

Модели вычислений.

Программа курса.

МИФИ (фак. кибернетики)

Автор: проф., д.т.н. Вольфенгаген В.Э., руководитель

Исполнители:

к.т.н., в.н.с. Исмаилова Л.Ю.,

с.н.с. Косиков С.В.

Курс рекомендуется студентам 3-6 курса и аспирантам.

По итогам курса сдается экзамен.

Лекций -- 34 час., семинарских занятий --17 час., лаб. работ –17 час.

1. Аннотация.

Курс моделей вычислений считается базовым для компьютерных наук и информационных технологий, а поиск новых моделей, изучение их свойств и выявление технологических преимуществ находятся на переднем крае научных исследований. Особое место отводится вычислениям с объектами и их связи с системами высших порядков. В начале рассматриваются вычисления на основе функций с оператором абстракции. Исходя из развитых представлений, изучается структура формальной системы комбинаторной логики и лямбда-исчисления. Показываются связи между возможными формами представления объектов, комбинаторная редукция, экспансия и конверсии. Подробно рассматривается задача синтеза объекта с заданной комбинаторной характеристикой.

Дается формулировка синтаксической теории вычислений, вводится представление о бестиповой системе. Изучаются комбинаторные базисы и их свойства, а также разложение произвольного объекта в базисах.

Доказывается теорема о неподвижной точке, рассматривается парадоксальный комбинатор Y и его применения к циклическим вычислениям. Рассматриваются приемы назначения типа объекту, исходя из структуры подобъектов.

Изучается метод погружения и рассматривается задача построения погруженной вычислительной системы. Изучается динамика вычислений, процесс компилирования комбинаторного кода. Рассматриваются механизмы вычислений, их параметризации и оптимизации.

Изучается вычисление значения объекта с применением декартово

замкнутой категории. Рассматривается строение категориальной абстрактной машины, реализация объектно-функциональных языков. Излагаются механизмы вычислений и вычисления в категории. На основе эквивалентных систем устанавливается взаимная сводимость различных форм теории вычислений. Рассматривается кодогенерация, даются доказательства оптимизационных равенств. Изучается цикл работы абстрактной машины, рекурсивная модификация среды вычислений. Даются примеры реализации в языках CAML и F#.

2. Основные цели курса.

Курс «Модели вычислений» показывает возможности семантической теории вычислений и дает представление о вычислении значения выражения, об основных приложениях к семантикам языков программирования, моделям объектов данных и языкам запросов, об установлении смысла вычисления значения в зависимости от среды вычислений. Охватываются вопросы использования лямбда-исчисления и комбинаторов. Демонстрируются возможности и преимущества комбинаторно полных теорий вычислений, в которых изучаются унифицированные представления выражений в комбинаторных базисах.

Курс развивает и формирует целостное представление о вычислениях с объектами и об их связи с системами высших порядков, дает знание структуры формальной системы комбинаторной логики и лямбда-исчисления, способствует овладению навыками применения форм представления объектов, комбинаторной редукции, экспансии и конверсии. Понятийная основа курса способствует развитию навыка выполнения исследований в области прикладного компьютерного программирования, а также овладению кругом идей наиболее актуальных прикладных вычислительных технологий и языков.

3. Место курса в системе профессиональной подготовки студента.

Для усвоения курса «Модели вычислений» желательно знакомство с формальными системами и элементами математической логики. Как минимум, необходимо владение представлением об объекте в информатике и о функции в анализе. Более глубокое изучение отдельных элементов курса достигается в дисциплинах: объектное программирование, теория типов, семантическое моделирование,

концептуальное моделирование и проектирование, модели данных и др.

4. Методы проведения занятий.

Читаются лекции 2 часа в неделю и проводятся семинарские занятия 1 час в неделю. На семинарских занятиях решаются задачи, а также рассматриваются дополнительные, не отраженные в лекциях, вопросы построения новых систем, методов и средств вычислений с объектами.

5. Формы контроля.

Предусмотрено 2-3 самостоятельных (контрольных) работы по группам, а также курсовая работа. Прием работы состоит в показе выполнения домашнего задания, беседы по теории и вопросов по ходу решения задач. Это позволяет контролировать как усвоение теоретического материала, так и уровень овладения практическим решением задач. Итоговым контролем является экзамен, включающий ответы на вопросы и решение задач. При определении итоговой оценки учитываются баллы, полученные студентами в семестре: за контрольные работы; за курсовую работу; за текущую работу в семестре, включая баллы за работу в семинаре; за выполнение домашних заданий.

6. Программа курса.

Курс состоит из 4-х разделов, которые объединяют 17 тем.

Разделы курса.

Раздел 1. Объекты, функции, абстракции

Тема 1. Вычисление значения.

- 1.1. Формальная система.
- 1.2. Выражения и означивания.
- 1.3. Определение объекта.
- 1.4. Индуктивные классы.
- 1.5. Вычисления без переменных.
- 1.6. Комбинаторы.
- 1.7. Операция абстракции.
- 1.8. Операция применения.

1.9. Операция связывания.

Тема 2. Объекты и вычисления с объектами.

- 2.1. Формальные и фактические параметры.
- 2.2. Передача параметров.
- 2.3. Подстановка.
- 2.4. Комбинаторная характеристика.
- 2.5. Системы постулатов. Правила вывода.
- 2.6. Отношения между объектами. Редукция, экспансия, конверсия.
- 2.7. Синтез объекта.

Тема 3. Связи между объектами.

- 3.1. Отображения.
- 3.2. неподвижные точки.
- 3.3. Теорема о неподвижной точке.
- 3.4. Представление циклов.
- 3.5. Рекурсия.
- 3.6. Структуры данных.

Раздел 2. Синтаксическая теория вычислений.

Тема 4. Системы типизации.

- 4.1. Представление о типе.
- 4.2. Приписывание типа.
- 4.3. Содержательная интерпретация.
- 4.4. Типизированное исчисление комбинаторов.
- 4.5. Типизированное исчисление абстракций.
- 4.6. Исходные типы.
- 4.7. Дедуктивные системы и вывод производного типа.
- 4.8. Типы высших порядков.
- 4.9. Функциональные пространства.

Тема 5. Решение задачи синтеза структуры данных.

- 5.1. Эквационная формулировка.
- 5.2. Итеративные уточнения.
- 5.3. Синтез операторов/функций.
- 5.4. Полнота.
- 5.5. Усиление выразительных возможностей.
- 5.6. Решение задачи погружения.

Тема 6. Базисы.

- 6.1. Определение базиса.
- 6.2. Свойство базисности.
- 6.3. Фиксированные базисы. Примеры.
- 6.4. Решение задачи разложения объекта в базисе. Границы применимости метода.
- 6.5. Нумералы. Комбинаторная арифметика.

Раздел 3. Динамика вычислений.

Тема 7. Динамические базисы.

- 7.1. Понятие суперкомбинатора.
- 7.2. Процесс компиляции, основанный на суперкомбинаторах.
- 7.3. Сведение абстракций к суперкомбинаторам.
- 7.4. Алгоритм подъема абстракции.

Тема 8. Использование параметров.

- 8.1. Устранение избыточных параметров.
- 8.2. Упорядочивание параметров.

Тема 9. Использование параметров (продолжение).

- 9.1. Модифицированные алгоритмы подъема.
- 9.2. Подъем при рекурсии.

Тема 10. Подвыражения.

- 10.1. Максимально свободные выражения.
- 10.2. Свойство ленивости вычисления.

Тема 11. Оптимизации.

Раздел 4. Абстрактные машины.

Тема 12. Механизмы вычислений.

- 12.1. Семантика вычислений.
- 12.2. Вычисления значения.
- 12.3. Вызов параметра.
- 12.4. Системы вычисления значений.
- 12.5. Коллизии переменных.

12.6. Вычисления в категории.

Тема 13. Теории вычислений.

13.1. Эквиационные системы вычисления значения.

Тема 14. Теории вычислений (продолжение).

14.1. Оптимизационные равенства.

Тема 15. Цикл работы категориальной абстрактной машины (КАМ).

15.1. Система инструкций.

15.2. Состояние вычисления.

15.3. Структура КАМ.

15.4. Оптимизация исполнения кода.

Тема 16. Циклические вычисления.

16.1. Рекурсивная модификация среды.

16.2. Примеры КАМ-программ.

Тема 17. CAML, F# и примеры программ.

17.1. Реализации аппликативных языков.

17.2. Примеры программ.

7. Краткое содержание тем (лекций)

Раздел 1. Объекты, функции, абстракции

В результате изучения раздела 1 нужно:

- иметь представление о вычислении значения выражений, объектах и вычислениях с объектами, связях между возможными формами представления объектов.
- усвоить основы комбинаторной редукции и начала лямбда-конверсии;

знать:

- постановку и решение задачи синтеза объекта с заданной комбинаторной характеристикой;
- подход к анализу цикличности вычислений на основе неподвижной точки.

уметь:

- синтезировать и анализировать объект с заданной комбинаторной характеристикой;
- производить вычисление (интерпретацию) комбинаторного программного кода, содержащего конструкции цикла.

Тема 1. Вычисление значения. Формальная система. Выражения и означивания. Определение объекта. Индуктивные классы. Вычисления без переменных. Комбинаторы. Операция абстракции. Операция применения. Операция связывания.

Учебная задача. Изучается строение формальной системы, роль и место термов и формул. На основе структурной индукции изучается построение значимой системы объектов. Рассматривается постановка и решение основной задачи комбинаторной логики, формулируемой как задача синтеза объекта с заданными свойствами из имеющихся объектов применением уже известных способов комбинирования.

Обзор темы. На начальном этапе предполагается наличие всего трех объектов-комбинаторов: I, K, S, а также их свойств, задаваемых характеристическими равенствами. Предполагается, что имеется система программирования с этими тремя инструкциями, пользуясь исключительно которыми предстоит построить довольно богатую по выразительным возможностям систему программирования. Результирующая система будет содержать исключительно объекты-комбинаторы.

В результате изучения темы 1 нужно:
знать:

- постановку и решение задачи синтеза объекта с заданной комбинаторной характеристикой;

уметь:

- синтезировать и анализировать объект с заданной комбинаторной характеристикой.

Тема 2. Объекты и вычисления с объектами. Формальные и фактические параметры. Передача параметров. Подстановка. Комбинаторная характеристика. Системы постулатов. Правила вывода. Отношения между объектами. Редукция, экспансия, конверсия. Синтез объекта.

Учебная задача. Изучается техника аппликативных вычислений, причем выделяется центральная идея вычисления -- замещение формального параметра на фактический. Изучаются приемы определения числа существенных параметров, пользуясь комбинаторной характеристикой объекта. Показывается строение системы постулатов, задающих отношения на классе объектов.

Обзор темы. Передача параметров рассматривается для изучения применения оператора функциональной абстракции. Характеристические свойства подстановки выводятся структурной индукцией по построению объекта. Обсуждаются отношения между объектами, взаимные переходы между формами представления объектов, их нормальные формы. Формулируется теорема Черча-Россера о конвертируемости объектов с учетом их нормальной формы. Рассматривается решение задачи синтеза объекта с заданной комбинаторной/вычислительной характеристикой.

В результате изучения темы 1 нужно:
знать:

- применение определения подстановки структурной индукцией по сложности объекта;

уметь:

- выполнять вычисления применением правила подстановки с учетом переименования связанных переменных.

Тема 3. Связи между объектами. Отображения. Неподвижные точки.
Теорема о неподвижной точке. Представление циклов. Рекурсия.
Структуры данных.

Учебная задача. Изучаются индуктивные классы объектов в комбинаторной логике. Вырабатывается подход, как, пользуясь простыми и конечными по своей природе объектами, представлять циклические процессы. Изучается, как, пользуясь комбинаторами, можно представлять процессы, в том числе циклические вычисления, которые представляют известную в программировании работу со стеком рекурсии.

Обзор темы. Формулируется определение неподвижной точки. Вводится парадоксальный комбинатор Y , пользуясь которым можно отыскивать неподвижную точку отображения. Формулируется теорема о неподвижной точке, пользуясь которой рекурсивные определения

преобразуются к стандартному виду.

В результате изучения темы 3 нужно:

знать:

- подход к анализу цикличности вычислений на основе неподвижной точки.

уметь:

- производить вычисление (интерпретацию) комбинаторного программного кода, содержащего конструкции цикла.

Раздел 2. Синтаксическая теория вычислений.

В результате изучения раздела 2 нужно:

- иметь представление о построении систем типизации;
- усвоить методы и технические приемы вывода типа объекту, исходя из заданного приписывания типов подобъектам;

знать:

- представление постановки и решения задачи синтеза структуры данных с заданными математическими свойствами;
- связь различных формулировок теории вычислений;
- различные варианты погружения теорий объектов;

уметь:

- устанавливать комбинаторный базис вычислений и применять его для решения задачи компилирования комбинаторного кода;
- строить эквациональные представления вычислений.

Тема 4. Системы типизации. Представление о типе. Приписывание типа. Содержательная интерпретация. Типизированное исчисление комбинаторов. Типизированное исчисление абстракций. Исходные типы. Дедуктивные системы и вывод производного типа. Типы высших порядков. Функциональные пространства.

Учебная задача. Изучается концепция класса, которая является одной из самых основных в объектно-ориентированных рассуждениях. Формируется подход к построению функциональных пространств высших порядков.

Обзор темы. Класс понимается как образец для создания экземпляров

конкретных объектов. Более того, классы рассматриваются как объекты. Точно также комбинаторы классифицируются, или типизируются. Существенным для комбинаторов оказывается высокий порядок функциональных пространств. Тем не менее, интуитивная ясность работы с комбинаторами как с объектами не теряется.

В результате изучения темы 4 нужно:

знать:

- установление типа объекта с оператором абстракции;
- установление типа объекта, построенного с применением комбинаторов;

уметь:

- давать конструктивное заключение о возможности или невозможности установления типа для объекта формальной системы;
- доказывать, почему тип выражения нельзя установить и давать содержательную интерпретации.

Тема 5. Решение задачи синтеза структуры данных. Эквациональная формулировка. Итеративные уточнения. Синтез операторов/функций. Полнота. Усиление выразительных возможностей. Решение задачи погружения.

Учебная задача. Изучается метод погружения. Вырабатывается подход к установлению подходящего представления содержательного объекта предметной области. Изучаются выразительные возможности комбинаторной логики, ее связь с лямбда-исчислением. Изучается, как построить предформализацию, которая выражает свойства содержательных объектов, а также ее перевод в эквациональную форму.

Обзор темы. Показывается, как ставить и решать задачу построения погруженной (встроенной) вычислительной системы. Рассматривается значимый пример синтеза вычислительной системы, представляющей средства обработки конечных последовательностей (списков).

Основные знания и умения.

В результате изучения темы 5 нужно:

знать:

- формулировку постановки задачи синтеза вычислительной системы с заданными свойствами в общем виде;

уметь:

- применять формулировку постановки задачи синтеза вычислительной системы с заданными свойствами в частных случаях;
- строить системы вычислений с конечными последовательностями (списками).

Тема 6. Базисы. Определение базиса. Свойство базисности.

Фиксированные базисы. Примеры. Решение задачи разложения объекта в базисе. Границы применимости метода. Нумералы. Комбинаторная арифметика.

Учебная задача. Изучаются вычислительные возможности наипростейшей системе программирования, в которой всего только три инструкции: I, K, S. Ставится и решается задача синтеза нового объекта чисто механическим использованием алгоритма разложения в базисе, который вполне аналогичен процессу компиляции. Показывается, что базис I, K, S не единственный, и свойство базисности проявляет также набор комбинаторов I, B, C, S. Изучаются приемы представления чисел и арифметических операций объектами комбинаторной логики и лямбда-исчисления.

Обзор темы. Вырабатывается представление о вычислительном базисе. Показывается, что компиляция (разложение) объекта в этом базисе также решает задачу синтеза объекта с заданными свойствами. Как оказывается, можно использовать свободу выбора базиса в зависимости от некоторых критериев. Поскольку в аппликативных вычислительных системах среди первичных объектов нет чисел, то строится их комбинаторное представление и соответствующая комбинаторная арифметика.

В результате изучения темы 6 нужно:

знать:

- формулировку постановки задачи синтеза объекта в заданном комбинаторном базисе;

- возможности комбинаторной арифметики;

уметь:

- применять алгоритм разложения объектов с заданными свойствами для комбинаторных базисов в частных случаях;

- выполнять вычисления в комбинаторной арифметике.

Раздел 3. Динамика вычислений.

В результате изучения раздела 3 нужно:

- иметь представление о статических и динамических комбинаторных базисах;
- усвоить подходы и приемы повышения эффективности вычислений;

знать:

- процесс компилирования комбинаторного кода;
- связь синтаксиса и семантики вычислений с избранными базисами;
- различные механизмы вычислений и пути их усовершенствования посредством различных параметризаций.

уметь:

- выполнять приведение абстракции к суперкомбинаторам;
- производить вычисление (интерпретацию) редуцированного выражения;
- оптимизировать вычисления, применяя параметризации.

Тема 7. Динамические базисы. Понятие суперкомбинатора. Процесс компиляции, основанный на суперкомбинаторах. Сведение абстракций к суперкомбинаторам. Алгоритм подъема абстракции.

Учебная задача. Изучается работа объектно-ориентированных систем, встроенных в комбинаторную логику. Показывается, как непосредственно удовлетворяется потребность в денотационном вычислении инструкций языков программирования, когда объектами выражается функциональный смысл программы. На основе определения суперкомбинатора устанавливается процедура преобразования абстракции и соответствующий алгоритм подъема абстракции.

Обзор темы. Вычисление начинается с некоторого заранее известного набора инструкций. Показывается, что в процессе вычисления значения программы динамически возникают заранее неизвестные, но необходимые по ходу дела инструкции, которые дополнительно фиксируются в системе программирования. Обсуждаются

сопутствующие этому процессу алгоритмы.

В результате изучения темы 7 нужно:

знать:

- процесс компилирования комбинаторного кода;

уметь:

- выполнять приведение абстракции к суперкомбинаторам.

Тема 8-9. Использование параметров. Устранение избыточных параметров. Упорядочивание параметров. Модифицированные алгоритмы подъема. Подъем при рекурсии.

Учебная задача. Изучаются различные виды параметров, приемы установления их избыточности. Рассматривается, как при необходимости модифицировать алгоритмы подъема, а также приемы обработки рекурсии.

Обзор темы. Показывается, как, используя различные критерии, определить избыточные параметры. Рассматривается применение алгоритмов подъема, в особенности при наличии рекурсии.

В результате изучения тем 8-9 нужно:

знать:

- различные механизмы вычислений и пути их

усовершенствования посредством различных параметризаций;

уметь:

- выполнять приведение абстракции к суперкомбинаторам;

- производить вычисление (интерпретацию) редуцированного

выражения.

Тема 10-11. Подвыражения. Максимально свободные выражения.

Свойство ленивости вычисления. Оптимизации.

Учебная задача. Изучаются вычислительные эффекты динамического формирования объектов “на лету”, когда эффективность результирующего кода может теряться из-за необходимости неоднократно вычислять значение одного и того же объекта. Выполняется исследование применения механизмов ленивого означивания, которые позволяют этого избежать: если значение объекта

уже однократно вычислено, то в дальнейшем используется именно это заранее вычисленное значение.

Изучаются приемы применения комбинаторов, которые открывают возможности строить оптимизированный программный код. При этом попутно, в ходе синтеза результирующего объекта, анализируется порядок возможного замещения формальных параметров на фактические.

Обзор темы. Водится представление о максимально свободных выражениях, которые отличаются относительной устойчивостью и относительной независимостью от остальной части программного кода. Рассматриваются способы повышения эффективности вычислений за счет снижения объема ненужных предвычислений. В этих условиях ставится и решается задача оптимизации вычислений.

В результате изучения темы 10-11 нужно:

знать:

- подходы и приемы повышения эффективности вычислений;
- различные механизмы вычислений и пути их

усовершенствования посредством различных параметризаций.

уметь:

- производить вычисление сформированного выражения, принимая меры по обеспечению эффективности этого процесса;
- оптимизировать вычисления, применяя параметризацию.

Раздел 4. Абстрактные машины.

В результате изучения раздела 4 нужно:

- иметь представление о состоянии исследований в области разработки механизмов вычислений;
- усвоить подход к анализу программного кода, основанный на абстрактной машине вычислений;

знать:

- пути и методы устранения коллизий переменных;
- различные формы, в том числе эквациональные, теории

вычислений;

- цикл работы абстрактной машины;

уметь:

- выполнять кодогенерацию исходного выражения в промежуточное представление;
- оптимизировать и исполнять сгенерированный код на основе инструкций абстрактной машины;
- выполнять вычисления, включающие неподвижную точку.

Тема 12. Механизмы вычислений. Семантика вычислений. Вычисления значения. Вызов параметра. Системы вычисления значений. Коллизии переменных. Вычисления в категории.

Учебная задача. Техника вычисления значения выражений пересматривается -- в свете систематического построения набора синтактико-семантических равенств, реализующих избранную парадигму объектно-ориентированных вычислений. Исследуется и изучается организация вычисления выражений в категории.

Обзор темы. Излагается связь построения семантики выражений и решения задачи вычисления значения выражений. Рассматриваются различные механизмы вычислений, применение для них формальных систем. Устанавливается необходимость систематического переименования связанных переменных, что позволяет устранить коллизии их связывания. Строится категория, которая является объемлющей средой вычислений.

В результате изучения темы 12 нужно:
знать:

- пути и методы устранения коллизий переменных;

уметь:

- выполнять кодогенерацию исходного выражения в промежуточное представление.

Тема 13-14. Теории вычислений. Эквациональные системы вычисления значения. Оптимизационные равенства.

Учебная задача. Вводится в рассмотрение техника переобозначения связанных переменных (формальных параметров), которая позволяет избежать коллизий связывания при замещении формальных параметров на фактические. Это прием переобозначения носит название *кодирования по де Брейну* и позволяет, фактически, аппаратом лямбда-исчисления

пользоваться на тех же самых правах, что и аппаратом комбинаторной логики. Изучаются наборы равенств, выражающих необходимый объем свойств для получения значения выражений. Изучается подход к построению равенств, позволяющих оптимизировать вычисления и обеспечить экономию в кодировании.

Обзор темы. Рассматриваются различные эквивалентные формулировки теории вычислений, каждая из которых характеризуется специальным кругом тех задач, которые решаются наилучшим образом. Вводятся и обосновываются равенства, которые являются следствиями теории вычислений и применимы для оптимизации вычислений и/или обеспечения экономии в кодировании выражений.

В результате изучения темы 13-14 нужно:

знать:

- различные формы, в том числе эквивалентные, теории вычислений;

уметь:

- оптимизировать и исполнять сгенерированный код на основе инструкций абстрактной машины.

Тема 15. Цикл работы категориальной абстрактной машины (КАМ). Система инструкций. Состояние вычисления. Структура КАМ. Оптимизация исполнения кода.

Учебная задача. Изучается метод построения специального варианта теории вычислений, называемого *категориальной абстрактной машиной* (КАМ). Для этого вводится в рассмотрение специальный фрагмент комбинаторной логики -- категориальная комбинаторная логика. Она представлена набором комбинаторов, каждый из которых имеет самостоятельное значение как инструкция системы программирования. Тем самым в комбинаторную логику встраивается еще одно полезное приложение -- система программирования, основанная на декартовой замкнутой категории. Это позволяет еще раз на новом уровне переосмыслить связь операторного и аппликативного стиля программирования.

Обзор темы. Для КАМ формулируется представление о состоянии. Строится перечисление всевозможных состояний, дающее основу для

цикла работы КАМ. Рассматривается назначение “регистров” КАМ -- термина, кода и стека. С применением цикла работы КАМ пересматриваются ранее выведенные оптимизационные равенства. Добавляются практические правила экономии в кодировании вычисления с сохранением его результата.

В результате изучения темы 15 нужно:

знать:

- цикл работы абстрактной машины;

уметь:

- оптимизировать и исполнять сгенерированный код на основе инструкций абстрактной машины.

Тема 16-17. Циклические вычисления. Рекурсивная модификация среды. Примеры КАМ-программ.

Учебная задача. Изучается среда вычислений, организация стека рекурсии, выполнение рекурсивной модификации среды (р.м.с.). Обосновывается связь вычислений с неподвижной точкой и равенств р.м.с.

Обзор темы. Рассматриваются простейшие примеры рекурсивных вычислений, для которых выполняется кодогенерация, оптимизация кода, его исполнение на КАМ. Иллюстрируется на примерах цикл работы КАМ как в стандартном, так и в расширенном виде. Обсуждаются возможные программные реализации на языках CAML, F# и др.

В результате изучения темы 16-17 нужно:

знать:

- состояние исследований в области разработки механизмов вычислений;

- цикл работы абстрактной машины, включая его расширение и формы реализации;

уметь:

- оптимизировать и исполнять сгенерированный код на основе инструкций абстрактной машины, включая условные выражения и конструкции цикла;

- выполнять вычисления, включающие неподвижную точку;

- при необходимости расширять набор инструкции КАМ, дополняя

цикл работы новыми переходами состояний.

Требования к уровню освоения содержания курса

В результате усвоения учебной дисциплины «Модели вычислений» нужно

знать:

- постановку и решение задачи синтеза объекта с заданной комбинаторной характеристикой;
- подход к анализу цикличности вычислений на основе неподвижной точки.
- постановку и решения задачи синтеза структуры данных с заданными математическими свойствами;
- связь различных формулировок теории вычислений;
- различные варианты погружения теорий объектов.
- процесс компилирования комбинаторного кода;
- связь синтаксиса и семантики вычислений с избранными базисами;
- различные механизмы вычислений и пути их усовершенствования посредством различных параметризаций;
- пути и методы устранения коллизий переменных;
- различные формы, в том числе эквациональные, теории вычислений;
- цикл работы абстрактной машины;
- перспективы аппликативных вычислительных технологий и языков CAML, Haskell, F#

уметь:

- синтезировать и анализировать объект с заданной комбинаторной характеристикой;
- производить вычисление (интерпретацию) комбинаторного программного кода, содержащего конструкции цикла;
- устанавливать комбинаторный базис вычислений и применять его для решения задачи компилирования комбинаторного кода;
- строить эквациональные представления вычислений;
- выполнять приведение абстракции к суперкомбинаторам;
- производить вычисление (интерпретацию) редуцированного выражения;
- оптимизировать вычисления, применяя параметризацию;
- выполнять кодогенерацию исходного выражения в промежуточное

представление;

- оптимизировать и исполнять сгенерированный код на основе инструкций абстрактной машины;
- выполнять вычисления, включающие неподвижную точку.

Перечень примерных контрольных вопросов и заданий для самостоятельной работы

См. раздел 10 настоящей программы. Задания уточняются в индивидуальном порядке.

8. Примерная тематика рефератов и курсовых работ

Целью курсовой работы является закрепление материала по разделам курса и выработка навыка решения практических задач. Работа состоит из трех частей, ее полная формулировка, варианты и указания содержатся в [3] списка основной литературы.

9. Практические занятия

Задачи, указания к их решению и ответы содержатся в источниках [1], [2], [3] списка основной литературы.

Тема 1-2. Синтез вычислительного процесса (2 час.).

Тема 3-4. Цикличность. Рекурсия. Комбинатор неподвижной точки (1 час).

Тема 5-6. Типы. Приписывание типов. Разложение объектов в базисах (2 час.).

Тема 7-8. Вычислительные процессы с неподвижными точками (1 час).

Тема 9-10. Машина рекурсии. Минимизации (1 час).

Тема 11-12. Синтез структуры данных (1 час).

Тема 13-14. Суперкомбинаторы (2 час.).

Тема 15-17. Вычисления в категории (7 час.).

При углубленном изучении тем может быть рекомендовано практическое выполнение лабораторных работ, список которых приводится далее.

Лабораторные работы (факультативно)

Тема 1. Формы записи объектов (2 час.).

- Тема 2. Редукция (2 час.).
- Тема 3. Полная редукция (2 час.).
- Тема 4. Базис $\langle I, K, S \rangle$ (2 час.).
- Тема 5. Базис $\langle I, B, C, S \rangle$ (2 час.).
- Тема 6. Расширение базисной системы объектов (2 час.).
- Тема 7. Компилирование кода (2 час.).
- Тема 8. Циклические вычисления (3 час.).

Для обеспечения выполнения работ предполагается использовать Web-ресурсы (по указанию преподавателя).

10. Примерный перечень вопросов к экзамену

Раздел I. Программирование с переменными. Объекты, абстракция, переменные, диапазоны, формальный и фактический параметр, подстановка, связывание формального и фактического параметра.

1. Определение объекта (на примере лямбда-исчисления).
2. Конвертирование программ и данных (на примере постулатов лямбда-исчисления). Примитивная система программирования со связанными переменными.

Раздел II. Программирование без связанных переменных. Объекты, константные формы, комбинирование, комбинаторный код. Эквивалентные преобразования объектов и отношение конвертирования. Редукции объектов.

3. Определение объекта (на примере комбинаторной логики).
4. Конвертирование программ и данных (на примере постулатов комбинаторной логики). Примитивная система программирования без связанных переменных.
5. Трансляция одной примитивной системы программирования в другую (на примере связи лямбда-исчисления и комбинаторной логики).

Раздел III. Синтез объекта. Неподвижная точка в вычислениях.

Циклические вычисления с объектами. Рекурсивные определения и их преобразования.

6. Постановка задачи синтеза нового объекта с заданными свойствами. Синтез алгоритма, реализующего объект, как вывод.
7. Определение неподвижной точки. Циклические вычисления с объектами. Применения.

8. Комбинатор неподвижной точки. Примеры реализаций.
9. Исследование свойств комбинатора неподвижной точки.
10. Теорема о неподвижной точке. Применения в программировании.

Раздел IV. Простейшая форма компилирования кода. Базисы объектов. Устранение переменных в процедуре или функции.

11. Базисы объектов. Алгоритм разложения в базисе I, K, S. Условия применимости и пример.
12. Алгоритм разложения в базисе I, B, C, S. Условия применимости и пример.

Раздел V. Типы в программе. Приписывание типов. Объекты с типами. Правильная типизация и вывод типа через уже известные типы. Правила работы с типами.

13. Исчисление объектов с типами (на примере комбинаторной логики). Правила приписывания типов (F), (I), (K), (S). Пример использования.
14. Приписывание типа объектам (на примере лямбда-исчисления). Правила приписывания типов (λ), (F). Пример использования.
15. Приписать тип объектам I, K, S.

Раздел VI. Приемы кодирования информации. Символьные преобразования. Программирование эквивалентных преобразований информации и конверсии объектов. Устранение коллизий и побочных эффектов. Компилирование кода. Кодогенерация. Оптимизация кода. Экономии в вычислениях. Исполнение кода. Абстрактная машина.

16. Пример коллизии переменных. Связь с постулатами (α) и (β).
17. Коды де Брейна. Применения. Ликвидация коллизий.
18. Понятие о среде. Правила вычисления значения объекта лямбда-исчисления.
19. Теория вычислений. Связь с кодами де Брейна.
20. Теория вычислений. Определение свойств объектов S, Λ , ϵ .
21. Теория вычислений в синтаксической форме. Равенства (ass), (fst), (snd), (dpair), (ac), (quote).
22. Для объекта M обосновать равенство $M = \Lambda(M \circ \text{Snd})$.
23. Понятие о категориальной абстрактной машине. Кодогенерация. Вычисление значения.
24. Цикл работы категориальной абстрактной машины.
25. Оптимизация кода. Правило (Beta).
26. Обоснование кодогенерации и КАМ-вычисления для 2-х местного

оператора.

27. Обоснование вычисления свертывания. Связь постулатов (β) и правила (Beta).
28. Экономии в кодировании.
29. Расширение и реализация категориальной абстрактной машины.
30. Схема вычисления на категориальной абстрактной машине объекта с комбинаторной характеристикой $YM = M(YM)$.
31. Схема вычисления на КАМ объекта, содержащего неподвижную точку. Решение при ограничении $\|M\| = \|\lambda\lambda.P\|$.
32. Стекло рекурсии на КАМ. Его представление гиперграфом.
33. Вычислить на КАМ 1!. Записать выражение, выполнить кодогенерацию и произвести оптимизацию кода, дать его полное табличное исполнение, указав точки рекурсивной модификации среды.
34. Вычислить на КАМ 2!. Записать выражение, выполнить кодогенерацию и произвести оптимизацию кода, дать его полное табличное исполнение, указав точки рекурсивной модификации среды.
35. Вычислить на КАМ 3!. Записать выражение, выполнить кодогенерацию и произвести оптимизацию кода, дать его полное табличное исполнение, указав точки рекурсивной модификации среды.

11. Список литературы

Основная литература

1. Вольфенгаген В.Э. *Методы и средства вычислений с объектами. Аппликативные вычислительные системы*. -- М.:АО "Центр ЮрИнфоР", 2004. - xvi+789 с.
2. Вольфенгаген В.Э. *Категориальная абстрактная машина*. -- М.:МИФИ, 1993. 2-е изд. – М.: АО "Центр ЮрИнфоР", 2002. – 96 с.
3. Вольфенгаген В.Э. *Комбинаторная логика в программировании. Вычисления с объектами в примерах и задачах*. -- М.:МИФИ, 1994; 2-е изд. – М.: АО "Центр ЮрИнфоР", 2003. – 336 с.
4. Косиков С.В. *Информационные системы: категорный подход*. -- М.: ЮрИнфоР-Пресс, 2005. – 96 с.
5. Wolfengagen V.E. *Combinatory logic in programming*. – М.: "Center JurInfoR", 2003. – 336 p.

Дополнительная литература

1. Бердж В. *Методы рекурсивного программирования.* -- М.:Машиностроение, 1983
2. Вольфенгаген В.Э. *Конструкции языков программирования.* – М.: АО “ Центр ЮрИнфоР ”, 2001. – 276 с.
3. Вольфенгаген В.Э., Горюнова И.А., Косиков С.В. *Методы и средства построения систем знания.* -- М.:МИФИ, 1992 (ч. 1, ч. 2)
4. Косиков С.В., Мясников А.В. *Математические методы и средства в новой информационной технологии.* -- М.:МИФИ, 1990
5. Вольфенгаген В.Э., Гольцева Л.В. *Аппликативные вычисления на основе комбинаторов и лямбда-исчисления.* -- М.:МИФИ, 1992
6. Илюхин А.А., Исмаилова Л.Ю., Шаргатова З.И. *Экспертные системы на реляционной основе.* -- М.:МИФИ, 1990
7. Барендрегт Х. *Лямбда-исчисление. Его синтаксис и семантика.* -- М.: Мир, 1985. 606 с.
8. Вольфенгаген В.Э. *Логика. Конспект лекций: техника рассуждений.* 2-е изд., дополн. и перераб. – М.: АО “ Центр ЮрИнфоР ”, 2004. – 229 с.
9. Голдблатт Р. *Топосы. Категорный анализ логики.* – М.: Мир, 1983. – 488 с.
10. Кузин Л.Т. *Основы кибернетики.* – В 2-х т. Т. 1. Математические основы кибернетики. 2-е изд., перераб. и доп. – М.: Энергоатомиздат, 1994. – 576 с.
11. Lambek J., Scott P.J. *Introduction to higher order categorical logic.* – Cambridge University Press, Cambridge, 1986; 1994. – 293 pp.
12. Карри Х. *Основания математической логики.* – М.: Мир, 1969. – 568 с.
13. Рассева Е., Сикорский Р. *Математика метаматематики.* – М.: Наука, 1972. – 592 с.

Дополнительная форма отчетности по проекту

Сайты вариантов курса, программы и рабочие материалы см. на URL <http://jurinfor.exponenta.ru>

По некоторым ключевым вопросам курса может быть подготовлен видеокурс. Например, по компьютингу в категории на примере д.з.к. с разработкой среды вычислений, абстрактной машины и т.д.

Примерный перечень тем видеолекций (6-13).

Часть II. Компьютинг в декартово замкнутой категории (д.з.к.).

6. Вычисления в категории.

- 6.1. Представление о категориальной абстрактной машине (КАМ). Вычисление значения в д.з.к.
- 6.2. Оценивающее отображение. Среда. Примеры вычисления значения выражений.
- 6.3. Коллизии переменных. Устранение коллизий. Кодирование по Дебрейну. Числа Дебрейна.
- 6.4. Вычисление значения в д.з.к. с учетом кодирования по Дебрейну.
- 6.5. Комбинаторный “клей”.
- 6.6. Формулировки теорий вычисления и обоснование их свойств.

7. Значение выражений: теория вычислений в категории

- 7.1. Значение выражений и техника вычисления значений.
- 7.2. Среда вычисления значений и ее представление.
- 7.3. Теория вычисления значений по Дебрейну.
- 7.4. Синтаксическая теория вычислений.
- 7.5. Различные формы вычисления значения выражений.

8. Значение выражений: способы вычисления в категории

- 8.1. Способы вычислений.
- 8.2. Исполнение скомпилированного кода.
- 8.3. Применение сборки кода.
- 8.4. Сравнение способов вычислений.

9. Конструирование в категории абстрактной машины (АМ).

- 9.1. Представление о конструировании абстрактной машины.
 - 9.2. Работа абстрактной машины. Состояния.
 - 9.3. Цикл работы абстрактной машины.
 - 9.4. Примеры вычислений. Компилирование кода и его исполнение.
10. Цикл работы абстрактной машины.
- 10.1. Описание всевозможных переходов состояний.
 - 10.2. Пример вычисления значения 2-местного предиката.
 - 10.2.1. Компилирование кода.
 - 10.2.2. Исполнение кода.
11. Оптимизация вычислений в категории.
- 11.1. Вычисление на абстрактной машине свертывания по постулату (β).
 - 11.2. Компилирование кода и его эквивалентные преобразования
 - 11.3. Экономии кодирования и обоснование вычисления β -свертывания.
 - 11.4. Принцип оптимизации (Beta) и его вывод.
 - 11.5. Оптимизация и экономия на примере вычисления значения 2-местного оператора.
12. Расширение и реализация абстрактной машины.
- 12.1. Неподвижная точка в вычислениях и инструкция ветвления.
 - 12.2. Кодогенерация для выражения с неподвижной точкой.
 - 12.3. Рекурсивная модификация среды (р.м.с.).
 - 12.4. Пример вычисления с р.м.с.
 - 12.5. Большой пример. Выполнение кодогенерации с оптимизацией.
13. Исполнение кода с рекурсивной модификацией среды на абстрактной машине.
- 13.1. Большой пример. Вспомогательные обозначения для упрощения кода.
 - 13.2. Большой пример. Исполнение кода с р.м.с.
 - 13.3. Анализ цикличности в вычислениях. Параметры цикла.