**Лекция 4**

**Машинный перевод и понимание текстов на естественном языке.**

Началом работ по *машинному переводу* следует считать 1954 год, когда в США с помощью ЭВМ было переведено шестьдесят фраз. Этот известный "Джорджтаунский эксперимент" произвел неизгладимое впечатление на специалистов. Тогда казалось, что достаточно создать большие хранилища словарей для перевода с одного языка на другой, разработать правила перевода - и проблема будет решена. Когда выяснилось, что проблема не так проста, был создан язык-посредник, облегчающий сопоставление фраз на разных языках. Во второй половине семидесятых годов этот язык-посредник превратился в семантическую модель представления смысла переводимых текстов. Таким образом, достижения в области *машинного перевода* оказались очень полезными для *ИИ*, так как они показали, что анализ естественно-языковых конструкций невозможен без создания *семантической модели*, в которой интерпретируется смысл слов. Создание *семантической модели* привело к пониманию необходимости создания языка для внутреннего представления *знаний*. И, наконец, важным достижением явилось понимание того, что анализ текстов на естественном языке (ЕЯ) состоит из четырех основных этапов: *морфологический анализ*, *синтаксический анализ*, семантический анализ, прагматический анализ [[ 1.10 ]](https://intuit.ru/studies/courses/46/46/literature#literature.1.10).

Задача разбора предложения на естественном языке даже с ограниченным словарем в сотню слов является типичной задачей *ИИ*. Для достаточно формальных предметных областей и конкретных наборов предметов, обладающих однозначными, недвусмысленными характеристиками и названиями, такая система была создана Т. Виноградом в 1971 году. Его знаменитая программа SHRDLU [[ 1.11 ]](https://intuit.ru/studies/courses/46/46/literature#literature.1.11)работает в абстрактном мире, состоящем из разноцветных кубиков и пирамид. В этом мире функционирует "однорукий" робот Робби, способный общаться с человеком на естественном (английском) языке и совершать манипуляции с объектами этого мира. Этот робот также способен понимать достаточно сложные выражения и самостоятельно планировать свою деятельность и объяснять мотивы тех или иных действий.

Вот небольшой пример - фрагмент реального диалога с этой системой:

- Найди блок, который выше, чем тот, что в твоей руке, и положи его в коробку.

- ОК.

- Поставь, пожалуйста, один из красных блоков на другой, а сверху либо зеленый блок, либо *пирамиду*.

- ОК.

- Служит ли самая короткая вещь, которая опирается на опору самой высокой *пирамиды*, опорой для чего-то еще?

- Да, для зеленой *пирамиды*.

- Когда ты брался за зеленую *пирамиду*?

- В то время, когда я строил башенку из красного куба, большого красного блока и большого зеленого куба.

- Почему?

- Чтобы от нее избавиться.

- Почему?

- Чтобы освободить красный куб.

Многократное повторение вопроса "Почему?" в итоге приведет к ответу: "Потому что вы меня об этом попросили". Самым интересным здесь является то, что программа не отвечает на заранее заданные типовые фразы. SHRDLU, можно сказать, "понимает", о чем ее спрашивают, и отвечает достаточно разумно. Сам алгоритм Винограда настолько элегантен, что занимает всего несколько сотен строк кода на языке *LISP*, любимом языке разработчиков *ИИ*, занимающихся анализом ЕЯ. Этот пример с роботом Робби весьма показателен и мы будем обращаться к нему в разных лекциях.

Надо отметить, что даже для английского языка, который служит основой для всех современных языков программирования в силу своей лаконичности и достаточно формальной семантики, до сего дня не удалось создать более-менее эффективную программную систему, способную адекватно понимать СМЫСЛ фраз из достаточно больших областей *знаний*, например, нашего обыденного мира.

В разборе и *понимании естественного русского языка* массу проблем создает сложная падежная система, склонения, времена, отсутствие формального порядка следования членов предложения. Тем не менее российскими учеными созданы эффективные системы разбора фраз ограниченного естественного языка (ОЕЯ) [[ 1.12 ]](https://intuit.ru/studies/courses/46/46/literature#literature.1.12), [[ 1.13 ]](https://intuit.ru/studies/courses/46/46/literature#literature.1.13), [[ 1.14 ]](https://intuit.ru/studies/courses/46/46/literature#literature.1.14).

**Игровые программы.**

К числу первых *игровых программ* можно отнести программу Артура Самуэля по игре в чекерс (американские шашки), написанную в 1947 году, причем в ней использовался ряд основополагающих идей *ИИ*, таких, как перебор вариантов и самообучение.

Научить компьютер играть в шахматы - одна из интереснейших задач в сфере *игровых программ*, использующих методы *ИИ*. Она была поставлена уже на заре вычислительной техники, в конце 50-х годов. В шахматах существуют определенные уровни мастерства, степени качества игры, которые могут дать четкие критерии интеллектуального роста машины. Поэтому компьютерными шахматами активно занимались ученые умы во всем мире. Но шахматы - игра, соревнование, и чтобы продемонстрировать свои логические способности, компьютеру необходим непосредственный противник. В 1974 году впервые прошел чемпионат мира среди шахматных программ в рамках очередного конгресса *IFIP* (International *Federation* of Information Processing) в Стокгольме. Победителем этого состязания стала советская шахматная программа "Каисса" (Каисса - богиня, покровительница шахмат). Эта программа была создана в Москве, в Институте проблем управления Академии наук в команде разработчиков программы-чемпиона, лидерами которой были Владимир Арлазаров, Михаил Донской и Георгий Адельсон-Вельский. "Каисса" показала всему миру способности русских специалистов в области *эвристического программирования*.

**Машинное творчество.**

В 1957 году американские исследователи М. Мэтьюз и Н. Гутман посетили концерт одного малоизвестного пианиста. Концерт им обоим не понравился, и, придя домой, М. Мэтьюз тут же стал писать программу, играющую музыку. Идея Мэтьюза, развиваясь, породила целый класс музыкальных языков программирования, которые вначале назывались MUSIC с номером версии. Язык C-*Sound* произошел как раз из этих программ. А отделение Стэндфордского института исследований, где работал тогда М. Мэтьюз, выросло в музыкальный исследовательский центр под названием CCRMA.

В 1959 году советский математик Рудольф Зарипов начал "сочинять" одноголосные музыкальные пьесы на машине "Урал" [[ 1.15 ]](https://intuit.ru/studies/courses/46/46/literature#literature.1.15). Они назывались "Уральские напевы" и носили характер эксперимента. При их сочинении использовались случайные процессы для различных элементов музыкальной фактуры (форма, ритм, звуковысотность и т. д.). С тех пор появилось очень много программ для алгоритмической композиции. Для различных музыкальных задач было создано специальное программное обеспечение: системы многоканального сведения; системы обработки звука; системы синтеза звука; системы интерактивной композиции; программы алгоритмической композиции и др.

В 1975-1976 годах были проведены эксперименты по сравнению машинной и "человеческой" музыки. Для эксперимента были выбраны мелодии песен известных советских композиторов, опубликованные в сборниках избранных песен, и мелодии, сочиненные на вычислительной машине "Урал-2" по программе Р. Зарипова. Результаты экспериментов таковы: машинные сочинения жюри признало в большинстве случаев наиболее интересными и, "без сомнения, написанными человеком". Таким образом, деятельность машины удовлетворяла критерию Тьюринга - слушатели-эксперты не узнали ее.

Д. А. Поспелов в своем интервью "Литературной газете" [№1, 1976] слегка иронизирует над методом Р. Зарипова, вспоминая, что примерно такой же способ "творчества" предложил еще Остап Бендер в "Золотом теленке", продав журналисту Ухудшанскому свое "Незаменимое пособие для сочинения юбилейных статей, табельных фельетонов, а также парадных стихотворений, од и тропарей", избавляющее от "необходимости ждать, покуда вас окатит потный вал вдохновенья". Из раздела первого (словарь) берутся нужные существительные, прилагательные, глаголы, смешиваются по образцам раздела второго (творческая часть) и получается "шедевр". Такой метод можно запрограммировать и можно написать повести, рассказы, стихи. Но вряд ли это можно назвать творчеством. Практически очевидно, что таким образом не будет создано гениальное в общечеловеческом смысле произведение.

Не будем требовать от *интеллектуальных систем* гениальности. *ИС* уже сейчас способны делать много полезной и разумной работы, которая требует какой-то доли интеллекта.

Среди направлений работ в области *ИИ* следует также выделить **НЕЙРОКИБЕРНЕТИКУ**, или иначе говоря, подход к разработке машин, демонстрирующих "разумное" поведение, на основе архитектур, напоминающих устройство мозга и называемых **нейронными сетями** (НС). В 1942 году, когда Н. Винер определил концепции кибернетики, В. Мак-Каллок и В. Питс опубликовали первый фундаментальный труд по НС, где говорилось о том, что любое хорошо заданное отношение вход-выход может быть представлено в виде формальной НС [[ 1.16 ]](https://intuit.ru/studies/courses/46/46/literature#literature.1.16). Одна из ключевых особенностей нейронных сетей состоит в том, что они способны обучаться на основе опыта, полученного в обучающей среде. В 1957 году Ф. Розенблат изобрел устройство для распознавания на основе НС - *персептрон*, который успешно различал буквы алфавита, хотя и отличался высокой чувствительностью к их написанию [[ 1.17 ]](https://intuit.ru/studies/courses/46/46/literature#literature.1.17).

Читателю, возможно, интересно узнать, что у рядовых муравьев и пчел примерно 80 нейронов на особь (у царицы - 200-300 нейронов), у тараканов - 300 нейронов и эти существа показывают отличные адаптационные свойства в процессе эволюции. У человека число нейронов более 1010.

Пик интереса к НС приходится на 60-е и 70-е годы, но в последние десять лет наблюдается резко возросший объем исследований и разработок НС. Это стало возможным в связи с появлением нового аппаратного обеспечения, повысившего производительность вычислений в НС (нейропроцессоры, *транспьютеры* и т. п.). НС хорошо подходят для *распознавания образов* и решения задач классификации, оптимизации и прогнозирования. Поэтому основными областями применения НС являются:

1. промышленное производство и *робототехника* ;
2. военная промышленность и аэронавтика;
3. банки и страховые компании;
4. службы безопасности;
5. биомедицинская промышленность;
6. телевидение и связь; и другие области.

Заканчивая *исторический обзор* работ в области *ИИ*, следует вернуться в 1981 год. В это время японские специалисты, объединившие свои усилия под эгидой научно-исследовательского центра по обработке информации JIPDEC, опубликовали программу *НИОКР* с целью создания к 1991 году прототипа ЭВМ нового поколения. Эта программа, получившая на Западе название "японский вызов", была представлена как попытка построить интеллектуальный компьютер, к которому можно было бы обращаться на естественном языке и вести беседу.

Серьезность, с которой основные конкуренты Японии откликнулись на брошенный им вызов, объясняется тем, что прежде переход от одного поколения к другому характеризовался изменением элементной базы, ростом производительности и расширением сервисных возможностей для пользователей, владеющих в той или иной мере профессиональными навыками программирования. Переход к ЭВМ пятого поколения означал резкий рост "интеллектуальных" способностей компьютера и возможность диалога между компьютером и непрофессиональным пользователем на естественном языке, в том числе в речевой форме или путем обмена графической информацией - с помощью чертежей, схем, графиков, рисунков. В состав ЭВМ пятого поколения также должна войти система решения задач и логического мышления, обеспечивающая способность машины к *самообучению*, ассоциативной обработке информации и получению *логических выводов*. Уровень "дружелюбия" ЭВМ по отношению к пользователю повысится настолько, что специалист из любой предметной области, не имеющий навыков работы с компьютером, сможет пользоваться ЭВМ при помощи естественных для человека средств общения - речи, рукописного текста, изображений и образов.

В литературе того времени [[ 1.18 ]](https://intuit.ru/studies/courses/46/46/literature#literature.1.18)достаточно подробно описываются все эти вопросы. Здесь отметим только основные компоненты программного обеспечения (ПО), планируемые для систем пятого поколения:

* базовая программная система, включающая систему управления базой *знаний* (СУБЗ), систему приобретения и представления *знаний*, систему решения задач и получения выводов, систему обучения и объяснения решений;
* базовая прикладная система, включающая *интеллектуальную систему* автоматизированного проектирования (САПР) сверхбольших *интегральных схем* (СБИС) и архитектур ЭВМ, *интеллектуальную систему* программирования, систему *машинного перевода* и понимания ЕЯ, систему *распознавания образов* и обработки изображений (не менее 100 000 единиц информации в виде изображений), систему распознавания речи (не менее 10 000 слов), базы *знаний* (*БЗ*) о предметных областях, а также утилитные системы для ввода программ и данных, обеспечивающие диагностику и обслуживание.

Теперь с позиции нашего времени можно сказать, что фирма Microsoft постаралась частично ответить на "японский вызов" в своих версиях операционной системы Windows для персональных компьютеров серии IBM PC AT/486 и выше. Уровень "дружелюбия" ЭВМ пятого поколения по отношению к пользователю действительно значительно повысился по сравнению с другими поколениями ЭВМ. В эти же годы стремительное развитие Internet стало мощным шагом по пути создания распределенных баз *знаний*.