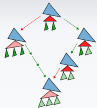




## Дедлайнерская Лаб. работа №1

Базовая стоимость этой лабораторной работы — 6 баллов.

- ❶ (ваш номер — 0 в поле  $\mathbb{Z}_3$ ) Автоматическое построение таблицы классов эквивалентности по Майхиллу–Нероуду по ДКА.
- ❷ (ваш номер — 1 в поле  $\mathbb{Z}_3$ ) Построение пересечения регулярной и контекстно-свободной грамматики.
- ❸ (ваш номер — 2 в поле  $\mathbb{Z}_3$ ) LL-корректное устранение  $\varepsilon$ -правил в грамматике.



## Синтаксис входных данных

- ❶ (0 в поле  $\mathbb{Z}_3$ ) Синтаксис ДКА такой же, как в нечётном варианте 2 лабораторной работы, но состояния-ловушки могут не присутствовать в автомате явно. Гарантируется, что по каждой букве из каждого состояния не больше одного перехода.
- ❷ (1 в поле  $\mathbb{Z}_3$ ) Синтаксис КС-грамматики такой же, как в чётном варианте 3 лабораторной работы (и так же в псевдо-GNF и без  $\varepsilon$ -правил). Синтаксис регулярной грамматики такой же, как в чётном варианте 2 лабораторной работы (строго праволинейный, без  $\varepsilon$ -правил). Читаются из двух отдельных файлов, CFG\_x.txt и RG\_x.txt.
- ❸ (2 в поле  $\mathbb{Z}_3$ ) Синтаксис КС-грамматики такой же, как в чётной 4 лабораторной работе.



## Извлечение классов эквивалентности

- 1 В ДКА находятся все пути из начального состояния в другие состояния, которые не проходят дважды ни через одну вершину. Слова, распознаваемые вдоль таких путей, считаются потенциальными классами эквивалентности. Если на этом этапе вдоль разных путей получились одинаковые слова, необходимо отбросить дубликаты.
- 2 Для каждого построенного слова  $\beta$  подбираются два суффикса  $\gamma_1$  и  $\gamma_2$ : так что слово  $\beta\gamma_1$  распознаётся автоматом, а слово  $\beta\gamma_2$  не распознаётся. Иногда такие  $\gamma_1$  или  $\gamma_2$  не удастся построить. Поэтому необходимо определить множество состояний-ловушек и множество «совершенно конечных» состояний (тех, переходы из которых делаются только в конечные состояния). Если слово  $\beta$  распознаётся в ловушке,  $\gamma_1$  не существует; если в совершенно конечном состоянии – не существует  $\gamma_2$ . Если хотя бы одно из них построить не удалось, тогда не добавляем в таблицу ничего.
- 3 Добавить в список суффиксов  $\varepsilon$ . Для каждого построенного префикса  $\beta_i$  и суффикса  $\gamma_j$  проверить, распознаётся ли слово  $\beta_i\gamma_j$  ДКА. Представить таблицу-результат в формате markdown.



## Построение пересечения грамматик

- 1 Базовый алгоритм можно взять из лекций (лекция 7).
- 2 По исчерпанию удалить все недостижимые и непорождающие нетерминалы и содержащие их правила.
- 3 До неподвижной точки применить алгоритм альфа-преобразования: если для всех правил, содержащих нетерминал  $A$ , можно заменить  $A$  на  $B$ , и при этом получатся уже существующие правила грамматики, тогда заменить  $A$  на  $B$  во всех правилах и удалить все дубликаты.



## LL-корректное удаление $\varepsilon$ -правил

- 1 Базовый алгоритм можно взять из лекций (лекция 9). До его применения удалить все недостижимые и непорождающие нетерминалы и содержащие их правила. Учсть, что исходная грамматика допускает нетерминалы вида  $[N_1 N_2]$ , поэтому при присоединении контекста использовать разделитель (например, +, или по Дьячеку,  $->$ ).
- 2 Если грамматика не LL(k), тогда алгоритм может зациклиться. Произойти это может, по лемме Розенкранца и Стирнса, только если в результате присоединения обнуляемого контекста возникнет нетерминал вида  $[(\Phi_1 ->)? A(-> \Phi_2)? -> A(-> \Phi_3)?]$  (т.е. такой, в котором дважды присоединяется один и тот же обнуляемый нетерминал). В таких случаях прерывать исполнение, печатать промежуточную грамматику, в которой появился проблемный нетерминал, и сообщать, что исходная грамматика — не LL(k).



## Дополнительные баллы

- 1 (0 в поле  $\mathbb{Z}_3$ ) (+2) Отрисовывать ДКА в виде dot-графа, а таблицу — в  $\text{\LaTeX}$ -формате, после чего собирать их в единый  $\text{\LaTeX}$ -файл с двумя рисунками (окружение figure): автомат и таблица его классов.
- 2 (1 в поле  $\mathbb{Z}_3$ ) (+2) На вход алгоритма может приниматься как регулярная грамматика, так и конечный автомат, описанный в синтаксисе 2 лабораторной работы. В этом случае входной файл будет DFA\_x.txt.
- 3 (2 в поле  $\mathbb{Z}_3$ ) (+2) Проверить исходную и результирующую грамматику на LL(1)-свойства, стандартным алгоритмом построения LL-таблицы.