Оглавление

[1. Введение 3](#_Toc326149023)

[2. Решение уравнений 5](#_Toc326149024)

[2.1. Свойства конкатенации 5](#_Toc326149025)

[2.2. Алгоритмы решения уравнений 6](#_Toc326149026)

[3. Решение систем 8](#_Toc326149027)

[3.1. Свойства возведения в степень 8](#_Toc326149028)

[3.2. Решение уравнений вида X \* Y = A 8](#_Toc326149029)

[3.3. Совмещение решений отдельных уравнений 9](#_Toc326149030)

[4. Пример работы алгоритма 11](#_Toc326149031)

[5. Направления дальнейшей работы 12](#_Toc326149032)

# Введение

*Унарный конечный автомат —* конечный автомат, входной алфавит которого состоит из одного символа. Другими словами, в унарном автомате при переходе между состояниями не может быть ветвлений. Так как кол-во состояний конечно, «топологически» такой автомат может принимать только одну форму, кольца с «хвостом»:

Рис. 1. Вид унарных конечных автоматов: а) кольцо с «хвостом»; б) вырожденная форма — простое кольцо. Нулем отмечено начальное состояние. Все стрелки означают переход по единственному символу. Какие-то состояния могут быть допускающими.

0

0

а

б

Также, принимаемые унарным автоматом строки можно отождествить с их длинами, ведь все они состоят только из одного символа.

С кольцами без «хвостов» удобно работать в алгебраических терминах: рассматривать допускающие состояния как вычеты по модулю длины кольца **N**. Тогда множество принимаемых автоматом длин равняется множеству допускающих состояний – вычетов по модулю **N**. Далее в зависимости от контекста «кольцо» может означать конкретный автомат или кольцо вычетов.

Автомат будет обозначаться так: , где — допускающие вычеты, **N**— модуль. Множество принимаемых длин обозначается **L(A)**,

*Конкатенация 2 автоматов —* автомат, принимающий язык, слова которого могут быть получены конкатенацией слов из языков первого и второго исходных автоматов соответственно. Обозначение: **C = A \* B**. При конкатенации унарных автоматов вместо «слов» и их «конкатенации» можно сразу рассматривать принимаемые длины и их суммы.

*«Возведение» автомата в степень* ***m*** *—* автомат, принимающий строки, полученные повторением принимаемых исходным автоматом строк **m** раз. Для колец:

**Цель этой курсовой работы** — изучение свойств операций конкатенации и возведения в степень применительно к унарным конечным автоматам и составление алгоритмов для приближенного решения уравнений вида и систем вида

где **X**, **Y** — неизвестные унарные конечные автоматы, **A** и **B** — кольца.

Решение таких систем (а также систем с большим количеством уравнений) может помочь оптимизировать работу виртуальной машины языка РЕФАЛ. (...)

Ввиду практического смысла (оптимизация, сокращение перебора) уравнений и систем над автоматами важно получить не точное решение, а автомат, язык которого *включает* язык точного решения. Такое послабление будет использоваться чтобы упростить алгоритмы и оставаться в множестве колец без «хвостов».

С другой стороны, нет смысла различать кольца, принимающие один язык. Поэтому далее на алгоритмическом уровне под «кольцами» (автоматами) понимаются классы эквивалентности (здесь и далее имеются ввиду классы эквивалентности по равенству языка):, в т. ч. для колец:

**На практике полезно сводить кольца к минимальным представителям их класса эквивалентности. Эта операция и результат называются *упрощением кольца***.

# Решение уравнений

## Свойства конкатенации

Операция конкатенации унарных автоматов коммутативна, потому что ее «база» — сложение целых чисел (длин) — коммутативная операция. Следовательно решения уравнений и совпадают.

Если **A** — унарный конечный автомат, — обозначение для кольца, получаемого механическим «отбрасыванием» «хвоста», начальным состоянием становится точка входа в кольцо. называется *собственным кольцом автомата* ***A***.

*Подкольцо кольца* ***A*** — кольцо, язык которого — подмножество языка кольца **A**.

*Свойство.* Результат конкатенации колец длин **N1** и **N2** — кольцо с «хвостом» длины не больше и модулем , его язык — подмножество языка его же собственного кольца.

*Доказательство.* Исходя из определения легко заметить, что конкатенация допускает «начальные» длины, полученные попарными сложениями базовых представителей вычетов исходных колец, которые могут произвольно удлиняться 2 модулями. Такое удлинение является подмножеством, а начиная с совпадает с удлинением одним модулем: :

*.*

*Следствие:* конкатенация колец с взаимнопростыми модулями **(A)** — цепочка с петлей на конце, т. е. .

Если **, то кольцо называется *сокращением A по модулю N (в m раз)*. A является подкольцом любого своего сокращения.**

***Следствие.* Если , то на место A и B можно подставить любого представителя классов эквивалентности и  соответственно без изменения результата.**

Свойство показывает, что операция конкатенации и решение уравнения (следовательно и системы) могут выводить за пределы множества простых колец. Нахождение точных решений выходит за рамки этой курсовой работы. Далее под «решением» уравнения  **понимается нахождения множества — собственных колец точных решений. Это имеет смысл, потому что кольцевая структура стабильна: . Языки колец из S *не* включают языки точных решений, поэтому формально для вывода предназначены автоматы с «хвостом» некоторой длины, в котором каждое состояние — допускающее, и найденными кольцами.**

## Алгоритмы решения уравнений

*Утверждение.* Если у уравнения есть решения, то это кольцо *(максимальное решение)* такой же длины, как и **B**, и, возможно, некоторые его подкольца.

*Доказательство.* Пусть **B** — минимальный представитель своего класса эквивалентности, длины **N**. Из свойства конкатенации следует: чтобы у уравнения были решения, длина кольца **A** и всех решений должна делиться на **N**. Пусть решением является кольцо . Но тогда  **— тоже решение, а C — его подкольцо. Если — решения модуля N**, решением является и объединение их вычетов по модулю **N**, подкольца колец  **и они сами также являются подкольцами объединения. Следовательно, любое решение является подкольцом единственного максимального решения.**

***Следствие.* Отдельные вычеты по модулю N, рассматриваемые как кольца, при конкатенации с кольцом A могут давать подкольцо B. Если объединение языков таких подколец равно языку B, максимальное решение равно объединению исходных вычетов по модулю N.**

Алгоритм нахождения максимального решения уравнения основан на следствии из утверждения о максимальном решении. На вход подается 2 кольца: левый или правый множитель **и результат конкатенации**. Выводится максимальное решение уравнения или сообщение о том, что решений нет.

1. Maximum\_Solution (**, ):**
2. **Упростить кольца A и B (обозначения cохраняются для удобства)**
3. if return
4. if **Решения нет**
5. for :
6. if :

9. if return
10. else **Решения нет**

Для решения систем помимо максимальных решений уравнений требуются условно «минимальные»: такие кольца модуля **N**, что ни одно из подколец, полученных исключением 1 вычета из списка допускающих, не является решением уравнения. Задача поиска минимальных решений не проще известной NP-сложной задачи о поиске минимального подпокрытия конечного множества.

1. Minimum\_Solutions (**, ):**
2. **Упростить кольца A и B**
3. if Maximum\_Solution :
4. return
5. //
6. for :
7. if and :
9. return

При реализации алгоритмов благодаря 2-му следствию из свойства конкатенации можно заменить кольцо **A** его сокращением по модулю **N** и работать не с автоматами, а напрямую с кольцом вычетов по модулю **N** и множествами на нем.

# Решение систем

План решения систем вида :

1. Решение уравнений и , т. е. нахождение множеств и : существуют решения уравнений и соответственно.
2. , аналогично для .
3. «Совмещение» решений 1-го и 2-го уравнений (подробнее ниже).

## Свойства возведения в степень

Все степени действуют аналогично, поэтому в этом разделе любое неизвестное кольцо на любой позиции будет обозначаться буквой **X**, **m** — степень, в которую возводится **X** на этой позиции, **N** — длина соотв. конкатенации.

При решении каждого уравнения с 2 неизвестными разумно сразу ввести ограничение: надо, чтобы найденные кольца **R** длины **N** могли быть получены возведением некоторого кольца **X** в степень **m**: . Это ограничение задается подмножеством вычетов в , которые могут присутствовать в **R**:

В этом случае кольца **X** будут иметь модуль . Возведение во взаимнопростую с модулем степень **m** приводит к «растяжению» и перестановке вычетов друг относительно друга.

Это словесное описание требуется оформить в виде вспомогательной функции, переводящей кольцо из «локальной» формы (решение уравнения) в «глобальную», общую для системы:

1. Log (**, ):**
2. return
3. //
4. //

Изложенная выше схема следует из свойств кольца вычетов и его мультипликативной группы.

## Решение уравнений вида **X \* Y = A**

Алгоритм реализует «спецификацию» из 1-го пункта в плане решения системы. Он опирается на утверждение о максимальном решении и его следствия.

1. Unbound\_Solutions (**, , ):**
3. for :
4. if Maximum\_Solution
5. and :
7. return

## Совмещение решений отдельных уравнений

*Утверждение.* Если  **и , то сокращение можно «проносить» через возведение в степень, т. е.**

*Доказательство:*

**Если , то кольцо называется *удлинением кольца A в m раз.***

***Утверждение.* Если для**

**то множество**

**Если же , построенное так кольцо удовлетворяет условиям (\*). Обе части утверждения — следствия свойств прямого произведения колец вычетов.**

**Понадобится функция, определяющая, совместимы ли 2 кольца.**

1. Compatible (, **):**
2. if and :
3. return
4. else return

Пусть уже получены *и переведены в «глобальную» форму* наборы решений и  (см. 2‑й пункт плана решения системы). Требуется найти такие пары

что

**Тогда пары будут решениями всей системы: это следствие утверждения о возведении в степень и сокращении.**

**Утверждение о пересечении колец дает простой способ проверить, совместимы ли левые решения. Если совместимы, нужно подобрать соотв. пару совместимых правых решений. По аналогии с утверждением о максимальном решении уравнения с одним неизвестным, будет существовать единственная «максимальная» пара, потому что если существует 2 совместимые пары, объединением вычетов колец из 1-го и 2-го уравнения получается также совместимая пара.**

**Но максимальная правая пара не обязательно состоит из максимальных решений «полуопределенных» уравнений (левые решения зафиксированы в пределах шага). Например, если** , , и — максимальные решения уравнений и соответственно, и совместимы, а и  — нет, т. е. **, или аналогично с , пока**

Minimum\_Solutions   
можно принять за «новый» и попробовать совместить и  снова.

Рассматривать подкольца минимальных правых решений смысла нет, потому что если не совместимы сокращения по модулям и , тем более не будут совместимы исходные кольца. Поэтому если на каком-то шаге или  меньше любого минимального решения своего уравнения, этап можно завершить и совмещение и в ответ не включается.

Все готово, чтобы записать итоговый алгоритм решения систем.

1. Solve\_System (**, ,, , , ):**
2. Unbound\_Solutions (**, , ),** Log (**, ))**
3. Unbound\_Solutions (**, , ),** Log (**, ))**
4. for , , if Compatible :
5. Log (Maximum\_Solution **, ,**
6. Log (Maximum\_Solution **, ,**
7. Minimum\_Solutions (**, ),** Log (**, ))**
8. Minimum\_Solutions (**, ),** Log (**, ))**
9. while not Compatible and and :
11. if :
13. if :


17. if Compatible :
19. return

# Пример работы алгоритма

С помощью алгоритма можно решить систему

(1)

В 1-м уравнении единичные степени идентичное действие функции Log. Левым решением может быть любое кольцо модуля 3 с 1 вычетом:

Во 2-м уравнении степень левого кольца сокращает кол-во возможных решений вдвое,

но на уровне системы множество совпадает со множеством .

В данном случае совпадают только равные левые решения. Если , то соответствующие правые максимальные (и единственные) решения совпадают с результатами конкатенации: , .

поэтому правые решения не совпадают.

Если , то , , — максимальные правые решения.

«сокращение» кольца модуля 6 до 6 — пустое, таким образом пара является решением системы.

Наконец, если , тогда , соответственно , — не совпадающие правые решения, поэтому левого решения в ответе не будет.

Итог: «максимальным» (также пары, состоящие из любых подколец, но в контексте практического смысла решения подобных систем это не интересно) решением системы (1) является пара пара . В конечном счете результатом будут 2 унарных автомата с «хвостами» длины 6, в которых каждое состояние — допускающее, и приведенными выше собственными кольцами.

# Направления дальнейшей работы

Совмещение решений очень просто обобщается на случай множества (больше 2) уравнений вместе с функциями и Compatible. А пункты 1 и 2 в плане решения систем касаются каждого уранения по отдельности, поэтому остаются независимыми. То есть, можно быстро доработать алгоритм для решения систем из многих уравнений.

«Приближение» ответов к точным решениям не кажется перспективным. Во-первых, точных решений систем в подавляющем большинстве случаев просто не существует, потому что в этом случае совпадение «хвостов» точных левых или правых решений (причем не в конечных алгебрах, а на ) разных уравнений будет чистой случайностью.

Во-вторых, предполагаемые текущим алгоритмом «хвосты», состоящие из допускающих состояний, не имеют тенденцию к удлинению, конкретнее, по длине они равны максимальному хвосту среди предполагаемых решениями отдельных уравнений (см. параграф 2.1).

В-третьих, специфика практического применения алгоритма в большом количестве уравнений и множестве простых результатов конкатенации, вплоть до . Учитывая предыдущий аргумент, в результате «лишних» принимаемых длин может быть пренебрежимо мало по сравнению с количеством строк «в основных» кольцах.

Нахождение менее тривиальных свойств конкатенации и возведения в степень применительно к кольцевой структуре может помочь существенно оптимизировать работу алгоритма, в т. ч. снизить максимальную асимптотическую сложность.

С одной стороны, при определенных входных данных (конкатенациях и степенях) некоторые блоки алгоритма могут гарантированно работать заранее известным образом. Например, в приведенном примере в cтроках 10 и 11 функции Solve\_System всегда возвращалось множество «минимальных» решений, состоящее из одного элемента — вычисленного буквально на предыдущем шаге максимального решения. Соответственно, «фильтрация» в строках 18 и 19, да и вообще весь цикл (строки 12–19) оказывались ненужными.

Кроме того, свойства кольцевой структуры, возможно, позволяют сократить полный перебор подмножеств в строке 3 функции Unbound\_Solutions и возможных левых решений в строке 7 функции Solve\_System.