Finite State Machines (Konečné Automaty)

Assoc. prof. Roman Senkerik

A.I.Lab

Obsah Prezentace

- Úvod do formálních modelů výpočtů
- Formální popis konečných automatů
- Deterministické automaty - příklady
- Nedeterministické automaty

1. Formální modely výpočtů

Formální modely výpočtů

- V počítačové vědě a teorii automatů (strojů) je abstraktní stroj teoretickým modelem počítače, tj. hardwaru i softwaru.
- Abstrakce výpočetních procesů obvykle předpokládá paradigma diskrétního času.

- Ve výpočetní teorii jsou abstraktní stroje často používány v experimentech s výpočetním myšlením (logikou, důkazy) nebo k analýze složitosti algoritmů (teorie výpočetní složitosti).
- Typický abstraktní stroj se skládá z definice z hlediska vstupu, výstupu a sady povolených operací.

Formální modely výpočtů - motivace

- V současné době se používají různé typy počítačů.
- Nejčastěji se setkáváme s počítači s takzvanou von Neumannovou architekturou.
- Ty se skládají z procesoru, hlavní paměti a různých přídavných zařízení pro vstup a výstup dat, do kterých můžeme zahrnout i externí paměti.
- Technické řešení konkrétního počítače lze poměrně dobře popsat, i když takový popis nebude dostatečně jasný.
- ***Nedává nám moc šancí vědět, jaké všechny úkoly lze s tímto strojem vyřešit, ani jak efektivně je bude možné vyřešit.***

Formální modely výpočtů - motivace

- Abychom přesně formulovali problémy, jako jsou např.: co lze vyřešit pomocí počítače, jaké jsou „otázky“, na které nám dokáže zodpovědět, a jaká bude účinnost výpočtu, je nutné mít formální model, který popisuje práci počítače.

- Tento model musí být do značné míry univerzální, aby vždy obsahoval celou sadu počítačů, které mají společné určité podstatné faktory, ale v mnoha detailech se mohou lišit.

- Takových modelů je více, od jednoduchých konečných automatů po relativně komplikovaný Turingův stroj.

2. Konečné automaty (Finite state machines)

Konečné automaty

- Konečné automaty (KA), nebo Finite state machines (FSM) nebo Finite state automata (FSA), nebo Finite Automata (FA)…, jsou teoretickým výpočetním modelem pro studium formálních jazyků.

- Rozpoznávají pouze jazyk typu 3 podle Chomského hierarchie jazyků, tj. regulární jazyk.

- Vyznačují se tím, že mají konečnou sadu stavů a přechod z jednoho stavu do druhého je dán tzv. „reakcí“ na externí vstupy.

Konečné automaty

- Jedním z nejdůležitějších rozdílů mezi třídami konečných automatů je, zda je přechod mezi stavy takzvaně „deterministický“.
- To znamená, že existuje pouze jeden stav, do kterého stroj může přejít.
- Nebo „nedeterministické“, tj. že existuje několik stavů, do kterých může automat vstoupit v jednom časovém okamžiku.
- S deterministickým automatem můžeme definovat jakýkoli jazyk, který lze definovat nedeterministicky, ale přidáním nedeterminismu můžeme výrazně zvýšit efektivitu výpočtu, s formálním popisem aplikace (úkolu).

Konečné automaty

- Chování „stavových“ strojů (konečných automatů) lze pozorovat na mnoha zařízeních
- Takové stroje provádějí předem stanovený sled akcí v závislosti na sledu (vstupních) událostí.

- Jednoduchými příklady jsou prodejní automaty, které vydávají produkty, když je uložena správná kombinace mincí, výtahy, jejichž pořadí zastávek je určeno podlažími požadovanými osobami, semafory, které mění pořadí, když čekají auta, a kombinační zámky, které vyžadují zadání sekvence čísel ve správném pořadí.

Konečný automat - definice

- Konečný automat (KA/FSM) je nejjednodušší z formálních výpočetních modelů
- Má konečný počet stavů a může být v přesně daném čase přesně v jednom z konečného počtu stavů.
- Konečný automat zpracovává posloupnost symbolů vybraných z dané konečné sady, kterou nazýváme abeceda konečného automatu
- Taková sekvence se nazývá vstupní slovo.
- Toto vstupní slovo může být posloupnost bitů, znaků abecedy nebo dokonce mincí, v případě automatu na jídlo ...
- KA/FSM nemá paměť - je realizována stavy, ve kterých automat může být.

Konečný automat - definice

- Stroj pracuje v diskrétním čase - čte jeden znak vstupního slova najednou.
- Na základě přečteného znaku vstupního slova a aktuálního stavu, ve kterém se nachází, pak změní svůj stav.
- Do komplexního souboru všech stavů patří také takzvané koncové stavy. Pokud je po přečtení posledních znaků vstupního slova automat v jednom z koncových stavů - automat slovo přijme. Pokud je v jakémkoli jiném stavu - stroj toto slovo nepřijme.
- Soubor přijatých slov pak tvoří jazyk přijatý automatem.
- Stroj proto PŘIJME nebo ODMÍTNE vstupní slova.

Konečný automat – formální definice

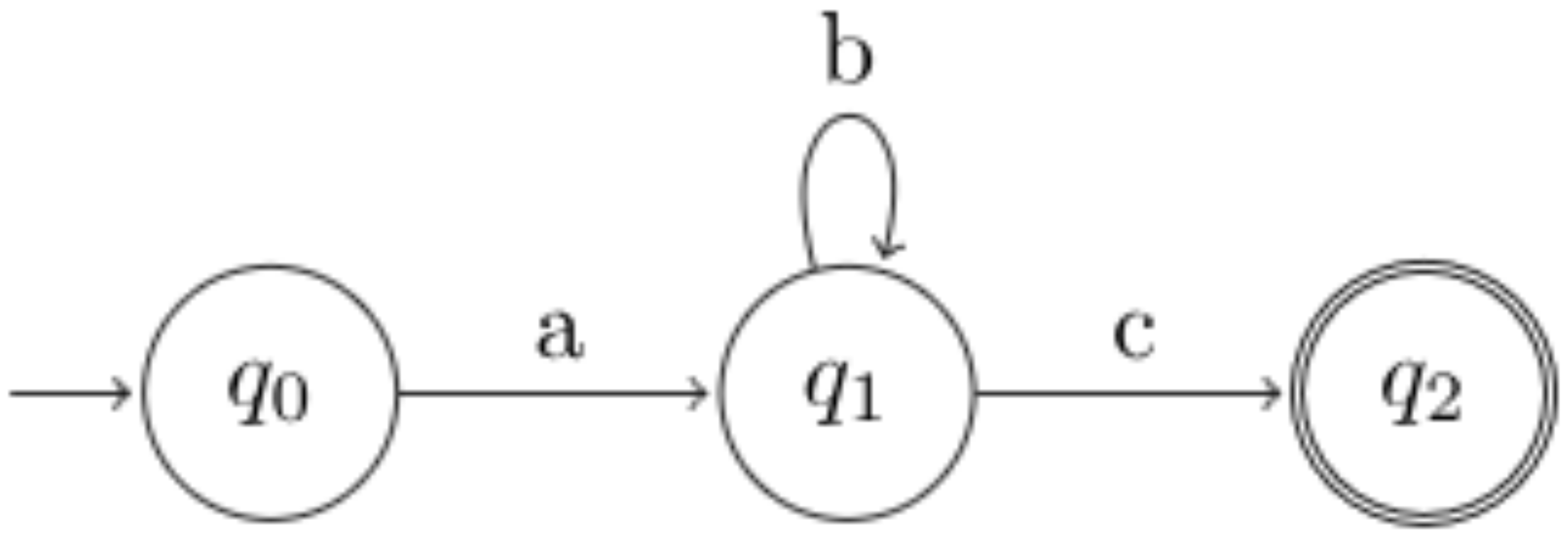
Formální definice:

Formálně lze konečný automat popsat jako uspořádanou pětici: (Q, I, f, q_0, P), kde:

- Q je jakákoliv konečná množina stavů,
- I je libovolná neprázdná konečná množina, kterou budeme nazývat vstupní abeceda,
- f je mapování z kartézského součinu Q x I do množiny Q. Pro uspořádané dvojice (vnitřní stav, vstupní symbol) je tedy přiřazen nový vnitřní stav automatu. Reprezentace f se nazývá přechodová funkce konečného automatu.
- q_0 je jeden významný prvek množiny Q, ve kterém automat začíná svou práci. Říkáme tomu počáteční stav .
- P je jakákoli podmnožina množiny stavů Q, nazývaná množina konečných stavů, někdy také množina přijímajících stavů automatu.

3. Deterministický automat - Příklad

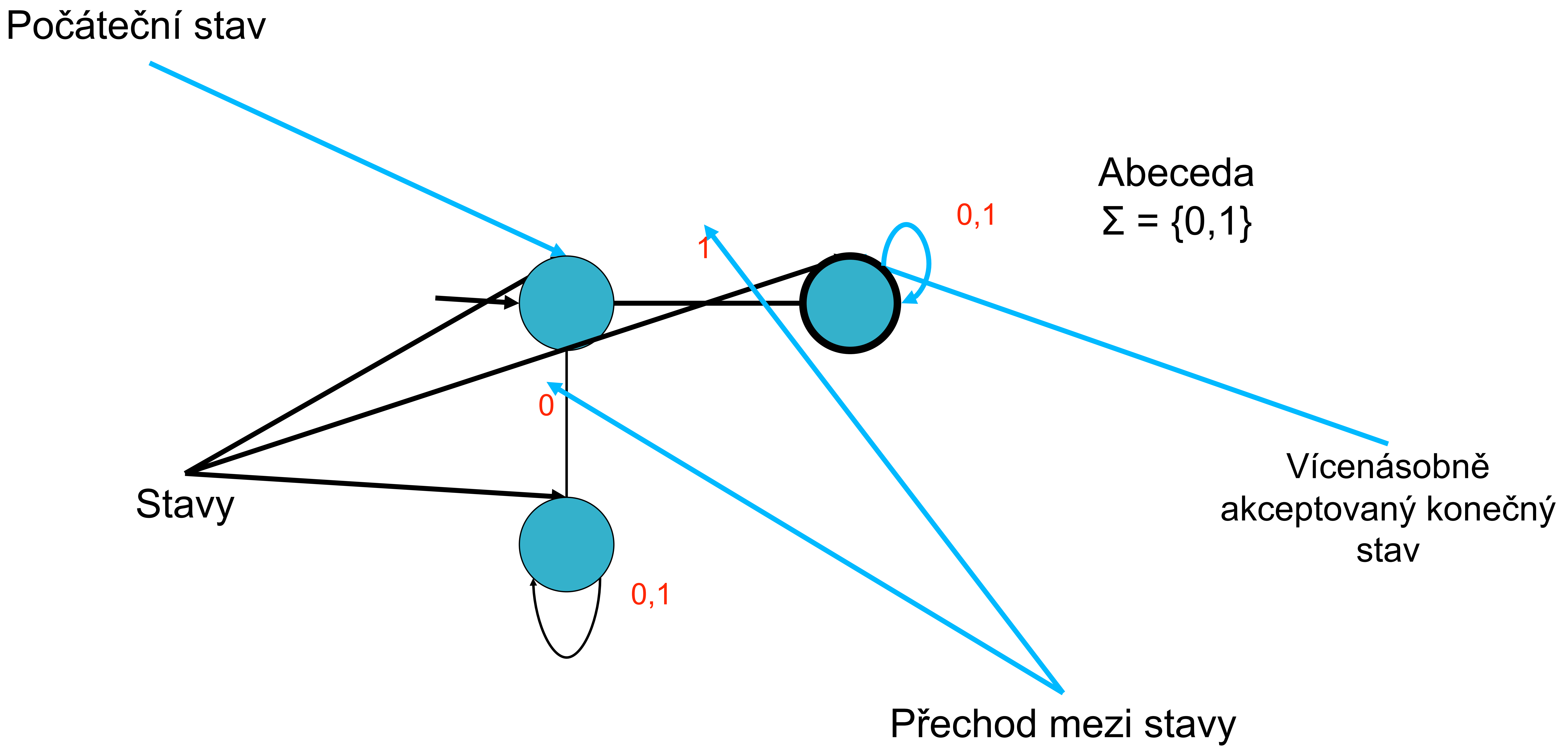
Konečný automat – jednoduchý příklad



Funkce je taková, že se automat pokusí přijmout slova, která se skládají z písmen „a, b, c“.

Pokud jsme na vstupu měli slovo „abbc“, slovo projde zleva doprava. Začíná v počátečním stavu q0 a ze slova převezme písmeno „a“, přičemž zůstane „bbc“. Uzel (stav) q1 ukazuje na sebe, takže tímto uzlem projdeme dvakrát a ze vstupního slova zůstane symbol „c“, který nás zavede do uzlu q2. V tomto okamžiku jsou načteny všechny symboly a stroj je v koncovém stavu, což znamená, že stroj přijme slovo „abbc“.

KA jako stavový diagram

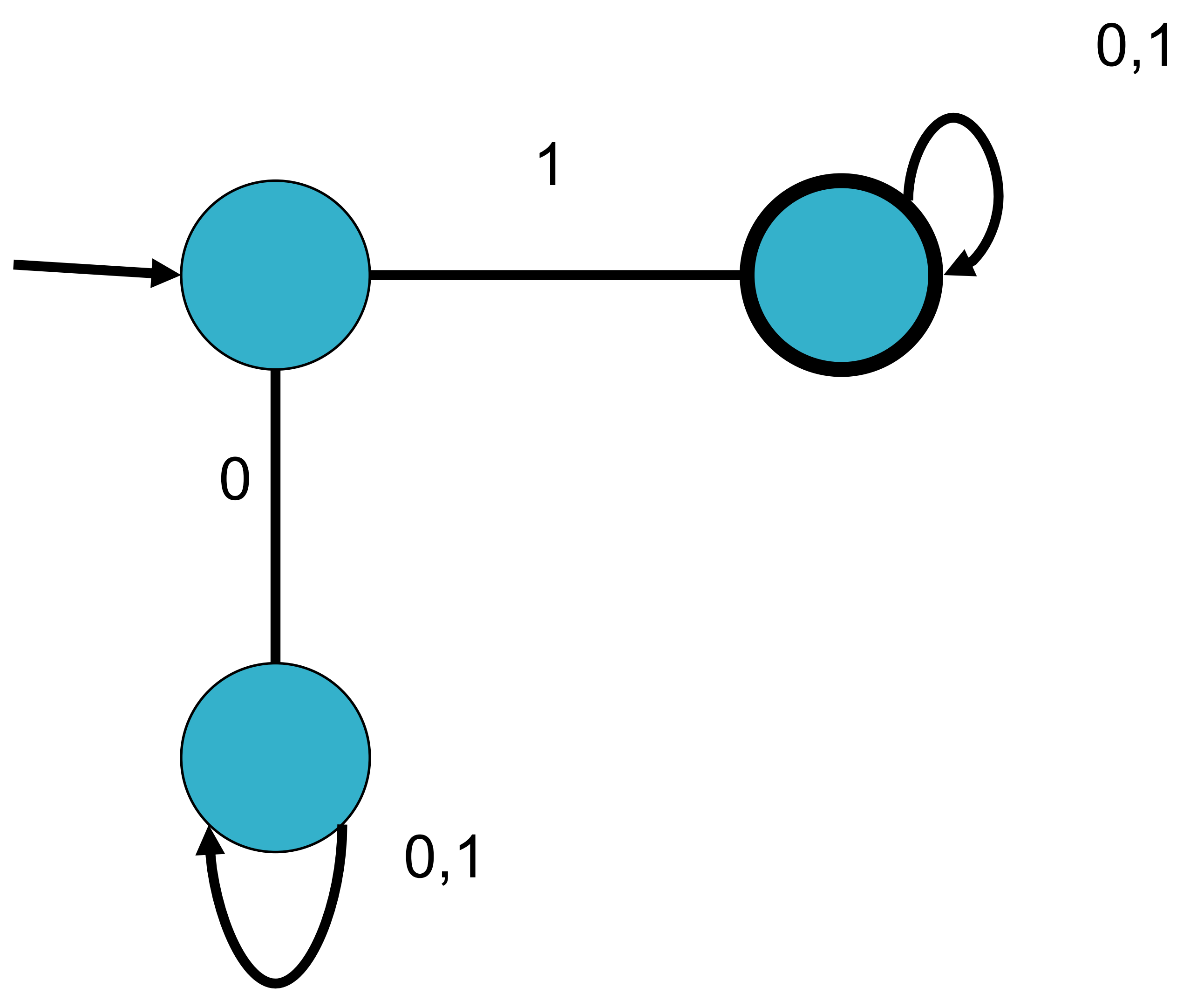


Automat lze vyjádřit pomocí grafu, kde uzly představují stavy a hrany přechod mezi stavy

KA jako stavový diagram

- Příklad automatu:

1010 akceptuje
0101 zamítá

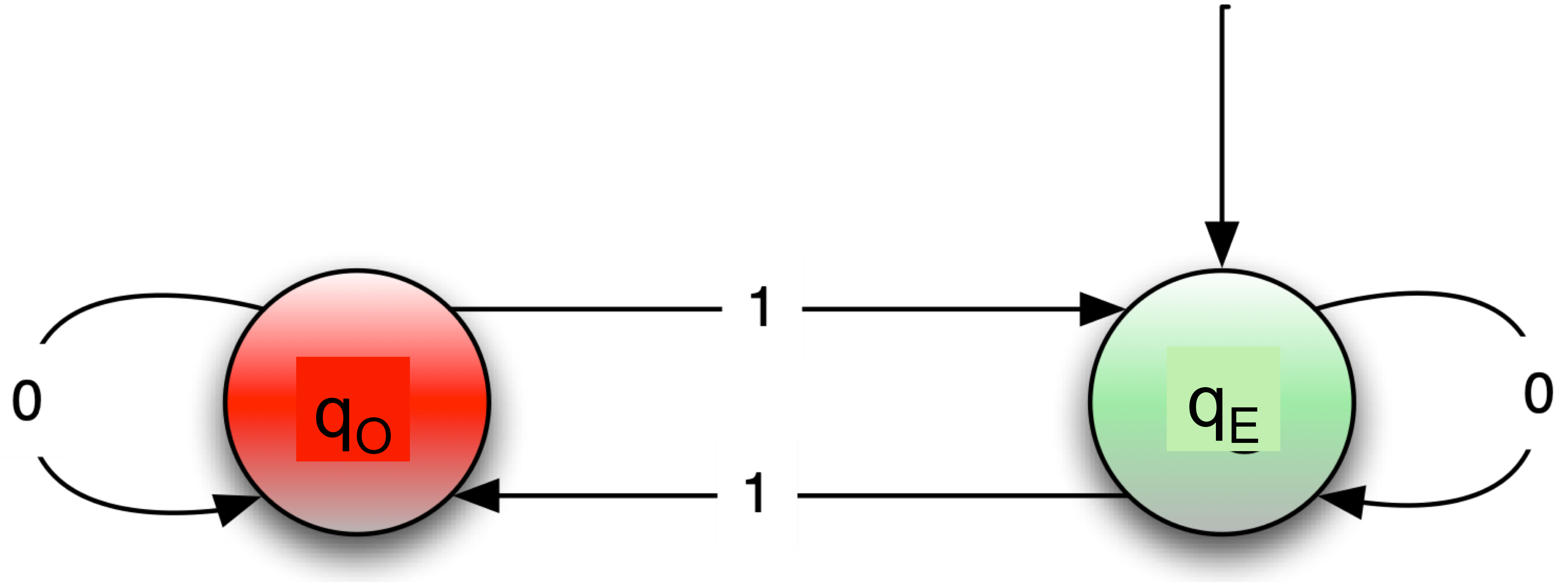


Jaký jazyk stroj rozpoznává?
$L = \{x : x \in \{0,1\}^*, x_1 = 1\}$

KA jako stavový diagram

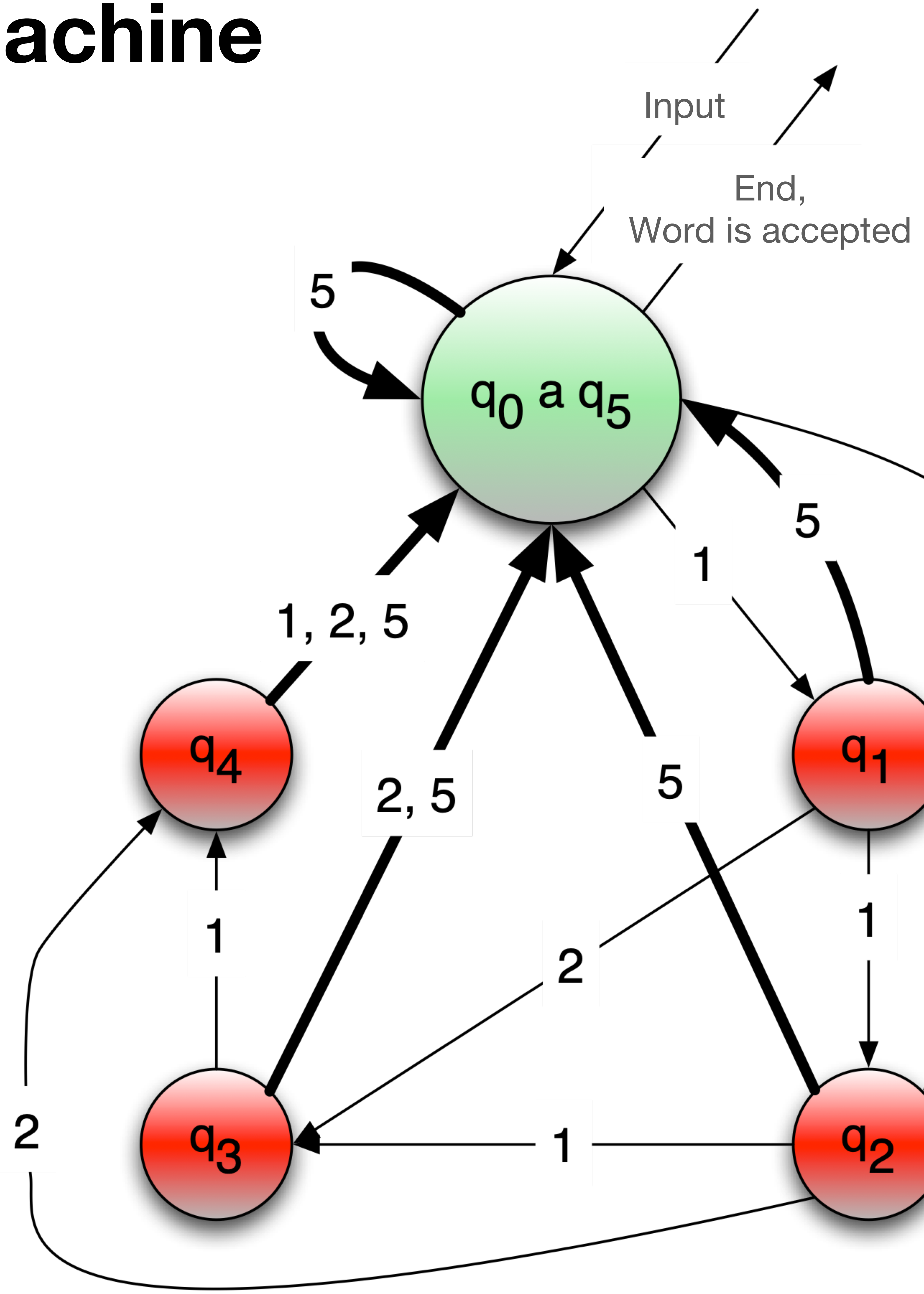
Rozpoznávání sudého a lichého počtu jedniček

$Q = \{q_O, q_E\}$,
$I=\{0, 1\}$,
$f(q_O, 0) = q_O$, $f(q_E, 0) = q_E$, $f(q_O, 1) = q_E$, $f(q_E, 1) = q_O$,
$q_0= q_E$,
$P = \{q_E\}$.



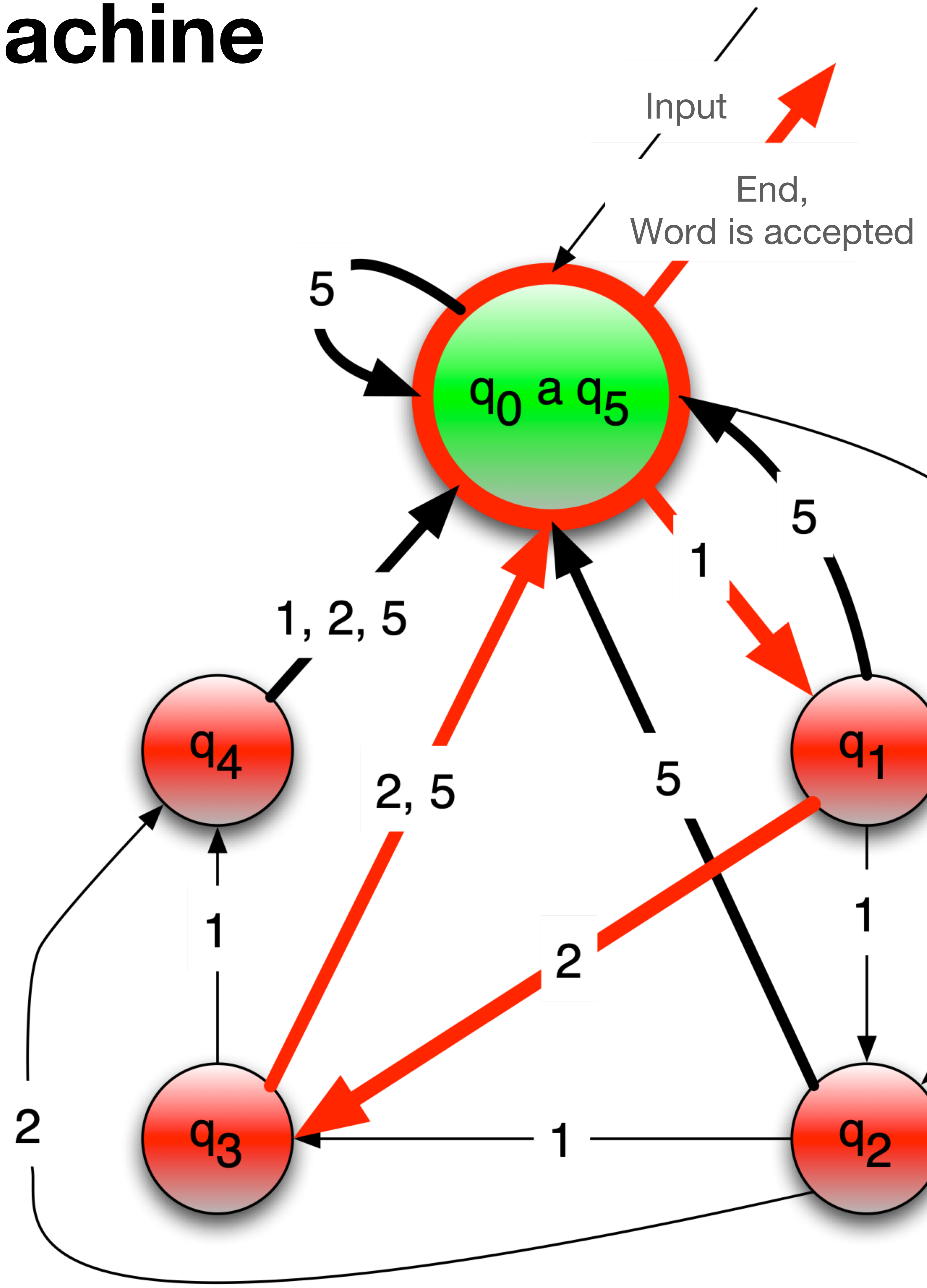
KA jako stavový diagram

Coffee vending machine



KA jako stavový diagram

Coffee vending machine



KA jako stavový diagram

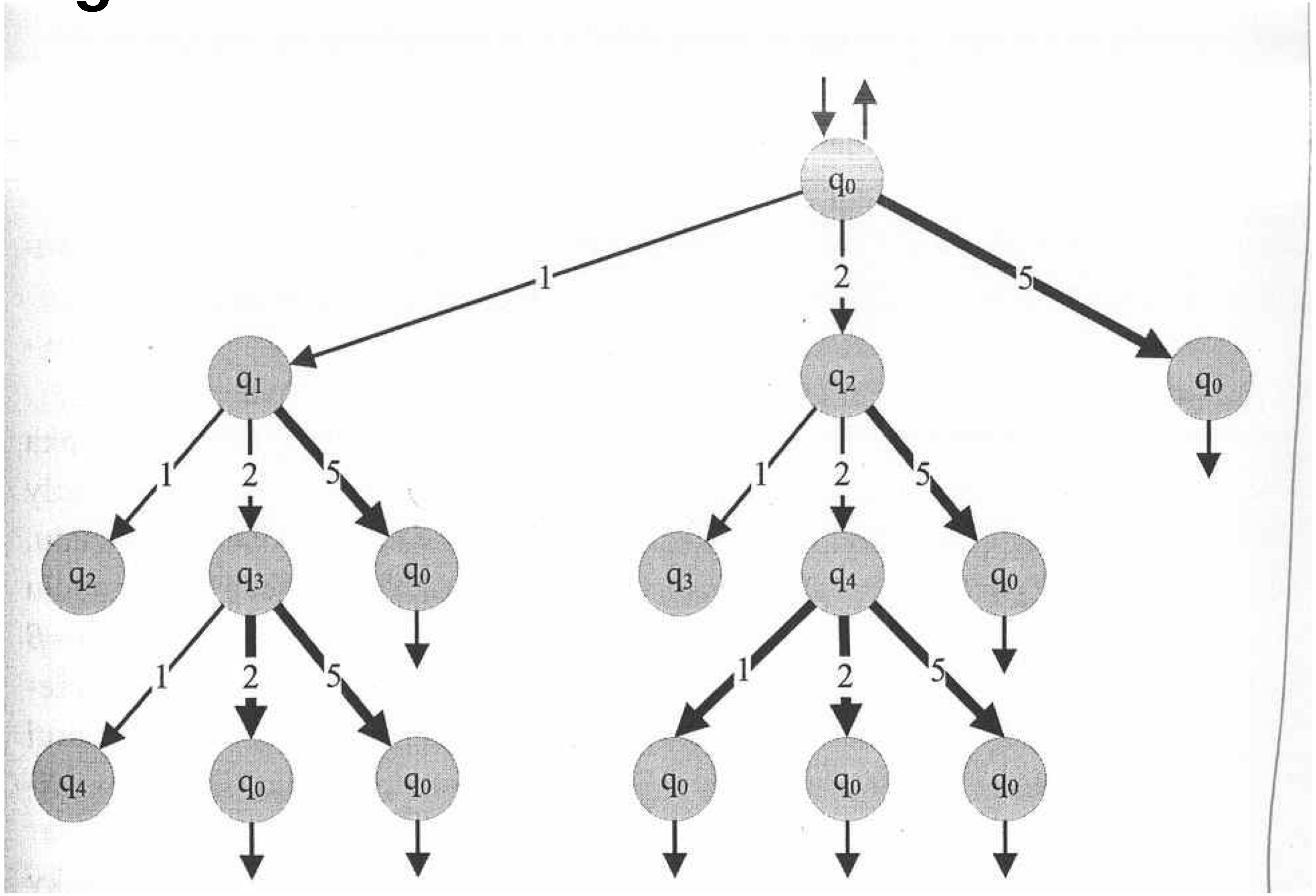
Coffee vending machine

Tento příklad by byl formálně popsán následovně: (Q, I, f, qo, {qo}), kde: Q = {q0, q1, q2, q3, q4}, I = {1, 2, 5}, a přechodová funkce f = (Q x I) → I je dána následující tabulkou:

	1	2	5
q_0	q_1	q_2	q_0
q_1	q_2	q_3	q_5
q_2	q_3	q_4	q_0
q_3	q_4	q_0	q_0
q_4	q_0	q_0	q_0

KA jako stavový diagram

Coffee vending machine



4. Nedeterministické automaty

Nedeterministický konečný automat

- NFSM (NFSA) – také NFA (nondeterministic finite acceptor), také NKA.
- Toto je zobecnění deterministického přístupu, což znamená, že jakýkoli deterministický konečný automat může být automaticky nedeterministickým konečným automatem.
- Nedeterminismus je často vyjádřen jako schopnost „hádat“ něco o vstupním řetězci (slovo).
- Což v tomto smyslu znamená nejednoznačnost.
- Z jednoho stavu může vést několik přechodů pro stejný vstupní symbol. Automat vždy zvolí (uhádne) takový přechod, aby přijal dané slovo, pokud je to možné.

Nedeterministický konečný automat

- Je tedy určen soubor různých reakcí automatu na danou situaci, tj. na danou kombinaci vnitřního stavu a vstupního symbolu.
- Formálně v definici konečného automatu stroje se popis přechodové funkce mění.
- Nyní to nebude mapování z Q x I na Q, ale mapování z Q x I na sadu neprázdných podmnožin Q.
- Pokud je k dané situaci přiřazena pouze jednoprvková podmnožina Q, je chování automatu v této situaci jasně dané, deterministické. V jiných případech však tato podmnožina může být víceprvková a automat se může chovat nedeterministicky, tj. přejít z této podmnožiny do jakéhokoli (možného) stavu.

Nedeterministický konečný automat

- Kromě toho, že některé přechody nemusí být určeny jednoznačně, je u nedeterministických automatů akceptováno, že počáteční stav, ve kterém automat začíná svou práci, nemusí být jediný.

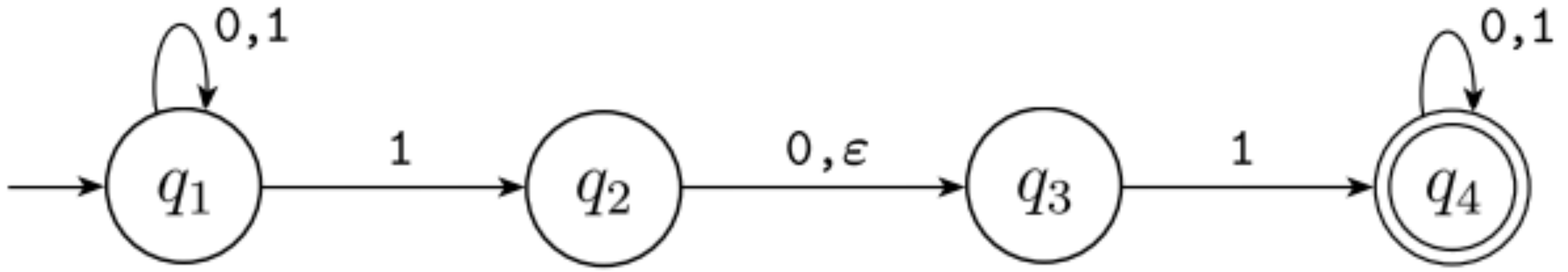
- Lze zadat libovolnou neprázdnou sadu možných počátečních stavů.

- Pro pohodlnější provádění některých operací jsou pro nedeterministické automaty povoleny také takzvané ε-transitions (epsilon přechody).

- V těchto případech může stroj změnit svůj stav bez čtení symbolu vstupního slova. Toto rozšíření není nutné.

Nedeterministický konečný automat

- Podle obrázku, ve stavu q_1 existuje jedna výstupní cesta pro symbol 0, ale dvě pro symbol 1.
- Ve stavu q_2 neexistuje žádný výstupní směr pro symbol 1.
- NFA tedy nemusí mít žádný výstupní směr nebo jeden nebo několik výstupních směrů pro každý symbol.



Nedeterministický konečný automat

- Na nedeterminismus lze pohlížet jako na druh paralelního výpočtu, kde lze spouštět více nezávislých procesů nebo vláken současně.

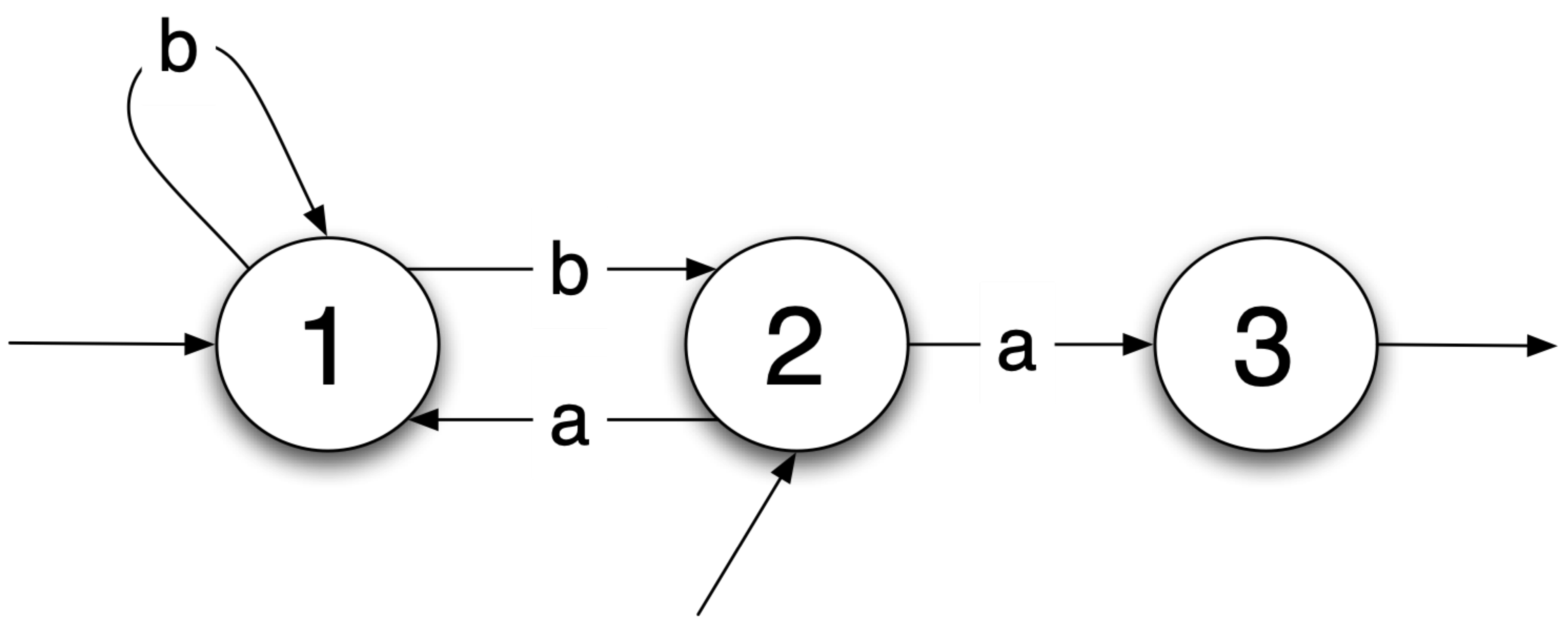
- V případě, že pro daný vstup neexistuje jednoznačný přechod do dalšího stavu, probíhá proces větvení, každou nově vytvořenou větev lze řešit samostatně.

- Pokud se stav „přijmout“ vyskytne alespoň v jedné větvi NFA, je přijat celý výpočet.

- Podobně si celý výpočet lze představit jako strom všech možností, kde kořen znamená začátek výpočtu a každý uzel spojující jednotlivé větve jako bod, kde došlo k situaci, kdy NFA měl na výběr přejít na další stav.

NFA - příklad

Zvažte následující nedeterministický konečný automat:

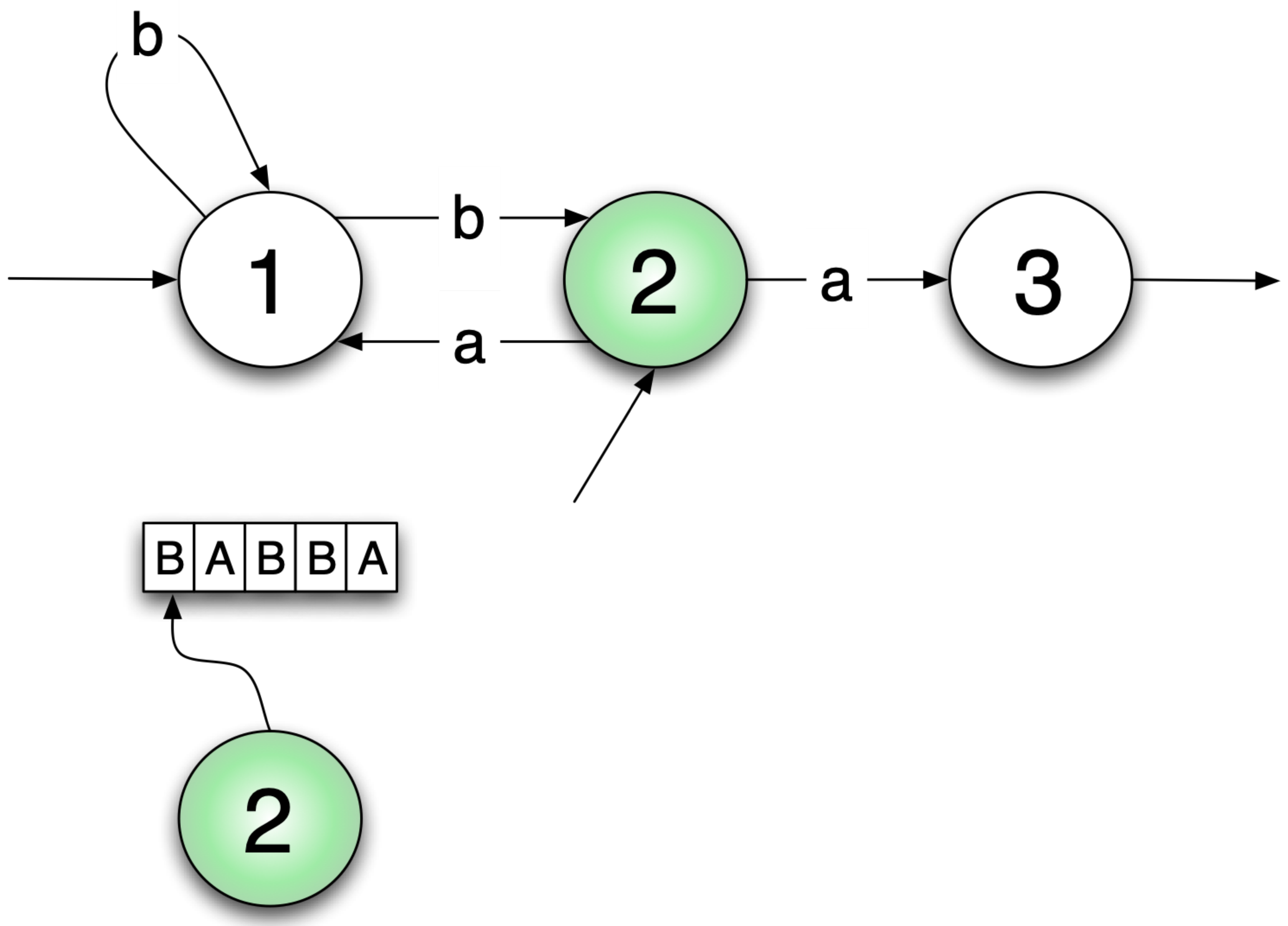


Přijímá stroj slovo „babba“?

NFA - příklad

Nejjednodušší případ je, když výpočet začíná ve stavu 2.
Pro b neexistuje žádný přechod a výpočet končí.

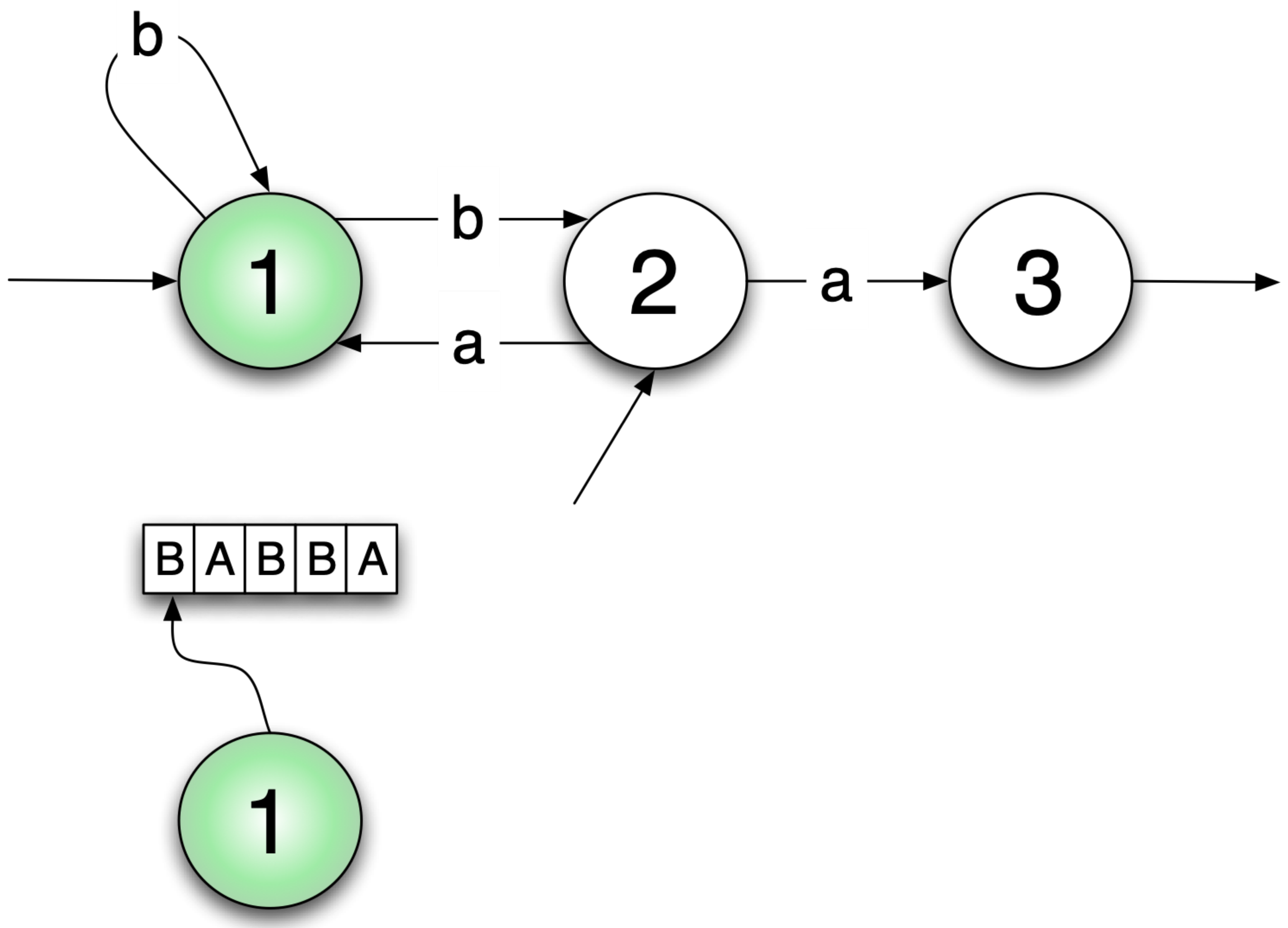
(2, babba)
Incomplete calculation



NFA - příklad

Pokud začneme ve stavu 1, máme již dva možné přechody pro první symbol.

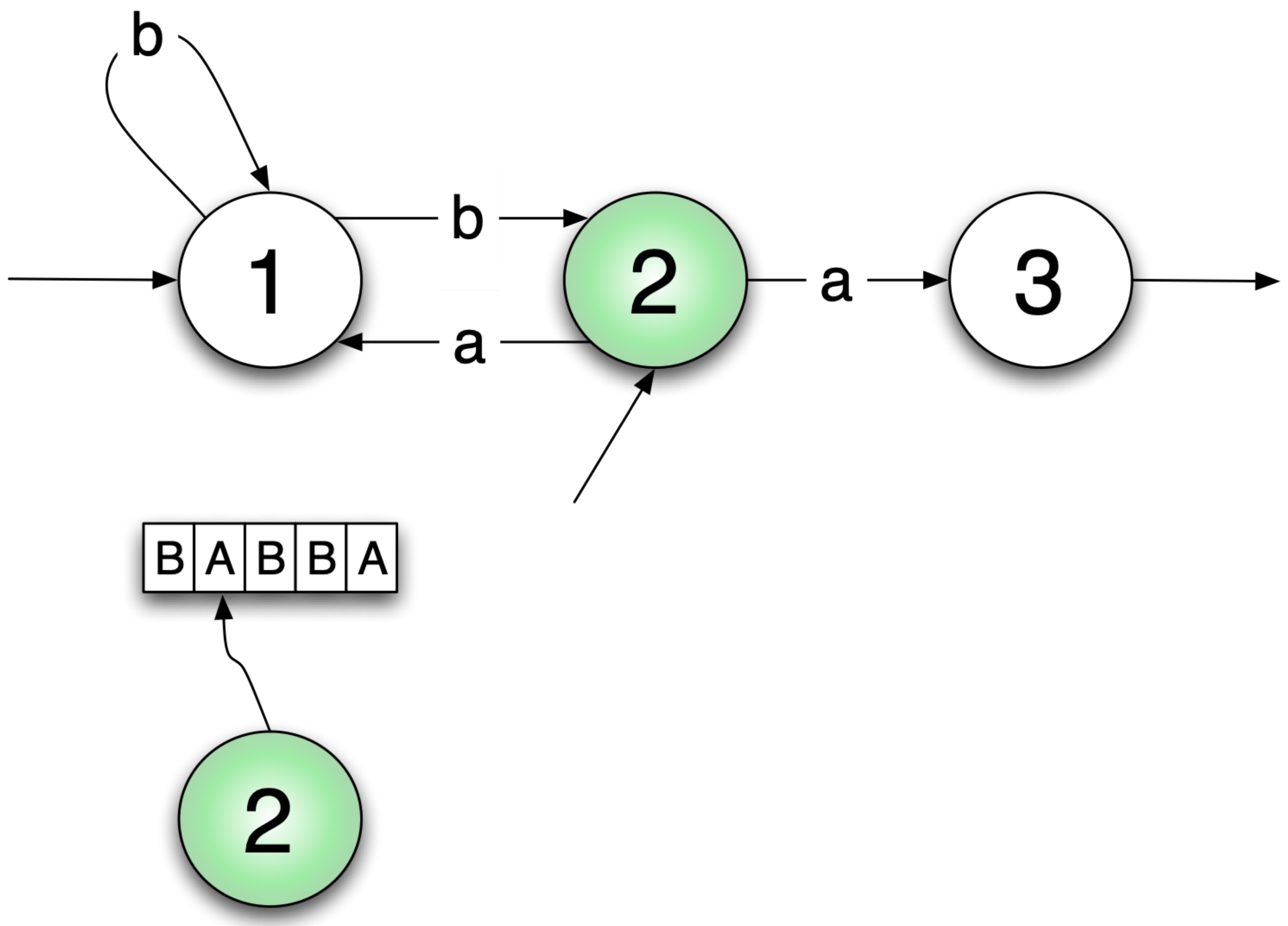
(1, babba)



NFA - příklad

Rozhodli jsme se jít do stavu 2. Opět máme na výběr.

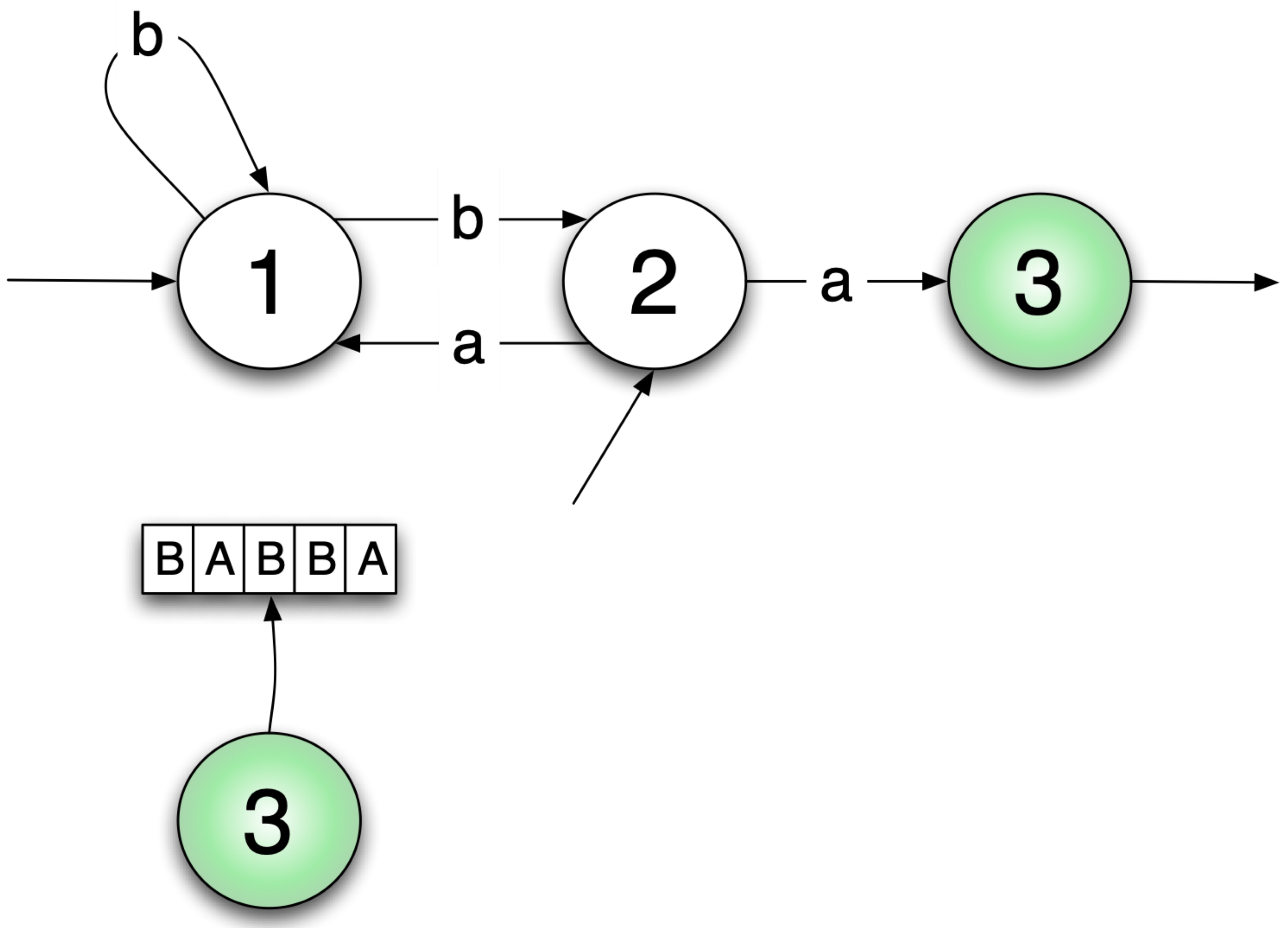
(1, babba) ⊢ (2, abba)



NFA - příklad

Pokud přejdeme do stavu 3, není možné ve výpočtu pokračovat.

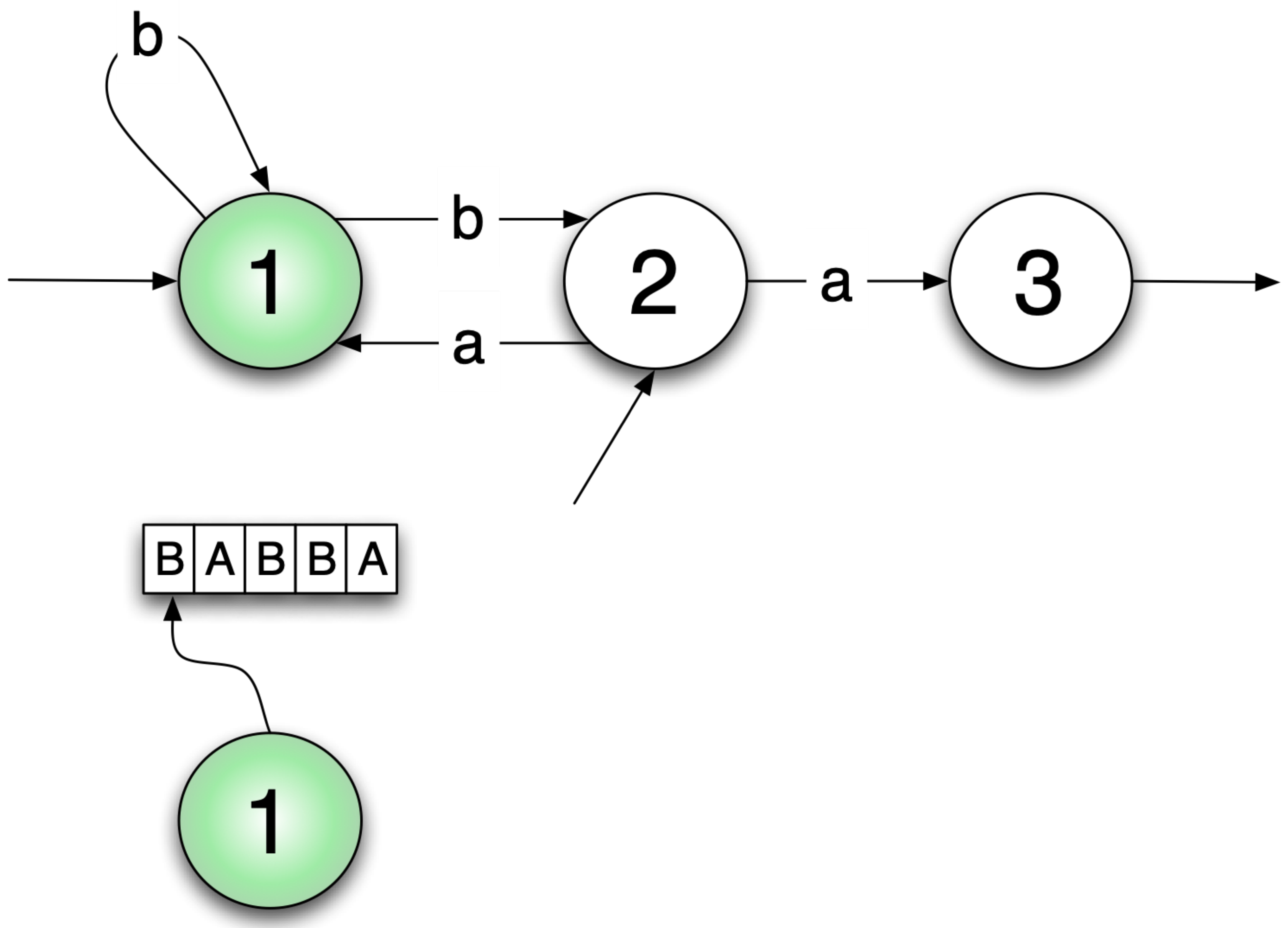
$(1, babba) \vdash (2, abba) \vdash (3, bba)$
Incomplete computation



NFA - příklad

Další možný výpočet zahájíme stejným přechodem jako v předchozím pokusu.

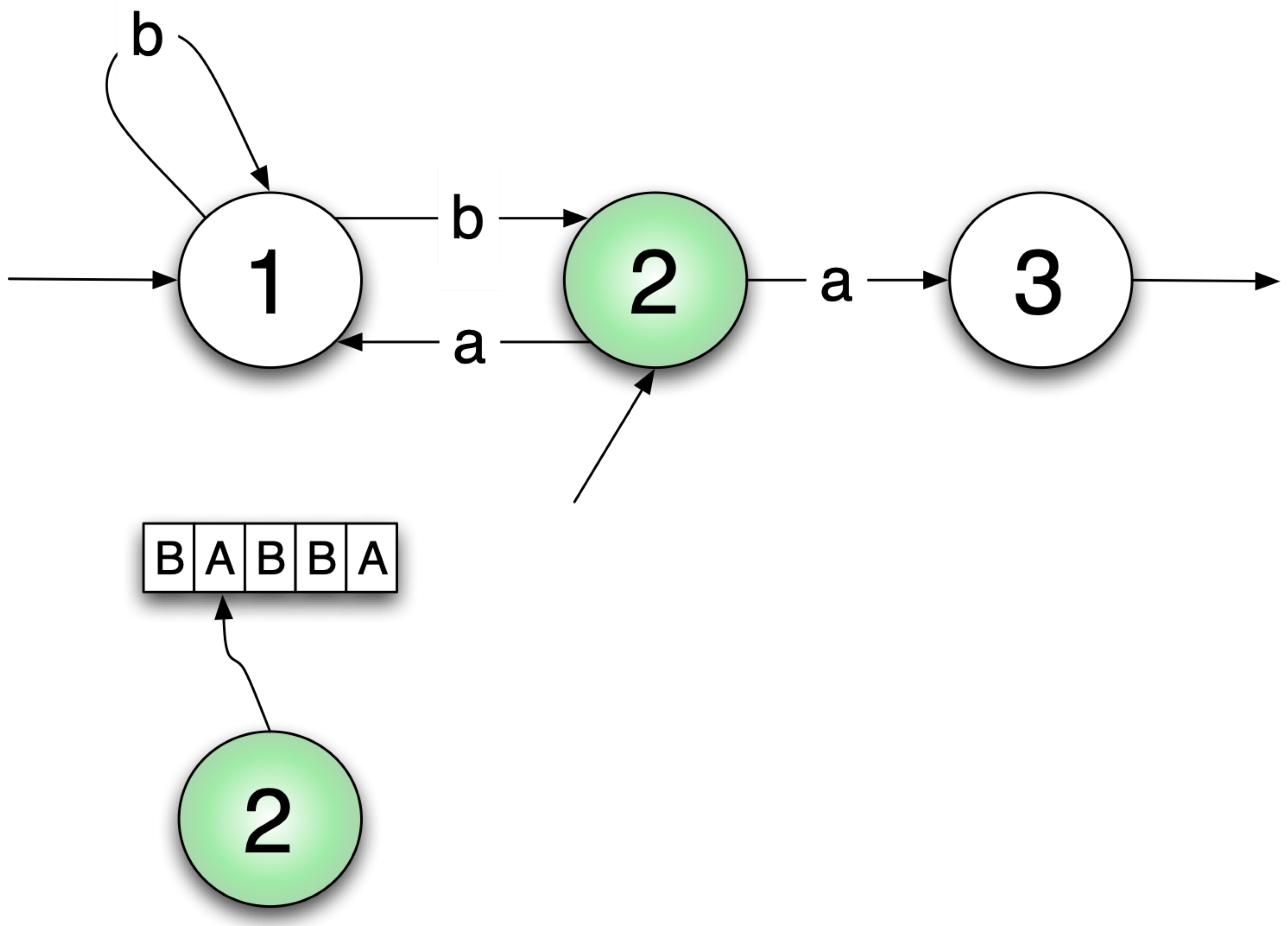
(1, babba)



NFA - příklad

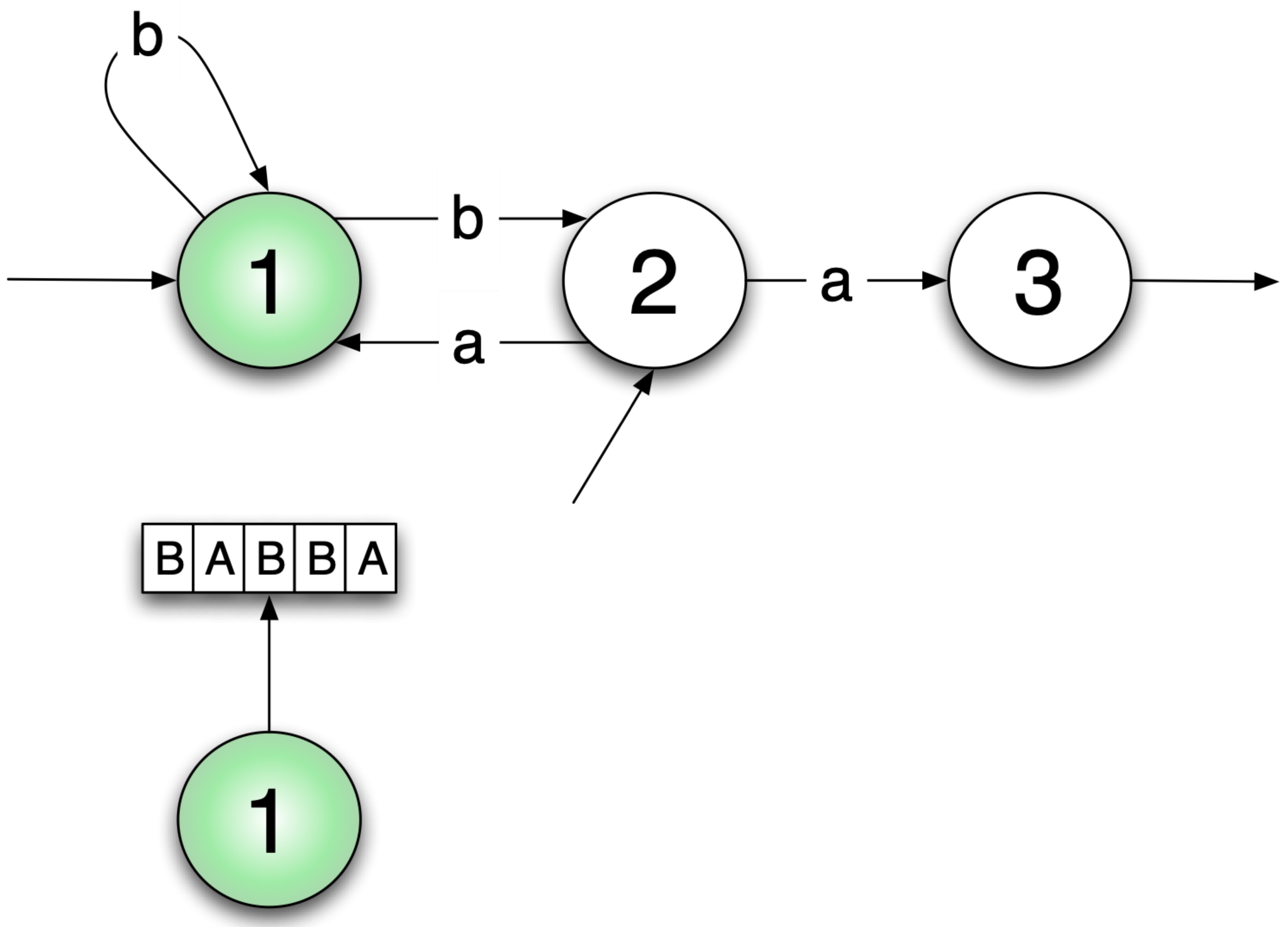
Zahájíme (pokračujeme) dalším možným výpočtem se stejnými přechody jako v předchozím případě.

(1, babba) ⊢ (2, abba)



NFA - příklad

Místo přechodu do stavu 3 jsme provedli přechod do stavu 1. Opět máme možnost pokračovat.

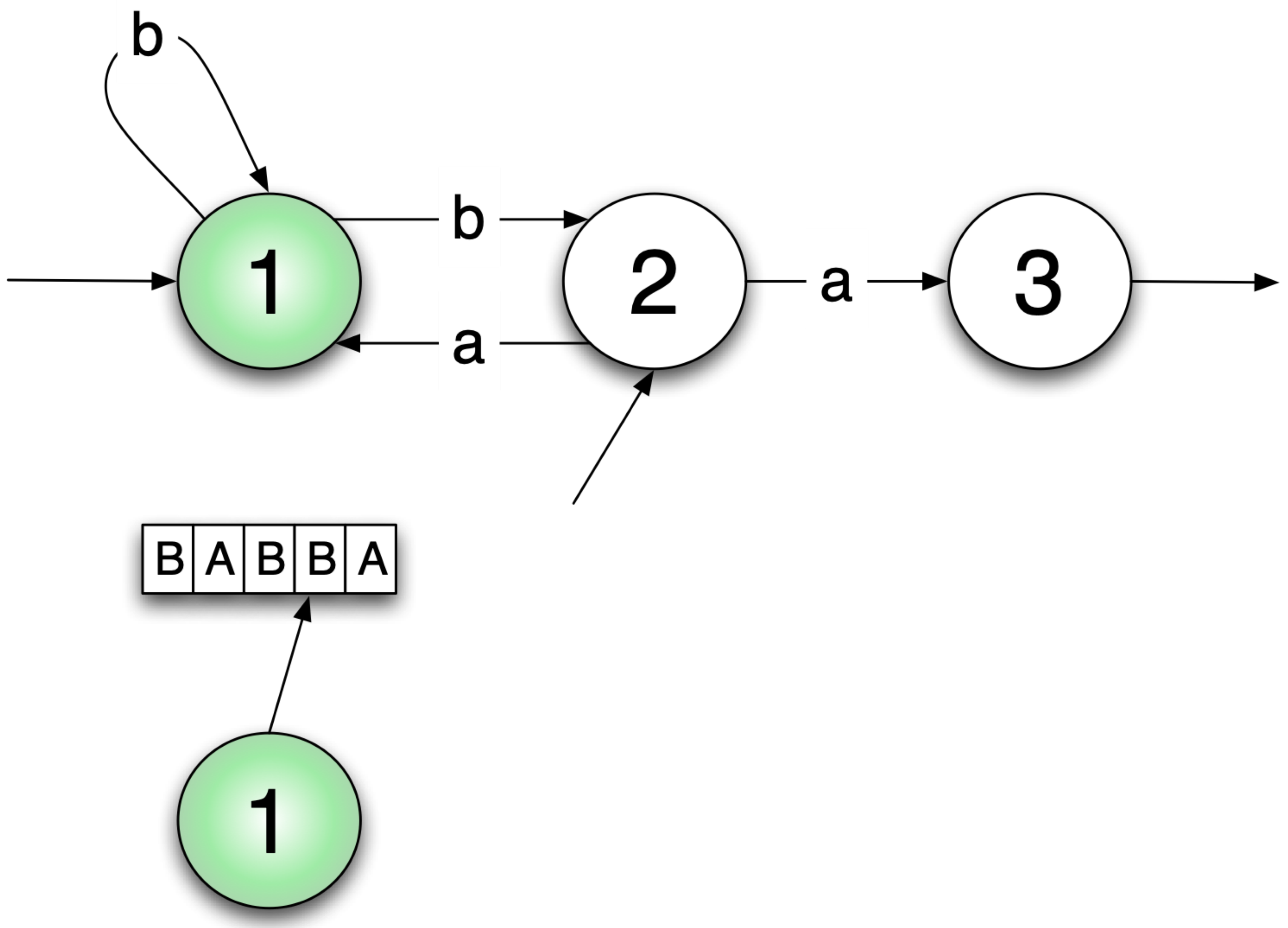


(1, babba) ⊢ (2, abba) ⊢ (1, bba)

NFA - příklad

Stroj zůstal ve stavu 1 a opět máme na výběr.

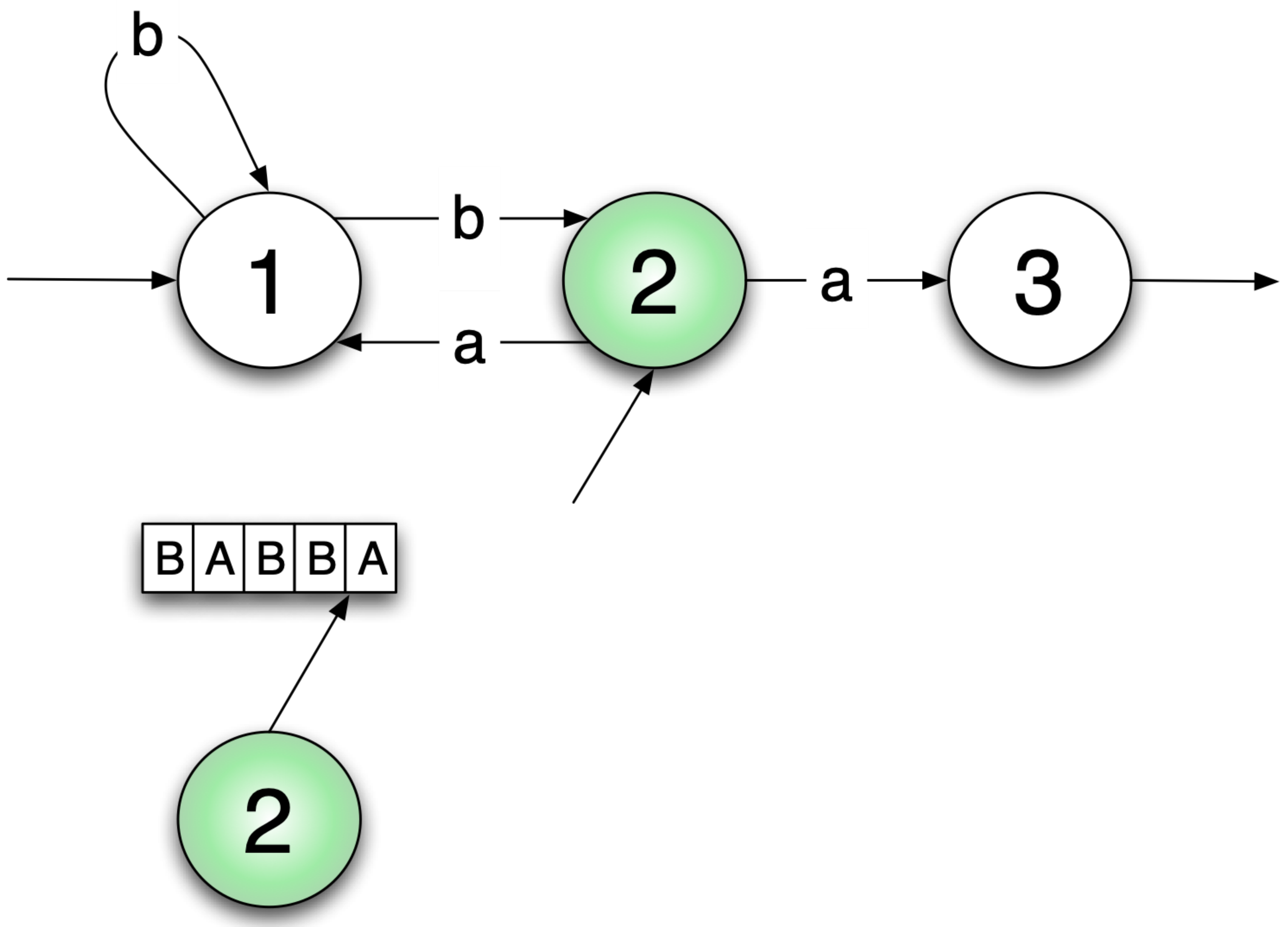
(1, babba) ⊢ (2, abba) ⊢ (1, bba) ⊢ (1, ba)



NFA - příklad

Tentokrát byl zvolen přechod do stavu 2 a my máme možnost pokračovat pro písmeno a.

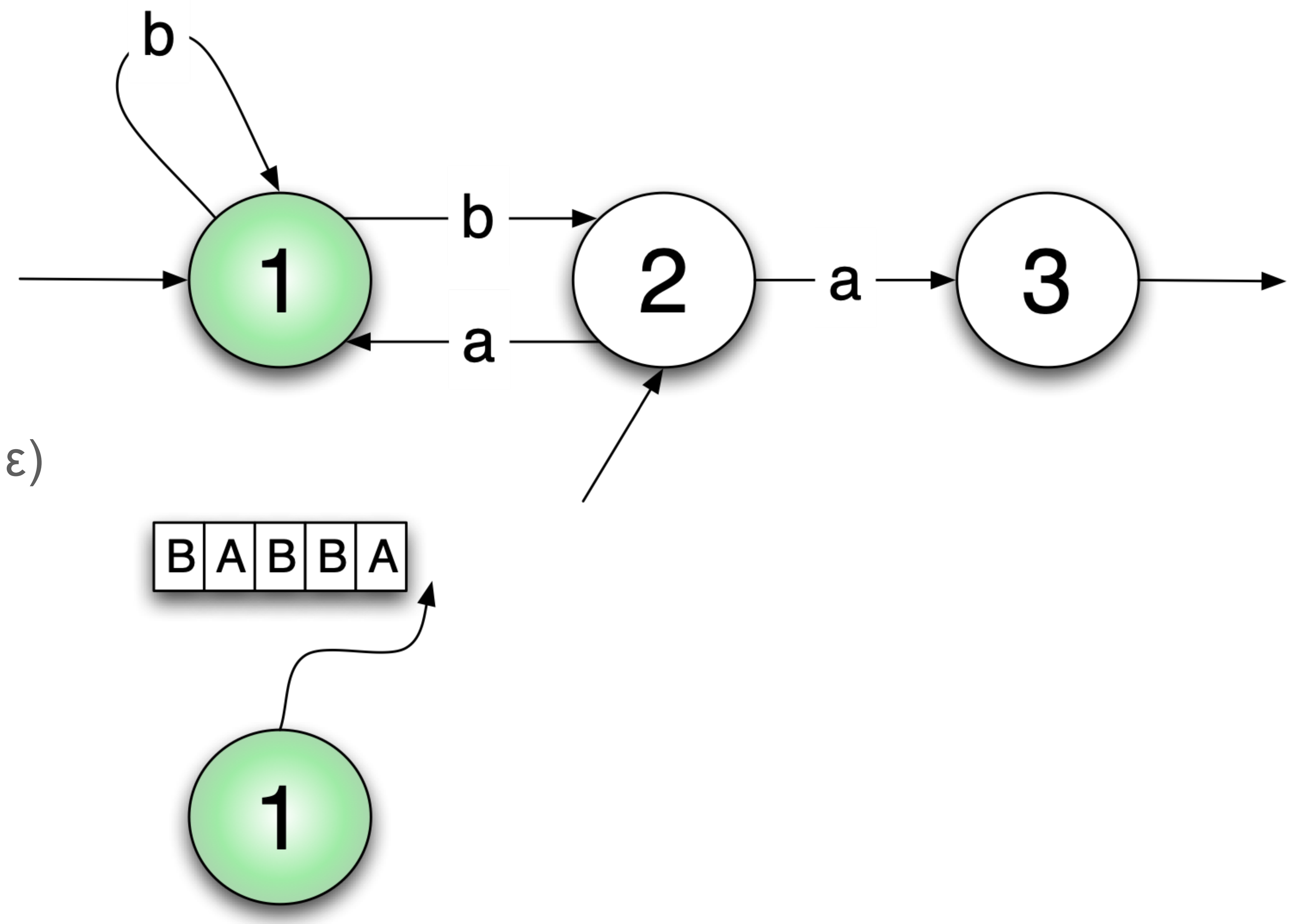
(1, babba) ⊢ (2, abba) ⊢ (1, bba) ⊢ (1, ba) ⊢ (2, a)



NFA - příklad

Úplný, ale odmítnutý (nepřijatý) výpočet

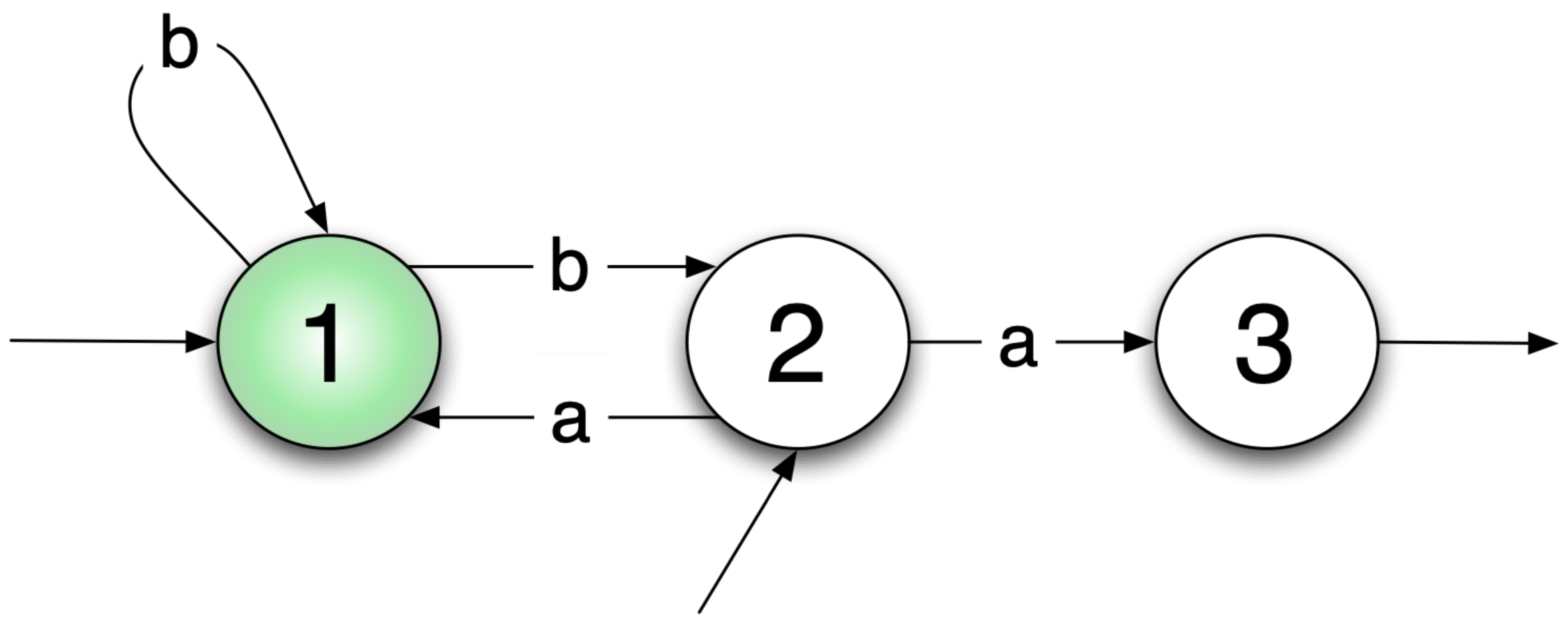
Pokud se rozhodneme přejít do stavu 1, pak jsme přečetli celé slovo, ale jsme v neakceptujícím (konečném) stavu, takže tímto výpočtem slovo nepřijmeme.



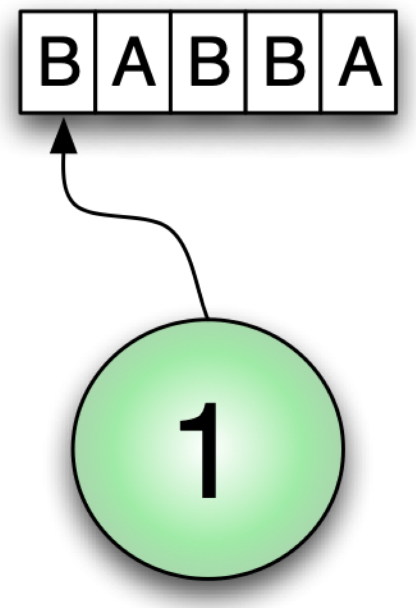
(1, babba) ⊢ (2, abba) ⊢ (1, bba) ⊢ (1, ba) ⊢ (2, a) ⊢ (1, ε)

NFA - příklad

Další možný výpočet je stejný jako předchozí, kromě posledního přechodu.



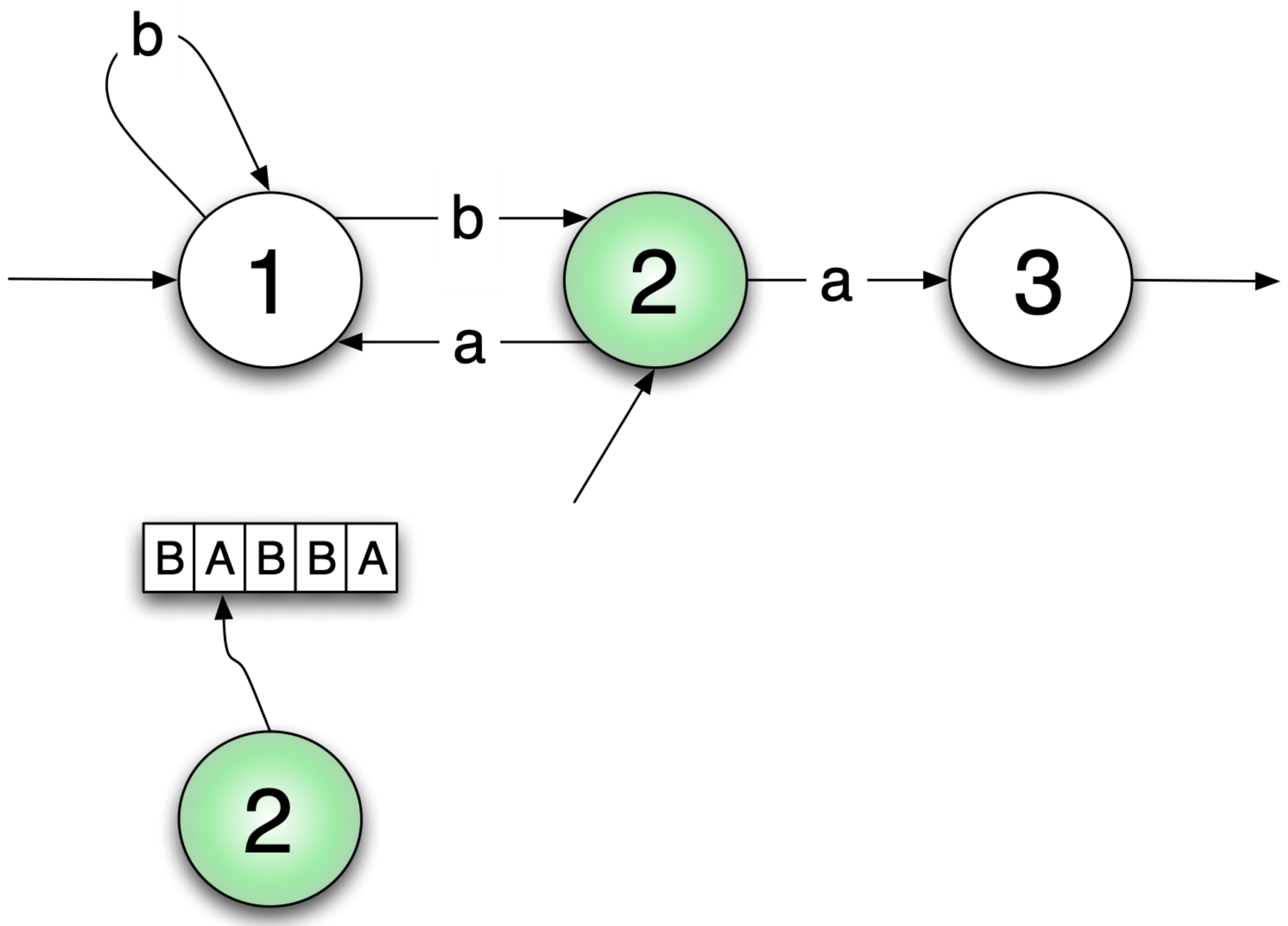
(1, babba)



NFA - příklad

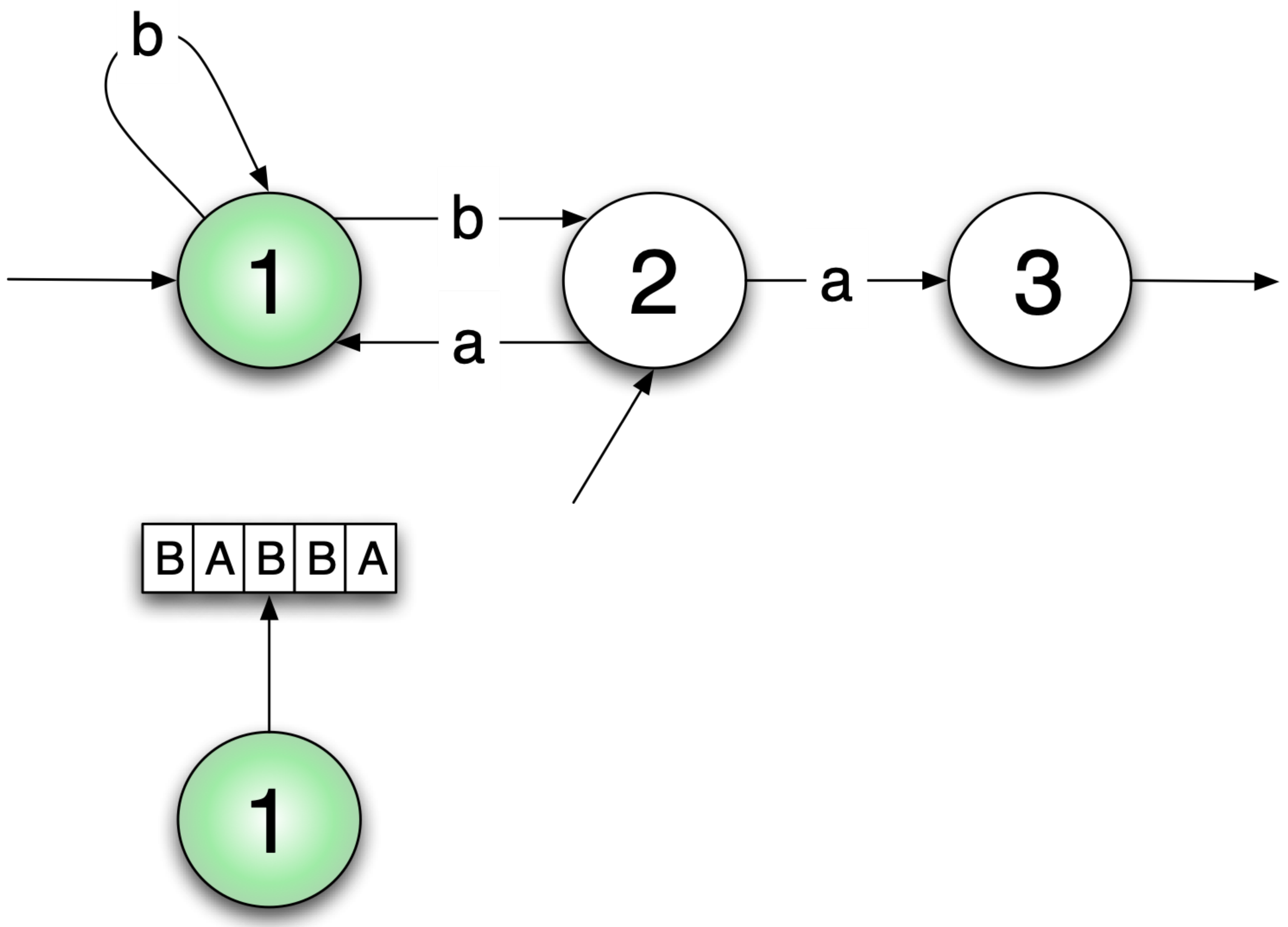
Další možný výpočet je stejný jako předchozí, kromě posledního přechodu.

(1, babba) ⊢ (2, abba)



NFA - příklad

Další možný výpočet je stejný jako předchozí, kromě posledního přechodu.

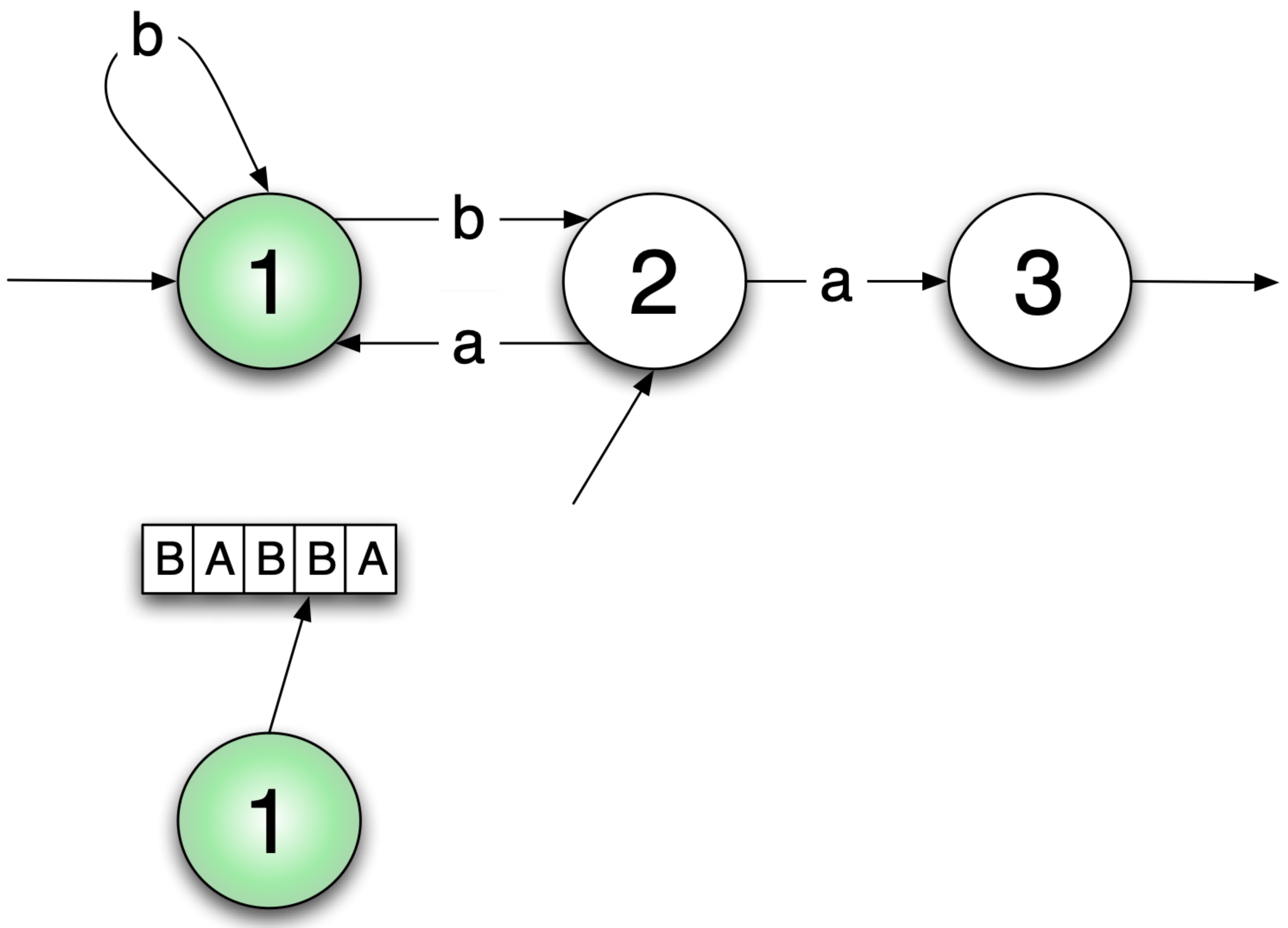


(1, babba) ⊢ (2, abba) ⊢ (1, bba)

NFA - příklad

Další možný výpočet je stejný jako předchozí, kromě posledního přechodu.

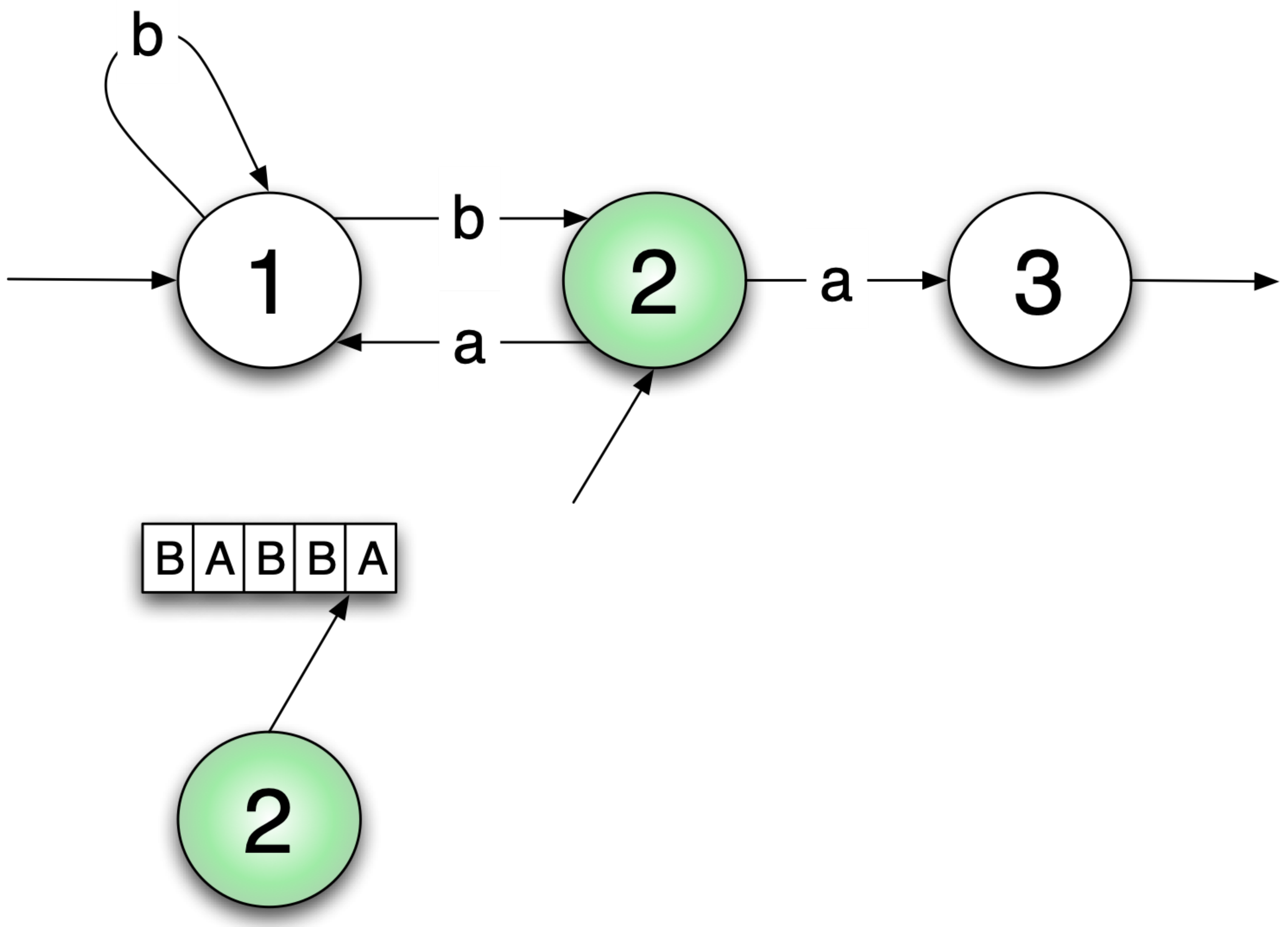
(1, babba) ⊢ (2, abba) ⊢ (1, bba) ⊢ (1, ba)



NFA - příklad

Další možný výpočet je stejný jako předchozí, kromě posledního přechodu.

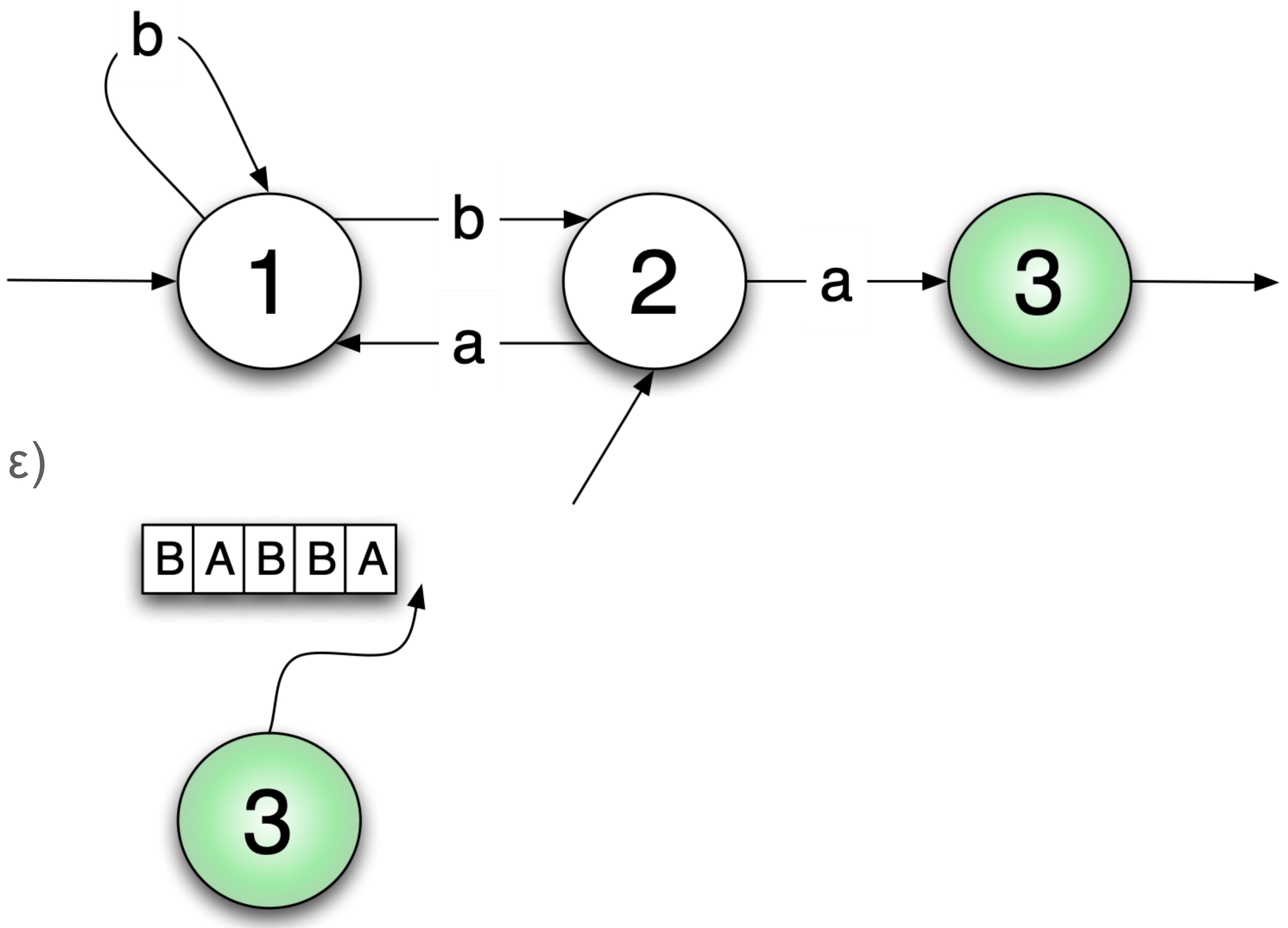
(1, babba) ⊢ (2, abba) ⊢ (1, bba) ⊢ (1, ba) ⊢ (2, a)



NFA - příklad

Našli jsme alespoň jeden „přijímající výpočet“, takže slovo je automaticky přijato.

(1, babba) ⊢ (2, abba) ⊢ (1, bba) ⊢ (1, ba) ⊢ (2, a) ⊢ (3, ε)



Děkuji za pozornost