

# Формальные языки

домашнее задание до 23:59 01.10

1. Существует ли такой регулярный язык  $L$ , что язык всех его подстрок не является регулярным? Обосновать.
2. Является ли язык, задаваемый следующей грамматикой регулярным? Если является, привести регулярное выражение или конечный автомат. Если нет — обосновать.

$$S \rightarrow TU$$

$$U \rightarrow TU \mid b$$

$$T \rightarrow UU \mid a$$

3. Равны ли регулярные выражения  $b^*a((a \mid b) b^*a)^*$  и  $((a \mid b)^* ba \mid a)(aa)^*$  над алфавитом  $\{a, b\}$ ? Обосновать.
4. Изучить спецификации синтаксиса бинарных операторов трех языков программирования, на которых вы никогда не писали. В отчете привести ссылки на спецификации, а также указать, что было в них для вас неожиданным.
5. Релизовать синтаксический анализатор методом рекурсивного спуска для упрощенного донельзя подмножества пролога (оставим только определения отношений, лишим отношения аргументов).

- Описание синтаксиса

- Программа на прологе — последовательность *определений отношений*.
- Определение отношений состоит из *головы* и *тела*, разделенных штопором  $(:-)$ , в конце стоит точка  $(.)$ .
- Голова — идентификатор.
- Тело — выражение с правоассоциативными бинарными операторами конъюнкции  $(,)$  и дизъюнкции  $(;)$  над идентификаторами. Конъюнкция имеет более высокий приоритет, чем дизъюнкция. Возможно использование скобок  $((, ))$  для управления порядком вычислений.
- Тело и штопор могут отсутствовать.
- Где угодно могут встречаться пробельные символы: советую использовать лексер.

- Примеры корректных определений отношений.

- $f.$
- $f :- g.$
- $f :- g, h; t.$
- $f :- g, (h; t).$

- 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.
- Код должен быть сопровожден инструкцией по сборке и запуску. Желательно выложить его на гитхаб и сопроводить тестами.
- Примеры простых реализаций парсеров методом рекурсивного спуска:
  - <https://vey.ie/2018/10/04/RecursiveDescent.html>
  - <http://www.cs.utsa.edu/~wagner/CS3723/rdparse/rdparser6.html>
  - <https://prgwonders.blogspot.com/2017/10/recursive-descent-parser-in-c-for.html>
  - <https://craftinginterpreters.com/parsing-expressions.html>