

# Теория программирования

*Бульонков Михаил Алексеевич, Филаткина Наталья Николаевна*

**Et.nsu.ru, дата размещения 7.11.2014**

## Аннотация

Дисциплина «Теория программирования» является частью математического цикла ООП по направлению подготовки «010200 – Математика и компьютерные науки» и «010400 – Прикладная математика и информатика». Дисциплина реализуется на Механико-математическом факультете Новосибирского государственного университета кафедрой Программирования ММФ НГУ.

Содержание дисциплины охватывает круг вопросов, связанных с анализом сложности алгоритмов для разных видов вычислительных моделей, анализом свойств моделей программ, а также с проблематикой смешанных вычислений.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, практические занятия, контрольная работа, реферат, самостоятельная работа студента.

Курс изучается в первом семестре, итоговая аттестация - экзамен.

## 1. Цели и задачи учебной дисциплины

Целью курса является освоение студентами базовых понятий теоретического программирования, которые, в свою очередь, необходимы для понимания современных методологий и технологий разработки программного обеспечения. Первая часть курса, в отличие от курса «Теория алгоритмов», где рассматриваются вопросы существования алгоритмов, посвящена элементам теории сложности, как средства для оценки существенных характеристик модельных вычислительных систем. Знакомство с различными моделями вычислений, их сравнительный анализ на основе моделирования даёт студентам представление о программной реализации вычислительной модели и их сложности. Методы оценки сложности демонстрируются на широко используемых алгоритмах поиска и сортировки, а также задаче умножения матрицы на вектор. Дается введение в машинно-независимую теорию сложности, которая позволяет оценить границы применимости понятий, связанных со сложностью алгоритмов.

Вторая часть курса также посвящена методам анализа программных моделей. Использование достаточно высокого уровня абстракции предоставляет единый механизм для определения разнообразных свойств программ и, в общем случае, динамических систем переходов, что позволяет распространить эти методы анализа на широкий класс задач. Студент получает представление о таких фундаментальных свойствах программы как поток управления, информационные зависимости, инварианты и т.п. Полезность этих понятий содержательно проявляется при определении систем преобразования программ, которая в свою очередь используются конструктивным доказательстве разрешимости логико-термальной эквивалентности.

Заключительная часть курса посвящена смешанным вычислениям программ, которые, с одной стороны, являются инструментом повышения эффективности программ и хорошим примером использования описанных ранее в курсе методов, с другой, – связующим звеном между основными подходами к реализации языков программирования: интерпретацией и трансляцией, а с третьей – связь с фундаментальными фактами, рассматриваемыми в теории вычислимости.

## 2. Содержание учебной дисциплины

Общая трудоемкость дисциплины составляет 4 зачетных единиц, 144 часа.

№ п/п	Раздел дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)					Формы текущего контроля успеваемости (по неделям семестра)  Форма промежуточной аттестации (по семестрам)
				Лекция	практич. занятия	Самост. работа	контр. работа	Экзамен	
1.1	Понятие машины Тьюринга. Запись программ на МТ. Функционирования МТ: конфигурация, протокол. Вычисляемая функция. Временная сложность. Ёмкостная сложность. Вариации МТ. Понятие РАМ-машины. Вычисление на РАМ: состояние памяти, конфигурация, операнды, команды. Функция, вычисляемая РАМ. Временная и ёмкостная сложность, равномерный и логарифмический весовой критерии. Понятие сложности в среднем.	3	1	2	2	2			
1.2	Понятие порядка сложности, полиномиальная связанность, экспоненциальная функция. Понятие моделирования, пошаговое моделирование. Моделирование РАМ на МТ, оценка сложности. Невозможность моделирование РАМ на МТ при равномерном весовом критерии. Моделирование РАМ на МТ при логарифмическом весовом критерии, оценка сложности.	3	2	2	2	2			

1.3	Теорема об ускорении. Понятие сигнализирующего оператора. Временная и ёмкостная сложность МТ как сигнализирующие. Машинно-независимая теория сложности. Теорема Цейтина, диагональный метод. Теорема Рабина.	3	3	2	2	2			
1.4	Нижние оценки. Задачи, допускающие матричные формулировки. Неветвящиеся программы, как модель вычислений. Теоремы о нижней оценке для задачи умножения матрицы на вектор по строкам и по столбцам. Примеры использования.	3	4	2	2	2			
1.5	Понятие конечного автомата. Графовое представление. Язык конечного автомата. Понятие регулярного выражения и его языка. Понятие регулярного выражения. Теорема о регулярности конечно-автоматных языков. Лемма о разрастании, примеры использования (существование нерегулярных языков).	3	5	2	2	2			
1.6	Проблемы пустоты и эквивалентности конечных автоматов. Переборный и алгебраический способы доказательства эквивалентности, оценка сложности. Понятие минимального автомата и его построение методом грубейшего разбиения. Оценка сложности. Доказательство эквивалентности методом сведения к задаче «Объединить-найти».	3	6	2	2	2			
1.7	Поиск в информационном массиве. Битовые шкалы: константное включение и поиск. Линейный поиск, оценка сложности в худшем и среднем. Бинарный поиск и двоичные деревья поиска. Сбалансированные деревья. 2-3 деревья: оценка сложности, процедуры вставки и удаления. В-деревья, как обобщение 2-3 деревьев: оценка сложности. АВЛ-деревья: оценка сложности, процедуры вставки в АВЛ дерево. Хэш-функции. Метод расстановки:	3	7	2	2	2			

	оценка сложности в худшем и среднем.								
1.8	Задача сортировки. Классификация методов сортировки: по области применения, по используемым методам. Критерии оценки сложности. Примеры сортировки вставкой, выбором и обменом. Понятие дерева решений. Оценка сложности в худшем случае для сортировки, основанной на сравнениях. Сортировка слиянием: оценка временной и ёмкостной сложности в худшем случае. Оценка сложности в среднем для сортировки, основанной на сравнениях. Быстрая сортировка: оценка временной сложности в среднем. Понятие цифровой сортировки.	3	8	2	2	2			
1.9	Метод разметки. Понятие ориентированного мультиграфа, примеры. Понятие полурешётки свойств, свойства ограниченности и обрыва цепей. Понятие монотонных и дистрибутивных преобразователей свойств. Формулировка задачи глобального анализа. Понятие точного решения ЗГА, Понятие недетерминированного процесса разметки. Понятие стационарной разметки, доказательство её существования. Понятие безопасной разметки. Теоремы о безопасности и единственности стационарной разметки. Оценка сложности алгоритмов ЗГА.	3	9	2	2	2			
1.10	Теорема о точности стационарной разметки в дистрибутивном случае. Пример формулировки ЗГА для анализа циклов в графе. Методы минимизации количества применений правила разметки. Теорема о невозможности построения точного решения в монотонном случае (путем сведения задачи анализа к построению системы Поста).	3	10	2		2	2		контрольная
2.1	Понятие стандартной схемы: базис, термы, операторы. Понятие аргументов и результатов, правильность стандартной схемы.	3	11	2	2	2			

	<p>Понятие интерпретации: значение терма, протокол, результат.</p> <p>Свободные интерпретации как пример интерпретации. Понятие функциональной эквивалентности стандартных схем. Понятие логико-термальной эквивалентности: логико-термальная история, детерминант. Теорема о корректности лт-эквивалентности.</p>							
3.1	<p>Понятие информационных связей и информационного графа.</p> <p>Нахождение информационного графа путём сведения к ЗГА.</p> <p>Компоненты связности информационного графа, их зацеплённость. Переименование переменных, минимизация количества переменных в стандартной схеме путём сведения к задаче раскраски графов.</p>	3	12	2	2	2		
3.2	<p>Понятие инвариантного соотношения. Функциональные сети, как средство представления инвариантных соотношений.</p> <p>Множество утверждений функциональной сети, приведённые сети. Операция пересечения функциональных сетей.</p> <p>Функциональные сети как полурешётка свойств.</p> <p>Преобразователи функциональных сетей, их дистрибутивность.</p> <p>Решение задачи нахождения инвариантных соотношений методом ЗГА.</p>	3	13	2	2	2		
3.3	<p>Понятие фрагмента стандартной схемы. Фрагмент, как обобщение стандартной схемы.</p> <p>Эквивалентность фрагментов, вхождения фрагментов. Понятие система преобразований. Система <math>\Sigma_{лт}</math>. Теорема о корректности.</p> <p>Теорема о полноте: согласование стандартных схем, Л-графы, Оценка сложности.</p>	3	14	2	2	2		
4.1	<p>Понятие смешанного вычисления программ: программы как данные, доступные и задержанные вычисления, основное уравнение СВ. Связь смешанных вычислений с S-m-n теоремой Клини. Условия</p>	3	15	2	2	2		

	выгодности СВ, Связь СВ и трансляции программ: понятие транслятора и интерпретатора. Проекция Фугамуры. Понятие метаинтерпретатора и автоинтерпретатора, критерий оптимальности Джонса. Другие потенциальные применения СВ.								
4.2	Пример функции $x^n$ . Интерпретационный подход к реализации СВ. Понятие генерирующего расширения, как объектного кода для семантики, заданной СВ, Трансформационный подход.	3	16	2	2	2			
4.3	Проблема задержанного управления при СВ, Поливариантные СВ, Пример интерпретатора конечных автоматов. Методы улучшения специализируемости программ. Построение транслятора конечных автоматов.	3	17	2	2	2			
4.4	Понятие анализа периода связывания (ВТА). Сведение задачи ВТА к ЗГА, Решение ВТА методом нахождения неподвижной точки. Решение ВТА методом решения системы неравенств. Понятие поливариантного анализа периода связывания: преобразование программы в процессе анализа.	3	18	2	2	2			
		3	19					36	Экзамен
				36	34	36	2	36	

### 3. Учебно-методические материалы дисциплины

а) основная литература:

1. Ахо А., Хопкрофт Дж., Ульман Дж. *Структуры данных и алгоритмы*: Пер. с англ. – М.: Издательский дом “Вильямс”, 2000.
2. Бульонков М.А. *Смешанные вычисления. Учебное пособие*. – Новосибирск: Изд-во НГУ, 1995.
3. Карпов Ю. *Теория автоматов. Учебник для вузов*. – СПб.: Издательский дом Питер, 2002.
4. Кормен Т., Лейзерсон Ч., Ривест Р. *Алгоритмы: построение и анализ*: Пер. с англ. – М.: МЦНМО, 1999.
5. Котов В.Е. *Введение в теорию схем программ*. – М.: Наука, 1978.
6. Котов В.Е., Сабельфельд В.К. *Теория схем программ*. – М.: Наука, 1991.
7. Лавров С. *Программирование. Математические основы, средства, теория*. – СПб.: БХВ-Петербург, 2001.

8. Мотвани Р., Хопкрофт Дж., Ульман Дж. *Введение в теорию автоматов, языков и вычислений*: Пер. с англ, 2-е издание. – М.: Издательский дом “Вильямс”, 2002.
9. Сабельфельд В.К. *Теория программирования. Учебное пособие*. – Новосибирск: Изд-во НГУ, 1993.
10. *Handbook of Theoretical Computer Science*. Volume A: Algorithms and Complexity. Volume B: Formal Models and Semantics / Edited by J. van Leeuwen. – Cambridge, MA: The MIT Press and – Amsterdam, New York, Oxford, Tokyo: Elsevier, 1996.

б) дополнительная литература:

1. Ахо А., Сети Р., Ульман Дж.Д. *Компиляторы: принципы, технологии и инструменты*: Пер. с англ. – М.: Издательский дом “Вильямс”, 2001.
2. Ахо А., Хопкрофт Дж., Ульман Дж. *Построение и анализ вычислительных алгоритмов*: Пер. с англ. – М.: Мир, 1979.
3. Барендрегт Х. *Лямбда-исчисление. Его синтаксис и семантика*: Пер. с англ. – М.: Мир, 1985.
4. Биркгоф Г. *Теория решеток*: Пер. с англ. – М.: Наука, 1984.
5. Грин Д., Кнут Д. *Математические методы анализа алгоритмов*: Пер. с англ. – М.: Мир, 1987.
6. Ершов А.П. *Введение в теоретическое программирование (беседы о методе)*. – М.: Наука, 1977.
7. Касьянов В.Н. *Лекции по теории формальных языков, автоматов и сложности вычислений. Учебное пособие*. – Новосибирск: Изд-во НГУ, 1995.
8. Катленд Н. *Вычислимость. Введение в теорию рекурсивных функций*: Пер. с англ. – М.: Мир, 1983.
9. Кнут Д. *Искусство программирования*: Пер. с англ. – М., СПб, Киев: Вильямс, 2000. – Т. 1-3.
10. Трахтенброт Б.А. *Сложность алгоритмов и вычислений*. – Новосибирск: Изд-во НГУ, 1967.
11. Hehnera E.C.R. *A Practical Theory of Programming. Second Edition*. – Toronto: University of Toronto, 2004. – url: [www.cs.utoronto.ca/~hehner/aPTOP](http://www.cs.utoronto.ca/~hehner/aPTOP).

#### 4. Контроль изучения дисциплины

а) Вопросы для экзамена

1. Машина Тьюринга. Конфигурация, протокол, временная и емкостная сложности. Моделирование машины Тьюринга на РАМ.
2. РАМ. Конфигурация, протокол, временная и емкостная сложности. Равномерный и логарифмический весовые критерии. Моделирование машины Тьюринга на РАМ.
3. Моделирование вычислений. Моделирование РАМ на машине Тьюринга.
4. Теорема об ускорении.
5. Теорема о существовании сколь угодно сложных функций (диагональ Рабина).
6. Теорема о нижней оценке количества умножений для вычисления  $Mx+y$
7. Конечные автоматы и регулярные выражения. Теорема об эквивалентности конечно-автоматных и регулярных языков.
8. Распознавание пустоты и эквивалентности конечных автоматов.
9. Задача поиска в информационном массиве. Деревья поиска, 2-3 деревья. Оценки сложности.

10. Задача поиска в информационном массиве. AVL-деревья. Оценки сложности.
11. Задача поиска в информационном массиве. Таблицы расстановки. Оценки сложности.
12. Сортировка массивов. Классификация алгоритмов. Быстрая сортировка. Оценка сложности.
13. Теоремы о сложности в среднем и в худшем сортировки, основанной на сравнениях.
14. Задача анализа свойств графа методом разметки. Существование, единственность и безопасность стационарной разметки.
15. Задача анализа свойств графа методом разметки. Возможность построения точного решения.
16. Понятие стандартной схемы. Интерпретация, функциональная эквивалентность.
17. Понятие стандартной схемы. Определение логико-термальной эквивалентности.
18. Функциональные сети для представления множества утверждений. Полурешетка.
19. Система преобразований  $S_{лт}$ . Полнота и корректность.
20. Рекурсивные схемы. Невозможность трансляции рекурсивных схем в стандартные.
21. Рекурсивные схемы. Алгоритм трансляции стандартных схем в рекурсивные схемы.
22. Многоленточные и многоголовочные конечные автоматы. Проблемы пустоты и эквивалентности.
23. Оптимальные деревья двоичного поиска
24. Порядковые статистики
25. Древоподобные структуры для задачи ОБЪЕДИНИТЬ-НАЙТИ.
26. Определение смешанных вычислений. Проекция Футамуры.
27. Интерпретативный, трансляционный и трансформационный подходы к смешанным вычислениям
28. Поливариантные смешанные вычисления.
29. Анализ периода связывания как ЗГА. Поливариантный анализ периода связывания.

б) Примеры задач, решаемых на занятиях.

1. Написать РАМ-программу для извлечения целой части корня квадратного из натурального числа без использования умножения и со сложностью  $O(\sqrt{n})$ .
2. В первой ячейке на входной ленте одноадресной РАМ машины задана размерность квадратной матрицы, а затем сама матрица по строкам. Написать программу, которая выводит на выходную ленту нечётные элементы побочной диагонали

матрицы. Например, для входной последовательности

2 3 4 7 8

результатом должно быть 4.

3. По данной РАМ машине построить РАМ машину, которая не использует ячейки с номерами с 1 по 10.
4. Являются ли классы машин Тьюринга с произвольным и бинарным алфавитами эквивалентными.
5. Промоделировать на РАМ машине машину Тьюринга.
6. Моделирование на РАМ-машине арифметического перехода с помощью логического и наоборот.
7. Построить регулярные выражения и конечные автоматы для следующих множеств:
  - а. идентификаторы Фортрана;
  - б. идентификаторы С.
8. Построить регулярное выражение и конечный автомат для множества цепочек из нулей и единиц, состоящих из четного числа нулей и четного числа единиц.
9. Доказать или опровергнуть два следующих утверждения: а) конечные автоматы могут складывать произвольные двоичные числа; б) конечные автоматы могут умножать произвольные двоичные числа. Входом автомата является цепочка в которой нечетные символы представляют биты первого числа, четные - второго. Выход – двоичная цепочка, являющаяся результатом. На любом такте работы автомата, в выход может быть записан либо один двоичный либо один пустой символ.
10. Доказательство теоремы Клини о замкнутости класса регулярных языков относительно операций объединения, пересечения, дополнения и конкатенации.
11. Является ли 2-скобочный язык Дика (язык всевозможных правильных скобочных цепочек, где скобками являются { } и [ ]) регулярным.
12. Оценка времени поиска в худшем в сбалансированных по высоте деревьях.
13. Построить алгоритм потокового анализа, вычисляющего для всякой вершины графа управления программы множество всех вершин графа, достижимых из этой вершины.
14. Стандартная схема задана в операторной форме:

```
start(x)
y:=x
1: if p(x) then goto 2
x:=f(x)
```

```

goto 1
2: if q(y,x) then goto 4
   if q(y,x) then goto 3
   goto 5
3: y:=f(y)
   goto 2
4: if q(y,x) then goto 5
   goto 3
5: stop(x)

```

Требуется построить для всех дуг инвариантные соотношения равенства термов.

15. Для заданных множеств равенств термов (необязательно транзитивно замкнутых относительно отношения равенства) построить замкнутое множество равенств термов, которое содержит те, и только те, равенства из исходных множеств, которые имеются в обоих множествах. Множество  $A = \{ y=f(z)=g(z,a)=g(z,x), z=f(a), x=a \}$ . Множество  $B = \{ f(x)=g(y,a)=f(y), x=y=f(z)=f(f(x)), z=a=f(x) \}$ .
16. Для заданного недетерминированного конечного автомата над алфавитом  $\{a,b\}$  построить эквивалентный детерминированный автомат:

```

1a -> 1
1a -> 2
1b -> 1
2a -> 3
2b -> 3
3a -> 2
3a -> 3
3b -> 1

```

Начальное состояние - 1, конечное - 2.

17. По заданному конечному автомату построить регулярное выражение, описывающее множество цепочек, допускаемых этим автоматом:

```

1a -> 2
1b -> 3
1c -> 1
2a -> a
2b -> 1
2c -> 4
3a -> 4

```

3b -> 4

3c -> 4

4a -> 3

4b -> 4

4c -> 4

Начальное состояние - 1, конечные - 3, 4.

18. По заданному регулярному выражению над алфавитом  $\{a,b,c\}$  построить конечный автомат без эpsilon-переходов, допускающий множество цепочек, описываемых данным регулярным выражением:

$((ab+aac)^*)(bac+bcc+(bac+bcc)^*)(cab+cba+abc)$ .

19. Расширим РАМ машину командами со стеком:

- a. PUSH - добавляет в стек значение сумматора
- b. POP – выталкивает значение из стека, засылая его в сумматор
- c. EMPTY – засылает в сумматор 1, если стек пуст, 0 – иначе.

Стек бесконечный, находится вне основной памяти. Доказать, что стековую РАМ машину можно моделировать на РАМ машине без стека.