Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра математической
кибернетики и компьютерных наук

ПОДСЧЕТ ОБОБЩЕННОЙ ЗВЕЗДНОЙ ВЫСОТЫ РЕГУЛЯРНОГО ЯЗЫКА

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

Студентки 2 курса 273 группы
направления 01.04.02 — Прикладная математика и информатика
факультета КНиИТ
Хомяковой Екатерины Сергеевны

Научный руководитель
к. ф.-м. н. _____ С. В. Миронов

Заведующий кафедрой
к. ф.-м. н. _____ С. В. Миронов

Саратов 2017

ВВЕДЕНИЕ

Теория регулярных языков изучается уже долгое время, и многие проблемы, связанные с этими языками или соответствующими конечными автоматами, были решены. Однако, остается открытым сложный вопрос, а именно проблема определения звездной высоты регулярного языка. Эта проблема важна, потому что звездная высота представляет собой меру сложности регулярных языков, а минимальная звездная высота будет очень желательна в любом каноническом виде регулярных выражений [1].

Область практического применения регулярных языков и автоматов включает многие компьютерные приложения, такие как, например, текстовые редакторы и обработка текста, компиляторы и интерпретаторы командной строки. В частности, регулярные выражения, которые наравне с автоматами являются способом определения языка, принято использовать в автоматических генераторах лексических анализаторов в качестве формализма для задания классов однотипных лексем. Встроенная поддержка регулярных выражений имеется во многих языках программирования (например, Python, JavaScript, Ruby и пр.). Поэтому одной из актуальных задач является упрощение структуры конечных автоматов и регулярных выражений и понижение их сложности.

Цель работы. Исследование и реализация методов снижения обобщенной звездной высоты регулярного языка. Для достижения цели были поставлены следующие задачи:

- Реализация базовых возможностей для работы с конечными автоматами и регулярными выражениями, включающих в себя:
  - построение конечного автомата по регулярному выражению;
  - построение регулярного выражения по конечному автомату;
  - детерминирование и минимизацию конечных автоматов;
  - построение конечного автомата для языков, полученных при применении регулярных операций над произвольными регулярными языкам и пр.
- Разработка и реализация алгоритма поиска обобщенной звездной высоты регулярного языка.
- Реализация механизма для преобразования регулярных выражений с помощью применения правил.

Структура работы. Магистерская работа содержит 51 страницу (без

учета приложений) и состоит из введения, трех разделов («Теоретические сведения», «Проблема звездной высоты» и «Реализация алгоритмов»), заключения, списка использованных источников (22 наименования) и трех приложений, занимающих 15 страниц.

Научная новизна и практическая значимость. В магистерской работе реализован инструментарий для работы с регулярными языками, разработан и реализован алгоритм подсчета обобщенной звездной высоты. Полученные модули позволяют преобразовывать регулярные выражения и конечные автоматы, что может быть актуально как для теоретических исследований в области теории формальных языков, так и для программных средств, работающих с текстом.

Положения, выносимые на защиту. Изложены используемые подходы к решению задачи и примеры использования разработанных методов.

1 Основное содержание работы

Во введении обозначено направление исследований, актуальность рассматриваемой проблемы, а также цель работы и поставленные задачи.

В разделе 1 приводятся основные определения и утверждения, связанные с регулярными языками. Регулярный язык может быть представлен несколькими различными способами, например:

- как язык, принимаемый: детерминированным конечным автоматом (КА), недетерминированным КА (НКА) или НКА с ε-переходами [2, 3];
- как язык, заданный: регулярным выражением, допускающим операции объединения, конкатенации и замыкания Клини, или обобщенным регулярным выражением, допускающим дополнительные операции пересечения и дополнения [4];
- как язык, распознаваемый конечным моноидом [5, 6].

Далее приводятся основные теоретические сведения, описывающие каждый из способов представления. Подраздел 1.1 описывает алгебраический подход. Подраздел 1.2 содержит определение регулярных выражений, операции над регулярными языками и правила преобразования регулярных выражений. В подразделе 1.3 рассматриваются определения, связанные с конечными автоматами.

Раздел 2 посвящен проблеме звездной высоты регулярного языка.

Для некоторого конечного алфавита A звездная высота $h(E)$ регулярного выражения E определяется индуктивно следующим образом [7]:

1. $h(\varepsilon) = h(\varnothing) = h(a) = 0$ для всех $a \in A$;
2. $h(e + e') = h(ee') = \max\{h(e), h(e')\}$;
3. $h(e^*) = 1 + h(e)$.

Обобщенная звездная высота, т. е. высота для расширенных регулярных выражений, содержит дополнительные правила:

4. $h(e \cap e') = \max\{h(e), h(e')\}$;
5. $h(\overline{e}) = h(e)$ [8].

Звездная высота языка определяется как наименьшая высота регулярного выражения, определяющего язык. Проблема звездной высоты включает два вопроса: существуют ли языки произвольно большой высоты и есть ли алгоритм подсчета звездной высоты регулярного языка?

Далее в подразделах раздела 2 рассматриваются различные подходы к

решению проблемы и основные результаты. Подраздел 2.1 содержит описание истории возникновения и способов решения проблемы звездной высоты без операций дополнения и пересечения. Исходя из приведенных исследований делается вывод, что иерархия звездной высоты бесконечна, и алгоритмы для нахождения высоты существуют для нескольких семейств языков, тем не менее, в общем случае вопрос до сих пор открыт, т. е. для произвольного регулярного языка неизвестно, как найти его звездную высоту [8].

В подразделе 2.2 описываются основные результаты для проблемы обобщенной звездной высоты (в регулярных выражениях допустимы операции дополнения и пересечения). Приводится описание класса беззвездных языков, а также значимая для решения проблемы теорема, которая используется в работе для оптимизации алгоритма подсчета обобщенной звездной высоты.

Теорема Шютценберже. Пусть A — конечный алфавит и пусть $L \subseteq A^*$. Следующие условия эквивалентны:

1. Язык L беззвездный;
2. Синтаксический моноид для L конечен и апериодичен;
3. L распознается конечным апериодическим моноидом [9].

Так как апериодичность конечных моноидов разрешима, существует процедура решения для нулевой обобщенной звездной высоты. Но несмотря на это, основной вопрос остается открытым: конечна ли иерархия обобщенной звездной высоты? На данный момент нет ответа даже на более простой вопрос: есть ли язык обобщенной звездной высоты два [8]?

Раздел 3 содержит подробное описание реализации основных методов для работы с регулярными выражениями и конечными автоматами. В качестве языка программирования используется Python версии 3.4.

Подраздел 3.1 описывает применяемые структуры данных для представления конечных автоматов и регулярных выражений. Для представления автомата создан класс, который позволяет хранить и использовать основные составляющие автомата: алфавит, множество состояний, функцию переходов, множества начальных и заключительных состояний. Список состояний формируется из функции переходов, в то время как все остальные составляющие, представляющие собой множества, хранятся с использованием встроенной структуры данных `set`. Функция переходов в реализованном классе описывается с помощью словарей (`dict`), и может выглядеть следующим образом:

```
{
    {0: {'a': {2}, 'b': {1}}},
    {1: {'a': {2}}},
    {2: {'b': {0, 1}}},
}
```

В качестве ключей словаря используются номера состояний (из которых можно сформировать множество состояний автомата), а в качестве значений — соответствие меток перехода множеству состояний, в которое переходит состояние-ключ по заданному символу.

Регулярные выражения в реализованной программе задаются в виде строки. Регулярное выражение может включать операции объединения, конкатенации, итерации, положительной итерации, дополнения и пересечения. Заданная строка может быть разбита на токены и преобразована в соответствующий класс автомата, либо в класс, позволяющий проводить преобразования над регулярными выражениями с использованием правил. Основные методы работы с автоматами и выражениями объединены в класс для регулярного языка.

Для возможности работы с регулярными выражениями, был реализован способ их преобразования с помощью применения правил. Исходная строка с выражением представляется в виде вложенной структуры, в которой можно осуществлять поиск и замену подвыражений. В результате получается объект некоторого класса, представляющего символ алфавита или регулярную операцию, который в своих полях содержит аргументы операции, либо способ представления символа. Используемые классы содержат метод для сравнения объекта текущего класса с другим объектом, метод для формирования строкового представления объекта, а также метод для применения правила к регулярному выражению.

Для правил преобразования используется схожая структура классов, что и для регулярных выражений, за исключением возможности использования класса для произвольных подвыражений. Классы для правил содержат метод для проверки, подходит ли переданное выражение под шаблон правила, а также метод, возвращающий результат применения правила преобразования к регулярному выражению.

Подраздел 3.2 содержит подробное описание реализации алгоритма поиска обобщенной звездной высоты регулярного языка и необходимых для это-

го вспомогательных модулей для работы с конечными автоматами и регулярными выражениями. Также данный подраздел содержит примеры использования основных реализованных методов. Реализованный алгоритм рекурсивно разворачивает регулярные выражения с помощью операций дополнения, объединения и пересечения, поскольку при реализации переборного алгоритма необходимо рассмотреть как можно больше языков, из которых можно получить исходный, применяя регулярные операции.

Общую последовательность действий в алгоритме схематично можно представить следующим образом:

1. Получив на вход автомат $\mathcal{A}$, рекурсивно обработать автоматы $\mathcal{A}$ и $\overline{\mathcal{A}}$, перейдя к шагу 2, после чего вернуть наименьшую из высот, полученную для автоматов.
2. Если автомат содержит несколько заключительных состояний, то для каждого автомата с теми же состояниями, функцией переходов, начальными состояниями и с одним заключительным состоянием, выбранным из множества заключительных состояний исходного автомата, выполнить минимизацию и перейти либо к шагу 1, если полученный автомат не встречался ранее, либо к шагу 3, в противном случае. Для автомата с одним заключительным состоянием перейти к шагу 3.
3. С помощью метода исключения состояний построить регулярное выражение, удаляя состояния в оптимальном порядке, т. е., выбрав последовательность исключения состояний, при которой звездная высота будет наименьшей. Если регулярное выражение содержит подвыражение с итерацией, то построить для этого подвыражения минимальный автомат и перейти к шагу 1. Чтобы избежать зацикливание, регулярные выражения, которые уже обрабатывались ранее, записываются в базу данных вместе с полученной высотой. Таким образом, вычисления для каждого выражения производятся только один раз.

Шаг 3 алгоритма позволяет внедрять различные способы понижения высоты, одним из которых является поиск пар автоматов, пересечение языков которых равно языку исходного автомата, но имеет меньшую высоту, чем исходный язык. Для этого после нахождения наименьшей высоты с помощью метода исключения состояний используются два метода получения таких пар

автоматов: добавление новых заключительных состояний к исходному автомату и перенаправлении дуг исходного автомата из стока в другие состояния.

Для оптимизации приведенного алгоритма можно применить теорему Шютценберже о беззвездных языках. Для этого проверяется, является ли синтаксический моноид языка апериодическим (конечным он будет в любом случае, т. к. рассматриваются регулярные языки). Реализованные модули позволяют построить синтаксический моноид по минимальному автомату с помощью моноида переходов, после чего для всех элементов моноида x проверяется условие $x^n = x^{n+1}$ для n, равному количеству состояний автомата. Внедрение проверки на беззвездность на этапе вычисления звездной высоты регулярного выражения позволяет значительно сократить вычисления.

Подраздел 3.3 содержит подробное описание реализованных методов для преобразования регулярных выражений. Каждое правило преобразования представляет собой шаблон, с которым могут совпадать или не совпадать какие-либо подвыражения исходного регулярного выражения. Поэтому при реализации замены подвыражений на соответствующие выражения необходим, в первую очередь, механизм поиска подвыражений, попадающих под заданный шаблон с сохранением нужных аргументов операций. Для проверки, подходит ли подвыражение под заданный шаблон, у объектов соответствующих классов вызывается метод сравнения, который основан на проверке типа и атрибутов переданного объекта.

Метод сравнения класса для шаблона произвольных подвыражений не проверяет на соответствие какому-то определенному типу, но при вызове метода для сравнения, переданный аргумент сохраняется. Впоследствии, если метод сравнения был вызван у того же объекта класса (т. е. если в правиле есть условие, что подвыражения на определенных позициях должны совпадать), то переданный аргумент сравнивается с ранее переданным. Если они не совпадают, значит, выражение не подходит под шаблон. После проверки правила (и замены подвыражения, если правило может быть применено), сохраненные объекты стираются, чтобы не затрагивать последующие сравнения.

Таким образом, классы для преобразований содержат всю необходимую им информацию для того, чтобы сгенерировать новое регулярное выражение, используя подвыражения исходного выражения в качестве аргументов новых операций. Например, все классы для шаблонов, за исключением класса для

шаблона произвольных подвыражений, используют свои соответствия классам операций или символов, чтобы создать объект этого класса при замене подвыражения.

Далее в подразделе приводятся примеры применения правил преобразования к регулярным выражениям и различные способы использования реализованных методов.

ЗАКЛЮЧЕНИЕ

В работе были рассмотрены и реализованы основные методы и подходы для снижения обобщенной звездной высоты регулярного языка. В частности, реализованные модули позволяют использовать:

- базовые возможности для работы с конечными автоматами и регулярными выражениями, включающие:
  - построение конечного автомата по регулярному выражению;
  - построение регулярного выражения по конечному автомату;
  - детерминирование и минимизацию автомата;
  - построение конечного автомата для языков, полученных при применении регулярных операций над произвольными регулярными языкам и пр.
- алгоритм подсчета обобщенной звездной высоты регулярного языка;
- модуль для преобразования регулярных выражений с помощью применения правил.

Значимость работы определяется реализацией возможности снижать сложность регулярных выражений и конечных автоматов, что может быть актуально как для программных средств, используемых для обработки текста, так и во многих других сферах применения регулярных языков.

СПИСОК ОСНОВНЫХ ИСТОЧНИКОВ

1 *Cohen, R.* General properties of star height of regular events / R. Cohen, J. Brzozowski // *Journal of computer and system sciences.* — 1970. — Vol. 4. — Pp. 260–280.

2 Введение в теорию автоматов, языков и вычислений / Под ред. Дж. Хопкрофт, Р. Мотвани, Дж. Ульман. — Москва — Санкт-Петербург — Киев: Вильямс, 2002.

3 Теория синтаксического анализа, перевода и компиляции / Под ред. А. Ахо, Дж. Ульман. — Москва: Мир, 1978. — Т. 1.

4 *Ellul, K.* Regular expressions. new results and open problems / K. Ellul, B. Krawetz, J. Shallit, M. Wang // *Journal of Automata, Languages and Combinatorics.* — 2004. — Vol. 9. — Pp. 233–256.

5 *Myhill, J.* Finite automata and the representation of events / J. Myhill // *WADD TR-57-624, Wright Patterson AFB.* — 1957. — Pp. 112–137.

6 *Nerode, A.* Linear automata transformation / A. Nerode // *Proceedings of the American Mathematical Society.* — 1958. — Vol. 9. — Pp. 541–544.

7 *Hashiguchi, K.* Algorithms for determing relative star height and star height / K. Hashiguchi // *Inform. and Control.* — 1988. — Vol. 78. — Pp. 124–169.

8 *Brzozowski, J.* Open problems about regular languages / J. Brzozowski // *R.V. Book, ed., Formal Language Theory. Perspectives and Open Problems.* — 1980.

9 *Kufleitner, M.* Star-free languages and local divisors / M. Kufleitner // *CoRR.* — 2014. — Vol. abs/1408.2842.

05.06.17