

Математическая логика и теория алгоритмов

Лекция 16

Принцип нормализации

Введение в интеллектуальные системы

Куценко Дмитрий Александрович

Белгородский государственный технологический университет
имени В. Г. Шухова

Институт информационных технологий и управляющих систем
Кафедра программного обеспечения вычислительной техники
и автоматизированных систем

20 декабря 2010 г.

Принцип нормализации

Вариант тезиса Чёрча—Тьюринга, сформулированный применительно к нормальным алгоритмам, принято называть **принципом нормализации**.

Пусть задан алфавит \mathcal{A} . Добавим к нему новые буквы и получим алфавит \mathcal{A}_1 с условием $\mathcal{A} \subseteq \mathcal{A}_1$.

Алфавит \mathcal{A}_1 называется **расширением** алфавита \mathcal{A} .

Принцип нормализации

Пусть задан произвольный вербальный алгоритм \mathcal{A} над алфавитом \mathcal{A} . Тогда существует расширение \mathcal{A}_1 алфавита \mathcal{A} и нормальный алгоритм \mathcal{A}_1 в алфавите \mathcal{A}_1 с условием: произвольное слово P в алфавите \mathcal{A} перерабатывается нормальным алгоритмом \mathcal{A}_1 в тот же самый результат, в который слово P перерабатывается исходным вербальным алгоритмом \mathcal{A} .

Любой нормальный алгоритм эквивалентен некоторой машине Тьюринга, и наоборот — любая машина Тьюринга эквивалентна некоторому нормальному алгоритму.

Принцип нормализации

Вариант тезиса Чёрча—Тьюринга, сформулированный применительно к нормальным алгоритмам, принято называть **принципом нормализации**.

Пусть задан алфавит \mathcal{A} . Добавим к нему новые буквы и получим алфавит \mathcal{A}_1 с условием $\mathcal{A} \subseteq \mathcal{A}_1$.

Алфавит \mathcal{A}_1 называется **расширением** алфавита \mathcal{A} .

Принцип нормализации

Пусть задан произвольный вербальный алгоритм \mathcal{A} над алфавитом \mathcal{A} . Тогда существует расширение \mathcal{A}_1 алфавита \mathcal{A} и нормальный алгоритм \mathcal{A}_1 в алфавите \mathcal{A}_1 с условием: произвольное слово P в алфавите \mathcal{A} перерабатывается нормальным алгоритмом \mathcal{A}_1 в тот же самый результат, в который слово P перерабатывается исходным вербальным алгоритмом \mathcal{A} .

Любой нормальный алгоритм эквивалентен некоторой машине Тьюринга, и наоборот — любая машина Тьюринга эквивалентна некоторому нормальному алгоритму.

Принцип нормализации

Вариант тезиса Чёрча—Тьюринга, сформулированный применительно к нормальным алгоритмам, принято называть **принципом нормализации**.

Пусть задан алфавит \mathcal{A} . Добавим к нему новые буквы и получим алфавит \mathcal{A}_1 с условием $\mathcal{A} \subseteq \mathcal{A}_1$.

Алфавит \mathcal{A}_1 называется **расширением** алфавита \mathcal{A} .

Принцип нормализации

Пусть задан произвольный вербальный алгоритм \mathcal{A} над алфавитом \mathcal{A} . Тогда существует расширение \mathcal{A}_1 алфавита \mathcal{A} и нормальный алгоритм \mathcal{A}_1 в алфавите \mathcal{A}_1 с условием: произвольное слово P в алфавите \mathcal{A} перерабатывается нормальным алгоритмом \mathcal{A}_1 в тот же самый результат, в который слово P перерабатывается исходным вербальным алгоритмом \mathcal{A} .

Любой нормальный алгоритм эквивалентен некоторой машине Тьюринга, и наоборот — любая машина Тьюринга эквивалентна некоторому нормальному алгоритму.

Принцип нормализации

Вариант тезиса Чёрча—Тьюринга, сформулированный применительно к нормальным алгоритмам, принято называть **принципом нормализации**.

Пусть задан алфавит \mathcal{A} . Добавим к нему новые буквы и получим алфавит \mathcal{A}_1 с условием $\mathcal{A} \subseteq \mathcal{A}_1$.

Алфавит \mathcal{A}_1 называется **расширением** алфавита \mathcal{A} .

Принцип нормализации

Пусть задан произвольный вербальный алгоритм \mathcal{A} над алфавитом \mathcal{A} . Тогда существует расширение \mathcal{A}_1 алфавита \mathcal{A} и нормальный алгоритм \mathcal{A}_1 в алфавите \mathcal{A}_1 с условием: произвольное слово P в алфавите \mathcal{A} перерабатывается нормальным алгоритмом \mathcal{A}_1 в тот же самый результат, в который слово P перерабатывается исходным вербальным алгоритмом \mathcal{A} .

Любой нормальный алгоритм эквивалентен некоторой машине Тьюринга, и наоборот — любая машина Тьюринга эквивалентна некоторому нормальному алгоритму.

Принцип нормализации

Вариант тезиса Чёрча—Тьюринга, сформулированный применительно к нормальным алгоритмам, принято называть **принципом нормализации**.

Пусть задан алфавит \mathcal{A} . Добавим к нему новые буквы и получим алфавит \mathcal{A}_1 с условием $\mathcal{A} \subseteq \mathcal{A}_1$.

Алфавит \mathcal{A}_1 называется **расширением** алфавита \mathcal{A} .

Принцип нормализации

Пусть задан произвольный вербальный алгоритм \mathfrak{A} над алфавитом \mathcal{A} . Тогда существует расширение \mathcal{A}_1 алфавита \mathcal{A} и нормальный алгоритм \mathfrak{A}_1 в алфавите \mathcal{A}_1 с условием: произвольное слово P в алфавите \mathcal{A} перерабатывается нормальным алгоритмом \mathfrak{A}_1 в тот же самый результат, в который слово P перерабатывается исходным вербальным алгоритмом \mathfrak{A} .

Любой нормальный алгоритм эквивалентен некоторой машине Тьюринга, и наоборот — любая машина Тьюринга эквивалентна некоторому нормальному алгоритму.

Искусственный интеллект, интеллектуальные системы

Интеллектуальные задачи — задачи, в которых логическая (смысловая) обработка информации преобладает над вычислительной.

Примеры интеллектуальных задач

Анализ ситуаций и принятие решений, понимание и синтез текстов на естественном языке, понимание и синтез речи, управление роботами, анализ графической информации и т. д.

Искусственный интеллект (ИИ) — направление науки и техники, ориентированное на создание программно-аппаратных средств решения интеллектуальных задач.

Термин ввёл Джон Маккарти в 1956 году на конференции в Дартмутском университете.

Интеллектуальная система (ИС) — техническая или программная система, способная решать интеллектуальные задачи.

Искусственный интеллект, интеллектуальные системы

Интеллектуальные задачи — задачи, в которых логическая (смысловая) обработка информации преобладает над вычислительной.

Примеры интеллектуальных задач

Анализ ситуаций и принятие решений, понимание и синтез текстов на естественном языке, понимание и синтез речи, управление роботами, анализ графической информации и т. д.

Искусственный интеллект (ИИ) — направление науки и техники, ориентированное на создание программно-аппаратных средств решения интеллектуальных задач.

Термин ввёл Джон Маккарти в 1956 году на конференции в Дартмутском университете.

Интеллектуальная система (ИС) — техническая или программная система, способная решать интеллектуальные задачи.

Искусственный интеллект, интеллектуальные системы

Интеллектуальные задачи — задачи, в которых логическая (смысловая) обработка информации преобладает над вычислительной.

Примеры интеллектуальных задач

Анализ ситуаций и принятие решений, понимание и синтез текстов на естественном языке, понимание и синтез речи, управление роботами, анализ графической информации и т. д.

Искусственный интеллект (ИИ) — направление науки и техники, ориентированное на создание программно-аппаратных средств решения интеллектуальных задач.

Термин ввёл Джон Маккарти в 1956 году на конференции в Дартмутском университете.

Интеллектуальная система (ИС) — техническая или программная система, способная решать интеллектуальные задачи.

Искусственный интеллект, интеллектуальные системы

Интеллектуальные задачи — задачи, в которых логическая (смысловая) обработка информации преобладает над вычислительной.

Примеры интеллектуальных задач

Анализ ситуаций и принятие решений, понимание и синтез текстов на естественном языке, понимание и синтез речи, управление роботами, анализ графической информации и т. д.

Искусственный интеллект (ИИ) — направление науки и техники, ориентированное на создание программно-аппаратных средств решения интеллектуальных задач.

Термин ввёл Джон Маккарти в 1956 году на конференции в Дартмутском университете.

Интеллектуальная система (ИС) — техническая или программная система, способная решать интеллектуальные задачи.

Искусственный интеллект, интеллектуальные системы

Интеллектуальные задачи — задачи, в которых логическая (смысловая) обработка информации преобладает над вычислительной.

Примеры интеллектуальных задач

Анализ ситуаций и принятие решений, понимание и синтез текстов на естественном языке, понимание и синтез речи, управление роботами, анализ графической информации и т. д.

Искусственный интеллект (ИИ) — направление науки и техники, ориентированное на создание программно-аппаратных средств решения интеллектуальных задач.

Термин ввёл Джон Маккарти в 1956 году на конференции в Дартмутском университете.

Интеллектуальная система (ИС) — техническая или программная система, способная решать интеллектуальные задачи.

Предметная и проблемная области

Реальный мир представляет собой сущности и отношения между ними.

Предметная область — часть реального мира, рассматриваемая в пределах данного контекста.

Под **контекстом** здесь может пониматься, например, область исследования или область, которая является объектом некоторой деятельности.

Существует множество задач и проблем, которые необходимо решать, используя сущности и отношения в предметной области, поэтому в ИИ применяется более широкое понятие **проблемной области**, под которой понимается предметная область плюс решаемые в ней задачи.

Предметная и проблемная области

Реальный мир представляет собой сущности и отношения между ними.

Предметная область — часть реального мира, рассматриваемая в пределах данного контекста.

Под **контекстом** здесь может пониматься, например, область исследования или область, которая является объектом некоторой деятельности.

Существует множество задач и проблем, которые необходимо решать, используя сущности и отношения в предметной области, поэтому в ИИ применяется более широкое понятие **проблемной области**, под которой понимается предметная область плюс решаемые в ней задачи.

Предметная и проблемная области

Реальный мир представляет собой сущности и отношения между ними.

Предметная область — часть реального мира, рассматриваемая в пределах данного контекста.

Под **контекстом** здесь может пониматься, например, область исследования или область, которая является объектом некоторой деятельности.

Существует множество задач и проблем, которые необходимо решать, используя сущности и отношения в предметной области, поэтому в ИИ применяется более широкое понятие **проблемной области**, под которой понимается предметная область плюс решаемые в ней задачи.

Предметная и проблемная области

Реальный мир представляет собой сущности и отношения между ними.

Предметная область — часть реального мира, рассматриваемая в пределах данного контекста.

Под **контекстом** здесь может пониматься, например, область исследования или область, которая является объектом некоторой деятельности.

Существует множество задач и проблем, которые необходимо решать, используя сущности и отношения в предметной области, поэтому в ИИ применяется более широкое понятие **проблемной области**, под которой понимается предметная область плюс решаемые в ней задачи.

Данные

Возникает вопрос: «Что такое знания и чем они отличаются от данных?».

Данные — это отдельные факты, характеризующие объекты, процессы и явления предметной области, а также их свойства. Традиционно выделяют три уровня представления данных:

- ❶ **внешний** — результаты наблюдений над объектами;
- ❷ **логический** — модель данных (некоторая схема описания, связывающая несколько объектов);
- ❸ **физический** — данные на машинных носителях информации (в памяти ЭВМ).

Для хранения данных используются базы данных (БД).

Для БД характерны большой объём и относительно небольшая стоимость информации.

Данные

Возникает вопрос: «Что такое знания и чем они отличаются от данных?».

Данные — это отдельные факты, характеризующие объекты, процессы и явления предметной области, а также их свойства.

Традиционно выделяют три уровня представления данных:

- 1 **внешний** — результаты наблюдений над объектами;
- 2 **логический** — модель данных (некоторая схема описания, связывающая несколько объектов);
- 3 **физический** — данные на машинных носителях информации (в памяти ЭВМ).

Для хранения данных используются базы данных (БД).

Для БД характерны большой объём и относительно небольшая стоимость информации.

Данные

Возникает вопрос: «Что такое знания и чем они отличаются от данных?».

Данные — это отдельные факты, характеризующие объекты, процессы и явления предметной области, а также их свойства. Традиционно выделяют три уровня представления данных:

- 1 **внешний** — результаты наблюдений над объектами;
- 2 **логический** — модель данных (некоторая схема описания, связывающая несколько объектов);
- 3 **физический** — данные на машинных носителях информации (в памяти ЭВМ).

Для хранения данных используются базы данных (БД).

Для БД характерны большой объём и относительно небольшая стоимость информации.

Данные

Возникает вопрос: «Что такое знания и чем они отличаются от данных?».

Данные — это отдельные факты, характеризующие объекты, процессы и явления предметной области, а также их свойства. Традиционно выделяют три уровня представления данных:

- 1 **внешний** — результаты наблюдений над объектами;
- 2 **логический** — модель данных (некоторая схема описания, связывающая несколько объектов);
- 3 **физический** — данные на машинных носителях информации (в памяти ЭВМ).

Для хранения данных используются базы данных (БД).

Для БД характерны большой объём и относительно небольшая стоимость информации.

Данные

Возникает вопрос: «Что такое знания и чем они отличаются от данных?».

Данные — это отдельные факты, характеризующие объекты, процессы и явления предметной области, а также их свойства. Традиционно выделяют три уровня представления данных:

- 1 **внешний** — результаты наблюдений над объектами;
- 2 **логический** — модель данных (некоторая схема описания, связывающая несколько объектов);
- 3 **физический** — данные на машинных носителях информации (в памяти ЭВМ).

Для хранения данных используются **базы данных (БД)**.

Для БД характерны большой объём и относительно небольшая стоимость информации.

Данные

Возникает вопрос: «Что такое знания и чем они отличаются от данных?».

Данные — это отдельные факты, характеризующие объекты, процессы и явления предметной области, а также их свойства. Традиционно выделяют три уровня представления данных:

- 1 **внешний** — результаты наблюдений над объектами;
- 2 **логический** — модель данных (некоторая схема описания, связывающая несколько объектов);
- 3 **физический** — данные на машинных носителях информации (в памяти ЭВМ).

Для хранения данных используются **базы данных (БД)**.

Для БД характерны большой объём и относительно небольшая стоимость информации.

Данные

Возникает вопрос: «Что такое знания и чем они отличаются от данных?».

Данные — это отдельные факты, характеризующие объекты, процессы и явления предметной области, а также их свойства. Традиционно выделяют три уровня представления данных:

- ❶ **внешний** — результаты наблюдений над объектами;
- ❷ **логический** — модель данных (некоторая схема описания, связывающая несколько объектов);
- ❸ **физический** — данные на машинных носителях информации (в памяти ЭВМ).

Для хранения данных используются **базы данных (БД)**.

Для БД характерны большой объём и относительно небольшая стоимость информации.

Данные

Возникает вопрос: «Что такое знания и чем они отличаются от данных?».

Данные — это отдельные факты, характеризующие объекты, процессы и явления предметной области, а также их свойства. Традиционно выделяют три уровня представления данных:

- 1 **внешний** — результаты наблюдений над объектами;
- 2 **логический** — модель данных (некоторая схема описания, связывающая несколько объектов);
- 3 **физический** — данные на машинных носителях информации (в памяти ЭВМ).

Для хранения данных используются **базы данных (БД)**.

Для БД характерны большой объём и относительно небольшая стоимость информации.

Знания

Знания связаны с данными, основываются на них, но представляют собой результат мыслительной деятельности человека, обобщают его опыт, полученный в ходе практической деятельности. Можно сказать, что **знания** — это выявленные закономерности предметной области.

Аналогично уровням представления данных выделяют три уровня представления знаний:

- 1 **знания** — знания в памяти человека;
- 2 **поле знаний** — структурированное полужформализованное описание знаний и модель представления знаний;
- 3 **база знаний** — знания на языках представления знаний в ЭВМ (на машинных носителях информации).

Для хранения знаний используются базы знаний (БЗ). БЗ, наоборот, отличаются сравнительно небольшими объёмами, но исключительно дорогими информационными массивами.

Знания

Знания связаны с данными, основываются на них, но представляют собой результат мыслительной деятельности человека, обобщают его опыт, полученный в ходе практической деятельности. Можно сказать, что **знания** — это выявленные закономерности предметной области.

Аналогично уровням представления данных выделяют три уровня представления знаний:

- 1 **знания** — знания в памяти человека;
- 2 **поле знаний** — структурированное полужформализованное описание знаний и модель представления знаний;
- 3 **база знаний** — знания на языках представления знаний в ЭВМ (на машинных носителях информации).

Для хранения знаний используются базы знаний (БЗ). БЗ, наоборот, отличаются сравнительно небольшими объёмами, но исключительно дорогими информационными массивами.

Знания

Знания связаны с данными, основываются на них, но представляют собой результат мыслительной деятельности человека, обобщают его опыт, полученный в ходе практической деятельности. Можно сказать, что **знания** — это выявленные закономерности предметной области.

Аналогично уровням представления данных выделяют три уровня представления знаний:

- 1 **знания** — знания в памяти человека;
- 2 **поле знаний** — структурированное полужформализованное описание знаний и модель представления знаний;
- 3 **база знаний** — знания на языках представления знаний в ЭВМ (на машинных носителях информации).

Для хранения знаний используются базы знаний (БЗ).

БЗ, наоборот, отличаются сравнительно небольшими объёмами, но исключительно дорогими информационными массивами.

Знания

Знания связаны с данными, основываются на них, но представляют собой результат мыслительной деятельности человека, обобщают его опыт, полученный в ходе практической деятельности. Можно сказать, что **знания** — это выявленные закономерности предметной области.

Аналогично уровням представления данных выделяют три уровня представления знаний:

- 1 **знания** — знания в памяти человека;
- 2 **поле знаний** — структурированное полужформализованное описание знаний и модель представления знаний;
- 3 **база знаний** — знания на языках представления знаний в ЭВМ (на машинных носителях информации).

Для хранения знаний используются **базы знаний (БЗ)**. БЗ, наоборот, отличаются сравнительно небольшими объёмами, но исключительно дорогими информационными массивами.

Знания

Знания связаны с данными, основываются на них, но представляют собой результат мыслительной деятельности человека, обобщают его опыт, полученный в ходе практической деятельности. Можно сказать, что **знания** — это выявленные закономерности предметной области.

Аналогично уровням представления данных выделяют три уровня представления знаний:

- 1 **знания** — знания в памяти человека;
- 2 **поле знаний** — структурированное полужормализованное описание знаний и модель представления знаний;
- 3 **база знаний** — знания на языках представления знаний в ЭВМ (на машинных носителях информации).

Для хранения знаний используются **базы знаний (БЗ)**.

БЗ, наоборот, отличаются сравнительно небольшими объёмами, но исключительно дорогими информационными массивами.

Знания

Знания связаны с данными, основываются на них, но представляют собой результат мыслительной деятельности человека, обобщают его опыт, полученный в ходе практической деятельности. Можно сказать, что **знания** — это выявленные закономерности предметной области.

Аналогично уровням представления данных выделяют три уровня представления знаний:

- ❶ **знания** — знания в памяти человека;
- ❷ **поле знаний** — структурированное полужормализованное описание знаний и модель представления знаний;
- ❸ **база знаний** — знания на языках представления знаний в ЭВМ (на машинных носителях информации).

Для хранения знаний используются **базы знаний (БЗ)**.

БЗ, наоборот, отличаются сравнительно небольшими объёмами, но исключительно дорогими информационными массивами.

Знания

Знания связаны с данными, основываются на них, но представляют собой результат мыслительной деятельности человека, обобщают его опыт, полученный в ходе практической деятельности. Можно сказать, что **знания** — это выявленные закономерности предметной области.

Аналогично уровням представления данных выделяют три уровня представления знаний:

- ❶ **знания** — знания в памяти человека;
- ❷ **поле знаний** — структурированное полужормализованное описание знаний и модель представления знаний;
- ❸ **база знаний** — знания на языках представления знаний в ЭВМ (на машинных носителях информации).

Для хранения знаний используются **базы знаний (БЗ)**.

БЗ, наоборот, отличаются сравнительно небольшими объёмами, но исключительно дорогими информационными массивами.

Знания в программных системах

Существует два способа наделить знаниями программную систему.

- 1 Поместить знания непосредственно в программный код.
В этом случае знания не вынесены в отдельную категорию, поэтому трудно оценить их роль и понять, каким образом они используются в процессе решения задачи, очень сложно модифицировать и пополнять такие знания.
- 2 Вынести знания в отдельную категорию, т. е. представить их в определённом формате и поместить в хранилище — базу знаний.
БЗ легко пополняется и модифицируется.

В современных интеллектуальных системах используется второй способ.

Знания в программных системах

Существует два способа наделить знаниями программную систему.

1 Поместить знания непосредственно в программный код.

В этом случае знания не вынесены в отдельную категорию, поэтому трудно оценить их роль и понять, каким образом они используются в процессе решения задачи, очень сложно модифицировать и пополнять такие знания.

2 Вынести знания в отдельную категорию, т. е. представить их в определённом формате и поместить в хранилище — базу знаний.

БЗ легко пополняется и модифицируется.

В современных интеллектуальных системах используется второй способ.

Знания в программных системах

Существует два способа наделить знаниями программную систему.

- 1 Поместить знания непосредственно в программный код.
В этом случае знания не вынесены в отдельную категорию, поэтому трудно оценить их роль и понять, каким образом они используются в процессе решения задачи, очень сложно модифицировать и пополнять такие знания.
- 2 Вынести знания в отдельную категорию, т. е. представить их в определённом формате и поместить в хранилище — базу знаний.
БЗ легко пополняется и модифицируется.

В современных интеллектуальных системах используется второй способ.

Знания в программных системах

Существует два способа наделить знаниями программную систему.

- ❶ **Поместить знания непосредственно в программный код.**
В этом случае знания не вынесены в отдельную категорию, поэтому трудно оценить их роль и понять, каким образом они используются в процессе решения задачи, очень сложно модифицировать и пополнять такие знания.
- ❷ **Вынести знания в отдельную категорию**, т. е. представить их в определённом формате и поместить в хранилище — базу знаний.
БЗ легко пополняется и модифицируется.

В современных интеллектуальных системах используется второй способ.

Знания в программных системах

Существует два способа наделить знаниями программную систему.

- ❶ **Поместить знания непосредственно в программный код.**
В этом случае знания не вынесены в отдельную категорию, поэтому трудно оценить их роль и понять, каким образом они используются в процессе решения задачи, очень сложно модифицировать и пополнять такие знания.
- ❷ **Вынести знания в отдельную категорию,** т. е. представить их в определённом формате и поместить в хранилище — базу знаний.
БЗ легко пополняется и модифицируется.

В современных интеллектуальных системах используется второй способ.

Знания в программных системах

Существует два способа наделить знаниями программную систему.

- ❶ **Поместить знания непосредственно в программный код.**
В этом случае знания не вынесены в отдельную категорию, поэтому трудно оценить их роль и понять, каким образом они используются в процессе решения задачи, очень сложно модифицировать и пополнять такие знания.
- ❷ **Вынести знания в отдельную категорию,** т. е. представить их в определённом формате и поместить в хранилище — базу знаний.
БЗ легко пополняется и модифицируется.

В современных интеллектуальных системах используется второй способ.

Процедурные и декларативные знания

По своей природе знания можно разделить на процедурные и декларативные.

Процедурные знания — это знания, представленные в виде описаний процедур, с помощью которых их можно получить. В виде процедурных знаний обычно описываются информация о предметной области, характеризующая способы решения задач в этой области, а также различные инструкции, методики, алгоритмы.

Декларативные знания описывают факты и явления, фиксируют наличие или отсутствие таких фактов, а также включают описания основных связей и закономерностей, в которые эти факты и явления входят.

В отличие от процедурных знаний, отвечающих на вопрос «Как сделать X ?», декларативные знания отвечают, скорее, на вопросы: «Что есть X ?» или «Какие связи имеются между X и Y ?», «Почему X ?» и т. д.

Процедурные и декларативные знания

По своей природе знания можно разделить на процедурные и декларативные.

Процедурные знания — это знания, представленные в виде описаний процедур, с помощью которых их можно получить.

В виде процедурных знаний обычно описываются информация о предметной области, характеризующая способы решения задач в этой области, а также различные инструкции, методики, алгоритмы.

Декларативные знания описывают факты и явления, фиксируют наличие или отсутствие таких фактов, а также включают описания основных связей и закономерностей, в которые эти факты и явления входят.

В отличие от процедурных знаний, отвечающих на вопрос «Как сделать X ?», декларативные знания отвечают, скорее, на вопросы: «Что есть X ?» или «Какие связи имеются между X и Y ?», «Почему X ?» и т. д.

Процедурные и декларативные знания

По своей природе знания можно разделить на процедурные и декларативные.

Процедурные знания — это знания, представленные в виде описаний процедур, с помощью которых их можно получить. В виде процедурных знаний обычно описываются информация о предметной области, характеризующая способы решения задач в этой области, а также различные инструкции, методики, алгоритмы.

Декларативные знания описывают факты и явления, фиксируют наличие или отсутствие таких фактов, а также включают описания основных связей и закономерностей, в которые эти факты и явления входят.

В отличие от процедурных знаний, отвечающих на вопрос «Как сделать X ?», декларативные знания отвечают, скорее, на вопросы: «Что есть X ?» или «Какие связи имеются между X и Y ?», «Почему X ?» и т. д.

Процедурные и декларативные знания

По своей природе знания можно разделить на процедурные и декларативные.

Процедурные знания — это знания, представленные в виде описаний процедур, с помощью которых их можно получить. В виде процедурных знаний обычно описываются информация о предметной области, характеризующая способы решения задач в этой области, а также различные инструкции, методики, алгоритмы.

Декларативные знания описывают факты и явления, фиксируют наличие или отсутствие таких фактов, а также включают описания основных связей и закономерностей, в которые эти факты и явления входят.

В отличие от процедурных знаний, отвечающих на вопрос «Как сделать X ?», декларативные знания отвечают, скорее, на вопросы: «Что есть X ?» или «Какие связи имеются между X и Y ?», «Почему X ?» и т. д.

Процедурные и декларативные знания

По своей природе знания можно разделить на процедурные и декларативные.

Процедурные знания — это знания, представленные в виде описаний процедур, с помощью которых их можно получить. В виде процедурных знаний обычно описываются информация о предметной области, характеризующая способы решения задач в этой области, а также различные инструкции, методики, алгоритмы.

Декларативные знания описывают факты и явления, фиксируют наличие или отсутствие таких фактов, а также включают описания основных связей и закономерностей, в которые эти факты и явления входят.

В отличие от процедурных знаний, отвечающих на вопрос «Как сделать X ?», декларативные знания отвечают, скорее, на вопросы: «Что есть X ?» или «Какие связи имеются между X и Y ?», «Почему X ?» и т. д.

Представление декларативных знаний

Существуют десятки способов представления декларативных знаний для различных предметных областях.

Обычно выделяют следующие классы: фреймы, семантические сети и продукционные модели.

Фрейм — это модель абстрактного образа, минимально возможное описание сущности какого-либо объекта, явления, события, ситуации, процесса.

Фрейм состоит из имени и отдельных единиц, называемых слотами. Фреймы и слоты похожи на записи и их поля в языке «Паскаль».

Семантическая сеть — информационная модель предметной области, имеющая вид ориентированного графа, вершины которого соответствуют объектам предметной области, а дуги (рёбра) задают отношения между ними.

Представление декларативных знаний

Существуют десятки способов представления декларативных знаний для различных предметных областях.

Обычно выделяют следующие классы: фреймы, семантические сети и продукционные модели.

Фрейм — это модель абстрактного образа, минимально возможное описание сущности какого-либо объекта, явления, события, ситуации, процесса.

Фрейм состоит из имени и отдельных единиц, называемых **слотами**. Фреймы и слоты похожи на записи и их поля в языке «Паскаль».

Семантическая сеть — информационная модель предметной области, имеющая вид ориентированного графа, вершины которого соответствуют объектам предметной области, а дуги (рёбра) задают отношения между ними.

Представление декларативных знаний

Существуют десятки способов представления декларативных знаний для различных предметных областях.

Обычно выделяют следующие классы: фреймы, семантические сети и продукционные модели.

Фрейм — это модель абстрактного образа, минимально возможное описание сущности какого-либо объекта, явления, события, ситуации, процесса.

Фрейм состоит из имени и отдельных единиц, называемых **слотами**. Фреймы и слоты похожи на записи и их поля в языке «Паскаль».

Семантическая сеть — информационная модель предметной области, имеющая вид ориентированного графа, вершины которого соответствуют объектам предметной области, а дуги (рёбра) задают отношения между ними.

Представление декларативных знаний

Существуют десятки способов представления декларативных знаний для различных предметных областях.

Обычно выделяют следующие классы: фреймы, семантические сети и продукционные модели.

Фрейм — это модель абстрактного образа, минимально возможное описание сущности какого-либо объекта, явления, события, ситуации, процесса.

Фрейм состоит из имени и отдельных единиц, называемых **слотами**. Фреймы и слоты похожи на записи и их поля в языке «Паскаль».

Семантическая сеть — информационная модель предметной области, имеющая вид ориентированного графа, вершины которого соответствуют объектам предметной области, а дуги (рёбра) задают отношения между ними.

Представление декларативных знаний

Существуют десятки способов представления декларативных знаний для различных предметных областях.

Обычно выделяют следующие классы: фреймы, семантические сети и продукционные модели.

Фрейм — это модель абстрактного образа, минимально возможное описание сущности какого-либо объекта, явления, события, ситуации, процесса.

Фрейм состоит из имени и отдельных единиц, называемых **слотами**. Фреймы и слоты похожи на записи и их поля в языке «Паскаль».

Семантическая сеть — информационная модель предметной области, имеющая вид ориентированного графа, вершины которого соответствуют объектам предметной области, а дуги (рёбра) задают отношения между ними.

Продукционные модели

Продукционная модель — модель, основанная на правилах, позволяет представить знание в виде предложений типа «**ЕСЛИ условие, ТО действие**».

Это наиболее часто используемый способ представления знаний в современных экспертных системах (более 80 % от общего числа ЭС).

Основными **преимуществами** продукционной модели являются наглядность, высокая модульность, лёгкость внесения изменений и дополнений, простота механизма логического вывода.

К недостаткам можно отнести неясность взаимных отношений правил, сложность оценки целостного образа знаний, низкая эффективность обработки, отличие от человеческой структуры знаний, отсутствие гибкости в логическом выводе.

Развитием продукционной модели является модель доски объявлений (blackboard).

Продукционные модели

Продукционная модель — модель, основанная на правилах, позволяет представить знание в виде предложений типа «**ЕСЛИ условие, ТО действие**».

Это наиболее часто используемый способ представления знаний в современных экспертных системах (более 80 % от общего числа ЭС).

Основными **преимуществами** продукционной модели являются наглядность, высокая модульность, лёгкость внесения изменений и дополнений, простота механизма логического вывода.

К **недостаткам** можно отнести неясность взаимных отношений правил, сложность оценки целостного образа знаний, низкая эффективность обработки, отличие от человеческой структуры знаний, отсутствие гибкости в логическом выводе.

Развитием продукционной модели является модель доски объявлений (blackboard).

Продукционные модели

Продукционная модель — модель, основанная на правилах, позволяет представить знание в виде предложений типа «**ЕСЛИ условие, ТО действие**».

Это наиболее часто используемый способ представления знаний в современных экспертных системах (более 80 % от общего числа ЭС).

Основными **преимуществами** продукционной модели являются наглядность, высокая модульность, лёгкость внесения изменений и дополнений, простота механизма логического вывода.

К **недостаткам** можно отнести неясность взаимных отношений правил, сложность оценки целостного образа знаний, низкая эффективность обработки, отличие от человеческой структуры знаний, отсутствие гибкости в логическом выводе.

Развитием продукционной модели является модель **доски объявлений (blackboard)**.

Продукционные модели

Продукционная модель — модель, основанная на правилах, позволяет представить знание в виде предложений типа «**ЕСЛИ условие, ТО действие**».

Это наиболее часто используемый способ представления знаний в современных экспертных системах (более 80 % от общего числа ЭС).

Основными **преимуществами** продукционной модели являются наглядность, высокая модульность, лёгкость внесения изменений и дополнений, простота механизма логического вывода.

К **недостаткам** можно отнести неясность взаимных отношений правил, сложность оценки целостного образа знаний, низкая эффективность обработки, отличие от человеческой структуры знаний, отсутствие гибкости в логическом выводе.

Развитием продукционной модели является модель **доски объявлений** (blackboard).

Продукционные модели

Продукционная модель — модель, основанная на правилах, позволяет представить знание в виде предложений типа «**ЕСЛИ условие, ТО действие**».

Это наиболее часто используемый способ представления знаний в современных экспертных системах (более 80 % от общего числа ЭС).

Основными **преимуществами** продукционной модели являются наглядность, высокая модульность, лёгкость внесения изменений и дополнений, простота механизма логического вывода.

К **недостаткам** можно отнести неясность взаимных отношений правил, сложность оценки целостного образа знаний, низкая эффективность обработки, отличие от человеческой структуры знаний, отсутствие гибкости в логическом выводе.

Развитием продукционной модели является модель **доски объявлений** (blackboard).

Продукции

В наиболее общем виде **продукционное правило** можно представить так:

$$\langle i; Q; P; C; A \rightarrow B; N \rangle.$$

Здесь i — собственное **имя** (метка) продукции;

Q — **сфера применения** продукции, вычлняющая из предметной области некоторую её часть, в которой знание, заключённое в продукции, имеет смысл;

P — **предусловие**, содержащее информацию об истинности данной продукция, её приоритетности и т. п., используемую в стратегиях управления выводом для выбора данной продукции для исполнения;

...

Продукции

В наиболее общем виде **продукционное правило** можно представить так:

$$\langle i; Q; P; C; A \rightarrow B; N \rangle.$$

Здесь *i* — собственное **имя** (метка) продукции;

Q — **сфера применения** продукции, вычлняющая из предметной области некоторую её часть, в которой знание, заключённое в продукции, имеет смысл;

P — **предусловие**, содержащее информацию об истинности данной продукция, её приоритетности и т. п., используемую в стратегиях управления выводом для выбора данной продукции для исполнения;

...

Продукции

В наиболее общем виде **продукционное правило** можно представить так:

$$\langle i; Q; P; C; A \rightarrow B; N \rangle.$$

Здесь *i* — собственное **имя** (метка) продукции;

Q — **сфера применения** продукции, вычленяющая из предметной области некоторую её часть, в которой знание, заключённое в продукции, имеет смысл;

P — **предусловие**, содержащее информацию об истинности данной продукция, её приоритетности и т. п., используемую в стратегиях управления выводом для выбора данной продукции для исполнения;

...

Продукции

В наиболее общем виде **продукционное правило** можно представить так:

$$\langle i; Q; P; C; A \rightarrow B; N \rangle.$$

Здесь i — собственное **имя** (метка) продукции;

Q — **сфера применения** продукции, вычлняющая из предметной области некоторую её часть, в которой знание, заключённое в продукции, имеет смысл;

P — **предусловие**, содержащее информацию об истинности данной продукция, её приоритетности и т. п., используемую в стратегиях управления выводом для выбора данной продукции для исполнения;

...

Продукции (продолжение)

В наиболее общем виде продукционное правило можно представить так:

$$\langle i; Q; P; C; A \rightarrow B; N \rangle.$$

...

C — **условие**, представляющее собой предикат, истинное значение которого разрешает применять на некотором шаге данную продукцию;

$A \rightarrow B$ — **ядро** продукции, интерпретация которого может быть различной, например: «Если A истинно, то B истинно», «Если A имеется в базе знаний, то B надо внести в базу знаний», «Если A — текущая ситуация, то надо делать B » и т. п.;

N — **постусловие** продукции, содержащее информацию о том, какие изменения надо внести в данную продукцию или другие продукции, входящие в систему продукций, после выполнения данной продукции.

Продукции (продолжение)

В наиболее общем виде продукционное правило можно представить так:

$$\langle i; Q; P; C; A \rightarrow B; N \rangle.$$

...

C — **условие**, представляющее собой предикат, истинное значение которого разрешает применять на некотором шаге данную продукцию;

$A \rightarrow B$ — **ядро** продукции, интерпретация которого может быть различной, например: «Если A истинно, то B истинно», «Если A имеется в базе знаний, то B надо внести в базу знаний», «Если A — текущая ситуация, то надо делать B » и т. п.;

N — **постусловие** продукции, содержащее информацию о том, какие изменения надо внести в данную продукцию или другие продукции, входящие в систему продукций, после выполнения данной продукции.

Продукции (продолжение)

В наиболее общем виде продукционное правило можно представить так:

$$\langle i; Q; P; C; A \rightarrow B; N \rangle.$$

...

C — **условие**, представляющее собой предикат, истинное значение которого разрешает применять на некотором шаге данную продукцию;

$A \rightarrow B$ — **ядро** продукции, интерпретация которого может быть различной, например: «Если A истинно, то B истинно», «Если A имеется в базе знаний, то B надо внести в базу знаний», «Если A — текущая ситуация, то надо делать B » и т. п.;

N — **постусловие** продукции, содержащее информацию о том, какие изменения надо внести в данную продукцию или другие продукции, входящие в систему продукций, после выполнения данной продукции.

Примеры простых продукционных правил

- ЕСЛИ холодно, ТО надеть шубу.
- ЕСЛИ идёт дождь, ТО взять зонтик.
- ЕСЛИ двигатель не заводится И стартёр двигателя не работает, ТО неполадки в системе электропитания стартера.

Противоречия в продукционной модели

Ещё одним недостатком продукционной модели является тот факт, что при накоплении достаточно большого числа (нескольких сотен) продукций они начинают **противоречить** друг другу.

Рост противоречивости продукционной модели может быть ограничен путём введения следующих механизмов:

Механизм исключений — вводятся специальные правила-исключения, отличающиеся большей конкретностью в сравнении с обобщёнными правилами. При наличии исключения основное правило не применяется.

Механизм возвратов — логический вывод может продолжаться и в том случае, если на каком-либо этапе вывод привёл к противоречию. В этом случае необходимо отказаться от одного из принятых ранее утверждений и осуществить возврат к предыдущему состоянию.

Противоречия в продукционной модели

Ещё одним недостатком продукционной модели является тот факт, что при накоплении достаточно большого числа (нескольких сотен) продукций они начинают **противоречить** друг другу.

Рост противоречивости продукционной модели может быть ограничен путём введения следующих механизмов:

Механизм исключений — вводятся специальные правила-исключения, отличающиеся большей конкретностью в сравнении с обобщёнными правилами. При наличии исключения основное правило не применяется.

Механизм возвратов — логический вывод может продолжаться и в том случае, если на каком-либо этапе вывод привёл к противоречию. В этом случае необходимо отказаться от одного из принятых ранее утверждений и осуществить возврат к предыдущему состоянию.

Противоречия в продукционной модели

Ещё одним недостатком продукционной модели является тот факт, что при накоплении достаточно большого числа (нескольких сотен) продукций они начинают **противоречить** друг другу.

Рост противоречивости продукционной модели может быть ограничен путём введения следующих механизмов:

Механизм исключений — вводятся специальные правила-исключения, отличающиеся большей конкретностью в сравнении с обобщёнными правилами. При наличии исключения основное правило не применяется.

Механизм возвратов — логический вывод может продолжаться и в том случае, если на каком-либо этапе вывод привёл к противоречию. В этом случае необходимо отказаться от одного из принятых ранее утверждений и осуществить возврат к предыдущему состоянию.

Противоречия в продукционной модели

Ещё одним недостатком продукционной модели является тот факт, что при накоплении достаточно большого числа (нескольких сотен) продукций они начинают **противоречить** друг другу.

Рост противоречивости продукционной модели может быть ограничен путём введения следующих механизмов:

Механизм исключений — вводятся специальные правила-исключения, отличающиеся большей конкретностью в сравнении с обобщёнными правилами. При наличии исключения основное правило не применяется.

Механизм возвратов — логический вывод может продолжаться и в том случае, если на каком-либо этапе вывод привёл к противоречию. В этом случае необходимо отказаться от одного из принятых ранее утверждений и осуществить возврат к предыдущему состоянию.

Формализация знаний

Знания специалиста в какой-либо ПО можно разделить на **формализованные (точные)** и **неформализованные (неточные)**.

Формализованные знания излагаются в книгах и руководствах в виде общих и строгих суждений — законов, формул, моделей, алгоритмов.

Формализованные знания характерны для точных наук, таких как математика, физика, химия, астрономия.

Описательные науки, такие как биология, экология, социология, педагогика, медицина и пр. обычно оперируют с неформализованными знаниями.

Неформализованные знания представляют собой множество эмпирических и эвристических приёмов и правил.

Формализация знаний

Знания специалиста в какой-либо ПО можно разделить на **формализованные (точные)** и **неформализованные (неточные)**.

Формализованные знания излагаются в книгах и руководствах в виде общих и строгих суждений — законов, формул, моделей, алгоритмов.

Формализованные знания характерны для точных наук, таких как математика, физика, химия, астрономия.

Описательные науки, такие как биология, экология, социология, педагогика, медицина и пр. обычно оперируют с неформализованными знаниями.

Неформализованные знания представляют собой множество эмпирических и эвристических приёмов и правил.

Формализация знаний

Знания специалиста в какой-либо ПО можно разделить на **формализованные (точные)** и **неформализованные (неточные)**.

Формализованные знания излагаются в книгах и руководствах в виде общих и строгих суждений — законов, формул, моделей, алгоритмов.

Формализованные знания характерны для точных наук, таких как математика, физика, химия, астрономия.

Описательные науки, такие как биология, экология, социология, педагогика, медицина и пр. обычно оперируют с неформализованными знаниями.

Неформализованные знания представляют собой множество эмпирических и эвристических приёмов и правил.

Формализация знаний

Знания специалиста в какой-либо ПО можно разделить на **формализованные (точные)** и **неформализованные (неточные)**.

Формализованные знания излагаются в книгах и руководствах в виде общих и строгих суждений — законов, формул, моделей, алгоритмов.

Формализованные знания характерны для точных наук, таких как математика, физика, химия, астрономия.

Описательные науки, такие как биология, экология, социология, педагогика, медицина и пр. обычно оперируют с неформализованными знаниями.

Неформализованные знания представляют собой множество эмпирических и эвристических приёмов и правил.

Формализация знаний

Знания специалиста в какой-либо ПО можно разделить на **формализованные (точные)** и **неформализованные (неточные)**.

Формализованные знания излагаются в книгах и руководствах в виде общих и строгих суждений — законов, формул, моделей, алгоритмов.

Формализованные знания характерны для точных наук, таких как математика, физика, химия, астрономия.

Описательные науки, такие как биология, экология, социология, педагогика, медицина и пр. обычно оперируют с неформализованными знаниями.

Неформализованные знания представляют собой множество эмпирических и эвристических приёмов и правил.

Задачи проблемной области

Задачи, решаемые на основе точных знаний, называют **формализованными**, а задачи, решаемые с помощью неточных знаний, — **неформализованными**.

Замечание

Не следует путать неформализованные задачи с неформализуемыми, т.к. для первых, возможно, существует формализация, но она пока неизвестна.

Класс задач, относящихся к неформализованным, значительно больше класса формализованных задач.

Задачи проблемной области

Задачи, решаемые на основе точных знаний, называют **формализованными**, а задачи, решаемые с помощью неточных знаний, — **неформализованными**.

Замечание

Не следует путать неформализованные задачи с неформализуемыми, т.к. для первых, возможно, существует формализация, но она пока неизвестна.

Класс задач, относящихся к неформализованным, значительно больше класса формализованных задач.

Задачи проблемной области

Задачи, решаемые на основе точных знаний, называют **формализованными**, а задачи, решаемые с помощью неточных знаний, — **неформализованными**.

Замечание

Не следует путать неформализованные задачи с неформализуемыми, т.к. для первых, возможно, существует формализация, но она пока неизвестна.

Класс задач, относящихся к неформализованным, значительно больше класса формализованных задач.

Признаки неформализованных задач

Задача является неформализованной, если она обладает хотя бы одной из следующих особенностей:

- Алгоритм решения задачи неизвестен;
- Алгоритм известен, но не может быть использован из-за ограниченности ресурсов ЭВМ (времени, памяти);
- Задача не может быть определена в числовой форме (требуется символьное представление);
- Цели задачи не могут быть выражены в терминах точно определённой целевой функции.

Для решения неформализованных задач с помощью компьютерных технологий применяются экспертные системы.

Признаки неформализованных задач

Задача является неформализованной, если она обладает хотя бы одной из следующих особенностей:

- Алгоритм решения задачи неизвестен;
- Алгоритм известен, но не может быть использован из-за ограниченности ресурсов ЭВМ (времени, памяти);
- Задача не может быть определена в числовой форме (требуется символьное представление);
- Цели задачи не могут быть выражены в терминах точно определённой целевой функции.

Для решения неформализованных задач с помощью компьютерных технологий применяются экспертные системы.

Признаки неформализованных задач

Задача является неформализованной, если она обладает хотя бы одной из следующих особенностей:

- Алгоритм решения задачи неизвестен;
- Алгоритм известен, но не может быть использован из-за ограниченности ресурсов ЭВМ (времени, памяти);
- Задача не может быть определена в числовой форме (требуется символьное представление);
- Цели задачи не могут быть выражены в терминах точно определённой целевой функции.

Для решения неформализованных задач с помощью компьютерных технологий применяются экспертные системы.

Признаки неформализованных задач

Задача является неформализованной, если она обладает хотя бы одной из следующих особенностей:

- Алгоритм решения задачи неизвестен;
- Алгоритм известен, но не может быть использован из-за ограниченности ресурсов ЭВМ (времени, памяти);
- Задача не может быть определена в числовой форме (требуется символьное представление);
- Цели задачи не могут быть выражены в терминах точно определённой целевой функции.

Для решения неформализованных задач с помощью компьютерных технологий применяются экспертные системы.

Признаки неформализованных задач

Задача является неформализованной, если она обладает хотя бы одной из следующих особенностей:

- Алгоритм решения задачи неизвестен;
- Алгоритм известен, но не может быть использован из-за ограниченности ресурсов ЭВМ (времени, памяти);
- Задача не может быть определена в числовой форме (требуется символьное представление);
- Цели задачи не могут быть выражены в терминах точно определённой целевой функции.

Для решения неформализованных задач с помощью компьютерных технологий применяются экспертные системы.

Признаки неформализованных задач

Задача является неформализованной, если она обладает хотя бы одной из следующих особенностей:

- Алгоритм решения задачи неизвестен;
- Алгоритм известен, но не может быть использован из-за ограниченности ресурсов ЭВМ (времени, памяти);
- Задача не может быть определена в числовой форме (требуется символьное представление);
- Цели задачи не могут быть выражены в терминах точно определённой целевой функции.

Для решения неформализованных задач с помощью компьютерных технологий применяются **экспертные системы**.

Задачи, решаемые с помощью ЭС

На основе многолетнего опыта разработки интеллектуальных систем выделен ряд неформализованных задач, при решении которых с использованием технологий ИИ достигаются ощутимые результаты.

- 1 **Диагностика** — процесс соотношения объекта с некоторым классом объектов и/или обнаружения неисправностей и их причин в некоторой системе (в технике и живых организмах).
- 2 **Интерпретация данных** — определение смысла данных, т. е. построение описаний по наблюдаемым данным (в т. ч. выделение шума, распознавание параметров и др.).
- 3 **Слежение (мониторинг)** — непрерывная интерпретация в реальном масштабе времени и сигнализация о выходе параметров за допустимые пределы.

Задачи, решаемые с помощью ЭС

На основе многолетнего опыта разработки интеллектуальных систем выделен ряд неформализованных задач, при решении которых с использованием технологий ИИ достигаются ощутимые результаты.

- 1 **Диагностика** — процесс соотношения объекта с некоторым классом объектов и/или обнаружения неисправностей и их причин в некоторой системе (в технике и живых организмах).
- 2 **Интерпретация данных** — определение смысла данных, т. е. построение описаний по наблюдаемым данным (в т. ч. выделение шума, распознавание параметров и др.).
- 3 **Слежение (мониторинг)** — непрерывная интерпретация в реальном масштабе времени и сигнализация о выходе параметров за допустимые пределы.

Задачи, решаемые с помощью ЭС

На основе многолетнего опыта разработки интеллектуальных систем выделен ряд неформализованных задач, при решении которых с использованием технологий ИИ достигаются ощутимые результаты.

- ❶ **Диагностика** — процесс соотношения объекта с некоторым классом объектов и/или обнаружения неисправностей и их причин в некоторой системе (в технике и живых организмах).
- ❷ **Интерпретация данных** — определение смысла данных, т. е. построение описаний по наблюдаемым данным (в т. ч. выделение шума, распознавание параметров и др.).
- ❸ **Слежение (мониторинг)** — непрерывная интерпретация в реальном масштабе времени и сигнализация о выходе параметров за допустимые пределы.

Задачи, решаемые с помощью ЭС

На основе многолетнего опыта разработки интеллектуальных систем выделен ряд неформализованных задач, при решении которых с использованием технологий ИИ достигаются ощутимые результаты.

- ❶ **Диагностика** — процесс соотношения объекта с некоторым классом объектов и/или обнаружения неисправностей и их причин в некоторой системе (в технике и живых организмах).
- ❷ **Интерпретация данных** — определение смысла данных, т. е. построение описаний по наблюдаемым данным (в т. ч. выделение шума, распознавание параметров и др.).
- ❸ **Слежение** (мониторинг) — непрерывная интерпретация в реальном масштабе времени и сигнализация о выходе параметров за допустимые пределы.

Задачи, решаемые с помощью ЭС (продолжение)

- 4 **Прогнозирование** — предсказание последствий, явлений в будущем на базе моделей прошлого и настоящего.
- 5 **Планирование** — конструирование плана, т. е. программы действий.
- 6 **Проектирование** — построение спецификаций на создание объектов с заранее определёнными свойствами.
- 7 **Отладка** — выработка рекомендаций по устранению неисправностей.
- 8 **Обучение** — использование компьютера для обучения какой-то дисциплине или предмету.
Обучение = диагностика + интерпретация +
+ планирование + проектирование.

Задачи, решаемые с помощью ЭС (продолжение)

- 4 **Прогнозирование** — предсказание последствий, явлений в будущем на базе моделей прошлого и настоящего.
- 5 **Планирование** — конструирование плана, т. е. программы действий.
- 6 **Проектирование** — построение спецификаций на создание объектов с заранее определёнными свойствами.
- 7 **Отладка** — выработка рекомендаций по устранению неисправностей.
- 8 **Обучение** — использование компьютера для обучения какой-то дисциплине или предмету.
Обучение = диагностика + интерпретация +
+ планирование + проектирование.

Задачи, решаемые с помощью ЭС (продолжение)

- ④ **Прогнозирование** — предсказание последствий, явлений в будущем на базе моделей прошлого и настоящего.
- ⑤ **Планирование** — конструирование плана, т. е. программы действий.
- ⑥ **Проектирование** — построение спецификаций на создание объектов с заранее определёнными свойствами.
- ⑦ **Отладка** — выработка рекомендаций по устранению неисправностей.
- ⑧ **Обучение** — использование компьютера для обучения какой-то дисциплине или предмету.
Обучение = диагностика + интерпретация +
+ планирование + проектирование.

Задачи, решаемые с помощью ЭС (продолжение)

- ④ **Прогнозирование** — предсказание последствий, явлений в будущем на базе моделей прошлого и настоящего.
- ⑤ **Планирование** — конструирование плана, т. е. программы действий.
- ⑥ **Проектирование** — построение спецификаций на создание объектов с заранее определёнными свойствами.
- ⑦ **Отладка** — выработка рекомендаций по устранению неисправностей.
- ⑧ **Обучение** — использование компьютера для обучения какой-то дисциплине или предмету.
Обучение = диагностика + интерпретация +
+ планирование + проектирование.

Задачи, решаемые с помощью ЭС (продолжение)

- ④ **Прогнозирование** — предсказание последствий, явлений в будущем на базе моделей прошлого и настоящего.
- ⑤ **Планирование** — конструирование плана, т. е. программы действий.
- ⑥ **Проектирование** — построение спецификаций на создание объектов с заранее определёнными свойствами.
- ⑦ **Отладка** — выработка рекомендаций по устранению неисправностей.
- ⑧ **Обучение** — использование компьютера для обучения какой-то дисциплине или предмету.

Обучение = диагностика + интерпретация +
+ планирование + проектирование.

Задачи, решаемые с помощью ЭС (продолжение)

- ④ **Прогнозирование** — предсказание последствий, явлений в будущем на базе моделей прошлого и настоящего.
- ⑤ **Планирование** — конструирование плана, т. е. программы действий.
- ⑥ **Проектирование** — построение спецификаций на создание объектов с заранее определёнными свойствами.
- ⑦ **Отладка** — выработка рекомендаций по устранению неисправностей.
- ⑧ **Обучение** — использование компьютера для обучения какой-то дисциплине или предмету.
Обучение = диагностика + интерпретация +
+ планирование + проектирование.

Задачи, решаемые с помощью ЭС (окончание)

- 9 **Управление** — функция организованной системы, поддерживающая определённый режим деятельности.
Управление = интерпретация + прогнозирование +
+ планирование + моделирование +
+ оптимизация выбранных решений + слежение.
- 10 **Поддержка принятия решений** — совокупность процедур, обеспечивающая лицо, принимающее решения (ЛПР), необходимой информацией и рекомендациями, облегчающие процесс принятия решения.

Задачи, решаемые с помощью ЭС (окончание)

- 9 **Управление** — функция организованной системы, поддерживающая определённый режим деятельности.
Управление = интерпретация + прогнозирование +
+ планирование + моделирование +
+ оптимизация выбранных решений + слежение.
- 10 **Поддержка принятия решений** — совокупность процедур, обеспечивающая лицо, принимающее решения (ЛПР), необходимой информацией и рекомендациями, облегчающие процесс принятия решения.

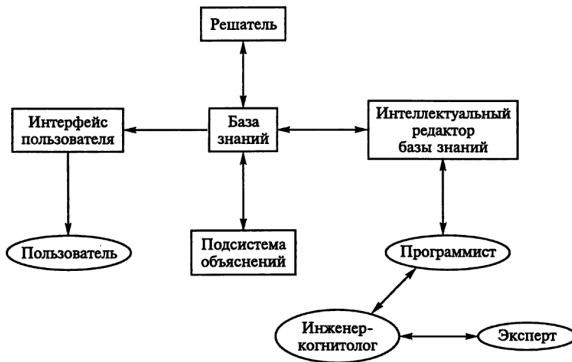
Задачи, решаемые с помощью ЭС (окончание)

- 9 **Управление** — функция организованной системы, поддерживающая определённый режим деятельности.
Управление = интерпретация + прогнозирование +
+ планирование + моделирование +
+ оптимизация выбранных решений + слежение.
- 10 **Поддержка принятия решений** — совокупность процедур, обеспечивающая лицо, принимающее решения (ЛПР), необходимой информацией и рекомендациями, облегчающие процесс принятия решения.

Экспертные системы. Обобщённая структура

Экспертные системы (ЭС) — это сложные программные комплексы, аккумулирующие знания специалистов (экспертов) в конкретных предметных областях и тиражирующие эти знания для консультаций менее квалифицированных пользователей.

Обобщённая схема ЭС:



Компоненты ЭС

Обычно в состав ЭС входят следующие модули:

- **База знаний (БЗ)** — ядро ЭС, совокупность знаний предметной области, записанная на машинном носителе в форме, понятной эксперту и пользователю.
- **Интеллектуальный редактор БЗ** — позволяет инженеру по знаниям составить БЗ в диалоговом режиме, облегчает работу с БЗ.
- **Интерфейс пользователя** — реализует диалог пользователя с ЭС на стадии как ввода информации, так и получения результатов.
- **Решатель (машина вывода)** — моделирует ход рассуждений эксперта на основании знаний, имеющихся в БЗ.
- **Подсистема объяснений** — позволяет пользователю получить ответы на вопросы «Как?» и «Почему?», например, «Как была получена та или иная рекомендация» и «Почему система приняла такое решение».

Компоненты ЭС

Обычно в состав ЭС входят следующие модули:

- **База знаний (БЗ)** — ядро ЭС, совокупность знаний предметной области, записанная на машинном носителе в форме, понятной эксперту и пользователю.
- **Интеллектуальный редактор БЗ** — позволяет инженеру по знаниям составить БЗ в диалоговом режиме, облегчает работу с БЗ.
- **Интерфейс пользователя** — реализует диалог пользователя с ЭС на стадии как ввода информации, так и получения результатов.
- **Решатель (машина вывода)** — моделирует ход рассуждений эксперта на основании знаний, имеющихся в БЗ.
- **Подсистема объяснений** — позволяет пользователю получить ответы на вопросы «Как?» и «Почему?», например, «Как была получена та или иная рекомендация» и «Почему система приняла такое решение».

Компоненты ЭС

Обычно в состав ЭС входят следующие модули:

- **База знаний (БЗ)** — ядро ЭС, совокупность знаний предметной области, записанная на машинном носителе в форме, понятной эксперту и пользователю.
- **Интеллектуальный редактор БЗ** — позволяет инженеру по знаниям составить БЗ в диалоговом режиме, облегчает работу с БЗ.
- **Интерфейс пользователя** — реализует диалог пользователя с ЭС на стадии как ввода информации, так и получения результатов.
- **Решатель (машина вывода)** — моделирует ход рассуждений эксперта на основании знаний, имеющихся в БЗ.
- **Подсистема объяснений** — позволяет пользователю получить ответы на вопросы «Как?» и «Почему?», например, «Как была получена та или иная рекомендация» и «Почему система приняла такое решение».

Компоненты ЭС

Обычно в состав ЭС входят следующие модули:

- **База знаний (БЗ)** — ядро ЭС, совокупность знаний предметной области, записанная на машинном носителе в форме, понятной эксперту и пользователю.
- **Интеллектуальный редактор БЗ** — позволяет инженеру по знаниям составить БЗ в диалоговом режиме, облегчает работу с БЗ.
- **Интерфейс пользователя** — реализует диалог пользователя с ЭС на стадии как ввода информации, так и получения результатов.
- **Решатель (машина вывода)** — моделирует ход рассуждений эксперта на основании знаний, имеющихся в БЗ.
- **Подсистема объяснений** — позволяет пользователю получить ответы на вопросы «Как?» и «Почему?», например, «Как была получена та или иная рекомендация» и «Почему система приняла такое решение».

Компоненты ЭС

Обычно в состав ЭС входят следующие модули:

- **База знаний (БЗ)** — ядро ЭС, совокупность знаний предметной области, записанная на машинном носителе в форме, понятной эксперту и пользователю.
- **Интеллектуальный редактор БЗ** — позволяет инженеру по знаниям составить БЗ в диалоговом режиме, облегчает работу с БЗ.
- **Интерфейс пользователя** — реализует диалог пользователя с ЭС на стадии как ввода информации, так и получения результатов.
- **Решатель (машина вывода)** — моделирует ход рассуждений эксперта на основании знаний, имеющихся в БЗ.
- **Подсистема объяснений** — позволяет пользователю получить ответы на вопросы «Как?» и «Почему?», например, «Как была получена та или иная рекомендация» и «Почему система приняла такое решение».

Компоненты ЭС

Обычно в состав ЭС входят следующие модули:

- **База знаний (БЗ)** — ядро ЭС, совокупность знаний предметной области, записанная на машинном носителе в форме, понятной эксперту и пользователю.
- **Интеллектуальный редактор БЗ** — позволяет инженеру по знаниям составить БЗ в диалоговом режиме, облегчает работу с БЗ.
- **Интерфейс пользователя** — реализует диалог пользователя с ЭС на стадии как ввода информации, так и получения результатов.
- **Решатель (машина вывода)** — моделирует ход рассуждений эксперта на основании знаний, имеющихся в БЗ.
- **Подсистема объяснений** — позволяет пользователю получить ответы на вопросы «Как?» и «Почему?», например, «Как была получена та или иная рекомендация» и «Почему система приняла такое решение».

Коллектив разработчиков ЭС

В коллектив разработчиков ЭС входят четыре группы специалистов — эксперты, инженеры по знаниям, программисты и пользователи.

- **Эксперт** — высококвалифицированный специалист, согласившийся поделиться своим опытом решения конкретного класса задач в конкретной предметной/проблемной области.
- **Инженер по знаниям** (инженер-когнитолог) — специалист в области ИИ, возглавляет коллектив, руководит проектом, организует весь процесс создания ЭС, извлекает и формализует знания эксперта и передаёт их программисту.
- **Программист** — специалист по разработке инструментальных средств, кодирует знания и помещает их в БЗ ЭС.
- **Пользователь** — лицо, для которого предназначена система.

Коллектив разработчиков ЭС

В коллектив разработчиков ЭС входят четыре группы специалистов — эксперты, инженеры по знаниям, программисты и пользователи.

- **Эксперт** — высококвалифицированный специалист, согласившийся поделиться своим опытом решения конкретного класса задач в конкретной предметной/проблемной области.
- **Инженер по знаниям** (инженер-когнитолог) — специалист в области ИИ, возглавляет коллектив, руководит проектом, организует весь процесс создания ЭС, извлекает и формализует знания эксперта и передаёт их программисту.
- **Программист** — специалист по разработке инструментальных средств, кодирует знания и помещает их в БЗ ЭС.
- **Пользователь** — лицо, для которого предназначена система.

Коллектив разработчиков ЭС

В коллектив разработчиков ЭС входят четыре группы специалистов — эксперты, инженеры по знаниям, программисты и пользователи.

- **Эксперт** — высококвалифицированный специалист, согласившийся поделиться своим опытом решения конкретного класса задач в конкретной предметной/проблемной области.
- **Инженер по знаниям** (инженер-когнитолог) — специалист в области ИИ, возглавляет коллектив, руководит проектом, организует весь процесс создания ЭС, извлекает и формализует знания эксперта и передаёт их программисту.
- **Программист** — специалист по разработке инструментальных средств, кодирует знания и помещает их в БЗ ЭС.
- **Пользователь** — лицо, для которого предназначена система.

Коллектив разработчиков ЭС

В коллектив разработчиков ЭС входят четыре группы специалистов — эксперты, инженеры по знаниям, программисты и пользователи.

- **Эксперт** — высококвалифицированный специалист, согласившийся поделиться своим опытом решения конкретного класса задач в конкретной предметной/проблемной области.
- **Инженер по знаниям** (инженер-когнитолог) — специалист в области ИИ, возглавляет коллектив, руководит проектом, организует весь процесс создания ЭС, извлекает и формализует знания эксперта и передаёт их программисту.
- **Программист** — специалист по разработке инструментальных средств, кодирует знания и помещает их в БЗ ЭС.
- **Пользователь** — лицо, для которого предназначена система.

Коллектив разработчиков ЭС

В коллектив разработчиков ЭС входят четыре группы специалистов — эксперты, инженеры по знаниям, программисты и пользователи.

- **Эксперт** — высококвалифицированный специалист, согласившийся поделиться своим опытом решения конкретного класса задач в конкретной предметной/проблемной области.
- **Инженер по знаниям** (инженер-когнитолог) — специалист в области ИИ, возглавляет коллектив, руководит проектом, организует весь процесс создания ЭС, извлекает и формализует знания эксперта и передаёт их программисту.
- **Программист** — специалист по разработке инструментальных средств, кодирует знания и помещает их в БЗ ЭС.
- **Пользователь** — лицо, для которого предназначена система.

Этапы разработки ЭС

В процессе своей разработки ЭС проходят стадии, в результате которых создаются различные версии, называемые **прототипами**:

- 1 Демонстрационный прототип — ЭС, которая решает часть требуемых задач, демонстрируя жизнеспособность метода инженерии знаний.
БЗ содержит 50—100 правил, время разработки — 6—12 мес.
- 2 Исследовательский прототип — ЭС, которая решает все требуемые задачи, но неустойчива в работе и неполностью проверена.
БЗ содержит 200—500 правил, время разработки — 3—6 мес.

Этапы разработки ЭС

В процессе своей разработки ЭС проходят стадии, в результате которых создаются различные версии, называемые **прототипами**:

- 1 **Демонстрационный прототип** — ЭС, которая решает часть требуемых задач, демонстрируя жизнеспособность метода инженерии знаний.
БЗ содержит 50—100 правил, время разработки — 6—12 мес.
- 2 **Исследовательский прототип** — ЭС, которая решает все требуемые задачи, но неустойчива в работе и неполностью проверена.
БЗ содержит 200—500 правил, время разработки — 3—6 мес.

Этапы разработки ЭС

В процессе своей разработки ЭС проходят стадии, в результате которых создаются различные версии, называемые **прототипами**:

- ❶ **Демонстрационный прототип** — ЭС, которая решает часть требуемых задач, демонстрируя жизнеспособность метода инженерии знаний.
БЗ содержит 50—100 правил, время разработки — 6—12 мес.
- ❷ **Исследовательский прототип** — ЭС, которая решает все требуемые задачи, но неустойчива в работе и неполностью проверена.
БЗ содержит 200—500 правил, время разработки — 3—6 мес.

Этапы разработки ЭС

В процессе своей разработки ЭС проходят стадии, в результате которых создаются различные версии, называемые **прототипами**:

- 1 **Демонстрационный прототип** — ЭС, которая решает часть требуемых задач, демонстрируя жизнеспособность метода инженерии знаний.
БЗ содержит 50—100 правил, время разработки — 6—12 мес.
- 2 **Исследовательский прототип** — ЭС, которая решает все требуемые задачи, но неустойчива в работе и неполностью проверена.
БЗ содержит 200—500 правил, время разработки — 3—6 мес.

Этапы разработки ЭС

В процессе своей разработки ЭС проходят стадии, в результате которых создаются различные версии, называемые **прототипами**:

- 1 **Демонстрационный прототип** — ЭС, которая решает часть требуемых задач, демонстрируя жизнеспособность метода инженерии знаний.
БЗ содержит 50—100 правил, время разработки — 6—12 мес.
- 2 **Исследовательский прототип** — ЭС, которая решает все требуемые задачи, но неустойчива в работе и неполностью проверена.
БЗ содержит 200—500 правил, время разработки — 3—6 мес.

Этапы разработки ЭС (продолжение)

- 3 Действующий прототип — надёжно решает задачи, но для решения сложных задач может потратить много времени и памяти.

БЗ содержит 500—1000 правил, время разработки — 6—12 мес.

- 4 Промышленная ЭС — обеспечивает высокое качество решения всех задач при минимуме времени и памяти, что достигается в результате оптимизации программной системы.

БЗ содержит 1000—1500 правил, время разработки — 1—1,5 года.

- 5 Коммерческая ЭС — отличается от промышленной тем, что может продаваться различным потребителям.

БЗ содержит 1500—3000 правил, время разработки — 1,5—3 года, стоимость — 0,3—5 млн. долларов США.

Этапы разработки ЭС (продолжение)

- ③ **Действующий прототип** — надёжно решает задачи, но для решения сложных задач может потратить много времени и памяти.
БЗ содержит 500—1000 правил, время разработки — 6—12 мес.
- ④ **Промышленная ЭС** — обеспечивает высокое качество решения всех задач при минимуме времени и памяти, что достигается в результате оптимизации программной системы.
БЗ содержит 1000—1500 правил, время разработки — 1—1,5 года.
- ⑤ **Коммерческая ЭС** — отличается от промышленной тем, что может продаваться различным потребителям.
БЗ содержит 1500—3000 правил, время разработки — 1,5—3 года, стоимость — 0,3—5 млн. долларов США.

Этапы разработки ЭС (продолжение)

- ③ **Действующий прототип** — надёжно решает задачи, но для решения сложных задач может потратить много времени и памяти.
БЗ содержит 500—1000 правил, время разработки — 6—12 мес.
- ④ **Промышленная ЭС** — обеспечивает высокое качество решения всех задач при минимуме времени и памяти, что достигается в результате оптимизации программной системы.
БЗ содержит 1000—1500 правил, время разработки — 1—1,5 года.
- ⑤ **Коммерческая ЭС** — отличается от промышленной тем, что может продаваться различным потребителям.
БЗ содержит 1500—3000 правил, время разработки — 1,5—3 года, стоимость — 0,3—5 млн. долларов США.

Этапы разработки ЭС (продолжение)

- ③ **Действующий прототип** — надёжно решает задачи, но для решения сложных задач может потратить много времени и памяти.
БЗ содержит 500—1000 правил, время разработки — 6—12 мес.
- ④ **Промышленная ЭС** — обеспечивает высокое качество решения всех задач при минимуме времени и памяти, что достигается в результате оптимизации программной системы.
БЗ содержит 1000—1500 правил, время разработки — 1—1,5 года.
- ⑤ **Коммерческая ЭС** — отличается от промышленной тем, что может продаваться различным потребителям.
БЗ содержит 1500—3000 правил, время разработки — 1,5—3 года, стоимость — 0,3—5 млн. долларов США.

Этапы разработки ЭС (продолжение)

- ③ **Действующий прототип** — надёжно решает задачи, но для решения сложных задач может потратить много времени и памяти.
БЗ содержит 500—1000 правил, время разработки — 6—12 мес.
- ④ **Промышленная ЭС** — обеспечивает высокое качество решения всех задач при минимуме времени и памяти, что достигается в результате оптимизации программной системы.
БЗ содержит 1000—1500 правил, время разработки — 1—1,5 года.
- ⑤ **Коммерческая ЭС** — отличается от промышленной тем, что может продаваться различным потребителям.
БЗ содержит 1500—3000 правил, время разработки — 1,5—3 года, стоимость — 0,3—5 млн. долларов США.

Этапы разработки ЭС (продолжение)

- ③ **Действующий прототип** — надёжно решает задачи, но для решения сложных задач может потратить много времени и памяти.
БЗ содержит 500—1000 правил, время разработки — 6—12 мес.
- ④ **Промышленная ЭС** — обеспечивает высокое качество решения всех задач при минимуме времени и памяти, что достигается в результате оптимизации программной системы.
БЗ содержит 1000—1500 правил, время разработки — 1—1,5 года.
- ⑤ **Коммерческая ЭС** — отличается от промышленной тем, что может продаваться различным потребителям.
БЗ содержит 1500—3000 правил, время разработки — 1,5—3 года, стоимость — 0,3—5 млн. долларов США.

Технология разработки ЭС

Технология разработки ЭС обычно состоит из следующих этапов.

- ❶ **Идентификация (постановка задачи)** — устанавливаются задачи, которые подлежат решению, выявляются цели разработки, требования к ЭС, участники процесса разработки, ресурсы, используемые понятия и взаимосвязи, определяются методы решения задач. Цель этапа — сформулировать задачу, охарактеризовать поддерживающую её БЗ и т.о. обеспечить начальный импульс для развития БЗ.
- ❷ **Концептуализация** — проводится содержательный анализ проблемной области, выявляются используемые понятия и их взаимосвязи, определяются методы решения задач.

Технология разработки ЭС

Технология разработки ЭС обычно состоит из следующих этапов.

- ❶ **Идентификация (постановка задачи)** — устанавливаются задачи, которые подлежат решению, выявляются цели разработки, требования к ЭС, участники процесса разработки, ресурсы, используемые понятия и взаимосвязи, определяются методы решения задач. Цель этапа — сформулировать задачу, охарактеризовать поддерживающую её БЗ и т.о. обеспечить начальный импульс для развития БЗ.
- ❷ **Концептуализация** — проводится содержательный анализ проблемной области, выявляются используемые понятия и их взаимосвязи, определяются методы решения задач.

Технология разработки ЭС

Технология разработки ЭС обычно состоит из следующих этапов.

- ❶ **Идентификация (постановка задачи)** — устанавливаются задачи, которые подлежат решению, выявляются цели разработки, требования к ЭС, участники процесса разработки, ресурсы, используемые понятия и взаимосвязи, определяются методы решения задач. Цель этапа — сформулировать задачу, охарактеризовать поддерживающую её БЗ и т.о. обеспечить начальный импульс для развития БЗ.
- ❷ **Концептуализация** — проводится содержательный анализ проблемной области, выявляются используемые понятия и их взаимосвязи, определяются методы решения задач.

Технология разработки ЭС

Технология разработки ЭС обычно состоит из следующих этапов.

- ❶ **Идентификация (постановка задачи)** — устанавливаются задачи, которые подлежат решению, выявляются цели разработки, требования к ЭС, участники процесса разработки, ресурсы, используемые понятия и взаимосвязи, определяются методы решения задач. Цель этапа — сформулировать задачу, охарактеризовать поддерживающую её БЗ и т.о. обеспечить начальный импульс для развития БЗ.
- ❷ **Концептуализация** — проводится содержательный анализ проблемной области, выявляются используемые понятия и их взаимосвязи, определяются методы решения задач.

Технология разработки ЭС (продолжение)

- ③ **Формализация** — определяются способы представления всех видов знаний, формализуются основные понятия, определяются способы интерпретации знаний, оценивается адекватность целям системы зафиксированных понятий, методов решения, средств представления и манипулирования знаниями.
- ④ **Выполнение** — осуществляется наполнение экспертом БЗ. Процесс приобретения знаний разделяется на извлечение знаний из эксперта, организацию знаний, обеспечивающую эффективную работу системы, и представление знаний в виде, понятном ЭС. Приобретение знаний является очень трудоёмким процессом.

Технология разработки ЭС (продолжение)

- ③ **Формализация** — определяются способы представления всех видов знаний, формализуются основные понятия, определяются способы интерпретации знаний, оценивается адекватность целям системы зафиксированных понятий, методов решения, средств представления и манипулирования знаниями.
- ④ **Выполнение** — осуществляется наполнение экспертом БЗ. Процесс приобретения знаний разделяется на извлечение знаний из эксперта, организацию знаний, обеспечивающую эффективную работу системы, и представление знаний в виде, понятном ЭС. Приобретение знаний является очень трудоёмким процессом.

Технология разработки ЭС (продолжение)

- ③ **Формализация** — определяются способы представления всех видов знаний, формализуются основные понятия, определяются способы интерпретации знаний, оценивается адекватность целям системы зафиксированных понятий, методов решения, средств представления и манипулирования знаниями.
- ④ **Выполнение** — осуществляется наполнение экспертом БЗ. Процесс приобретения знаний разделяется на извлечение знаний из эксперта, организацию знаний, обеспечивающую эффективную работу системы, и представление знаний в виде, понятном ЭС.

Приобретение знаний является очень трудоёмким процессом.

Технология разработки ЭС (продолжение)

- ③ **Формализация** — определяются способы представления всех видов знаний, формализуются основные понятия, определяются способы интерпретации знаний, оценивается адекватность целям системы зафиксированных понятий, методов решения, средств представления и манипулирования знаниями.
- ④ **Выполнение** — осуществляется наполнение экспертом БЗ. Процесс приобретения знаний разделяется на извлечение знаний из эксперта, организацию знаний, обеспечивающую эффективную работу системы, и представление знаний в виде, понятном ЭС. Приобретение знаний является очень трудоёмким процессом.

Технология разработки ЭС (окончание)

- 5 **Тестирование** — эксперт и инженер по знаниям в интерактивном режиме проверяют комплектность ЭС. Тестирование продолжается до тех пор, пока эксперт не решит, что ЭС достигла требуемого уровня комплектности.
- 6 **Опытная эксплуатация** — проверяется пригодность ЭС для конечных пользователей. По результатам этого этапа может потребоваться модификация ЭС.
- 7 **Модификация** — производится почти постоянно при разработке ЭС (возврат на предыдущие этапы и их переработка).

Технология разработки ЭС (окончание)

- 5 **Тестирование** — эксперт и инженер по знаниям в интерактивном режиме проверяют комплектность ЭС. Тестирование продолжается до тех пор, пока эксперт не решит, что ЭС достигла требуемого уровня комплектности.
- 6 **Опытная эксплуатация** — проверяется пригодность ЭС для конечных пользователей.
По результатам этого этапа может потребоваться модификация ЭС.
- 7 **Модификация** — производится почти постоянно при разработке ЭС (возврат на предыдущие этапы и их переработка).

Технология разработки ЭС (окончание)

- 5 **Тестирование** — эксперт и инженер по знаниям в интерактивном режиме проверяют комплектность ЭС. Тестирование продолжается до тех пор, пока эксперт не решит, что ЭС достигла требуемого уровня комплектности.
- 6 **Опытная эксплуатация** — проверяется пригодность ЭС для конечных пользователей.
По результатам этого этапа может потребоваться модификация ЭС.
- 7 **Модификация** — производится почти постоянно при разработке ЭС (возврат на предыдущие этапы и их переработка).

Технология разработки ЭС (окончание)

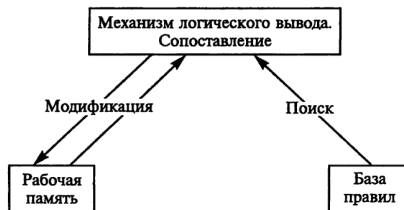
- 5 **Тестирование** — эксперт и инженер по знаниям в интерактивном режиме проверяют комплектность ЭС. Тестирование продолжается до тех пор, пока эксперт не решит, что ЭС достигла требуемого уровня комплектности.
- 6 **Опытная эксплуатация** — проверяется пригодность ЭС для конечных пользователей. По результатам этого этапа может потребоваться модификация ЭС.
- 7 **Модификация** — производится почти постоянно при разработке ЭС (возврат на предыдущие этапы и их переработка).

Технология разработки ЭС (окончание)

- 5 **Тестирование** — эксперт и инженер по знаниям в интерактивном режиме проверяют комплектность ЭС. Тестирование продолжается до тех пор, пока эксперт не решит, что ЭС достигла требуемого уровня комплектности.
- 6 **Опытная эксплуатация** — проверяется пригодность ЭС для конечных пользователей. По результатам этого этапа может потребоваться модификация ЭС.
- 7 **Модификация** — производится почти постоянно при разработке ЭС (возврат на предыдущие этапы и их переработка).

Пример простой продукционной ЭС

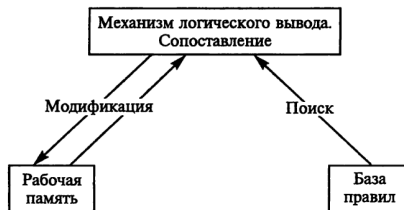
Рассмотрим упрощённую продукционную экспертную систему, которая состоит из следующих компонент:



- 1 База правил (БП) типа ЕСЛИ условие, ТО действие.
- 2 Рабочая память (РП), в которой хранятся исходные данные к задаче и выводы, полученные в ходе работы системы.
- 3 Механизм логического вывода, использующий правила в соответствии с содержимым РП.

Пример простой продукционной ЭС

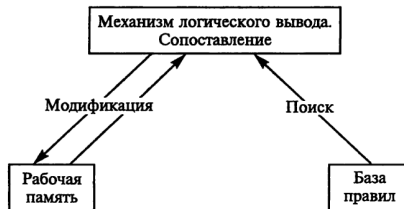
Рассмотрим упрощённую продукционную экспертную систему, которая состоит из следующих компонент:



- 1 База правил (БП) типа ЕСЛИ условие, ТО действие.
- 2 Рабочая память (РП), в которой хранятся исходные данные к задаче и выводы, полученные в ходе работы системы.
- 3 Механизм логического вывода, использующий правила в соответствии с содержимым РП.

Пример простой продукционной ЭС

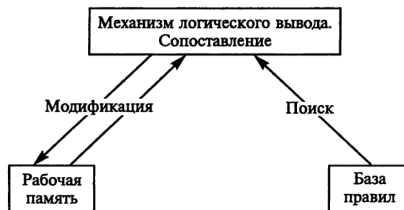
Рассмотрим упрощённую продукционную экспертную систему, которая состоит из следующих компонент:



- 1 База правил (БП) типа ЕСЛИ условие, ТО действие.
- 2 Рабочая память (РП), в которой хранятся исходные данные к задаче и выводы, полученные в ходе работы системы.
- 3 Механизм логического вывода, использующий правила в соответствии с содержимым РП.

Пример простой продукционной ЭС

Рассмотрим упрощённую продукционную экспертную систему, которая состоит из следующих компонент:



- 1 База правил (БП) типа ЕСЛИ условие, ТО действие.
- 2 Рабочая память (РП), в которой хранятся исходные данные к задаче и выводы, полученные в ходе работы системы.
- 3 Механизм логического вывода, использующий правила в соответствии с содержимым РП.

Пример прямого вывода

Вывод знаний — получение новых информационных единиц из ранее известных.

Рассмотрим пример **прямого вывода** — от данных к цели.

Пусть БП имеет следующие правила для решения задачи диагностики:

- 1 ЕСЛИ Двигатель не заводится И Фары не горят,
ТО Сел аккумулятор.
- 2 ЕСЛИ Указатель уровня топлива находится на нуле,
ТО Двигатель не заводится.

Пусть в РП от пользователя поступили факты:

- 1 Фары не горят.
- 2 Указатель уровня топлива находится на нуле.

Пример прямого вывода

Вывод знаний — получение новых информационных единиц из ранее известных.

Рассмотрим пример **прямого вывода** — от данных к цели.

Пусть БП имеет следующие правила для решения задачи диагностики:

- 1 ЕСЛИ Двигатель не заводится И Фары не горят,
ТО Сел аккумулятор.
- 2 ЕСЛИ Указатель уровня топлива находится на нуле,
ТО Двигатель не заводится.

Пусть в РП от пользователя поступили факты:

- 1 Фары не горят.
- 2 Указатель уровня топлива находится на нуле.

Пример прямого вывода

Вывод знаний — получение новых информационных единиц из ранее известных.

Рассмотрим пример **прямого вывода** — от данных к цели.

Пусть БП имеет следующие правила для решения задачи диагностики:

- 1 ЕСЛИ Двигатель не заводится И Фары не горят,
ТО Сел аккумулятор.
- 2 ЕСЛИ Указатель уровня топлива находится на нуле,
ТО Двигатель не заводится.

Пусть в РП от пользователя поступили факты:

- 1 Фары не горят.
- 2 Указатель уровня топлива находится на нуле.

Пример прямого вывода

Вывод знаний — получение новых информационных единиц из ранее известных.

Рассмотрим пример **прямого вывода** — от данных к цели.

Пусть БП имеет следующие правила для решения задачи диагностики:

- 1 ЕСЛИ Двигатель не заводится И Фары не горят,
ТО Сел аккумулятор.
- 2 ЕСЛИ Указатель уровня топлива находится на нуле,
ТО Двигатель не заводится.

Пусть в РП от пользователя поступили факты:

- 1 Фары не горят.
- 2 Указатель уровня топлива находится на нуле.

Пример прямого вывода

Вывод знаний — получение новых информационных единиц из ранее известных.

Рассмотрим пример **прямого вывода** — от данных к цели.

Пусть БП имеет следующие правила для решения задачи диагностики:

- 1 ЕСЛИ Двигатель не заводится И Фары не горят,
ТО Сел аккумулятор.
- 2 ЕСЛИ Указатель уровня топлива находится на нуле,
ТО Двигатель не заводится.

Пусть в РП от пользователя поступили факты:

- 1 Фары не горят.
- 2 Указатель уровня топлива находится на нуле.

Пример прямого вывода (продолжение)

Рассмотрим основные шаги алгоритма прямого вывода.

- 1 Сопоставление фактов из РП с антецедентами правил из БП.

Правило 1 не может сработать, а правило 2 срабатывает, так как факт, совпадающий с его антецедентом, присутствует в РП.

- 2 Выполняется действие сработавшего правила 2 — в РП заносится консеквент этого правила — «Двигатель не заводится».

- 3 Второй цикл сопоставления фактов из РП с антецедентами правил.

Теперь срабатывает правило 1, т. к. конъюнкция в его антецеденте становится истинной.

- 4 Выполняется действие правила 1, которое и выводит окончательный диагноз — «Сел аккумулятор».

- 5 Конец работы (БП исчерпана).

Пример прямого вывода (продолжение)

Рассмотрим основные шаги алгоритма прямого вывода.

- 1 Сопоставление фактов из РП с антецедентами правил из БП.

Правило 1 не может сработать, а правило 2 срабатывает, так как факт, совпадающий с его антецедентом, присутствует в РП.

- 2 Выполняется действие сработавшего правила 2 — в РП заносится консеквент этого правила — «Двигатель не заводится».

- 3 Второй цикл сопоставления фактов из РП с антецедентами правил.

Теперь срабатывает правило 1, т. к. конъюнкция в его антецеденте становится истинной.

- 4 Выполняется действие правила 1, которое и выводит окончательный диагноз — «Сел аккумулятор».

- 5 Конец работы (БП исчерпана).

Пример прямого вывода (продолжение)

Рассмотрим основные шаги алгоритма прямого вывода.

- 1 Сопоставление фактов из РП с антецедентами правил из БП.

Правило 1 не может сработать, а правило 2 срабатывает, так как факт, совпадающий с его антецедентом, присутствует в РП.

- 2 Выполняется действие сработавшего правила 2 — в РП заносится консеквент этого правила — «Двигатель не заводится».

- 3 Второй цикл сопоставления фактов из РП с антецедентами правил.

Теперь срабатывает правило 1, т. к. конъюнкция в его антецеденте становится истинной.

- 4 Выполняется действие правила 1, которое и выводит окончательный диагноз — «Сел аккумулятор».

- 5 Конец работы (БП исчерпана).

Пример прямого вывода (продолжение)

Рассмотрим основные шаги алгоритма прямого вывода.

- 1 Сопоставление фактов из РП с антецедентами правил из БП.

Правило 1 не может сработать, а правило 2 срабатывает, так как факт, совпадающий с его антецедентом, присутствует в РП.

- 2 Выполняется действие сработавшего правила 2 — в РП заносится консеквент этого правила — «Двигатель не заводится».

- 3 Второй цикл сопоставления фактов из РП с антецедентами правил.

Теперь срабатывает правило 1, т. к. конъюнкция в его антецеденте становится истинной.

- 4 Выполняется действие правила 1, которое и выводит окончательный диагноз — «Сел аккумулятор».

- 5 Конец работы (БП исчерпана).

Пример прямого вывода (продолжение)

Рассмотрим основные шаги алгоритма прямого вывода.

- 1 Сопоставление фактов из РП с антецедентами правил из БП.

Правило 1 не может сработать, а правило 2 срабатывает, так как факт, совпадающий с его антецедентом, присутствует в РП.

- 2 Выполняется действие сработавшего правила 2 — в РП заносится консеквент этого правила — «Двигатель не заводится».

- 3 Второй цикл сопоставления фактов из РП с антецедентами правил.

Теперь срабатывает правило 1, т. к. конъюнкция в его антецеденте становится истинной.

- 4 Выполняется действие правила 1, которое и выводит окончательный диагноз — «Сел аккумулятор».

- 5 Конец работы (БП исчерпана).

Пример прямого вывода (продолжение)

Рассмотрим основные шаги алгоритма прямого вывода.

- 1 Сопоставление фактов из РП с антецедентами правил из БП.

Правило 1 не может сработать, а правило 2 срабатывает, так как факт, совпадающий с его антецедентом, присутствует в РП.

- 2 Выполняется действие сработавшего правила 2 — в РП заносится консеквент этого правила — «Двигатель не заводится».

- 3 Второй цикл сопоставления фактов из РП с антецедентами правил.

Теперь срабатывает правило 1, т. к. конъюнкция в его антецеденте становится истинной.

- 4 Выполняется действие правила 1, которое и выводит окончательный диагноз — «Сел аккумулятор».

- 5 Конец работы (БП исчерпана).

Пример прямого вывода (продолжение)

Рассмотрим основные шаги алгоритма прямого вывода.

- 1 Сопоставление фактов из РП с антецедентами правил из БП.

Правило 1 не может сработать, а правило 2 срабатывает, так как факт, совпадающий с его антецедентом, присутствует в РП.

- 2 Выполняется действие сработавшего правила 2 — в РП заносится консеквент этого правила — «Двигатель не заводится».

- 3 Второй цикл сопоставления фактов из РП с антецедентами правил.

Теперь срабатывает правило 1, т. к. конъюнкция в его антецеденте становится истинной.

- 4 Выполняется действие правила 1, которое и выводит окончательный диагноз — «Сел аккумулятор».

- 5 Конец работы (БП исчерпана).

Пример обратного вывода

Кроме прямого, применяют и **обратный вывод** — от цели для её подтверждения к данным.

Пусть БД и РП содержат ту же информацию, что и в начале предыдущего примера.

Алгоритм обратного вывода содержит следующие шаги.

1. Выдвигается гипотеза окончательного диагноза — «Сел аккумулятор».
2. Отыскивается правило, консеквент которого соответствует гипотезе — в данном примере это правило 1.
3. Решается, может ли это правило сработать. Для этого в РП должны присутствовать факты, совпадающие с фактами из антецедента правила. В данном примере правило 1 сработать не может из-за отсутствия в РП факта «Двигатель не заводится». Этот факт становится новой гипотезой.

Пример обратного вывода

Кроме прямого, применяют и **обратный вывод** — от цели для её подтверждения к данным.

Пусть БД и РП содержат ту же информацию, что и в начале предыдущего примера.

Алгоритм обратного вывода содержит следующие шаги.

- 1 Вывдвигается гипотеза окончательного диагноза — «Сел аккумулятор».
- 2 Отыскивается правило, консеквент которого соответствует гипотезе — в данном примере это правило 1.
- 3 Решается, может ли это правило сработать. Для этого в РП должны присутствовать факты, совпадающие с фактами из антецедента правила. В данном примере правило 1 сработать не может из-за отсутствия в РП факта «Двигатель не заводится». Этот факт становится новой гипотезой.

Пример обратного вывода

Кроме прямого, применяют и **обратный вывод** — от цели для её подтверждения к данным.

Пусть БД и РП содержат ту же информацию, что и в начале предыдущего примера.

Алгоритм обратного вывода содержит следующие шаги.

- 1 Вывдвигается гипотеза окончательного диагноза — «Сел аккумулятор».
- 2 Отыскивается правило, консеквент которого соответствует гипотезе — в данном примере это правило 1.
- 3 Решается, может ли это правило сработать. Для этого в РП должны присутствовать факты, совпадающие с фактами из антецедента правила. В данном примере правило 1 сработать не может из-за отсутствия в РП факта «Двигатель не заводится». Этот факт становится новой гипотезой.

Пример обратного вывода

Кроме прямого, применяют и **обратный вывод** — от цели для её подтверждения к данным.

Пусть БД и РП содержат ту же информацию, что и в начале предыдущего примера.

Алгоритм обратного вывода содержит следующие шаги.

- 1 Вывдвигается гипотеза окончательного диагноза — «Сел аккумулятор».
- 2 Отыскивается правило, консеквент которого соответствует гипотезе — в данном примере это правило 1.
- 3 Решается, может ли это правило сработать. Для этого в РП должны присутствовать факты, совпадающие с фактами из антецедента правила. В данном примере правило 1 сработать не может из-за отсутствия в РП факта «Двигатель не заводится». Этот факт становится новой гипотезой.

Пример обратного вывода

Кроме прямого, применяют и **обратный вывод** — от цели для её подтверждения к данным.

Пусть БД и РП содержат ту же информацию, что и в начале предыдущего примера.

Алгоритм обратного вывода содержит следующие шаги.

- 1 Вывдвигается гипотеза окончательного диагноза — «Сел аккумулятор».
- 2 Отыскивается правило, консеквент которого соответствует гипотезе — в данном примере это правило 1.
- 3 Решается, может ли это правило сработать. Для этого в РП должны присутствовать факты, совпадающие с фактами из антецедента правила. В данном примере правило 1 сработать не может из-за отсутствия в РП факта «Двигатель не заводится». Этот факт становится новой гипотезой.

Пример обратного вывода

Кроме прямого, применяют и **обратный вывод** — от цели для её подтверждения к данным.

Пусть БД и РП содержат ту же информацию, что и в начале предыдущего примера.

Алгоритм обратного вывода содержит следующие шаги.

- 1 Выводится гипотеза окончательного диагноза — «Сел аккумулятор».
- 2 Отыскивается правило, консеквент которого соответствует гипотезе — в данном примере это правило 1.
- 3 Решается, может ли это правило сработать. Для этого в РП должны присутствовать факты, совпадающие с фактами из антецедента правила. В данном примере правило 1 сработать не может из-за отсутствия в РП факта «Двигатель не заводится». Этот факт становится новой гипотезой.

Пример обратного вывода (второй этап)

- ❶ Поиск правила, консеквент которого соответствует новой гипотезе — находится правило 2.
- ❷ Решается, может ли это правило сработать. Оно срабатывает, т. к. в РП присутствует факт, совпадающий с его антецедентом.
- ❸ Выполняется действие правила 2 — в РП заносится факт «Двигатель не заводится».
- ❹ Правило 1 теперь срабатывает, и выдвинутая в начале гипотеза подтверждается.
- ❺ Конец работы.

Преимущества обратного вывода в задачах диагностики становятся очевидными при наличии в БП конфликтующих правил.

Пример обратного вывода (второй этап)

- ❶ Поиск правила, консеквент которого соответствует новой гипотезе — находится правило 2.
- ❷ Решается, может ли это правило сработать. Оно срабатывает, т. к. в РП присутствует факт, совпадающий с его антецедентом.
- ❸ Выполняется действие правила 2 — в РП заносится факт «Двигатель не заводится».
- ❹ Правило 1 теперь срабатывает, и выдвинутая в начале гипотеза подтверждается.
- ❺ Конец работы.

Преимущества обратного вывода в задачах диагностики становятся очевидными при наличии в БП конфликтующих правил.

Пример обратного вывода (второй этап)

- 4 Поиск правила, консеквент которого соответствует новой гипотезе — находится правило 2.
- 5 Решается, может ли это правило сработать. Оно срабатывает, т. к. в РП присутствует факт, совпадающий с его антецедентом.
- 6 Выполняется действие правила 2 — в РП заносится факт «Двигатель не заводится».
- 7 Правило 1 теперь срабатывает, и выдвинутая в начале гипотеза подтверждается.
- 8 Конец работы.

Преимущества обратного вывода в задачах диагностики становятся очевидными при наличии в БП конфликтующих правил.

Пример обратного вывода (второй этап)

- ④ Поиск правила, консеквент которого соответствует новой гипотезе — находится правило 2.
- ⑤ Решается, может ли это правило сработать. Оно срабатывает, т. к. в РП присутствует факт, совпадающий с его антецедентом.
- ⑥ Выполняется действие правила 2 — в РП заносится факт «Двигатель не заводится».
- ⑦ Правило 1 теперь срабатывает, и выдвинутая в начале гипотеза подтверждается.
- ⑧ Конец работы.

Преимущества обратного вывода в задачах диагностики становятся очевидными при наличии в БП конфликтующих правил.

Пример обратного вывода (второй этап)

- ④ Поиск правила, консеквент которого соответствует новой гипотезе — находится правило 2.
- ⑤ Решается, может ли это правило сработать. Оно срабатывает, т. к. в РП присутствует факт, совпадающий с его антецедентом.
- ⑥ Выполняется действие правила 2 — в РП заносится факт «Двигатель не заводится».
- ⑦ Правило 1 теперь срабатывает, и выдвинутая в начале гипотеза подтверждается.
- ⑧ Конец работы.

Преимущества обратного вывода в задачах диагностики становятся очевидными при наличии в БП конфликтующих правил.

Пример обратного вывода (второй этап)

- ④ Поиск правила, консеквент которого соответствует новой гипотезе — находится правило 2.
- ⑤ Решается, может ли это правило сработать. Оно срабатывает, т. к. в РП присутствует факт, совпадающий с его антецедентом.
- ⑥ Выполняется действие правила 2 — в РП заносится факт «Двигатель не заводится».
- ⑦ Правило 1 теперь срабатывает, и выдвинутая в начале гипотеза подтверждается.
- ⑧ Конец работы.

Преимущества обратного вывода в задачах диагностики становятся очевидными при наличии в БП конфликтующих правил.

Методы вывода знаний

Методами вывода знаний, кроме рассмотренных двух, могут быть также следующие:

- абдуктивный (от частного к частному),
- индуктивный (от частного к общему),
- дедуктивный (от общего к частному),
- естественный (на основе «здравого смысла»),
- вывод по аналогии (перенесение рассуждений из одной проблемной области в другую, похожую),
- правдоподобный (на каждом шаге вычисляется оценка достоверности полученного утверждения) и др.

В последнее время для представления и вывода знаний чаще всего используют нечёткие множества и нечёткую логику.

Методы вывода знаний

Методами вывода знаний, кроме рассмотренных двух, могут быть также следующие:

- **абдуктивный** (от частного к частному),
- индуктивный (от частного к общему),
- дедуктивный (от общего к частному),
- естественный (на основе «здравого смысла»),
- вывод по аналогии (перенесение рассуждений из одной проблемной области в другую, похожую),
- правдоподобный (на каждом шаге вычисляется оценка достоверности полученного утверждения) и др.

В последнее время для представления и вывода знаний чаще всего используют нечёткие множества и нечёткую логику.

Методы вывода знаний

Методами вывода знаний, кроме рассмотренных двух, могут быть также следующие:

- абдуктивный (от частного к частному),
- индуктивный (от частного к общему),
- дедуктивный (от общего к частному),
- естественный (на основе «здравого смысла»),
- вывод по аналогии (перенесение рассуждений из одной проблемной области в другую, похожую),
- правдоподобный (на каждом шаге вычисляется оценка достоверности полученного утверждения) и др.

В последнее время для представления и вывода знаний чаще всего используют нечёткие множества и нечёткую логику.

Методы вывода знаний

Методами вывода знаний, кроме рассмотренных двух, могут быть также следующие:

- абдуктивный (от частного к частному),
- индуктивный (от частного к общему),
- дедуктивный (от общего к частному),
- естественный (на основе «здравого смысла»),
- вывод по аналогии (перенесение рассуждений из одной проблемной области в другую, похожую),
- правдоподобный (на каждом шаге вычисляется оценка достоверности полученного утверждения) и др.

В последнее время для представления и вывода знаний чаще всего используют нечёткие множества и нечёткую логику.

Методы вывода знаний

Методами вывода знаний, кроме рассмотренных двух, могут быть также следующие:

- абдуктивный (от частного к частному),
- индуктивный (от частного к общему),
- дедуктивный (от общего к частному),
- естественный (на основе «здравого смысла»),
- вывод по аналогии (перенесение рассуждений из одной проблемной области в другую, похожую),
- правдоподобный (на каждом шаге вычисляется оценка достоверности полученного утверждения) и др.

В последнее время для представления и вывода знаний чаще всего используют нечёткие множества и нечёткую логику.

Методы вывода знаний

Методами вывода знаний, кроме рассмотренных двух, могут быть также следующие:

- абдуктивный (от частного к частному),
- индуктивный (от частного к общему),
- дедуктивный (от общего к частному),
- естественный (на основе «здравого смысла»),
- вывод по аналогии (перенесение рассуждений из одной проблемной области в другую, похожую),
- правдоподобный (на каждом шаге вычисляется оценка достоверности полученного утверждения) и др.

В последнее время для представления и вывода знаний чаще всего используют нечёткие множества и нечёткую логику.

Методы вывода знаний

Методами вывода знаний, кроме рассмотренных двух, могут быть также следующие:

- **абдуктивный** (от частного к частному),
- **индуктивный** (от частного к общему),
- **дедуктивный** (от общего к частному),
- **естественный** (на основе «здравого смысла»),
- **вывод по аналогии** (перенесение рассуждений из одной проблемной области в другую, похожую),
- **правдоподобный** (на каждом шаге вычисляется оценка достоверности полученного утверждения) и др.

В последнее время для представления и вывода знаний чаще всего используют нечёткие множества и нечёткую логику.

Методы вывода знаний

Методами вывода знаний, кроме рассмотренных двух, могут быть также следующие:

- **абдуктивный** (от частного к частному),
- **индуктивный** (от частного к общему),
- **дедуктивный** (от общего к частному),
- **естественный** (на основе «здравого смысла»),
- **вывод по аналогии** (перенесение рассуждений из одной проблемной области в другую, похожую),
- **правдоподобный** (на каждом шаге вычисляется оценка достоверности полученного утверждения) и др.

В последнее время для представления и вывода знаний чаще всего используют нечёткие множества и нечёткую логику.