Курс «Алгоритмы и алгоритмические языки»
1 семестр 2014/2015

Лекции 1-2

Курс «Алгоритмы и алгоритмические языки»

Лектор Белеванцев Андрей Андреевич
кафедра СП

Лекции 2 раза в неделю: среда/суббота, 8.45

Практические и лабораторные занятия 2 раза в неделю

Структура курса:

- Элементы теории алгоритмов
- Язык Си
- Алгоритмы и структуры данных

В конце курса зачет с оценкой и письменный экзамен

Курс «Алгоритмы и алгоритмические языки»

Сайт курса: http://algcourse.cs.msu.su/

Новости и объявления

Материалы лекций

Вопросы к экзамену

Рекомендуемая литература

Среда разработки программ и опции компилятора

Стиль кодирования

Практические и лабораторные занятия

Контрольные и коллоквиумы
(по лекциям - 2, по семинарам - 3)

Анкета студента: заполните и верните лектору

Курс «Алгоритмы и алгоритмические языки»

Рекомендуемая литература:

<u>По алгоритмам и машинам Тьюринга</u>

1. Г. Эббинхауз, К. Якобс, Ф. Манн, Г. Хермес. Машины Тьюринга и рекурсивные функции. «Мир», М. – 1972
2. М.Минский. Вычисления и автоматы. «Мир», М. – 1971

<u>По языку Си</u>

1. Б. Керниган, Д. Ритчи. Язык программирования Си. Издание 2-е, «Вильямс» – 2013
2. Стандарт языка Си C99 + TC{1,2,3} http://www.open-std.org/JTC1/SC22/WG14/www/docs/n1256.pdf
3. Stephen Prata. C Primer Plus. Fifth Edition. Sams Publishing 2004. http://www.9wy.net/onlinebook/CPrimerPlus5/main.html
4. Г. Шилдт. Полный справочник по Си. Издание 4-е, «Вильямс» – 2010

<u>По алгоритмам и структурам данных</u>

1. Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. Алгоритмы. Построение и анализ. Издание 2-е, «Вильямс» – 2011
2. Harry R. Lewis, Larry Denenberg. Data Structures and Their Algorithms. HarperCollins, 1991.

Курс «Алгоритмы и алгоритмические языки»

Методические пособия:

1. А.А. Белеванцев, С.С. Гайсарян, Л.С. Корухова, Е.А. Кузьменкова, В.С. Махнычев. Семинары по курсу “Алгоритмы и алгоритмические языки” (учебно-методическое пособие для студентов 1 курса). М., 2012: Изд. отдел ф-та ВМК МГУ имени М.В. Ломоносова.
2. А.А. Белеванцев, С.С. Гайсарян, В.П. Иванников, Л.С. Корухова, В.А. Падарян. Задачи экзаменов по вводному курсу программирования (учебно-методическое пособие). М., 2012: Изд. отдел ф-та ВМК МГУ имени М.В. Ломоносова.
3. К.А. Батузов, А.А. Белеванцев, Р.А. Жуйков, А.О. Кудрявцев, В.А. Падарян, М.А. Соловьев. Практические задачи по вводному курсу программирования (методическое пособие). М., 2012: Изд. отдел ф-та ВМК МГУ имени М.В. Ломоносова.

Неформальное (интуитивное) определение алгоритма.

Под ***алгоритмом*** (или эффективной процедурой) в математике понимают ***точное предписание, задающее вычислительный процесс, ведущий от начальных данных***, *которые могут варьироваться,* ***к искомому результату***.

Алгоритм должен обладать следующими свойствами:

(1) ***Конечность*** (результативность). *Алгоритм должен заканчиваться за конечное* (хотя и не ограниченное сверху) *число шагов*.

(2) ***Определенность*** (детерминированность). Каждый шаг алгоритма и переход от шага к шагу должны быть точно определены и каждое применение алгоритма к одним и тем же исходным данным должно приводить к одинаковому результату.

Неформальное (интуитивное) определение алгоритма.

Под ***алгоритмом*** (или эффективной процедурой) в математике понимают
точное предписание, задающее вычислительный процесс, ведущий от начальных данных*, которые могут варьироваться,* ***к искомому результату***.
Алгоритм должен обладать следующими свойствами:

(3) ***Простота и понятность***. Каждый шаг алгоритма должен быть четко и ясно определен, чтобы выполнение алгоритма можно было «поручить» *любому* исполнителю (человеку или механическому устройству).

(4) ***Массовость***. Алгоритм задает процесс вычисления для множества исходных данных (чисел, строк букв и т.п.), он представляет общий метод решения класса задач.

Неформальное (интуитивное) определение алгоритма.

Пример: ***Алгоритм Евклида*** нахождения наибольшего общего делителя двух целых положительных чисел $НОД(a, b)$

(в геометрической форме это алгоритм нахождения общей меры двух отрезков).

Даны два целых числа a и b.

Выполнить следующие шаги:

(0) Если $a < b$, то поменять их местами.

(1) Разделить нацело a на b; получить остаток r.

(2) Если $r = 0$, то $НОД(a, b) = b$

(3) Если $r \neq 0$, заменить: a на b, b на r и вернуться к шагу (1).

***Почему необходимо формальное определение алгоритма*.**

Не имея такого определения, **невозможно** доказать, что задача алгоритмически неразрешима, т.е. алгоритм ее решения никогда не удастся построить.

***Тезис Тьюринга – Черча*.** Для любой интуитивно вычислимой функции существует вычисляющая её значения машина Тьюринга.

Тезис Тьюринга – Черча невозможно строго доказать или опровергнуть, так как он устанавливает эквивалентность между строго формализованным понятием частично вычислимой функции и неформальным понятием *вычислимости*.

Формализация понятия алгоритма

Алфавиты и отображения.

Алфавит $A_p = \{a_1, a_2, \ldots, a_p\}$.

Символы a_i

Слова $a_{i_1} a_{i_2} \ldots a_{i_m}$

Пустое слово ε

Множество всех слов над алфавитом A_p

$$A_p^* = \{\varepsilon\} \cup A_p \cup A_p^2 \cup \ldots \cup A_p^m \cup \ldots = \bigcup_{m=0}^{\infty} A_p^m$$

Длина слова w обозначается $|w|$, в частности $|\varepsilon| = 0$

Формализация понятия алгоритма

Кодирование

Утверждение. Для любой пары алфавитов A и B существует простой алгоритм, определяющий взаимно-однозначное отображение.

Следствие. Кодирование позволяет ограничиться одним алфавитом. Обычно рассматриваются A_1 или A_2.

Формализация понятия алгоритма

Обработка информации.

Задача обработки информации – это задача построения частичного отображения (функции)

$$F : A^* \to A^*$$

Утверждение. Существует взаимно-однозначное отображение (функция) $\#: A^* \to N$, где N – множество целых неотрицательных чисел, которое любому слову $w \in A^*$ ставит в соответствие его номер $n \in N$.

Формализация понятия алгоритма

Обработка информации.

$$
\begin{tikzcd}
A^* \arrow[r, "F"] \arrow[d, "\#"'] & A^* \\
N \arrow[r, "f"] & N \arrow[u, "\#^{-1}"']
\end{tikzcd}
$$

Таким образом:

(1) Каждый алгоритм определяет частично вычислимую функцию;

(2) Каждая частично вычислимая функция определяет алгоритм.

Машина Тьюринга (МТ)

Вычислимость по Тьюрингу

Машина-автомат: предъявляется исходное слово $w \in A^*$, в результате обработки получается слово $v = F(w)$.

Каждая частичная функция F, для которой можно построить МТ, называется *вычислимой по Тьюрингу*.

Машина Тьюринга (МТ)

Алфавит состояний $Q = \{q_0, q_1, q_2, \ldots, q_n\}$

Рабочий алфавит $S = A \cup A'$:

A – алфавит входных символов,

A' – алфавит вспомогательных символов (маркеров).

Лента, размеченная на ячейки (пустая ячейка - Λ)

Управляющая головка (УГ)

Рабочая ячейка (РЯ)

Начальное состояние q_0, состояние останова q_s.

Начальные данные – слова из A^*.

Машина Тьюринга (МТ)

$$\ldots\Lambda\Lambda\Lambda 00101110010\underset{\mathbf{q}}{\boxed{1}}1100000\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\ldots$$

Конфигурация МТ: $\langle n, \mathcal{F}, q\rangle$, где $\mathcal{F}: Z \to S$

Такт работы МТ:

$\langle$состояние, символ$\rangle \to \langle$состояние, символ, направление$\rangle$

Машина Тьюринга (МТ)

Пример. Проверка правильности скобочных выражений:
МТ должна записать на ленту для правильного скобочного выражения результат 1 (для неправильного 0) и остановиться.

Правильное скобочное выражение:

(1) число открывающих скобок равно числу закрывающих,

(2) каждая открывающая скобка предшествует парной ей закрывающей скобке.

(( ))( ) – правильное скобочное выражение

)(или (( ) – неправильные скобочные выражения

Машина Тьюринга (МТ)

Пример. Проверка правильности скобочных выражений:
МТ должна записать на ленту для правильного скобочного выражения результат 1 (для неправильного 0) и остановиться.

Рабочий алфавит: $S = \{(,), 0, 1\} \cup \{\Lambda, X\}$ (X – маркер)

Алфавит состояний $Q = \{q_0, q_1, q_2, q_3, q_s\}$:

- q_0 – начальное состояние МТ: поиск закрывающей скобки;
- q_s – состояние останова;
- q_1 – поиск парной открывающей скобки;
- q_2 – стирание маркеров, запись результата 1 и переход в q_s;
- q_3 – стирание маркеров, запись результата 0 и переход в q_s.

В начальном состоянии УГ обозревает самый левый символ входного слова

Машина Тьюринга (МТ)

Пример. Проверка правильности скобочных выражений:
МТ должна записать на ленту для правильного скобочного выражения результат 1 (для неправильного 0) и остановиться.

Программа

$\boldsymbol{q_0}$, **(** $\to q_0$, (, R;	$\boldsymbol{q_0}$, **)** $\to q_1, X, L$;	$\boldsymbol{q_0, X} \to q_0, X, R$;	$\boldsymbol{q_0, \Lambda} \to q_2, \Lambda, L$;
$\boldsymbol{q_1}$, **(** $\to q_0, X, R$;	$\boldsymbol{q_1}$, **)** $\to q_1$,), L;	$\boldsymbol{q_1, X} \to q_1, X, L$;	$\boldsymbol{q_1, \Lambda} \to q_3, \Lambda, R$;
$\boldsymbol{q_2}$, **(** $\to q_3, \Lambda, L$;	$\boldsymbol{q_2}$, **)** невозможна;	$\boldsymbol{q_2, X} \to q_2, \Lambda, L$;	$\boldsymbol{q_2, \Lambda} \to q_s, \boldsymbol{1}, H$;
$\boldsymbol{q_3}$, **(** $\to q_3, \Lambda, R$;	$\boldsymbol{q_3}$, **)** невозможна;	$\boldsymbol{q_3, X} \to q_3, \Lambda, R$;	$\boldsymbol{q_3, \Lambda} \to q_s, 0, H$;

Машина Тьюринга (МТ)

Пример. Проверка правильности скобочных выражений:
МТ должна записать на ленту для правильного скобочного выражения результат 1 (для неправильного 0) и остановиться.

Программа (другой способ записи)

$\boldsymbol{q_i} \downarrow \backslash\, \boldsymbol{s_j} \rightarrow$	(	)	X	Λ
q_0	q_0, (, R	q_1, X, L	q_0, X, R	q_2, Λ, L
q_1	q_0, X, R	q_1,), L	q_1, X, L	q_3, Λ, R
q_2	q_3, Λ, L	—	q_2, Λ, L	$q_s, 1, H$
q_3	q_3, Λ, R	—	q_3, Λ, R	$q_s, 0, H$

Машина Тьюринга (МТ)
Нормальные МТ.

Любую МТ можно перестроить таким образом, что она будет, вычисляя ту же функцию, удовлетворять следующим двум условиям:

(1) в начальном состоянии (q_0) УГ установлена напротив пустой ячейки, которая следует за всеми исходными символами:

$$\ldots\Lambda(()())((()))\underset{q_0}{\boxed{\Lambda}}\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\ldots$$

(2) в состоянии останова (q_s) УГ установлена напротив пустой ячейки, которая следует за всеми символами результата:

$$\ldots\Lambda 1\underset{q_s}{\boxed{\Lambda}}\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\Lambda\ldots$$

МТ, удовлетворяющая условиям (1) и (2), называется *нормальной* МТ.

Диаграммы Тьюринга (ДТ)

Перестройка МТ к виду, более удобному для ДТ

(1) МТ с лентой, ограниченной с левого конца

Для произвольной МТ ***T*** с неограниченной лентой построим МТ ***T'*** с лентой, ограниченной с левого конца, которая работает так же

…	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	…
…	Λ	Λ	Λ	**k**	$\boxed{\mathbf{a}}$	**b**	**c**	**d**	**a**	**f**	Λ	Λ	…
					q_0								

(а) перегнем ленту по ячейке с номером 0;

(б) раздвинем ячейки правой части ленты, помещая содержимое ячейки с номером n в ячейку с номером $2 \cdot n$

(в) в освободившиеся ячейки с нечетными номерами поместим содержимое ячеек левой части ленты, помещая содержимое ячейки с номером n в ячейку с номером $2 \cdot |n|-1$

Диаграммы Тьюринга (ДТ)

Перестройка МТ к виду, более удобному для ДТ

(1) МТ с лентой, ограниченной с левого конца

(а) перегнем ленту по ячейке с номером 0;

(б) раздвинем ячейки правой части ленты (с неотрицательными номерами), помещая содержимое ячейки с номером n в ячейку с номером $2 \cdot n$

(в) в освободившиеся ячейки с нечетными номерами поместим содержимое ячеек левой части ленты (с отрицательными номерами), помещая содержимое ячейки с номером n в ячейку с номером $2 \cdot |n| - 1$

	0	1	2	3	4	5	6	7	8	9	10	11	12	…
	0	-1	1	-2	2	-3	3	-4	4	-5	5	-6	6	…
|	**b**	$\boxed{\mathbf{a}}$	**c**	**k**	**d**	Λ	**a**	Λ	**f**	Λ	Λ	Λ	Λ	…
		q_0												

Диаграммы Тьюринга (ДТ)

Перестройка МТ к виду, более удобному для ДТ

(1) МТ с лентой, ограниченной с левого конца

T

…	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	…
…	Λ	Λ	Λ	**k**	$\boxed{\mathbf{a}}$	**b**	**c**	**d**	**a**	**f**	Λ	Λ	…
					q_0								

T′

	0	1	2	3	4	5	6	7	8	9	10	11	12	…
	0	-1	1	-2	2	-3	3	-4	4	-5	5	-6	6	…
край ленты |	**b**	$\boxed{\mathbf{a}}$	**c**	**k**	**d**	Λ	**a**	Λ	**f**	Λ	Λ	Λ	Λ	…
		q_0												

T	T′ (четные)	T′ (нечетные)	T′ (ячейка 0)	T′ (ячейка 1)
вправо	на две вправо	на две влево	на две вправо	на одну влево
влево	на две влево	на две вправо	на одну вправо	на две вправо

Диаграммы Тьюринга (ДТ)

Перестройка МТ к виду, более удобному для ДТ

(2) МТ с укороченными инструкциями.

Рассмотрим произвольную инструкцию МТ **T**:

$q, a \rightarrow q', b, R$;

Разобьем ее на две инструкции:

(1) $q, a \rightarrow q'', b, H$ (только записывает символ в РЯ);

(2) $q'', b \rightarrow q', b, R$ (только сдвигает головку).

Можно доказать, что для любой МТ **T** можно построить МТ **T'**, каждая инструкция которой либо только сдвигает головку, либо только записывает символ в РЯ.

МТ **T'** и есть МТ с укороченными инструкциями.

Диаграммы Тьюринга (ДТ)

Перестройка МТ к виду, более удобному для ДТ

Далее будем рассматривать класс МТ, который содержит только МТ с укороченными инструкциями, лентой, ограниченной слева, выполняющие нормальные вычисления по Тьюрингу.

Все эти предположения не являются ограничением общности, так как по произвольной МТ нетрудно построить МТ рассматриваемого класса.

Основным преимуществом рассматриваемого класса МТ является возможность ввести понятие ***действия.***

$$v_{ij}=\{L, R, H, s_i \in S\}$$

Диаграммы Тьюринга (ДТ)

Построение диаграмм Тьюринга

Запись символа в РЯ или сдвиг УГ вправо или влево называются ***элементарными действиями.***

МТ, выполняющие элементарные действия

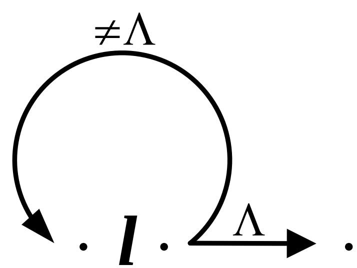
Элементарная МТ	Программа	Диаграмма
$\boldsymbol{l}$	$q_0\,\Lambda \to l\,q_1,\ q_0\,a_1 \to l\,q_1,\ \ldots,\ q_0\,a_p \to l\,q_1,$ q_1 – состояние останова*	$\cdot\boldsymbol{l}\cdot$
$\boldsymbol{r}$	$q_0\,\Lambda \to r\,q_1,\ldots,\ q_0\,a_p \to rq_1,$ q_1 – состояние останова*	$\cdot\boldsymbol{r}\cdot$
$\boldsymbol{a_i}$	$q_0\,\Lambda \to a_i\,q_1,\ q_0\,a_1 \to a_i\,q_1,\ \ldots,\ q_0\,a_p \to a_i\,q_1,$ q_1 – состояние останова*	$\cdot\boldsymbol{a_i}\cdot$

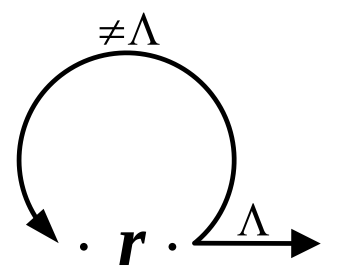
*Иногда пишут правила вида $q_1\,a_i \to hq_s$

Диаграммы Тьюринга (ДТ)

Построение диаграмм Тьюринга

Примеры не элементарных МТ.

$L =$ 

$R =$ 

МТ L переводит конфигурацию

$$[\Lambda\Lambda\ldots\Lambda a_1a_2a_3\ldots a_n\underset{q_0}{\boxed{\Lambda}}\Lambda\ldots$$

в конфигурацию

$$[\Lambda\Lambda\ldots\underset{q_1}{\boxed{\Lambda}}a_1a_2a_3\ldots a_n\Lambda\Lambda\ldots \qquad (1)$$

В дальнейшем слово на ленте $a_1a_2a_3\ldots a_n$ будем обозначать w, т.е. (1) будем записывать в виде $[\Lambda\Lambda\ldots\underset{q_1}{\boxed{\Lambda}}w\Lambda\Lambda\ldots$

Диаграммы Тьюринга (ДТ)

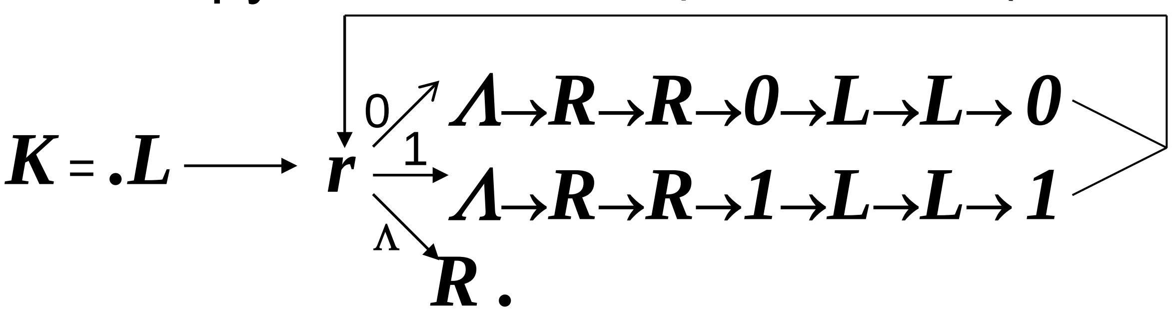
Построение диаграмм Тьюринга

МТ $\boldsymbol{K}$ переводит конфигурацию : в конфигурацию :

$$[\Lambda\ \ \Lambda\ \ \ldots\ \ \Lambda\ \ \mathtt{w}\ \boxed{\Lambda}\ \Lambda\ \ \ldots \qquad [\Lambda\ \ \Lambda\ \ \ldots\ \ \Lambda\ \ \mathtt{w}\ \ \Lambda\ \ \mathtt{w}\ \boxed{\Lambda}\ \Lambda\ \ \ldots$$

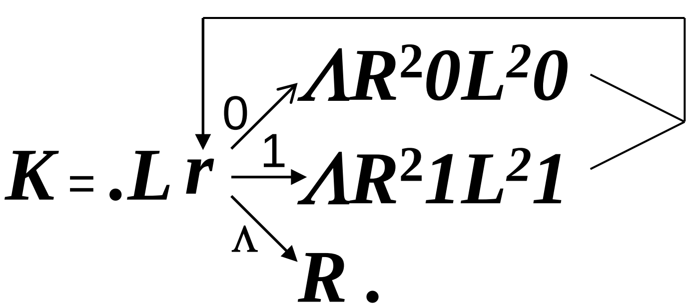
$$\mathtt{q_0} \qquad\qquad\qquad \mathtt{q_s}$$

т.е. копирует слово **w**. Диаграмма МТ $\boldsymbol{K}$ (над алфавитом {**0**,**1**})



Соглашение: стрелочки, над которыми ничего не надписано, опускаются.

Получаем *упрощенную диаграмму*:



Левая точка соответствует состоянию $\mathtt{q_0}$, правая – состоянию $\mathtt{q_s}$

В дальнейшем при построении новых МТ можно использовать диаграмму МТ $\boldsymbol{K}$