Восстановление после ошибок. Недетерминизм

Теория формальных языков

2021 г.

Синхронизирующиеся автоматы

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

Критерий синхронизации

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

Синхронизирующиеся автоматы

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

Критерий синхронизации

DFA $\mathscr{A}$ 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

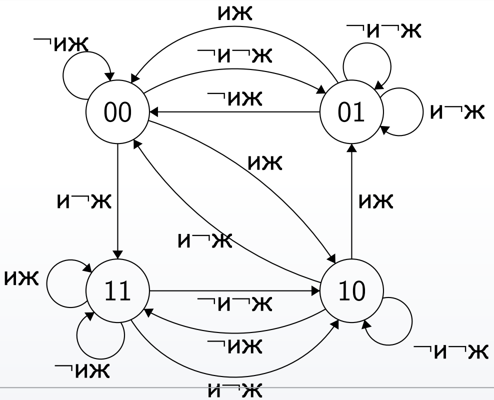
Дорогой друг! Недавно я купил старый дом, в котором обитают два призрака: Певун и Хохотун. Я установил, что их поведение подчиняется определенным законам, и что я могу воздействовать на них, играя на органе или сжигая ладан. В течение каждой минуты каждый из призраков либо шумит, либо молчит. Поведение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

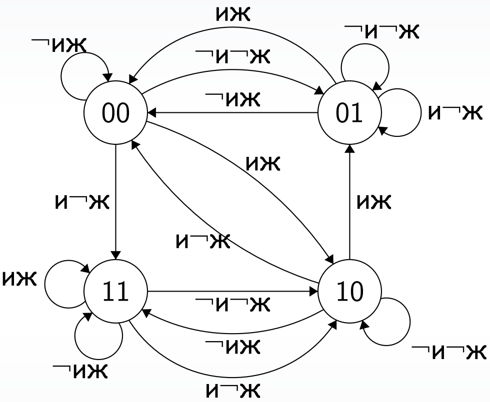
- Если не играли на органе или Хохотун шумел, Певун не меняет поведение, иначе меняет.
- Если горел ладан, Хохотун делает то же, что делал Певун, иначе — противоположное.

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: ¬и¬ж, и¬ж, ¬иж.

Префиксное кодирование

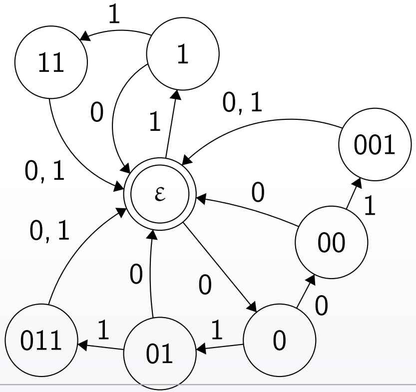
Двоичное префиксное кодирование — это гомоморфизм $h : \Sigma^+ \to \{0,1\}^+$ такой, что $\forall a, b \in \Sigma \, \forall w \in \{0,1\}^* (h(a) \neq h(b)w)$.

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

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 $\forall y \in \{0,1\}^+$ слово yz 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 $\forall y \in \{0,1\}^+$ слово yz 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 $\Rightarrow$ синхронизация учитывает стек.

Подходы к восстановлению после ошибок в PDA

- Множество к.э. по Майхиллу–Нероуду бесконечно $\Rightarrow$ 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 $\Rightarrow$ 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 $\Rightarrow$ 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 FOLLOW(A).
- Возможная вставка $\Rightarrow$ синхронизирующие — FIRST(A). Если конфликт терминалов — интерпретируем как возможную вставку.

Panic mode для LR-разбора

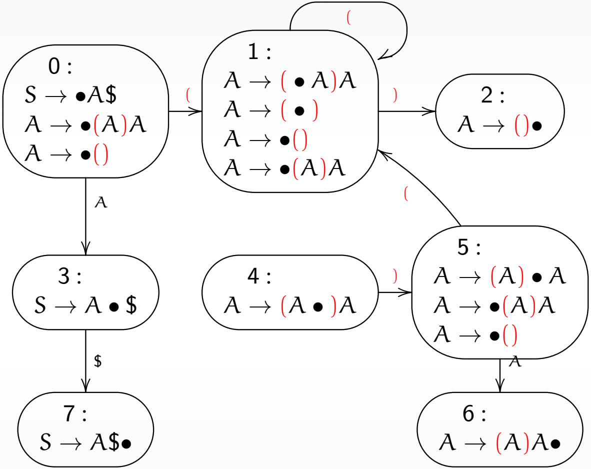
Ошибочная ситуация

Переход по таблице правил приводит к ошибке.

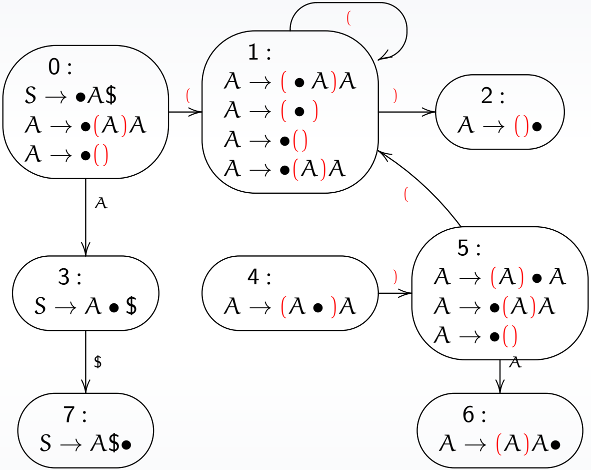
- Вводим специальный токен «ошибка» в правиле $A \to \beta \bullet \alpha$, на котором она произошла.
- Отбрасываем вершину стека до свёртки по правилу $A \to$«ошибка»$\alpha\bullet$, не добавляя ничего в стек (если есть lookahead, то до совпадения с lookahead-ом). Продолжаем разбор дальше.

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 $()()\$$: $([0], ()()\$) \to ([1,0],)()\$) \to ([2,1,0], ()\$) \to ([3,0], ()\$)$
На этом шаге происходит ошибка. Строим $S \to \bullet$«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 $\mathscr{A}$ допускает декомпозицию на DPDA, между которыми есть максимум k недетерминированных переходов, но не допускает такую декомпозицию при $i < k$ переходов, скажем, что $\mathscr{A}$ задаёт КС-язык с k-недетерминированностью.

Степень недетерминизма языка

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

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

Степень недетерминизма языка

Если PDA $\mathscr{A}$ допускает декомпозицию на DPDA, между которыми есть максимум k недетерминированных переходов, но не допускает такую декомпозицию при $i < k$ переходов, скажем, что $\mathscr{A}$ 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 $\{a^n b^n\} \cup \ldots \cup \{a^n b^{k*n}\}$?

Степень недетерминизма языка

Если PDA $\mathscr{A}$ допускает декомпозицию на DPDA, между которыми есть максимум k недетерминированных переходов, но не допускает такую декомпозицию при $i < k$ переходов, скажем, что $\mathscr{A}$ 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 $\{a^n b^n\} \cup \ldots \cup \{a^n b^{k*n}\}$? 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 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, xyz_1, xyz_2 такие, что исправляемая позиция находится в подслове y, причём в подслове y слова xyz_1 происходит наполнение стека, а в слове xyz_2 стек опустошается либо игнорируется.
- убеждаемся, что отдельно x накачать нельзя, после чего рассматриваем накачки yz_1 и yz_2. Из-за разного поведения стека на их префиксах, скорее всего, эти накачки будут выводить из языка L.

Пример применения

Проанализировать контекстно-свободный язык $L = \{wa^n c^n w^R \mid w \in \{a, b\}^*\}$.

- В словах языка есть произвольные подслова из $\{a, b\}^*$, что усложняет анализ. К тому же есть блок c^n, который на первый взгляд однозначно указывает на детерминизм, однако нет условия $n \geqslant 1$, поэтому в некоторых случаях на его существование нельзя положиться. Воспользуемся замкнутостью DCFL относительно пересечений с регулярными языками, избавимся от c^n и сузим область накачек.
- Простейший язык, с которым мы можем пересечь L для этой цели: $a^*b^*a^*$, после чего взять $xy = a^m$, $z_1 = a^{n_1}$, $z_2 = a^{n_2}b^{2*n_3}a^{m+n_2}$.

Пример применения

Проанализировать контекстно-свободный язык $L = \{wa^n c^n w^R \mid w \in \{a, b\}^*\}$.

- Нужно избавиться от подслова с буквами c и сузить область накачек.
- Простейший язык, с которым мы можем пересечь L для этой цели: $a^*b^*a^*$, после чего взять $xy = a^m$, $z_1 = a^{n_1}$, $z_2 = a^{n_2}b^{2*n_3}a^{m+n_2}$.
- Хотя поведение стека на этих фрагментах слов соответствует рекомендуемому, анализ ни к чему не приводит: мы без проблем можем накачивать в этих словах одновременно суффикс послова xy и элементы z_1 и z_2, а всё потому, что слова в языке a^*, являющиеся палиндромами, описываются регулярными выражениями. Искомое пересечение языков неудачное, выберем то, которое чётче обозначит нерегулярную структуру палиндрома.

Пример применения

Проанализировать контекстно-свободный язык $L = \{w a^n c^n w^R \mid w \in \{a, b\}^*\}$.

- Рассмотрим пересечение L с языком $a^* b^* a^* b^* a^*$. В нём уже будут два типа палиндромов, не распознаваемые регулярками (с одним или двумя подсловами, состоящими из букв b).
- Абеляр (т.е. антагонист) выбирает длину накачки p.
- Элоиза (т.е. мы) выбирает слова $a^{p+1} b a^{p+1}$ и $a^{p+1} b a^{p+1} a^{p+1} b a^{p+1}$ и $xy = a^{p+1} b a^p$, $z_1 = a$, $z_2 = a^{p+2} b a^{p+1}$.
- Абеляр не может накачивать только $a^{p+1} b a^p$: при накачке только второго a^p произойдёт рассинхронизация с суффиксом z_1, а при любой накачке с участием первого a^{p+1} — рассинхронизация с суффиксом z_2.
- Значит, Абеляру остаётся только накачивать подслово суффикса a^p синхронно с подсловом z_1 (т.е. a) (и некоторым подсловом z_2, но это уже не важно), что также приводит к выходу из языка палиндромов.

Пример применения

Проанализировать контекстно-свободный язык $L = \{wa^n c^n w^R \mid w \in \{a, b\}^*\}$.

- Рассмотрим пересечение L с языком $a^*b^*a^*b^*a^*$.
- Абеляр (т.е. антагонист) выбирает длину накачки p.
- Элоиза (т.е. мы) выбирает слова $a^{p+1}ba^{p+1}$ и $a^{p+1}ba^{p+1}a^{p+1}ba^{p+1}$ и $xy = a^{p+1}ba^p$, $z_1 = a$, $z_2 = a^{p+2}ba^{p+1}$.
- Абеляр не может накачивать только $a^{p+1}ba^p$: при накачке только второго a^p произойдёт рассинхронизация с суффиксом z_1, а при любой накачке с участием первого a^{p+1} — рассинхронизация с суффиксом z_2.
- Значит, Абеляру остаётся только накачивать подслово суффикса a^p синхронно с подсловом z_1 (т.е. a) (и некоторым подсловом z_2, но это уже не важно), что также приводит к выходу из языка палиндромов.
- Заметим, что если взять слова a^pba^p и $a^pba^pa^pba^p$ и $xy = a^pba^{p-1}$, тогда синхронную накачку придумать можно: накачивать в xy букву b (она ещё в пределах длины накачки), в z_2 её же, а в z_1 ничего.
- Мы показали, что язык пересечения — NCFL, значит, язык L — NCFL.

Иерархия недетерминированных КС-языков

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

Иерархия недетерминированных КС-языков

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

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