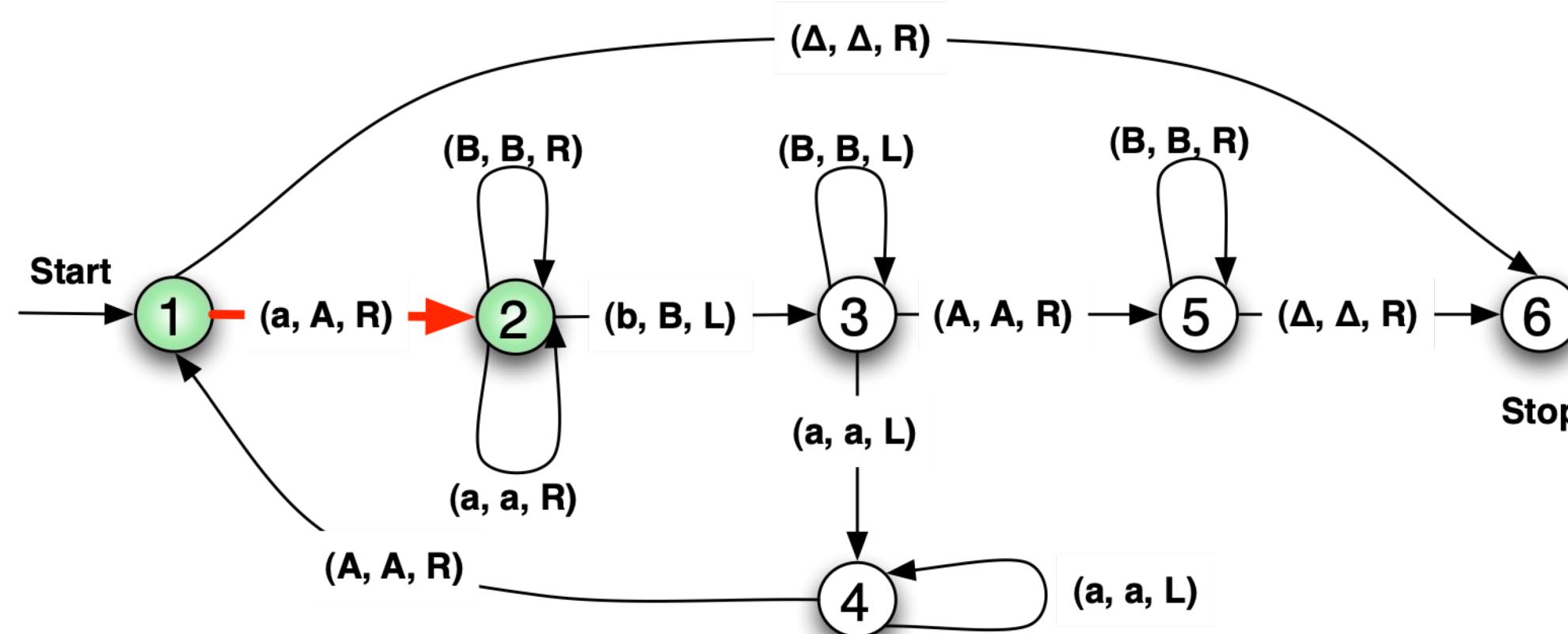


aabb Δ



Turing Machine – příklad II

TS akceptuje $a^n b^n$ pro $n \geq 0$

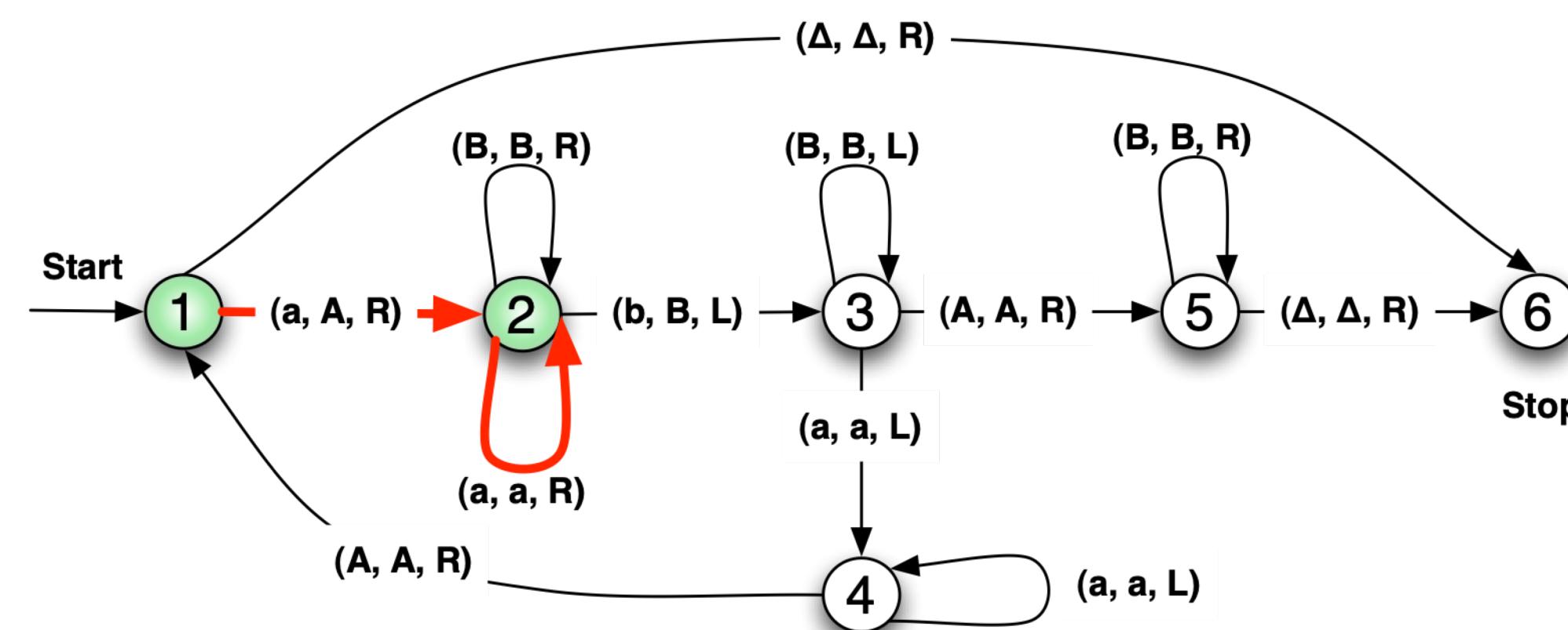


Aabb Δ



Turing Machine – příklad II

TS akceptuje $a^n b^n$ pro $n \geq 0$

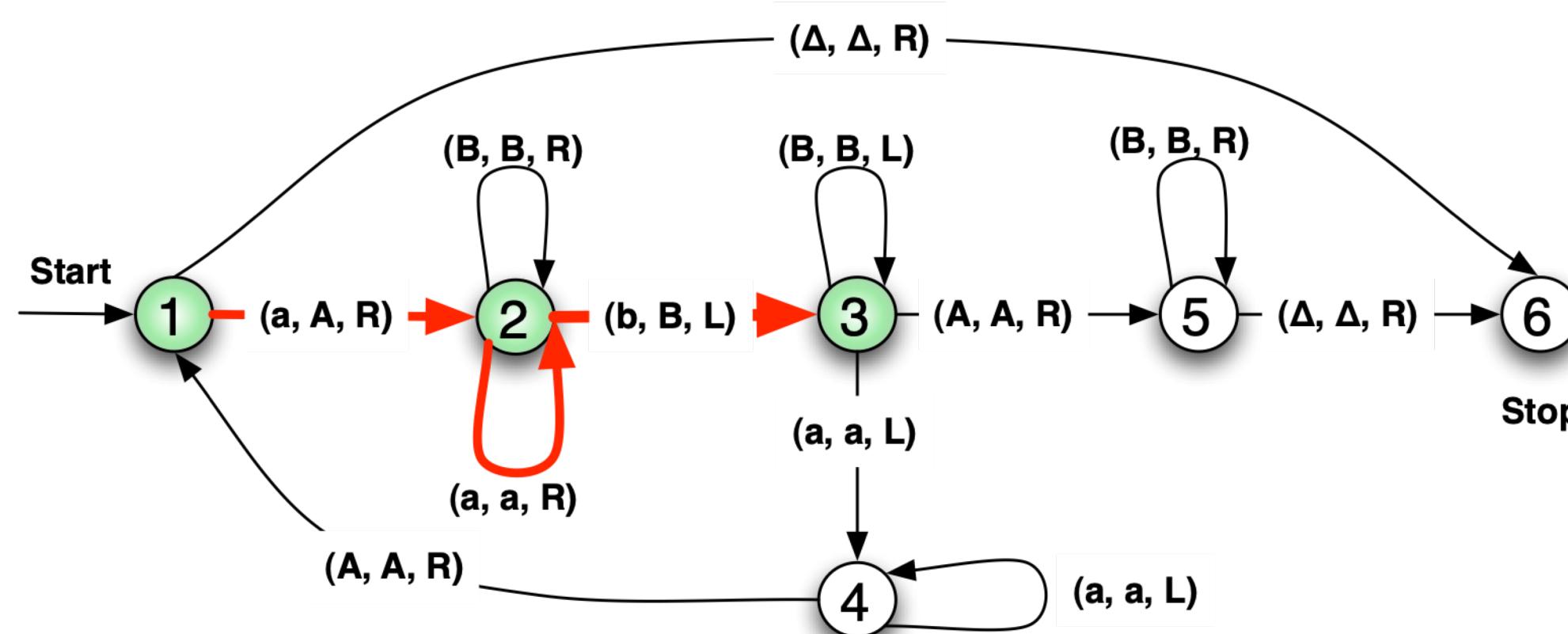


Aabb Δ



Turing Machine – příklad II

TS akceptuje $a^n b^n$ pro $n \geq 0$

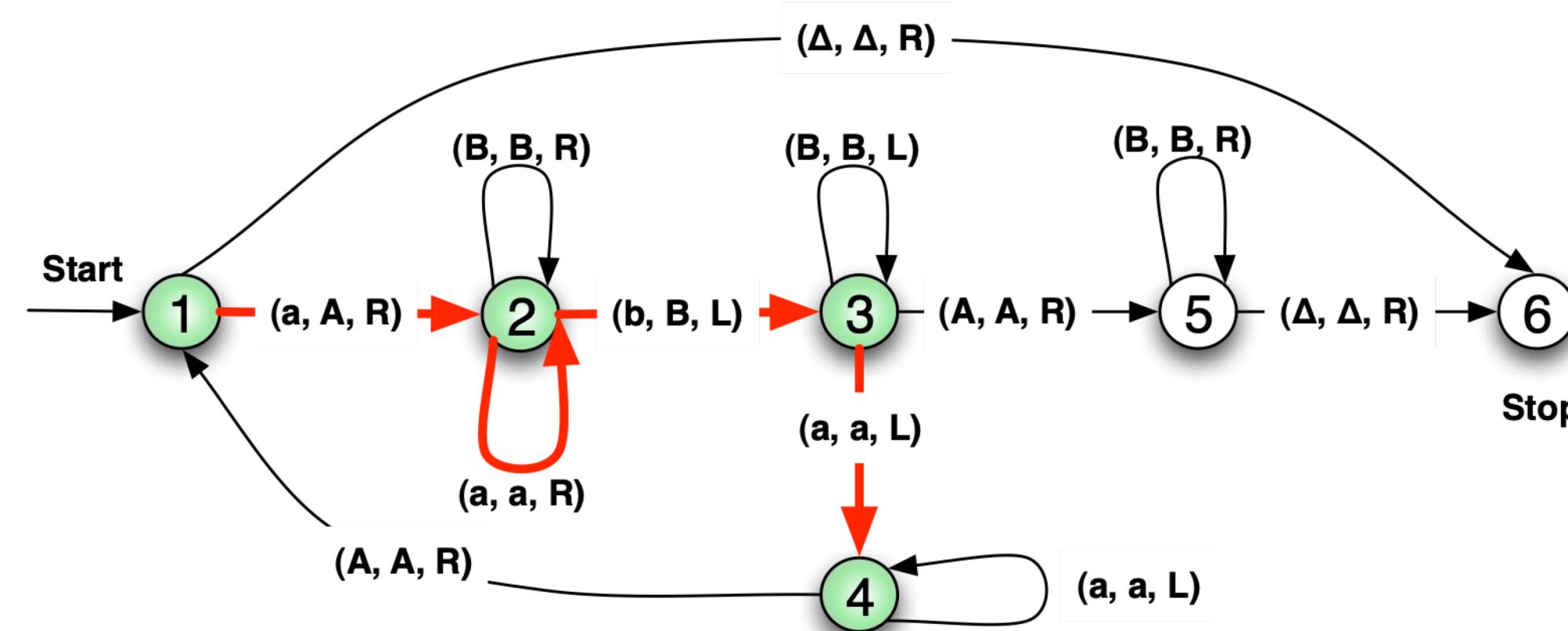


AaBbΔ



Turing Machine – příklad II

TS akceptuje $a^n b^n$ pro $n \geq 0$

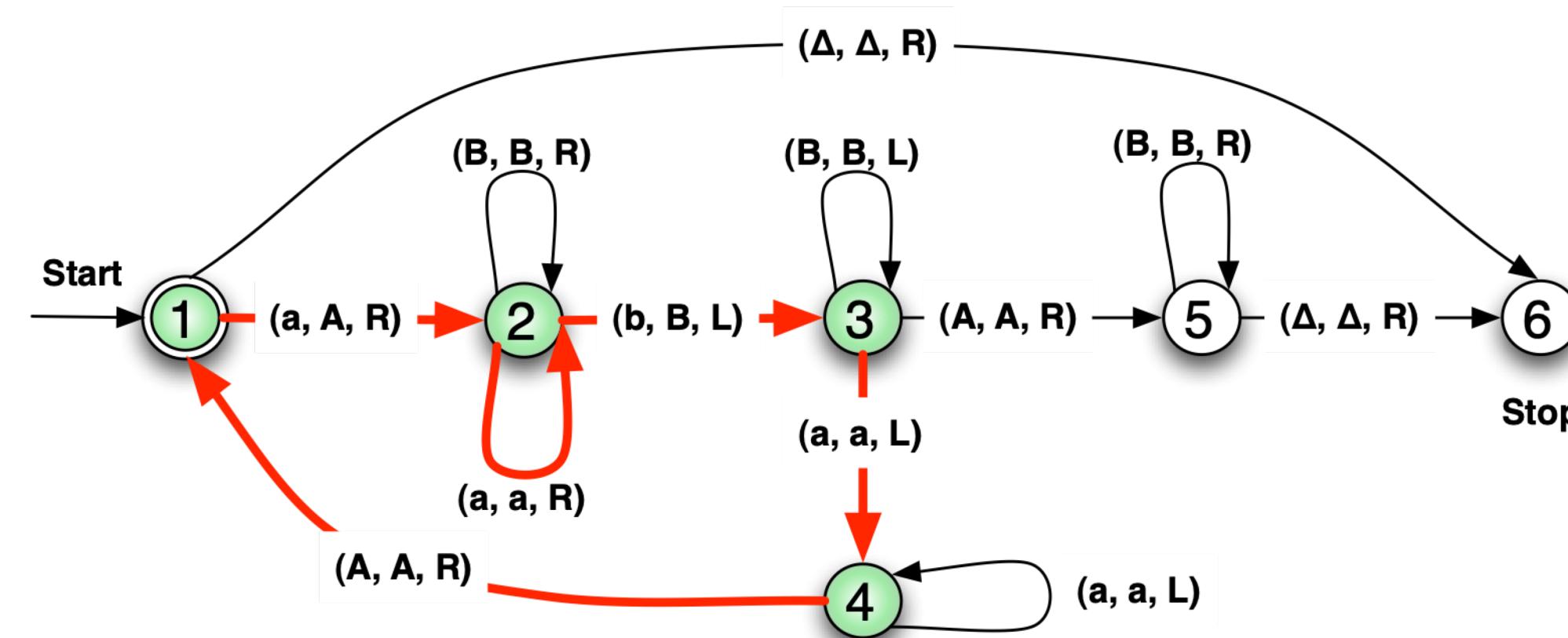


AaBb Δ



Turing Machine – příklad II

TS akceptuje $a^n b^n$ pro $n \geq 0$

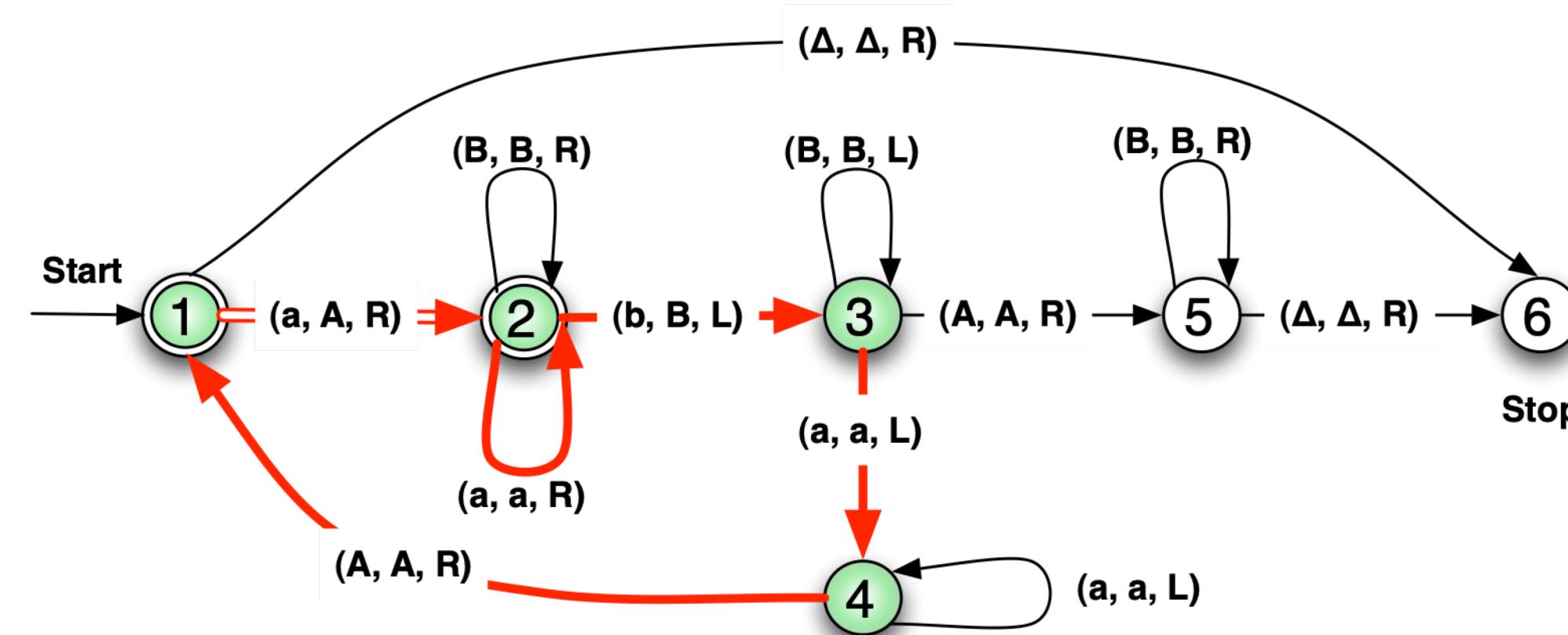


AaBb Δ



Turing Machine – příklad II

TS akceptuje $a^n b^n$ pro $n \geq 0$

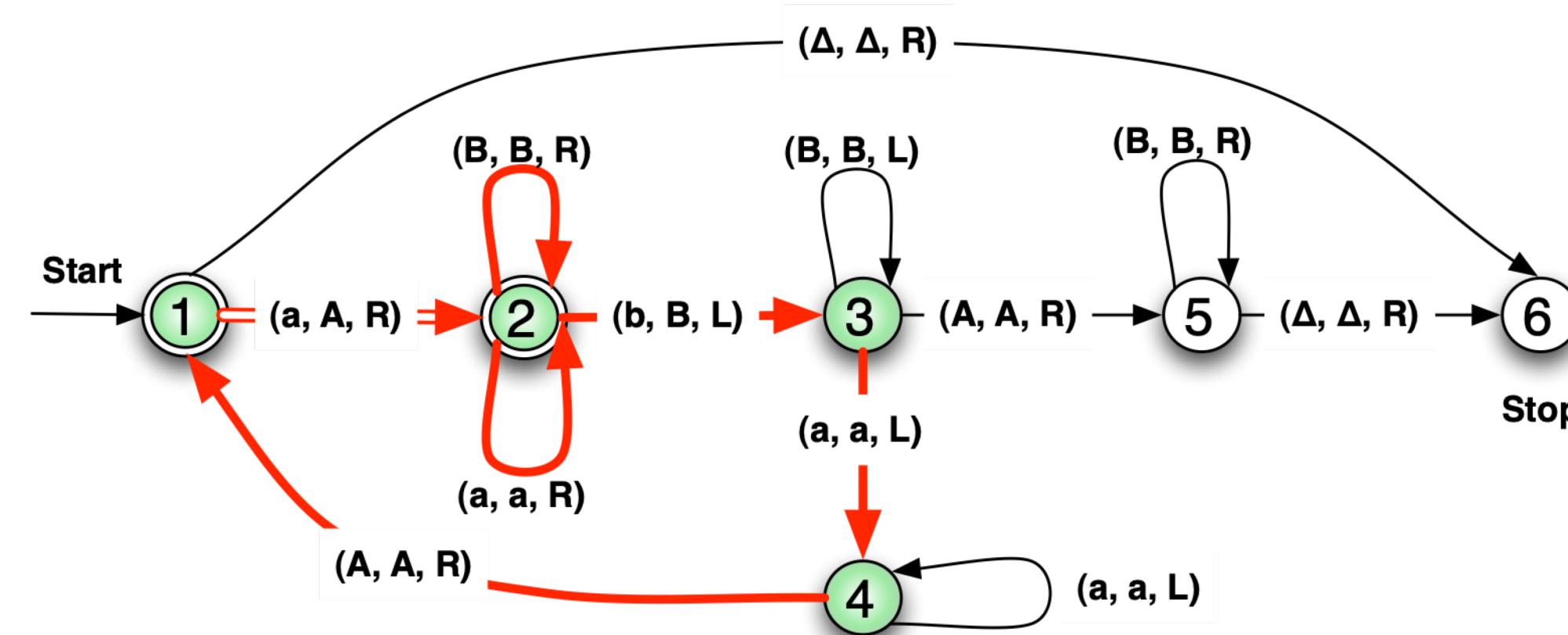


AABb Δ



Turing Machine – příklad II

TS akceptuje $a^n b^n$ pro $n \geq 0$

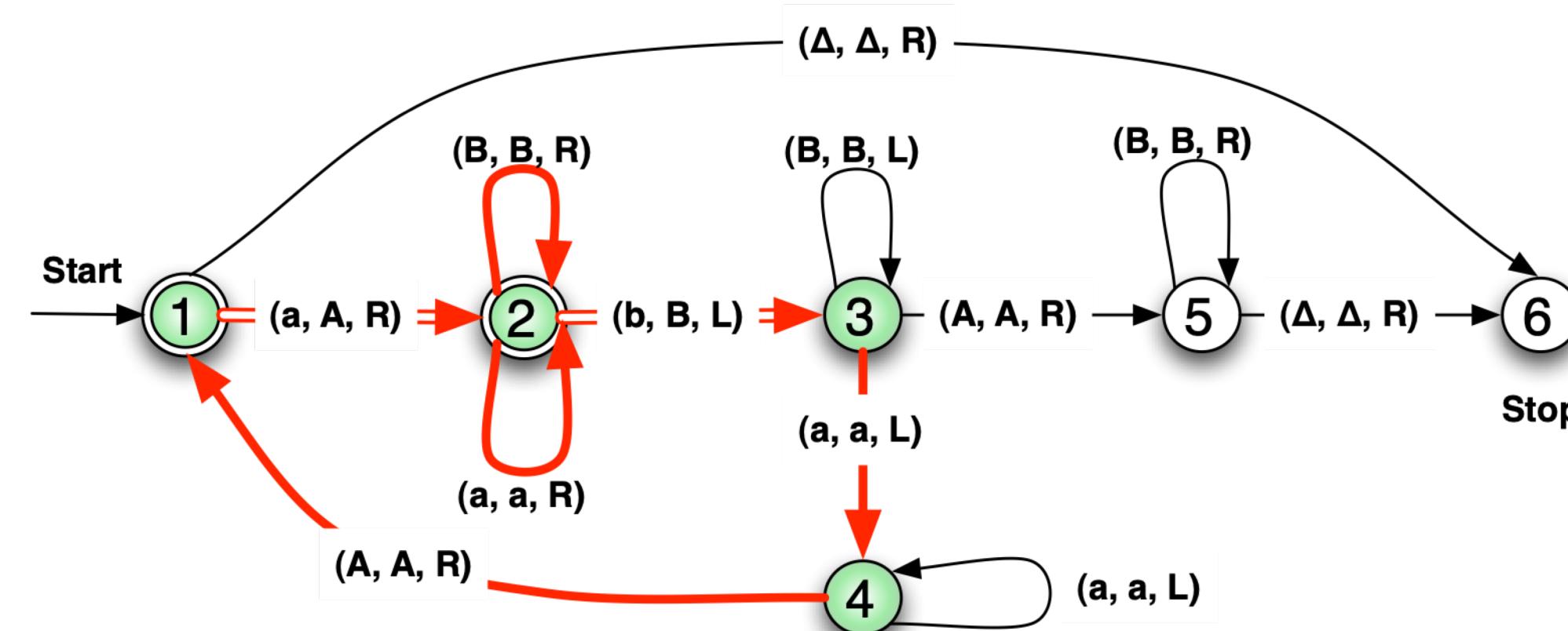


AABb Δ



Turing Machine – příklad II

TS akceptuje $a^n b^n$ pro $n \geq 0$

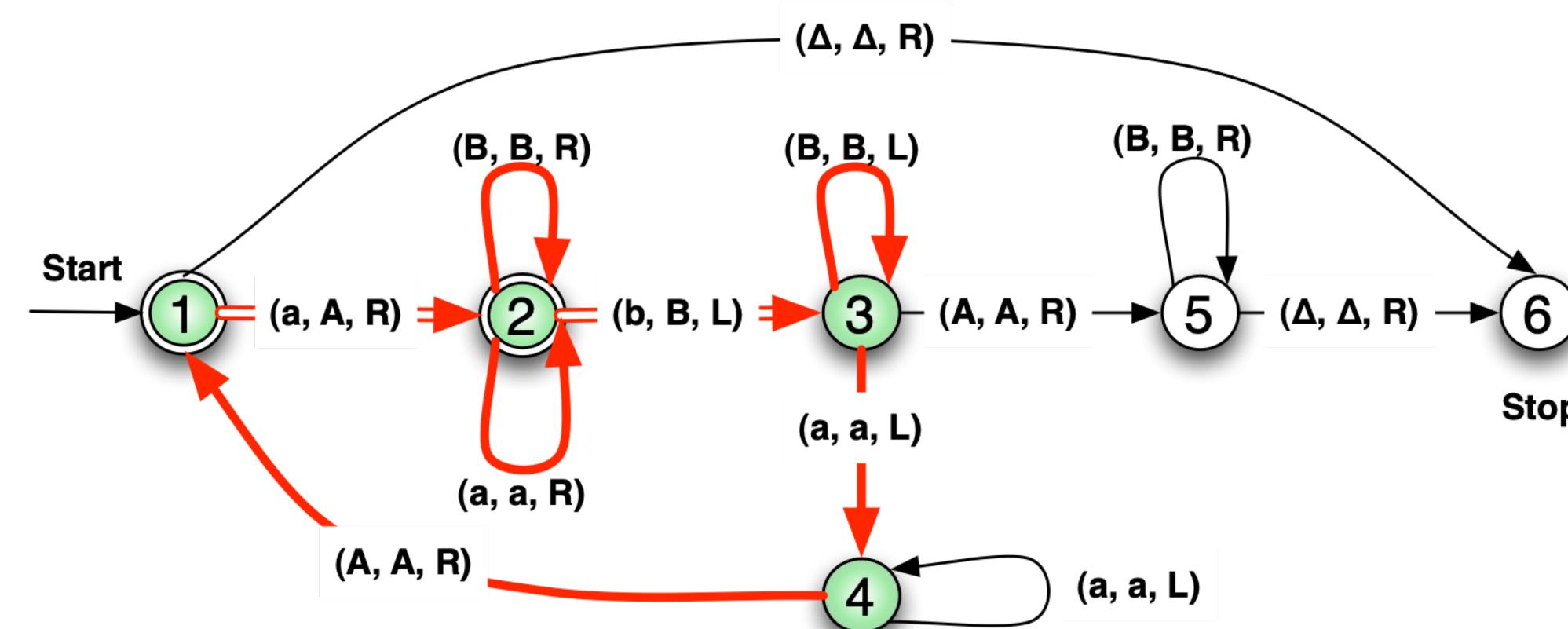


AABBA Δ



Turing Machine – příklad II

TS akceptuje $a^n b^n$ pro $n \geq 0$

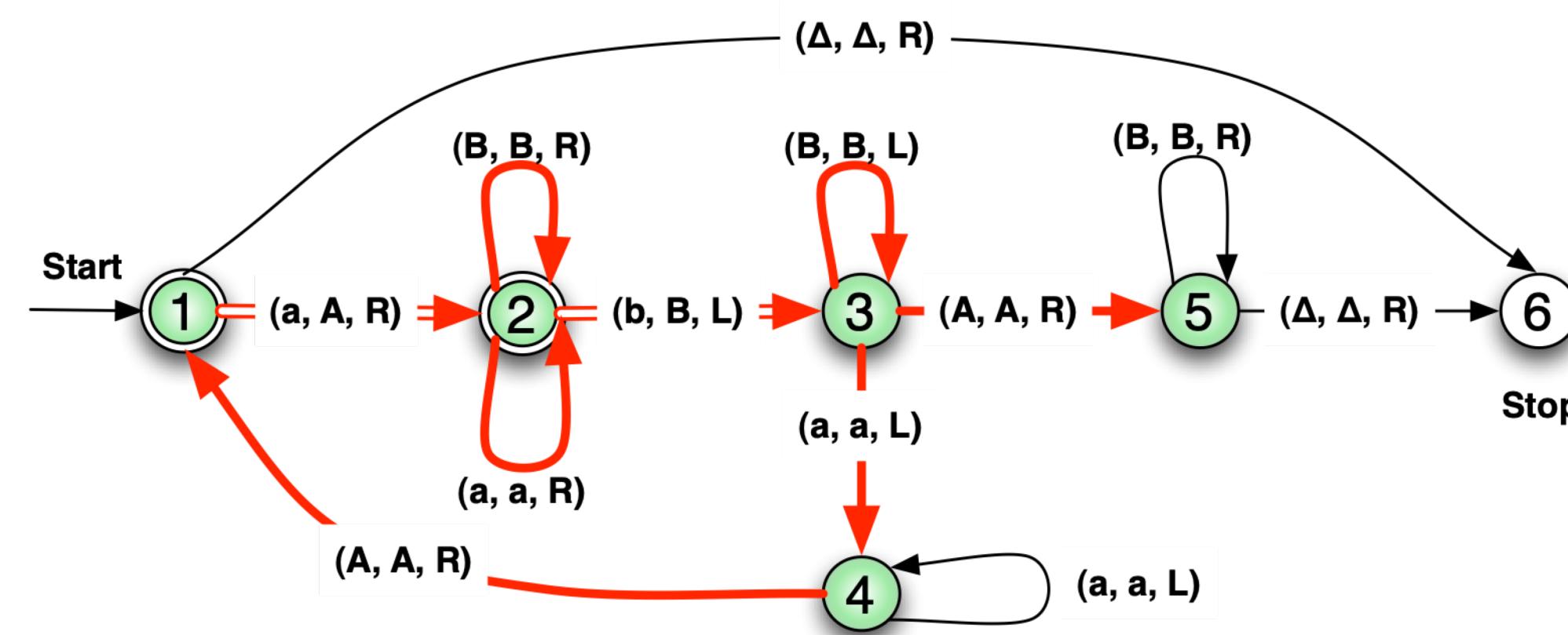


AABBA Δ



Turing Machine – příklad II

TS akceptuje $a^n b^n$ pro $n \geq 0$

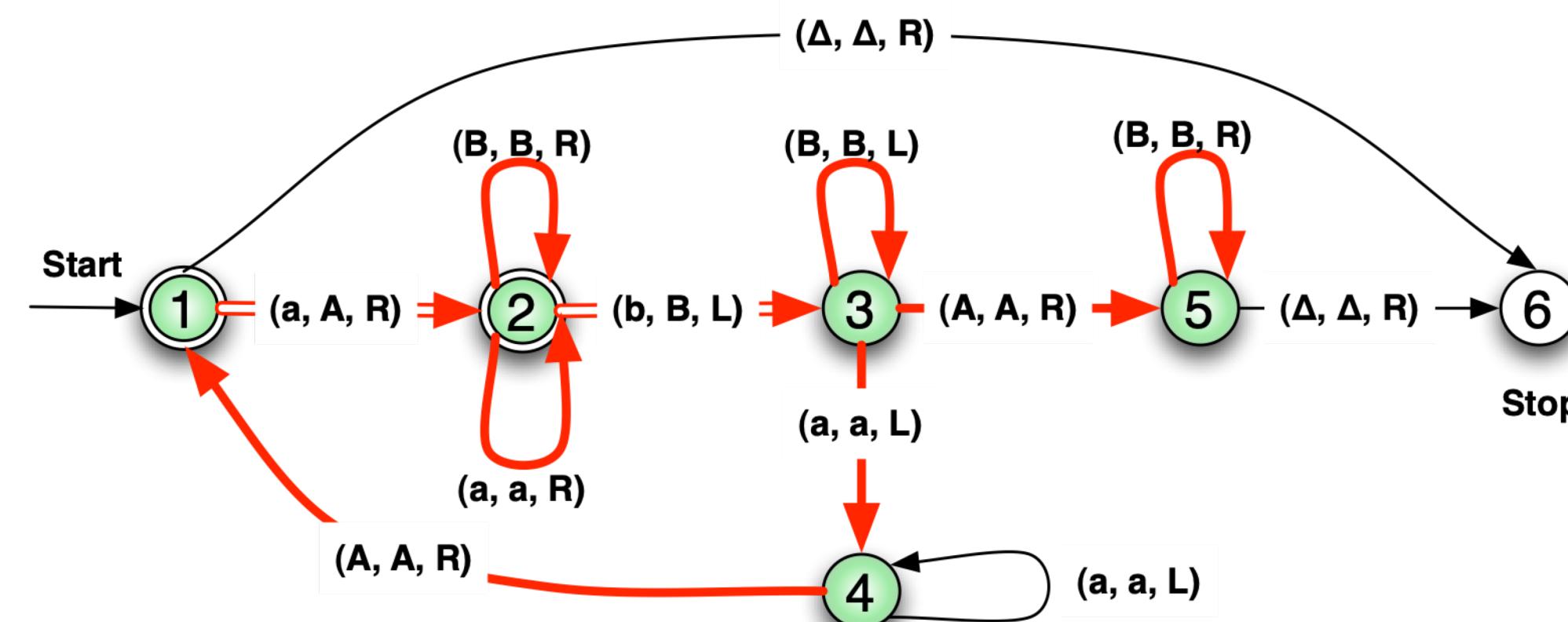


AABBA Δ



Turing Machine – příklad II

TS akceptuje $a^n b^n$ pro $n \geq 0$

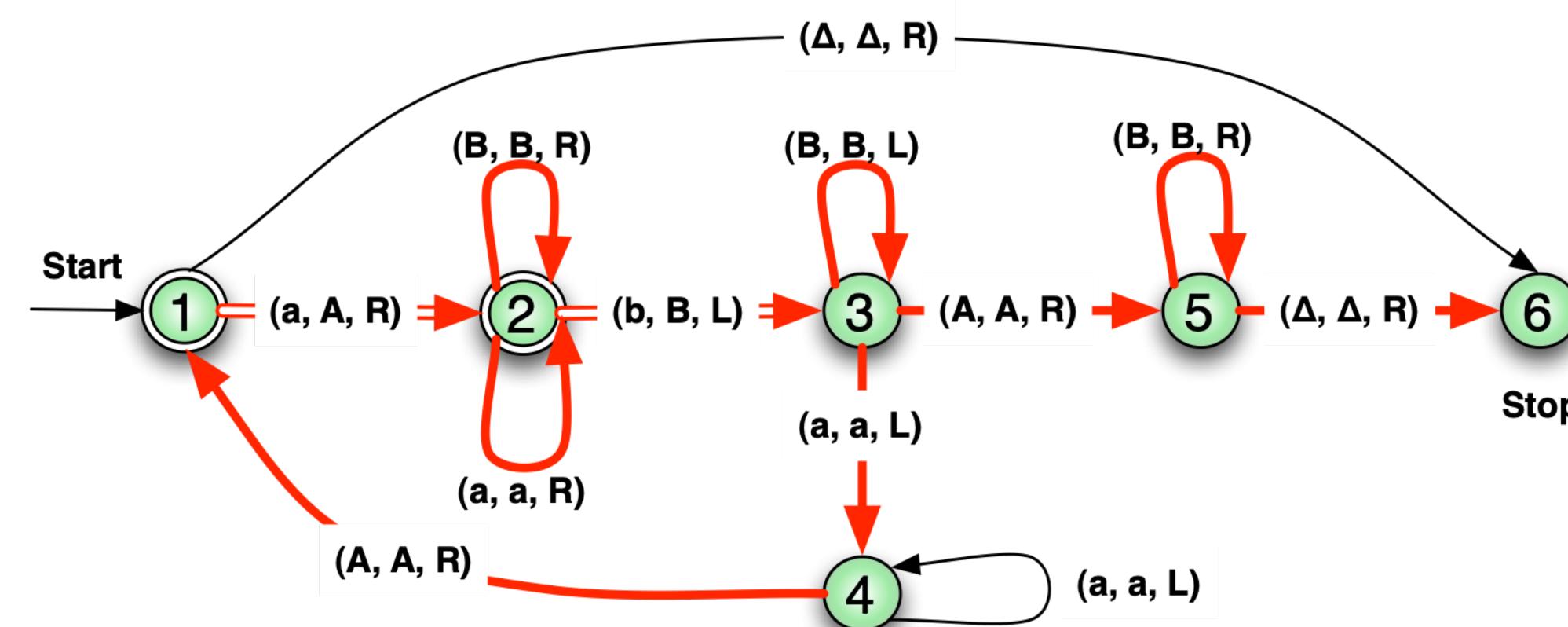


AABBA Δ



Turing Machine – příklad II

TS akceptuje $a^n b^n$ pro $n \geq 0$



AABBA Δ



Děkuji za pozornost