Формальные языки и трансляции. Подготовка к коллоквиуму (2021)

София Бондарь

Алексей Горбулев

Тимур Харисов

2021

Содержание

Программу коллоквиума смотреть тут.

Часть 1. Автоматы и регулярки

Вопрос 1

Основные определения

Def. Алфавит Σ — непустое конечное множество, элементы которого называются символами.

Σ^* — множество слов, состоящее из всех слов алфавита Σ. Пустое слово ε принадлежит Σ^*.

Def. Формальный язык L — некоторое подмножество Σ^*.

Def. Недетерминированный конечный автомат (НКА) — кортеж $M = \langle Q, \Sigma, \Delta, q_0, F\rangle$, где:

1. Q — множество состояний, Q — конечное множество, то есть $|Q| < \infty$;
2. Σ — алфавит;
3. $\Delta \subset Q \times \Sigma^* \times Q$ — множество переходов;
4. $q_0 \in Q$ — стартовое состояние;
5. $F \subset Q$ — множество завершающих состояний.

Почему множество переходов выглядит так страшно на первый взгляд? Дело в том, что переход используется для того, чтобы перейти из вершины (состояния) q_1 по слову $word$ в вершину (состояние) q_2, q_1 и q_2 — состояния, $word$ — слово. Поэтому формально множество переходов определяется как подмножество соответствующего декартового произведения: состояние — слово — состояние.

Def. Конфигурация в автомате $\langle Q, \Sigma, \Delta, q_0, F\rangle$ — элемент $\langle q, w\rangle \in Q \times \Sigma^*$.

Def. Отношение $\vdash$ достижимости по M — наименьшее рефлексивное транзитивное отношение над $Q \times \Sigma^*$, такое что:

1. $\forall w \in \Sigma^* : (\langle q_1, w\rangle \to q_2) \in \Delta \Longrightarrow \langle q_1, w\rangle \vdash \langle q_2, \varepsilon\rangle$
2. $\forall u, v \in \Sigma^* : \langle q_1, u\rangle \vdash \langle q_2, \varepsilon\rangle,\ \langle q_2, v\rangle \vdash \langle q_3, \varepsilon\rangle \Longrightarrow \langle q_1, uv\rangle \Longrightarrow \langle q_3, \varepsilon\rangle$
3. $\forall u \in \Sigma^* : \langle q_1, u\rangle \vdash \langle q_2, \varepsilon\rangle \Longrightarrow \forall v \in \Sigma^* \langle q_1, uv\rangle \vdash \langle q_2, v\rangle$

Def. Для автомата $M = \langle Q, \Sigma, \Delta, q_0, F\rangle$ языком $L(M)$, задаваемым автоматом M, является множество $\{w \in \Sigma^* | \exists q \in F : \langle q_0, w\rangle \vdash \langle q, \varepsilon\rangle\}$.

Def. Язык L называется автоматным, если существует такой НКА M, что $L = L(M)$.

Утверждение об НКА с одним завершающим состоянием

Утверждение. Для любого автоматного языка L существует НКА $M' = \langle Q', \Sigma, \Delta', q_0', F' \rangle$, такой что $L(M') = L$, и $|F'| = 1$.

Доказательство: L — автоматный язык, значит, существует НКА $M = \langle Q, \Sigma, \Delta, q_0, F \rangle$, такой что $L(M) = L$. Введём $M' = \langle Q \cup \{q_f\}, \Sigma, \Delta', q_0, \{q_f\} \rangle$, где $\Delta' = \Delta \cup \{\langle q, \varepsilon \rangle \to q_f | q \in F\}$. Далее нужно доказать, что $L(M) = L(M')$.

Докажем, что $L(M) \subset L(M')$. По определению из того, что $w \in L(M)$, следует, что существует состояние $q \in F$, что $\langle q_0, w \rangle \vdash \langle q, \varepsilon \rangle$. В автомате M' $\langle q, \varepsilon \rangle \to q_f \Longrightarrow \langle q, \varepsilon \rangle \vdash \langle q_f, \varepsilon \rangle$. Так как $\langle q_0, w \rangle \vdash \langle q, \varepsilon \rangle \vdash \langle q_f, \varepsilon \rangle$, то $w \in L(M')$.

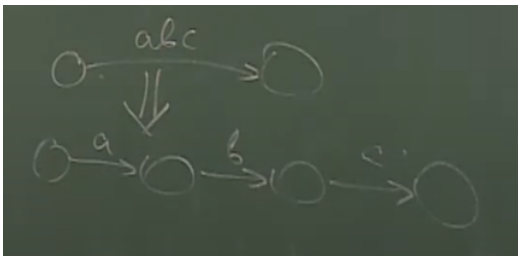
Докажем, что $L(M) \supset L(M')$. Из того, что $w \in L(M')$, следует, что $\langle q_0, w \rangle \vdash \langle q_f, \varepsilon \rangle$. Но так как в q_f можно добраться только по ε-переходу, то существует состояние q', что $\langle q_0, w \rangle \vdash \langle q', \varepsilon \rangle \vdash_1 \langle q_f, \varepsilon \rangle \Longrightarrow q' \in F$. А в автомате M $\langle q_0, w \rangle \vdash \langle q', \varepsilon \rangle$. Значит, $w \in L(M)$. ■

Утверждение об НКА с не более однобуквенными переходами

Утверждение. Для любого автоматного языка L существует НКА $M = \langle Q, \Sigma, \Delta, q_0, F \rangle$, такой что $L = L(M)$ и:

$$\forall (\langle q_1, w \rangle \to q_2) \in \Delta \ |w| \leqslant 1$$

Доказательство: Идею доказательства проще нарисовать:



Расширим множество вершин Q так, чтобы более однобуквенным переходам из одного состояния в другое соответствовало бы несколько не более однобуквенных переходов через цепочку состояний. Формально на лекции этого проделано не было. ■

Теорема об НКА с однобуквенными переходами

Th. Для любого НКА $M = \langle Q, \Sigma, \Delta, q_0, F \rangle$ существует НКА $M' = \langle Q, \Sigma, \Delta', q_0, F' \rangle$, такой что $L(M) = L(M')$ и:

$$\forall (\langle q_1, w \rangle \to q_2) \in \Delta \ |w| = 1$$

Доказательство: Обозначим множество вершин, достижимых из q до q' как $\Delta(q, w) = \{q' | \langle q, w\rangle \vdash \langle q', \varepsilon\rangle\}$. Считаем, что в любом переходе $|w| \leqslant 1$. Введём следующие множества:

$$F' := \{q | \Delta(q, \varepsilon) \cap F \neq \emptyset\}^1$$

$$\Delta' = \{\langle q_1, a\rangle \to q_2 | \exists q_3 \in \Delta(q_1, \varepsilon) : \langle q_3, a\rangle \to q_2\}$$

Докажем следующее утверждение:

$$\exists q \in F : \langle q_0, w\rangle \vdash_M \langle q, \varepsilon\rangle \iff \exists q' \in F' : \langle q_0, w\rangle \vdash_{M'} \langle q', \varepsilon\rangle$$

Пусть $w = w_1 w_2 \dots w_n$, $w \in L(M') \implies \exists q \in F' : \langle q_0, w\rangle \vdash_{M'} \langle q, \varepsilon\rangle$. Из однобуквенности всех переходов:

$$\exists q_1, \dots, q_n : \langle q_0, w_1 \dots w_n\rangle \vdash \langle q_1, w_2 \dots w_n\rangle \vdash \dots \vdash \langle q_{n-1}, w_n\rangle \vdash \langle q_n, \varepsilon\rangle$$

$$q_n \in F' \implies \exists q'' \in F : q'' \in \Delta(q_n, \varepsilon) \implies \langle q_n, \varepsilon\rangle \vdash_M \langle q'', \varepsilon\rangle \ (1)$$

$$\left(\langle q_{k-1}, w_k\rangle \overset{M'}{\to} q_k\right) \implies \exists \widetilde{q_k} \in Q$$

$$\widetilde{q_k} = \Delta(q_{k-1}, \varepsilon)$$

$$\langle \widetilde{q_k}, w_k\rangle \to q_k \in \Delta$$

$$\langle q_{k-1}, \varepsilon\rangle \vdash_M \langle \widetilde{q_k}, \varepsilon\rangle$$

$$\langle q_{k-1}, w_k\rangle \vdash_M \langle q_k, \varepsilon\rangle \ (2)$$

Из (1) и (2) следует, что существует $q'' \in F$, такой что $\langle q_0, w_1 \dots w_n\rangle \overset{(2)}{\vdash_M} \langle q_1, w_2 \dots w_k\rangle \overset{(2)}{\vdash_M} \dots \overset{(2)}{\vdash_M} \langle q_n, \varepsilon\rangle \overset{(1)}{\vdash_M} \langle q'', \varepsilon\rangle$, откуда $w = w_1 \dots w_n \in L(M)$. ■

Вопрос 2

Определение детерминированного конечного автомата

Def. НКА $M = \langle Q, \Sigma, \Delta, q_0, F\rangle$ называется детермированным (ДКА), если выполнено:

1. $\forall (\langle q_1, w\rangle \to q_2) \in \Delta' : |w| = 1$ *(все переходы являются однобуквенными)*;
2. $\forall a \in \Sigma, q \in Q \ |\Delta(q, a)| \leqslant 1$ *(из одного состояния по одному символу можно перейти не более, чем в одно состояние)*;

[1] На лекции Павел Ахтямов вводил F' как $F' := \{q' | \Delta(q, \varepsilon) \cap F \neq \emptyset\}$. Однако если F' определить именно так, то, как заметила София Бондарь, можно получить противоречие с пунктом (1) доказательства. Само доказательство вообще само по себе спорное, например, явным образом не показывается, что $w \in L(M) \implies w \in L(M')$. Более конструктивное доказательство от Алексея Сорокина можно посмотреть, например, здесь.

Эквивалентность ДКА и НКА

Th. Для любого НКА $M = \langle Q, \Sigma, \Delta, q_0, F\rangle$ существует ДКА M', такой что $L(M) = L(M')$.

Доказательство:

Обозначим $\Delta(S, w) = \bigcup\limits_{q \in S} \Delta(q, w)$, где $w \in \Sigma^*$, $S \subset Q$. Построим ДКА $M' = \langle 2^Q, \Sigma, \Delta', \{q_0\}, F'\rangle$, где:

1. $F' = \{S \subset Q | S \cap F \neq \emptyset\}$;
2. $\Delta' = \{\langle S, a\rangle \to \Delta(S, a)\}$.

Чтобы понять, что из себя представляет новое множество переходов Δ', докажем следующую лемму:

Лемма. $\Delta'(\{q_0\}, w) = \Delta(\{q_0\}, w)$

Докажем лемму индукцией по длине слова w.

База. $w = \varepsilon$, тогда $\Delta(\{q_0\}, \varepsilon) = \{q_0\} = \Delta'(\{q_0\}, \varepsilon)$, так как все переходы в автомате M' являются однобуквенными.

Переход. $w = ua$, $a \in \Sigma$, $u \in \Sigma^*$.

Сначала покажем, что $\Delta(\{q_0\}, ua) = \Delta(\Delta(\{q_0\}, u), a)$:

$$\Delta(\{q_0\}, ua) = \{q | \langle q_0, ua\rangle \vdash \langle q, \varepsilon\rangle\}$$

По однобуквенности переходов:

$$\{q | \langle q_0, ua\rangle \vdash \langle q, \varepsilon\rangle\} = \{q | \exists q' : \langle q_0, ua\rangle \vdash \langle q', a\rangle \vdash \langle q, \varepsilon\rangle\} = \{q | \exists q' \in \Delta(q_0, u) : \langle q', a\rangle \vdash \langle q, \varepsilon\rangle\} = \Delta(\Delta(\{q_0\}, u), a)$$

По предположению индукции:

$$\Delta(\Delta(\{q_0\}, u), a) = \Delta(\Delta'(\{q_0\}, u), a)$$

$$S := \Delta'(\{q_0\}, u)$$

$$\Delta(S, a) = \Delta'(S, a) \text{ (следует из определения переходов в ДКА)}$$

$$\Delta'(\Delta'(\{q_0\}, u), a) = \Delta'(\{q_0\}, ua)$$

Лемма доказана.

Теперь покажем, что $w \in L(M) \iff w \in L(M')$.

$$w \in L(M) \iff \exists q \in F : \langle q_0, w\rangle \vdash \langle q, \varepsilon\rangle \iff \Delta(q_0, w) \cap F \neq \emptyset \iff \Delta(\{q_0\}, w) \cap F \neq \emptyset \overset{lemma}{\iff} \Delta'(\{q_0\}, w) \cap F \neq \emptyset$$

$$T := \Delta'(\{q_0\}, w)$$

$$T \cap F \neq \emptyset \Longleftrightarrow T \in F',\ \Delta'\left(q_0', w\right) \in F',\ q_0' = \{q_0\}$$

$$T \in F' \Longleftrightarrow w \in L\left(M'\right)$$

■

О полных ДКА

Def. ДКА $M = \langle Q, \Sigma, \Delta, q_0, F\rangle$ называется полным, если верно, что $\forall q \in Q \forall a \in \Sigma \exists q' \in Q : (\langle q, a\rangle \to q') \in \Delta$

Утверждение. ДКА $M = \langle Q, \Sigma, \Delta, q_0, F\rangle$ эквивалентен полному ДКА $M' = \langle Q', \Sigma, \Delta', q_0, F\rangle$, где $Q' = Q \cup \{q_{fict}\}$, $\Delta' = \Delta \cup \Delta_{new}$, где q_{fict} — фиктивная вершина, $\Delta_{new} = \{(\langle q_i, a\rangle \to q_{fict}) \,|q_i \in Q, a \in \Sigma, \Delta_M\left(q_i, a\right) = \emptyset\} \cup \{(\langle q_{fict}, a\rangle \to q_{fict}) \,|a \in \Sigma\}$.

Доказательство: Следует из построения: добавляем новую вершину q_{fict}, добавляем переходы, которых не хватает для того, чтобы ДКА был полным, проводим циклы из q_{fict} на себя. Множество F не изменилось, новых принимающих путей не появилось, и если $w \notin L\left(M\right)$, то $\Delta_{M'}\left(q_0, w\right) = \{q_{fict}\}$. Если сделать «откат» из M' в M, то множество принимающих путей не изменится. ■

Вопрос 3

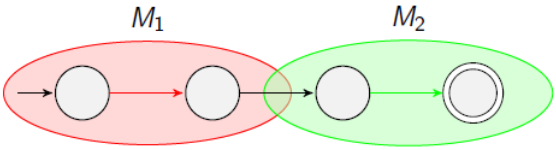
Замкнутость автоматных языков относительно операций

Th. Автоматные языки замкнуты относительно операций:

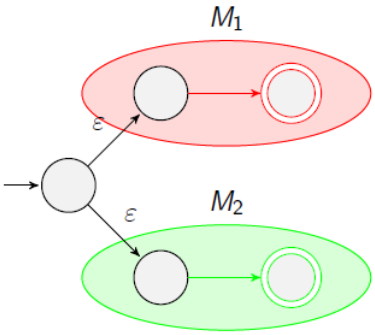
1. Конкатенации
2. Объединения
3. Пересечения
4. Итерации Клини
5. Дополнения

Доказательство: Пусть $L_1\left(M_1\right)$ и $L_2\left(M_2\right)$ — автоматные языки, которым соответствуют НКА M_1 и M_2 соответственно.

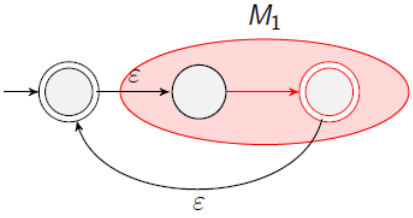
Укажем явно конкатенацию языков L_1 и L_2:



Укажем явно объединению языков L_1 и L_2:



Укажем явно итерацию Клини языка L_1:



Для того, чтобы показать, что автоматные языки замкнуты относительно операции пересечения, рассмотрим эквивалентные M_1 и M_2 ПДКА $M_1' = \langle Q_1', \Sigma, \Delta_1', q_{01}, F_1' \rangle$ и $M_2' = \langle Q_2', \Sigma, \Delta_2', q_{02}, F_2' \rangle$ соответственно. Рассмотрим новый автомат:

$$M = \langle Q_1' \times Q_2', \Sigma, \Delta, (q_{01}, q_{02}), F_1' \times F_2' \rangle$$

$$\Delta = \{ \langle (q_1, q_2), a \rangle \to (q_1', q_2') \,|\, \langle q_1, a \rangle \to q_1' \in \Delta_1', \langle q_2, a \rangle \to q_2' \in \Delta_2' \}$$

Покажем корректность данной конструкции с помощью леммы:

Лемма. $\forall w \in \Sigma^* \Delta_M ((q_1, q_2), w) = \left(\Delta_{M_1'}(q_1, w), \Delta_{M_2'}(q_2, w) \right)$

Докажем индукцией по длине слова w. База следует из построения. Переход состоит в следующем: пусть $w = w'a$, $|w| = |w'| + 1$, по предположению индукции можно перейти по всему слову w', далее переходим по a.

Из леммы следует, что $w \in L_1 \cap L_2 \iff \Delta_{M_1'}(q_{01}, w) \in F_1'$, $\Delta_{M_2'}(q_{02}, w) \in F_2'$, по лемме это равносильно тому, что $\Delta_M ((q_{01}, q_{02}), w) \in F_1' \times F_2'$, и по определению $w \in L(M)$.

Теперь покажем, что автоматные языки замкнуты относительно дополнения. Считаем, что L задан ПДКА $M = \langle Q, \Sigma, \Delta, q_0, F\rangle$. Тогда $\overline{M} = \langle Q, \Sigma, \Delta, q_0, Q \setminus F\rangle$, $L\left(\overline{M}\right) = \Sigma^* \setminus L$. В силу того, что автомат является ПДКА, по каждому слову можно прийти либо в завершающее состояние, либо в состояние, которое не является завершающим. Чтобы автомат принимал слова, принадлежащие дополнению, достаточно состояния, которые ранее не были завершающими, сделать завершающими, а с завершающими состояниями поступить аналогично, но в обратную сторону. ■

Вопрос 4

Регулярные выражения

Def. Множество регулярных выражений $Reg\,(\Sigma)$ — наименьшее множество, такое что:

1. $\Sigma \subseteq Reg\,(\Sigma)$;
2. $0,\ 1 \in Reg\,(\Sigma)$;
3. Если $\alpha,\ \beta \in Reg\,(\Sigma)$, то $(\alpha \cdot \beta) \in Reg\,(\Sigma)$ и $(\alpha + \beta) \in Reg\,(\Sigma)$;
4. Если $\alpha \in Reg\,(\Sigma)$, то $(\alpha^*) \in Reg\,(\Sigma)$.

Приоритет операций в регулярных выражениях (левее — приоритетнее): $* \to \cdot \to +$ (сначала итерация Клини, затем конкатенация, затем сложение).

Регулярные языки

Def. $L\,(R)$ — регулярный язык, задаваемый регулярным выражением R. Определение регулярного языка рекурсивное, как и регулярного выражения:

$Regex(R)$	$Language(L_i = L(R_i))$
0	$\emptyset$
1	$\{\varepsilon\}$
$a,\ a \in \Sigma$	$\{a\}$
$R_1 + R_2$	$L_1 \cup L_2$
$R_1 \cdot R_2$	$L_1 \cdot L_2$
R^*	L^*

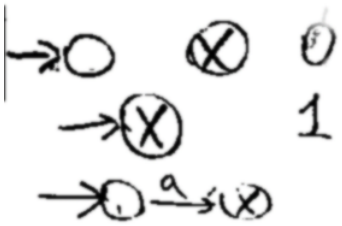
Теорема Клини о совпадении классов автоматных и регулярных языков

Th. Классы автоматных и регулярных языков совпадают.

Доказательство: Докажем два вложения:

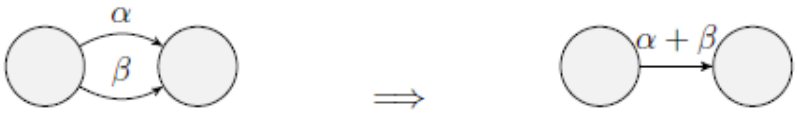
Регулярные $\subseteq$ автоматные: докажем индукцией по построению регулярного выражения.

База. Построим автоматы. соответствующим языкам $\emptyset$, $\{\varepsilon\}$, $\{a\}$:

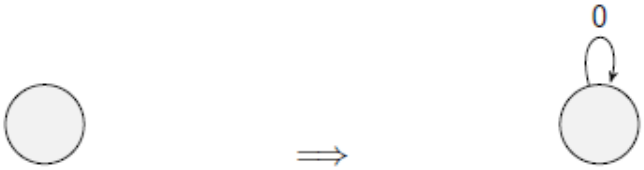


Переход. Следует из теоремы о замкнутости класса автоматных языков относительно операций.

Автоматные $\subseteq$ регулярные: рассмотрим регулярный автомат $\langle Q, \Sigma, \Delta, q_0, F\rangle$, где $\Delta \subset Q \times R(\Sigma) \times Q$, $R(\Sigma)$ — множество регулярных выражений над Σ. Кроме того, применим утверждение об НКА с одним завершающим состоянием (доказательство есть в вопросе 1). Удалим кратные рёбра:



Добавим циклы из состояние в то же состояние, если таковых нет:



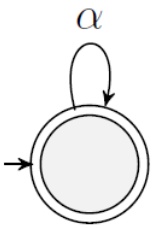
Это требуется для следующего:

$$0^* = 1$$

$$0^* = \{\varepsilon\} \cup L(0) \cup (L(0))^2 \cup \cdots = \{\varepsilon\} = 1$$

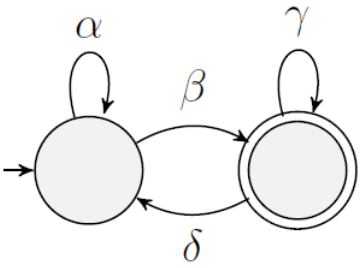
Теперь преобразуем регулярный автомат в регулярное выражение индукцией по числу состояний регулярного автомата.

База. $|Q| = 1$. Тогда в регулярном автомате стартовое состояние является завершающим, и можно однозначно построить регулярное выражение. Например:



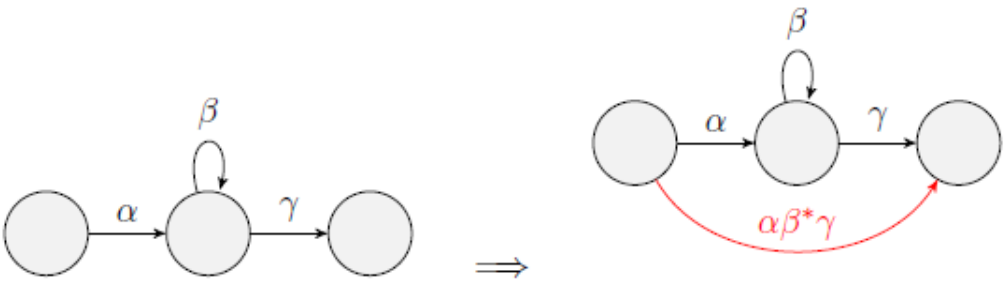
Данному автомату соответствует регулярное выражение α^*.

$|Q| = 2$. Если стартовое состояние является завершающим, то можно свести к случаю $|Q| = 1$. Иначе стартовое состояние и завершающее состояние различны, и можно тоже однозначно построить регулярное выражение. Например:



Данному автомату соответствует регулярное выражение $\alpha^*\beta\,(\gamma + \delta\alpha^*\beta)^*$.

Переход. $|Q| \geqslant 3$. Тогда в автомате существуют состояния, которые не являются ни завершающими, ни стартовыми. Тогда из автомата можно убрать состояние, не являющееся ни стартовым, ни завершающимся, следующим образом:



Повторим процесс для всех путей длины 2, пока не останется ни одного состояния, не являющегося стартовым или завершающимся. Тем самым число состояний будет уменьшено либо до 1, либо до 2, а эти случаи уже рассмотрены. ■

Вопрос 5

Вопросы 5, 6 и 8 довольно тесно связаны между собой. Если вдруг чего-то не хватает в одном из вопросов, то можно посмотреть в каком-либо из двух других вопросов.

Основные определения

Def. Детерминированный конечный автомат M называется полным (ПДКА), если для M выполнено следующее условие: для любого символа a из алфавита Σ и любого состояния q из множества состояний Q верно, что $|\Delta(q, a)| = 1$.

Пусть $L \subset \Sigma^*$ — автоматный язык $L(M)$, M — ПДКА для L.

Отношение $\sim_L$

Def. Отношение $\sim_L$ — такое отношение на Σ^*, что $u \sim_L v \iff \forall w \in \Sigma^* (uw \in L \iff vw \in L)$.

Утверждение. $\sim_L$ является отношением эквивалентности.

Доказательство:

Рефлексивность: для любого u верно, что $u \sim_L u \iff \forall w \in \Sigma^* (uw \in L \iff uw \in L)$.

Симметричность:

$$u \sim_L v \iff \forall w \in \Sigma^* (uw \in L \iff vw \in L) \ (1)$$
$$v \sim_L u \iff \forall w \in \Sigma^* (vw \in L \iff uw \in L) \ (2)$$

Если выполнено (1), то выполнено и (2), так как если выполнено (1), то $uw \in L \iff vw \in L$ для любого слова $w \in \Sigma^*$, и $\forall w \in \Sigma^* (uw \in L \iff vw \in L) \iff \forall w \in \Sigma^* (vw \in L \iff uw \in L)$, откуда (1) $\iff$ (2).

Транзитивность:

$$u \sim_L v \iff \forall w \in \Sigma^* (uw \in L \iff vw \in L) \ (1)$$
$$v \sim_L x \iff \forall w \in \Sigma^* (vw \in L \iff xw \in L) \ (2)$$

Если выполнено (1) и (2), то $uw \in L \iff vw \in L \iff xw \in L$ для любого слова $w \in \Sigma^*$, и по определению отношения $\sim_L$ это эквивалентно тому, что $u \sim_L x$. Если хотя бы одно из (1) и (2) не выполнено, то и отношение $u \sim_L x$ не выполнится.

■

Классы эквивалентности

Def. Множество классов эквивалентности $\Sigma^*/_{\sim_L}$ — это множество $\{\{u | u \sim_L v\} | v \in \Sigma^*\}$.

Def. Отношение $\sim_M$ — такое отношение $\sim_M$ над Q, что $q_1 \sim q_2 \iff \forall w \in \Sigma^* (\Delta(q_1, w) \subset F \iff \Delta(q_2, w) \subset F)$.

Если $q_1 \sim_M q_2$, то их можно попробовать объединить.

Лемма. Пусть $L_q := \{w|\Delta(q_0, w) = q\}$. Тогда каждый класс эквивалентности из $\Sigma^*/_{\sim_L}$ — объединение классов в L_q.

Доказательство:

Рассмотрим слово $u \in \Sigma^*$. $u \in [u] \in \Sigma^*/_{\sim_L}$, где $[u]$ — класс эквивалентности для u. Обозначим $q_u = \Delta(q_0, u)$. Для любого слова $w \in [u]$ $q_w = \Delta(q_0, w)$. Тогда $[u] = \bigcup\limits_{q_w, w \in [u]} L_{q_w}$. Далее докажем, почему это так.

Пусть $v \in [u]$. Тогда $v \sim_L u$, $q_v = \Delta(q_0, v)$, по определению $v \in L_{q_v}$. Тогда $v \in \bigcup\limits_{q_w, w \in [u]} L_{q_w}$.

Пусть $v \in \bigcup\limits_{q_w, w \in [u]} L_{q_w}$. Тогда существует состояние q_z, $z \in [u]$, что $v \in L_{q_z} = \{w|\Delta(q_0, w) = q_z\}$.

$$z \in [u] \Longrightarrow z \sim_L u \Longrightarrow \forall w\,(zw \in L \Longleftrightarrow uw \in L)$$

$$v \in L_{q_z} \Longrightarrow \Delta(q_0, v) = q_z \ (1)$$

$$\Delta(q_0, z) = q_z \ (2)$$

$$(1),\ (2) \Longrightarrow v \sim_L z \ (3)$$

(3) верно, так для любого слова $w \in \Sigma^*$ $\Delta(q_0, vw) = \Delta(q_0, zw)$:

$$\Delta(q_0, vw) = \Delta(\Delta(q_0, v), w) = \Delta(q_z, w)$$

$$\Delta(q_0, zw) = \Delta(\Delta(q_0, z), w) = \Delta(q_z, w)$$

Так как $v \sim_L z$, $z \in [u]$, то $v \in [u]$. Значит, $[u] = \bigcup\limits_{q_w, w \in [u]} L_{q_w}$, и каждый класс эквивалентности из $\Sigma^*/_{\sim_L}$ — объединение классов в L_q.

■

Следствие. $|\Sigma^*/_{\sim_L}| \leqslant |Q|$.

Теорема о существовании и единственности минимального ПДКА

Th. Для любого автоматного языка L существует единственный с точностью до изоморфизма минимальный ПДКА M, такой что $L = L(M)$.

Доказательство:

Пусть M — минимальный ПДКА, Q_M — множество его состояний.

Построим автомат $M_0 = \langle \Sigma^*/_{\sim_L}, \Sigma, \Delta, [\varepsilon], \{[w]|w \in L\}\rangle$. Для любых $u \in \Sigma^*$, $a \in \Sigma$ верно, что $\Delta([u], a) = [ua]$ (факт 1). Так как количество состоянии автомата, соответствующему языку L, конечно, то по следствию из леммы о классах эквивалентности и L_q $|\Sigma^*/_{\sim_L}| < \infty$.

Факт 1 верен вследствие следующего:

$$u \sim_L v \Longrightarrow ua \sim_L va \Longleftrightarrow \forall w \left(uaw \in L \Longleftrightarrow vaw \in L\right)$$

$$w' = aw,\ \forall w' \left(uw' \in L \Longleftrightarrow vw' \in L\right) \Longleftrightarrow u \sim_L v$$

Так как $u \sim_L v$, то если $u \in L$, то $v \in L$, и наоборот.

Теперь рассмотрим $\psi : Q_M \to \Sigma^*/_{\sim_L}$, $\psi\left(q\right) = \{w|\Delta\left(q_0, w\right) = q\} = L_q$. Покажем, что ψ — изоморфизм. Для этого нужно показать, что:

1. $\Delta\left(\psi(q), a\right) = [\psi\left(q\right) a]$;
2. $q \in F \Longleftrightarrow \psi\left(q\right) \subseteq L$.

Покажем, почему выполняется (1).

$$\psi\left(q\right) = [u],\ \psi\left(q'\right) = [u'],\ \Delta\left(q, a\right) = q'$$

$$\Delta\left(\psi[q], a\right) = \Delta\left(\{w|\Delta(q_0, w) = q\}, a\right) = \{w' = wa|\langle q_0, w'\rangle = q'\} = [wa]$$

$$[u] = [w],\ [u'] = [wa] \text{ по транзитивности переходов в автомате}$$

Покажем, почему выполняется (2). Из того, что $q \in F$, следует, что слова из множества $\psi\left(q\right) = \{w|\Delta\left(q_0, w\right) = q\}$ принадлежат языку L, так они распознаются автоматом, поскольку q является завершающим состоянием. А так как $\psi\left(q\right) = \{w|\Delta\left(q_0, w\right) = q\} \subseteq L$, то так как они распознаются автоматом, соответствующему языку L, то $q \in F$.

Пусть ψ_1 — изоморфизм между минимальными ПДКА M_1 и M_0, ψ_2 — изоморфизм между минимальными ПДКА M_2 и M_0. Тогда M_1 и M_2 изоморфны между собой — этому соответствует изоморфизм $\psi_2^{-1} \circ \psi_1$, композиция изоморфизмов является изоморфизмом.

■

Вопрос 6

Повторение определений из вопроса 5

Def. Отношение $\sim_L$ — такое отношение на Σ^*, что $u \sim_L v \Longleftrightarrow \forall w \in \Sigma^* \left(uw \in L \Longleftrightarrow vw \in L\right)$.

Def. Множество классов эквивалентности $\Sigma^*/_{\sim_L}$ — это множество $\{\{u|u \sim_L v\}|v \in \Sigma^*\}$.

Def. Отношение $\sim_M$ — такое отношение $\sim_M$ над Q, что $q_1 \sim q_2 \Longleftrightarrow \forall w \in \Sigma^* \left(\Delta(q_1, w) \in F \Longleftrightarrow \Delta(q_2, w) \in F\right)$.

Лемма. Пусть $L_q := \{w|\Delta\left(q_0, w\right) = q\}$. Тогда каждый класс эквивалентности из $\Sigma^*/_{\sim_L}$ — объединение классов в L_q.

Следствие. $|\Sigma^*/_{\sim_L}| \leqslant |Q|$.

Утверждения, связанные с ними, были доказаны ранее в вопросе 5. На коллоквиуме те же доказательства нужно повторить.

Лемма о ПДКА

Лемма. Для любого автоматного языка L существует ПДКА M', такой что все состояния в M' попарно неэквивалентны.

Доказательство: Рассмотрим автомат над классами эквивалентности $\sim_M$. Класс эквивалентности q обозначим за $[q]$. $M' = \langle Q/_{\sim_M}, \Sigma, \Delta', q_0, F' \rangle$, где:

$$\Delta' = \{\langle [q_1], a\rangle \to [q_2] | \exists \langle q_1, a\rangle \to q_2 \in \Delta\}$$
$$F' = \{[q_f] | q_f \in F\}$$

Проверим, что множества Δ', F' заданы корректно.

Для Δ': Пусть $q_1 \sim_m q_1'$, и существует a такое, что $\Delta(q_1, a) \nsim_m \Delta(q_1', a)$.

$$q_1 \sim_M q_1' \Longrightarrow \forall w \in \Sigma^* \Delta(q, w) \in F \Longleftrightarrow \Delta(q_1', w) \in F$$
$$w = au \Longrightarrow \forall u \in \Sigma^* \Delta(q_1, au) \in F \Longleftrightarrow \Delta(q_1', au) \in F$$

Далее обозначим $\Delta(q_1, a)$ за q_2, $\Delta(q_1', a)$ обозначим за q_2'.

$$\Delta(q_1, au) = \Delta(\Delta(q, a), u) = \Delta(q_2, u)$$
$$\Delta(q_1', au) = \Delta(q_2', u)$$
$$\Delta(q_2, u) \in F \Longleftrightarrow \Delta(q_2', u) \in F.$$

Приходим к противоречию.

Для F':

$$q_1 \in F,\ q_2 \sim_M q_1 \overset{w=\varepsilon}{\Longrightarrow} \Delta(q_1, \varepsilon) \in F \Longleftrightarrow \Delta(q_2, \varepsilon) \in F$$
$$q_1 \in F \Longleftrightarrow q_2 \in F$$

Теперь покажем, что $L(M) = L(M')$. Для этого нужно показать, что $w \in L(M) \Longleftrightarrow \Delta(q_0, w) \in F \Longleftrightarrow \Delta([q_0], w) \in F'$.

Докажем утверждение: $\forall u : \Delta(q_0, u) = q_1 \Longleftrightarrow \Delta([q_0], u) = [q_1]$.

Индукция по длине слова u.

База. $|u| = 0 \Longrightarrow u = \varepsilon$. Тогда $\Delta(q_0, \varepsilon) = q_0$, $\Delta([q_0], \varepsilon) = [q_0]$.

Переход. Пусть $u = va$, $v \in \Sigma^*$, $a \in \Sigma$.

$$\Delta(q_0, va) = q_1 \Longrightarrow \exists q_2 \Delta(q_0, v) = q_2, \Delta(q_2, a) = q_1$$

По предположению индукции, $\Delta([q_0], u) = [q_2]$, $\Delta([q_2], a) = [q_1]$, так как переход $\langle q_2, a\rangle \to q_1 \in \Delta$ тогда и только тогда, когда $\langle [q_2], a\rangle \to [q_1] \in \Delta'$. По транзитивности, $\Delta([q_0], ua) = [q_1]$.

Теперь покажем, что состояния попарно неэквивалентны. Пусть $[q_1] \sim_{M'} [q_2]$. Тогда $\forall w$: $\Delta_{M'}([q_1], w) \in F' \Longleftrightarrow \Delta_{M'}([q_2], w) \in F'$ по определению.

$$[q_{1f}] = \Delta_M([q_1], w) \in F$$

$$[q_{2f}] = \Delta_{M'}([q_2], w)$$

$$\exists q_{1f} \in F : \Delta_m(q_1, w) = q_{1f} \in F \Longleftrightarrow \exists q_{2f} \in F \Delta(q_2, w) = q_{2f} \in F$$

$$[q_1] \sim_{M'} [q_2] \Longrightarrow [q_1] = [q_2]$$

■

Теорема о минимальном ПДКА

Th. M — минимальный ПДКА, распознающий язык L, тогда и только тогда, когда любые два состояния попарно неэквивалентны и все состояния достижимы из стартового.

Теперь запишем более формально:

$$M \text{ — минимальный ПДКА} \Longleftrightarrow \begin{cases} \forall q_1, q_2 \in Q \ q_1 \nsim q_2 \\ \forall q \in Q \ \exists w \in \Sigma^* : \ \langle q_0, w\rangle \vdash \langle q, \varepsilon\rangle \end{cases}$$

Доказательство:

$\Longrightarrow$ Если $q_1 \sim_M q_2$, то $[q_1] = [q_2]$, и их можно объединить в одно состояние, значит, M не был бы минимальным, и тогда из минимальности следует, что $q_1 \nsim_M q_2$. Если среди состояний есть недостижимые, то если их удалить, то множество принимаемых слов не изменится.

$\Longleftarrow$ По следствию из леммы о L_q $|\Sigma^*/_{\sim_L}| \leqslant |Q||$. Рассмотрим w_1, w_2 такие, что $\Delta(q_0, w_1) \neq \Delta(q_0, w_2)$. Введём обозначения:

$$\Delta(q_0, w_1) = q_1$$

$$\Delta(q_0, w_2) = q_2$$

Неэквивалентность состояний q_1, q_2 означает, что существует слово w, что без ограничения общности:

$$\Delta(q_1, w) = \Delta(q_0, w_1 w) \in F \Longleftrightarrow w_1 w \in L$$

$$\Delta(q_2, w) = \Delta(q_0, w_2 w) \notin F \Longleftrightarrow w_2 w \notin L$$

$$w_1 \nsim_L w_2$$

Тогда для автомата M со множеством состояний Q' выполняется, что $|\Sigma^*/_{\sim_L}| \geqslant |Q'|$, но тогда $|Q| \geqslant |\Sigma^*/_{\sim_L}| \geqslant |Q'|$, и M — минимальный. ■

Алгоритм построения минимального ПДКА

Введём отношение эквивалентности по словам длины не более, чем n $\sim_n$.

Def. $q_1 \sim_n q_2$, если для любого слова $w : |w| \leqslant n$ выполняется, что:

$$\Delta(q_1, w) \in F \iff \Delta(q_2, w) \in F$$

Введём $Q/_{\sim_n}$.

Лемма. $q_1 \sim q_2 \Longrightarrow q_1 \sim_{|Q|-2} q_2$

Доказательство: Если $q_1 \sim_{i+1} q_2 \Longrightarrow q_1 \sim_i q_2$, тогда $|Q/_{\sim_i}| \leqslant |Q/_{\sim_{i+1}}|$.

Покажем, что если $|Q/_{\sim_i}| = |Q/_{\sim_{i+1}}|$, то $|Q/_{\sim_{i+1}}| = |Q/_{\sim_{i+2}}|$. Пусть существуют состояния q_1 и q_2. такие что $q_1 \nsim_{i+2} q_2$, $q_1 \sim_{i+1} q_2$.

$$q_1 \nsim_{i+2} q_2 \Longrightarrow \exists u, |u| \leqslant i+2, \text{ что без ограничения общности } \Delta(q_1, u) \in F,\ \Delta(q_2, u) \notin F$$

$$u := aw$$

$$\Delta(\Delta(q_1, a), w) \in F,\ \Delta(\Delta(q_2, a), w) \notin F$$

$$|w| \leqslant i+1 \Longrightarrow \Delta(q_1, a) \nsim_{i+1} \Delta(q_2, a) \Longrightarrow \Delta(q_1, a) \nsim_i \Delta(q_2, a) \Longrightarrow q_1 \nsim_{i+1} q_2$$

Приходим к противоречию. ■

Отсюда следует, что если $|Q/_{\sim_i}| = |Q/_{\sim_{i+1}}|$, то $|Q/_{\sim_{i+1}}| = |Q/_{\sim_{i+2}}|$ и так далее.

Множество $Q/_{\sim_0}$ представляет собой $\{F, Q \setminus F\}$, то есть это множество из множества завершающих состояний и множества состояний, не являющихся завершающими.

Класс расширяется, когда найдутся два состояния $q_1 \sim_n q_2$, что $q_1 \nsim_{n+1} q_2$. Тогда до тех пор, пока не найдётся m, что $Q/_{\sim_m} = Q/_{\sim_{m+1}}$, если уже нашли $Q/_{\sim_i}$, то найдём $Q/_{\sim_{i+1}}$ через различия в слове длины $i+1$, что можно посмотреть по переходу по первой букве: в состояния каких классов идёт переход. Далее перенумеруем классы эквивалентности в соответствии с тем, в состояния каких классов $Q/_{\sim_i}$ был осуществлён переход.

Вопрос 7

Лемма о разрастании для автоматных языков

Лемма. Пусть L — автоматный язык, $|L| = \infty$. Тогда существует такое P, что для любого слова $w \in L$ такого, что $|w| \geqslant P$, существуют такие x, y, z, что $w = xyz$, $|xy| \leqslant P$, $|y| \neq 0$, что для любого $k \in \mathbb{N}$ выполняется, что $xy^k z \in L$.

Кванторная версия:

$$\exists P\ \forall w \in L : |w| \geqslant P\ \exists x, y, z : w = xyz,\ |xy| \leqslant P,\ |y| \neq 0 : \forall k \in \mathbb{N} : xy^k z \in L$$

Доказательство: Рассмотрим НКА с однобуквенными переходами M. Положим P равным $|Q|$ — количеству состояний в данном автомате. Тогда если $|w|$ — длина слова w — не меньше, чем P, то с повторениями будет посещено хотя бы $|w| + 1 > |w| \geqslant P = |Q|$ состояний. По принципу Дирихле существует состояние q, посещённое хотя бы дважды. Рассмотрим начальное состояние q_0, состояние q, конечное состояние $q_f \in F$. x будет соответствовать префиксу w, соответствующему пути из q_0 в q в автомате, y будет соответствовать пути из q в q в автомате («цикл»), z будет соответстовать суффиксу w, соответствующему пути из q в q_f в автомате. Теперь покажем следующее:

1. $|xy| \leqslant P = |Q|$: допустим, что $|xy| > P = |Q|$, тогда в xy нашёлся бы цикл, где q не будет являться первым пересечением;
2. $|y| \neq 0$, так как среди переходов в автомате M встречаются только однобуквенные, то существует хотя бы одно состояние $q_{new} \neq q$, которое будет посещено.

Слово $xy^k z$ принадлежит языку L для любого $k \in \mathbb{N}$, так как цикл, соответствующий y, может быть повторён k раз. ■

Примеры неавтоматных языков

Пример: Язык $\{a^n b^n c^n | n \in \mathbb{N}\}$ не является автоматным.

Доказательство: Пусть язык $L = \{a^n b^n c^n | n \in \mathbb{N}\}$ является автоматным. Тогда для него выполняется лемма о разрастании: $\exists P\ \forall w \in L : |w| \geqslant P\ \exists x,\ y,\ z : w = xyz, |xy| \leqslant P,\ |y| \neq 0 : \forall i \in \mathbb{N} : xy^i z \in L$.

Для любого $P \in \mathbb{N}$ рассмотрим слово $w = a^P b^P c^P,\ |w| = 3P$. Тогда если $|xy| \leqslant P$, $|y| \neq 0$, то xy состоит из символов a и только из них, как и y, $xy \neq \varepsilon$, $y \neq \varepsilon$. Укажем явно x, y и z: $x = a^{P-m}$, $y = a^m$, $z = b^P c^P$, $1 \leqslant m \leqslant P$. Для любого i $|xy^i z|_b = |xyz|_b = P$, $|xy^i z|_c = |xyz|_c = P$, при этом $|xy^i z|_a - |xyz|_a \geqslant 1$, и $|xy^i z|_a \geqslant P + 1 \neq P$, и существует $i \in \mathbb{N}$ такое, что $xy^i z \notin L$. Тогда лемма о разрастании не выполняется, L не является автоматным. ■

Вопрос 8

Повторение определений из вопроса 5

Def. Отношение $\sim_L$ — такое отношение на Σ^*, что $u \sim_L v \iff \forall w \in \Sigma^* (uw \in L \iff vw \in L)$.

Def. Множество классов эквивалентности $\Sigma^*/_{\sim_L}$ — это множество $\{\{u | u \sim_L v\} | v \in \Sigma^*\}$.

Def. Отношение $\sim_M$ — такое отношение $\sim_M$ над Q, что $q_1 \sim q_2 \iff \forall w \in \Sigma^* (\Delta(q_1, w) \in F \iff \Delta(q_2, w) \in F)$.

Лемма. Пусть $L_q := \{w | \Delta(q_0, w) = q\}$. Тогда каждый класс эквивалентности из $\Sigma^*/_{\sim_L}$ — объединение классов в L_q.

Следствие. $|\Sigma^*/_{\sim_L}| \leqslant |Q|$.

Утверждения, связанные с ними, были доказаны ранее в вопросе 5. На коллоквиуме те же доказательства нужно повторить.

Вообще намного проще доказать следующее утверждение.

Утверждение. Пусть $L = L(M)$, $M = \langle Q, \Sigma, \Delta, q_0, F\rangle$ — ПДКА. Тогда:

$$\Delta(q_0, u) = \Delta(q_0, v) \Longrightarrow u \sim_L v$$

Доказательство: Предположим, что существует слово $w \in \Sigma^*$, что $uw \in L$, $vw \notin L$. Тогда $uw \in L \iff \Delta(q_0, uw) \in F \iff \Delta(\Delta(q_0, u), w) \in F \iff \Delta(\Delta(q_0, v), w) \in F \iff \Delta(q_0, vw) \in F \iff vw \in L$. Противоречие. ■

Отсюда тоже следует, что $|\Sigma^*/_{\sim_L}| \leqslant |Q|$.

Теорема Майхилла-Нероуда

Th. Язык L является автоматным тогда и только тогда, когда $\Sigma^*/_{\sim_L}$ содержит конечное количество классов эквивалентности.

Доказательство:

$\Longrightarrow$ Так как L является автоматным. то для него существует минимальный ПДКА, $|\Sigma^*/_{\sim_L}| \leqslant |Q|$, а множество состояний в автомате конечно по определению.

$\Longleftarrow$ Множество $\Sigma^*/_{\sim_L}$ конечно. Построим канонический ПДКА $M_0 = \langle \Sigma^*/_{\sim_L}, \Sigma, \Delta, [\varepsilon], \{[w] | w \in L\}\rangle$. Для любых $u \in \Sigma^*$, $a \in \Sigma$ верно, что $\Delta([u], a) = [ua]$ (факт 1). Так как количество состоянии автомата, соответствующему языку L, конечно, то по следствию из леммы о классах эквивалентности и L_q $|\Sigma^*/_{\sim_L}| < \infty$.

Факт 1 верен вследствие следующего:

$$u \sim_L v \Longrightarrow ua \sim_L va \iff \forall w (uaw \in L \iff vaw \in L)$$

$$w' = aw, \forall w' (uw' \in L \iff vw' \in L) \iff u \sim_L v$$

Так как $u \sim_L v$, то если $u \in L$, то $v \in L$, и наоборот. Автомат построен. ■

Алгоритм проверки регулярных выражений на равенство

Пусть R_1 и R_2 — регулярные выражения.

Способ 1. Построим по ним минимальные ПДКА M_1 и M_2 соответственно. По теореме о существовании и единственности минимального ПДКА для автоматного языка $L(M)$ минимальный ПДКА единственен с точности до изоморфизма. Если M_1 и M_2 изоморфны, то регулярные выражения равны, иначе нет.

Способ 2. Построим по R_1 и R_2 ПДКА M_1 и M_2 соответственно: сначала построим регулярные автоматы, затем детерминируем их, если будет нужно, и затем приведём их к ПДКА. Проверим, что $L(M_1) = L(M_2)$. Важное условие, следующее из теоретико-множественных рассуждений:

$$L(M_1) \subset L(M_2) \iff L(M_1) \cap \overline{L(M_2)} = \emptyset$$

Из него следует, что нужно проверить, что:

1. $L(M_1) \cap \overline{L(M_2)} = \emptyset$
2. $L(M_2) \cap \overline{L(M_1)} = \emptyset$

Если оба условия выполнены, то $L(M_1) = L(M_2)$, и $R_1 = R_2$.

Часть 2. КС-грамматики и МП-автоматы

Вопрос 9

Иерархия Хомского

Иерархия Хомского позволяет классифицировать классы грамматик по типу правил.

Тип	Грамматики	Правила	Автоматы
3	Праволинейные	$A \to wB$, $A \to w$	НКА
2	Контестно-свободные	$a \to \alpha$	Автоматы с магазинной памятью
1	Контекстно-зависимые	$\varphi A \psi \to \varphi \alpha \psi$	Линейно-ограниченные недетерминированные автоматы, машины Тьюринга
0	Порождающие	любые	Машины Тьюринга

Вводные определения

Def. Порождающей грамматикой G называется кортеж $\langle N, \Sigma, P, S\rangle$, где:

1. N — множество вспомогательных (нетерминальных) символов, N — конечное множество, то есть $|N| < \infty$;
2. Σ — алфавит — множество терминальных символов, $|\Sigma| < \infty$, N и Σ не имеют общих элементов, то есть $N \cap \Sigma = \emptyset$;
3. P — множество правил, $P \subset (N \cup \Sigma)^+ \times (N \cup \Sigma)^*$, $|P| < \infty$ *(чтобы понять, почему берётся подмножество именно $(N \cup \Sigma)^+ \times (N \cup \Sigma)^*$, посмотрим, какой вид имеет правило $p \in P$: $\alpha \to \beta$, где α должен содержать хотя бы один, как правило, нетерминальный символ, в зависимости от типа грамматики в левой части правила могут быть как терминальные, так и нетерминальные символы, а β может быть равным ε, то есть пустому слову)*;
4. $S \in N$ — стартовый нетерминальный символ.

Def. Отношением достижимости в грамматике $\vdash_G$ называется наименьшее рефлексивное транзитивное отношение, такое что для любого правила $(\alpha \to \beta) \in P$ и для любых элементов $\varphi, \psi \in (N \cup \Sigma)^*$ выполняется, что $\varphi\alpha\psi \vdash_G \varphi\beta\psi$. *По сути, это операция замены левой части на правую часть несколько раз, возможно, нуль.*

Def. Говорят, что слово w выводимо в грамматике $G = \langle N, \Sigma, P, S\rangle$, если $S \vdash_G w$, то есть из стартового символа достижимо слово w.

Def. Говорят, что язык L распознаётся грамматикой G, если $L = \{w \in \Sigma^* | S \vdash_G w\}$, то есть язык L состоит из таких слов, которые выводимы в грамматике G. Если L распознаётся грамматикой G, то его обозначают как $L(G)$.

Праволинейные грамматики и праволинейные языки

Def. Грамматика G называется праволинейной, если правила из P имеют вид либо $A \to wB$, либо $A \to w$, где A, B — нетерминальные символы, то есть A, $B \in N$, и $w \in \Sigma^*$.

Def. Язык $L(G)$ называется праволинейным, если грамматика G, которой распознаётся (задаётся) язык L, является праволинейной.

Теорема о совпадении классов автоматных и праволинейных языков

Th. Множество автоматных языков равно множеству языков, задаваемых праволинейными грамматиками, *то есть для любого языка L существует НКА M, такой что $L = L(M) \iff$ существует праволинейная грамматика G, такая что $L = L(G)$.*

Доказательство от Павла Ахтямова не особо оказалось понятным, поэтому ниже будет доказательство, подобное тому, которое делал Алексей Сорокин.

Доказательство:

$\Longrightarrow$ Докажем, что если язык является автоматным ($L = L(M)$), то он задаётся праволинейной грамматикой. Автомат M считаем детерминированным. $M = \langle Q, \Sigma, \Delta, q_o, F\rangle$. Построим праволинейную грамматику $G_M = \langle N, \Sigma, P, S\rangle$, где $N = Q$, $S = q_0$, а $P = \{q_1 \to aq_2 | \langle q_1, a\rangle \to q_2 \in \Delta\} \cup \{f \to \varepsilon | f \in F\}$. Далее докажем следующую лемму:

Лемма. $\langle q_1, w\rangle \vdash_M \langle q_2, \varepsilon\rangle \iff q_1 \vdash_{G_M} wq_2$

База: $w = \varepsilon$.

$\langle q_1, \varepsilon\rangle \vdash_M \langle q_2, \varepsilon\rangle \iff q_1 = q_2 \iff q_1 \vdash_{G_M} q_2 \iff q_1 \vdash \varepsilon q_2$

Шаг индукции: $w = au$.

Так как $|u| < |w|$, то применим предположение индукции.

$$\langle q_1, a\rangle \vdash_M \langle q_3, \varepsilon\rangle$$

$$\langle q_1, au\rangle \vdash_M \langle q_3, u\rangle$$

$$\langle q_3, u\rangle \vdash_M \langle q_2, \varepsilon\rangle$$

Далее по предположению индукции и построению:

$$q_3 \vdash_{G_M} uq_2$$

$$(q_1 \to aq_3) \in P \iff q_1 \vdash_1 aq_3 \vdash auq_2$$

$$auq_2 = wq_2$$

Лемма доказана по индукции.

По определению конечного автомата:

$$w \in L(M) \iff \exists q \in F : \langle q_0, w\rangle \vdash_M \langle q, \varepsilon\rangle$$

По лемме это равносильно следующему:

$$\exists q \in F : q_0 \vdash_{G_M} wq \iff q_0 \vdash_{G_M} wq \vdash_{G_M,1} w \iff w \in L(G)$$

Здесь воспользовались тем, что правило $(q \to \varepsilon) \in P$.

$\Longleftarrow$ Докажем, что если язык $L = L(G)$ задаётся праволинейной грамматикой $G = \langle N, \Sigma, P, S\rangle$, то он является автоматным. Заменим правила вида $A \to w$ на правила $A \to wF$, где $F \in N$ — новый нетерминальный символ общий для всех правил, введём также правило $F \to \varepsilon$. Таким образом, правила имеют теперь два вида:

$$A \to wB,\ A,\ B \in N$$

$$F \to \varepsilon$$

Таким образом, праволинейной грамматике G сопоставим конечный автомат $M = \langle N, \Sigma, \Delta, S, \{F\}\rangle$, где $\Delta = \{\langle A, w\rangle \to B | A \to wB \in P\}$. Эквивалентность доказывается аналогично первой части, но вместо индукции по длине слова будет индукция по числу переходов.

■

Вопрос 10

Основные определения

Def. Грамматика $G = \langle N, \Sigma, P, S\rangle$ называется контестно-свободной, если правила грамматики имеют вид $P = \{A \to \alpha | A \in N, \alpha \in (N \cup \Sigma)^*\}$.

Def. Контестно-свободный язык $L = L(G)$ — язык, задаваемый контестно-свободной грамматикой G.

Примеры контестно-свободных языков

Пример: Язык $L = \{w | w \in (0+1)^*\}$ является КС-языком.

Доказательство: Укажем явно контекстно-свободную грамматику: $G = \langle N, \Sigma, P, S\rangle$, где $N = \{S\}$, $\Sigma = \{0, 1\}$, правила: $(S \to 0S)$, $(S \to 1S)$, $(S \to \varepsilon)$. ■

Пример: Язык $L = a^n b^{2n}$ является КС-языком.

Доказательство: Укажем явно контекстно-свободную грамматику: $G = \langle N, \Sigma, P, S\rangle$, где $N = \{S\}$, $\Sigma = \{a, b\}$, правила: $(S \to aSbb)$, $(S \to abb)$. ■

Замкнутость КС-языков относительно операций

Утверждение. КС-языки замкнуты относительно операции объединения.

Доказательство: Пусть L_1 и L_2 — КС-языки, S_1 и S_2 — стартовые нетерминальные символы соответствующих грамматик. Построим КС-грамматику для языка $L_1 \cup L_2$. Введём новый стартовый символ S' такой, что выполняются правила $(S' \to S_1)$ и $(S' \to S_2)$, остальные правила добавим из языков L_1 и L_2.

Покажем, что $S' \vdash w \iff S_1 \vdash w \vee S_2 \vdash w$. Пусть $S' \vdash w$. Так как правила $(S' \to S_1)$ и $(S' \to S_2)$ — единственные правила, где S' присутствует в левой части, то тогда либо $S' \vdash S_1 \vdash w$, либо $S' \vdash S_2 \vdash w$. Пусть $S_1 \vdash w$. Так как присутствует правило $(S' \to S_1)$, то по определению получим, что $S' \vdash S_1 \vdash w$. Аналогично в случае, когда $S_2 \vdash w$. ■

Утверждение. КС-языки замкнуты относительно операции конкатенации.

Доказательство: Доказывается аналогично. Построим КС-грамматику для языка L_1L_2, добавив правило $(S' \to S_1S_2)$. ■

Утверждение. КС-языки замкнуты относительно итерации Клини.

Доказательство: Пусть L — КС-язык, S — соответствующий стартовый нетерминал. Покажем, что $L*$ тоже является КС-языком. Построим для него контекстно-свободную грамматику: S' — новый стартовый нетерминал, новые правила: $(S' \to SS')$, $(S' \to \varepsilon)$. ■

Незамкнутость КС-языков относительно операций

Утверждение. КС-языки не замкнуты относительно пересечения.

Доказательство: Рассмотрим КС-язык $L_1 = \{a^nb^mc^m\}$, правила соответствующей грамматики: $(S \to AT)$, $(A \to aA)$, $(T \to bTc)$, $(A \to \varepsilon)$, $(T \to \varepsilon)$. Также рассмотрим КС-язык $L_2 = \{a^nb^nc^m\}$. Здесь n, $m \in \mathbb{N}$. $L_1 \cap L_2 = \{a^nb^nc^n\}$ не является КС-языком по лемме о разрастании для КС-языков. ■

Утверждение. КС-языки не замкнуты относительно дополнения.

Доказательство: Рассмотрим язык $L = \{a^nb^nc^n\}$, не являющийся КС-языком. Язык $\overline{L} = \{a^kb^lc^m | k \neq l \vee l \neq m \vee k \neq m\}$ является КС-языком. ■

Альтернативное доказательство: Пусть КС-языки замкнуты относительно дополнения. Тогда если L — КС-язык, то $\overline{L}$ — тоже КС-язык. Так как КС-языки замкнуты относительно объединения, то $\overline{L_1} \cup \overline{L_2}$ — КС-язык. По закону де Моргана, $\overline{\overline{L_1} \cup \overline{L_2}} = L_1 \cap L_2$, но КС-языки не замкнуты относительно пересечения. Если $L_1 \cap L_2$ - не КС-язык, то $\overline{\overline{L_1} \cup \overline{L_2}}$ тоже не КС-язык. Противоречие. ■

Вопрос 11

Основные определения

Пусть дана контекстно-свободная грамматика $G = \langle N, \Sigma, P, S \rangle$, где $N \cap \Sigma = \emptyset$, $P = \{A \to \alpha | A \in N, \alpha \in (N \cup \Sigma)^*\}$ — правила грамматики.

Def. Символ $Y \in N$ называется порождающим, если существует слово $w \in \Sigma^*$, такое что $Y \vdash w$.

Def. Символ $D \in N$ называется достижимым, если существуют некоторые φ, $\psi \in (N \cup \Sigma)^*$, такие что $S \vdash \varphi D \psi$.

Def. Символ $U \in N$ называется бесполезным, если он непорождающий или недостижимый.

Def. Символ $E \in N$ называется ε-порождающим, если $E \vdash \varepsilon$.

Удаление бесполезных символов

Утверждение. Для любой контекстно-свободной грамматики существует эквивалентная КС-грамматика без бесполезных символов.

Доказательство: Приведём алгоритм преобразования КС-грамматики: сначала найдём непорождающие символы, удалим их из грамматики, затем найдём и удалим недостижимые символы.

Пусть G_1 — грамматика, преобразованная из G путём удаления непорождающих символов и всех правил, содержащих непорождающие символы. Покажем, почему $L(G) = L(G_1)$. $L(G_1) \subset L(G)$, так как количество правил уменьшается. Пусть $w \in L(G) \setminus L(G_1)$, тогда в дереве вывода есть непорождающий символ C:

$$S \vdash \varphi C \psi \vdash w,\ w = xyz$$

Но $\varphi \vdash x$, $C \vdash y$, $\psi \vdash z$, откуда C — порождающий символ. Противоречие.

Пусть G_2 — грамматика, преобразованная из G_1 путём удаления всех недостижимых символов и правил, содержащих их. Покажем, почему $L(G_1) = L(G_2)$. $L(G_2) \subset L(G_1)$, так как количество правил уменьшается. Пусть $w \in L(G_1) \setminus L(G_2)$, тогда существует недостижимый символ D, что $S \vdash \varphi D \psi \vdash w$, но тогда D по определению является достижимым. Противоречие.

Проверим, что не появилось новых непорождающих символов. Пусть B — непорождающий символ в G_2. Тогда B — порождающий символ в G_1, и $B \vdash u$, но тогда B достижим в G_1.

Проверим, что не появилось новых недостижимых символов. Пусть на пути вывода $B \vdash u$ был недостижимый символ C, но тогда $S \vdash_{G_1} B \vdash C \vdash u$, и C — достижимый символ. ■

Удаление ε-правил

Приведём алгоритм нахождения ε-порождающих символов. Если правило $(A \to \varepsilon) \in P$. то A — ε-порождающий. Если $(A \to B_1, \ldots, B_n) \in P$, $B_1 \ldots B_n$ — ε-порождающий, то A — ε-порождающий.

Th. Для любого контекстно-свободного языка существует грамматика без ε-правил, порождающая язык $L \setminus \{\varepsilon\}$.

Доказательство: Пусть $\langle N, \Sigma, P, S \rangle$ — контекстно-свободная грамматика, $L = L(G)$. Построим новую грамматику $G' = \langle N, \Sigma, P', S' \rangle$, где P' обладает следующими свойствами:

1. $P' \supset \{(A \to \alpha) \in P | \alpha \neq \varepsilon\}$;
2. Пусть B — ε-порождающий в исходной грамматике G, тогда если $(A \to \alpha B \gamma) \in P'$, $\alpha\gamma \neq \varepsilon$, то $A \to \alpha\gamma \in P'$

Индукцией по длине вывода можно доказать, что $A \vdash_G w \iff A \vdash_{G'} w$, $w \neq \varepsilon$. ■

Вопрос 12

О нормальной форме Хомского

Пусть дана контекстно-свободная грамматика $G = \langle N, \Sigma, P, S \rangle$, $P = \{A \to \alpha | A \in N, \alpha \in (N \cup \Sigma)^*\}$ — правила грамматики.

Def. КС-грамматика находится в нормальной форме Хомского, если все правила имеют такой и только такой вид:

1. $A \to a \, (A \in N, a \in \Sigma)$;
2. $A \to BC \, (B, C \in N; B, C \neq S)$;
3. $S \to \varepsilon$.

Алгоритм приведения к нормальной форме Хомского

Любую КС-грамматику можно привести к нормальной форме Хомского с помощью алгоритма, который состоит из следующих шагов:

1. Удаление непорождающих символов
2. Удаление недостижимых символов
3. Удаление смешанных правил $D \to aBc$
4. Удаление длинных правил $A \to A_1 A_2 A_3 A_4$
5. Удаление ε-порождающих символов
6. Обработка пустого слова
7. Удаление унарных (одиночных) правил $A \to B$

После удаления непорождающих и недостижимых символов будет получена эквивалентная грамматика. Подробное доказательство этого есть в вопросе 11. Обозначим полученную грамматику за G_2.

Для удаления смешанных правил сделаем замену правила вида $A \to dBcEf$ на правила следующего вида:

$$A \to DBCEF$$

$$D \to d$$

$$C \to c$$

$$F \to f$$

Обозначим полученную грамматику за G_3. Покажем, что $w \in L(G_2) \iff w \in L(G_3)$. В основе доказательства лежит идея, что в дереве вывода нетерминальные символы можно выводить в любом порядке.

$\Longrightarrow$

$$A \vdash_1 dBcEf \vdash dw_BcEf \ (G_2)$$
$$A \vdash DBCEF \vdash dBcEf \vdash dw_BcEf \ (G_3)$$

$\Longleftarrow$

Аналогично, раскрываем в первую очередь правила вида $D \to d$.

Удалим длинные правила вида $B \to A_1A_2 \dots A_n$ с помощью замены на правила следующего вида:

$$B \to A_1B_1$$
$$B_1 \to A_2B_2$$
$$B_2 \to A_3B_3$$
$$\vdots$$
$$B_{n-1} \to A_{n-1}A_n$$

Обозначим полученную грамматику за G_4. Покажем, что $w \in L(G_3) \iff w \in (G_4)$.

$\Longrightarrow$ Пусть $w \in L(G_3)$. Тогда $S \vdash \varphi B\psi \vdash_1 \varphi A_1A_2 \dots A_n\psi \vdash w$. Посмотрим, что происходит в G_4:

$$S \vdash \varphi B\psi \vdash \varphi A_1B_1\psi \vdash \varphi A_1A_2B_2\psi \vdash \dots \vdash \varphi A_1A_2 \dots A_{n-2}B_{n-1}\psi \vdash \varphi A_1 \dots A_n\psi \vdash w$$

$\Longleftarrow$ Пусть $w \in L(G_4)$. Рассмотрим вывод $S \vdash w$. Возможны два варианта:

1. B_k не встречается на пути вывода. Тогда слово выводимо и в G_4, и в G_3;
2. B_K встречается на пути вывода. Тогда встречаются и все нетерминалы вида B_1, …, B_{n-1} по построению правил. Тогда делаем в обратную сторону то, что делали, когда доказывали из G_3 в G_4.

Теперь удалим ε-правила. Это можно сделать так, как в вопросе 11, получим эквивалентную грамматику G_5.

Однако если $S \vdash \varepsilon$, то возможны варианты:

1. $S \to \varepsilon$ — и правило убрано;

2. $S \to AB$, где $A \vdash \varepsilon$, $B \vdash \varepsilon$ — рекурсивный спуск.

Построим грамматику G_6 следующим образом:

1. S' — новый стартовый нетерминал;
2. $S' \to S$ — новое правило;
3. Если $S \vdash \varepsilon$, то $S' \to \varepsilon$ — новое правило.

Данная грамматика будет эквивалентной.

Теперь остались правила вида:

1. $A \to a$;
2. $A \to BC$;
3. $A \to B$;
4. $S \to \varepsilon$.

Рассмотрим последовательности правил:

$$B \to B_1 \to B_2 \to \cdots \to B_n \to CD$$

$$B \to B_1 \to B_2 \to \cdots \to B_n \to a$$

Удалим унарные одиночные правила, заменим последовательности правил на $B \to CD$ и $B \to a$ соответственно. Обозначим полученную грамматику за G_7.

Покажем, что $L(G_6) = L(G_7)$.

$\Longleftarrow$ Для всех правил вида $B \to a$ или $B \to CD$ верно, что $B \vdash_{G_6} a$, $B \vdash_{G_6} CD$. Никаких новых выводов добавлено не было, тем самым $L(G_7) \subseteq L(G_6)$.

$\Longrightarrow$ Покажем, что $B \vdash_{G_6} w \Longrightarrow B \vdash_{G_7} w$ индукцией по длине вывода.

База. Вывод за один шаг: $B \vdash_{G_6,1} w \Longrightarrow (B \to w) \in P_{G_6}, w \in \Sigma \Longrightarrow (B \to w) \in P_{G_7} \Longrightarrow B \vdash_{G_7,1} w$.

Переход. Добавленное правило имеет вид $B \to C_1 \dots C_n$, где $n \in \{1, 2\}$, $B \to C_1 \dots C_n \vdash_{G_6} w_1 \dots w_n$, по предположению индукции $B_i \vdash_{G_7} w_i$.

Пусть $n = 2$. Тогда правило имеет вид $B \to CD$, $A \to_{G_7} CD \vdash_{G_7} w_1 w_2$.

Пусть $n = 1$. Тогда вывод начинается с правила $B \to C_1$. Рассмотрим первое правило вывода, не являющееся одиночным. Тогда:

1. $B \vdash C_1 \vdash C_2 \vdash \cdots \vdash C_m \vdash a$, $a \in \Sigma$, тогда по построению существует правило $B \to a$, $B \vdash_{G_7} a$;
2. $B \vdash C_1 \vdash C_2 \vdash \cdots \vdash CD \vdash w_1 w_2$. По построению существует правило $B \to CD$, по предположению индукции $B \vdash_{G_7} CD \vdash w_1 w_2$.

Построенная грамматика G_7 — грамматика в нормальной форме Хомского.

Вопрос 13

Основные сведения

Алгоритм Кока-Янгера-Касами принимает на вход грамматику G в нормальной форме Хомского и слово w. В качестве выхода выдаётся информация, принадлежит ли слово w языку L, который задаётся грамматикой G.

Алгоритм и его реализация

В основе алгоритма Кока-Янгера-Касами лежит динамическое программирование по подотрезкам. Пусть $dp[A][i][j]$ — трёхмерный булев массив, где поддерживается инвариант: $dp[A][i][j] = True \iff A \vdash w[i:j]$. Инициализируем массив dp следующим образом:

$$dp[A][i][i] = True \iff \begin{cases} a = w[i] \\ (A \to a) \in P \end{cases}$$

Ниже приведена реализация на псевдодиалекте Python:

```
def check_word(grammar, word) -> bool:
    dp = array of False

    if word == '':
        return (S -> e) in grammar.rules

    dp = init_one_letters(dp, grammar, word)

    for word_length in range(2, len(word) + 1):
        dp = process_words(dp, grammar, word_length)

return dp[S][0][len(word)]

def init_one_letters(dp, grammar, word):
    for index, letter in enumerate(word):
        for (A -> a) in grammar.rules:
            if letter == a:
```

```
                dp[A][index][index + 1] = True
    return dp

def process_words(dp, grammar, word_length):
    for start in range(len(word)):
        end = word_length + start
        for A -> BC in grammar.rules:
            for mid_position in range(start + 1, end - 1):
                dp[A][start][end] |= dp[B][start][mid] & dp[C][mid][end]
    return dp
```

Расскажем вкратце, в чём заключается алгоритм. Во внешнем цикле перебирается длина подслова от 2 до $|w|$. Во внутреннем цикле перебирается начало подслова, запоминаем конец соответствующего подслова. От начала до конца перебираем позицию разреза, и перебираем правила грамматики вида $(A \to BC)$. Если $dp[B][i][k] = True$ и $dp[C][k][j] = True$, то $dp[A][i][j] = True$. Если уже однажды обновили значение $dp[A][i][j]$, то больше его не изменяем. Слово w выводимо в грамматике G, если и только если $dp[S][0][n]$ (если динамика по полуинтервалам) или $dp[S][0][n-1]$ (если динамика по подотрезкам) равен $True$.

Доказательство корректности

Проведём индукцию по длине слова.

База. Пусть $j = i$. Тогда $dp[A][i][j] = True$ появилась на этапе инициализации, что по построению равносильно тому, что правило $(A \to w[i]) \in P$, что эквивалентно тому, что $A \vdash w[i : j]$, $j = i$, то есть $A \vdash w[i]$.

Переход. Пусть $dp[A][i][j] = True$. Тогда по построению и предположению индукции существуют число k и нетерминальные символы B и C, такие что $A \vdash_1 BC$, $B \vdash w[i : k]$, $C \vdash w[k : j]$, откуда $dp[B][i][k] = True$ и $dp[C][k][j] = True$, так как $A \vdash_1 BC$, то $(A \to BC) \in P$, $A \vdash w[i : j]$.

Корректность доказана.

Асимптотика алгоритма

Обработка правил вида $A \to a$ выполняется за $\mathcal{O}(|w| \cdot |P|)$. Проход по всем подстрокам занимает $\mathcal{O}\left(|w|^2\right)$ времени, обработка каждой подстроки занимает $\mathcal{O}(|w| \cdot |P|)$ времени, так присутствует цикл по всем правилам вывода, среди которых правила вида $A \to BC$. Итоговая асимптотика: $\mathcal{O}\left(|w|^3 \cdot |P|\right)$.

Вопрос 14

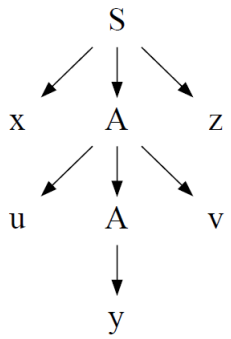
Лемма о разрастании для КС-языков

Лемма. Пусть L — КС-язык. Тогда существует p, такое что для любого слова $w \in L$, длина которого не меньше, чем p, существуют такие слова x, u, y, v, z, принадлежащие Σ^*, что $w = xuyvz$, $|uv| > 0$, $|uyv| \leqslant p$, что для любого $k \in \mathbb{N}$ выполняется, что $xu^k yv^k z \in L$.

Кванторная версия:

$$\exists p : \forall w \in L : |w| \geqslant p : \exists x,\ u,\ y,\ v,\ z \in \Sigma^* : w = xuyvz,\ |uv| > 0,\ |uyv| \leqslant p : \forall k \in \mathbb{N} : xu^k yv^k z \in L$$

Доказательство: Рассмотрим грамматику G в нормальной форме Хомского: $L = L(G)$. Выберем $p = 2^{|N|}$, где $|N|$ — количество нетерминальных символов. Тогда $w \geqslant p = 2^{|N|}$. Дерево вывода является бинарным деревом, и тогда существует «ветвь» дерева вывода уровня хотя бы $|N|$. Воспользуемся принципом Дирихле, рассмотрим «ветвь» максимальной глубины, в ней количество нетерминалов будет хотя бы $|N| + 1$. Тогда существует нетерминал A, такой что $S \vdash xAz \vdash xuAvz \vdash xuyvz$ и $A \vdash uAv$, который повторяется не менее двух раз. Среди всех возможных нетерминалов A выберем тот, который находится ниже всех, то есть его глубина относительно корня наибольшая.



Покажем, что $|uyv| \leqslant p = 2^{|N|}$. Пусть $|uyv| > p = 2^{|N|}$, тогда для дерева со стартом в A можно сделать те же самые операции, значит, существует уровень, который больше $|N|$, значит, существует в поддереве пара $B \vdash B$, и A — не самый глубокий нетерминал.

Покажем, что $|uv| > 0$. Для любого нетерминального символа $C \in N$ верно, что C не является ε-порождающим. Рассмотрим D такой, что $D \underset{1}{\vdash} KA$, $K \vdash r$, где r — суффикс u, $|r| > 0$. Отсюда следует, что $|uv| \geqslant |u| \geqslant |r| > 0$. ■

Примеры языков, не являющихся КС-языками

Пример: Язык $L = \{a^n b^n c^n | n \in \mathbb{Z}^+\}$ не является КС-языком.

Доказательство: Рассмотрим фиксированное p, $w = a^p b^p c^p$, $w = xuyvz$, $|uv| > 0$, $|uyv| \leqslant p$. Заметим, что в uyv не может быть трёх разных букв из $\{a, b, c\}$. То же самое верно и для uv. Не умаляя

общности, полагаем $|uyv|_c = 0$, $|uyv|_b > 0$. Пусть $k = 2$. Тогда $|xu^2yv^2z|_b = |xuyvz|_b + |uv|_b = p + |uv|_b > p$, $|xu^2yv^2z|_c = p + |uv|_c = p + 0 = p$. Но тогда $xu^2yv^2z \notin L$. ■

Вопрос 15

Определения

Def. Автомат с магазинной памятью (МП-автомат) — кортеж $M = \langle Q, \Sigma, \Gamma, \Delta, q_0, F \rangle$, где:

1. Q — множество состояний, Q — конечное множество, то есть $|Q| < \infty$;
2. Σ — алфавит, $|\Sigma| < \infty$;
3. Γ — стековый алфавит, $|\Gamma| < \infty$;
4. $\Delta \subset (Q \times \Sigma^* \times \Gamma^*) \times (Q \times \Gamma^*)$ — множество переходов, $|\Delta| < \infty$;
5. $q_0 \in Q$ — стартовое состояние;
6. $F \subset Q$ — множество завершающих состояний.

Переходы имеют вид $\langle q_1, w, \alpha \rangle \to \langle a_2, \beta \rangle$, $w \in \Sigma^*$, $\alpha \in \Gamma^*$, *то есть когда находимся в состоянии* q_1, *снимаем со стека слово* α, *стек растёт слева направо, читаем слово* w, *переходим в состояние* q_2, *добавляем на стек слово* β.

Def. Конфигурация МП-автомата M — кортеж $\langle q, u, \gamma \rangle$, где $q \in Q$, $u \in \Sigma^*$, $\gamma \in \Gamma^*$.

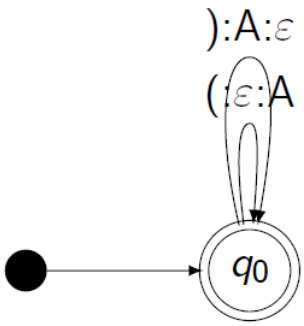
Def. Отношение выводимости $\vdash$ — наименьшее рефлексивное транзитивное отношение, что для любого перехода $(\langle q_1, u, \alpha \rangle \to \langle q_2, \beta \rangle) \in \Delta$ выполнено следующее:

$$\forall v \in \Sigma^*, \eta \in \Gamma^* : \langle q_1, uv, \eta\alpha \rangle \vdash \langle q_2, v, \eta\beta \rangle$$

Языки, распознаваемые МП-автоматами

Def. Пусть M — МП-автомат, язык $L(M)$, распознаваемый МП-автоматом M — множество $\{w \in \Sigma^* | \exists q \in F : \langle q_0, w, \varepsilon \rangle \vdash \langle q, \varepsilon, \varepsilon \rangle\}$.

Пример: Язык правильных скобочных последовательностей распознаётся следующим МП-автоматом:

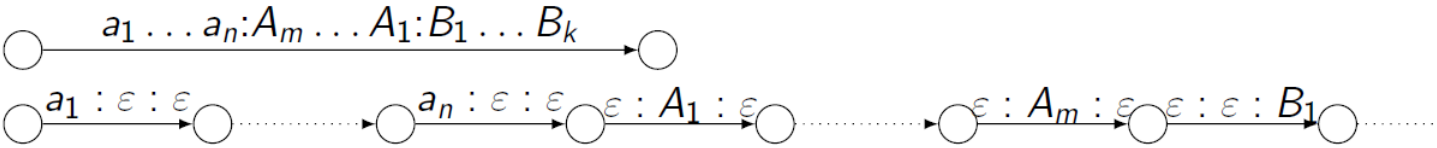


Упрощения МП-автоматов

Утверждение. Для любого МП-автомата существует эквивалентный МП-автомат, для которого выполнено соотношение:

$$\forall\,(\langle q_1, u, \alpha\rangle \to \langle q_2, \beta\rangle) \in \Delta : |u| \leqslant 1, |\alpha| + |\beta| \leqslant 1$$

Доказательство:

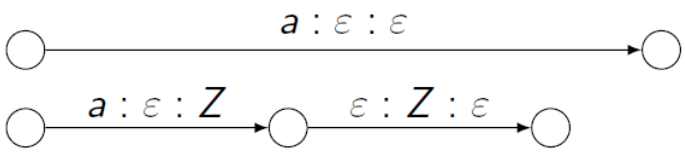


■

Утверждение. Для любого МП-автомата существует эквивалентный МП-автомат, для которого выполнено соотношение:

$$\forall\,(\langle q_1, u, \alpha\rangle \to \langle q_2, \beta\rangle) \in \Delta : |u| \leqslant 1, |\alpha| + |\beta| = 1$$

Доказательство:



■

Вопрос 16

В вопросах 16 и 17 предлагается доказать в одну из сторон следующую теорему:

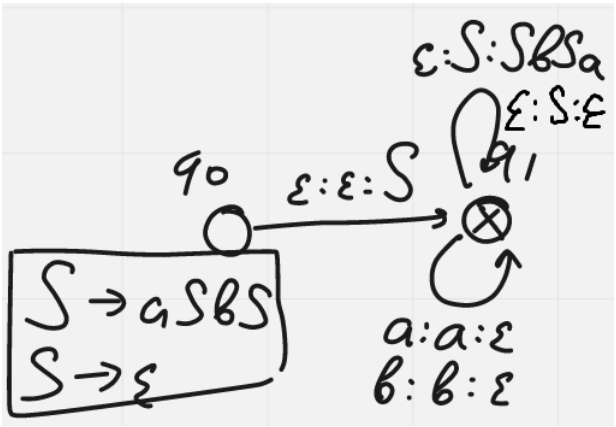
Th. Язык L является МП-автоматным тогда и только тогда, когда L является контекстно-свободным.

Построение автомата по грамматике

Доказательство:

$\Longleftarrow$ Рассмотрим контекстно-свободную грамматику $G = \langle N, \Sigma, P, S\rangle$. Автомат строим следующим образом: q_0 — стартовое (начальное) состояние, q_1 — единственное завершающее состояние, $Q = \{q_0, q_1\}$. Переходы из q_1 в q_1 имеют вид либо $\langle q_1, a, a\rangle \to \langle q_1, \varepsilon\rangle$, если a — некоторый терминальный символ, либо $\langle q_1, \varepsilon, S\rangle \to \langle q_1, SbSa\rangle$, если существует правило вида $S \to aSbS$, где $S \in N$. *Заметим, что при обработке правил, где левая часть — некоторый нетерминал, мы добавляем в стек развёрнутую правую часть правила.*

Чтобы было видно, что происходит, приведём пример. Пусть правила КС-грамматики G следующие: $(S \to \varepsilon), (S \to aSbS)$. Тогда МП-автомат по КС-грамматике будет таким:



Докажем следующую лемму:

Лемма. $A \vdash w \Longleftrightarrow \langle q_1, w, A\rangle \vdash \langle q_1, \varepsilon, \varepsilon\rangle$

Докажем индукцией по длине дерева вывода (количеству рёбер в дереве). Считаем её равной k.

База. $k = 1$, $A \vdash_1 w$. Пусть $w = w_1 w_2 \dots w_n$. Так как $A \to w_1 \dots w_n$, то $\langle q_1, \varepsilon, A\rangle \to \langle q_1, w^R\rangle$. Значит:

$$\langle q_1, w, A\rangle \vdash \langle q_1, w, w_n w_{n-1} \dots w_1\rangle \vdash$$
$$\vdash \langle q_1, w_2 \dots w_n, w_n w_{n-1} \dots w_2\rangle \vdash$$
$$\vdash \langle q_1, w_3 \dots w_n, w_n \dots w_3\rangle \vdash \dots$$
$$\vdash \langle q_1, w_n, w_n\rangle \vdash \langle q_1, \varepsilon, \varepsilon\rangle$$

Переход. Посмотрим на первое раскрытие: $A \vdash_1 \alpha \vdash w$. Здесь $\alpha = \alpha_1 \dots \alpha_n \vdash w_1 \dots w_n = w$, $\alpha_i \in N \cup \Sigma$. Тогда $\langle q_1, w, A\rangle \vdash \langle q_1, w, \alpha_n \dots \alpha_1\rangle$.

Если $\alpha_n \in N$, то тогда $\alpha_n \vdash w_n$, и по предположению индукции $\langle q_1, w_n, \alpha_n\rangle \vdash \langle q_1, \varepsilon, \varepsilon\rangle$.

Если $\alpha_n \in \Sigma$, то $\alpha_n \vdash w_n$, $\alpha_n = w_n$, и тогда $\langle q_1, w_n, w_n\rangle \vdash \langle q_1, \varepsilon, \varepsilon\rangle$, так как это правило грамматики.

В итоге:

$$\langle q_1, w, A\rangle \vdash \langle q_1, w, \alpha_n \dots \alpha_1\rangle \vdash \langle q_1, w_1 \dots w_n, \alpha_n \dots \alpha_1\rangle \vdash \langle q_1, w_2 \dots w_n, \alpha_n \dots \alpha_2\rangle \vdash \dots \vdash \langle q_1, w_n, \alpha_n\rangle \vdash \langle q_1, \varepsilon, \varepsilon\rangle.$$

Теперь нужно показать в обратную сторону: если $\langle q_1, w, A\rangle \vdash \langle q_1, \varepsilon, \varepsilon\rangle$, то $A \vdash w$. Это сделаем с помощью индукции по количеству переходов.

База. Пусть всего один переход, тогда $A \in \Sigma$, и $\langle q_1, w, A\rangle \vdash_1 \langle q_1, \varepsilon, \varepsilon\rangle$, откуда $w = A$, и $A \vdash w$, так как $\vdash$ обладает свойством рефлексивности.

Переход. $\langle q_1, w, A\rangle \vdash_k \langle q_1, \varepsilon, \varepsilon\rangle$, где $k > 1$. Тогда $A \in N$, откуда если $A \to \alpha_1 \dots \alpha_n$, где $\alpha_m \in (N \cup \Sigma)$, то:

$$\langle q_1, w, A\rangle \vdash_1 \langle q_1, w, \alpha_n \dots \alpha_1\rangle \vdash \langle q_1, \varepsilon, \varepsilon\rangle$$

Note. За один шаг мы снимаем ровно один элемент со стека:

$$\alpha_1 \to \beta_1$$
$$\alpha_2 \to \beta_2$$
$$\vdots$$
$$\alpha_n \to \beta_n$$

Пусть $\alpha_1 \to \beta_1$, $\alpha_1 \in N$, $\langle q_1, w, \alpha_n \dots \alpha_1\rangle \vdash_1 \langle q_1, w, \alpha_n \dots \alpha_2 \beta_1^R\rangle$. Дождёмся, пока на стеке останется $\alpha_n \dots \alpha_2$: так как в конце стек пустой и мы считаем ровно 1 символ. Тогда в этом моменте:

$$\langle q_1, w, \alpha_n \dots \alpha_1\rangle \vdash \langle q_1, w', \alpha_n \dots \alpha_2\rangle \Longrightarrow \exists w_1 : w = w_1 w' : \langle q_1, w_1, \alpha_1\rangle \vdash \langle q_1, \varepsilon, \varepsilon\rangle \overset{hypothesis}{\Longrightarrow} \alpha_1 \vdash w_1$$

По аналогии, $\alpha_m \vdash w_m$ для любого $m = \overline{1, n}$.

Итого имеем, что $A \to \alpha_1 \dots \alpha_n$, $\alpha_m \vdash w_m$, $w_1 \dots w_n = w$. Из всего этого следует, что $A \vdash w_1 \dots w_n = w$. Переход доказан.

Из доказанной леммы следует, что:

$$w \in L(G) \iff S \vdash w \iff \langle q_1, w, S\rangle \vdash \langle q_1, w, S\rangle \vdash \langle q_1, \varepsilon, \varepsilon\rangle$$
$$w \in L(M) \iff \langle q_0, w_\varepsilon\rangle \vdash_1 \langle q_1, w, S\rangle \vdash \langle q_1, \varepsilon, \varepsilon\rangle$$

■

Вопрос 17

Построение грамматики по автомату

Доказательство:

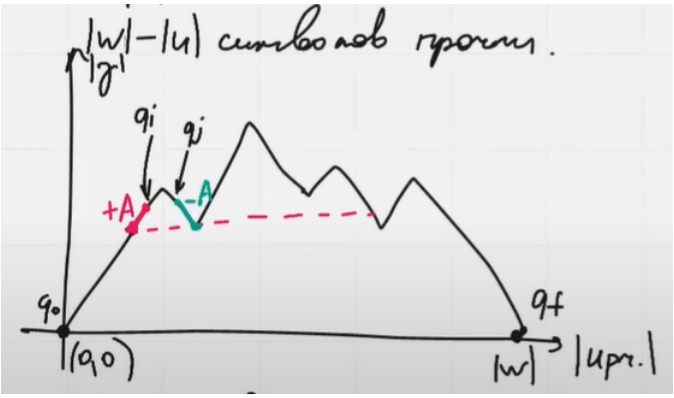
$\Longrightarrow$ Пусть $w \in L(M)$, M — МП-автомат, для которого выполнено соотношение:

$$\forall (\langle q_1, u, \alpha\rangle \to \langle q_2, \beta\rangle) \in \Delta : |u| \leqslant 1, |\alpha| + |\beta| = 1$$

Для вывода построим график «длина стека» от префикса w:

$$\langle q_0, w, \varepsilon\rangle \to \langle q, u, \gamma\rangle$$

На графике — точка $(|\gamma|, |w| - |u|)$.



Далее зададим грамматику $G = \langle N, \Sigma, P, S\rangle$, где:

$N = S \cup \{A_{ij} | q_i, q_j \in Q\}$ *Под* A_{ij} *подразумевается то, что выводится на пути между* q_i *и* q_j *без изменения стека.* P — объединение следующих множеств:

1. $\{A_{ii} \to \varepsilon | q_i \in Q\}$
2. $\{S \to A_{0j} | q_j \in F\}$
3. $\{A_{ij} \to aA_{rs}bA_{tj} \ | \alpha \vee \beta\}$, где:
   (a) α: условие, что $\langle q_i, a, \varepsilon\rangle \to \langle q_r, A\rangle \in \Delta$;
   (b) β: условие, что $\langle q_r, b, A\rangle \to \langle q_j, \varepsilon\rangle \in \Delta$.

Теперь нужно доказать следующую лемму:

Лемма. $A_{ij} \vdash_G w \iff \langle q_i, w, \varepsilon\rangle \vdash_M \langle q_j, \varepsilon, \varepsilon\rangle$

$\Longrightarrow$ Докажем индукцией по длине вывода в грамматике.

База. Вывод за один шаг. $A_{ij} \to \varepsilon$. Тогда $i = j$, $w = \varepsilon$, $\langle q_i, \varepsilon, \varepsilon\rangle \vdash \langle q_i, \varepsilon, \varepsilon\rangle$.

Переход. Пусть $A_{ij} \vdash w$ за k шагов. Тогда $A_{ij} \vdash aA_{rs}bA_{tj}$, при этом:

$$\langle q_i, a, \varepsilon\rangle \to \langle a_r, A\rangle \ (1)$$

$$\langle q_s, b, A\rangle \to \langle q_t, \varepsilon\rangle \ (3)$$

Слово w имеет вид $aubv$. Тогда, используя предположение индукции, можем получить:

$$A_{rs} \vdash u \Longrightarrow \langle q_r, u, \varepsilon\rangle \overset{(2)}{\vdash} \langle q_s, \varepsilon, \varepsilon\rangle$$

$$A_{tj} \vdash v \Longrightarrow \langle q_t, v, \varepsilon\rangle \overset{(4)}{\vdash} \langle q_j, \varepsilon, \varepsilon\rangle$$

Тогда $\langle q_i, aubv, \varepsilon\rangle \overset{(1)}{\vdash} \langle q_r, ubv, A\rangle \overset{(2)}{\vdash} \langle q_s, bv, A\rangle \overset{(3)}{\vdash} \langle q_t, v, \varepsilon\rangle \overset{(4)}{\vdash} \langle q_j, \varepsilon, \varepsilon\rangle$.

$\Longleftarrow$ Проведём индукцию по количеству переходов k, которые необходимы для того, чтобы $\langle q_i, w, \varepsilon\rangle \vdash \langle q_j, \varepsilon, \varepsilon\rangle$.

База. $k = 0$. Тогда $\langle q_i, w, \varepsilon\rangle \vdash_0 \langle q_j, \varepsilon, \varepsilon\rangle$, откуда $w = \varepsilon$ и $q_i = q_j$, $A_{ij} = A_{ii}$, так как есть правило $A_{ii} \to \varepsilon$, то $A_{ii} \vdash \varepsilon$.

Переход. $\langle q_i, w, \varepsilon\rangle \vdash_k \langle q_j, \varepsilon, \varepsilon\rangle$. Так как стек пустой, то:

$$\langle q_i, w, \varepsilon\rangle \vdash_1 \langle q_r, u, A\rangle$$

A существует, так как мы либо кладём, либо снимаем со стека. Пусть $q_s \to q_t$ — это момент, когда A снят со стека. Тогда:

$$\langle q_s, v, A\rangle \vdash_1 \langle q_t, x, \varepsilon\rangle \vdash \langle q_j, \varepsilon, \varepsilon\rangle$$

$$\langle q_i, a, \varepsilon\rangle \to \langle q_r, A\rangle \in \Delta : w = au$$

$$\exists u' : u = u'v,\ \langle q_r, u', \varepsilon\rangle \vdash \langle q_s.\varepsilon, \varepsilon\rangle$$

Пользуясь предположением индукции, получаем:

$$A_{rs} \vdash u'$$

$$\langle q_s, b, A\rangle \vdash \langle q_t, \varepsilon\rangle \in \Delta : v = bx$$

$$\langle q_t, x, \varepsilon\rangle \vdash \langle q_j, \varepsilon, \varepsilon\rangle \Longrightarrow A_{tj} \vdash x$$

Так как правило $A_{ij} \to aA_{rs}bA_{tj} \in P$, то $A_{ij} \vdash aA_{rs}bA_{tj} \vdash au'bx = au'v = au = w$. Предположение доказано.

Из леммы будет следовать теорема следующим образом:

$$w \in L(G) \Longleftrightarrow S \vdash w \Longleftrightarrow \exists A_{0j}\,(q_j \in F) : S \vdash_1 A_{0j} \vdash w \Longleftrightarrow \exists A_{0j}\,(q_j \in F) : \langle q_0, w, \varepsilon\rangle \vdash \langle q_j, \varepsilon, \varepsilon\rangle \Longleftrightarrow$$
$$\exists q_j \in F : \langle q_0, w, \varepsilon\rangle \vdash \langle q_j, \varepsilon, \varepsilon\rangle \Longleftrightarrow w \in L(M)$$

■

Часть 3. Парсеры

Основная информация об алгоритме Эрли

Основные определения

Пусть $w \in \Sigma^*$ — слово на входе. На вход подаётся контекстно-свободная грамматика $G = \langle N, \Sigma, P, S \rangle$.

Def. Ситуация — объект вида $(A \to \alpha \cdot \beta, i)$, где правило $(A \to \alpha\beta) \in P$, $\cdot$ — вспомогательный символ, который не принадлежит ни Σ, ни N, $i \in [0; |w|]$.

Def. D_j — множество ситуаций вида $(A \to \alpha \cdot \beta, i)$ таких, что $\alpha \vdash w[i : j]$.

Note. Вывод считаем левосторонним: пусть правило $(A \to \beta) \in P$, тогда:

$$S \vdash \varphi A \psi \vdash_1 \varphi\beta\psi \Longrightarrow \varphi \in \Sigma^*$$

Note. Ситуация $(A \to \alpha \cdot \beta, i) \in D_j$ означает, что:

1. $S \vdash w[0 : i]$, где S — стартовый нетерминальный символ;
2. $(A \to \alpha\beta) \in P$;
3. $\alpha \vdash w[i : j]$.

Если вдруг с алгоритмом Эрли вы встречаетесь впервые, то представьте, что точка играет роль курсора, слева от которого то, что уже было введено, а справа находится то, что предстоит обработать.

Note. Для удобства вводится новый стартовый нетерминальный символ S', а также в грамматику G добавляется правило $(S' \to S)$. На выводимость слова это не влияет.

Операции

Всего в алгоритме Эрли поддерживаются три операции: **Scan**, **Predict**, **Complete**. *Проще говоря, Scan отвечает за «прочтение» нового символа слова, то есть появляются ситуации, которые соответствуют тому, как префикс слова мог быть выведен, Predict отвечает за генерацию возможных ситуаций при прочтении следующего символа, который является нетерминальным, то есть как бы «предсказывает», по каким правилам слово может быть выведено дальше, Complete отвечает как бы за проверку того, было ли правильным «предсказание» со стороны операции Predict.* Теперь переходим к формальным определениям операций:

$$\text{Scan: } \begin{cases} (A \to \alpha \cdot a\beta, i) \in D_j \\ w[j] = a \end{cases} \Longrightarrow (A \to \alpha a \cdot \beta, i) \in D_{j+1}$$

$$\text{Predict: } \begin{cases} (A \to \alpha \cdot B\beta, i) \in D_j \\ (B \to \gamma) \in P \end{cases} \Longrightarrow (B \to \cdot\gamma, j) \in D_j$$

$$\text{Complete: } \begin{cases} (B \to \gamma\cdot, k) \in D_j \\ (A \to \alpha \cdot B\beta, i) \in D_k \end{cases} \Longrightarrow (A \to \alpha B \cdot \beta, i) \in D_j$$

Инициализация: $(S' \to S, 0) \in D_0$. Слово выводимо в грамматике G тогда и только тогда, когда ситуация $(S' \to S\cdot, 0) \in D_{|w|}$.

Вопрос 18

См. вопрос 19 (доказательство слева направо).

Вопрос 19

Вводная информация

Def. Алгоритм называется полным, если для любого ввода гарантируется, что хотя бы одно решение, если такое существует, будет выведено.

Note. В этом вопросе предстоит доказать, что слово выводимо в грамматике G тогда и только тогда, когда ситуация $(S' \to S\cdot, 0) \in D_{|w|}$. Если внимательно посмотреть на лемму, которая использовалась для доказательства корректности алгоритма Эрли, то можно понять, что из этой леммы следует полнота алгоритма Эрли. *Для доказательства полноты достаточно привести доказательство справа налево.* Сначала докажем эту лемму в общем случае:

Основная лемма об инварианте

Лемма. Ситуация $(A \to \alpha \cdot \beta, i) \in D_j$ тогда и только тогда, когда выполнено следующее:

$$\exists \varphi, \psi \in (N \cup \Sigma)^* : \varphi \vdash w[0:i], \alpha \vdash w[i:j]$$
$$S' \vdash \varphi A \psi \vdash w[0:i] A \psi \vdash_1 \varphi\alpha\beta\psi$$

Доказательство:

$\Longrightarrow$ Сначала докажем необходимость. Сделаем это индукцией по числу t эффективных шагов в алгоритме:

1. Выполнения операции Scan;
2. Добавления в D_i нового элемента

База. $t = 0$, $(S' \to \cdot S, 0) \in D_0$. Покажем, что:

$$S' \vdash \varphi S' \psi \vdash_1 \varphi S \psi \Longrightarrow \varphi = \varepsilon = w[:0]$$

Здесь $\beta = S$, $\alpha = \varepsilon = w[0:0]$, откуда существует $\varphi = \varepsilon = w[0:0] = w[0:i]$, и $\alpha = \varphi$. Тогда $S' \vdash \varepsilon S' \varepsilon \vdash \underbrace{\varepsilon}_{\varphi}\underbrace{\varepsilon}_{\alpha} S\varepsilon$.

Переход. Рассмотрим, как могла появиться ситуация $(A \to \alpha \cdot \beta, i) \in D_j$:

1. Scan
2. Predict
3. Complete

Рассмотрим случай, когда ситуация появилась в результате выполнения операции Scan. Тогда $\alpha = \alpha' a$, при этом $(A \to \alpha' \cdot a\beta, i) \in D_{j-1}$, $w[j-1] = a$. По предположению индукции:

$$S' \vdash w[:i] A \psi \vdash_1 w[:i] \alpha' a \beta \psi, \text{ при этом:}$$
$$\alpha' \vdash w[i : j-1] \Longrightarrow \alpha' a \vdash w[i:j] \Longrightarrow \alpha \vdash w[i:j]$$
$$w[:i]\alpha' a \beta \psi \vdash_0 w[:i]\alpha\beta\psi = \varphi\alpha\beta\psi$$
$$S' \vdash \varphi A \psi \vdash w[0:i] A \psi \vdash \underbrace{\varphi}_{0:i}\underbrace{\alpha}_{i:j}\beta\psi$$

Рассмотрим случай, когда ситуация появилась в результате выполнения операции Predict. Пусть появилась ситуация $(A \to \alpha \cdot \beta, i) \in D_j$, тогда $\alpha = \varepsilon$, $i = j$, но существовало некоторое k и некоторая ситуация вида $(B \to \gamma \cdot A\delta, k) \in D_j$. По предположению индукции:

$$S' \vdash w[0:k] B \psi \vdash_1 w[0:k] \gamma A \delta$$
$$\gamma \vdash w[k:j]$$
$$\text{Раскрывая все } \gamma, \text{ выведем } w[0:k]w[k:j]A\delta \overset{(A\to B)}{\vdash_1} w[0:j]\underbrace{\varepsilon}_{\alpha}\beta\delta$$
$$\alpha \vdash w[i:j]$$

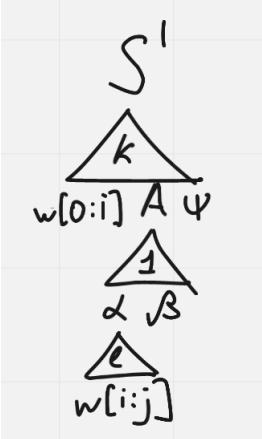
Рассмотрим случай, когда ситуация $(A \to \alpha \cdot \beta, i) \in D_j$ появилась в результате выполнения операции Complete. Тогда $\alpha = \alpha' B$, и при этом $(A \to \alpha' \cdot B\beta, i) \in D_k$ для некоторого k, $(B \to \gamma \cdot, k) \in D_j$. По предположению индукции:

$$S' \vdash w[0:i] A \psi \vdash_1 w[0:i] \alpha' B \beta \psi \ (\alpha' \vdash w[i:k])$$
$$\text{Раскрывая } \alpha', \text{ выведем } w[0:i]w[i:k]B\beta\psi \vdash_1 w[0:k]\gamma\beta\psi \ (\gamma \vdash w[k:j])$$

Почему $\alpha \vdash w[i:j]$? Дело в том, что $\alpha = \alpha' B \vdash \alpha'\gamma \vdash w[i:k]w[k:j] = w[i:j]$.

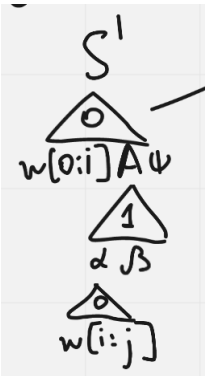
Необходимость доказана индукцией.

$\Longleftarrow$ Докажем достаточность индукцией по нескольким параметрам: j, $l+k$, l. Покажем, за что именно отвечают эти параметры:



Далее поддеревья назовём сверху вниз: верхнее, среднее, нижнее.

База. $j = 0$, $l + k = 0 \Longrightarrow l = 0$. Дерево вывода примет вид:



Из верхнего поддерева получим, что $S' = w[0:i]A\psi$, где $A \in N$, откуда $A = S'$, $w[0:i] = \varepsilon$, $i = 0$. Так как $j = 0$, $\alpha = \varepsilon$. Теперь за A возьмём S', $\alpha = \varepsilon$. Тогда $S' \to S$ — единственное правило грамматики, тогда $\alpha\beta = S$, $\alpha = \varepsilon$, откуда $\beta = S$. Этому соответствует единственная ситуация:

$$(S' \to \cdot S, 0) \in D_0$$

Эта ситуация появляется при инициализации алгоритма Эрли.

Переход. Существует вывод:

$$S' \vdash w[0:i]A\psi$$

$$A \vdash_1 \alpha\beta$$

$$\alpha \vdash w[i:j]$$

Рассмотрим последний символ в α. Разберём три случая:

1. $\alpha = \alpha' a$
2. $\alpha = \alpha' B$
3. $\alpha = \varepsilon$

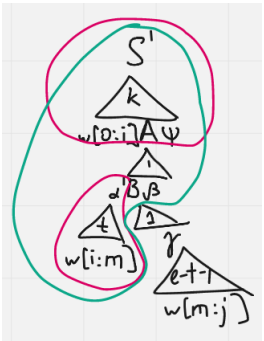
Разберём случай, когда $\alpha = \alpha' a$.

$$\alpha = \alpha' a \vdash_l w[i:j] \Longrightarrow \alpha' \vdash_l w[i:j-1]$$

$$a \vdash w[j] \Longrightarrow w[j] = a$$

$$\begin{cases} S' \vdash_k w[0:j] A\psi \\ A \vdash_1 \alpha' a \beta \\ \alpha' \vdash_l w[i:j-1] \end{cases} \overset{(j-1,k+l,l)}{\Longrightarrow} \begin{cases} (A \to \alpha' \cdot a\beta, i) \in D_{j-1} \\ w[j] = a \end{cases} \overset{Scan}{\Longrightarrow} (A \to \alpha' a \cdot \beta, i) \in D_j$$

Разберём случай, когда $\alpha = \alpha' B$. Тогда $\alpha' B \vdash_l w[i:j]$, откуда существуют такие m, t, что $\alpha' \vdash_t w[i:m]$, $t < l$. Нетерминал B раскрывается по правилу $B \vdash_1 \gamma \vdash_{l-t-1} w[m:j]$. Дерево вывода имеет вид:



Применим предположение индукции:

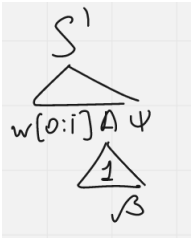
$$\begin{cases} S' \vdash_k w[0:i] A\psi \\ \alpha' \vdash_t w[i:m] \end{cases} \overset{(m,k+t,t)}{\Longrightarrow} (A \to \alpha' \cdot B\beta, i) \in D_m$$

Далее рассмотрим следующее:

$$\begin{cases} S' \vdash_{k+t+1} w[0:m] B\beta\psi \\ \gamma \vdash_{l-t-1} w[m:j] \\ (B \to \gamma) \in P \end{cases}$$

Параметры будут следующими: $(j, k+t+1+l-t-1, l-t-1)$. Сравним это с параметрами $(j, k+l, l)$. Вторые элементы совпадают: $k+t+1+l-t-1 = k+l$, а $l-t-1 < l$. По предположению индукции заключаем, что $(B \to \gamma\cdot, m) \in D_j$. Применим операцию Complete, получим ситуацию $(A \to \alpha' B \cdot \beta, i) \in D_j$.

Разберём случай, когда $\alpha = \varepsilon$. Дерево вывода имеет вид:



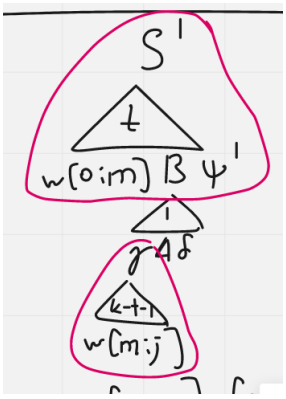
Рассмотрим, как получено A:

$$(B \to \gamma A \delta) \in P$$

$$S' \vdash_t w[0:m] B\psi' \vdash_1 w[0:m]\gamma A\delta\psi' \vdash_{k-t-1} w[0:m]w[m:i]A\delta\psi'$$

$$\gamma \vdash_{k-t-1} w[m:j]$$

Дерево вывода приобрело вид:



Далее используем предположение индукции, параметры: $(j, t+(k-t-1), k-t-1) = (i=j, k-1, k-t-1)$. Сравнение с параметрами $(i=j, k+0, 0)$ позволит сделать следующее:

$$\begin{cases} S' \vdash_t w[0:m] B\psi' \\ \gamma \vdash_{k-t-1} w[m:i] \end{cases} \overset{(i=j, k-1, k-t-1)}{\Longrightarrow} \begin{cases} (B \to \gamma \cdot A\delta, m) \in D_j \\ (A \to \beta) \in P \end{cases} \Longrightarrow (A \to \cdot\beta, j) \in D_j$$

Ситуация $(A \to \cdot\beta, j) \in D_j$ получена по правилу *Predict*. Сравним с тем, что мы хотели: $(A \to \varepsilon \cdot \beta, j) \in D_j$, так как $i = j$.

■

Полнота алгоритма

Утверждение. Алгоритм Эрли является полным.

Доказательство: Рассмотрим слово w. Если слово w выводимо в грамматике $G = \langle N, \Sigma, P, S\rangle$, то верно, что $S \vdash w$. Пусть $i = 0$, $j = |w|$. Тогда существуют $\varphi, \psi \in (N \cup \Sigma)^*$, что $\varphi \vdash w[0:0] = \varepsilon$, $S \vdash w[0:|w|] = w$, что $S' \vdash \varphi S' \psi \vdash S'\psi \vdash_1 S\psi$. Укажем явно φ и ψ: $\varphi = \varepsilon$, $\psi = \varepsilon$, тогда выполняется, что:

$$S' \vdash_1 S \vdash w$$

По сути, только что расписали подробно $S' \vdash w$. A соответствует нетерминальному символу S' (вспомогательному стартовому нетерминалу), α соответствует нетерминальному символу S, из которого выводится слово w, β соответствует пустому слову ε. По основной лемме об инварианте, доказанной ранее, ситуация $(S' \to S\cdot, 0) \in D_{|w|}$ тогда и только тогда, когда $S' \vdash w$. Так как из основной леммы следует корректность алгоритма Эрли, то все возможные ситуации будут рассмотрены, и если слово w выводимо в грамматике G, то это эквивалентно тому, что будет рассмотрена ситуация $(S' \to S\cdot, 0) \in D_{|w|}$, и будет выведено, что $w \in L(G)$. Если слово w не является выводимым в грамматике G, то ситуация $(S' \to S\cdot, 0) \notin D_{|w|}$, и будет выведено, что $w \notin L(G)$. Значит, алгоритм Эрли является полным. ■

Вопрос 20

Эффективное хранение ситуаций и правил

Требуется эффективным образом хранить ситуации типа $(A \to A \cdot X\beta, i) \in D_j$. Множества D_j можно хранить в массиве $D[j][X]$. Возможны три случая, связанные с символом X:

1. $X = a$, $a \in \Sigma$
2. $X = B$, $B \in N$
3. $X = \$$, $\$$ — конец слова

Правила будем хранить в массиве $G[A]$, где в $G[A]$ будут храниться все правила, начинающиеся с A, то есть правила вида $A \to \beta$.

Об операциях

Рассмотрим операцию Scan. Пусть w — слово, $w[j] = a$. Тогда нужно рассмотреть все элементы $D[j][a]$ и расположить их в $D[j+1]$ в соответствии с символом, следующим за a. Асимптотика этой операции

соответствует $\mathcal{O}(D[j][a])$, то есть она растёт в соответствии с количеством элементов, находящихся в $D[j][a]$.

Рассмотрим операцию Predict. Пусть ситуация имеет вид $(A \to \alpha \cdot B\beta, i)$, и эта ситуация принадлежит множеству D_j. Она лежит в $D[j][B]$. Нужно рассмотреть все правила вида $B \to \gamma \in P$, они лежат в $G[B]$.

Рассмотрим операцию Complete. Пусть ситуация имеет вид $(B \to \gamma\cdot, i)$, она принадлежит множеству D_j, лежит в $D[j][\$]$. Нужно рассмотреть все ситуации вида $(A \to \alpha \cdot B\beta, k)$, они принадлежат множеству D_i, они лежат в $D[i][B]$.

Оценки

Оценим, как растёт величина $|D_j|$. $\mathcal{O}(|D_j|) = (j+1)|G|$, где $|G|$ — сумма всех длин правых частей правил. Теперь рассмотрим ситуации $(A \to \alpha \cdot \beta, i)$. Значение i может быть от 0 до j, а точка может быть расположена в $\mathcal{O}(|G|)$ мест.

Асимптотика операции Scan соответствует $\mathcal{O}(|D_j|) = \mathcal{O}(|w| \cdot |G|)$.

Для оценки асимптотики операции Predict рассмотрим множества вида:

$$(A \to \alpha \cdot B\beta, i) \in D_j$$
$$(B \to \cdot\gamma, j) \in D_j$$

Так как операция Predict сводится к перебору по i и точке, то асимптотика операции соответствует $\mathcal{O}(|w| \cdot |G|)$.

Рассмотрим теперь операцию Complete. Ситуации из $D[j][\$]$ указывают на номер множества i и нетерминальный символ B, который соответствует левой части правила из ситуации. Поэтому для каждой ситуации из $D[j][\$]$ будут рассмотрены все ситуации из $D[i][B]$. Поэтому асимптотика операции Complete равна $\mathcal{O}(|P| \cdot |D_0| + |P| \cdot |D_1| + \cdots + |P| \cdot |D_j|) = \mathcal{O}(|P| \cdot |w| \cdot |w| \cdot |G|) = \mathcal{O}\left(|w|^2|G|^2\right)$.

Так как итераций $\mathcal{O}(|w|)$, то итоговая сложность алгоритма составляет $\mathcal{O}\left(|w|^3|G|^2\right)$. Однако данную оценку можно сильно улучшить, если грамматически удастся доказать, что количество появлений каждого правила ограничена сверху некоторой константой C, то асимптотика уже будет равна $\mathcal{O}\left(|w|^2 \cdot C\right)$.