

Министерство образования и науки Российской Федерации

Иркутский государственный технический университет

Факультет кибернетики

Кафедра вычислительной техники

УТВЕРЖДАЮ

Председатель Методической комиссии
факультета кибернетики

_____ А.В. Петров

" ____ " _____ 2008 г.

**ИНТЕЛЛЕКТНЫЕ
ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ**

УЧЕБНАЯ ПРОГРАММА ДИСЦИПЛИНЫ

**Укрупненная группа
направлений и
специальностей**

230000 «Информатика и вычислительная техника»

**Направление
подготовки:**

230100 «Информатика и вычислительная техника»

Специальность:

230101 «Вычислительные машины, комплексы,
системы и сети»

Иркутск

2008 г.

1. ИНФОРМАЦИЯ ИЗ ГОС (ДОПОЛНЕНИЕ К ГОС)

Индекс	Наименование дисциплины и ее основные разделы	Всего часов
СД.07	Системы искусственного интеллекта	100
	Искусственный интеллект как научное направление, представление знаний, рассуждений и задач; эпистемологическая полнота представления знаний и эвристически эффективные стратегии поиска решения задач; модели представления знаний: алгоритмические, логические, сетевые и продукционные модели; сценарии; экспертные системы: классификация и структура; инструментальные средства проектирования, разработки и отладки; этапы разработки; примеры реализации.	

2 ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

Учебная дисциплина «Системы искусственного интеллекта» является обязательной инженерной дисциплиной в фундаментальной подготовке специалистов технического и профиля, направленной на развитие у

студентов представления о современных и классических методах решения задач, автоматизирующих принятие решения и обработку символьной информации. Фактически, такая обработка представляет собой самый верхний абстрактный уровень, управляющий и использующий в качестве своих подсистем другие алгоритмы, программы, математические методы и методики решения частных задач.

Цель изучения дисциплины заключается в изучении качественных характеристик задач искусственного интеллекта и классификации этих задач по признакам; изучение методов решения задач в зависимости от класса в классификации (уделяется основное внимание); ознакомление с существующим программным обеспечением, реализующим тот или иной метод, а также самостоятельная разработка подсистем такого программного обеспечения. Курс предполагает наличие у обучаемого предварительных знаний в области программирования на языке высокого уровня, объектно-ориентированного программирования, реляционных баз данных, а также методов программирования с использованием рекурсии. Приветствуются также хорошие знания формальной логики первого порядка, реляционной алгебры, компьютерной математики.

Основными задачами изучения дисциплины являются:

- познакомить студентов с классификацией задач искусственного интеллекта, методами их решения и программным обеспечением, научить выявлять такие задачи среди производственных задач, требующих автоматизации;
- выработать навыки реализации программ в логическом языке программирования, научить логически анализировать программу и доказывать ее правильность (корректность и остановку), давать процедурную и декларативную интерпретацию программ языка ПРОЛОГ;
- научить разрабатывать и реализовывать алгоритмы, реализующие методы искусственного интеллекта, в т.ч. игровые программы для игр с полной информацией, решения задач (problem solving);
- познакомить студента с методами представления символьной информации (в т.ч. знаний) и методами ее обработки;
- развить навыки логического мышления.

3. ТРЕБОВАНИЯ К УРОВНЮ ОСВОЕНИЯ ДИСЦИПЛИНЫ

После изучения дисциплины студенты должны:

уметь:

- проводить формализацию задачи в терминах логики, рекурсии и обработки информации на основе преобразования строковых и древовидных (нечисловых) данных;
- анализировать алгоритмы на предмет их оптимизации эвристическими подсистемами;
- разрабатывать стратегии решения задач;
- выбирать и реализовывать алгоритмы поиска решений или оценивания ситуации;
- разрабатывать экспертные системы;
- владеть методами обработки неполной, нечеткой, противоречивой информации;
- анализировать полученные результаты.

знать:

- современные методы обработки нечисловой информации;
- методы извлечения из задачи эвристической информации, имеющей смысл дополнительной;

иметь представление:

- о методах и методиках автоматизации доказательства логических теорем;
- о методиках интеллектуализации вычислительных систем на основе логических средств программирования и подсистем, основанных на формализованных знаниях;
- каким образом возможно проводить анализ алгоритмов с последующим формальным доказательством их корректности, понимать, что применяемое в промышленности тестирование не обеспечивает полной гарантии корректного функционирования алгоритмов и программ.

4. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРНО-ЛОГИЧЕСКОЙ СХЕМЕ

Для изучения дисциплины, необходимы знания и умения из дисциплин:

- программирование на языке высокого уровня,
- дискретная математика (формальная логика),
- основы лингвистики (базовые знания одного иностранного языка).

Знания и умения, приобретаемые студентами после изучения дисциплины будут использоваться в следующих видах инженерной и исследовательской деятельности:

- Системы автоматического проектирования;
- Анализ данных баз данных;
- Построение баз знаний экспертных систем;
- Интеллектуализация вычислительных систем;
- Анализ алгоритмов.

5. ВИДЫ УЧЕБНОЙ РАБОТЫ И ИХ ТРУДОЕМКОСТЬ

Вид учебной работы	Всего часов	Семестр	
		9	10
1	2	3	4
Общая трудоемкость дисциплины	100	100	-
Аудиторные занятия	68	68	-
Лекции	34	34	-
Лабораторные работы	34	34	-

Практические занятия	-	-	-
Самостоятельная работа	32	32	-
Вид итогового контроля	Экзамен	Экзамен	-

6. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

6.1. Содержание разделов дисциплины

Раздел 1. Введение

Вводный курс: определение понятий искусственного интеллекта, данные и знания.

1.1 Задачи, Виды обработки информации

Классификация задач искусственного интеллекта, их свойства. Понятия символьной обработки информации, автоматизация принятия решения.

1.2 Представление знаний, формализмы представления знаний

Способы формального представления знаний. Формализмы и их свойства: продукционный, логический, фреймовый и сетевой. О преобразовании представлений.

Раздел 2. Язык программирования ПРОЛОГ

Логическое программирование. Представление знаний в виде фраз Хорна (логический формализм).

2.1 Целевое программирование, структура языка

Логическое программирование. Представление знаний в виде фраз Хорна (логический формализм). Сравнение формализма хорновских фраз с языком исчисления предикатов первого порядка.

2.2 Представление данных в языке

Синтаксические структуры языка ПРОЛОГ. Факты. Правила. Понятия цели и подцели. Дополнительные описательные структуры. Структура ПРОЛОГ-программы.

2.3 Задание и программирование отношений

Представление отношений между объектами предметной области в виде множеств и логических зависимостей. Рекурсия.

2.4 Семантики программы (процедурная и декларативная)

Интерпретация ПРОЛОГ-программ: декларативная и процедурная. Доказательство правильности ПРОЛОГ-программы. Математическая индукция: простая и обобщенная индукция. Структурная индукция. Вполне упорядоченные множества.

2.5 Логический вывод и управление им.

Методы поиска логического вывода: метод резолюции, семантических таблиц. Представление процесса поиска вывода в методе резолюции в виде И-ИЛИ дерева. Отсечение в И-ИЛИ-дереве как оптимизация решения некоторых задач. Предикаты определения состояния переменных.

2.6 Списки, обработка списков. Функторы.

Представление рекурсивных структур на языке ПРОЛОГ: списки и деревья. Представление графов с вершинами разного качества.

2.7 Базы данных в ПРОЛОГЕ.

Реляционные базы данных как набор фактов и правил языка ПРОЛОГ. Предикаты изменения содержимого базы данных.

Раздел 3. Решение задач

Понятия планирование действий, допустимое состояние, допустимые переходы из состояния в состояние, цели, и т.п.

3.1 Граф пространства состояний.

Постановка задачи планирования. Формальное определение графа пространства состояний. Задачи планирования, решаемые с помощью поиска пути в графе пространства состояний. Понятие о решении задачи планирования.

3.2 Стратегии поиска решения без учета дополнительной информации.

Алгоритмы поиска пути в графе. Стратегии поиска. Стратегии и алгоритмы поиска «в глубину» и «в ширину». Свойства этих алгоритмов.

3.3 Стратегии поиска решения с учетом дополнительной информации.

Понятия штрафов и стоимости решения, эвристик. Эвристический поиск. Алгоритм A*. Теорема о допустимости алгоритма A*. Модификации классического алгоритма A*: Iterative Deeping A* (IDA*). Алгоритм SMA*.

3.4 И-ИЛИ графы. Алгоритмы поиска решения

Представление задачи с помощью подзадач. Понятия задач и подзадач. Отношение между задачами и подзадачами. И-ИЛИ-графы. Понятие о решении задачи с использованием И-ИЛИ-графов.

3.5 Алгоритмы эвристического поиска.

Стоимость решения задачи в И-ИЛИ-графе. Алгоритм эвристического поиска решения в И-ИЛИ-графе.

3.6 CSP-задачи. Алгоритмы поиска решения.

Определение задачи в виде поиска значений при удовлетворении ограничений (Constrain Satisfaction Problem, CSP). Правило «большого пальца». Итеративные алгоритмы: оценки ограничений и градиентного спуска.

3.7 Эвристические функции оценивания. Методы разработки этих функций.

Методы конструирования оценочных функций: усиление условий задачи, статистические и т.п. Комбинирование оценочных функций. Теорема об оптимальности оценочной функции по отношению к алгоритму A*.

Раздел 4. Игры. Представление позиционных игр с полной информацией. Оценочные функции.

Формальное описание пошаговых игр с полной информацией для двух игроков. Оценочные функции и методы их разработки.

4.1 Алгоритм MiniMax. Альфа-бета – отсечение.

Алгоритм оценки игровых позиций MiniMax. Обход дерева MiniMaxa вглубину. Понятие горизонта. Сужение области поиска с помощью Альфа-Бета отсечения.

Раздел 5. Экспертные системы. Структура экспертной системы.

Определение экспертной системы. Базовая и расширенная структура экспертной системы и взаимодействие ее модулей. Машина логического вывода.

5.2 Классификация экспертных систем.

5.3 Принципы построения машин вывода экспертных систем. Программирование в терминах образов.

Представление знаний в экспертных системах. Обработка знаний экспертных систем машиной логического вывода. Программирование в терминах типовых конфигураций: принцип функционирования. Обратный вывод.

5.4 Продукции. Представление знаний с помощью продукций. Продукции системы CLIPS.

Продукционный формализм представления знаний. Представление фактов и знаний в языке CLIPS.

5.5 Принципы построения подсистем объяснения вывода в экспертных системах.

Представление процесса консультации с экспертной системой как построение дерева рассуждений. Анализ дерева рассуждений и синтез ответов на вопросы «Как?» и «Почему?».

5.6 Инженерия знаний. Полнота базы знаний.

Представление о процессе наполнения базы знаний формализованными знаниями, диалог с экспертом-человеком. Понятие о полноте базы знаний.

Раздел 6. Нечеткая логика. Нечеткие системы управления.

Нечеткая логика как обобщение формальной пропозициональной логики. Нечеткие правила и логические связки. Представление и формализация нечетких понятий. Системы правления, основанные на нечеткой логике. Нечеткие контроллеры.

6.1 Обработка неопределенности в экспертных системах.

Представление нечетких данных в экспертных системах. Коэффициенты достоверности. Система FuzzyCLIPS.

Раздел 7. Эволюционные вычисления.

О задачах и о пространстве решений.

7.1 Генетические алгоритмы.

Определение абстрактного генетического алгоритма. Кодирование информации. Решение задач с помощью генетических алгоритмов.

7.2 Оптимизационные алгоритмы. Градиентный спуск. Алгоритмы последовательного улучшения.

Решение CSP-задач с помощью оптимизационных алгоритмов. Градиентный спуск. Алгоритмы последовательного улучшения. Задача о составлении расписания.

7.3 Нейронные сети.

Формализованный нейрон, его структура. Определение нейронной сети. Разновидности нейронных сетей: Формальные нейроны МакКаллока и Питса, Сеть Хемминга, Сеть Хопфилда. Структуры нейронных сетей, Перцептрон.

Раздел 8. Обучение интеллектуальных систем.

Что такое обучение интеллектуальных систем. Методы обучения: с учителем (метод «кнути и пряника»), без учителя.

8.1 Деревья решений.

Анализ набора фактов и исходов. Представление этого набора в виде дерева решений. Консультация с деревом решений. Энтропия.

8.2 Обучение нейронных сетей.

Обучение сети Хопфилда. Синтез сети на формальных нейронах.

Раздел Ф) Доказательство правильности программ.**

Понятие о правильной программе. Остановка и полнота алгоритма. Доказательство правильности программ с использованием методов математической индукции. Понятия о предусловии и постусловии алгоритма. Инварианты цикла.

Ф.1 Итерационные и рекурсивные алгоритмы. Хвостовая рекурсия. Структурная рекурсия.

Преобразование алгоритмов от рекурсивных к итерационным: хвостовая рекурсия. Понятие схем программ. Структурная рекурсия, обобщение метода математической индукции на вполне упорядоченные множества.

6.2. Распределение разделов дисциплины по видам занятий

№ пп	Раздел дисциплины	Лекции	ЛР	СРС
1	Введение	2		
1.1	Задачи, Виды обработки информации	1		
1.2	Представление знаний, формализмы представления знаний	1		1
2	Язык программирования ПРОЛОГ	7	10	6
2.1	Целевое программирование, структура языка	1		
2.2	Представление данных в языке	1		
2.3	Задание и программирование отношений	1		
2.4	Семантики программы (процедурная и декларативная)	1		
2.5	Логический вывод и управление им	1		
2.6	Списки, обработка списков. Функторы.	1		

2.7	Базы данных в ПРОЛОГЕ	1		
3	Решение задач	7	8	7
3.1	Граф пространства состояний.	1		
3.2	Стратегии поиска решения без учета дополнительной информации.	1		
3.3	Стратегии поиска решения с учетом дополнительной информации. Эвристический поиск. Алгоритм A*. Теорема о допустимости алгоритма A*.	1		
3.4	И-ИЛИ графы. Алгоритмы поиска решения	1		
3.5	Алгоритмы эвристического поиска	1		
3.6	CSP-задачи. Алгоритмы поиска решения	1		
3.7	Эвристические функции оценивания. Методы разработки этих функций	1		
4	Игры. Представление позиционных игр с полной информацией. Оценочные функции	2	8	2
4.2	Алгоритм MiniMax. Альфа-бета – отсечение	2		
5	Экспертные системы. Структура экспертной системы.	5	4	5
5.2	Классификация экспертных систем.	1		
5.3	Принципы построения машин вывода экспертных систем. Программирование в терминах образцов.	1		
5.4	Продукции. Представление знаний с помощью продукций. Продукции системы CLIPS.	1		
5.5	Принципы построения подсистем объяснения вывода в экспертных системах.	1		
5.6	Инженерия знаний. Полнота базы знаний.	1		
6	Нечеткая логика. Нечеткие системы управления.	1	2	1
6.2	Обработка неопределенности в экспертных системах.	1		
7	Эволюционные вычисления	2		2
7.1	Генетические алгоритмы	1		
7.2	Оптимизационные алгоритмы. Градиентный спуск. Алгоритмы последовательного улучшения.	1		
7.3	Нейронные сети.	1		
8	Обучение интеллектуальных систем	4	2	4
8.1	Деревья решений	2		
8.2	Обучение нейронных сетей	2		
Ф*)	Доказательство правильности программ	2		4
Ф.1	Итерационные и рекурсивные алгоритмы. Хвостовая рекурсия. Структурная рекурсия.	2		
	Итого	34	34	32

*) Факультативная тема, проводится только для заинтересовавшихся студентов с целью выявления потенциальных аспирантов.

7. ЛАБОРАТОРНЫЕ РАБОТЫ

Главной целью лабораторных работ по дисциплине является:

Выработка навыков анализа предметной области задачи как задачи искусственного интеллекта, научиться формализовывать задачу в терминах логики и обработки символьной информации, разрабатывать эвристические алгоритмы. Познакомиться с базовыми методами анализа и обобщения данных.

В каждой лабораторной работе студенту необходимо выполнить формализацию задачи, разработать структуры данных и построить алгоритм решения задачи, провести тестирование и анализ перспектив развития программы, точки улучшения в смысле повышения производительности.

Рекомендуемый перечень лабораторных работ.

1. Представление данных и знаний в языке ПРОЛОГ. Формализация высказываний естественного языка.
2. Обработка рекурсивных структур. Списки. Алгоритм Британского Музея.
3. Базы данных в прологе. Программирование в терминах типовых конфигураций. Игры. Решение задач. CSP-задачи. Эволюционные вычисления. Нечеткая логика.
4. Построение экспертной системы в CLIPS.

Для каждой лабораторной работы оформляется отчет.

Отчеты-проекты по лабораторным работам оформляются в соответствии с требованиями методических указаний по выполнению каждой лабораторной работы и требованиями стандарта СТО ИрГТУ.027-2009.

8. ПРАКТИЧЕСКИЕ ЗАНЯТИЯ

Практические занятия в явном виде в курсе не представлены. Практические задачи решаются в рамках лабораторных работ. В частности, практической задачей является формализация задачи и разработка структур данных.

9. САМОСТОЯТЕЛЬНАЯ РАБОТА

9.1 ОБЩИЙ ПЕРЕЧЕНЬ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Рекомендуется следующие виды самостоятельной работы:

- изучение литературы по теме лекций.
- анализ примеров решения задачи, найденных в интернет.
- реализация метода искусственного интеллекта как подсистемы в рамках курсового или дипломного проекта.

9.2. Наименование вида СРС

1. Представление знаний, формализмы представления знаний
2. Реализация программы нахождения производных.
3. Алгоритм IDA*. Реализовать решение задачи этим алгоритмом.
4. Алгоритмы поиска решения полным перебором в задачах CSP.
5. Альфа-бета – отсечение. Реализация языка модификации алгоритма MiniMax для игры в шахматном эндшпиле.
6. Построение подсистемы объяснения вывода в экспертных системах.
7. Обработка неопределенности в экспертных системах. Система FuzzyCLIPS.
8. Алгоритмы последовательного улучшения в задачах оптимизации. Алгоритм Роя.
9. Обучение нейронных сетей. Реализация адгоритма.
10. Программа Пролог как схема доказательства корректности при помощи метода индукции.

9.3. Курсовой проект

Курсовой проект не предусмотрен.

10. РЕКОМЕНДУЕМОЕ ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

10.1. НОРМАТИВНОЕ СОПРОВОЖДЕНИЕ

Основой нормативного сопровождения дисциплины являются:

- Государственный образовательный стандарт высшего профессионального образования по направлению подготовки 230100 (654600) «Информатика и вычислительная техника».
- СТО ИрГТУ 007-2007 СИСТЕМА МЕНЕДЖМЕНТА КАЧЕСТВА. Учебно-методическая деятельность. Планирование учебной деятельности.
- СТО ИрГТУ 008-2007 СИСТЕМА МЕНЕДЖМЕНТА КАЧЕСТВА. Общие требования к оформлению и управлению учебно-методического комплекса дисциплины.
- СТО ИрГТУ 015-2007 СИСТЕМА МЕНЕДЖМЕНТА КАЧЕСТВА. Учебно-методическая деятельность. Контроль успеваемости студентов.

10.2. МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

В состав учебно-методического комплекса дисциплины входят следующие материалы:

- учебная программа;
- рабочая программа;
- конспектом лекций;
- методические указания по выполнению лабораторных работ;
- методические указания по выполнению самостоятельных работ.

10.3. ОСНОВНАЯ ЛИТЕРАТУРА

1. И.Братко, Алгоритмы искусственного интеллекта на языке PROLOG // М.: «Вильямс» - 2004, 640 с. ISBN 5-8459-0664-4.
2. С. Рассел, П. Норвиг. Искусственный интеллект: современный подход (AIMA-2). 2-е издание. 2007. 1408 С., с ил.; ISBN 978-5-8459-0887-2, 0-13-790395-2.
3. П. Джексон «Экспертные системы» – М.: Издательский дом «Вильямс», 2001 – 624 С., ил.

10.4. ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

1. Ж.-Л. Лорьер, Системы искусственного интеллекта: пер. с франц. // М.: «Мир», 1991 – 568 с., ил.
2. Дж. Малпас, Реляционный язык ПРОЛОГ и его применение // М.: «Наука» - 1990, 464 с.
3. Л.Стерлинг, Э.Шапиро, Искусство программирования на языке ПРОЛОГ // М.: «Мир» - 1990, 235 с.
4. Приобретение знаний: пер. с японского, под ред. С.Осуги, Ю.Саэки // М.: «Мир» - 1990, 304 с.
5. Искусственный интеллект: справочник в трех томах // М.: «Радио и связь», 1990.
6. К.Нейлор, Как построить свою экспертную систему // М.: «Энергоатомиздат», 1991 - 286 с.
7. А.Тей и др., Логический подход к искусственному интеллекту // М.: «Мир», 1990 – 432 с.
8. П. Андерсен. Доказательство правильности программ. 1982. 168 С.

10.5. ПРОГРАММНЫЕ СРЕДСТВА

Все программные средства доступны в интернет.

1. The GNU Prolog web site. Stable version is gprolog-1.4.0. URL: <http://www.gprolog.org/>
2. SWI-Prolog's home. SWI Prolog-6.3.1. <http://www.swi-prolog.org/>

3. Любая бесплатная среда программирования любого языка высокого уровня с возможностью обработки динамической памяти.
4. CLIPS: A Tool for Building Expert Systems. CLIPS-6.3.1 beta.

11. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ

Задачи искусственного интеллекта — особый класс задач обработки информации, который характеризуется тремя важными свойствами:

1. Не существует алгоритма решения задачи.
2. В процессе решения задачи производится обработка символьной информации (различного рода структур и отношений).
3. Результатом решения задачи является принятие некоторого решения.

В связи с этим студенту как будущему лицу, принимающему решение о способе автоматизации решения задач на производстве, необходимо научиться выявлять такие задачи из ряда актуальных задач в предметной области. Для этого в процессе изложения лекций и при выполнении лабораторных работ приводятся много примеров из личного опыта преподавателя и его коллег, указывается наличие того или иного свойства задачи, метод и методика ее решения. Кроме того, студенту предлагается самостоятельно принять решение об автоматизации задачи на производстве, где он работает или проходит производственную практику, разумеется, при согласовании с преподавателем.

При реализации лабораторной работы № 3 (с широкой тематикой) студенту предлагается использовать тот язык программирования, который он считает нужным без согласования с преподавателем. Такой подход позволяет студенту развить навыки использования инструментария, которым он владеет, в более широком классе задач.

Учебная программа составлена в соответствии с дополнением к Государственному образовательному стандарту высшего профессионального образования по направлению подготовки: 230100 «Информатика и вычислительная техника»

Программу составил:

Черкашин Евгений Александрович, доцент, к.т.н.,
доцент каф. вычислительной техники

Программа согласована с кафедрой вычислительной техники

“ ____ ” _____ 2008 г. Зав. кафедрой, к.т.н., доц. _____ **Дорофеев А.С.**

Программа одобрена на заседании кафедры вычислительной техники

протокол № ____ от “ ____ ” _____ 2008 г.

Зав. кафедрой А.С. Дорофеев
“ ____ ” _____ 2008 г.

Программа одобрена на заседании Методической комиссии Факультета кибернетики

Протокол № 1 от “15” сентября 2008 г.