

Восстановление после ошибок. Недетерминизм



Теория формальных языков
2021 г.

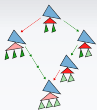


Синхронизирующиеся автоматы

DFA \mathcal{A} называется синхронизирующимся, если $\exists w, q_s \forall q_i (q_i \xrightarrow{w} q_s)$.

Критерий синхронизации

DFA \mathcal{A} синхронизирующийся $\Leftrightarrow \forall q, q' \exists w, q_x (q \xrightarrow{w} q_x \ \& \ q' \xrightarrow{w} q_x)$.



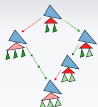
Синхронизирующиеся автоматы

DFA \mathcal{A} называется синхронизирующимся, если $\exists w, q_s \forall q_i (q_i \xrightarrow{w} q_s)$.

Критерий синхронизации

DFA \mathcal{A} синхронизирующийся $\Leftrightarrow \forall q, q' \exists w, q_x (q \xrightarrow{w} q_x \ \& \ q' \xrightarrow{w} q_x)$.

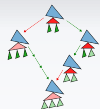
Рассмотрим слово w_1 , синхронизирующее q_1 и q_2 . Если w_1 синхронизирует все состояния, доказывать нечего. Иначе построим множество $Q_1 = \{q \mid q_i \xrightarrow{w_1} q\}$. По построению, $Q_1 \subset \{q_1, \dots, q_n\}$. Выберем в нём два первых состояния, q_i, q_j , и слово w_2 , синхронизирующее их. Построим множество $Q_2 = \{q \mid q_i \in Q_1 \ \& \ q_i \xrightarrow{w_2} q\}$. По построению, $Q_2 \subset Q_1$. Продолжив так не более чем $n - 1$ раз, построим синхронизирующее слово.



Задача Эшби о привидениях

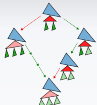
Дорогой друг! Недавно я купил старый дом, в котором обитают два призрака: Певун и Хохотун. Я установил, что их поведение подчиняется определенным законам, и что я могу воздействовать на них, играя на органе или сжигая ладан. В течение каждой минуты каждый из призраков либо шумит, либо молчит. Поведение же их в каждую минуту зависит только от минуты до этого, и эта зависимость такова.

Певун всегда ведет себя так же, как и в предыдущую минуту (звучит или шумит), если только в эту предыдущую минуту не было игры на органе при молчании Хохотуна. В последнем случае Певун меняет свое поведение на противоположное. Что касается Хохотуна, то, если в предыдущую минуту горел ладан, он будет вести себя так же, как Певун минутой раньше. Если, однако, ладан не горел, Хохотун будет вести себя противоположно Певуну в предыдущую минуту. Что мне делать, чтобы установить и поддерживать тишину в доме?



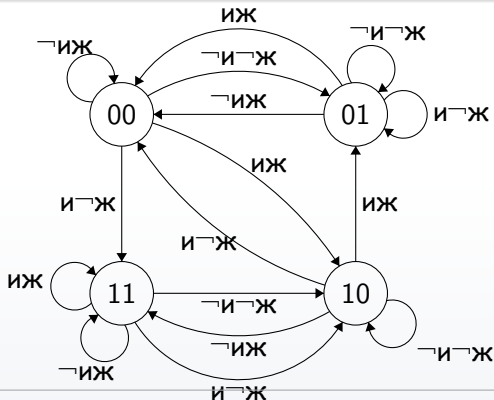
Задача Эшби о привидениях

- Если не играли на органе или Хохотун шумел, Певун не меняет поведение, иначе меняет.
- Если горел ладан, Хохотун делает то же, что делал Певун, иначе — противоположное.



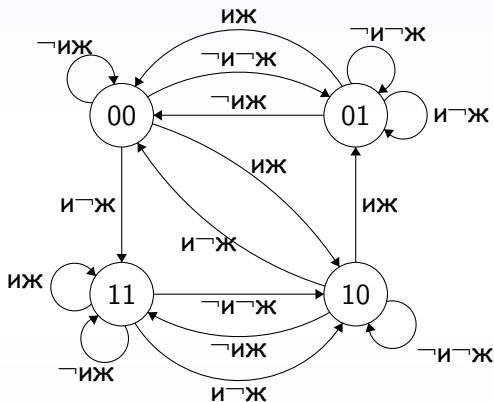
Задача Эшби о привидениях

- Если не играли на органе или Хохотун шумел, Певун не меняет поведение, иначе меняет.
- Если горел ладан, Хохотун делает то же, что делал Певун, иначе — противоположное.





Задача Эшби о привидениях



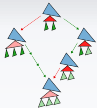
Синхронизирующее к состоянию 00 слово: $\neg \text{и}\neg\text{Ж}$, $\text{и}\neg\text{Ж}$, $\neg \text{иЖ}$.



Префиксное кодирование

Двоичное префиксное кодирование — это гомоморфизм $h : \Sigma^+ \rightarrow \{0, 1\}^+$ такой, что $\forall a, b \in \Sigma \forall w \in \{0, 1\}^* (h(a) \neq h(b)w)$.

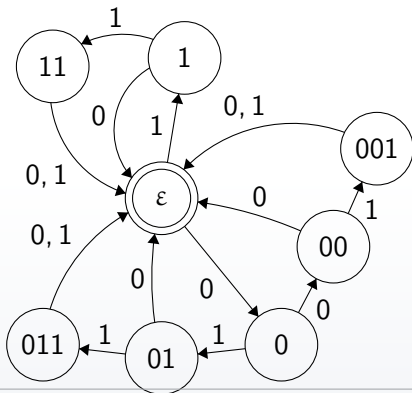
Рассмотрим префиксный код из 9-буквенного алфавита: $C = \{000, 0010, 0011, 010, 0110, 0111, 10, 110, 111\}$.



Префиксное кодирование

Рассмотрим префиксный код из 9-буквенного алфавита:
 $\mathcal{C} = \{000, 0010, 0011, 010, 0110, 0111, 10, 110, 111\}$.

Автомат-декодер для \mathcal{C} :



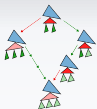


Коды, исправляющие ошибки

Префиксный код максимален, если к множеству кодирующих слов нельзя добавить ни одно слово без нарушения префикс-свойства.

Максимальный префиксный двоичный код \mathcal{C} называют синхронизированным, если $\exists z \in \{0, 1\}^+$, такое что $\forall u \in \{0, 1\}^+$ слово uz можно представить как конкатенацию слов из \mathcal{C} .

Если код \mathcal{C} синхронизирован, тогда ошибки в передаче закодированного слова будут исправляться сами при передаче достаточно длинной закодированной последовательности.



Коды, исправляющие ошибки

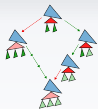
Префиксный код максимален, если к множеству кодирующих слов нельзя добавить ни одно слово без нарушения префикс-свойства.

Максимальный префиксный двоичный код \mathcal{C} называют синхронизированным, если $\exists z \in \{0, 1\}^+$, такое что $\forall u \in \{0, 1\}^+$ слово uz можно представить как конкатенацию слов из \mathcal{C} .

Если код \mathcal{C} синхронизирован, тогда ошибки в передаче закодированного слова будут исправляться сами при передаче достаточно длинной закодированной последовательности.

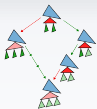
Утверждение

Максимальный префиксный код синхронизирован \Leftrightarrow его декодер — синхронизирующийся DFA.



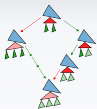
Подходы к восстановлению после ошибок в PDA

- Множество к.э. по Майхиллу–Нероуду бесконечно
⇒ синхронизация учитывает стек.



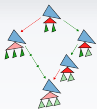
Подходы к восстановлению после ошибок в PDA

- Множество к.э. по Майхиллу–Нероуду бесконечно
⇒ синхронизация учитывает стек.
- Стандартный подход: множество синхронизирующих терминалов строится для каждого нетерминала отдельно.



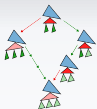
Подходы к восстановлению после ошибок в PDA

- Множество к.э. по Майхиллу–Нероуду бесконечно
⇒ синхронизация учитывает стек.
- Стандартный подход: множество синхронизирующих терминалов строится для каждого нетерминала отдельно.
- (режим паники) При восстановлении после ошибки отбрасывается не только префикс ошибочного входа, но и вершина стека.



Подходы к восстановлению после ошибок в PDA

- Множество к.э. по Майхиллу–Нероуду бесконечно
⇒ синхронизация учитывает стек.
- Стандартный подход: множество синхронизирующих терминалов строится для каждого нетерминала отдельно.
- (режим паники) При восстановлении после ошибки отбрасывается не только префикс ошибочного входа, но и вершина стека.
- (режим починки) При восстановлении после ошибки стек не отбрасывается, а вход подгоняется под стек. Набор действий может зависеть от ячейки таблицы, содержащей ошибку.



Panic mode для LL-разбора

Ошибочная ситуация

Терминал в стеке не совпадает с терминалом на ленте, либо переход по таблице правил приводит к ошибке.

- Отбрасываем вершину стека до синхронизирующего токена и входные символы до успеха перехода по нему.
- Возможное удаление \Rightarrow для токена A синхронизирующими могут предполагаться элементы $\text{FOLLOW}(A)$.
- Возможная вставка \Rightarrow синхронизирующие — $\text{FIRST}(A)$. Если конфликт терминалов — интерпретируем как возможную вставку.



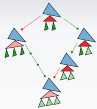
Panic mode для LR-разбора

Ошибочная ситуация

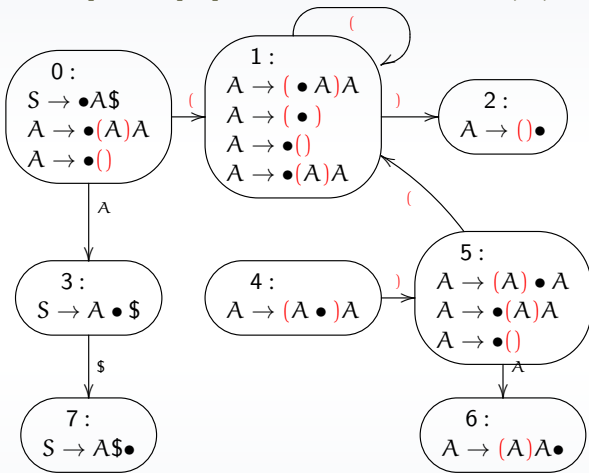
Переход по таблице правил приводит к ошибке.

- Вводим специальный токен «ошибка» в правиле $A \rightarrow \beta \bullet \alpha$, на котором она произошла.
- Отбрасываем вершину стека до свёртки по правилу $A \rightarrow \text{«ошибка»} \alpha \bullet$, не добавляя ничего в стек (если есть lookahead, то до совпадения с lookahead-ом).
Продолжаем разбор дальше.

Альтернатива: поиск «починки» — минимального количества действий, позволяющего возобновить парсинг.

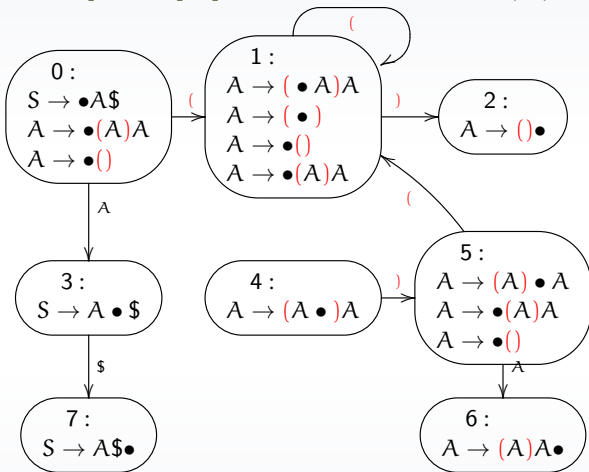


Пример panic mode в LR(0)-парсере

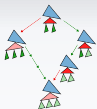




Пример panic mode в LR(0)-парсере



Разбор строки $()() \$$: $([0], ()() \$) \rightarrow ([1, 0],)() \$) \rightarrow ([2, 1, 0], () \$) \rightarrow ([3, 0], () \$)$
 На этом шаге происходит ошибка. Строим $S \rightarrow \bullet \text{«ошибка»} \$$, отбрасываем $()$ и редуцируемся в S .



Бурке–Фишер и его вариации

Идея алгоритма

При заранее заданном k и ошибке на i -ом терминале входа рассмотреть возможные последовательности терминалов от i -ого до $i + k - 1$ -ого, продолжающие парсинг, и выбрать в качестве «починки» ту из них, расстояние Левенштейна до которой от реального входа наименьшее.

- (Corchuello et al) Также разрешается делать операции сдвига по lookahead-у.
- (Diekmann et al) Ищутся все возможные варианты «починки» и выбирается тот из них, который позволяет продолжить разбор на наибольшую глубину.



Степень недетерминизма языка

Если PDA \mathcal{A} допускает декомпозицию на DPDA, между которыми есть максимум k недетерминированных переходов, но не допускает такую декомпозицию при $i < k$ переходов, скажем, что \mathcal{A} задаёт КС-язык с k -недетерминированностью.



Степень недетерминизма языка

Если PDA \mathcal{A} допускает декомпозицию на DPDA, между которыми есть максимум k недетерминированных переходов, но не допускает такую декомпозицию при $i < k$ переходов, скажем, что \mathcal{A} задаёт КС-язык с k -недетерминированностью.

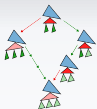
- 1 Степень недетерминированности языка $\{a^n b^n\} \cup \{a^n b^{2n}\}$?



Степень недетерминизма языка

Если PDA \mathcal{A} допускает декомпозицию на DPDA, между которыми есть максимум k недетерминированных переходов, но не допускает такую декомпозицию при $i < k$ переходов, скажем, что \mathcal{A} задаёт КС-язык с k -недетерминированностью.

- 1 Степень недетерминированности языка $\{a^n b^n\} \cup \{a^n b^{2n}\}$?
Ответ: 1
- 2 Степень недетерминированности языка $\{a^n b^n\} \cup \dots \cup \{a^n b^{k \cdot n}\}$?



Степень недетерминизма языка

Если PDA \mathcal{A} допускает декомпозицию на DPDA, между которыми есть максимум k недетерминированных переходов, но не допускает такую декомпозицию при $i < k$ переходов, скажем, что \mathcal{A} задаёт КС-язык с k -недетерминированностью.

- 1 Степень недетерминированности языка $\{a^n b^n\} \cup \{a^n b^{2n}\}$?
Ответ: 1
- 2 Степень недетерминированности языка $\{a^n b^n\} \cup \dots \cup \{a^n b^{k \cdot n}\}$? Ответ: тоже 1 (см. критерий исправляемости)
- 3 Степень недетерминированности языка $\{ww^R\}$ — также 1.
- 4 Степень недетерминированности языка $\{ww^R vv^R\}$ равна 2.



Исправление недетерминированности

Пусть L — недетерминированный КС-язык и $k > 0$. Язык L — k -исправляемый, если существует алфавит Δ , $\Delta \cap \Sigma = \emptyset$ и $\text{DCFL } L(k) \subseteq (\Sigma \cup \Delta)^*$ такой, что для $h(\Delta) = \varepsilon$, $h(L(k)) = L$ и все слова языка $L(k)$ содержат не больше k букв из Δ .

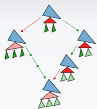
Язык L имеет k -ую степень недетерминизма $\Leftrightarrow L$ k -исправляемый, но не $k-1$ -исправляемый.



Исправляемость и анализ на DCFL

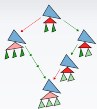
Техника использования леммы о накачке для DCFL

- анализируем позиции в словах языка L , в которых может произойти смена наполнения стека на его опустошение, а может не произойти. Такие позиции считаем подозрительными на исправляемость.
- подбираем два слова из L , xuz_1 , xuz_2 такие, что исправляемая позиция находится в подслове u , причём в подслове u слова xuz_1 происходит наполнение стека, а в слове xuz_2 стек опустошается либо игнорируется.
- убеждаемся, что отдельно x накачать нельзя, после чего рассматриваем накачки uz_1 и uz_2 . Из-за разного поведения стека на их префиксах, скорее всего, эти накачки будут выводить из языка L .



Иерархия недетерминированных КС-языков

Семейство языков $w_1 w_1^R \$ \dots w_k w_k^R \$$ ($\$ \notin \Sigma$) задаёт бесконечную иерархию недетерминированных языков с k -недетерминизмом.



Иерархия недетерминированных КС-языков

Семейство языков $w_1 w_1^R \$ \dots w_k w_k^R \$$ ($\$ \notin \Sigma$) задаёт бесконечную иерархию недетерминированных языков с k -недетерминизмом.

- Введение вложенных структур с совпадающими маркерами начала и конца приводит к неограниченному недетерминизму.