



1. Основные понятия теории автоматов

Разделы:

- Введение
- Проблемы разрешимости и труднорешаемости
- Алфавиты, цепочки, языки
- Определения языков с помощью множеств

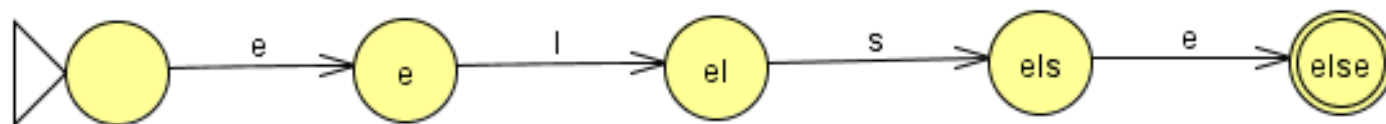
Введение

- КА являются моделью многих компонентов аппаратного и программного обеспечения
- Наиболее важные примеры использования КА:
 - ПО для верификации цифровых схем
 - Лексический анализатор типичного транслятора
 - ПО для сканирования Web-страниц для обеспечения поиска шаблонов
 - ПО для верификации сетевых протоколов и прочих систем, которые могут находиться в конечном числе различных состояний

Введение



Введение



Структурные представления

- Неавтоматные системы записи, играющие важную роль в теории автоматов и ее приложениях:
 - **Грамматики** как модели при проектировании ПО, обрабатывающего данные рекурсивной природы
 - Правила (продукции) вида $E \rightarrow E * E$
 - **Регулярные выражения** задают структуры данных, в том числе текстовые цепочки
 - $[A-Z][a-z]^*[] [A-Z][A-Z][A-Z]$ для названий городов и стран вида *Moscow RUS, Prague CZE, ...*
 - $([A-Z][a-z]^*[])^*[] [A-Z][A-Z][A-Z]$ для названий городов и стран вида *Nizhny Novgorod RUS*, в дополнение к предыдущему шаблону

Автоматы и сложность

- Проблемы, связанные с определением пределов вычислимости
 - Что **может** вычислительная машина?
 - Это **проблема разрешимости**, а задачи, которые могут быть решены на такой машине, называются **разрешимыми**
 - Что вычислитель **может** делать **эффективно**?
 - Это **проблема труднорешаемости** задач
 - Если на решение какой-то из них компьютеру требуется время, зависящее от размера входных данных как некая медленно растущая функция, то задача называется **легкоразрешимой**

Основные понятия теории автоматов

- **Алфавит** - конечное непустое множество символов
- Мы будем обозначать алфавиты символом Σ
- Примеры алфавитов:
 - $\Sigma = \{0,1\}$
 - $\Sigma = \{0,1,2\}$
 - $\Sigma = \{0,1,2,3,4,5,6,7,8,9\}$
 - $\Sigma = \{A,B,\dots,Z,a,b,\dots,z\}$
 - Σ = Все символы набора ASCII
 - Σ = Все печатаемые символы набора ASCII

Основные понятия теории автоматов

- **Цепочка (слово, строка)** – это конечная последовательность символов алфавита
- 01011 – это цепочка в двоичном алфавите
- Цепочка 00 также является цепочкой в этом алфавите
- Произвольную цепочку мы обозначаем символом w
- **Пустая цепочка** – это цепочка, не содержащая ни одного символа
- Ее обозначают символом ε
- Длину некоторой цепочки w принято обозначать $|w|$
- $|1011| = 4$, а $|\varepsilon| = 0$

Основные понятия теории автоматов

- Если Σ – некоторый алфавит, то множество всех строк определенной длины, состоящих из символов алфавита, выражается с использованием знака **степени**
- Определим Σ^k как множество всех строк длины k , состоящих из символов алфавита Σ , при $k \geq 0$
- Независимо от алфавита, $\Sigma^0 = \{\varepsilon\}$
- Значит, ε – это единственная цепочка длины 0
- Если $\Sigma = \{0,1\}$, то $\Sigma^1 = \{0,1\}$, $\Sigma^2 = \{00,01,10,11\}, \dots$

Основные понятия теории автоматов

- Множество всех цепочек над алфавитом Σ обозначим через Σ^*
 - $\{0, 1\}^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, \dots\}$
- Или эквивалентно,
$$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \dots$$
- Через Σ^+ принято обозначать множество всех непустых строк в алфавите Σ
- Тогда справедливы:
$$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \dots$$
$$\Sigma^* = \Sigma^+ \cup \{\varepsilon\}$$

Основные понятия теории автоматов

- Пусть x и y – строки в некотором алфавите
- Тогда xy – их **конкатенация**
- Результат — строка, в которой последовательно записаны элементы x и y
 - если x – строка из i символов ($x = a^1a^2...a^i$) и y – строка из j символов ($y = b^1b^2...b^j$),
 - то xy – это строка длины $i+j$ ($xy = a^1a^2...a^ib^1b^2...b^j$)
- Для любой строки w справедливы равенства $\varepsilon w = w\varepsilon = w$

Основные понятия теории автоматов

- Множество строк, каждая из которых принадлежит Σ^* , где Σ – некоторый алфавит, называется **формальным языком**
- Если Σ – алфавит и $L \subseteq \Sigma^*$
- то L – это **язык над**
- Если известно, что L является языком в Σ , то L – это язык над любым алфавитом, содержащим Σ

Основные понятия теории автоматов

- Примеры языков:

- $\{\varepsilon, 01, 0011, 000111, \dots\}$

- $\{\varepsilon, 01, 10, 0011, 1100, 1001, \dots\}$

- $\{01, 11, 101, 111, 1011, \dots\}$

- Σ^*

- \emptyset

- $\{\varepsilon\}$

Основные понятия теории автоматов

- Под **проблемой** понимается вопрос о том, является ли некоторая строка элементом некоторого конкретного языка
- Если Σ – алфавит, и L язык в Σ , то проблема L выглядит следующим образом
 - Дана строка w из Σ^* , требуется выяснить, принадлежит ли w языку L или нет
- Пример: задачу проверки заданного числа на простоту можно выразить в терминах принадлежности языку L_p , состоящего из двоичных строк, выражающих простые числа
- Цепочка 001101 не может представлять простое число по понятным причинам
- Решение данной проблемы для 1101 не очевидно

Основные понятия теории автоматов

- Языки часто задаются с помощью конструкций, принятых для описания множеств или параметризованного выражения:
 - $\{w \mid \text{сведения о } w\}$
- Примеры:
 - $\{w \mid w \text{ содержит одинаковое число } 0 \text{ и } 1\}$
 - $\{w \mid w \text{ является двоичным представлением простого числа}\}$
 - $\{w \mid w \text{ является синтаксически правильной программой на языке программирования высокого уровня}\}$
 - $\{0^n 1^n \mid n \geq 1\}$
 - $\{0^n 1^m \mid 0 \leq n \leq m\}$

Основные понятия теории автоматов

- На практике под проблемами понимаются не вопросы разрешения, а запросы на обработку или преобразование некоторого набора входных данных
- Задача анализатора в компиляторе языка C – определить, **принадлежит** ли данная строка символов множеству L_C корректных программ на C , отвечает определению проблемы
- Компилятор решает задачу **перевода** программы в объектный код для целевой вычислительной машины, а ответ на вопрос о правильности такой программы не лежит в плоскости «принадлежит»/ «не принадлежит»

Дополнительные источники

1. Машина Тьюринга// Лекция А.Шеня в проекте ПостНаука – Видео - <http://postnauka.ru/video/10777>
2. Кук, Д. Компьютерная математика / Д. Кук, Г.Бейз. – М.: Наука, 1990. – 384 с.
3. Молчанов, А. Ю. Системное программное обеспечение. 3-е изд. / А.Ю. Молчанов. – СПб.: Питер, 2010. – 400 с.
4. Stephen A. Cook: The Complexity of Theorem-Proving Procedures - <http://4mhz.de/cook.html>
5. Л. А. Левин Универсальные задачи перебора // Проблемы передачи информации. — 1973. — Т. 9. — № 3. — С. 115—116.
6. Теорема Кука-Левина – http://ru.wikipedia.org/wiki/Теорема_Кука
7. Кормен, Т.Х. Алгоритмы. Построение и анализ / Т.Х. Кормен, Ч.И. Лейзерсон, Р.Л. Ривест, К.Штайн. – М.: Вильямс, 2012. – 1296 с.
8. Aho, A.V. Foundations of Computer Science: C Edition (Principles of Computer Science Series) / A.V. Aho, J.D. Ullman. - New York: Computer Science Press, 1994. – 786 p.
9. Закон Мура – <http://cs.usu.edu.ru/study/moore/>