

Формальные языки

Индивидуальное домашнее задание для Н. Козубаева

29.10.2018

1. Доказать или опровергнуть следующее свойство языков: $(M \cup N) \cdot L = M \cdot L \cup N \cdot L$ для произвольных языков L, M, N .
2. Построить минимальный детерминированный конечный автомат, распознающий язык

$$\{a, b, c\}^+ \setminus (\{a^n \mid n \geq 0\} \cup \{c^n \mid n \geq 0\})$$

3. Построить регулярную грамматику, задающую язык

$$\{a \cdot \alpha \cdot ab \cdot \beta \cdot bb \mid \alpha, \beta \in \{a, b\}^*\} \cap \{aa \cdot \gamma \cdot bb \cdot \delta \cdot b \mid \gamma, \delta \in \{a, b\}^*\}$$

4. Проверить регулярность языка (если регулярный, построить автомат, регулярное выражение или регулярную грамматику, иначе — доказать нерегулярность)

$$\{a^k \omega b^k \mid k \geq 0, \omega \in \{a, b\}^*, |\omega|_a \leq 3\}$$

5. Построить недетерминированный автомат по регулярному выражению $a^*ba^+ba^*(ba^* \mid \varepsilon)$, построить эpsilon-замыкание автомата
6. Используя любой понравившийся способ писать парсер, реализовать синтаксический анализатор для подмножества языка пролог, описание которого приведено ниже.

Описание синтаксиса

- Лексический синтаксис

- Идентификатор — последовательность букв латинского алфавита в любом регистре, цифр и символа нижнего подчеркивания ($_$), начинающаяся на букву латинского алфавита в нижнем регистре.
 - * Корректные идентификаторы: `x`, `list`, `listNat_123`.
 - * Некорректные идентификаторы: `Abc`, `123`, `_List`.
- Переменная — последовательность букв латинского алфавита в любом регистре, цифр и символа нижнего подчеркивания ($_$), начинающаяся на букву латинского алфавита в верхнем регистре.
 - * Корректные переменные: `X`, `XyZ`, `ABC`.
 - * Некорректные переменные: `abc`, `123`, `X Y`.
- Ключевые слова: `module`, `type` — не могут быть идентификаторами.
- Другие символы, которые есть в алфавите: штихор `-`, стрелка `->`, точка `.`, запятая `,`, точка с запятой `;`, вертикальная черта `|`, скобки `(` и `)`.

- Базовый синтаксис

- Программа на прологе начинается с *объявления модуля*, за которым идет последовательность *определений отношений*.

- Объявление модуля это ключевое слово `module`, за которым идет идентификатор (не переменная), завершается все это точкой `.: module example`.
- Определение отношения состоит из *головы* и *тела*, разделенных штопором `(:-)`, в конце стоит точка `(.)`. *Перед* определением отношения может быть приведено *определение его типа* (а может не быть).
- Голова отношения — атом.
- Атом — идентификатор, за которым в скобках `((,))` идет последовательность атомов или переменных, разделенных запятыми `(,)`.
- У атома может не быть аргументов, тогда он имеет вид идентификатора (без пустых скобок).
- Тело отношения — выражение с правоассоциативными бинарными операторами конъюнкции `(,)` и дизъюнкции `(;)` над атомами. Конъюнкция имеет более высокий приоритет, чем дизъюнкция. Возможно использование скобок `((,))` для управления порядком вычислений.
- Тело и штопор могут отсутствовать (вместе).
- Где угодно могут встречаться пробельные символы: советую использовать лексер.
- Определение типа это строка, начинающаяся с ключевого слова `type`, за которым идет идентификатор-имя типа, за которым идет тип, в конце — точка `..`. Тип это последовательность атомов, переменных или типов, разделенных стрелкой `->`.
 - * Корректные определения типов
 - `type filter (A -> o) -> list(A) -> list(A) -> o.`
 - `type fruit string -> o.`
 - * Некорректные определения типов
 - `type type type -> type.` — ключевое слово используется как идентификатор
 - `type lst list nat.` — `list nat` не является корректным типом
 - `type x -> y -> z.` — нет имени типа
 - `tupe x o.` — идентификатор вместо ключевого слова `type`
- Синтаксический сахар для списков:
 - * Список это возможно пустая последовательность атомов или переменных, разделенных запятой `(,)`, находящаяся в квадратных скобках `[,]`.
 - Корректные списки: `[], [X, Y, Z], [a(b(c)), d, Z]`.
 - Некорректные списки: `[,]a, b, c[`
 - * Можно специфицировать список, состоящий из головы `H` и хвоста `T`: `[H|T]`. В голове может быть не только переменная, но и произвольный атом, хвост — обязательно переменная.
 - Корректные списки: `[H | T], [a(b(c)) | T]`
 - Некорректные списки: `[H | abc], [H | A(b, c)]`
 - * Комбинация двух предыдущих способов: несколько первых элементов через запятую, дальше `|`, дальше переменная.
 - Корректные списки: `[H, H1 | T], [a(b(c)), Second, Third | T]`
 - Некорректные списки: `[H, | abc], [| A]`
 - * Список может вкладываться в другой список: `[[X, [H | T]] | Z]`
 - * Список не может быть головой атома, но может быть аргументом.

- Некорректное использование списка: $[X](Y) :- f(X, Y).$
- Корректное использование списка: $g([X], Y) :- f(X, Y).$
- * Важно: это синтаксический сахар, он не должен добавлять новые сущности в абстрактный синтаксис.
 - $[X, Y \mid T] = \text{cons}(X, \text{cons}(Y, T))$
 - $[X, Y, Z] = \text{cons}(X, \text{cons}(Y, \text{cons}(T, \text{nil})))$
 - $[a(b(c)) \mid T] = \text{cons}(a(b(c)), T)$
- Примеры корректных определений отношений.
 - $f.$
 - $f :- g.$
 - $f :- g, h; t.$
 - $f :- g, (h; t).$
 - $f(a) :- g, h(t, c, d).$
 - $f(\text{cons}(h, t)) :- g(h), f(t).$
- Примеры некорректных определений отношений.
 - f — нет точки.
 - $:- f.$ — нет головы.
 - $f :- .$ — нет тела.
 - $f :- g; h, .$ — нет правого подвыражения у конъюнкции.
 - $f :- (g; (f)).$ — несбалансированные скобки.
 - $f ().$ — пустые скобки
- Требования к оформлению
 - Результатом должно быть консольное приложение, которое принимает на вход программу и печатает результат синтаксического анализа в файл с таким же названием и дополнительным расширением `.out`. (`input.txt` \rightarrow `input.txt.out`)
 - Результатом синтаксического анализа является абстрактное синтаксическое дерево в случае успешного разбора и сообщение об ошибке иначе.
 - * Если произошла лексическая ошибка, то сообщить о ней и завершиться, не пытаясь дальше парсить.
 - * Если произошла любая ошибка — сообщить о ней и завершиться, не пытаясь восстанавливаться после ошибки.
 - Код может быть написан на любом языке программирования с использованием любых инструментов, но должен быть сопровожден инструкцией по сборке и запуску. Желательно выложить его на гитхаб и сопроводить тестами.