

Математические модели в лингвистике

Математическая теория формальных языков

Мати Пентус, Александр Пиперски, Алексей Сорокин

МГУ им. М. В. Ломоносова, межфакультетский курс,
осенний семестр 2015–2016 учебного года

Мотивация

- При описании фонологических и морфологических процессов часто используются контекстные замены вида $u \rightarrow v \mid x_y$.
- Требуется подобрать математическую модель для их описания.

Мотивация

- При описании фонологических и морфологических процессов часто используются контекстные замены вида $u \rightarrow v \mid x_y$.
- Требуется подобрать математическую модель для их описания.
- То есть хочется задавать условия вида:
 - “Последний прочитанный символ — a ”.

Мотивация

- При описании фонологических и морфологических процессов часто используются контекстные замены вида $u \rightarrow v \mid x_y$.
- Требуется подобрать математическую модель для их описания.
- То есть хочется задавать условия вида:
 - “Последний прочитанный символ — a ”.
 - Символ b находится между символами a и c .

Мотивация

- При описании фонологических и морфологических процессов часто используются контекстные замены вида $u \rightarrow v \mid x_y$.
- Требуется подобрать математическую модель для их описания.
- То есть хочется задавать условия вида:
 - “Последний прочитанный символ — a ”.
 - Символ b находится между символами a и c .
 - В слове содержится ровно 2 символа a , и т.д.

Формальные определения

- Σ — алфавит (произвольное конечное множество), его элементы — буквы.

Формальные определения

- Σ — алфавит (произвольное конечное множество), его элементы — буквы. Например:
 - $\Sigma = \{a\}$, или $\Sigma = \{a, b\}$ или $\Sigma = \{a, b, c\}$;

Формальные определения

- Σ — алфавит (произвольное конечное множество), его элементы — буквы. Например:
 - $\Sigma = \{a\}$, или $\Sigma = \{a, b\}$ или $\Sigma = \{a, b, c\}$;
 - Σ — буквы кириллического или латинского алфавита;

Формальные определения

- Σ — алфавит (произвольное конечное множество), его элементы — буквы. Например:
 - $\Sigma = \{a\}$, или $\Sigma = \{a, b\}$ или $\Sigma = \{a, b, c\}$;
 - Σ — буквы кириллического или латинского алфавита;
 - Σ — элементы международного фонетического алфавита IPA;

Формальные определения

- Σ — алфавит (произвольное конечное множество), его элементы — буквы. Например:
 - $\Sigma = \{a\}$, или $\Sigma = \{a, b\}$ или $\Sigma = \{a, b, c\}$;
 - Σ — буквы кириллического или латинского алфавита;
 - Σ — элементы международного фонетического алфавита IPA;
 - Σ — грамматические категории и т.д.

Формальные определения

- Σ — алфавит (произвольное конечное множество), его элементы — буквы. Например:
 - $\Sigma = \{a\}$, или $\Sigma = \{a, b\}$ или $\Sigma = \{a, b, c\}$;
 - Σ — буквы кириллического или латинского алфавита;
 - Σ — элементы международного фонетического алфавита IPA;
 - Σ — грамматические категории и т.д.
- Слова — конечные последовательности элементов Σ , Σ^* — множество всех слов. ε — пустое слово (слово длины 0).

Формальные определения

- Σ — алфавит (произвольное конечное множество), его элементы — буквы. Например:
 - $\Sigma = \{a\}$, или $\Sigma = \{a, b\}$ или $\Sigma = \{a, b, c\}$;
 - Σ — буквы кириллического или латинского алфавита;
 - Σ — элементы международного фонетического алфавита IPA;
 - Σ — грамматические категории и т.д.
- Слова — конечные последовательности элементов Σ , Σ^* — множество всех слов. ε — пустое слово (слово длины 0).
- Языки — произвольные подмножества Σ^* .

Формальные определения

- Σ — алфавит (произвольное конечное множество), его элементы — буквы. Например:
 - $\Sigma = \{a\}$, или $\Sigma = \{a, b\}$ или $\Sigma = \{a, b, c\}$;
 - Σ — буквы кириллического или латинского алфавита;
 - Σ — элементы международного фонетического алфавита IPA;
 - Σ — грамматические категории и т.д.
- Слова — конечные последовательности элементов Σ , Σ^* — множество всех слов. ε — пустое слово (слово длины 0).
- Языки — произвольные подмножества Σ^* .
- Нам также понадобятся функции из Σ^* в Σ^* и бинарные отношения на множестве Σ^* , то есть подмножества $\Sigma^* \times \Sigma^*$.

Регулярные выражения

Пусть зафиксирован конечный алфавит Σ .

Регулярные выражения

Пусть зафиксирован конечный алфавит Σ .

- Базовые регулярные выражения: элементы алфавита,

Регулярные выражения

Пусть зафиксирован конечный алфавит Σ .

- Базовые регулярные выражения: элементы алфавита,
- Также есть константы 0 (пустой язык) и 1 (язык, содержащий только пустое слово ε).

Регулярные выражения

Пусть зафиксирован конечный алфавит Σ .

- Базовые регулярные выражения: элементы алфавита,
- Также есть константы 0 (пустой язык) и 1 (язык, содержащий только пустое слово ε).
- Бинарные операции: $|$ (объединение) и $\cdot$ (конкатенация или приписывание): $u \cdot v = uv$,

Регулярные выражения

Пусть зафиксирован конечный алфавит Σ .

- Базовые регулярные выражения: элементы алфавита,
- Также есть константы 0 (пустой язык) и 1 (язык, содержащий только пустое слово ε).
- Бинарные операции: $|$ (объединение) и $\cdot$ (конкатенация или приписывание): $u \cdot v = uv$,
- Унарная операция $*$ (итерация, взять любое количество раз): L^* состоит из слов вида $u_1 \dots u_r$, где $r \in \mathbb{N}$, $u_1, \dots, u_r \in L$.

Регулярные выражения

Пусть зафиксирован конечный алфавит Σ .

- Базовые регулярные выражения: элементы алфавита,
- Также есть константы 0 (пустой язык) и 1 (язык, содержащий только пустое слово ε).
- Бинарные операции: $|$ (объединение) и $\cdot$ (конкатенация или приписывание): $u \cdot v = uv$,
- Унарная операция $*$ (итерация, взять любое количество раз): L^* состоит из слов вида $u_1 \dots u_r$, где $r \in \mathbb{N}$, $u_1, \dots, u_r \in L$.
- Если α — регулярное выражение, то $L(\alpha)$ — задаваемый им язык.

Регулярные выражения

Пусть зафиксирован конечный алфавит Σ .

- Базовые регулярные выражения: элементы алфавита,
- Также есть константы 0 (пустой язык) и 1 (язык, содержащий только пустое слово ε).
- Бинарные операции: $|$ (объединение) и $\cdot$ (конкатенация или приписывание): $u \cdot v = uv$,
- Унарная операция $*$ (итерация, взять любое количество раз): L^* состоит из слов вида $u_1 \dots u_r$, где $r \in \mathbb{N}$, $u_1, \dots, u_r \in L$.
- Если α — регулярное выражение, то $L(\alpha)$ — задаваемый им язык.
- Например, $L((a|b)^*) = \{\varepsilon, a, b, aa, ab, ba, bb, \dots\}$.

Регулярные выражения

Пусть зафиксирован конечный алфавит Σ .

- Базовые регулярные выражения: элементы алфавита,
- Также есть константы 0 (пустой язык) и 1 (язык, содержащий только пустое слово ε).
- Бинарные операции: $|$ (объединение) и $\cdot$ (конкатенация или приписывание): $u \cdot v = uv$,
- Унарная операция $*$ (итерация, взять любое количество раз): L^* состоит из слов вида $u_1 \dots u_r$, где $r \in \mathbb{N}$, $u_1, \dots, u_r \in L$.
- Если α — регулярное выражение, то $L(\alpha)$ — задаваемый им язык.
- Например, $L((a|b)^*) = \{\varepsilon, a, b, aa, ab, ba, bb, \dots\}$.
- Приоритет операций: итерация, конкатенация, объединение.
При этом значок конкатенации можно опускать.

Регулярные выражения

Пусть зафиксирован конечный алфавит Σ .

- Базовые регулярные выражения: элементы алфавита,
- Также есть константы 0 (пустой язык) и 1 (язык, содержащий только пустое слово ε).
- Бинарные операции: $|$ (объединение) и $\cdot$ (конкатенация или приписывание): $u \cdot v = uv$,
- Унарная операция $*$ (итерация, взять любое количество раз): L^* состоит из слов вида $u_1 \dots u_r$, где $r \in \mathbb{N}$, $u_1, \dots, u_r \in L$.
- Если α — регулярное выражение, то $L(\alpha)$ — задаваемый им язык.
- Например, $L((a|b)^*) = \{\varepsilon, a, b, aa, ab, ba, bb, \dots\}$.
- Приоритет операций: итерация, конкатенация, объединение.
При этом значок конкатенации можно опускать.

Язык регулярный, если он задаётся регулярным выражением.

Примеры регулярных языков

- Все слова в алфавите $\{a, b\}$: $(a|b)^*$,

Примеры регулярных языков

- Все слова в алфавите $\{a, b\}$: $(a|b)^*$,
- Слова в алфавите $\{a, b, c\}$, где предпоследняя буква — b : $(a|b|c)^*b(a|b|c)$.

Примеры регулярных языков

- Все слова в алфавите $\{a, b\}$: $(a|b)^*$,
- Слова в алфавите $\{a, b, c\}$, где предпоследняя буква — b : $(a|b|c)^*b(a|b|c)$.
- Слова в алфавите $\{a, b, c\}$, содержащие ровно 2 буквы a : $(b|c)^*a(b|c)^*a(b|c)^*$.

Примеры регулярных языков

- Все слова в алфавите $\{a, b\}$: $(a|b)^*$,
- Слова в алфавите $\{a, b, c\}$, где предпоследняя буква — b : $(a|b|c)^*b(a|b|c)$.
- Слова в алфавите $\{a, b, c\}$, содержащие ровно 2 буквы a : $(b|c)^*a(b|c)^*a(b|c)^*$.
- Слова нечётной длины в алфавите $\{a, b\}$: $((a|b)(a|b))^*$.

Примеры регулярных языков

- Все слова в алфавите $\{a, b\}$: $(a|b)^*$,
- Слова в алфавите $\{a, b, c\}$, где предпоследняя буква — b : $(a|b|c)^*b(a|b|c)$.
- Слова в алфавите $\{a, b, c\}$, содержащие ровно 2 буквы a : $(b|c)^*a(b|c)^*a(b|c)^*$.
- Слова нечётной длины в алфавите $\{a, b\}$: $((a|b)(a|b))^*$.
- Слова в алфавите $\{a, b, c\}$, содержащие чётное число букв a : .

Примеры регулярных языков

- Все слова в алфавите $\{a, b\}$: $(a|b)^*$,
- Слова в алфавите $\{a, b, c\}$, где предпоследняя буква — b : $(a|b|c)^*b(a|b|c)$.
- Слова в алфавите $\{a, b, c\}$, содержащие ровно 2 буквы a : $(b|c)^*a(b|c)^*a(b|c)^*$.
- Слова нечётной длины в алфавите $\{a, b\}$: $((a|b)(a|b))^*$.
- Слова в алфавите $\{a, b, c\}$, содержащие чётное число букв a : $((b|c)^*a(b|c)^*a)(b|c)^*$.

Примеры регулярных языков

- Все слова в алфавите $\{a, b\}$: $(a|b)^*$,
- Слова в алфавите $\{a, b, c\}$, где предпоследняя буква — b : $(a|b|c)^*b(a|b|c)$.
- Слова в алфавите $\{a, b, c\}$, содержащие ровно 2 буквы a : $(b|c)^*a(b|c)^*a(b|c)^*$.
- Слова нечётной длины в алфавите $\{a, b\}$: $((a|b)(a|b))^*$.
- Слова в алфавите $\{a, b, c\}$, содержащие чётное число букв a : $((b|c)^*a(b|c)^*a)(b|c)^*$.

Примеры регулярных языков

- Все слова в алфавите $\{a, b\}$: $(a|b)^*$,
- Слова в алфавите $\{a, b, c\}$, где предпоследняя буква — b : $(a|b|c)^*b(a|b|c)$.
- Слова в алфавите $\{a, b, c\}$, содержащие ровно 2 буквы a : $(b|c)^*a(b|c)^*a(b|c)^*$.
- Слова нечётной длины в алфавите $\{a, b\}$: $((a|b)(a|b))^*$.
- Слова в алфавите $\{a, b, c\}$, содержащие чётное число букв a : $((b|c)^*a(b|c)^*a)(b|c)^*$.
- Слова в алфавите $\{a, b, c\}$, где перед a идёт только b :

.

Примеры регулярных языков

- Все слова в алфавите $\{a, b\}$: $(a|b)^*$,
- Слова в алфавите $\{a, b, c\}$, где предпоследняя буква — b : $(a|b|c)^*b(a|b|c)$.
- Слова в алфавите $\{a, b, c\}$, содержащие ровно 2 буквы a : $(b|c)^*a(b|c)^*a(b|c)^*$.
- Слова нечётной длины в алфавите $\{a, b\}$: $((a|b)(a|b))^*$.
- Слова в алфавите $\{a, b, c\}$, содержащие чётное число букв a : $((b|c)^*a(b|c)^*a)(b|c)^*$.
- Слова в алфавите $\{a, b, c\}$, где перед a идёт только b : $((b|c)^*ba)^*(b|c)^*$.

Примеры регулярных языков

- Все слова в алфавите $\{a, b\}$: $(a|b)^*$,
- Слова в алфавите $\{a, b, c\}$, где предпоследняя буква — b : $(a|b|c)^*b(a|b|c)$.
- Слова в алфавите $\{a, b, c\}$, содержащие ровно 2 буквы a : $(b|c)^*a(b|c)^*a(b|c)^*$.
- Слова нечётной длины в алфавите $\{a, b\}$: $((a|b)(a|b))^*$.
- Слова в алфавите $\{a, b, c\}$, содержащие чётное число букв a : $((b|c)^*a(b|c)^*a)(b|c)^*$.
- Слова в алфавите $\{a, b, c\}$, где перед a идёт только b : $((b|c)^*ba)^*(b|c)^*$.

Примеры регулярных языков

- Все слова в алфавите $\{a, b\}$: $(a|b)^*$,
- Слова в алфавите $\{a, b, c\}$, где предпоследняя буква — b : $(a|b|c)^*b(a|b|c)$.
- Слова в алфавите $\{a, b, c\}$, содержащие ровно 2 буквы a : $(b|c)^*a(b|c)^*a(b|c)^*$.
- Слова нечётной длины в алфавите $\{a, b\}$: $((a|b)(a|b))^*$.
- Слова в алфавите $\{a, b, c\}$, содержащие чётное число букв a : $((b|c)^*a(b|c)^*a(b|c)^*)^*$.
- Слова в алфавите $\{a, b, c\}$, где перед a идёт только b : $((b|c)^*ba)^*(b|c)^*$.
- Непустые слова в алфавите $\{a, b\}$, в которых одинаковые буквы не идут подряд: .

Примеры регулярных языков

- Все слова в алфавите $\{a, b\}$: $(a|b)^*$,
- Слова в алфавите $\{a, b, c\}$, где предпоследняя буква — b : $(a|b|c)^*b(a|b|c)$.
- Слова в алфавите $\{a, b, c\}$, содержащие ровно 2 буквы a : $(b|c)^*a(b|c)^*a(b|c)^*$.
- Слова нечётной длины в алфавите $\{a, b\}$: $((a|b)(a|b))^*$.
- Слова в алфавите $\{a, b, c\}$, содержащие чётное число букв a : $((b|c)^*a(b|c)^*a(b|c)^*)^*$.
- Слова в алфавите $\{a, b, c\}$, где перед a идёт только b : $((b|c)^*ba)^*(b|c)^*$.
- Непустые слова в алфавите $\{a, b\}$, в которых одинаковые буквы не идут подряд: $(a(ba)^*(b|1))|(b(ab)^*(a|1))$.

Примеры регулярных языков

- Все слова в алфавите $\{a, b\}$: $(a|b)^*$,
- Слова в алфавите $\{a, b, c\}$, где предпоследняя буква — b : $(a|b|c)^*b(a|b|c)$.
- Слова в алфавите $\{a, b, c\}$, содержащие ровно 2 буквы a : $(b|c)^*a(b|c)^*a(b|c)^*$.
- Слова нечётной длины в алфавите $\{a, b\}$: $((a|b)(a|b))^*$.
- Слова в алфавите $\{a, b, c\}$, содержащие чётное число букв a : $((b|c)^*a(b|c)^*a(b|c)^*)^*$.
- Слова в алфавите $\{a, b, c\}$, где перед a идёт только b : $((b|c)^*ba)^*(b|c)^*$.
- Непустые слова в алфавите $\{a, b\}$, в которых одинаковые буквы не идут подряд: $(a(ba)^*(b|1))|(b(ab)^*(a|1))$.

Примеры регулярных языков

- Все слова в алфавите $\{a, b\}$: $(a|b)^*$,
- Слова в алфавите $\{a, b, c\}$, где предпоследняя буква — b : $(a|b|c)^*b(a|b|c)$.
- Слова в алфавите $\{a, b, c\}$, содержащие ровно 2 буквы a : $(b|c)^*a(b|c)^*a(b|c)^*$.
- Слова нечётной длины в алфавите $\{a, b\}$: $((a|b)(a|b))^*$.
- Слова в алфавите $\{a, b, c\}$, содержащие чётное число букв a : $((b|c)^*a(b|c)^*a)(b|c)^*$.
- Слова в алфавите $\{a, b, c\}$, где перед a идёт только b : $((b|c)^*ba)^*(b|c)^*$.
- Непустые слова в алфавите $\{a, b\}$, в которых одинаковые буквы не идут подряд: $(a(ba)^*(b|1))|(b(ab)^*(a|1))$.
- Непустые слова в алфавите $\{a, b, c\}$, в которых одинаковые буквы не идут подряд:

Примеры регулярных языков

- Все слова в алфавите $\{a, b\}$: $(a|b)^*$,
- Слова в алфавите $\{a, b, c\}$, где предпоследняя буква — b : $(a|b|c)^*b(a|b|c)$.
- Слова в алфавите $\{a, b, c\}$, содержащие ровно 2 буквы a : $(b|c)^*a(b|c)^*a(b|c)^*$.
- Слова нечётной длины в алфавите $\{a, b\}$: $((a|b)(a|b))^*$.
- Слова в алфавите $\{a, b, c\}$, содержащие чётное число букв a : $((b|c)^*a(b|c)^*a(b|c)^*)^*$.
- Слова в алфавите $\{a, b, c\}$, где перед a идёт только b : $((b|c)^*ba)^*(b|c)^*$.
- Непустые слова в алфавите $\{a, b\}$, в которых одинаковые буквы не идут подряд: $(a(ba)^*(b|1))|(b(ab)^*(a|1))$.
- Непустые слова в алфавите $\{a, b, c\}$, в которых одинаковые буквы не идут подряд: $(H|1)(cH)^*(1|c|cH)$, где H — ответ на предыдущий пункт.

Примеры регулярных выражений

Пусть $\Sigma = \{C, V, -\}$ (согласный, гласный, слогораздел).

- Открытый слог с непустой инициальной: .

Примеры регулярных выражений

Пусть $\Sigma = \{C, V, -\}$ (согласный, гласный, слогораздел).

- Открытый слог с непустой инициальной: CC^*V или C^+V .

Примеры регулярных выражений

Пусть $\Sigma = \{C, V, -\}$ (согласный, гласный, слогораздел).

- Открытый слог с непустой инициальнойю: CC^*V или C^+V .
- Слово с 2 гласными, разделёнными хотя бы одним согласным:

Примеры регулярных выражений

Пусть $\Sigma = \{C, V, -\}$ (согласный, гласный, слогораздел).

- Открытый слог с непустой инициалью: CC^*V или C^+V .
- Слово с 2 гласными, разделёнными хотя бы одним согласным: $C^*VC^+VC^*$.

Примеры регулярных выражений

Пусть $\Sigma = \{C, V, -\}$ (согласный, гласный, слогораздел).

- Открытый слог с непустой инициалью: CC^*V или C^+V .
- Слово с 2 гласными, разделёнными хотя бы одним согласным: $C^*VC^+VC^*$.
- Корректное разбиение на слоги:

Примеры регулярных выражений

Пусть $\Sigma = \{C, V, -\}$ (согласный, гласный, слогораздел).

- Открытый слог с непустой инициалью: CC^*V или C^+V .
- Слово с 2 гласными, разделёнными хотя бы одним согласным: $C^*VC^+VC^*$.
- Корректное разбиение на слоги: $1|(C^*VC^*-)^*(C^*VC^*)$

Примеры регулярных выражений

Пусть $\Sigma = \{C, V, -\}$ (согласный, гласный, слогораздел).

- Открытый слог с непустой инициалью: CC^*V или C^+V .
- Слово с 2 гласными, разделёнными хотя бы одним согласным: $C^*VC^+VC^*$.
- Корректное разбиение на слоги: $1|(C^*VC^*-)^*(C^*VC^*)$
- Разбиение слова на слоги, содержащее ровно 1 открытый слог.

Примеры регулярных выражений

Пусть $\Sigma = \{C, V, -\}$ (согласный, гласный, слогораздел).

- Открытый слог с непустой инициалью: CC^*V или C^+V .
- Слово с 2 гласными, разделёнными хотя бы одним согласным: $C^*VC^+VC^*$.
- Корректное разбиение на слоги: $1|(C^*VC^*-)^*(C^*VC^*)$
- Разбиение слова на слоги, содержащее ровно 1 открытый слог. $(C^*VC^+-)^*(C^*V)(-C^*VC^+)^*$

Примеры регулярных выражений

Пусть $\Sigma = \{C, V, -\}$ (согласный, гласный, слогораздел).

- Открытый слог с непустой инициалью: CC^*V или C^+V .
- Слово с 2 гласными, разделёнными хотя бы одним согласным: $C^*VC^+VC^*$.
- Корректное разбиение на слоги: $1|(C^*VC^*-)^*(C^*VC^*)$
- Разбиение слова на слоги, содержащее ровно 1 открытый слог. $(C^*VC^+-)^*(C^*V)(-C^*VC^+)^*$
- “Гармония гласных” (встречаются либо только гласные типа V_1 , либо только типа V_2): $C^*(V_1(C|V_1)^*|V_2(C|V_2)^*)$

Определение конечного автомата

Пусть Σ — конечный алфавит.

Определение конечного автомата

Конечный автомат: помеченный граф $M = \langle Q, \Sigma, \Delta, q_0, F \rangle$, где

Определение конечного автомата

Пусть Σ — конечный алфавит.

Определение конечного автомата

Конечный автомат: помеченный граф $M = \langle Q, \Sigma, \Delta, q_0, F \rangle$, где

- Q — конечное множество состояний (вершин графа),

Определение конечного автомата

Пусть Σ — конечный алфавит.

Определение конечного автомата

Конечный автомат: помеченный граф $M = \langle Q, \Sigma, \Delta, q_0, F \rangle$, где

- Q — конечное множество состояний (вершин графа),
- Δ — конечное множество переходов (рёбер с метками) вида $\langle q_1, w \rangle \rightarrow q_2, q_1, q_2 \in Q, w \in \Sigma^*$.

Определение конечного автомата

Пусть Σ — конечный алфавит.

Определение конечного автомата

Конечный автомат: помеченный граф $M = \langle Q, \Sigma, \Delta, q_0, F \rangle$, где

- Q — конечное множество состояний (вершин графа),
- Δ — конечное множество переходов (рёбер с метками) вида $\langle q_1, w \rangle \rightarrow q_2, q_1, q_2 \in Q, w \in \Sigma^*$.
- $q_0 \in Q$ — стартовое состояние
- $F \subseteq Q$ — завершающие состояния.

Определение конечного автомата

Пусть Σ — конечный алфавит.

Определение конечного автомата

Конечный автомат: помеченный граф $M = \langle Q, \Sigma, \Delta, q_0, F \rangle$, где

- Q — конечное множество состояний (вершин графа),
- Δ — конечное множество переходов (рёбер с метками) вида $\langle q_1, w \rangle \rightarrow q_2, q_1, q_2 \in Q, w \in \Sigma^*$.
- $q_0 \in Q$ — стартовое состояние
- $F \subseteq Q$ — завершающие состояния.

Метка пути — конкатенация всех меток на входящих в него рёбрах.

Определение конечного автомата

Пусть Σ — конечный алфавит.

Определение конечного автомата

Конечный автомат: помеченный граф $M = \langle Q, \Sigma, \Delta, q_0, F \rangle$, где

- Q — конечное множество состояний (вершин графа),
- Δ — конечное множество переходов (рёбра с метками) вида $\langle q_1, w \rangle \rightarrow q_2, q_1, q_2 \in Q, w \in \Sigma^*$.
- $q_0 \in Q$ — стартовое состояние
- $F \subseteq Q$ — завершающие состояния.

Метка пути — конкатенация всех меток на входящих в него рёбрах.

Язык $L(M)$ — метки путей из начального состояния в завершающие.

Определение конечного автомата

Пусть Σ — конечный алфавит.

Определение конечного автомата

Конечный автомат: помеченный граф $M = \langle Q, \Sigma, \Delta, q_0, F \rangle$, где

- Q — конечное множество состояний (вершин графа),
- Δ — конечное множество переходов (рёбер с метками) вида $\langle q_1, w \rangle \rightarrow q_2, q_1, q_2 \in Q, w \in \Sigma^*$.
- $q_0 \in Q$ — стартовое состояние
- $F \subseteq Q$ — завершающие состояния.

Метка пути — конкатенация всех меток на входящих в него рёбрах.

Язык $L(M)$ — метки путей из начального состояния в завершающие.

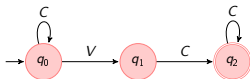
Язык — автоматный, если задаётся некоторым конечным автоматом.

Примеры конечных автоматов

- Закрытый слог

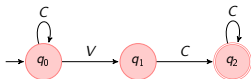
Примеры конечных автоматов

- Закрытый слог



Примеры конечных автоматов

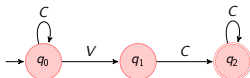
- Закрытый слог



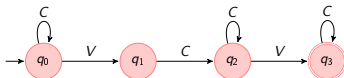
- Слово с 2 гласными, разделёнными хотя бы одним согласным:

Примеры конечных автоматов

- Закрытый слог

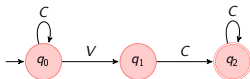


- Слово с 2 гласными, разделёнными хотя бы одним согласным:

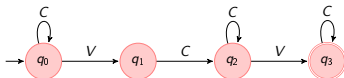


Примеры конечных автоматов

- Закрытый слог



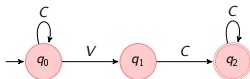
- Слово с 2 гласными, разделёнными хотя бы одним согласным:



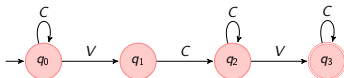
- Слогоделение ровно с одним открытым слогом:

Примеры конечных автоматов

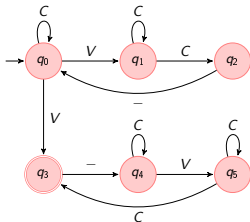
- Закрытый слог



- Слово с 2 гласными, разделёнными хотя бы одним согласным:



- Слогоделение ровно с одним открытым слогом:

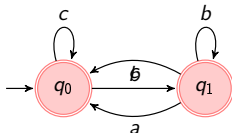


Примеры конечных автоматов

- Слова в алфавите $\{a, b, c\}$, где a встречается только после b :

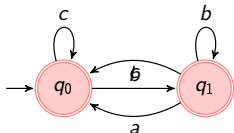
Примеры конечных автоматов

- Слова в алфавите $\{a, b, c\}$, где a встречается только после b :



Примеры конечных автоматов

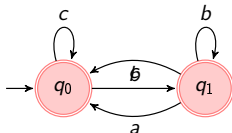
- Слова в алфавите $\{a, b, c\}$, где a встречается только после b :



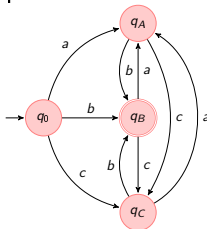
- Слова в алфавите $\{a, b, c\}$, где нет повторяющихся букв, причём слово завершается буквой b :

Примеры конечных автоматов

- Слова в алфавите $\{a, b, c\}$, где a встречается только после b :



- Слова в алфавите $\{a, b, c\}$, где нет повторяющихся букв, причём слово завершается буквой b :



Свойства автоматных языков

Теорема

*Любой автоматный язык распознаётся автоматом с
однобуквенными переходами.*

Свойства автоматных языков

Теорема

Любой автоматный язык распознаётся автоматом с однобуквенными переходами.

Определение

Автомат с однобуквенными переходами — детерминированный, если ни из какого состояния нет двух переходов по одной и той же букве.

Теорема

Любой автоматный язык распознаётся детерминированным конечным автоматом.

Свойства автоматных языков

Теорема

Любой автоматный язык распознаётся автоматом с однобуквенными переходами.

Определение

Автомат с однобуквенными переходами — детерминированный, если ни из какого состояния нет двух переходов по одной и той же букве.

Теорема

Любой автоматный язык распознаётся детерминированным конечным автоматом.

Теорема (Клини, 1956)

Классы автоматных и регулярных языков совпадают.

Конечные преобразователи

Неформально, конечный преобразователь — это автомат с добавленными на рёбра выходными символами.

Конечные преобразователи

Неформально, конечный преобразователь — это автомат с добавленными на рёбра выходными символами.

Пусть Σ , Γ — конечные алфавиты.

Определение конечного преобразователя

Конечный преобразователь: кортеж $M = \langle Q, \Sigma, \Gamma, \Delta, q_0, F \rangle$, где

- Q — конечное множество состояний (вершин графа)

Конечные преобразователи

Неформально, конечный преобразователь — это автомат с добавленными на рёбра выходными символами.

Пусть Σ , Γ — конечные алфавиты.

Определение конечного преобразователя

Конечный преобразователь: кортеж $M = \langle Q, \Sigma, \Gamma, \Delta, q_0, F \rangle$, где

- Q — конечное множество состояний (вершин графа)
- Δ — конечное множество переходов вида $\langle q_1, u \rangle \rightarrow \langle q_2, v \rangle$, $q_1, q_2 \in Q$, $u \in \Sigma^*$, $v \in \Gamma^*$.

Конечные преобразователи

Неформально, конечный преобразователь — это автомат с добавленными на рёбра выходными символами.

Пусть Σ , Γ — конечные алфавиты.

Определение конечного преобразователя

Конечный преобразователь: кортеж $M = \langle Q, \Sigma, \Gamma, \Delta, q_0, F \rangle$, где

- Q — конечное множество состояний (вершин графа)
- Δ — конечное множество переходов вида $\langle q_1, u \rangle \rightarrow \langle q_2, v \rangle$, $q_1, q_2 \in Q$, $u \in \Sigma^*$, $v \in \Gamma^*$.
- $q_0 \in Q$ — стартовое состояние
- $F \subseteq Q$ — завершающие состояния.

Конечные преобразователи

Неформально, конечный преобразователь — это автомат с добавленными на рёбра выходными символами.

Пусть Σ , Γ — конечные алфавиты.

Определение конечного преобразователя

Конечный преобразователь: кортеж $M = \langle Q, \Sigma, \Gamma, \Delta, q_0, F \rangle$, где

- Q — конечное множество состояний (вершин графа)
- Δ — конечное множество переходов вида $\langle q_1, u \rangle \rightarrow \langle q_2, v \rangle$, $q_1, q_2 \in Q$, $u \in \Sigma^*$, $v \in \Gamma^*$.
- $q_0 \in Q$ — стартовое состояние
- $F \subseteq Q$ — завершающие состояния.

В дальнейшем мы будем считать, что все переходы имеют вид $\langle q_1, a \rangle \rightarrow \langle q_2, b \rangle$, $a \in \Sigma \cup \{\varepsilon\}$, $b \in \Gamma \cup \{\varepsilon\}$. То есть на каждом шаге читается и пишется не более одного символа.

Конечные преобразователи

Конечный преобразователь принимает пары слов $\langle u, v \rangle \in \Sigma^* \times \Gamma^*$, которые являются метками путей из начального состояния в завершающие. Также можно считать, что конечный преобразователь задаёт многозначное отображение из Σ^* в Γ^* (возможно, не всюду определённое).

Конечные преобразователи

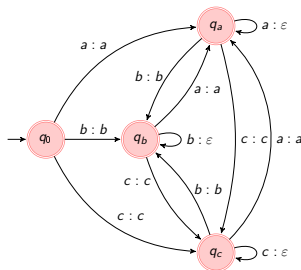
Конечный преобразователь принимает пары слов $\langle u, v \rangle \in \Sigma^* \times \Gamma^*$, которые являются метками путей из начального состояния в завершающие. Также можно считать, что конечный преобразователь задаёт многозначное отображение из Σ^* в Γ^* (возможно, не всюду определённое).

Пусть $\Sigma = \{a, b, c\}$, следующий преобразователь удаляет в слове повторяющиеся буквы, например, отображает *acccbbaac* в *acbac*.

Конечные преобразователи

Конечный преобразователь принимает пары слов $\langle u, v \rangle \in \Sigma^* \times \Gamma^*$, которые являются метками путей из начального состояния в завершающие. Также можно считать, что конечный преобразователь задаёт многозначное отображение из Σ^* в Γ^* (возможно, не всюду определённое).

Пусть $\Sigma = \{a, b, c\}$, следующий преобразователь удаляет в слове повторяющиеся буквы, например, отображает *acccbbaac* в *acbac*.



Контекстная замена

Контекстная замена

$X \rightarrow Y \parallel U_V$ (заменить X на Y между U и V)

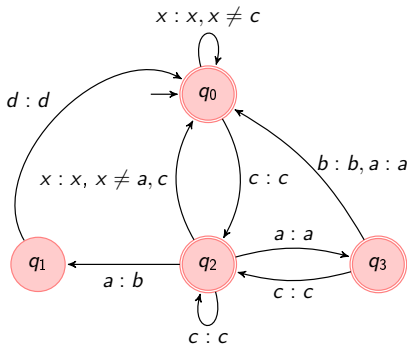
Контекстная замена

Контекстная замена

$X \rightarrow Y || U_V$ (заменить X на Y между U и V)

Конечный преобразователь для контекстной замены

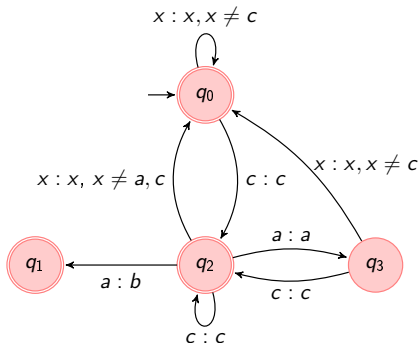
$a \rightarrow b || c_d$ (считаем, что $\Sigma = \Gamma = \{a, b, c, d\}$).



Контекстная замена

Контекстная замена

Конечный преобразователь для контекстной замены
 $a \rightarrow b \parallel c_ \#$ (то есть замена производится только в конце слова после символа c).



Свойства конечных автоматов

- Замкнутость относительно операций:

Операция	Автоматы	Преобразователи
Конкатенация	Да	Да
Итерация	Да	Да
Объединение	Да	Да
Пересечение	Да	Нет
Дополнение	Да	Нет

Свойства конечных преобразователей

- Конечные преобразования также замкнуты относительно
 - Композиции ($\circ$).
 - Приоритетного объединения ($\cup_p$).

Свойства конечных преобразователей

- Конечные преобразования также замкнуты относительно
 - Композиции ($\circ$).
 - Приоритетного объединения ($\cup_p$).

$$(T_1 \cup_p T_2)(u) = \begin{cases} T_1(u), & T_1(u) \neq \emptyset, \\ T_2(u), & T_1(u) = \emptyset. \end{cases}$$

Свойства конечных преобразователей

- Конечные преобразования также замкнуты относительно
 - Композиции ($\circ$).
 - Приоритетного объединения ($\cup_p$).

$$(T_1 \cup_p T_2)(u) = \begin{cases} T_1(u), & T_1(u) \neq \emptyset, \\ T_2(u), & T_1(u) = \emptyset. \end{cases}$$

- Обращения $((\cdot)^{-1})$: $\phi^{-1} = \{\langle y, x \rangle \mid \langle x, y \rangle \in \phi\}$.

Свойства конечных преобразователей

- Конечные преобразования также замкнуты относительно
 - Композиции ($\circ$).
 - Приоритетного объединения ($\cup_p$).

$$(T_1 \cup_p T_2)(u) = \begin{cases} T_1(u), & T_1(u) \neq \emptyset, \\ T_2(u), & T_1(u) = \emptyset. \end{cases}$$

- Обращения $((\cdot)^{-1})$: $\phi^{-1} = \{\langle y, x \rangle \mid \langle x, y \rangle \in \phi\}$.
- Применения операций:
 - Композиция: последовательное применение преобразований,

Свойства конечных преобразователей

- Конечные преобразования также замкнуты относительно
 - Композиции ($\circ$).
 - Приоритетного объединения ($\cup_p$).

$$(T_1 \cup_p T_2)(u) = \begin{cases} T_1(u), & T_1(u) \neq \emptyset, \\ T_2(u), & T_1(u) = \emptyset. \end{cases}$$

- Обращения $((\cdot)^{-1})$: $\phi^{-1} = \{\langle y, x \rangle \mid \langle x, y \rangle \in \phi\}$.
- Применения операций:
 - Композиция: последовательное применение преобразований,
 - Обращение: переход от синтеза к анализу и наоборот,

Свойства конечных преобразователей

- Конечные преобразования также замкнуты относительно
 - Композиции ($\circ$).
 - Приоритетного объединения ($\cup_p$).

$$(T_1 \cup_p T_2)(u) = \begin{cases} T_1(u), & T_1(u) \neq \emptyset, \\ T_2(u), & T_1(u) = \emptyset. \end{cases}$$

- Обращения $((\cdot)^{-1})$: $\phi^{-1} = \{\langle y, x \rangle \mid \langle x, y \rangle \in \phi\}$.
- Применения операций:
 - Композиция: последовательное применение преобразований,
 - Обращение: переход от синтеза к анализу и наоборот,
 - Приоритетное объединение: отдельная обработка нерегулярных форм.

Модельная задача: изменение по числу в английском

“Модельная задача”

Морфологический синтез: множественное число существительного в английском (+N, +Sg, +Pl — вспомогательные символы):

“Модельная задача”

Морфологический синтез: множественное число существительного в английском (+N, +Sg, +Pl — вспомогательные символы):

- $cat+N+Sg \Leftrightarrow cat$
- $cat+N+Pl \Leftrightarrow cats$
- $torch+N+Pl \Leftrightarrow torches$
- $monarch+N+Pl \Leftrightarrow monarchs$
- $ally+N+Pl \Leftrightarrow allies$
- $play+N+Pl \Leftrightarrow plays$
- $goose+N+Pl \Leftrightarrow geese$
- $formula+N+Pl \Leftrightarrow formulas/formulae$

Модельная задача: изменение по числу в английском

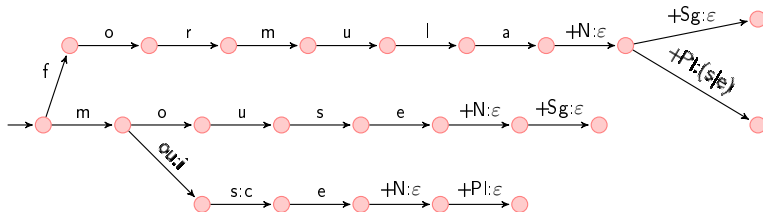
Промежуточные преобразования

- A_{in} — проверка корректности входа $\Sigma^+(+N)(+Sg|+Pl)$

Модельная задача: изменение по числу в английском

Промежуточные преобразования

- A_{in} — проверка корректности входа $\Sigma^+ (+N)(+Sg|+Pl)$
- T_{exc} — преобразователь для исключений $formula+N+Pl \mapsto formulas/formulae$, $goose+N+Pl \mapsto geese$, $mouse+N+Pl \mapsto mice$ и т.д. (фрагмент):



Промежуточные преобразования

- A_{in} — проверка корректности входа $\Sigma^+(+N)(+Sg|+Pl)$
- T_{exc} — преобразователь для исключений $formula+N+Pl \mapsto formulas/formulae$, $goose+N+Pl \mapsto geese$, $mouse+N+Pl \mapsto mice$ и т.д. (фрагмент):
- T_s : замены $+N+Sg \rightarrow \hat{_}||_ \#$, $+N+Pl \rightarrow \hat{s}||_ \#$ ($\#$ — конец строки, $\hat{_}$ — вспомогательный символ)

Промежуточные преобразования

- A_{in} — проверка корректности входа $\Sigma^+(+N)(+Sg|+Pl)$
- T_{exc} — преобразователь для исключений $formula+N+Pl \mapsto formulas/formulae$, $goose+N+Pl \mapsto geese$, $mouse+N+Pl \mapsto mice$ и т.д. (фрагмент):
- T_s : замены $+N+Sg \rightarrow \hat{_}||_ \#$, $+N+Pl \rightarrow \hat{s}||_ \#$ ($\#$ — конец строки, $\hat{_}$ — вспомогательный символ)
- T_y : замена $y \rightarrow ie||C_ \hat{s}\#$ (C — выражение, задающее все согласные.)

Промежуточные преобразования

- A_{in} — проверка корректности входа $\Sigma^+(+N)(+Sg|+Pl)$
- T_{exc} — преобразователь для исключений $formula+N+Pl \mapsto formulas/formulae$, $goose+N+Pl \mapsto geese$, $mouse+N+Pl \mapsto mice$ и т.д. (фрагмент):
- T_s : замены $+N+Sg \rightarrow \wedge || _ \#$, $+N+Pl \rightarrow \wedge s || _ \#$ ($\#$ — конец строки, $\wedge$ — вспомогательный символ)
- T_y : замена $y \rightarrow ie || C _ \wedge s \#$ (C — выражение, задающее все согласные.)
- $A_{exc,sib}$: проверка условия $\Sigma^+ arch \wedge s \#$

Промежуточные преобразования

- A_{in} — проверка корректности входа $\Sigma^+(+N)(+Sg|+Pl)$
- T_{exc} — преобразователь для исключений $formula+N+Pl \mapsto formulas/formulae$, $goose+N+Pl \mapsto geese$, $mouse+N+Pl \mapsto mice$ и т.д. (фрагмент):
- T_s : замены $+N+Sg \rightarrow \hat{_}||_ \#$, $+N+Pl \rightarrow \hat{s}||_ \#$ ($\#$ — конец строки, $\hat{_}$ — вспомогательный символ)
- T_y : замена $y \rightarrow ie||C_ \hat{s}\#$ (C — выражение, задающее все согласные.)
- $A_{exc,sib}$: проверка условия $\Sigma^+arch\hat{s}\#$
- T_{sib} : замена $\rightarrow e||[s|z|x|sh|ch]_ \hat{s}\#$

Промежуточные преобразования

- A_{in} — проверка корректности входа $\Sigma^+(+N)(+Sg|+Pl)$
- T_{exc} — преобразователь для исключений $formula+N+Pl \mapsto formulas/formulae$, $goose+N+Pl \mapsto geese$, $mouse+N+Pl \mapsto mice$ и т.д. (фрагмент):
- T_s : замены $+N+Sg \rightarrow \hat{_}||_ \#$, $+N+Pl \rightarrow \hat{s}||_ \#$ ($\#$ — конец строки, $\hat{_}$ — вспомогательный символ)
- T_y : замена $y \rightarrow ie||C_ \hat{s}\#$ (C — выражение, задающее все согласные.)
- $A_{exc,sib}$: проверка условия $\Sigma^+arch\hat{s}\#$
- T_{sib} : замена $\rightarrow e||[s|z|x|sh|ch]_ \hat{s}\#$
- T_{cl} : очистка $\hat{_} \rightarrow ||_$

Промежуточные преобразования

- A_{in} — проверка корректности входа $\Sigma^+(+N)(+Sg|+Pl)$
- T_{exc} — преобразователь для исключений $formula+N+Pl \mapsto formulas/formulae$, $goose+N+Pl \mapsto geese$, $mouse+N+Pl \mapsto mice$ и т.д. (фрагмент):
- T_s : замены $+N+Sg \rightarrow \hat{_}||_ \#$, $+N+Pl \rightarrow \hat{s}||_ \#$ ($\#$ — конец строки, $\hat{_}$ — вспомогательный символ)
- T_y : замена $y \rightarrow ie||C_ \hat{s}\#$ (C — выражение, задающее все согласные.)
- $A_{exc,sib}$: проверка условия $\Sigma^+ arch \hat{s}\#$
- T_{sib} : замена $\rightarrow e||[s|z|x|sh|ch]_ \hat{s}\#$
- T_{cl} : очистка $\hat{_} \rightarrow ||_$
- Общий преобразователь: $A_{in} \circ (T_{exc} \cup_p (T_s \circ T_y \circ (A_{exc,sib} \cup_p T_{sib})) \circ T_{cl}))$

Примеры вычисления

- Некорректный вход: $cat+N$, $dog+N+Pg$ — отвергается A_{in} ,

Примеры вычисления

- Некорректный вход: $cat+N$, $dog+N+Pg$ — отвергается A_{in} ,
- Исключения: $mouse+N+Pl \mapsto mice$, $cactus+N+Pl \mapsto cactuses$, $cacti$ — обрабатывается в T_{exc} и не идёт в основную ветку.

Примеры вычисления

- Некорректный вход: $cat+N$, $dog+N+Pg$ — отвергается A_{in} ,
- Исключения: $mouse+N+Pl \mapsto mice$, $cactus+N+Pl \mapsto cactuses$, $cacti$ — обрабатывается в T_{exc} и не идёт в основную ветку.
- Регулярные формы обрабатываются в ветке $A_{in} \circ T_s \circ T_y \circ T_{sib} \circ T_{cl}$:

Примеры вычисления

- Некорректный вход: $cat+N$, $dog+N+Pg$ — отвергается A_{in} ,
- Исключения: $mouse+N+Pl \mapsto mice$, $cactus+N+Pl \mapsto cactuses$, $cacti$ — обрабатывается в T_{exc} и не идёт в основную ветку.
- Регулярные формы обрабатываются в ветке $A_{in} \circ T_s \circ T_y \circ T_{sib} \circ T_{cl}$:
 - $day+N+Pl$

Примеры вычисления

- Некорректный вход: $cat+N$, $dog+N+Pg$ — отвергается A_{in} ,
- Исключения: $mouse+N+Pl \mapsto mice$, $cactus+N+Pl \mapsto cactuses$, $cacti$ — обрабатывается в T_{exc} и не идёт в основную ветку.
- Регулярные формы обрабатываются в ветке $A_{in} \circ T_s \circ T_y \circ T_{sib} \circ T_{cl}$:
 - $day+N+Pl \rightarrow day\hat{s}$

Примеры вычисления

- Некорректный вход: $cat+N$, $dog+N+Pg$ — отвергается A_{in} ,
- Исключения: $mouse+N+Pl \mapsto mice$, $cactus+N+Pl \mapsto cactuses$, $cacti$ — обрабатывается в T_{exc} и не идёт в основную ветку.
- Регулярные формы обрабатываются в ветке $A_{in} \circ T_s \circ T_y \circ T_{sib} \circ T_{cl}$:
 - $day+N+Pl \rightarrow day\hat{s} \rightarrow day\hat{s}$

Примеры вычисления

- Некорректный вход: $cat+N$, $dog+N+Pg$ — отвергается A_{in} ,
- Исключения: $mouse+N+Pl \mapsto mice$, $cactus+N+Pl \mapsto cactuses$, $cacti$ — обрабатывается в T_{exc} и не идёт в основную ветку.
- Регулярные формы обрабатываются в ветке $A_{in} \circ T_s \circ T_y \circ T_{sib} \circ T_{cl}$:
 - $day+N+Pl \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow day\hat{s}$

Примеры вычисления

- Некорректный вход: $cat+N$, $dog+N+Pg$ — отвергается A_{in} ,
- Исключения: $mouse+N+Pl \mapsto mice$, $cactus+N+Pl \mapsto cactuses$, $cacti$ — обрабатывается в T_{exc} и не идёт в основную ветку.
- Регулярные формы обрабатываются в ветке $A_{in} \circ T_s \circ T_y \circ T_{sib} \circ T_{cl}$:
 - $day+N+Pl \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow days$

Примеры вычисления

- Некорректный вход: $cat+N$, $dog+N+Pg$ — отвергается A_{in} ,
- Исключения: $mouse+N+Pl \mapsto mice$, $cactus+N+Pl \mapsto cactuses$, $cacti$ — обрабатывается в T_{exc} и не идёт в основную ветку.
- Регулярные формы обрабатываются в ветке $A_{in} \circ T_s \circ T_y \circ T_{sib} \circ T_{cl}$:
 - $day+N+Pl \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow days$
 - $rally+N+Pl$

Примеры вычисления

- Некорректный вход: $cat+N$, $dog+N+Pg$ — отвергается A_{in} ,
- Исключения: $mouse+N+Pl \mapsto mice$, $cactus+N+Pl \mapsto cactuses$, $cacti$ — обрабатывается в T_{exc} и не идёт в основную ветку.
- Регулярные формы обрабатываются в ветке $A_{in} \circ T_s \circ T_y \circ T_{sib} \circ T_{cl}$:
 - $day+N+Pl \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow days$
 - $rally+N+Pl \rightarrow rally\hat{s}$

Примеры вычисления

- Некорректный вход: $cat+N$, $dog+N+Pg$ — отвергается A_{in} ,
- Исключения: $mouse+N+Pl \mapsto mice$, $cactus+N+Pl \mapsto cactuses$, $cacti$ — обрабатывается в T_{exc} и не идёт в основную ветку.
- Регулярные формы обрабатываются в ветке $A_{in} \circ T_s \circ T_y \circ T_{sib} \circ T_{cl}$:
 - $day+N+Pl \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow days$
 - $rally+N+Pl \rightarrow rally\hat{s} \rightarrow ralli\hat{e}s$

Примеры вычисления

- Некорректный вход: $cat+N$, $dog+N+Pg$ — отвергается A_{in} ,
- Исключения: $mouse+N+Pl \mapsto mice$, $cactus+N+Pl \mapsto cactuses$, $cacti$ — обрабатывается в T_{exc} и не идёт в основную ветку.
- Регулярные формы обрабатываются в ветке $A_{in} \circ T_s \circ T_y \circ T_{sib} \circ T_{cl}$:
 - $day+N+Pl \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow days$
 - $rally+N+Pl \rightarrow rally\hat{s} \rightarrow ralli\hat{e}s \rightarrow ralli\hat{e}s$

Примеры вычисления

- Некорректный вход: $cat+N$, $dog+N+Pg$ — отвергается A_{in} ,
- Исключения: $mouse+N+Pl \mapsto mice$, $cactus+N+Pl \mapsto cactuses$, $cacti$ — обрабатывается в T_{exc} и не идёт в основную ветку.
- Регулярные формы обрабатываются в ветке $A_{in} \circ T_s \circ T_y \circ T_{sib} \circ T_{cl}$:
 - $day+N+Pl \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow days$
 - $rally+N+Pl \rightarrow rally\hat{s} \rightarrow ralli\hat{e}s \rightarrow ralli\hat{e}s \rightarrow rallies$

Примеры вычисления

- Некорректный вход: $cat+N$, $dog+N+Pg$ — отвергается A_{in} ,
- Исключения: $mouse+N+Pl \mapsto mice$, $cactus+N+Pl \mapsto cactuses$, $cacti$ — обрабатывается в T_{exc} и не идёт в основную ветку.
- Регулярные формы обрабатываются в ветке $A_{in} \circ T_s \circ T_y \circ T_{sib} \circ T_{cl}$:
 - $day+N+Pl \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow days$
 - $rally+N+Pl \rightarrow rally\hat{s} \rightarrow ralli\hat{e}s \rightarrow ralli\hat{e}s \rightarrow rallies$
 - $witch+N+Pl$

Примеры вычисления

- Некорректный вход: $cat+N$, $dog+N+Pg$ — отвергается A_{in} ,
- Исключения: $mouse+N+Pl \mapsto mice$, $cactus+N+Pl \mapsto cactuses$, $cacti$ — обрабатывается в T_{exc} и не идёт в основную ветку.
- Регулярные формы обрабатываются в ветке $A_{in} \circ T_s \circ T_y \circ T_{sib} \circ T_{cl}$:
 - $day+N+Pl \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow days$
 - $rally+N+Pl \rightarrow rally\hat{s} \rightarrow ralli\hat{e}s \rightarrow ralli\hat{e}s \rightarrow rallies$
 - $witch+N+Pl \rightarrow witch\hat{s}$

Модельная задача: изменение по числу в английском

Примеры вычисления

- Некорректный вход: $cat+N$, $dog+N+Pg$ — отвергается A_{in} ,
- Исключения: $mouse+N+Pl \mapsto mice$, $cactus+N+Pl \mapsto cactuses$, $cacti$ — обрабатывается в T_{exc} и не идёт в основную ветку.
- Регулярные формы обрабатываются в ветке $A_{in} \circ T_s \circ T_y \circ T_{sib} \circ T_{cl}$:
 - $day+N+Pl \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow days$
 - $rally+N+Pl \rightarrow rally\hat{s} \rightarrow ralli\hat{e}s \rightarrow ralli\hat{e}s \rightarrow rallies$
 - $witch+N+Pl \rightarrow witch\hat{s} \rightarrow witch\hat{s}$

Модельная задача: изменение по числу в английском

Примеры вычисления

- Некорректный вход: $cat+N$, $dog+N+Pg$ — отвергается A_{in} ,
- Исключения: $mouse+N+Pl \mapsto mice$, $cactus+N+Pl \mapsto cactuses$, $cacti$ — обрабатывается в T_{exc} и не идёт в основную ветку.
- Регулярные формы обрабатываются в ветке $A_{in} \circ T_s \circ T_y \circ T_{sib} \circ T_{cl}$:
 - $day+N+Pl \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow days$
 - $rally+N+Pl \rightarrow rally\hat{s} \rightarrow ralli\hat{e}s \rightarrow ralli\hat{e}s \rightarrow rallies$
 - $witch+N+Pl \rightarrow witch\hat{s} \rightarrow witch\hat{s} \rightarrow witch\hat{e}s$

Модельная задача: изменение по числу в английском

Примеры вычисления

- Некорректный вход: $cat+N$, $dog+N+Pg$ — отвергается A_{in} ,
- Исключения: $mouse+N+Pl \mapsto mice$, $cactus+N+Pl \mapsto cactuses$, $cacti$ — обрабатывается в T_{exc} и не идёт в основную ветку.
- Регулярные формы обрабатываются в ветке $A_{in} \circ T_s \circ T_y \circ T_{sib} \circ T_{cl}$:
 - $day+N+Pl \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow days$
 - $rally+N+Pl \rightarrow rally\hat{s} \rightarrow ralli\hat{e}s \rightarrow ralli\hat{e}s \rightarrow rallies$
 - $witch+N+Pl \rightarrow witch\hat{s} \rightarrow witch\hat{s} \rightarrow witch\hat{e}s \rightarrow witches$

Примеры вычисления

- Некорректный вход: $cat+N$, $dog+N+Pg$ — отвергается A_{in} ,
- Исключения: $mouse+N+Pl \mapsto mice$, $cactus+N+Pl \mapsto cactuses$, $cacti$ — обрабатывается в T_{exc} и не идёт в основную ветку.
- Регулярные формы обрабатываются в ветке $A_{in} \circ T_s \circ T_y \circ T_{sib} \circ T_{cl}$:
 - $day+N+Pl \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow day\hat{s} \rightarrow days$
 - $rally+N+Pl \rightarrow rally\hat{s} \rightarrow ralli\hat{e}s \rightarrow ralli\hat{e}s \rightarrow rallies$
 - $witch+N+Pl \rightarrow witch\hat{s} \rightarrow witch\hat{s} \rightarrow witch\hat{e}s \rightarrow witches$
- Частичное исключение $monarch$ обрабатывается веткой $A_{in} \circ T_s \circ T_y \circ A_{exc,sib} \circ T_{cl}$:
 $monarch+N+Pl \rightarrow monarch\hat{s} \rightarrow monarch\hat{s} \rightarrow monarch\hat{s} \rightarrow monarchs$.

Глагольные формы в языке йоулумни (америндская семья)

основа	герундий	дуратив
saw “кричать”	saw-inay	sawaa-ʔaa-n
siim “разрушать”	siim-inay	siimii-ʔaa-n
hoyoo “называть”	hoy-inay	hoyoo-ʔaa-n
diiyl “охранять”	diyl-inay	diyil-ʔaa-n
ʔilk “петь”	ʔilk-inay	ʔiliik-ʔaa-n
hiwiit “гулять”	hiwt-inay	hiwiit-ʔaa-n

Глагольные формы в языке йоулумни

Глагольные формы в языке йоулумни (америндская семья)

основа	герундий	дуратив
saw “кричать”	saw-inay	sawaa-ʔaa-n
siim “разрушать”	siim-inay	siimii-ʔaa-n
hoyoo “называть”	hoy-inay	hoyoo-ʔaa-n
diiyl “охранять”	diyl-inay	diyiil-ʔaa-n
ʔilk “петь”	ʔilk-inay	ʔiliik-ʔaa-n
hiwiit “гулять”	hiwt-inay	hiwiit-ʔaa-n

Глагольные формы в языке йоулумни

Если основа имела вид $\alpha_1 V(V)\alpha_2(V)(V)\alpha_3$, где $\alpha_1, \alpha_2 \in C$, $\alpha_3 \in \{C, \varepsilon\}$, то основа герундия имеет вид $\alpha_1 V\alpha_2\alpha_3$,

Глагольные формы в языке йоулумни (америндская семья)

основа	герундий	дуратив
saw “кричать”	saw-inay	sawaa-ʔaa-n
siim “разрушать”	siim-inay	siimii-ʔaa-n
hoyoo “называть”	hoy-inay	hoyoo-ʔaa-n
diiyl “охранять”	diyl-inay	diyil-ʔaa-n
ʔilk “петь”	ʔilk-inay	ʔiliik-ʔaa-n
hiwiit “гулять”	hiwt-inay	hiwiit-ʔaa-n

Глагольные формы в языке йоулумни

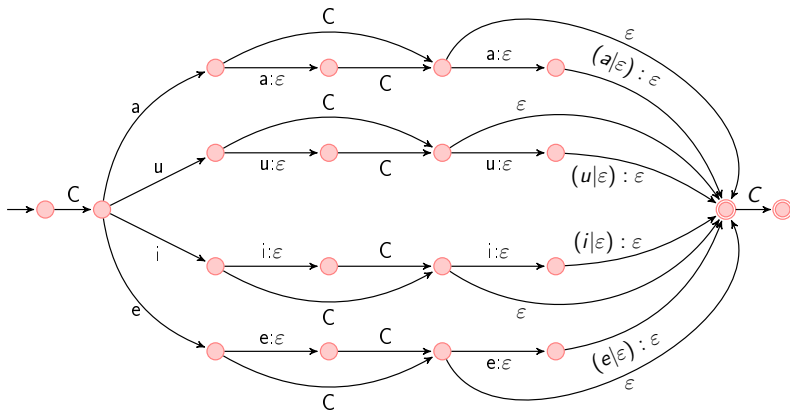
Если основа имела вид $\alpha_1 V(V)\alpha_2(V)(V)\alpha_3$, где $\alpha_1, \alpha_2 \in C$, $\alpha_3 \in \{C, \varepsilon\}$, то основа герундия имеет вид $\alpha_1 V\alpha_2\alpha_3$, а основа дуратива — $\alpha_1 V\alpha_2 VV\alpha_3$.

Преобразователи для глагольных форм в языке йоулумни

- Основа герундия:

Преобразователи для глагольных форм в языке йоулумни

- Основа герундия:



Преобразователи для глагольных форм в языке йоулумни

- Основа дуратива:

Преобразователи для глагольных форм в языке йоулумни

- Основа дуратива:

