Factor Oracle for Machine Improvisation

Jaime Arias

Université de Bordeaux, LaBRI, UMR 5800
Inria - Bordeaux Sud-Ouest

Septembre 2016

Preliminaries

Preliminaries

Word

A word s is a finite sequence $s = s_1 s_2 \dots s_m$ of length $|s| = m$ on a finite alphabet Σ.

$s =$

a	b	b	c	a	b	c	d	a	b	c

Factor

A word $x \in \Sigma^*$ is a factor of s if and only if s can be written $s = uxv$ with $u, v \in \Sigma^*$. Given integers i, j where $1 \leq i \leq j \leq m$, we denote a *factor* of s as $s[i \dots j] = s_i s_{i+1} \dots s_j$.

$s =$

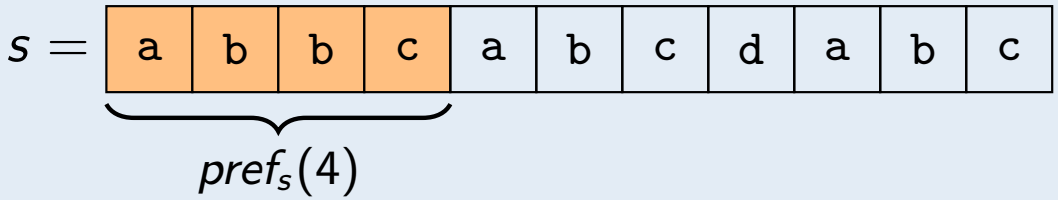
a	b	b	c	a	b	c	d	a	b	c

$s[3, 5]$

Preliminaries

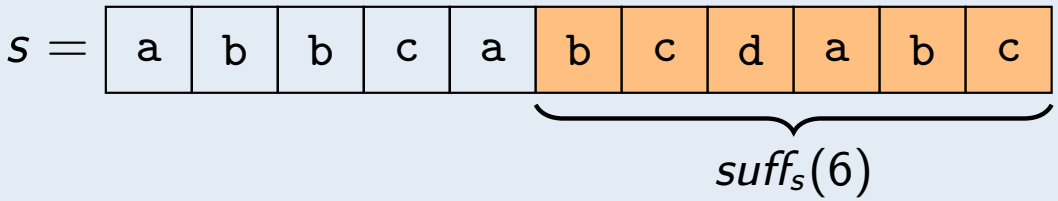
Prefix

A factor x of s is a prefix of s if $s = xu$ with $u \in \Sigma^*$. The ith *prefix* of s, denoted $pref_s(i)$, is the prefix $s[1 \dots i]$.



Suffix

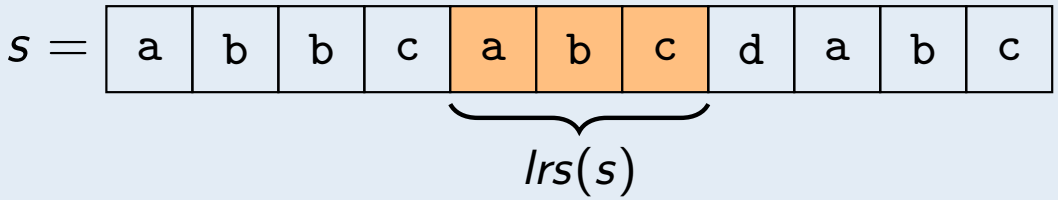
A factor x of s is a suffix of s if $s = ux$ with $u \in \Sigma^*$. The ith *suffix* of s, denoted $suff_s(i)$, is the suffix $s[i \dots m]$.



Preliminaries

Longest Repeated Suffix (LRS)

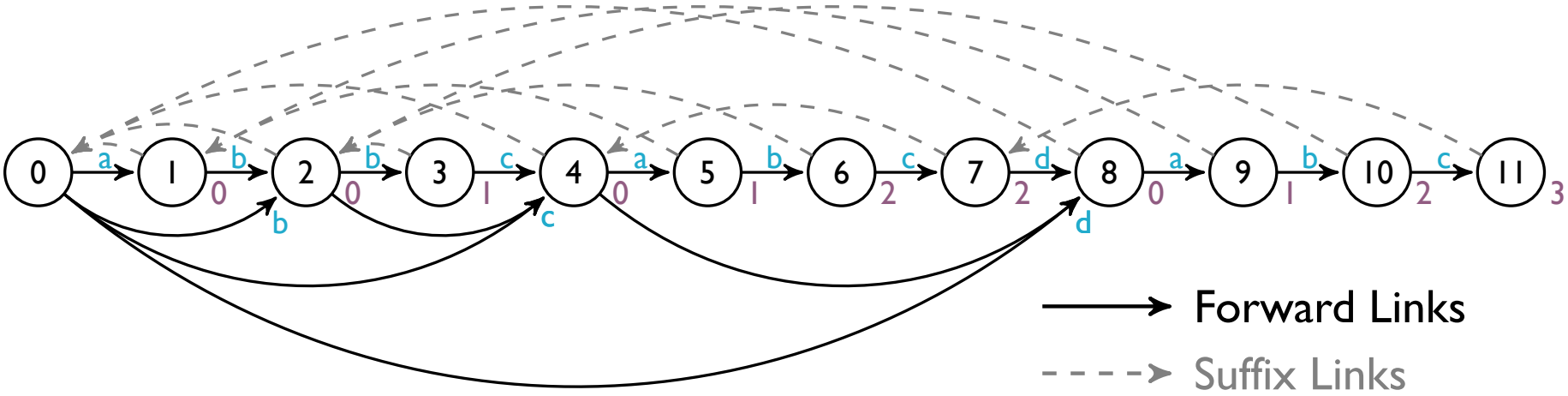
A factor x of s is the longest repeated suffix of s if x is a suffix of s and $|x|$ is maximal.



Factor Oracle

Factor Oracle

Overview

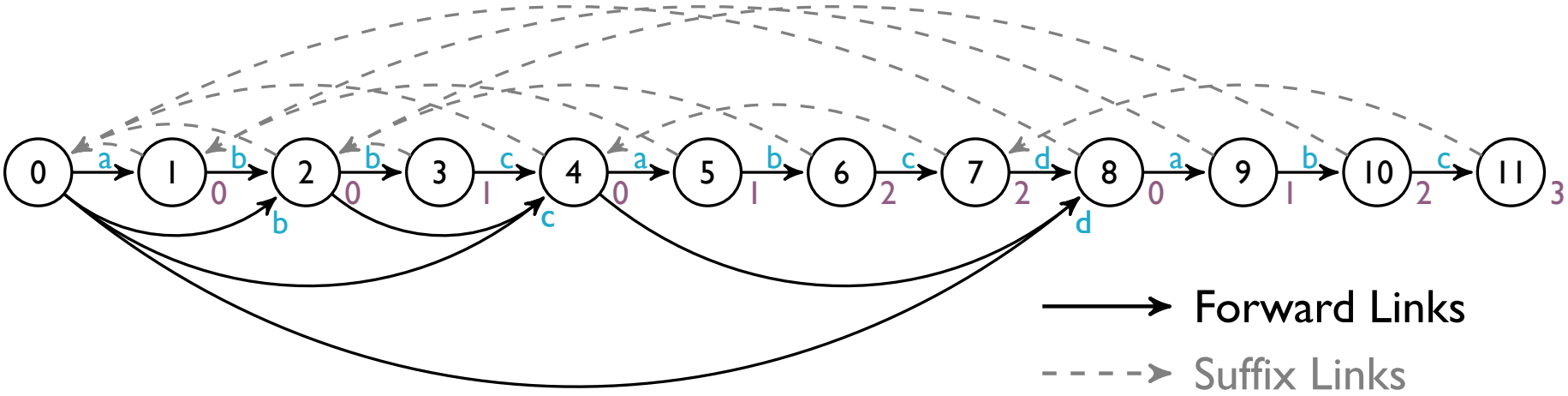


Factor Oracle

The factor oracle of a word s of length m is a *deterministic finite automaton* (Q, q_0, F, δ) where $Q = \{0, 1, \ldots, m\}$ is the set of states, $q_0 = 0$ is the starting state, $F = Q$ is the set of terminal states and δ is the transition function.

Factor Oracle

Overview

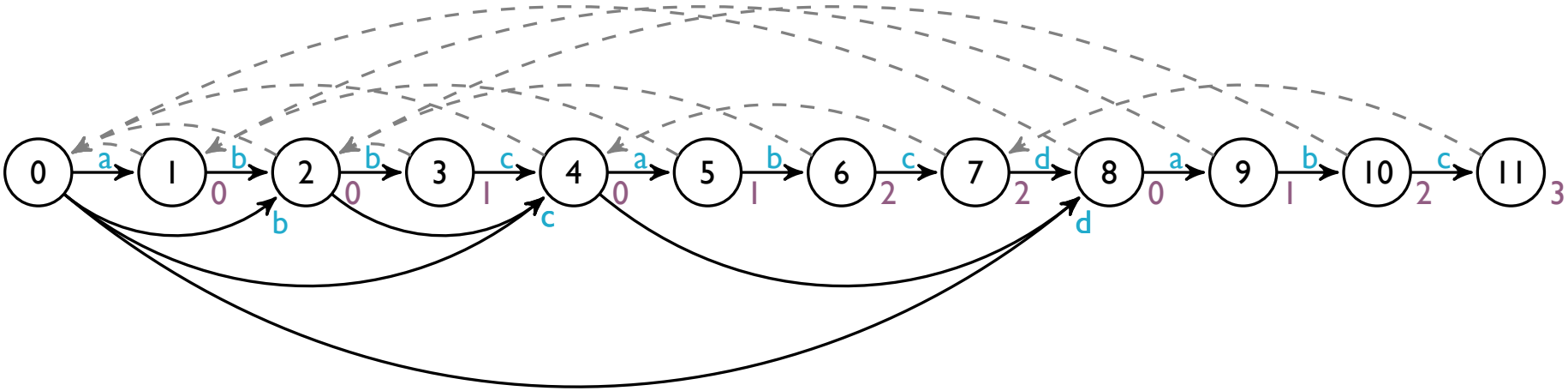


Suffix Link

The suffix link of a state i of the factor oracle of a word s, is equal to the state in which the *longest repeated suffix (lrs)* of $s[1 \dots i]$ is recognized.

Factor Oracle

Overview

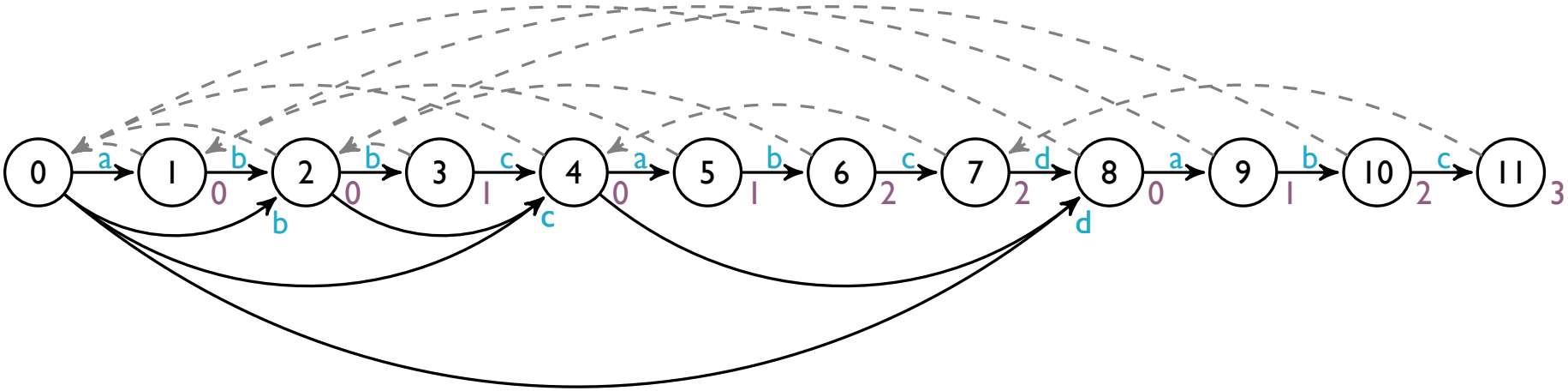


Suffix Links

- $s =$ abbcabcdabc

Factor Oracle

Overview

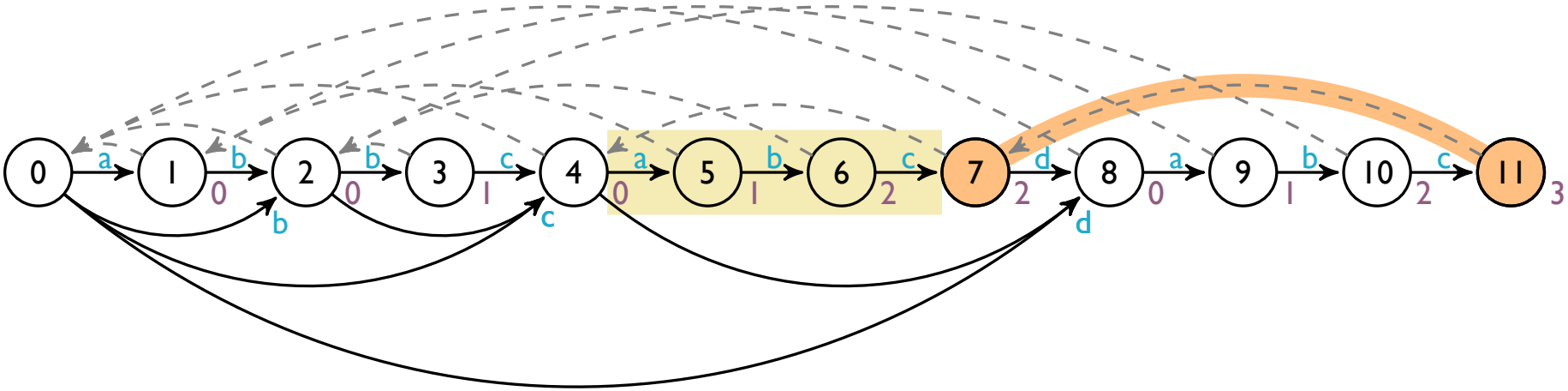


Suffix Links

- $s = \texttt{abbcabcdabc}$
- $lrs(s) = \texttt{abc}$

Factor Oracle

Overview



Suffix Links

- $s = \texttt{abbcabcdabc}$
- $lrs(s) = \texttt{abc}$
- $S(11) = 7$

Factor Oracle

Algorithm - Improvisation

Algorithm 1 FO-Generate function

Require: Oracle $P = p_1, p_2 \dots p_m$ in active state i, a generated sequence v, and a continuation parameter $0 \leq q \leq 1$.

1: Generate uniformly distribute random number u
2: **if** $u < q$ **then**
3: $\quad i \leftarrow i + 1 \qquad v \leftarrow vp_i$
4: **else**
5: $\quad$ Choose at random a symbol $\sigma \in \{\sigma_j \mid \delta(S(i), \sigma_j) \neq \perp\}$
6: $\quad i \leftarrow \delta(S(i), \sigma) \qquad v \leftarrow v\sigma$
7: **end if**
8: **return** Sequence v

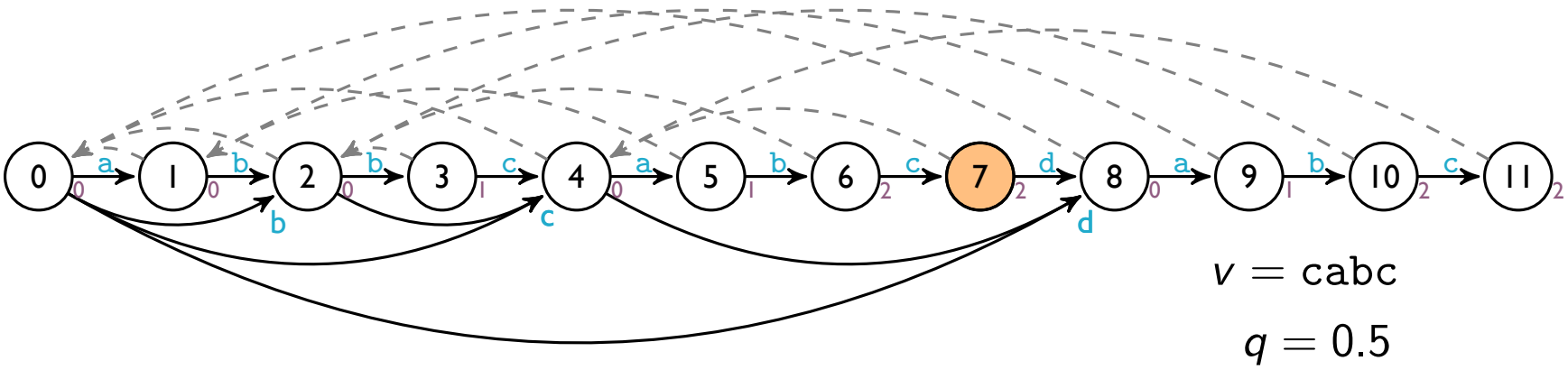
Factor Oracle

Algorithm - Improvisation

Algorithm 1 FO-Generate function

Require: Oracle $P = p_1, p_2 \dots p_m$ in active state i, a generated sequence v, and a continuation parameter $0 \leq q \leq 1$.

1: Generate uniformly distribute random number u
2: **if** $u < q$ **then**
3: $\quad i \leftarrow i + 1 \qquad v \leftarrow vp_i$
4: **else**
5: $\quad$ Choose at random a symbol $\sigma \in \{\sigma_j \mid \delta(S(i), \sigma_j) \neq \perp\}$
6: $\quad i \leftarrow \delta(S(i), \sigma) \qquad v \leftarrow v\sigma$
7: **end if**
8: **return** Sequence v



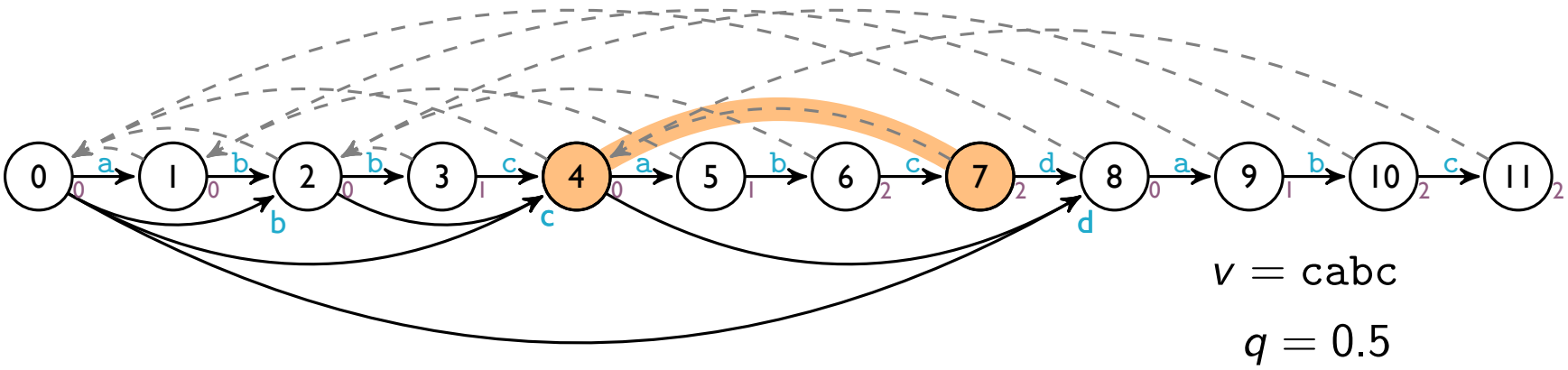
Factor Oracle

Algorithm - Improvisation

Algorithm 1 FO-Generate function

Require: Oracle $P = p_1, p_2 \dots p_m$ in active state i, a generated sequence v, and a continuation parameter $0 \leq q \leq 1$.

1: Generate uniformly distribute random number u
2: **if** $u < q$ **then**
3: $\quad i \leftarrow i + 1 \qquad v \leftarrow vp_i$
4: **else**
5: $\quad$ Choose at random a symbol $\sigma \in \{\sigma_j \mid \delta(S(i), \sigma_j) \neq \perp\}$
6: $\quad i \leftarrow \delta(S(i), \sigma) \qquad v \leftarrow v\sigma$
7: **end if**
8: **return** Sequence v



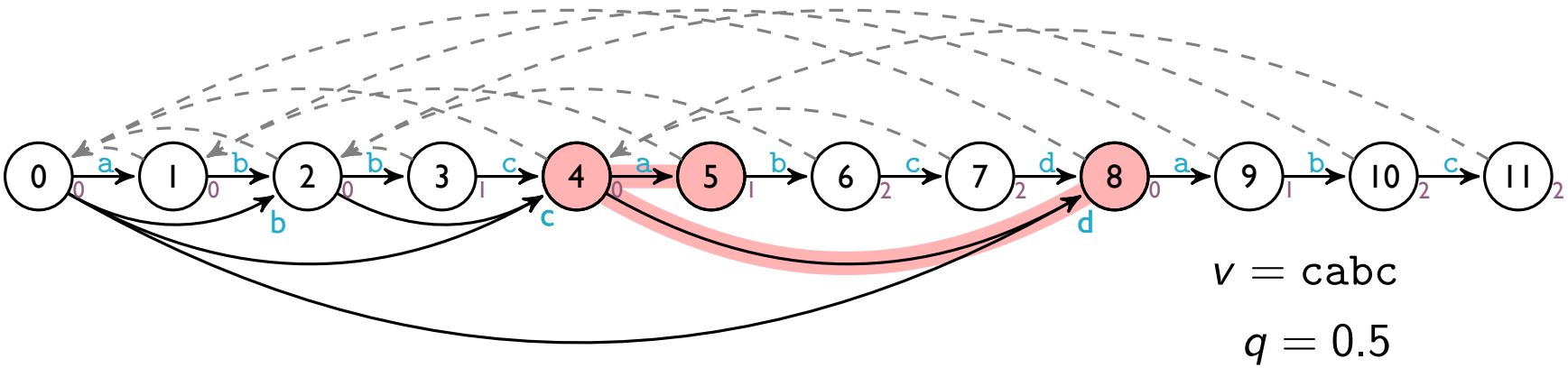
Factor Oracle

Algorithm - Improvisation

Algorithm 1 FO-Generate function

Require: Oracle $P = p_1, p_2 \dots p_m$ in active state i, a generated sequence v, and a continuation parameter $0 \leq q \leq 1$.

1: Generate uniformly distribute random number u
2: **if** $u < q$ **then**
3: $\quad i \leftarrow i + 1 \qquad v \leftarrow vp_i$
4: **else**
5: $\quad$ Choose at random a symbol $\sigma \in \{\sigma_j \mid \delta(S(i), \sigma_j) \neq \perp\}$
6: $\quad i \leftarrow \delta(S(i), \sigma) \qquad v \leftarrow v\sigma$
7: **end if**
8: **return** Sequence v



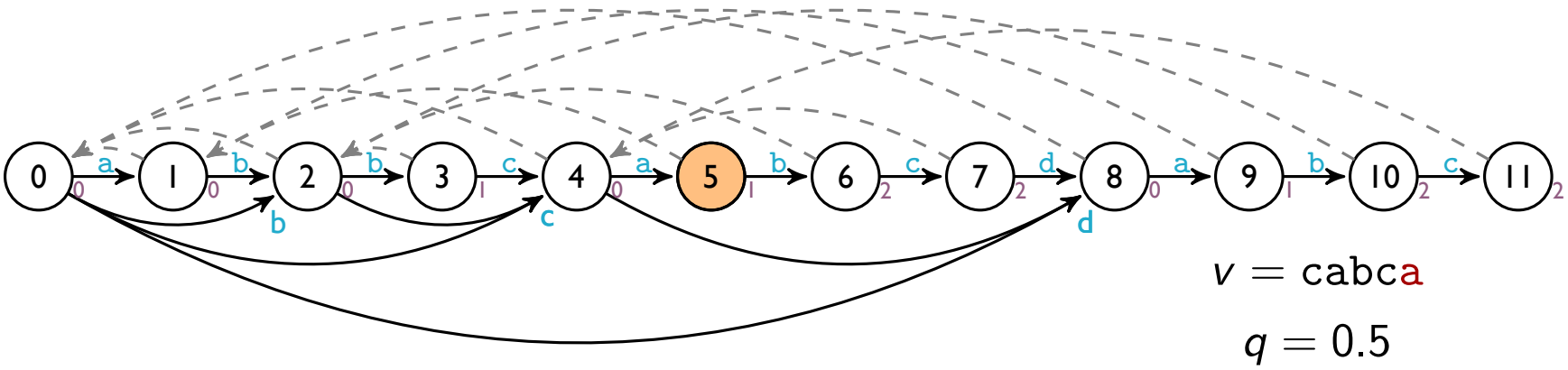
Factor Oracle

Algorithm - Improvisation

Algorithm 1 FO-Generate function

Require: Oracle $P = p_1, p_2 \dots p_m$ in active state i, a generated sequence v, and a continuation parameter $0 \leq q \leq 1$.

1: Generate uniformly distribute random number u
2: **if** $u < q$ **then**
3: $\quad i \leftarrow i + 1 \qquad v \leftarrow vp_i$
4: **else**
5: $\quad$ Choose at random a symbol $\sigma \in \{\sigma_j \mid \delta(S(i), \sigma_j) \neq \perp\}$
6: $\quad i \leftarrow \delta(S(i), \sigma) \qquad v \leftarrow v\sigma$
7: **end if**
8: **return** Sequence v



Thank you for your attention! ☺

Factor Oracle for Machine Improvisation

Jaime Arias

Université de Bordeaux, LaBRI, UMR 5800
Inria - Bordeaux Sud-Ouest

Septembre 2016