Министерство Образования и Науки Российской Федерации

Федеральное Государственное Бюджетное Образовательное Учреждение Высшего Профессионального Образования «Санкт-Петербургский Национальный Исследовательский Университет Информационных Технологий, Механики и Оптики»

Факультет Программной Инженерии и Компьютерной Техники

Отчёт по модулю 1

по дисциплине

“Системы искусственного интеллекта”

Выполнил:

Студент группы P3330,

Тарасов Иван Сергеевич

Проверила:

Авдюшина Анна Евгеньевна

 Санкт-Петербург

2024

*Введение*

**Цель работы:** разработать базу знаний и онтологию по градостроительному симулятору Tropico 6. Протестировать их запросами.

*Анализ требований*

**Определение основных требований к системе поддержки принятия решений**

1. Система должна выдавать подробную информацию о зданиях (стоимость, качество работы/жилья/обслуживания, минимальный достаток, образование работников, количество работников/жильцов), а также информацию о потребляемых ресурсах в случае с заводами и производимых товаров в случае с плантациями, фермами, шахтами и заводами.
2. Система должна сохранять в базу знаний или удалять из нее здания в реальном времени, то есть строить или разрушать здания. При строительстве нового здания нужно указать количество работников, не превышающее максимальное.
3. Система должна рассчитывать прибыль от экспорта товаров, сколько товаров производится в день, зарплата работников, подходящее здание для работников. Все эти расчеты должны быть реализованы в виде правил
4. Система должна иметь правило, которое выводит информацию о том, сколько зданий, производящий некоторой товар, должно быть построено и сколько оптимально работников должно работать в этих зданиях, чтобы прибыль в день была равна целевому значению. Также необходимо вывести информацию, о том сколько необходимо построить подходящих жилых зданий для работников. В конце вывести общее потребление энергии и бюджет всего проекта.
5. В системе должны быть реализованы законы, изменяющие параметры зданий, например индустриализация и аграрные субсидии, понижающие стоимость зданий в 2 раза.

**Выявление требований к базе знаний и онтологии для представления знаний.**

1. База знаний должна быть организована таким образом, чтобы информация была структурированной, связной и легко извлекаемой для анализа и принятия решений. Она может включать факты, правила, связи и контексты.
2. Использование логического вывода для формирования новых знаний и получения ответов на сложные вопросы.
3. Онтология должна охватывать все ключевые понятия предметной области и их взаимосвязи, фактически представить базу знаний из Prolog в виде онтологии.
4. Онтология должна поддерживать различные типы отношений между понятиями, включая иерархии, ассоциации и другие типы связей.
5. Онтология должна быть гибкой, чтобы ее можно было расширять и обновлять по мере появления новых знаний или изменений в предметной области.

*Изучение основных концепций и инструментов*

**Обзор основных концепций баз знаний и онтологий.**

1) Основными концепциями базы знаний являются

- субъект: сущность, к которой относят предикат и выполняющая какое-либо действие.

- объект: сущность, являющееся целью или действием предиката.

- предикат: функция или отношения между субъектом и объектом.

- факты: элементарные знания, представляющие утверждения о предметной области.

- правила: логические конструкции, которые задают алгоритм отношений между фактами.

2) Основными концепциями онтологии являются:

- Классы и подклассы: Определение понятий и их иерархии

- Отношения: Связи между классами, описывающие их взаимодействие

- Экземпляры: Конкретные объекты, принадлежащие к определенным классам.

- Семантические правила: Формальные правила, определяющие смысл понятий и их взаимодействие (функциональность, транзитивность и тд)

**Изучение Prolog и его возможностей для разработки систем искусственного интеллекта.**

Человеческий интеллект построен на логических рассуждениях: что-то является чем-то, что-то выполняет что-то, из одного следует другое и так далее. Язык Prolog имеет схожий принцип. Все знания и логические отношения хранятся в базе знаний. Используя эти знания, интерпретатор Prolog просматривает базу и логически находит верные результаты. Простые знания, которые представляют собой один субъект, предикат и n-ое количество объектов, называются фактами. Эти факты могут быть объединены в правила для более сложного управления логическими фактами. В разработке систем искусственного на Prolog имеется существенный минус. Дело в том, что интерпретатор перебирает все возможные варианты практически без каких-либо алгоритмов машинного обучения, что делает поиск решения в большой базе знаний неприемлемо долгим.

*Реализация системы искусственного интеллекта (системы поддержки принятия решений)*

**Создание правил и логики вывода для принятия решений на основе базы знаний и онтологии.**

В Prolog правила создаются с использованием знака :- после которого идёт последовательность логических фактов или других правил. Они разделяются запятыми, которые означают логическое «И», точкой с запятой – логическое «ИЛИ» и знак «\+» - логическое «НЕ».

Пример создания правила в моей работе, которое проводит расчет зарплаты работников в месяц

*salary(Build, Workers, Salary) :-*

*(build\_price(Build, Price); electricity(Build, Price)), job(Build, \_, \_, N, \_),*

*Salary is (Price / 100) \* (Workers / N).*

В онтологии, а конкретно в Protégé, правила можно задавать с помощью SWRL, например следующее правило определяет, что если гражданин имеет работу, а на этой работе работники бедные, то это будет значить, что достаток этого гражданина – бедный.

*Гражданин(?p) ^ работать\_на(?p, ?f) ^ Рабочие(?f) ^ достаток\_рабочих(?f, ?w) -> иметь\_достаток(?p, ?w)*

Prolog использует механизм обратного логического вывода, когда система пытается удовлетворить цели, начиная с конечного утверждения и подбирая правила, которые приводят к этому утверждению. Это позволяет системе постепенно формировать цепочки рассуждений на основе имеющихся фактов и правил.

**Тестирование и отладка системы, обеспечение ее функциональности и эффективности.**

Отладить систему можно, используя встроенную «функцию» trace. Далее после ввода запроса в базу знаний нам выведется последовательность выполнения целей. Интерпретатор Prolog имеет 4 метки:

Call – вызов цели (предиката),

Exit – выход из цели, если система успешно нашла решение,

Redo – возврат к предыдущим решениям с целью поиска других решений,

Fail – система не смогла найти решение, переход к следующем варианту.

Пример отладки запроса «Вывести здания, которые имеют стоимость и являются промышленными зданиями»: *build\_price(Build, \_), industry(Build, \_).*

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Несмотря на то, что интерпретатор Prolog делает полный обход дерева решений, его можно оптимизировать и повысить его эффективность за счёт использования отсечений. Обозначаются восклицательным знаком «!». При нахождении первого решения все остальные решения данного предиката будут пропущены и интерпретатор не будет пробегаться по этим решениям.

*Оценка и интерпретация результатов*

**Примеры запросов для БЗ и онтологии, сравнение разницы реализации.**

Пример запроса: найти все промышленные здания

Запрос для БЗ (Prolog): *industry(Build, \_).*

Запрос для онтологии (SPARQL):

*SELECT ?instance WHERE {*

*?instance a <http://www.semanticweb.org/iwant/ontologies/2024/8/lab1#Промышленные>*

*}*

Основное отличие обоих запросов заключается в том, что запросы в Prolog больше походят на вызов функций, как в любом алгоритмическом языке, хотя Prolog таковым не является; а в онтологии запросы выглядит как запросы к базе данных, как SQL.

**Оценка соответствия системы поставленным требованиям и достижению целей проекта.**

Система полностью соответствует целям и требованиям проекта. Вывод информации о зданиях, ресурсах, расчётах работает корректно. Также корректно работают динамические знания (dynamic).

**Интерпретация результатов и описание дальнейших возможностей развития и улучшения системы.**

На данном этапе система поддержки принятия решений (СППР), основанная на базе знаний и онтологиях, демонстрирует следующие ключевые результаты:

1. На основе заранее определенных правил и логики вывода, система способна автоматически генерировать решения для различных сценариев.
2. Результаты работы системы подтверждают, что предикаты и правила, созданные в базе знаний, обеспечивают логически обоснованные выводы
3. Система может корректно обрабатывать различные запросы пользователя.
4. Использование онтологий позволяет легко добавлять новые концепции и отношения в систему.

Дальнейшие возможности развития и улучшения системы: расширения базы знаний и онтологии, интеграция с внешними источниками данных (веб-ресурсами, внешними БД), использование машинного обучения, оптимизация логики вывода.

*Заключение*

Разработанная система может выводить информацию о состоянии города. Пользуясь ею, можно построить оптимальный план по развитию города, получению целевой прибыли от экспорта товаров и улучшению жизни граждан.