Дополнительное задание

Полностью обесценится 6 ноября. При ручном выполнении стоит 2 балла, с реализацией — 3 балла.

Рассматриваются следующие операции над грамматикой: F_1 — удаление ε-правил, F_2 — удаление цепных правил, F_3 — получение правил с длиной правой части не больше 2.

- Установить, при каких порядках применения этих правил более не потребуется применять ни одно из них (т.е. результат не изменится). Привести пример случая, когда все три операции были сделаны, но их повторное применение ещё требуется.
- Привести пример, когда при разной очерёдности применения правил F_i результирующая грамматика оказывается существенно разного размера.

Лабораторная номер 3

1. ($\equiv 0 \mod 3$) По данной контекстно-свободной грамматике построить её коммутативный образ по алгоритму Пиллинга.
2. ($\equiv 1 \mod 3$) Экспериментальная верификация гомоморфизма Хомского–Шютценберже.
3. ($\equiv 2 \mod 3$) По данной контекстно-свободной грамматике построить её нормальную форму Грейбах методом устранения левой рекурсии и по Блюму–Коху.

Коммутативный образ

Лемма Ардена для коммутативных образов

Пусть правила грамматики имеют следующий вид:

$$T \to W_1 T^{i_1} T \mid \ldots \mid W_{k-1} T^{i_{k-1}} T \mid W_k$$

где W_i не содержит T, и степени i_j различны. При этом W_i могут содержать любые регулярные операции над константами и нетерминалами, отличными от T. Коммутативный образ правил для T есть $T = (W_1 (W_k)^{i_1})^* W_k + \cdots + (W_{k-1} (W_k)^{i_{k-1}})^* W_k$.

Коммутативный образ

1. Удалить цепные и ε-правила.
2. Записать уравнения, соответствующие правилам грамматики. Выбрать порядок на нетерминалах, согласно которому будет применяться лемма Ардена.
3. Поочередно применить лемму Ардена для коммутативных образов ко всем уравнениям, рассматривая в каждом очередном уравнении все переменные, кроме выделенной, как константы. После каждого применения леммы подставить полученный результат в оставшиеся уравнения.
4. Когда будет обработано последнее уравнение, выполнить обратную подстановку и получить регулярные выражения — коммутативные образы языков всех нетерминалов грамматики.

Разбиение итерации

- Если требуется получить коммутативный образ уравнения $X = (\Phi(X))^*\Psi$, где Ψ не содержит вхождений X, то нужно разобрать случаи нулевой и ненулевой итерации подвыражения $\Phi(X)$, содержащего X (вне альтернатив и итераций), чтобы явным образом извлечь вхождение X.
- Например: $X = (((aX)^*c)|b)^*Y$ превращаем в сумму $X = (c|b)^*Y + ((((aX)^*c)|b)^*acY)X$. Первое слагаемое соответствует ситуации, когда итерация $(aX)^*$ всегда раскрывается в ε, второе — когда она хотя бы раз происходит, т.е. приписывается acX.

Дополнительные баллы

- Использовать упрощающие соотношения посредством применения модуля, реализованного в предыдущей Л.Р. (+2 балла) или путём непосредственной их реализации (+1 балл) для более компактного представления регулярного выражения.
- Коммутативный образ задаёт точные полулинейные отношения на вхождения определённых букв в слова языка. Определить эти соотношения (в форме векторов). Например: $(ab)^*a(bb)^*$ определяет соотношения: $|w|_a = x_0 + 1$, $|w|_b = x_0 + 2 \cdot x_1$ (3 балла).

Тест гомоморфизма Шютценберже

Входные данные лабораторной работы представляют собой КС-грамматику G, описание гомоморфизма h и регулярное выражение R над скобочными структурами. Требуется экспериментально проверить, действительно ли данный гомоморфизм h и регулярное выражение R являются описанием Шютценберже языка данной грамматики.

Синтаксис данных (кроме грамматики) приведён ниже. $\langle$br-code$\rangle$ — это способ кодировки скобок. Левая скобка типа Ψ помечена как $(\Psi($, правая — как $)\Psi)$.

$$\begin{array}{rcl}
\langle\text{Hom-rule}\rangle & ::= & h(\langle\text{br-code}\rangle)=\langle\text{term}\rangle^{*} \\
\langle\text{br-code}\rangle & ::= & ([A\text{–}z]^{+}(\mid)[A\text{–}z]^{+}) \\
\langle\text{regex}\rangle & ::= & \langle\text{regex}\rangle\ \langle\text{binary}\rangle\ \langle\text{regex}\rangle \mid (\langle\text{regex}\rangle) \\
 & & \mid \langle\text{regex}\rangle * \mid \langle\text{br-code}\rangle \mid \varepsilon \\
\langle\text{binary}\rangle & ::= & \mid\ \mid \varepsilon
\end{array}$$

Тест гомоморфизма Шютценберже

1. Построить дополнение образа $h(R)$, пересечь его с грамматикой G. Если полученная грамматика порождает не пустой язык, вывести слово, ему принадлежащее, и сказать, что язык $h(R)$ не является надъязыком G. Иначе п.2. Этот пункт является дополнительным и приносит 4 балла.
2. Построить грамматику D — пересечение R с языком ПСП из тех типов скобок, которые представлены в h, и применить к ней h. Вывести 100 самых коротких слов в $h(D)$ и сравнить их с 100 самыми короткими словами в G. Если множества не совпадают, предъявить слово-контрпример. Иначе сообщить, что неточностей в описании Шютценберже не найдено.
3. Для порождения слов ограниченной длины можно воспользоваться переходом к ХНФ: тогда каждое удлиняющее правило добавляет ровно 1 к длине слова, и 100 самых коротких слов будут соответствовать 100 самым коротким леволинейным выводам.

Бонусные баллы

- Самые быстрые реализации получат от +1 до бонусных +3 баллов.
- Если при этом будет строиться теоретический прообраз Шютценберже (по ХНФ) (т.е. соответствующее ему регулярное выражение и гомоморфизм), то реализация получит +3 балла.

Метод устранения левой рекурсии и метод Блюма–Коха

1. Результатом работы программы должна быть не только нормальная форма грамматики, но также диаграмма частичного порядка нетерминалов, построенная методом устранения левой рекурсии, и конечные автоматы, определяющие поведение сентенциальных форм в методе Блюма–Коха (после обращения).
2. +1 балл: подбирать частичный порядок таким образом, чтобы итоговая грамматика имела как можно меньше правил.
3. Ещё +2 балла: максимально упростить грамматику, используя эквивалентность по бисимуляции. Дальнейшие бонусы — по скорости преобразования.
4. Бонус за скорость: +1-2 балла.

ГНФ$\times$2

- Достаточно построить слабую форму Грейбах (Шейлы Грейбах! Не Грейбаха!).
- Правило подстановки в первом способе (для $A_i \to A_j\Phi$, где $i > j$) применяем уже после того как устранена левая рекурсия для A_j, иначе зациклимся.
- Автоматы сентенциальных форм строить нужно только для тех нетерминалов (плюс стартовый), которые входят в правила грамматики не только на первой позиции.
- Если в итоговой грамматике появились правила вида $A \to N_\gamma\Phi$, тогда нужно подставить вместо N_γ правые части правил переписывания для N_γ — по построению, они все будут начинаться с терминала.

Синтаксис грамматики

Чёрным обозначены элементы метаязыка, красным — элементы языка входных данных. Расстановка пробелов произвольна, могут встречаться табуляции, новая строка может начинаться с \n или с \r\n. Начальный нетерминал — [S].

Синтаксис входных данных КС-грамматики:

$$
\begin{aligned}
\langle \text{grammar} \rangle &::= \langle \text{rule} \rangle^{+} \\
\langle \text{rule} \rangle &::= \langle \text{nterm} \rangle \text{->} \; (\langle \text{term} \rangle \mid \langle \text{nterm} \rangle)^{*} \\
\langle \text{term} \rangle &::= [\text{A–z}] \mid [0\text{–}9] \\
\langle \text{nterm} \rangle &::= \text{[}[\text{A–z}]^{+}[0\text{–}9]^{*}\text{]}
\end{aligned}
$$

Расширение функционала Л.Р. 2

- (Первый вариант) Проверка регулярного языка на детерминированность (1-однозначность) методом орбит Брюггеманн–Вуда.
- (Второй вариант) Добавление метода порождения префиксной грамматики по НКА.
- (Третий вариант) Тест минимальности НКА по Глейстеру–Шаллиту (база — синтаксический моноид).
- (Четвёртый вариант) Расширение языка регулярных выражений оператором отрицания и встраивание его в существующие методы анализа регулярных выражений.
- (Пятый вариант)$\times$2 Подготовка документации по методам фреймворка и добавление функции help.
- (Шестой вариант) Добавление REPL-среды во фреймворк и добавление в интерфейс методов конкатенации, пересечения, итерации и объединения регулярных выражений и автоматов.