Федеральное агентство по образованию
Государственное образовательное учреждение
высшего профессионального образования
«Уральский государственный университет им. А. М. Горького»

ИОНЦ «Информационная безопасность»

Математико-механический факультет

Кафедра алгебры и дискретной математики

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС

Математическая логика и теория алгоритмов

Методические указания

Автор: доцент кафедры алгебры
и дискретной математики
А. П. Замятин

Екатеринбург
2008

Методические указания

Указания приведем с разбивкой по главам программы.

Глава 1. Логика высказываний

Учебный материал этой главы является несложным и трудностей для усвоения не представляет. При изложении материала на лекциях необходимо сделать акцент на понятии равносильности. Это понятие является основным в «техническом» плане. На практических занятиях необходимо освоить способы преобразования формул, основанные на законах логики высказываний. При этом надо обратить внимание студентов на законы дистрибутивности и закона де Моргана. При первоначальном изучении законов дистрибутивности полезно использовать аналогии с дистрибутивностью умножения чисел относительно сложения.

Одна из задач изучения математической логики в целом и логики высказываний в частности – задача изучения выразительных возможностей языка. В связи с этим на практических занятиях необходимо ряд задач перевода небольшого связного текста естественного языка на язык логики высказываний.

Важным понятием с практической точки является понятие логического следствия. Полезными для изучения этого понятия являются задачи, связанные с анализом на логичность различных рассуждений, проведенных на естественном языке. Кроме того, на практических занятиях надо освоить формальные приемы доказательства логичности и нелогичности.

Программой предусмотрено изучение трех нормальных форм в логике высказываний: ДНФ, СДНФ, КНФ. На практических занятиях необходимо продолжить изучение соответствующих алгоритмов, начатое на лекциях Обратить внимание студентов на метод построения СДНФ, использующий таблицу истинности формулы.

При решении задач минимизации контактных схем надо обратить внимание на то, что преобразования ведутся не в классе всех формул, а в классе формул, удовлетворяющих двум ограничениям: формулы не содержат импликации и эквиваленции и знак отрицания относится только к атомарным формулам.

Глава 2. Логика предикатов

В логике предикатов довольно сложным образом вводится понятие формулы. В силу этого рекомендуется на лекциях постоянно проводить аналогии с соответствующими конструкциями в языках программирования. Обратить внимание студентов на возможность коллизии переменных в формулах, а также на те способы введения понятия формулы, которые позволяют избежать коллизии переменных.

Изучение логики предикатов на практических занятиях полезно начать с рассмотрения выразительных возможностей языка этой логики. Обратить внимание студентов на то, что в ряде этих задач надо вначале подобрать сигнатуру.

Понятие интерпретации в логике предикатов существенно сложнее аналогичного понятия в логике высказываний. Для лучшего усвоения этого понятия рекомендуется на практических занятиях решить две-три соответствующие задачи.

Как и в случае логики высказываний, при изучении законов логики предикатов необходимо сделать акцент на их применение при преобразовании формул. Необходимо добиться довольно свободного оперирования законами при доказательстве равносильности.

При изучении нормальных форм необходимо подчеркнуть, что для бескванторных формул можно говорить о ДНФ, СДНФ и КНФ. Из двух специфичных (по сравнению с логикой высказываний) для логики предикатов нормальных форм – предваренной и сколемовской – обратить внимание на вторую форму.

Понятие логического следствия в логике предикатов сложнее соответствующего понятия в логике высказываний, поэтому изложение этого понятия на лекции рекомендуется побольше иллюстрировать примерами. На практических занятиях надо подчеркнуть, что средствами данной главы можно фактически доказывать только нелогичность рассуждения. Техника доказательства логичности будет разработана в главе 4.

Последняя тема главы посвящена применению логики предикатов в теории баз данных. Здесь рассматривается язык логики предикатов, как язык запросов к базам данных. На практических занятиях при решении соответствующих задач полезно провести сравнение выражения запроса на языке логики предикатов с соответствующим выражением на языке SQL.

Глава 3. Исчисление предикатов

Понятие формулы в этой главе является более узким, нежели в предыдущей главе. На лекциях необходимо подчеркнуть, что причины такого сужения две: минимизация числа аксиом и числа случаев в шаге индукции по построению формулы.

Центральным в этой главе является § 3. На лекциях понятия вывода и доказательства нужно рассмотреть значительным числом примеров. При этом надо обязательно привести примеры 1, 2 и 4. Примеры 1 и 4 используются в доказательстве теоремы о дедукции, пример 2 – для доказательства одного из свойств противоречивых множеств формул. В случае дефицита лекционного времени доказательства теорем о дедукции, об оправданности аксиоматизации, о непротиворечивости можно провести только для логики высказываний. При этом принципиальные моменты доказательств сохранятся.

На практических занятиях нужно освоить навыки построении выводов и доказательств. Это достигается, естественно, решением значительного числа примеров.

При рассмотрении вопросов независимости схем аксиом на практических занятиях полезно попытаться найти другие варианты интерпретации в трехэлементном множестве отрицания и импликации, доказывающие независимость схем аксиом. Предлагается также подумать над следующим вопросом: почему независимость третьей схемы аксиом на лекции не доказана тем же методом, что и независимость первой и второй.

Рассмотренная в третьем параграфе система аксиом и правил не является единственной. Пособие содержит изложение ряда других систем для логики высказываний. В случае дефицита времени материал можно исключить.

Глава 4. Метод резолюций

В начале изучения данной главы надо напомнить, что в случае логики высказываний логичность рассуждения можно доказать построением таблицы истинности. В случае же логики предикатов у нас аналогичного способа нет. Метод резолюций как раз и изучается как метод доказательства логичности в логике предикатов. Тем не менее, в методическом плане полезно начать его изучение с логики высказываний. Хотя теорема о полноте метода в логике высказываний будет очевидным следствием соответствующей теоремы в логике предикатов (и доказательство второй не опирается на доказательство первой теоремы), на лекции рекомендуется рассмотреть ее доказательство.

После изложения основных понятий метода резолюций в логике предикатов (подстановки, унификатора, правила резолюций, вывода) рекомендуется сразу (еще до доказательства теоремы о полноте) привести ряд примеров применений метода для доказательства логичности рассуждений. Доказательство теоремы о полноте базируется на лемме о подъеме, теореме об Н-интерпретациях и теореме Эрбрана. Традиционно трудным для студентов является доказательство леммы о подъеме. На лекции его можно не приводить, так оно является чисто техническим и принципиальных моментов, проясняющих понятия резольвенты и примера оно не содержит. Изложение понятий эрбрановского универсума, Н-интерпретации, семантического дерева необходимо сопровождать значительным числом примеров. Изучение стратегий метода резолюций и областей применения метода в случае недостатка лекционного времени можно отнести на практические занятия. В то же время настоятельно рекомендуется на лекции рассмотреть «взаимоотношения» метода резолюций и логического программирования.

На практических занятиях по данной главе надо вначале отработать технику доказательства логичности, когда необходимые формулы даны в условии задачи. Далее рекомендуется рассмотреть задачи, связанные с доказательством логичности рассуждений, проведенных на естественном языке. Полезно также на практических занятиях рассмотреть основных моменты доказательства теоремы о полноте. Решить задачи, связанные с понятиями Н-интерпретации и семантического дерева.

Глава 5. Функции k-значной логики

Начальный материал данной главы – булевы функции – изучается в курсе дискретной математики. Тем не менее, следует повторить определения понятий полноты и замкнутости классов булевых функций, определения самодвойственной, монотонной и линейной функции. Полезно также повторить формулировку и доказательство теоремы Поста в том числе и в терминах предполных классов.

Новый (по сравнению с курсом дискретной математики) материал, касающийся булевых функций – вопросы минимизации ДНФ. Соответствующий материал в пособии изложен в двух параграфах. Вначале обсуждается понятие сокращенной ДНФ. Для минимизации ДНФ это понятие является центральным, поэтому необходимо привести примеры нахождения сокращенной ДНФ при различных способах задания булевой функции (формулой, картой Карно, на n-мерном кубе). Способ нахождения тупиковых ДНФ рекомендуется свести к задаче о покрытиях множества.

Изучение функций многозначной логики необходимо начать с рассмотрения конкретных функций, поскольку именно они определят сигнатуру, формулами которой и будут в дальнейшем задаваться функции многозначной логики. На лекции надо привести доказательство полноты ряда классах функций, продемонстрировав при этом способ доказательства полноты нового класса сведением к известным полным классам.

Основная цель главы – изложение критерия Розенберга, критерия полноты классов многозначной логики. Формулировка критерия использует понятие класса сохранения отношения. На лекции изложение этого понятия надо сопроводить значительным числом примеров (в пособии соответствующий набор примеров имеется). Изложение критерия Розенберга рекомендуется провести в двух вариантах (так как это сделано в пособии). Весьма полезно на лекции доказать полноту одного-двух классов с использованием критерия Розенберга и сведением к известным полным классам.

На практических занятиях по этой главе рекомендуется повторить известные из курса дискретной математики способы определения самодвойственности, монотонности и линейности данной булевой функции. Необходимо также решить ряд задач на критерий Поста. Изучение функций собственно многозначной логики следует начать с повторения определений конкретных функций, введенных на лекции. Полезно это повторение иллюстрировать графиками. Далее надо решить ряд задач на доказательство полноты нового класса функций сведением к известным полным классам. Рекомендуется уделить значительное время обсуждению линейного, центрального и регулярного отношений, поскольку эти типы отношений в курсе дискретной математики не изучались. На критерий Розенберга на практических занятиях необходимо решить три-четыре задачи.

Глава 6. Алгоритмы и машины Тьюринга

В начале изучения данной главы надо акцентировать внимание студентов необходимости формализации понятия алгоритм для доказательства отсутствия алгоритма, решающего данную задачу. Подчеркнуть сразу, что вариантов формализации этого понятия разработано несколько, и что машина Тьюринга – это только один из вариантов. Изложение понятия машины Тьюринга надо сопроводить большим числом примеров, том числен и не совсем простых. Например, перевод из единичного кода натурального числа в двоичный. Из способов задания машины Тьюринга предпочтение рекомендуется отдавать диаграмме. В пособии примеров машин приведено довольно много. Надо иметь ввиду, что часть из них будет использоваться при доказательстве существования универсальной машины Тьюринга. Для изложения этого доказательства надо выбрать нужный уровень детализации. С одной стороны, дотошное вписывание всех необходимых команд займет много времени и может привести к известной ситуации, когда «за деревьями леса не видно». С другой стороны, если постоянно говорить «ясно, что это можно сделать», возникнет ситуация, когда не будут восприняты принципиальные моменты конструкции универсальной машины. Аналогичное замечание надо сделать относительно доказательства существования рекурсивно перечислимого и нерекурсивного множества.

Центральным с идейной точки зрения моментом является доказательство отсутствие алгоритма для некоторых задач, т. е. факт существования алгоритмически неразрешимых задач. Рекомендуется на лекции привести доказательство алгоритмической неразрешимости проблемы остановки машины Тьюринга и без доказательства привести примеры других алгоритмически неразрешимых проблем.

В конце изучения теории надо вернуться к утверждению о том, что машина Тьюринга – это один из вариантов формализации понятия алгоритм. В качестве других вариантов рекомендуется рассмотреть рекурсивные функции и алгоритмы Маркова.

На практических занятиях по данной главе нужно потренироваться «в конструировании» машин Тьюринга, решающих различные задачи. Полезно доказать правильную вычислимость некоторых функций и предикатов, заданных на множестве натуральных чисел. Важным с точки зрения приложений является понятие многоленточной машины Тьюринга. Его рекомендуется рассмотреть только на практических занятиях. Завершить практические занятия по данной главе надо решением ряда задач, связанных с понятием рекурсивной функции и алгоритма Маркова.

Глава 7. Сложность алгоритмов

При изложении первого параграфа главы на лекции полезно привести примеры конкретных практических ситуаций, в которых приходится решать обсуждаемые задачи. Сведение задач к языкам надо проиллюстрировать двумя-тремя примерами. Четвертый параграф этой главы содержит значительное число результатов по полиномиальной сводимости. Некоторые из них имеют сложное доказательство. Если возник дефицит лекционного времени, то доказательство ряда результатов можно не проводить. Наиболее трудным понятием главы является понятие полиномиальной вычислимости на недетерминированной машине Тьюринга. Изложение этого понятия, безусловно, надо сопровождать значительным числом примеров. Довольно трудно воспринимается студентами доказательство NP-полноты задачи «выполнимость». Тем не менее, это доказательство рекомендуется привести на лекции.

На практических занятиях по данной главе полезно для ряда задач придумать несколько вариантов сведения к языкам. Среди них можно рассмотреть примеры «неразумных» кодировок. Для одной-двух задач из класса $\boldsymbol{P}$-$time$ надо рассмотреть алгоритм на псевдокоде и для него найти временную сложность. При изучении недетерминированной полиномиальной вычислимости рекомендуется придумать несколько соответствующих алгоритмов для задач «раскраска» и «коммивояжер». Весьма полезно на практических занятиях проработать доказательство теоремы о NP-полноте задачи «выполнимость».