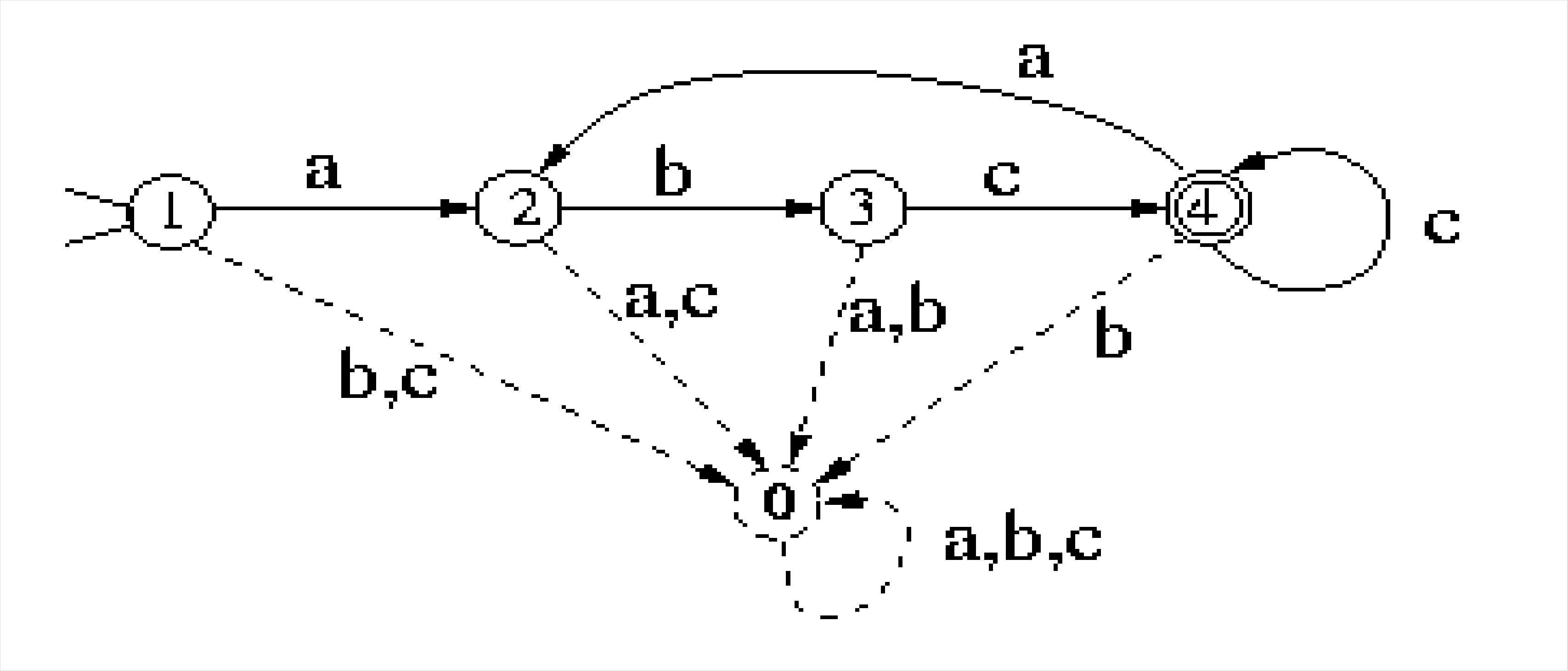
String matching



Announcements

Programming assignment due Sunday

Prefix vs suffix

$\sqsubset$

w is a prefix of x=w [x, means exists y s.t. wy = x (also implies $|w| \leq |x|$)

(w] y = w is a suffix of x)

An easy way to remember prefix vs suffix is: prefix = [, which looks like beginning of an array (similar suffix)

Finite Automata

A finite automata has 5 parts:

(1) A set of states Q

(2) An initial state q_0

(3) Some accepting states, A subset Q

(4) An alphabet, $\sum$

(5) A transition function δ, from

$Q \times \sum$ to Q, namely $\delta(q,a)=\sigma(P_q a)$

Finite Automata

Let $\sigma(x) = \max \{k : P_k \,]\, x\}$

So σ is the longest prefix of P that is also a suffix of x:

P = {a, b, a, a, b, c, a}

σ(b a a c b a b) = 2 (all longer bad)

Finite Automata

Compute-Transition-Function(P, $\sum$)

for q = 0 to |P|

    for each a in $\sum$

        k = min(|P|, q+1)//end P or q

        while: not P_k] $P_q a$

            k = k – 1

        $\delta(q,a) = k$ // runtime?

Finite Automata

$O(|P|^3|\Sigma|)$, but can get to $O(|P|\ |\Sigma|)$ if smart

$|P|$ - outside loop

$|\Sigma|$ - outside loop

$|P|$ - repeat runs at most $|P|$ times

$|P|$ - P_k] $P_q a$ checks $O(|P|)$ chars

Finite Automata

```
Finite-Automaton-Matcher(T,δ,|P|)
q=0 // q is state
for i = 1 to |T|
   q = δ(q,T[ i ])
   if q == |P|
      print "Pattern occurs at shift" i-|P|

Runtime = O(|T|)
```

Finite Automata

1, 2, 3, 4, 5, 6, 7

P = {a, b, a, a, b, c, a}, then δ is:

	0	1	2	3	4	5	6	7
a	1	1	3	4	1	3	7	1
b	0	2	0	2	5	0	0	2
c	0	0	0	0	0	6	0	0

(see FAsigma.py)

Finite Automata

	0	1	2	3	4	5	6	7
a	1	1	3	4	1	3	7	1
b	0	2	0	2	5	0	0	2
c	0	0	0	0	0	6	0	0

S = {a, b, c, a, b, a, a, b, c, a, c, a}

Start 0, see S_1='a', goto 1...

At 1, see S_2='b', goto 2...

At 2, see S_3='c', goto 0...

At 0, see S_4='a', goto 1...

Finite Automata

	0	1	2	3	4	5	6	7
a	1	1	3	4	1	3	7	1
b	0	2	0	2	5	0	0	2
c	0	0	0	0	0	6	0	0

S = {a, b, c, a, b, a, a, b, c, a, c, a}

At 1, see S_5='b', goto 2...

At 2, see S_6='a', goto 3...

At 3, see S_7='a', goto 4...

At 4, see S_8='b', goto 5...

Finite Automata

	0	1	2	3	4	5	6	7
a	1	1	3	4	1	3	7	1
b	0	2	0	2	5	0	0	2
c	0	0	0	0	0	6	0	0

S = {a, b, c, a, b, a, a, b, c, a, c, a}

At 5, see S_9='c', goto 6...

At 6, see S_{10}='a', goto 7... MATCH!

At 7, see S_{11}='a', goto 1...

At 1, see S_{12}='c', goto 0...

Finite Automata

	0	1	2	3	4	5	6	7
a	1	1	3	4	1	3	7	1
b	0	2	0	2	5	0	0	2
c	0	0	0	0	0	6	0	0

S = {a, b, c, a, b, a, a, b, c, a, c, a}

At 0, see S_{13}='a', goto a...

Done, one match found ending at S_{10} (so match starts S_4)

Finite Automata

You try it!

P={a, b, a, a}

S={a, a, b, a, c, a, a, b, a, a, b, a, a, a}

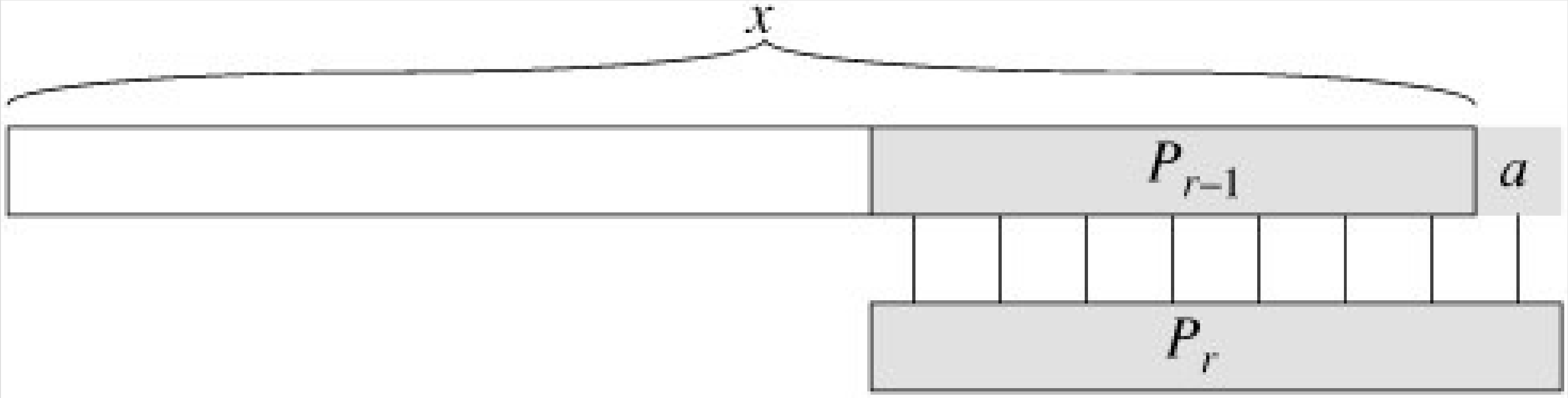
What is automata?

Where are matches?

FA correctness

Lemma 32.2: $\sigma(xa) \leq \sigma(x) + 1$

Obvious...



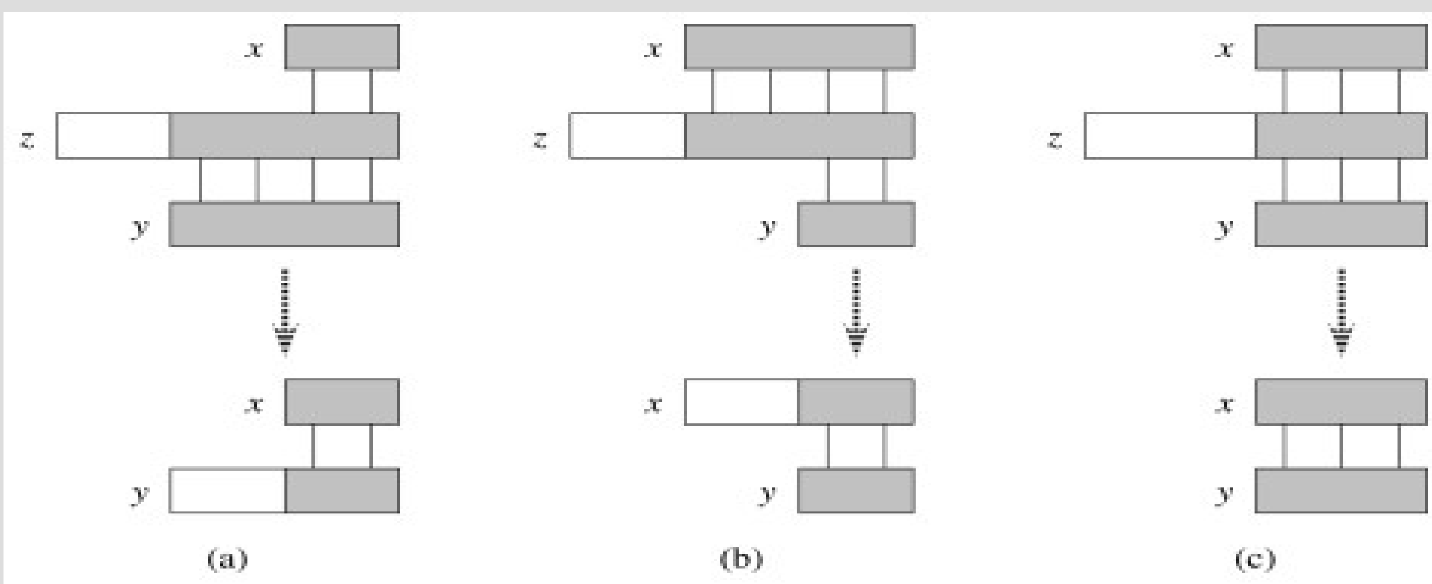
FA correctness

If x] z and y] z, then:

(a) If $|x| \le |y|$, x] y

(b) If $|y| \le |x|$, y] x

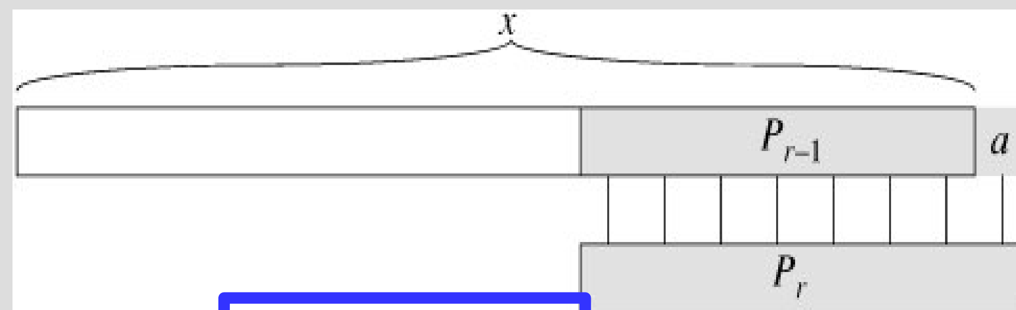
(c) If $|x| = |y|$, $x = y$



FA correctness

Lemma 32.3: if $q=\sigma(x)$,

then $\sigma(xa)=\sigma(P_q a)$



Proof:

P_q] x by def of $q=\sigma(x)$, then $\boxed{P_q a \ ]\ xa}$

Let $r=\sigma(xa)$ then P_r] xa and $r \leq q+1$

So $|P_r| < |P_q a|$ means P_r] $P_q a$

$\sigma(xa)\leq\sigma(P_q a)$,

$\boxed{P_q a \ ]\ xa,}$ so also $\sigma(P_q a)\leq\sigma(xa)$

FA correctness

Theorem 32.4: if Φ is the final-state function, then $\Phi(T_i) = \sigma(T_i)$

Base: $T_0 = \varepsilon$, so $\Phi(T_0)=0=\sigma(T_0)$

$\Phi(T_{i+1}) = \Phi(T_i a) = \delta(\Phi(T_i),a)$

$= \sigma(P_{\Phi(Ti)}a) = \sigma(T_i a) = \sigma(T_{i+1})$

Knuth-Morris-Pratt

Faster computation by using pattern symmetries within itself (vs transitions for each char/state)

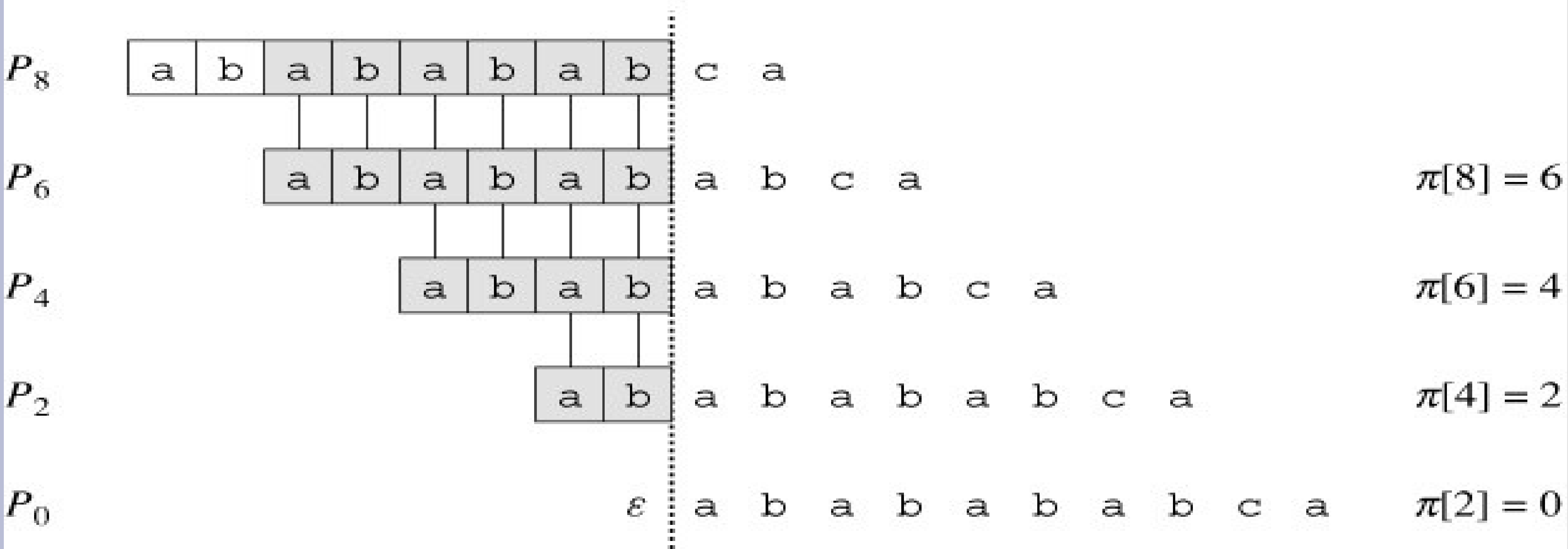
The function π does this, namely
$\pi(q) = \max(k : k < q \text{ and } P_k] P_q)$

Namely, π finds shifts of P on itself

Knuth-Morris-Pratt

i	1	2	3	4	5	6	7	8	9	10
$P[i]$	a	b	a	b	a	b	a	b	c	a
$\pi[i]$	0	0	1	2	3	4	5	6	0	1

(a)



(b)

Knuth-Morris-Pratt

i	1	2	3	4	5	6	7
P[i]	a	b	a	a	b	c	a
π(i)	0	0	1	1	2	0	1

(See: FAsigma.py ... again)

Knuth-Morris-Pratt

```
KMP-Matcher(T,P,π) // runtime?
q = 0
for i = 1 to |T|
  while q > 0 and P[q+1] ≠ T[ i ]
    q = π[q]
  if P[q+1] == T[ i ], then q = q+1
  if q == |P|
    match found, and set q = π[q]
```

Knuth-Morris-Pratt

The while loop decreases q, so it can only run as many times as q increases

q increases only if match in T, so at most |T| times

O(|T| + |T|) = O(|T|)
(why not |T|*|T|?)

Knuth-Morris-Pratt

```
Compute-Prefix-Function(P)
k = 0, π[1] = 0
for q = 2 to |P|
    while k > 0 and P[k+1] ≠ P[q]
        k = π[k]
    if P[k+1] == P[q]
        k = k+1
    π[q]=k              // Runtime = O(|P|)
```

KMP correctness

Let $\pi^*[q] = \{\pi[q], \pi[\pi[q]], \ldots 0\}$

Lemma 32.5: $\pi^*[q] = \{k : k < q$ and $P_k \,] \, P_q \}$

Remember:

$\pi(q) = \max(k : k < q$ and $P_k \,] \, P_q)$,

so fairly obvious (see next slide)

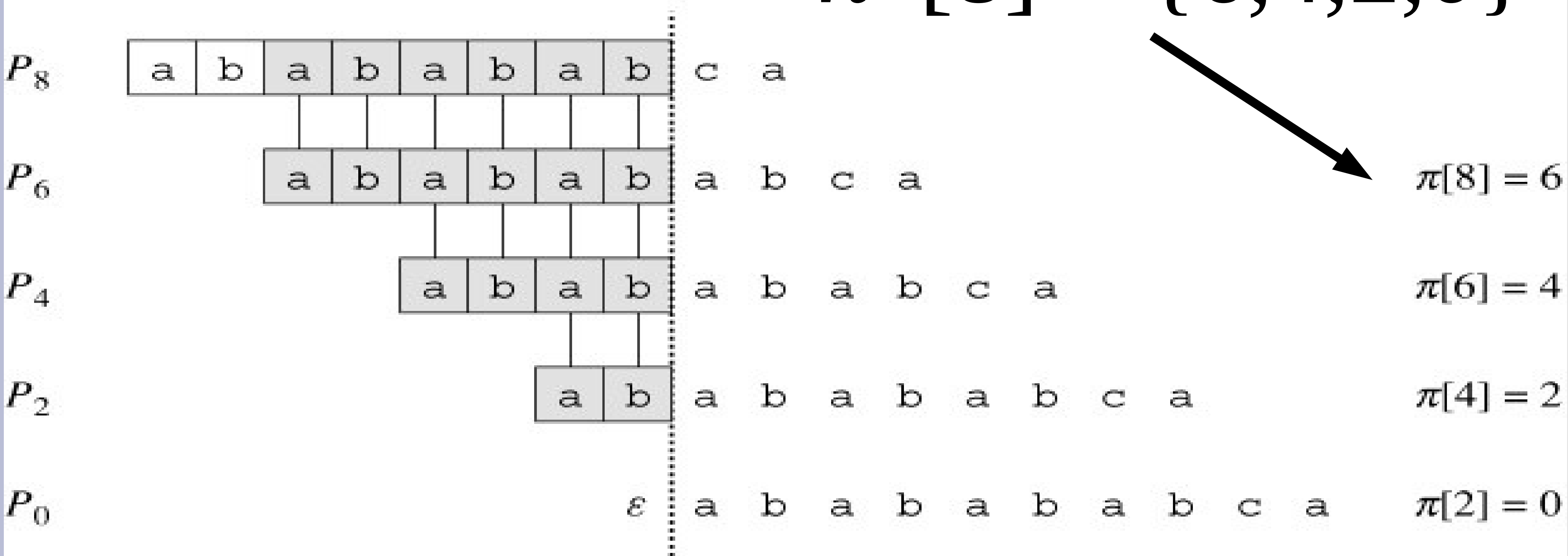
(Tip: prove 2 sets equal by showing A subset B and B subset A)

KMP correctness

i	1	2	3	4	5	6	7	8	9	10
$P[i]$	a	b	a	b	a	b	a	b	c	a
$\pi[i]$	0	0	1	2	3	4	5	6	0	1

(a)

$\pi^*[8] = \{6,4,2,0\}$



(b)

KMP correctness

Lemma 32.6: if $\pi[q] > 0$, then $\pi[q]-1$ in $\pi^*[q-1]$

Proof: $\pi[q] < q$ and $P_{\pi[q]}] P_q$, so $\pi[q] - 1 < q - 1$ and $P_{\pi[q]-1}] P_{q-1}$ (we know $\pi[q] > 0$, so we can drop a char)

Previous lemma says: $\pi^*[q] = \{k : k < q \text{ and } P_k] P_q \}$, above let $q=q-1$, $k=\pi[q]-1$, then done

KMP correctness

Let E_{q-1}={k in π*[q-1] : P[k+1]=P[q]}

Corollary 32.7: $\pi[q]$ = {0 or 1+max{k in E_{q-1}} if E_{q-1} not empty}

Proof:

Case 1: E_{q-1} empty, no match, so 0

Case 2: By def of E_{q-1}, k+1 < q and $P_{k+1}] P_q$ implies $\pi[q] \geq 1+\max\{k \text{ in } E_{q-1}\}$

KMP correctness

$(E_{q-1}=\{k \text{ in } \pi^*[q-1] : P[k+1]=P[q]\})$

Case 2 (cont): $\pi[q] \geq 1+\max\{k \text{ in } E_{q-1}\}$

Let $r = \pi[q] - 1$, then P_{r+1}] P_q so

$P[r+1] = P[q]$. Lemma 32.6 says

r in $\pi^*[q-1]$, so r in E_{q-1}.

Thus $\pi[q] \leq 1+\max\{k \text{ in } E_{q-1}\}$

Thus $\pi[q]=1+\max\{k \text{ in } E_{q-1}\}$

KMP correctness

$k=\pi[q-1]$ at the start of the for loop in Compute-Prefix-Function alg

The while loop finds max{k in E_{q-1}} and adds one for Corollary 32.7

If there k=0, then either the max was 0 and it will be incremented to 1 or no match and will stay 0

KMP correctness

KMP alg correctness (map to FA alg):

Base: both start with q=0

Step ($q'=\sigma(T_{i-1})$):

Case $\sigma(T_i)=0$: q=0 and same

Case $\sigma(T_i)=q'+1$: while does not run, then increases q, so $q=q'+1=\sigma(T_i)$

(continued)

KMP correctness

Step: $q'=\sigma(T_{i-1})$, Case $0<\sigma(T_i)<q'$:

while loop terminates when

$P[q+1]=T[i]$, so $q+1 = \sigma(P_{q'}T[i])$

$=\sigma(T_{i-1}T[i])$

$=\sigma(T_i)$, then q is incremented so...

$q=\sigma(T_i)$