**Лекция 3**

**Исторический обзор работ в области ИИ.**

Среди важнейших классов задач, которые ставились перед *ИИ* с момента его зарождения как научного направления (с середины 50-х годов ХХ века), следует выделить следующие *трудно формализуемые задачи*, важные для задач *робототехники*: *доказательство теорем*, *управление роботами*, *распознавание изображений*, *машинный перевод* и *понимание текстов на естественном языке*, *игровые программы*, *машинное творчество* ( *синтез музыки*, *стихотворений*, текстов).

**Доказательство теорем.**

Изучение приемов *доказательства теорем* сыграло важную роль в развитии *ИИ*. Формализация дедуктивного процесса с использованием *логики предикатов* помогает глубже понять некоторые компоненты рассуждений. Многие неформальные задачи, например, медицинская диагностика, допускают формализацию как задачу на *доказательство теорем*. Поиск *доказательства математической теоремы* требует не только произвести дедукцию, исходя из гипотез, но также создать интуитивные догадки и гипотезы о том, какие промежуточные утверждения следует доказать для вывода *доказательства* основной теоремы.

В 1954 году А. Ньюэлл задумал создать программу для игры в шахматы. Дж. Шоу и Г. Саймон объединились в работе по проекту Ньюэлла и в 1956 году они создали язык программирования *IPL*-I (предшественник LISPа) для работы с символьной информацией. Их первыми программами стала программа LT (Logic *Theorist*) для *доказательства теорем* и исчисления высказываний (1956 год), а также программа *NSS* (Newell, Shaw, Simon) для игры в шахматы (1957 год). LT и *NSS* привели к созданию А. Ньюэллом, Дж. Шоу и Г. Саймоном программы *GPS* (General *Problem* Solver) в 1957-1972 годах Программа *GPS* моделировала используемые человеком общие стратегии решения задач и могла применяться для решения шахматных и логических задач, *доказательства теорем*, грамматического разбора предложений, математического интегрирования, головоломок типа "Ханойская башня" и т. д. Процесс работы *GPS* воспроизводит методы решения задач, применяемые человеком: выдвигаются подцели, приближающие к решению, применяется *эвристический метод* (один, другой и т. д.), пока не будет получено решение. Попытки прекращаются, если получить решение не удается. Программа *GPS* могла решать только относительно простые задачи. Ее универсальность достигалась за счет эффективности. Специализированные "*решатели* задач" - STUDENT (Bobrov, 1964) и др. лучше проявляли себя при поиске решения в своих предметных областях. *GPS* оказалась первой программой (написана на языке *IPL*-V), в которой предусматривалось планирование стратегии решения задач.

Для решения трудно формализуемых задач и, в частности, для работы со *знаниями* были созданы языки программирования для задач *ИИ*: *LISP* (1958 год, J. MacCatthy), Пролог (1975-79 годы, D. Warren, F. Pereira), ИнтерLISP, FRL, *KRL*, *SMALLTALK*, *OPS5*, PLANNER, QA4, MACSYMA, *REDUCE*, *РЕФАЛ*, *CLIPS*. К числу наиболее популярных традиционных языков программирования для создания *ИС* следует также отнести С++ и Паскаль.

**Распознавание изображений.**

Рождение *робототехники* выдвинуло задачи машинного зрения и *распознавания изображений* в число первоочередных.

В традиционном *распознавании образов* появился хорошо разработанный математический аппарат, и для не очень сложных объектов оказалось возможным строить практически работающие *системы классификации* по признакам, по аналогии и т. д. В качестве признаков могут рассматриваться любые характеристики распознаваемых объектов. Признаки должны быть инвариантны к ориентации, размеру и вариациям формы объектов. Алфавит признаков придумывается разработчиком системы. Качество *распознавания* во многом зависит от того, насколько удачно придуман алфавит признаков. *Распознавание* состоит в априорном получении вектора признаков для выделенного на изображении отдельного распознаваемого объекта, и лишь затем в определении того, какому из эталонов этот вектор соответствует.

П. Уинстон в начале 80-х годов обратил внимание на необходимость реализации целенаправленного процесса машинного восприятия. Цель должна управлять работой всех процедур, в том числе и процедур нижнего уровня, т. е. процедур предварительной обработки и выделения признаков. Должна иметься возможность на любой стадии процесса в зависимости от получаемого результата возвращаться к его началу для уточнения результатов работы процедур предшествующих уровней. У П. Уинстона, так же как и у других исследователей, до решения практических задач дело не дошло, хотя в 80-е годы вычислительные мощности больших машин позволяли начать решение подобных задач. Таким образом, ранние традиционные системы *распознавания*, основывающиеся на последовательной организации процесса распознавания и классификации объектов, эффективно решать задачи восприятия сложной зрительной информации не могли.

**Экспертные системы.**

Методы *ИИ* нашли применение при создании автоматических консультирующих систем. До 1968 года исследователи в области *ИИ* работали на основе общего подхода - упрощения *комбинаторики*, базирующегося на уменьшении перебора альтернатив исходя из *здравого смысла*, применения *числовых функций* *оценивания* и различных *эвристик*.

В начале 70-х годов произошел качественный скачок и пришло понимание, что необходимы глубокие *знания* в соответствующей области и выделение *знаний* из данных, получаемых от эксперта. Появляются *экспертные системы* (ЭС), или системы, основанные на *знаниях*.

ЭС DENDRAL (середина 60-х годов, Стэнфордский университет) расшифровывала данные масс-спектрографического анализа.

ЭС MYCIN (середина 70-х годов, Стэнфордский университет) ставила диагноз при инфекционных заболеваниях крови.

ЭС PROSPECTOR (1974-1983 годы, Стэнфордский университет) обнаруживала полезные ископаемые.

ЭС SOPHIE обучала диагностированию неисправностей в электрических цепях. ЭС XCON помогала конфигурировать оборудование для систем *VAX* фирмы DEC, ЭС PALLADIO помогала проектировать и тестировать СБИС-схемы.

ЭС JUDITH помогает специалистам по гражданским делам и вместе с юристом и с его слов усваивает фактические и юридические предпосылки дела, а затем предлагает рассмотреть различные варианты подходов к разрешению дела.

ЭС LRS оказывает помощь в подборе и анализе информации о судебных решениях и правовых актах в области кредитно-денежного законодательства, связанного с использованием векселей и чеков.

ЭС "Ущерб" на основе российского трудового законодательства обеспечивает юридический анализ ситуации привлечения рабочих и служащих к материальной ответственности при нанесении предприятию материального ущерба действием или бездействием.

Список созданных ЭС можно перечислять очень долго. Были разработаны и внедрены тысячи реально работающих экспертных систем. Об этом мы будем говорить подробнее в 6 и 7 лекциях.

Разработка инструментальных средств для создания ЭС ведется постоянно. Появляются *экспертные системы* оболочки, совершенствуются технологии создания ЭС. Язык Пролог (1975-79 годы) становится одним из основных инструментов создания ЭС. Язык *CLIPS* (C Language Integrated *Production System*) начал разрабатываться в космическом центре Джонсона *NASA* в 1984 году [[ 1.5 ]](https://intuit.ru/studies/courses/46/46/literature#literature.1.5). Язык *CLIPS* свободен от недостатков предыдущих инструментальных средств для создания ЭС, основанных на языке *LISP*. Появляется инструментарий EXSYS, ставший в начале 90-х годов одним из лидеров по созданию ЭС [[ 1.6 ]](https://intuit.ru/studies/courses/46/46/literature#literature.1.6). В начале ХХI века появляется теория интеллектуальных агентов и экспертных систем на их основе [[ 1.7 ]](https://intuit.ru/studies/courses/46/46/literature#literature.1.7). Web-ориентированный инструментарий JESS (Java *Expert System* Shell), использующий язык представления *знаний* *CLIPS*, приобрел достаточную известность в настоящее время [[ 1.8 ]](https://intuit.ru/studies/courses/46/46/literature#literature.1.8). Среди отечественных инструментальных средств следует отметить веб-ориентированную версию комплекса АТ-ТЕХНОЛОГИЯ, разработанного на кафедре Кибернетики МИФИ. В этом комплексе вся прикладная логика как комплекса в целом, так и разработанных в нем веб-интегрированных ЭС, сосредоточена на стороне сервера [[ 1.9 ]](https://intuit.ru/studies/courses/46/46/literature#literature.1.9).

Практика внедрения ЭС показала, что нет чудодейственных рецептов - нужна кропотливая работа по вводу в ЭВМ опыта и *знаний* специалистов всех областей науки.