

Интеллектуальные информационные системы (ИИС)

Лекция 3

Логическая модель представления знаний

Знание и информация

Знания – это результат обработки информации умом, включающий понимание и интерпретацию.

Понятийные знания - набор понятий, используемых при решении данной задачи, например, в фундаментальных науках и теоретических областях наук, т.е. это понятийный аппарат науки.

Конструктивные знания - наборы структур, подсистем системы и взаимодействий между их элементами, например, в технике.

Процедурные знания - методы, процедуры (алгоритмы) их реализации и идентификации, например, в прикладных науках.

Фактографические - количественные и качественные характеристики объектов и явлений, например, в экспериментальных науках.

Метазнания - знания о порядке и правилах применения знаний (знания о знаниях).

Информация – это данные, представленные в структурированной форме, которые могут быть обработаны человеком или вычислительной техникой.

Логическая модель

Логическая модель представления знаний описывает, как знания могут быть формализованы и представлены с использованием формальных логических средств.

Формальные логические средства включают символы, правила инференции (умозаключения) и языки для представления знаний.

Пример

Пример: Логические символы - операторы И, ИЛИ, НЕ.

- Высказывание А: "Солнце светит".

- Высказывание В: «Сейчас день».

Логическое И (AND):

А И В = Истина (True) только если оба высказывания А и В истинны. В противном случае - Ложь (False).

А И В = "Солнце светит" И " Сейчас день " = Истина.

Логическое ИЛИ (OR):

А ИЛИ В = Истина, если хотя бы одно из высказываний А или В истинно.

А ИЛИ В = "Солнце светит" ИЛИ " Сейчас день " = Истина.

Логическое НЕ (NOT):

НЕ А = Истина, если А ложно, и наоборот.

НЕ А = НЕ "Солнце светит" = Ложь.

Пример формулы в логической модели представления знаний:

IF (Солнце светит) AND (Сейчас день) THEN (На улице хорошо).

В данной формуле мы используем операторы И/ИЛИ для объединения условий и вывода знаний.

Символы и символьное представление знаний

В логической модели представления знаний, знания и информация часто представляются с использованием символов и символьных выражений.

Символы представляют абстрактные объекты или понятия, которые имеют семантический смысл.

Пример

Примеры символьного представления знаний:

1. Логический символ "P":

- Представляет утверждение или высказывание. Например, "P" может означать "Солнце светит".

2 . Символы констант:

- Могут представлять конкретные объекты или значения. Например, "Солнце", "Сентябрь".

3. Символы функций:

- Позволяют создавать выражения, использующие функции или отношения. Например, "Светит (Солнце)".

4. Кванторы:

- Символы " \exists " (существует) и " \forall " (для всех) используются для создания выражений с кванторами, позволяющими говорить о существовании или универсальности объектов.

5. Операторы:

- Операторы, такие как "И" (AND), "ИЛИ" (OR), "НЕ" (NOT), используются для создания сложных логических выражений.

Символьное представление знаний включает создание высказываний, формул и правил, которые могут быть использованы для логического вывода и принятия решений в Интеллектуальных информационных системах.

Пример символьного представления знаний:

IF (Солнце светит) AND (День сегодня) THEN (На улице хорошо).

В этом примере, "Солнце светит" и "День сегодня" - это символы, представляющие состояния, а "На улице хорошо" - это символ, представляющий вывод знаний. Такие символьные выражения могут быть использованы для автоматического принятия решений в ИИС на основе имеющихся знаний.

Формальные языки для представления знаний

Формальные языки представляют собой системы символов и правил, используемые для описания и формализации знаний и информации. Формальные языки представления знаний:

1. Логический язык предикатов (Predicate Logic):

- Является одним из самых распространенных формальных языков для представления знаний в ИИС.

- Включает символы для предикатов, переменных, кванторов (существования и всеобщности) и логических операторов.

- Позволяет выражать утверждения в виде логических формул.

Пример формулы в логическом языке предикатов:

$$\exists x (\text{Светит}(x) \wedge \text{День}(x)) \rightarrow \text{Хорошо}(x)$$

(Существует такой x , что если светит и день, то хорошо).

2. OWL (Web Ontology Language):

- Используется для создания онтологий и представления знаний в семантическом вебе.
- Включает классы, свойства, индивидуальные экземпляры и аксиомы для описания отношений между объектами.

Пример OWL-аксиомы:

Класс "Человек" подкласс "Животное"

(Экземпляры класса "Человек" также являются экземплярами класса "Животное").

3. RDF (Resource Description Framework):

- Используется для описания ресурсов и их отношений в семантическом вебе.
- Основан на тройках, состоящих из субъекта, предиката и объекта.

Пример RDF-тройки:

(Субъект: "John", Предикат: "работает", Объект: "в компании А")

(John работает в компании А).

4. Продукционные правила:

- Используются для описания правил вывода на основе условий и действий.
- Формулируются в виде "IF [условие] THEN [действие]".

Пример продукционного правила:

IF (Температура > 30°C) THEN (Включить кондиционер)

5. Концептуальные графы:

- Используются для представления знаний в виде графов, где узлы представляют сущности, а ребра - отношения между ними.



Различные формальные языки обладают разными выразительными способностями и применяются в зависимости от конкретных задач в ИИС.

Понятие семантики в логическом представлении знаний

Семантика в логическом представлении знаний означает значение и интерпретацию символов, выражений и правил в данной логической системе. Семантика определяет, какие истинные факты и законы реального мира соответствуют символам и выражениям в формальной системе.

Семантики в различных логических языках:

1. Логический язык предикатов (Predicate Logic):

- Семантика этого языка определяется при помощи интерпретации, которая связывает символы и предикаты с доменами их значений.
- Семантика кванторов " \exists " (существует) и " \forall " (для всех) указывает на существование и всеобщность объектов в домене.

Пример интерпретации:

Пусть у нас есть домен $D = \{a, b, c\}$, и предикат $P(x) = "x - человек"$.

Тогда формула $\forall x P(x)$ будет истинна, если все элементы домена D являются людьми, и ложна в противном случае.

2. OWL (Web Ontology Language):

- Семантика OWL определена через классы, свойства и индивидуальные экземпляры.
- Она предоставляет формальные ограничения и правила для интерпретации онтологий.

Пример семантики:

Если класс "Птица" является подклассом "Животное" в онтологии, это означает, что все птицы также являются животными.

3. **RDF** (Resource Description Framework):

- Семантика RDF основана на идее о тройках, где субъект, предикат и объект образуют утверждение.
- Интерпретация RDF описывает, какие субъекты, предикаты и объекты представляют собой в реальном мире.

Пример интерпретации:

Тройка (субъект: "John", предикат: "работает", объект: "в компании А") означает, что John работает в компании А.

4. **Продукционные правила:**

- Семантика продукционных правил заключается в их логическом следствии. Если условие правила истинно, то выполняется действие.

Пример семантики:

IF (Температура > 30°C) THEN (Включить кондиционер).

Если температура выше 30°C, то кондиционер включается.

5. Концептуальные графы:

- Семантика концептуальных графов описывает отношения между узлами (сущностями) и ребрами (отношениями) в графе.
- Это позволяет интерпретировать, какие концепции и связи существуют в реальной ситуации.

Пример интерпретации:

Если в концептуальном графе есть узел "Автомобиль" и ребро "имеет модель" к узлу "Toyota Camry", это означает, что автомобиль Toyota Camry имеет модель.

Семантика играет важную роль в логическом представлении знаний, так как она определяет, какие утверждения и выводы являются истинными или ложными в данной логической системе, что является основой для принятия решений и инференции в информационных системах.

Представление знаний в форме утверждений

Логические высказывания представляют собой утверждения или фразы, которые могут быть либо истинными (правдивыми), либо ложными (неправдивыми).

В логическом представлении знаний, логические высказывания используются для формализации фактов и условий.

Примеры логических высказываний:

1. "Солнце светит":

- Это утверждение может быть истинным или ложным в зависимости от времени суток и погодных условий.

2. "День сегодня":

- Это утверждение также может быть истинным или ложным, и оно зависит от конкретного дня.

3. "Температура воздуха выше 30°C":

- Это утверждение истинно, если текущая температура воздуха превышает 30°C, и ложно в противном случае.

4. "Автомобиль Toyota Camry":

- Утверждение указывает на существование автомобиля определенной марки и модели.

5. "Человек работает в компании А":

- Это утверждение описывает трудовую связь между человеком и компанией.

Основные операторы логических высказываний

Оператор "И" (AND):

Обозначается как " \wedge " или "&&".

Истинное логическое И утверждение будет истинным только в том случае, если оба операнда истинны. В противном случае, оно будет ложным.

Пример:

"Солнце светит" \wedge "День сегодня" - это истинное утверждение, если и солнце светит, и сегодня день.

Оператор "ИЛИ" (OR):

Обозначается как " \vee " или "||".

Логическое ИЛИ утверждение будет истинным, если хотя бы один из операндов истинен.

Пример:

"Солнце светит" \vee "Дождь идет" - это истинное утверждение, если хотя бы одно из условий выполняется.

Оператор "НЕ" (NOT):

Обозначается как " \neg " или "!".

Логическое НЕ инвертирует истинность операнда. Если операнд истинен, то результат будет ложным, и наоборот.

Пример:

\neg "Солнце светит" - это ложное утверждение, если солнце действительно светит.

Кванторы (существование, всеобщность)

Кванторы используются для создания высказываний, связанных с существованием и всеобщностью объектов в домене.

Квантор \exists (существование):

Обозначается как " \exists " и означает "существует".

Выражение с квантором \exists указывает на наличие хотя бы одного объекта в домене, для которого высказывание истинно.

Пример:

$\exists x (\text{Человек}(x) \wedge \text{Работает}(x, \text{"Компания А"}))$ - это утверждение означает, что существует человек, который работает в "Компании А".

Квантор \forall (всеобщность):

Обозначается как " \forall " и означает "для всех" или "все".

Выражение с квантором \forall указывает на истинность утверждения для всех объектов в домене.

Пример:

$\forall x (\text{Человек}(x) \rightarrow \text{Возраст}(x) > 18)$ - это утверждение означает, что для всех людей возраст больше 18 лет.

Кванторы существования и всеобщности позволяют формализовать сложные логические высказывания, учитывая количественные аспекты и отношения между объектами в информационных системах и ИИС. Они часто используются в математической логике и логическом представлении знаний.

Онтология как логическая модель

Онтология представляет собой формальную логическую модель, которая описывает знания о концепциях, сущностях, отношениях и свойствах в некоторой предметной области. Онтология определяет структуру этой области и способы взаимодействия между элементами в ней.

Ключевые элементы определения онтологии:

1. **Формальная модель:** Онтология представляется в форме, которая позволяет формализовать знания с использованием логических и математических выражений. Это может включать символы, правила, ограничения и аксиомы.
2. **Предметная область:** Онтология фокусируется на определенной области знаний или предметной области, в которой она используется для описания и структурирования информации.
3. **Концепции и сущности** Онтология включает в себя определения концепций и сущностей, которые являются важными для данной предметной области. Эти концепции могут включать в себя классы, типы, объекты и т.д.
4. **Отношения и свойства:** Онтология описывает отношения и свойства, которые могут существовать между концепциями и сущностями в предметной области. Эти отношения могут быть иерархическими, частичными порядками, ассоциациями и т.д.
5. **Интерпретация:** Онтология имеет семантическую интерпретацию, которая определяет, какие знания и утверждения о предметной области могут быть сформулированы с использованием этой модели.

Онтологии играют важную роль в области искусственного интеллекта, семантического веба, баз данных и информационных систем, так как они позволяют структурировать и формализовать знания, что в свою очередь упрощает поиск, анализ, интеграцию и использование информации в различных приложениях и доменах.

Основные компоненты онтологии

Основные компоненты онтологии включают в себя классы, свойства и инстансы. Давайте более подробно разберем каждый из них:

1. Классы (Classes):

Классы в онтологии представляют собой группы или категории сущностей, которые имеют общие характеристики или свойства. Классы описывают концепции или типы объектов в предметной области.

Классы используются для структурирования и организации информации в онтологии.

Пример: Класс "Автомобиль" может включать в себя все автомобили в предметной области.

2. Свойства (Properties):

Свойства определяют отношения между классами и инстансами (экземплярами) в онтологии. Свойства могут быть двух типов: объектные (object properties) и данных (data properties).

Объектные свойства описывают отношения между инстансами классов. Например, свойство "имеет владельца" устанавливает отношение между экземпляром класса "Автомобиль" и экземпляром класса "Человек".

Данные свойства ассоциируют классы или инстансы с данными, такими как числа или строки. Например, свойство "год выпуска" может ассоциировать класс "Автомобиль" с числом, представляющим год выпуска.

Свойства используются для описания атрибутов и отношений между объектами в предметной области.

3. Инстансы (Instances):

Инстансы или экземпляры представляют конкретные объекты или сущности, которые относятся к определенным классам в онтологии.

Инстансы представляют конкретные факты или объекты, которые существуют в предметной области и которые можно описать с использованием классов и свойств.

Пример: Если "Автомобиль" - это класс в онтологии, то конкретный автомобиль, например, "Toyota Camry", будет являться инстансом этого класса.

Классы определяют типы сущностей, свойства описывают их атрибуты и отношения, а инстансы представляют конкретные экземпляры в предметной области. Онтологии широко используются для организации и формализации знаний в информационных системах, семантическом вебе, искусственном интеллекте и других областях.

Популярные онтологические модели

1. OWL (Web Ontology Language):

OWL является мощным и широко используемым стандартом для создания онтологий в семантическом вебе и других областях. Он предоставляет формальную спецификацию классов, свойств, индивидуальных экземпляров и правил для создания сложных онтологических моделей.

Разновидности OWL включают OWL Lite, OWL DL и OWL Full, каждая из которых предоставляет различные уровни выразительности и семантики.

2. RDF (Resource Description Framework):

RDF является стандартом для представления знаний и метаданных в веб-среде. Он представляет данные в форме троек (субъект-предикат-объект) и может быть использован для создания онтологий, хотя его выразительность ограничена по сравнению с OWL.

3. RDFS (RDF Schema):

RDFS является расширением RDF, предоставляющим более высокий уровень выразительности для создания онтологий. Он позволяет определять классы, свойства, иерархии и домен-кодоменные ограничения.

4. SKOS (Simple Knowledge Organization System) или антология тегов:

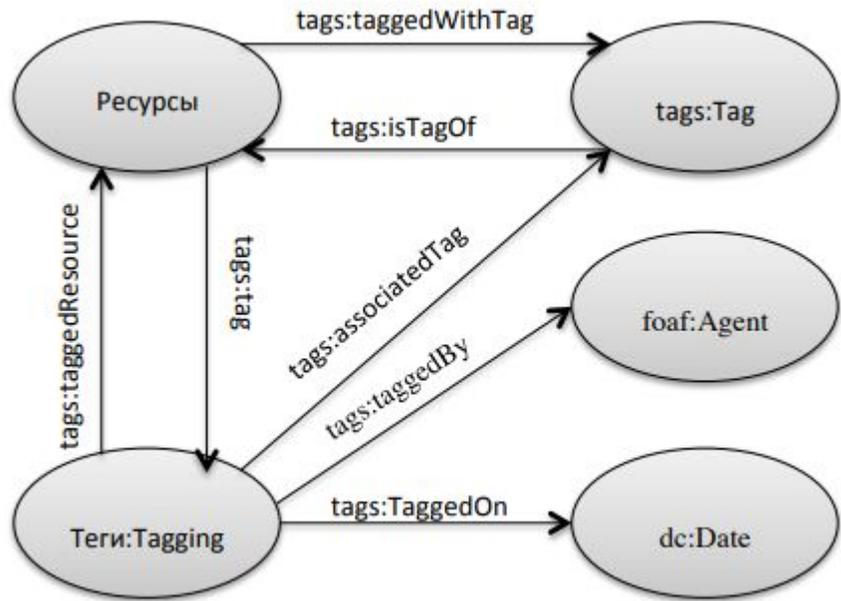
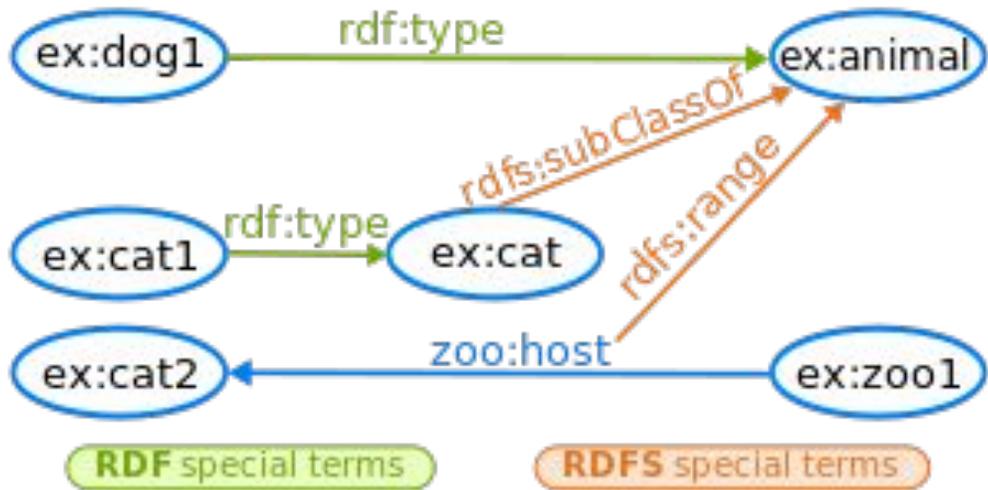
SKOS предназначен для представления тезаурусов, классификаций и терминологических словарей. Он используется для структурирования и организации терминологии и понятий в информационных системах.

5. Dublin Core:

Dublin Core предоставляет простую онтологическую модель для описания метаданных, таких как заголовки, авторы, ключевые слова и другие атрибуты для ресурсов в сети.

6. FOAF (Friend of a Friend):

FOAF представляет собой онтологию, разработанную для описания социальных сетей и связей между людьми в сети. Она включает в себя понятия, такие как "друг", "профиль" и "группа".



Группа	Атрибут	Спецификация FOAF
Основные атрибуты FOAF	Имя	<foaf.name>
		<foaf.familyName>
		<foaf.firstName>
		<foaf.givenName>
		<foaf.lastName>
		<foaf.nick>
	Email	<foaf.mbox>
Фото	<foaf.img>	
Пол	<foaf.gender>	
Возраст	<foaf.age>	
Личная информация	Дата рождения	<foaf.birthday>
	Блог	<foaf.weblog>
	Интерес	<foaf.interest>
		<foaf.topic_interest>
	Публикация	<foaf.publication>
Отношение	<foaf.knows>	
Проекты и группы	Проект	<foaf.project>
		<foaf.currentProject>
		<foaf.pastProject>
	Группа	<foaf.group>
Организация	<foaf.organization>	

Преимущества логической модели представления знаний

1. **Формальность и точность:** Логическая модель представления знаний предоставляет формальные и точные способы определения знаний и их отношений. Это позволяет избежать двусмысленности и неоднозначности в интерпретации информации.
2. **Структурирование знаний:** Онтологии и другие логические модели позволяют структурировать знания в иерархиях классов и определять отношения между ними. Это упрощает организацию и поиск информации.
3. **Интеграция данных:** Логические модели могут использоваться для интеграции данных из различных источников. Они позволяют объединять и сопоставлять информацию из разных систем и источников.
4. **Легкость обновления и расширения:** Онтологии могут быть легко обновлены и расширены с добавлением новых классов, свойств и экземпляров. Это делает их гибкими в использовании.
5. **Поддержка логического вывода:** Логические модели могут использоваться для автоматического логического вывода и решения задач на основе знаний. Это полезно в системах искусственного интеллекта и автоматизированных системах принятия решений.

Ограничения логической модели представления знаний

1. **Сложность разработки:** Создание и поддержка логических моделей может быть сложной задачей. Требуется экспертное знание в предметной области и логике для построения эффективных онтологий.
2. **Не всегда подходит:** Логические модели не всегда являются лучшим выбором для всех типов знаний. В некоторых случаях более подходящими могут быть структуры данных, например, графовые базы данных.
3. **Выразительность:** Несмотря на высокую выразительность некоторых логических моделей, некоторые концепции и отношения могут оказаться сложными для формализации, особенно в больших и сложных предметных областях.
4. **Трудности в интероперабельности:** Интеграция онтологий и логических моделей из разных источников может быть сложной из-за различий в используемых языках и семантике.
5. **Трудности в обновлении:** При изменении предметной области или добавлении новых данных может потребоваться пересматривать и обновлять онтологию, что может быть затратным процессом.

Применение логической модели представления знаний в области ИИ

- 1. Семантический веб (Semantic Web):** Логические модели, такие как OWL, играют важную роль в семантическом вебе. Они позволяют структурировать и описывать информацию в сети, что облегчает поиск и интеграцию данных, а также обеспечивает машинную обработку информации.
- 2. Экспертные системы:** Логические модели используются для создания экспертных систем, которые могут принимать решения и делать выводы на основе формализованных знаний. Это применяется в медицине, финансах, инженерии и других областях.
- 3. Обработка естественного языка (Natural Language Processing, NLP):** Онтологии и логические модели применяются в NLP для семантического анализа текста, извлечения информации и понимания смысла предложений. Это улучшает качество обработки текстов и ответов на запросы.
- 4. Рекомендательные системы:** Логические модели используются для построения рекомендательных систем, которые анализируют предпочтения пользователей и рекомендуют им товары, услуги или контент на основе формализованных знаний о предметной области и пользователях.

- 5. Системы управления знанием:** Логические модели применяются в системах управления знанием организаций. Они помогают организовывать и структурировать знания о бизнес-процессах, продуктах и клиентах.
- 6. Автоматическое планирование и решение задач:** В ИИ, логические модели используются для автоматического планирования и решения задач. Они позволяют моделировать предметные области и применять логический вывод для нахождения оптимальных решений.
- 7. Робототехника и автономные системы:** Логические модели используются в программировании роботов и автономных систем. Они помогают роботам анализировать окружающую среду, принимать решения и планировать действия.
- 8. Медицинская диагностика:** В медицинской области, логические модели применяются для создания систем поддержки принятия решений и диагностики на основе знаний в медицине.

Применение в сфере информационных систем и баз данных

- 1. Семантические базы данных:** Логические модели, включая онтологии, используются для создания семантических баз данных. Это позволяет представлять данные с учетом их смысла и семантики, что улучшает эффективность поиска и запросов.
- 2. Интеграция данных:** В мире большого объема данных с различных источников, логические модели используются для интеграции, сопоставления и объединения данных. Они позволяют создавать общую семантическую модель для данных из разных источников.
- 3. Управление данными и метаданными:** Логические модели используются для управления данными и метаданными в информационных системах и базах данных. Они описывают структуру и связи между данными, что упрощает их администрирование и обслуживание.
- 4. Бизнес-процессы и системы управления:** Логические модели представляют знания о бизнес-процессах и внутренних операциях организации. Это полезно для автоматизации и оптимизации бизнес-процессов.

5. Биоинформатика: В биоинформатике логические модели используются для описания геномов, молекулярных взаимодействий и биологических путей. Это позволяет биологам и исследователям анализировать и понимать биологические процессы.

6. Каталоги и тезаурусы: Логические модели, такие как онтологии, применяются для создания каталогов и тезаурусов, которые помогают организовывать и классифицировать информацию в библиотеках и архивах.

7. Системы поддержки решений: Логические модели используются для разработки систем поддержки решений, которые помогают анализировать данные и принимать стратегические решения в организации.

8. Метаданные и управление знаниями: Логические модели помогают управлять метаданными о структуре данных и знаний в информационных системах, что способствует их более эффективному использованию.

Примеры успешных проектов, использующих логическую модель

Проект "DBpedia":

DBpedia - это проект, который создает структурированную базу данных на основе информации, извлеченной из Википедии. Он использует онтологическую модель для представления знаний о сущностях, событиях и фактах. DBpedia является одним из основных источников семантической информации в семантическом вебе.

Проект "Linked Open Data" (LOD):

LOD - это инициатива по созданию открытых и связанных данных в Интернете. Множество проектов, включая собственные онтологии и семантические модели, присоединяются к этой инициативе, чтобы сделать данные более доступными и интероперабельными.

Системы семантического поиска:

Множество поисковых систем и информационных порталов используют онтологии и логические модели для улучшения точности и релевантности поисковых результатов. Проекты, такие как Wolfram Alpha, семантический поиск Google и YAGO, применяют логические модели для более интеллектуального поиска.

Проекты в биоинформатике:

В биоинформатике логические модели используются для описания геномов, биологических путей и молекулярных взаимодействий. Проекты, такие как Gene Ontology (GO) и Biological Pathway Exchange (BioPAX), используют онтологии для структурирования биологических данных.

Проект "OpenCyc":

OpenCyc - это проект, который разрабатывает и поддерживает открытую версию системы Cyc, которая является масштабной онтологической базой данных. Она используется в исследованиях и разработках в области искусственного интеллекта, семантических технологий и многих других областях.

Значимость логической модели представления знаний в современных информационных системах

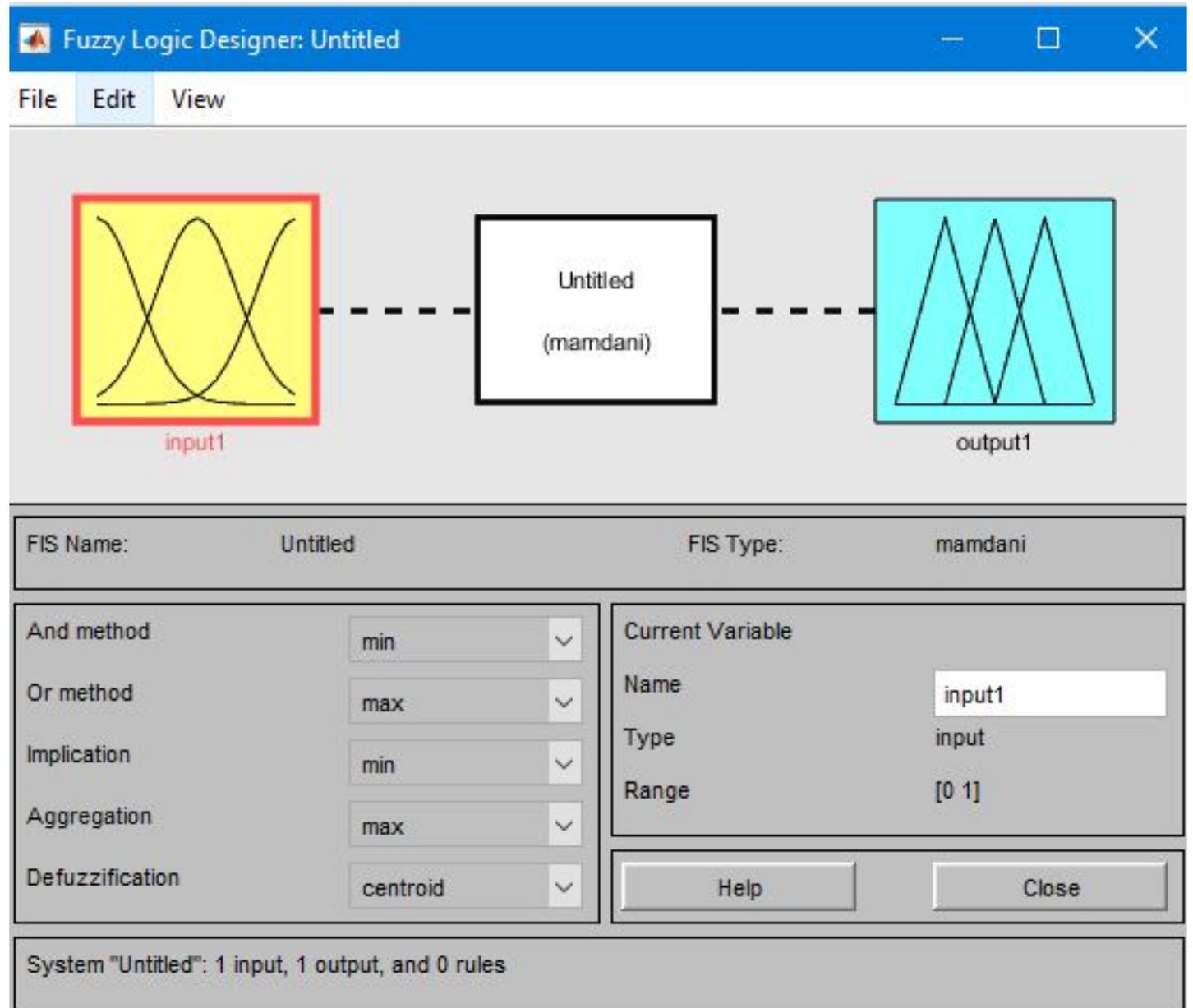
- 1. Семантическая точность и понимание данных:** Логическая модель позволяет формализовать знания и данные с точностью и семантической ясностью. Это делает возможным более точный и понятный обмен данными между различными системами и источниками.
- 2. Интеграция и сопоставление данных:** Логические модели позволяют интегрировать данные из разных источников и сопоставлять их с учетом семантики. Это решает проблему разнородных данных и облегчает создание единой семантической модели для использования в различных приложениях.
- 3. Семантический поиск и анализ текста:** Логическая модель является основой для развития семантических поисковых систем и инструментов анализа текста. Она позволяет системам понимать смысл запросов и текстовых данных, что улучшает релевантность результатов поиска.
- 4. Разработка экспертных систем и ИИ:** Логические модели используются в разработке экспертных систем и искусственного интеллекта для представления знаний и правил вывода. Они позволяют системам принимать решения и делать выводы на основе формализованных знаний.

5. **Семантический веб:** Логические модели играют ключевую роль в концепции семантического веба, где данные имеют более богатую семантику и связаны друг с другом. Это делает возможным создание более интеллектуальных и интероперабельных веб-приложений и сервисов.
6. **Управление знаниями в организациях:** В организациях логические модели используются для создания систем управления знаниями, которые помогают организовывать и делиться знаниями внутри компании, улучшая процессы принятия решений и обучения персонала.
7. **Семантический анализ данных в бизнесе:** Логические модели могут использоваться для анализа данных в бизнесе и выявления взаимосвязей и закономерностей в больших объемах информации. Это способствует более точному прогнозированию и оптимизации бизнес-процессов.
8. **Безопасность и конформность данных:** Логические модели могут быть использованы для определения правил доступа к данным и обеспечения их безопасности. Они также помогают обеспечивать конформность данных с требованиями законодательства и нормативами.

Заключение

Логическая модель представления знаний - это мощный инструмент, который обеспечивает семантическую точность, интеграцию данных, поддержку искусственного интеллекта и множество других преимуществ в современных информационных системах. Она играет важную роль в области семантического веба, искусственного интеллекта, бизнес-аналитики и управления знаниями.

Пример Fuzzy logic в Matlab



Новые переменны е

Fuzzy Logic Designer: Untitled

File Edit View

service

Untitled
(mamdani)

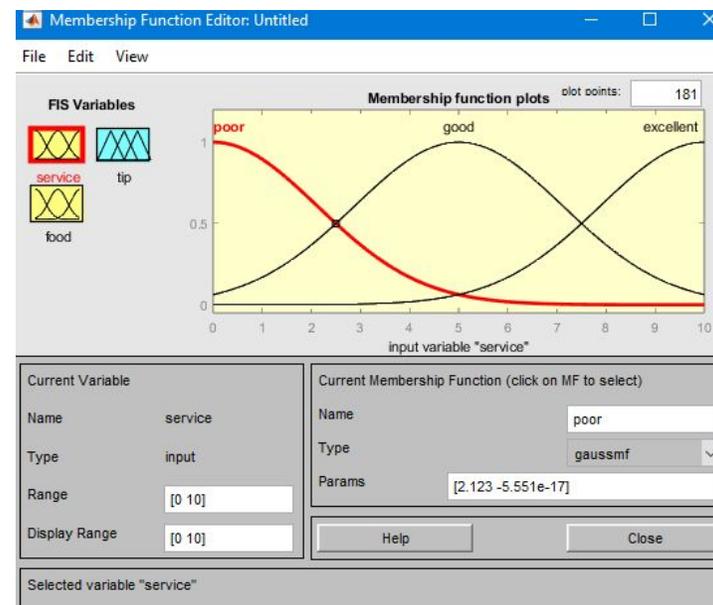
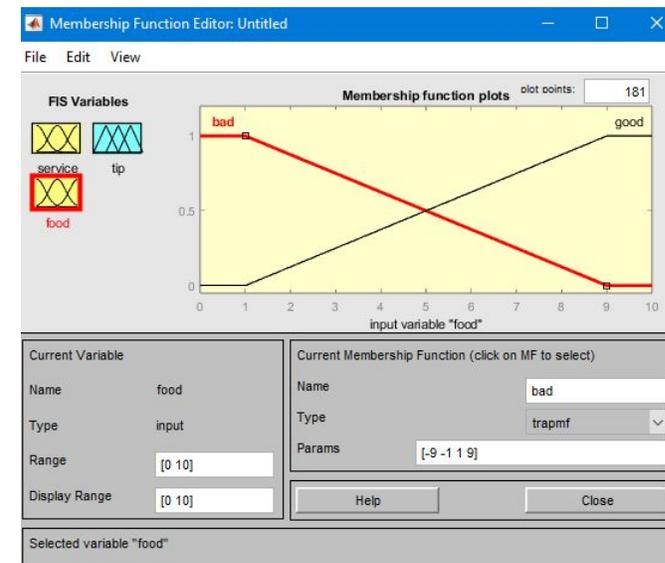
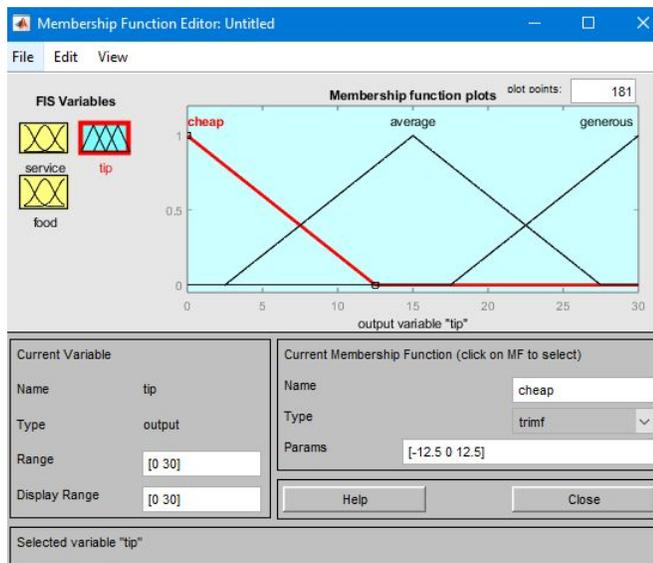
tip

food

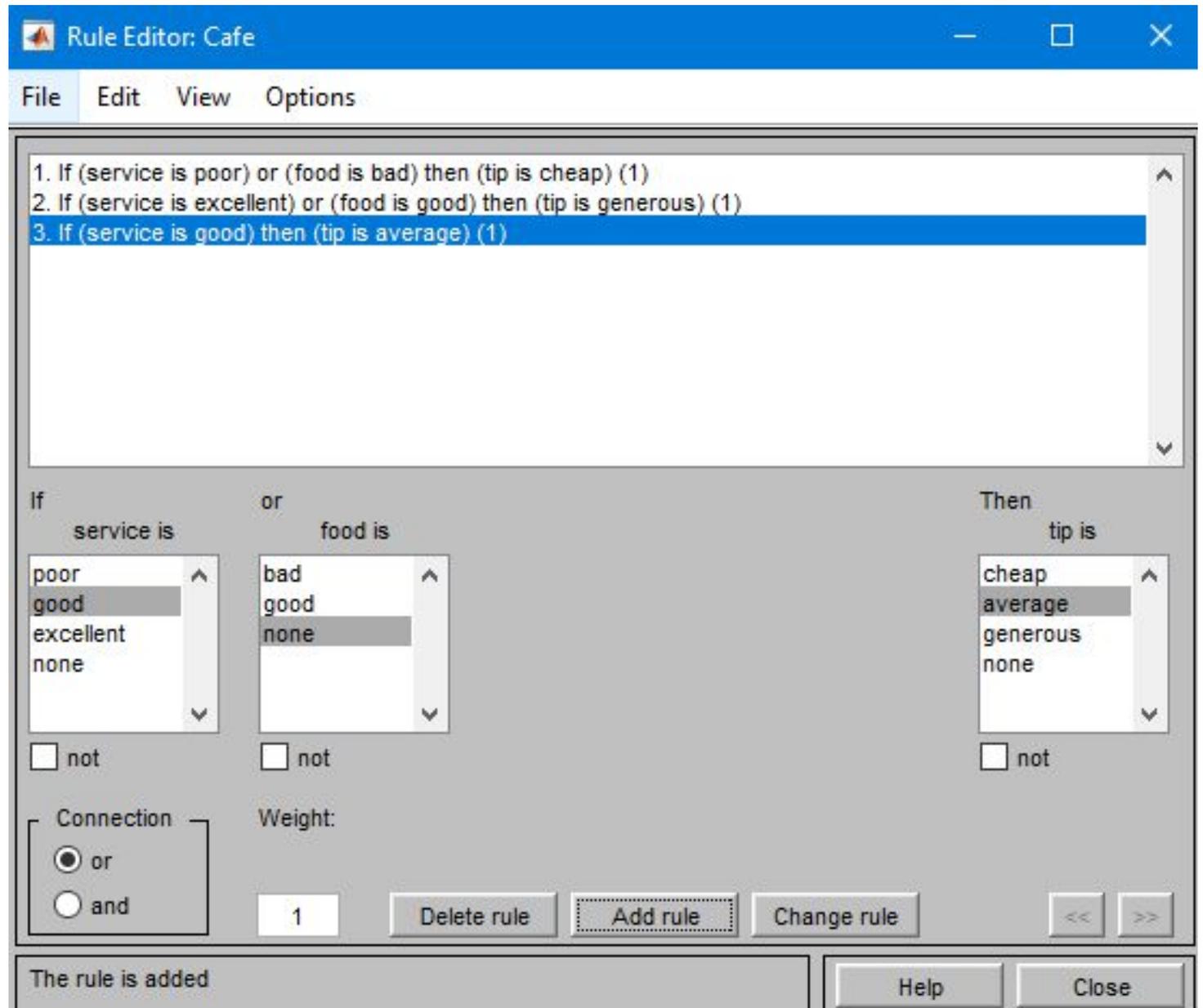
FIS Name:	Untitled	FIS Type:	mamdani
And method	min	Current Variable	
Or method	max	Name	tip
Implication	min	Type	output
Aggregation	max	Range	[0 1]
Defuzzification	centroid	Help Close	

Renaming output variable 1 to "tip"

Настройка функций принадлежности



Создание правил



Графики изменения входа и выхода

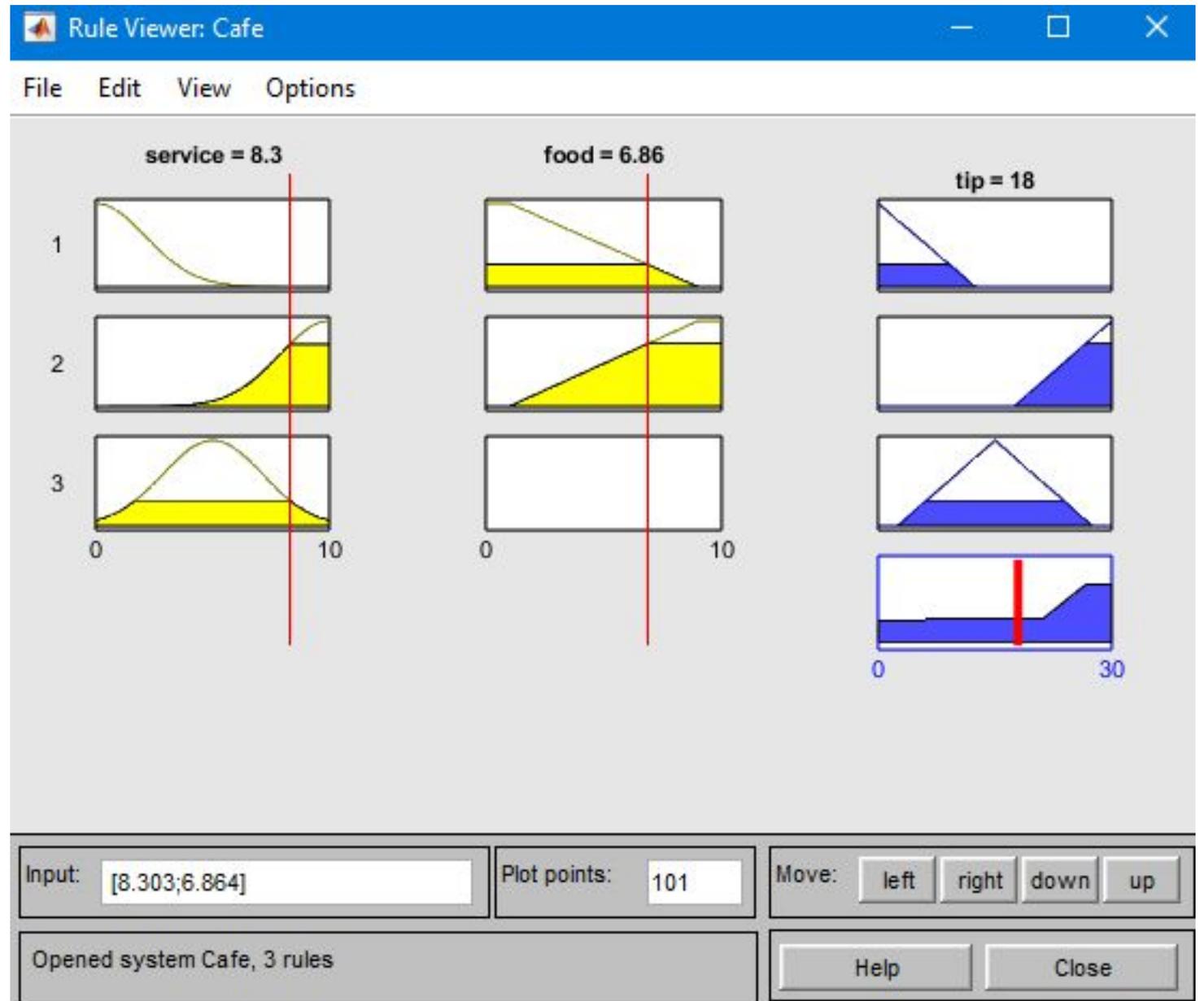


График поверхности и отклика

