PushDown Automaton

Nicolò Felicioni[1]

Dipartimento di Elettronica e Informazione
Politecnico di Milano

nicolo . felicioni @ polimi . it

March 10, 2021

[1]Mostly based on Nicholas Mainardi's material, enriched by few additional examples.

An extended computation model

Pushdown Automata

- Finite state automata are a handy model, but are not able to count an arbitrary number of items
- Idea: add a simple memory to the computation model: a *stack*
- The stack is a Last-In-First-Out memory (LIFO)
- The read operation on the memory erases the value (pop operation)
- The resulting automaton is known as a PushDown Automaton (PDA)
- For this computation model the non-determinism enhances the computation capability

Formalization

Definition

- A recognizer PDA is formally defined as a 7-tuple $(\mathbf{Q}, \mathbf{I}, \Gamma, \delta, q_0, \mathbf{F}, Z_0)$, where:
  - $\mathbf{Q}$ is the set of states of the automata
  - $\mathbf{I}$ is the alphabet of the input string which will be checked
  - Γ is the alphabet of the symbols on the stack
  - $\delta : \mathbf{Q} \times (\mathbf{I} \cup \epsilon) \times \Gamma \mapsto \mathbf{Q} \times \Gamma^*$ the transition function
  - $q_0 \in \mathbf{Q}$ the (unique) initial state from where the automaton starts
  - $\mathbf{F} \subseteq \mathbf{Q}$ the set of final accepting states of the automaton
  - Z_0 is the symbol which indicates the bottom of the stack

Structure

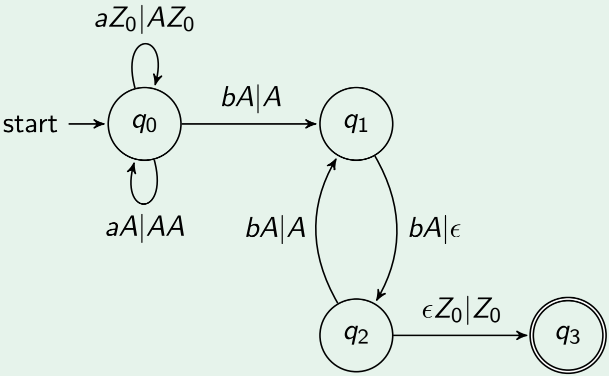
Advantages

- The main advantage of a PDA is that it is able to count letters
- The transition function relies on reading both a symbol from the input and a symbol from the stack to perform a transition
- Nondeterminism takes place when two transitions have both the same input and the same stack symbol as a trigger (from the same state)...
- ... or with ϵ-transitions, as always

A first attempt

$L = a^n b^{2n}, n \geq 1$

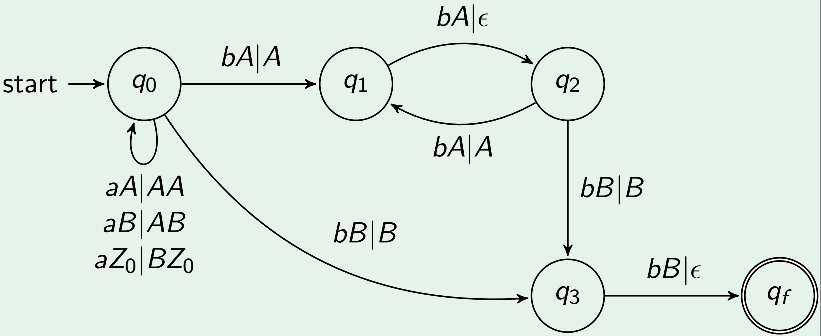
- To obtain a synthetic notation, the convention is to denote on an arc $<input><stack> \mid <stack> \cdots <stack>$



A first attempt

$L = a^n b^{2n}, n \geq 1$

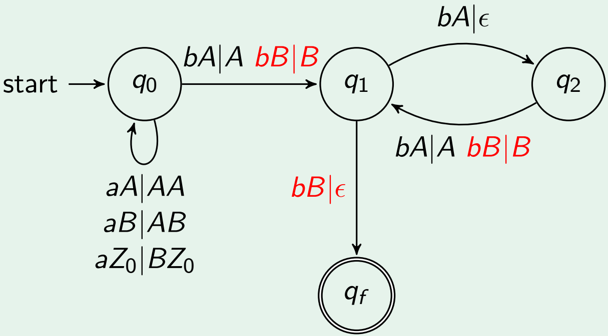
- Eliminating ϵ-transitions with an extra symbol



A first attempt

$L = a^n b^{2n}, n \geq 1$

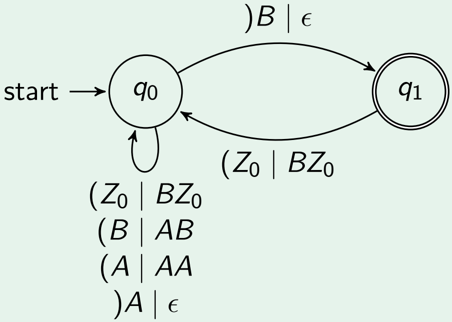
- Eliminating ϵ-transitions with an extra symbol
- q_1 and q_3 can be merged in a single state ...



Three Classic Languages Recognized By PDA

Well-parenthesized empty expressions

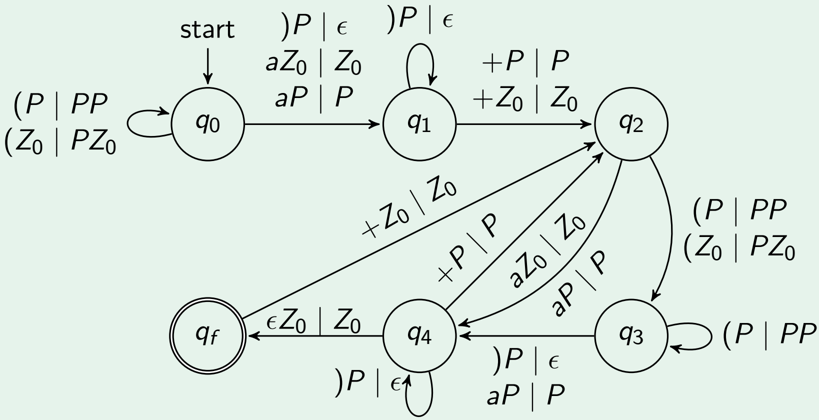
Examples: $((()))(), ()(), (()()) \in L$. $((), (,)(, \epsilon \notin L$.



Three Classic Languages Recognized By PDA

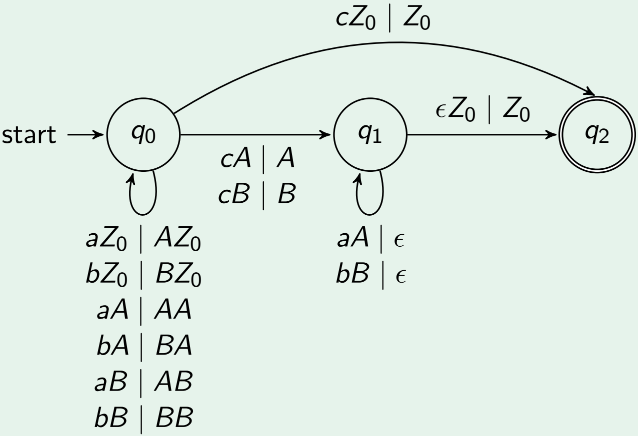
Well-parenthesized additions with empty parenthesis i.e. a+(())

$L_1 = \{a \vee ({}^n a^?)^n \mid n \geq 1\}$, $L = \{({}^n y + z)^n \mid n \geq 0, y, z \in L \cup L_1\}$



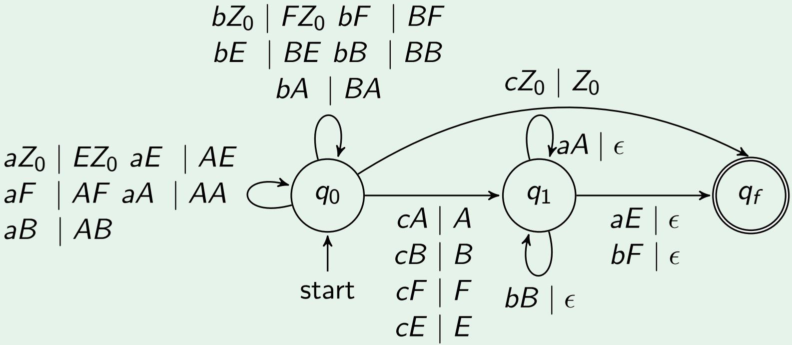
Three Classic Languages Recognized By PDA

$L = \{wcw^R \mid w \in \{a, b\}^*\}$



Three Classic Languages Recognized By PDA

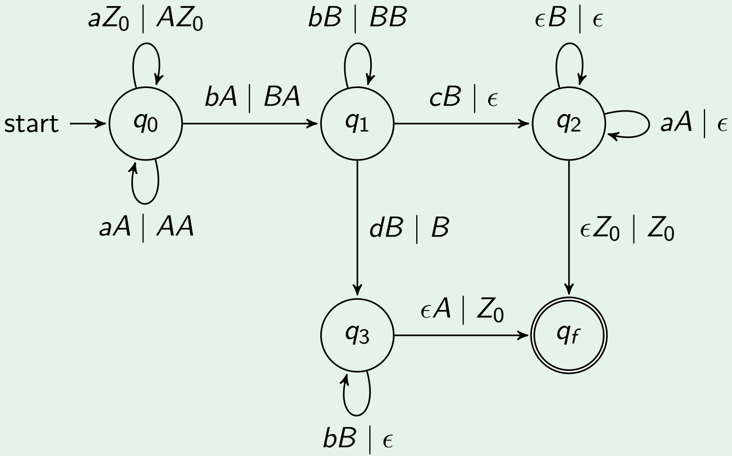
$L = \{wcw^R \mid w \in \{a, b\}^*\}$, no ϵ-transitions



Stack symbols E, F are used to show how to avoid ϵ-transitions to go from q_1 to q_f, but they are not strictly necessary

Is It Always Possible To Remove ϵ-transitions?

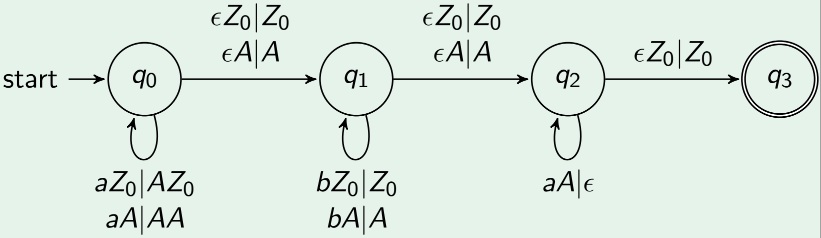
$L = a^n b^p c a^n \cup a^n b^p d b^p,\ n \geq 1,\ p \geq 1$



Another exercise

$L = a^n b^m a^n, n, m \geq 0$

Naive idea: for each a, push A on the stack, ignore the bs, match the last as with the As on the stack.

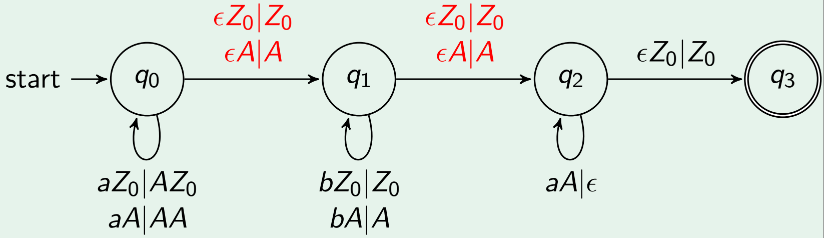


Another exercise

$L = a^n b^m a^n, n, m \geq 0$

The ϵ-transitions of q_0 and q_1 make the PDA non-deterministic.

(N.B.: the one from q_2 is not a problem)



Another exercise

$L = a^n b^m a^n, n, m \geq 0$

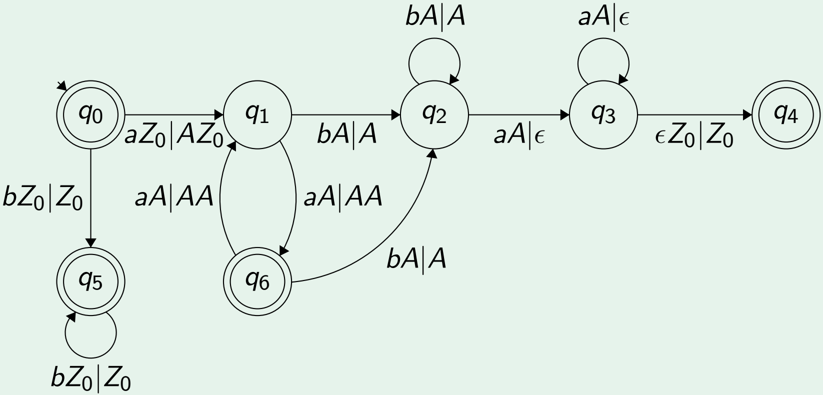
We notice how this language includes the following cases:

- ϵ (i.e. $n = m = 0$)
- b^+ (i.e. $n = 0, m \neq 0$)
- a^{2n} (i.e. $n \neq 0, m = 0$)
- $L_1 = a^n b^m a^n, n, m > 0$ (i.e. $n \neq 0, m \neq 0$)

We have to check for all of them at the same time.

Another exercise

$L = a^n b^m a^n, n, m \geq 0$



Complementing PDA

Strategy (same of FSA): complete the δ function introducing error states, then swap final and non final states

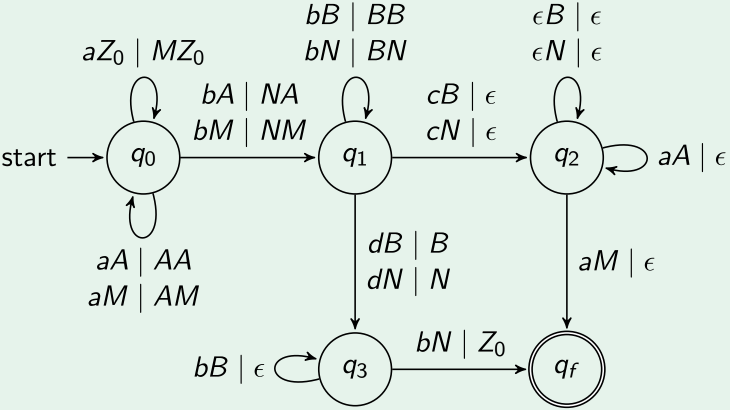
Difficulties Due to ϵ-transitions

1. What if there is a loop with ϵ-transitions which pile characters in the stack? $\rightarrow$ The automaton may not stop!
   - $\hookrightarrow$ it is always possible to make the PDA acyclic, that is no infinite cycles of $\epsilon-$transitions
2. What if from a final state q_f there is an ϵ-transition to a state $q_i \notin F$ (or vice versa)? $\rightarrow$ Suppose the PDA stops in q_f when a string s is accepted. When we perform the complement, $q_f \notin F$ while $q_i \in F$, thus s is accepted by the complement PDA too by performing an additional ϵ-transitions to q_i
   - $\hookrightarrow$ Luckily, removing these kind of sequences is always possible
3. Beware of introducing non-determinism while completing the δ function...

Complementing PDA

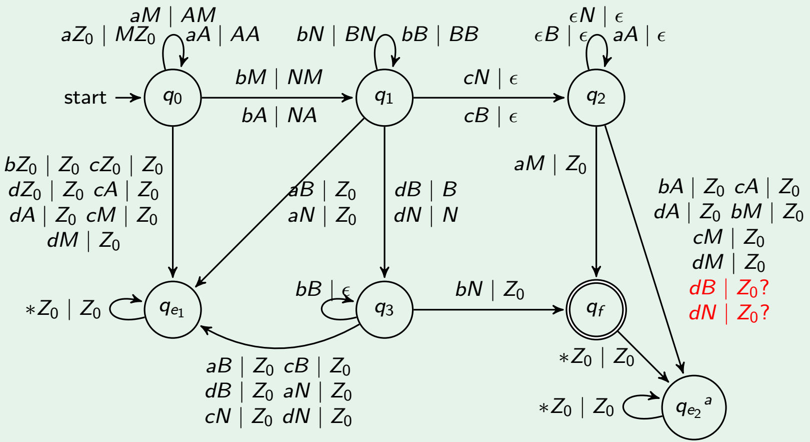
Removing troublesome ϵ-moves

ϵ-moves to q_f must be removed before applying the complement



Complementing PDA

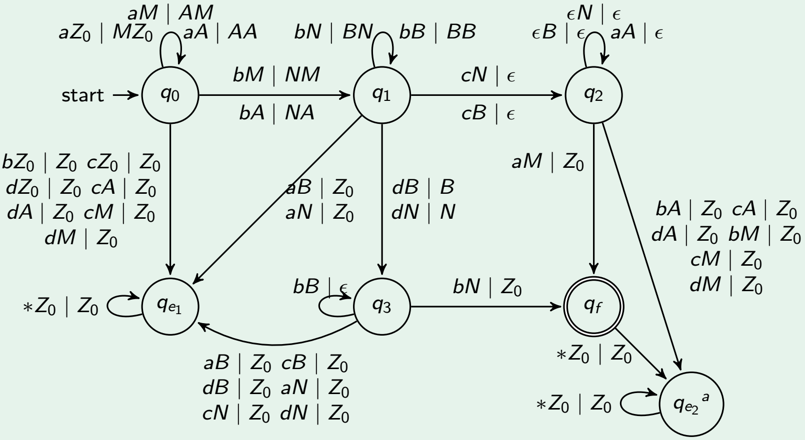
PDA for $L = \{a^n b^p c a^n \cup a^n b^p d b^p,\ n \geq 1,\ p \geq 1\}$ with total δ



[a] This state is present just for graphical purposes. It is equivalent to q_{e_1}

Complementing PDA

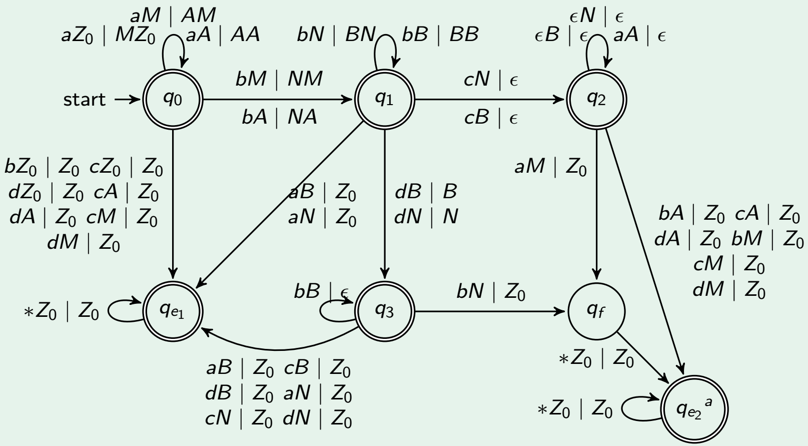
Deterministic PDA for L with total δ



[a] This state is present just for graphical purposes. It is equivalent to q_{e_1}

Complementing PDA

Deterministic PDA for $L^c = \{a^n b^p c a^n \cup a^n b^p d b^p,\ n \geq 1,\ p \geq 1\}^c$



[a] This state is present just for graphical purposes. It is equivalent to q_{e_1}

Formalization

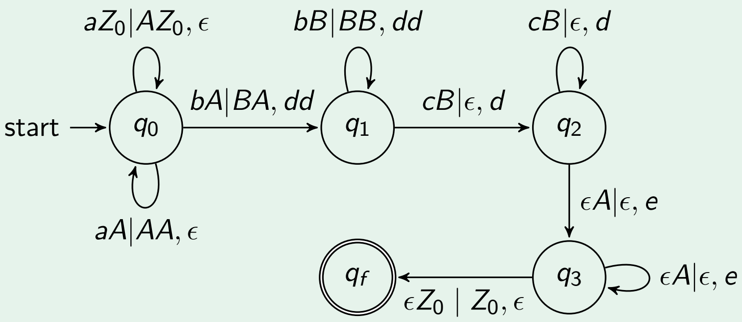
Definition

- A transducer PDA is formally defined as a 9-tuple $(\mathbf{Q}, \mathbf{I}, \Gamma, \delta, q_0, \mathbf{F}, Z_0, \mathbf{O}, \eta)$, where:
  - $\mathbf{Q}$ is the set of states of the automata
  - $\mathbf{I}$ is the alphabet of the input string which will be checked
  - Γ is the alphabet of the symbols on the stack
  - $\delta : \mathbf{Q} \times (\mathbf{I} \cup \epsilon) \times \Gamma \mapsto \mathbf{Q} \times \Gamma^*$ the transition function
  - $q_0 \in \mathbf{Q}$ the (unique) initial state from where the automaton starts
  - $\mathbf{F} \subseteq \mathbf{Q}$ the set of final accepting states of the automaton
  - Z_0 is the symbol which indicates the bottom of the stack
  - $\mathbf{O}$ the output alphabet (may coincide with $\mathbf{I}$)
  - $\eta : \mathbf{Q} \times (\mathbf{I} \cup \epsilon) \times \Gamma \mapsto \mathbf{O}^*$ the transduction function

A simple transducer

$\tau(a^k b^h c^h) = d^{3h} e^k, h, k \geq 1$

- Hint: "one b is worth two d" is a nice strategy
- Notation convention:
  $<input><stack> \mid <stack><stack>, < output >$



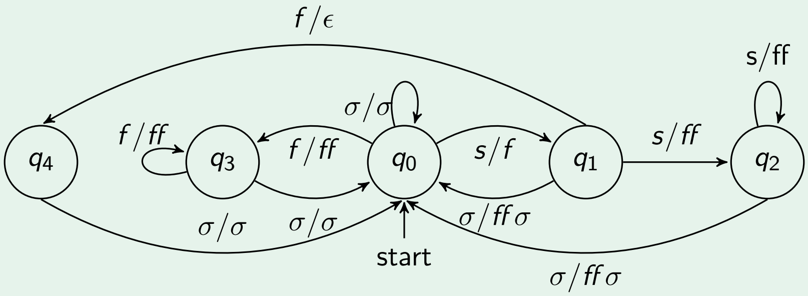
Jovanotti's Words

Suppose we want to encode/decode Jovanotti's words, with dash used as words separator:

- Encoding: replace each s with a f (e.g., $sasso \rightarrow faffo$).
- However, with such a replacement, during decoding we cannot know if an f corresponds to an s or an f.
- Instead, consider this encoding: $f^n \mapsto f^{2n}$ and $s^n \mapsto f^{2n+1}$. In this way, an even number of f denotes a sequence of f, while an odd number denotes a sequence of s
- Note that the pair fs cannot appear, as well as ss^+f and sff^+(at least in Italian words).
- There is still to deal with the sequence sf: we can just replace it with a single f
- The language is thus $L = \{(x.-)^+ \mid \neg\exists y, z(x = y.fs.z \vee y = y.ss^+f.z \vee x = y.sff^+.z)\}$

Jovanotti's Words

Encoder



- Symbol σ denotes an arbitrary letter, except for s and f.
- A final state to recognize – at the end of a word is missing to avoid a lot of crossing edges. This state has:
  - same outgoing transitions as q_0
  - same ingoing transitions as q_0, with σ being replaced by –

Jovanotti's Words

Decoder

Determining the translation of a sequence of f into a sequence of s or a sequence of f is equivalent to determine the parity of the lenght of the sequence of f in the encoded message

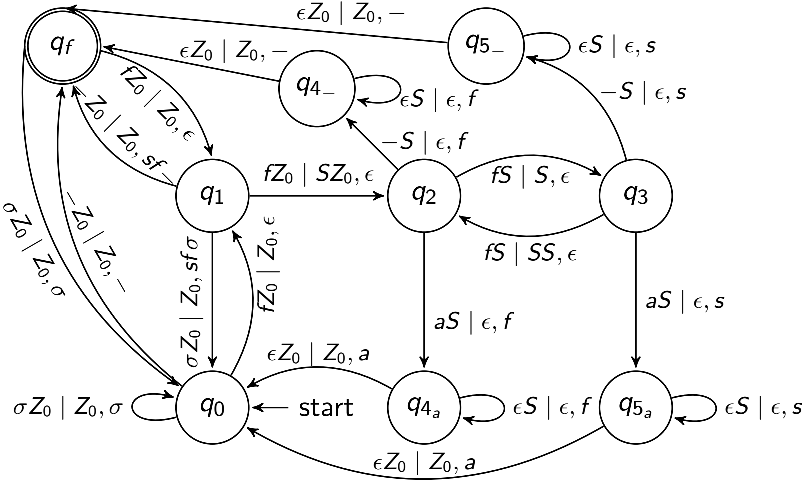
$\Downarrow$

While determining the parity can be done with a FSA, we need to store the number of f or s characters to be written once the parity has been determined

$\Downarrow$

We need a PDA based transducer!

Jovanotti's Words: Decoder



We use additional states $q_{4_b}, q_{4_c}, \ldots$ (resp. $q_{5_b}, q_{5_c} \ldots$), not depicted, to store the character to be written after sequence of f (resp. s).