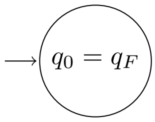
Теория и реализация языков программирования.
Задание 2: НКА и алгоритмы поиска подстрок

Сергей Володин, 272 гр.

задано 2013.09.11

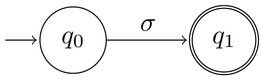
Упражнение 0

Автомат, распознающий $\varnothing$:



У него нет принимающих состояний, поэтому ни одно слово не будет принято.

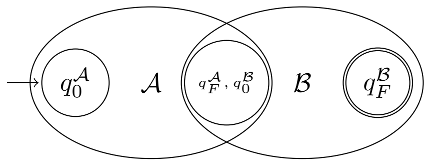
Автомат, распознающий $\{\sigma\}$:



Очевидно, автомат примет σ: $(q_0, \sigma) \vdash (q_1, \varepsilon)$, и $q_1 \in F$. Пусть автомат принял слово w. Поскольку начальных и принимающих состояний по одному, и между ними один переход $\delta(q_0, \sigma) = \{q_1\}$, получаем $(q_0, w) \vdash (q_1, \varepsilon)$. Отсюда $w = \sigma$.

Упражнение 1

$L(\mathcal{A}) = X, L(\mathcal{B}) = Y$. Докажем, что $L(\mathcal{C}) = X \cdot Y$, $\mathcal{C}$ — автомат из условия:



Докажем, что $w \in X \cdot Y \Rightarrow w \in L(\mathcal{C})$:
$w \in X \cdot Y \Rightarrow w = xy, x \in X, y \in Y$. Тогда $x \in L(\mathcal{A})$, то есть, $(q_0^{\mathcal{A}}, x) \vdash^* (q_F^{\mathcal{A}}, \varepsilon)$. Аналогично $(q_0^{\mathcal{B}}, y) \vdash^* (q_F^{\mathcal{B}}, \varepsilon)$.

Пусть для некоторого автомата $(q_1, cx) \vdash (q_2, x)$. Это значит, что $\delta(q_1, c) \ni q_2$. Но отсюда $(q_1, cxy) \vdash (q_2, xy)$, x, y — некоторые слова. Пусть $(q_1, x) \vdash^* (q_2, y)$. Это значит, что существует цепочка $(q_1, x) \vdash \cdots \vdash (q_2, y)$. Применяя утверждение выше, получаем $(q_1, xz) \vdash \cdots \vdash (q_2, yz)$, а отсюда $(q_1, xz) \vdash^* (q_2, yz)$. Очевидно, рассуждение верно и в обратную сторону: если $(q_1, xz) \vdash^* (q_2, yz)$, то $(q_1, x) \vdash^* (q_2, y)$.

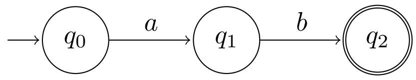
Из предыдущего утверждения получаем для автомата $\mathcal{A}$ $(q_0^{\mathcal{A}}, xy) \vdash^* (q_F^{\mathcal{A}}, y) \equiv (q_0^{\mathcal{B}}, y) \vdash^* (q_F^{\mathcal{B}}, \varepsilon)$, то есть, $(q_0^{\mathcal{A}}, w) \vdash^* (q_F^{\mathcal{B}}, \varepsilon) \Rightarrow w \in L(\mathcal{C})$.
Докажем обратное: $w \in L(\mathcal{C}) \Rightarrow w \in X \cdot Y$.
Пусть $w \in L(\mathcal{C})$. Тогда $(q_0^{\mathcal{A}}, w) \vdash \cdots \vdash (q_F^{\mathcal{B}}, \varepsilon)$. Докажем, что в этой цепочке встретилось состояние $(q_F^{\mathcal{A}}, y)$.
В этой цепочке был переход из $q_{\mathcal{A}} \in Q^{\mathcal{A}}$ в $q_{\mathcal{B}} \in Q^{\mathcal{B}}$, т.к. иначе получим, что все переходы были внутри множества состояний $Q^{\mathcal{A}}$. Поэтому $\exists q^{\mathcal{A}} \exists \sigma \in \Sigma : \delta(q^{\mathcal{A}}, \sigma) \ni q^{\mathcal{B}}$. Но изначально множество $Q^{\mathcal{A}}$ — множество состояний автомата $\mathcal{A}$, поэтому $\delta(Q^{\mathcal{A}} \times \Sigma) \subseteq 2^{Q^{\mathcal{A}}}$. Получаем $q^{\mathcal{B}} \in Q^{\mathcal{A}}$. По условию $Q^{\mathcal{A}} \cap Q^{\mathcal{B}} = \{q_F^{\mathcal{A}}\}$, откуда получаем требуемое: в цепочке встретилось $(q^{\mathcal{A}}, \sigma y) \vdash (q_F^{\mathcal{A}}, y)$.
Таким образом, $(q_0^{\mathcal{A}}, w) \vdash^* (q_F^{\mathcal{A}}, y) \vdash^* (q_F^{\mathcal{B}}, \varepsilon)$. Поскольку y — суффикс w, то $w = xy$. Из доказанного ранее получаем $(q_0^{\mathcal{A}}, x) \vdash^* (q_F^{\mathcal{A}}, \varepsilon)$, откуда $x \in L(\mathcal{A}) = X$. Аналогично $y \in L(\mathcal{B}) = Y$. Найдены $x \in X$ и $y \in Y$: $w = xy$. Отсюда $w \in X \cdot Y$.

Упражнение 2

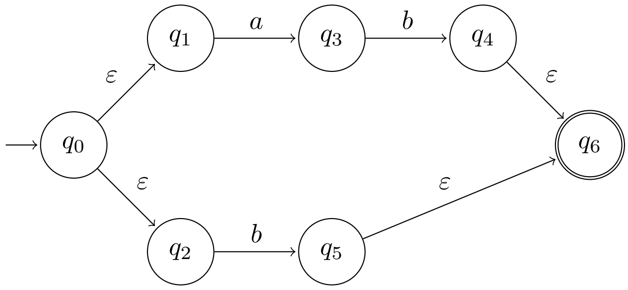
Упражнение 3

Задача 1

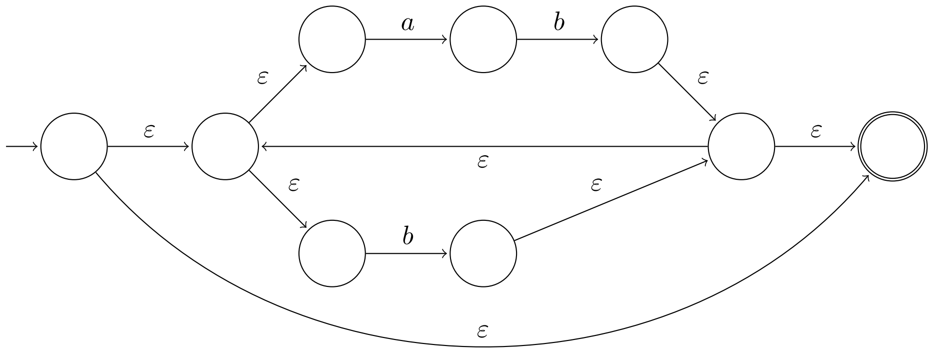
Пользуясь доказанным ранее, построим автомат для ab:



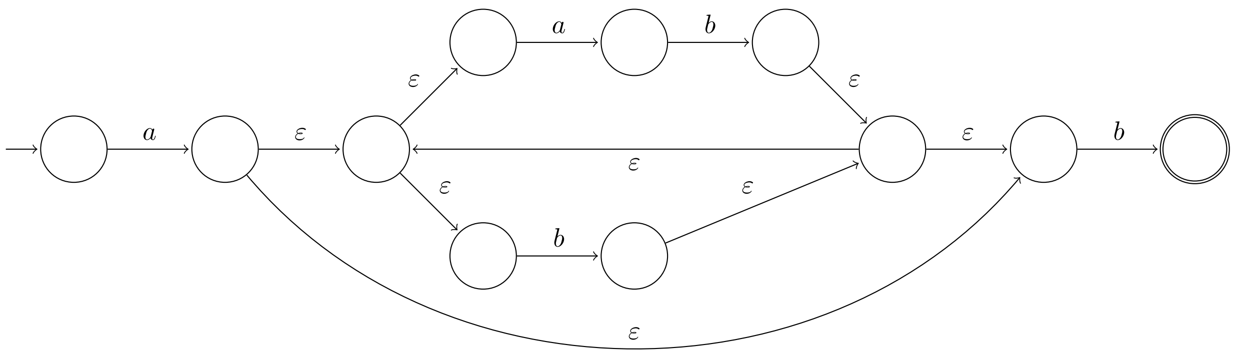
Для $ab|b$:



Для $(ab|b)^*$

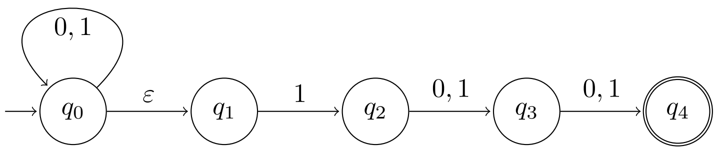


Для $a(ab|b)^*b$:



Задача 2

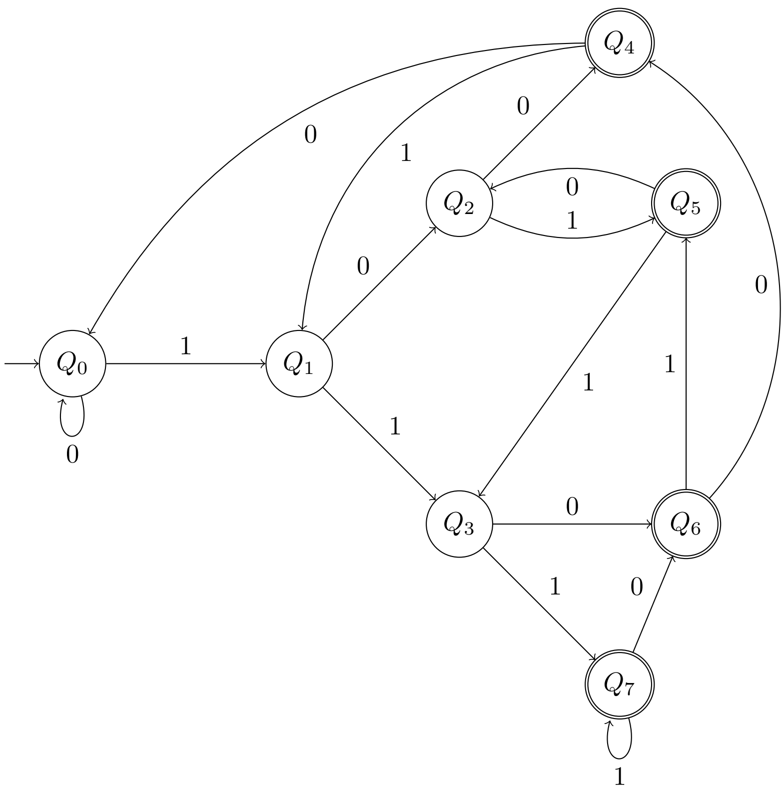
a. Докажем, что автомат $\mathcal{A}$ распознает L_3:



1. $w \in L_3$. Тогда $w[n-3] = 1 \Leftrightarrow w = x1\sigma_1\sigma_2$. $x \in \Sigma^* \Rightarrow$ после обработки x автомат может оказаться в q_0: $(q_0, x1\sigma_1\sigma_2) \vdash^* (q_0, 1\sigma_1\sigma_2)$. Далее $(q_0, 1\sigma_1\sigma_2) \vdash (q_1, 1\sigma_1\sigma_2) \vdash (q_2, \sigma_1\sigma_2)$. $\sigma_1, \sigma_2 \in \Sigma \Rightarrow (q_2, \sigma_1\sigma_2) \vdash (q_3, \sigma_2) \vdash (q_4, \varepsilon)$. Таким образом, $(q_0, w) \vdash^* (q_4, \varepsilon) \Rightarrow w \in L(\mathcal{A})$.
2. $w \in L(\mathcal{A}) \Rightarrow (q_0, w) \vdash^* (q_4, \varepsilon)$. Поскольку переход в q_4 только один: $\exists! q : \delta(q, \sigma) \ni q_4, q = q_3$, получаем, что в цепочке конфигураций на последнем месте переход $(q_3, \sigma) \vdash (q_4, \varepsilon)$, σ — суффикс w. Аналогично получаем $(q_1, 1\sigma_1\sigma_2) \vdash^* (q_4, \varepsilon)$, $1\sigma_1\sigma_2$ — суффикс w, откуда $w \in L_3$.

b. Построим ДКА $\mathcal{B}$ по НКА $\mathcal{A}$:

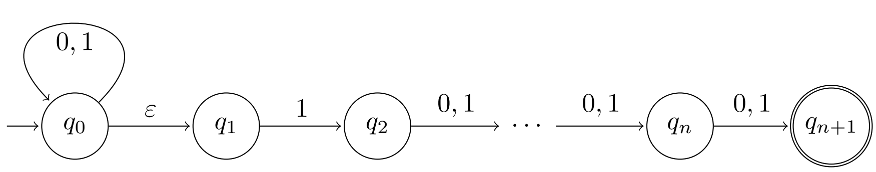
Q	q	$0\,(q)$	$0\,(Q)$	$1\,(q)$	$1\,(Q)$
Q_0	q_0, q_1	q_0, q_1	Q_0	q_0, q_1, q_2	Q_1
Q_1	q_0, q_1, q_2	q_0, q_1, q_3	Q_2	q_0, q_1, q_2, q_3	Q_3
Q_2	q_0, q_1, q_3	q_0, q_1, q_4	Q_4	q_0, q_1, q_2, q_4	Q_5
Q_3	q_0, q_1, q_2, q_3	q_0, q_1, q_3, q_4	Q_6	q_0, q_1, q_2, q_3, q_4	Q_7
Q_4	q_0, q_1, q_4	q_0, q_1	Q_0	q_0, q_1, q_2	Q_1
Q_5	q_0, q_1, q_2, q_4	q_0, q_1, q_3	Q_2	q_0, q_1, q_2, q_3	Q_3
Q_6	q_0, q_1, q_3, q_4	q_0, q_1, q_4	Q_4	q_0, q_1, q_2, q_4	Q_5
Q_7	q_0, q_1, q_2, q_3, q_4	q_0, q_1, q_3, q_4	Q_6	q_0, q_1, q_2, q_3, q_4	Q_7



Заметим, что в построенном автомате $8 = 2^3$ состояний.

Задача 3

В исходном автомате $\mathcal{A} : L(\mathcal{A}) = L_n$ будет $n+2$ состояний (при $n = 3$ состояний 5, при увеличении n на 1 количество состояний также будет увеличиваться на 1, так как в конце слова должен стоять еще один произвольный символ).



(Идея из Хопкрофта, 2.3.6: плохой случай для конструкции подмножеств)

Предположим, что в ДКА $\mathcal{B} : L(\mathcal{B}) = L_n$ меньше 2^n состояний. Тогда по принципу Дирихле (т.к. состояний меньше, чем строк из n символов) $\exists a = a_1 \dots a_n \, \exists b = b_1 \dots b_n : a \neq b$ и $(Q_0, q) \vdash^* (Q, x), (Q_0, b) \vdash^* (Q, y)$, то есть, после обработки a или b автомат переходит в состояние Q. Поскольку $a \neq b$, $\exists i : a_i \neq b_i$.

1. $i = 1$. Это значит, что отличаются первые символы строк. Без ограничения общности, $a_1 = 1, b_1 = 0$. Но тогда $a \in L(\mathcal{B})$, так как n-й символ с конца строки a равен 1. Поэтому $Q \in F_{\mathcal{B}}$.
   Аналогично, так как $b_1 = 0$, $b \notin L(\mathcal{B}) \Rightarrow Q \notin F_{\mathcal{B}}$ — противоречие.

2. $i > 1$. Тогда, без ограничения общности, $a_i = 1, b_i = 0$. Дополним строки a и b чем-либо (например, нулями) справа так, чтобы i-й символ стал n-м: $a' = a_1 \dots a_i \dots a_n \underbrace{0 \dots 0}_{i-1}, b'$ — аналогично. Тогда $a' \in L(\mathcal{B}), b' \notin L(\mathcal{B})$.

   Но $(Q_0, a') \vdash^* (Q, 0 \dots 0) \vdash^* (P, \varepsilon)$,
   $(Q_0, b') \vdash^* (Q, 0 \dots 0) \vdash^* (P, \varepsilon)$, откуда состояние P должно быть как принимающим, так и не принимающим — противоречие.

Получаем, что в НКА $\mathcal{A}$ $O(n)$ состояний, а в любом ДКА $\mathcal{B} : L(\mathcal{B}) = L_n$ 2^n состояний. В том числе 2^n состояний будет у того ДКА, который построен по $\mathcal{A}$ при помощи алгоритма■

Задача 4

Построим ДКА $\mathcal{B}$ по НКА $\mathcal{A}$:

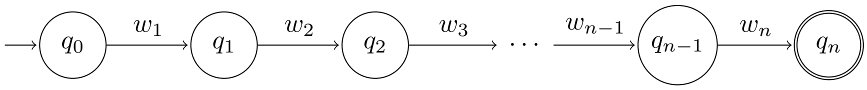
Q	q	a	b
Q_0	q_0	Q_1	Q_0
Q_1	q_0, q_1	Q_1	Q_2
Q_2	q_0, q_2	Q_3	Q_0
Q_3	q_0, q_1, q_3	Q_1	Q_4
Q_4	q_0, q_2, q_4	Q_5	Q_0
Q_5	q_0, q_1, q_3, q_5	Q_1	Q_6
Q_6	q_0, q_2, q_4, q_6	Q_7	Q_8
Q_7	q_0, q_1, q_3, q_5, q_6	Q_9	Q_6
Q_8	q_0, q_6	Q_9	Q_8
Q_9	q_0, q_1, q_6	Q_9	Q_{10}
Q_{10}	q_0, q_2, q_6	Q_{11}	Q_8
Q_{11}	q_0, q_1, q_3, q_6	Q_9	Q_6

Начальное состояние — Q_0, принимающие — $\{Q_6, Q_7, Q_8, Q_9, Q_{10}, Q_{11}\}$.

Задача 5

Задача 6

Пусть $n = |w|$. Тогда определим $Q = \{q_0, q_1, \ldots, q_n\}$, $F = \{q_n\}$.



Определим $l'(i) = |l(w_1 \ldots w_i)|$.
Определим функцию $f(i, \sigma)$ — длина максимального префикса слова $w_1 \ldots w_i$, который совпадает с его суффиксом, причем после префикса стоит символ σ. Построение f аналогично построению префикс-функции: пусть построены значения при $i < m$. Построим для m: рассмотрим следующий символ w_{m+1}. Если $w_{m+1} = \sigma$, то $f(m, \sigma) = m$, иначе найдем максимальный несобственный префикс слова $w_1 \ldots w_m$, являющийся его суффиксом, причем после префикса следует σ. Длина этого суффикса — $f(l'(m), \sigma)$ — уже найдена (т.к. $l'(m) < m$). Определим $f(0, \sigma) = -1$, если $w_1 \neq \sigma$.

Определим функцию переходов δ через f: $\delta(q_i, \sigma) = \begin{cases} \{q_{i+1}\} & \text{если } w_{i+1} = \sigma \\ \{q_{f(i,\sigma)+1}\} & \text{иначе} \end{cases}$.

Свойство: номер состояния — длина максимального префикса слова w, который входит во входное слово, причем вхождение заканчивается на рассматриваемом автоматом символе.

Первый случай, очевидно, корректный (сохраняет свойство выше): если совпало i символов, и следующий символ σ совпадает с w_{i+1}, то всего совпадает $i+1$ символ (и не больше) $\Rightarrow$ переход в q_{i+1}.

Во втором случае следующий входной символ отличается от w_{i+1}. Поскольку текущий символ — σ, необходимо рассматривать префиксы, заканчивающиеся на этот символ. Максимальный префикс p_2, требуемый свойством в текущей конфигурации, очевидно, по длине не больше $|p_1| + 1$, где p_1 — требуемый свойством префикс на предыдущей конфигурации: иначе префикс p_1 не был бы максимальным (максимальным был бы p_2 без последнего символа σ).

Также $|p_2| \neq |p_1| + 1$, так как иначе p_2 получался из p_1 приписыванием σ, что невозможно, так как $\sigma \neq w_{i+1}$. Итак, $p_2 < |p_1| + 1$. Рассмотрим p'_2 — p_2 без последнего символа σ. Тогда $|p'_2| < |p_1|$.

Заметим, что p'_2 — суффикс p_1, так как они оба подстроки входного слова, причем вхождение заканчивается в одной и той же позиции. Также p'_2 — префикс p_1, так как они — префиксы w. Таким образом, p'_2 — суффикс и префикс $p_1 = w_1 \ldots w_i$. Таким образом, необходимо найти в строке $w_1 \ldots w_i$ префикс, который является суффиксом. Ранее требовалось, чтобы после префикса следовал символ σ. Из свойства он должен быть максимальным по длине. Таким образом, необходимо взять префикс длины $f(i, \sigma)$, и переход должен быть в состояние с номером на 1 большее, так как будет обработан и символ σ (этим доказано сохранение Свойства во втором случае).

Пусть автомат находится на позиции, в которой заканчивается вхождение w. Тогда, по свойству, он будет находиться в последнем, принимающем состоянии $\Rightarrow$ слово будет принято.

Пусть входное слово не содержит w. Тогда длина префикса w, который входит во входное слово не достигнет $|w|$ $\Rightarrow$ по свойству автомат не перейдет в принимающее состояние.

Задача 7

Построение префикс-функции

```
Looking at [1], a: lastP=<> Found P=<>
Looking at [2], b: lastP=<> lastP=<> break Found P=<>
Looking at [3], b: lastP=<> lastP=<> break Found P=<>
Looking at [4], a: lastP=<> Found P=<a>
Looking at [5], #: lastP=<a> lastP=<> break Found P=<>
Looking at [6], a: lastP=<> Found P=<a>
Looking at [7], b: lastP=<a> Found P=<ab>
Looking at [8], b: lastP=<ab> Found P=<abb>
Looking at [9], b: lastP=<abb> lastP=<> break Found P=<>
Looking at [10], a: lastP=<> Found P=<a>
Looking at [11], b: lastP=<a> Found P=<ab>
Looking at [12], a: lastP=<ab> lastP=<> break Found P=<a>
Looking at [13], b: lastP=<a> Found P=<ab>
Looking at [14], b: lastP=<ab> Found P=<abb>
Looking at [15], a: lastP=<abb> Found P=<abba>
Looking at [16], b: lastP=<abba> lastP=<a> Found P=<ab>
```

Префикс-функция

Номер	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Символ	a	b	b	a	$\#$	a	b	b	b	a	b	a	b	b	a	b
Функция	0	0	0	1	0	1	2	3	0	1	2	1	2	3	4	2

Результат

```
Found, starting from 12
```

https://bitbucket.org/etoestja/inf/raw/HEAD/mipt/s3/TIPL/2/KMP.c — исходный код. Запускать:

```
$ cc KMP.c -o KMP
$ echo "abba abbbababbab" | ./KMP
```