

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение
высшего образования «Санкт-Петербургский политехнический
университет Петра Великого»

Институт компьютерных наук и кибербезопасности

Высшая школа технологий искусственного интеллекта

Направление: 02.03.01 Математика и компьютерные науки

Отчёт о выполнении курсовой работы по дисциплине
«Теория графов»

Реализация экспертной системы «Правила проезда
перекрёстков»

Студент,

группа 5130201/30001

_____ Васюк М. В.

Преподаватель

_____ Востров А. В.

« _____ » _____ 2025 г.

Санкт-Петербург, 2025

Содержание

Введение	3
1 Постановка задачи	4
2 Предметная область	5
3 Математическое описание	7
3.1 Экспертная система	7
3.2 Структура экспертной системы	7
3.3 Продукционная модель представления знаний	8
3.4 Бинарное дерево решений	9
3.4.1 Дерево решений для выбранной экспертной системы	9
4 Особенности реализации	11
4.1 Функция member	11
4.2 Функция ask-question	11
4.3 Функция YesOrNo	12
4.4 Шаблон status	12
4.5 Продукционные правила	12
4.5.1 Запуск программы	12
4.5.2 Узел дерева	13
4.5.3 Финальное правило	13
5 Результаты работы	14
Список использованной литературы	17

Введение

В рамках данной курсовой работы была разработана экспертная система, предназначенная для помощи в определении очередности проезда перекрёстков на основе вводимых пользователем условий. Система реализована с использованием продукционной модели представления знаний и среды CLIPS, широко применяемой для разработки интеллектуальных систем.

Разработка охватывает различные сценарии: регулируемые и нерегулируемые перекрёстки, наличие или отсутствие трамвая, особенности разметки и положения автомобиля. Знания о ПДД были представлены в виде бинарного дерева.

1 Постановка задачи

Целью курсовой работы является создание экспертной системы, которая по заданным условиям перекрёстка выдает рекомендацию о порядке проезда в соответствии с правилами дорожного движения.

Для достижения поставленной цели были посланы следующие задачи:

1. Выбрать предметную область и согласовать её с преподавателем.
2. Построить логическую структуру (бинарное дерево) с глубиной не менее 4 уровней и не менее 30 узлов, описывающую последовательность принятия решений.
3. На основе дерева составить правила и реализовать в среде CLIPS.
4. Провести тестирование системы на различных сценариях дорожных ситуаций.

2 Предметная область

В качестве предметной области были выбраны правила дорожного движения России.

Водитель, подъезжая к перекрёстку, должен правильно оценить ситуацию, чтобы определить, уступает он дорогу или имеет право проезда первым. При этом необходимо учитывать ряд критериев, таких как: тип перекрёстка (регулируемый или нерегулируемый), наличие регулировщика и его сигналы, направление движения, приоритет дорог (главная/второстепенная), наличие трамвайного движения, траектория других участников, а также дорожные знаки.

Выбранные для экспертной системы факты:

1. ПДД 13.1. При повороте направо или налево водитель обязан уступить дорогу пешеходам, лицам, использующим для передвижения средства индивидуальной мобильности, и велосипедистам, пересекающим проезжую часть дороги, на которую он поворачивает.
2. ПДД 13.2. Запрещается выезжать на перекресток, пересечение проезжих частей или участка перекрестка, обозначенного разметкой 1.26, если впереди по пути следования образовался затор, который вынудит водителя остановиться, создав препятствие для движения транспортных средств в поперечном направлении, за исключением поворота направо или налево в случаях, установленных настоящими Правилами.
3. ПДД 13.3. Перекресток, где очередность движения определяется сигналами светофора или регулировщика, считается регулируемым. При желтом мигающем сигнале, неработающих светофорах или отсутствии регулировщика перекресток считается нерегулируемым, и водители обязаны руководствоваться правилами проезда нерегулируемых перекрестков и установленными на перекрестке знаками приоритета.
4. ПДД 13.4. При повороте налево или развороте по зеленому сигналу светофора водитель безрельсового транспортного средства обязан уступить дорогу транспортным средствам, движущимся со встречного направления прямо или направо. Таким же правилом должны руководствоваться между собой водители трамваев.
5. ПДД 13.5. При движении в направлении стрелки, включенной в дополнительной секции одновременно с желтым или красным сигналом светофора, водитель обязан уступить дорогу транспортным средствам, движущимся с других направлений.
6. ПДД 13.6. Если сигналы светофора или регулировщика разрешают движение одновременно трамваю и безрельсовым транспортным средствам, то трамвай имеет преимущество независимо от направления его движения. Однако при движении в направлении стрелки, включенной в дополнительной секции одновременно с красным или желтым сигналом светофора, трамвай должен уступить дорогу транспортным средствам, движущимся с других направлений.

7. ПДД 13.7. Водитель, въехавший на перекресток при разрешающем сигнале светофора, должен выехать в намеченном направлении независимо от сигналов светофора на выходе с перекрестка. Однако, если на перекрестке перед светофорами, расположенными на пути следования водителя, имеются стоп-линии (знаки 6.16) водитель обязан руководствоваться сигналами каждого светофора.
8. ПДД 13.8. При включении разрешающего сигнала светофора водитель обязан уступить дорогу транспортным средствам, завершающим движение через перекресток, и пешеходам, не закончившим переход проезжей части данного направления.
9. ПДД 13.9. На перекрестке неравнозначных дорог водитель транспортного средства, движущегося по второстепенной дороге, должен уступить дорогу транспортным средствам, приближающимся по главной, независимо от направления их дальнейшего движения. На таких перекрестках трамвай имеет преимущество перед безрельсовыми транспортными средствами, движущимися в попутном или встречном направлении по равнозначной дороге, независимо от направления его движения.
10. ПДД 13.10. В случае, когда главная дорога на перекрестке меняет направление, водители, движущиеся по главной дороге, должны руководствоваться между собой правилами проезда перекрестков равнозначных дорог. Этими же правилами должны руководствоваться водители, движущиеся по второстепенным дорогам.
11. ПДД 13.11. На перекрестке равнозначных дорог, за исключением случая, предусмотренного пунктом 13.11(1) Правил, водитель безрельсового транспортного средства обязан уступить дорогу транспортным средствам, приближающимся справа. Этим же правилом должны руководствоваться между собой водители трамваев. На таких перекрестках трамвай имеет преимущество перед безрельсовыми транспортными средствами независимо от направления его движения.
12. ПДД 13.11(1) При въезде по дороге, не являющейся главной, на перекресток, на котором организовано круговое движение и который обозначен знаком 4.3, водитель транспортного средства обязан уступить дорогу транспортным средствам, движущимся по такому перекрестку. (Изменения вступили в силу: 1 марта 2023 года).
13. ПДД 13.12. При повороте налево или развороте водитель безрельсового транспортного средства обязан уступить дорогу транспортным средствам, движущимся по равнозначной дороге со встречного направления прямо или направо. Этим же правилом должны руководствоваться между собой водители трамваев.
14. ПДД 13.13. Если водитель не может определить наличие покрытия на дороге (темное время суток, грязь, снег и тому подобное), а знаков приоритета нет, он должен считать, что находится на второстепенной дороге.

3 Математическое описание

3.1 Экспертная система

Экспертная система – компьютерная система, способная частично заменить специалиста-эксперта в разрешении проблемной ситуации.

Экспертная система работает в двух основных режимах:

- 1) в режиме приобретения знаний;
- 2) в режиме решения задачи (называемом также режимом консультаций, или режимом использования экспертной системы).

В режиме приобретения знаний работу с экспертной системой осуществляет эксперт при посредничестве инженера по знаниям. В этом режиме эксперт, используя компонент приобретения знаний, наполняет систему знаниями (данными), которые, в свою очередь, позволяют системе в режиме решения уже без участия эксперта решать задачи из данной предметной области.

В режиме решения задачи общение с экспертными системами осуществляет непосредственно конечный пользователь, которого интересует конечной итог работы и иногда способ его получения.

В зависимости от назначения экспертной системы пользователь не обязательно должен быть специалистом в данной проблемной области. В этом случае он обращается к экспертным системам за результатом, не имея достаточных знаний для получения результатов.

3.2 Структура экспертной системы

Структура экспертной системы представлена на [Рис. 1]

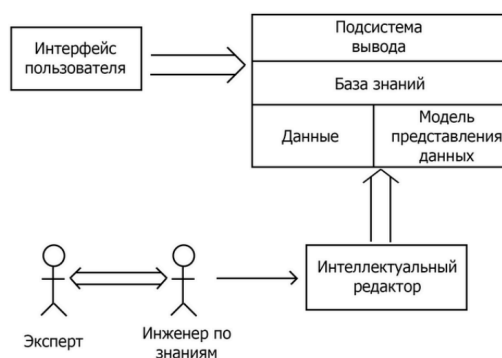


Рис. 1. Структура экспертной системы

Знания – это правила, законы, закономерности получены в результате профессиональной деятельности в пределах предметной области.

База знаний – база данных содержащая правила вывода и информацию о человеческом опыте и знаниях в некоторой предметной области.

Данные – это совокупность фактов и идей представленных в формализованном виде.

Механизм логического вывода данных (подсистема вывода)

Механизм логического вывода данных выполняет анализ и прodelывает работу по получению новых знаний исходя из сопоставления исходных данных из базы данных и правил из базы знаний.

Механизм логического вывода данных концептуально можно представить в виде $\langle A, B, C, D \rangle$: A — функция выбора из базы знаний и из базы данных закономерностей и фактов соответственно; B — функция проверки правил, результатом которой определяется множество фактов из базы данных к которым применимы правила; C — функция, которая определяет порядок применения правил, если в результате правила указаны одинаковые факты; D — функция, которая применяет действие.

3.3 Продукционная модель представления знаний

В основе продукционной модели представления знаний находится конструктивная часть, продукция(правило): IF \langle условие \rangle , THEN \langle действие 1 \rangle , ELSE \langle действие 2 \rangle .

Продукция состоит из двух частей: условие — антецедент, действие — консеквент.

Условия можно сочетать с помощью логических функций AND, OR.

Антецеденты и консеквенты составленных правил формируются из атрибутов и значений.

В базе данных продукционной системы хранятся правила, истинность которых установлена к заранее при решении определенной задачи. Правило срабатывает, если при сопоставлении фактов, содержащихся в базе данных с антецедентом правила, которое подвергается проверке, имеет место совпадение. Результат работы правила заносится в базу данных.

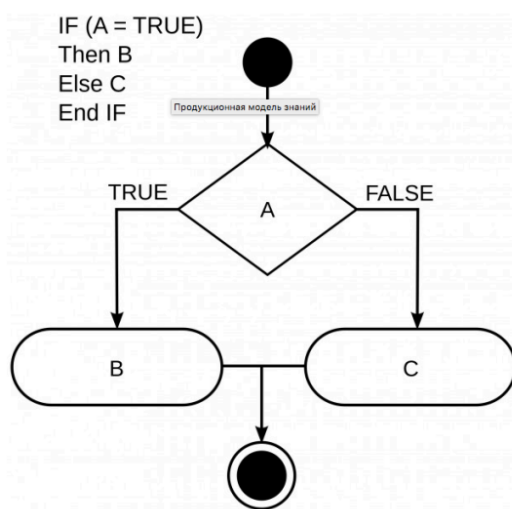


Рис. 2. Структура правил

3.4 Бинарное дерево решений

Экспертная система представлена в виде бинарного дерева решений. Граф без циклов называется ациклическим или лесом. Связный ациклический граф называется (свободным) деревом.

Дерево решений — метод представления решающих правил в определенной иерархии, включающей в себя элементы двух типов — узлов и листьев. Узлы включают в себя решающие правила и производят проверку примеров на соответствие выбранного атрибута обучающего множества.

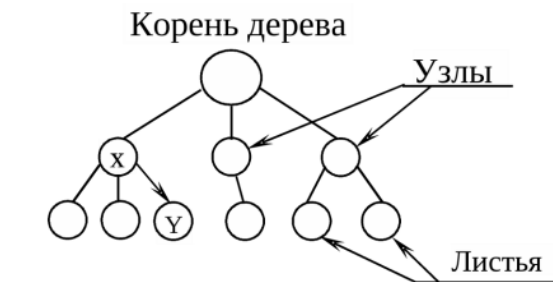


Рис. 3. Схема дерева

Дерево решений отображает логическую последовательность принятия решений в экспертной системе. Оно является графическим представлением компонентов и связей в экспертной системе, включая факты, правила и процедуры вывода.

3.4.1 Дерево решений для выбранной экспертной системы

На Рис. 4. изображено бинарное дерево решений по правилам проезда перекрестка. Оно состоит из 26 узлов, 27 листьев и 8 ярусов.

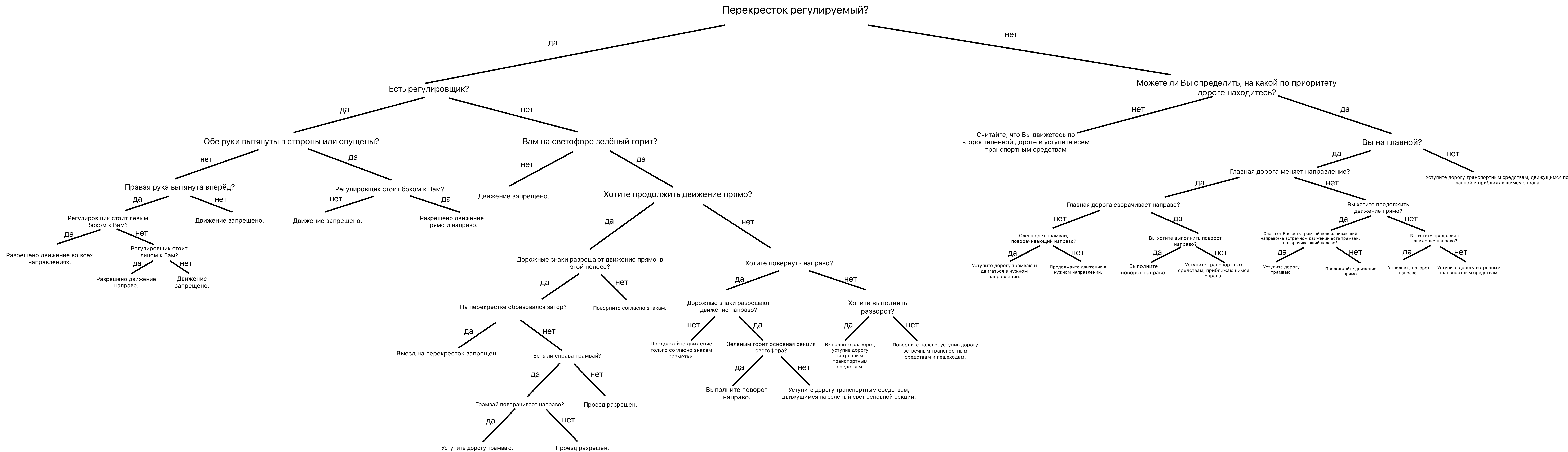


Рис. 4. Бинарное дерево решений

4 Особенности реализации

Реализация экспертной системы выполнена в среде CLIPS («C Language Integrated Production System») — специализированной платформе для разработки продукционных экспертных систем. В качестве модели представления знаний использована продукционная модель, реализованная в виде набора правил (defrule) и пользовательских функций (deffunction).

4.1 Функция member

Проверяет, содержится ли элемент item в списке list. Используется для проверки допустимости пользовательского ответа (например, "да" "нет" "д" "н"). Если элемент является текстовым (lexemep), он предварительно переводится в нижний регистр. Затем вызывается встроенная функция member, которая возвращает истину, если элемент найден. Ознакомиться с кодом можно в [«Листинг 1»](#).

Листинг 1. Функция member

```
1 (deffunction member (?item $?list)
2 (if (lexemep ?item)
3 then (bind ?item (lowercase ?item)))
4 (member$ ?item ?list))
```

4.2 Функция ask-question

Выводит вопрос пользователю и считывает ввод, проверяя его корректность по списку допустимых значений (allowed-values). Все ответы приводятся к нижнему регистру для устойчивости к регистру ввода. В случае недопустимого ответа — пользователю повторно задаётся тот же вопрос. Возвращает валидный ответ пользователя. Ознакомиться с кодом можно в [«Листинг 2»](#).

Листинг 2. Функция ask-question

```
1 (deffunction ask-question (?question $?allowed-values)
2 (printout t ?question " ")
3 (bind ?answer (read))
4 (if (lexemep ?answer)
5 then (bind ?answer (lowercase ?answer)))
6 (while (not (member ?answer ?allowed-values)) do
7 (printout t "Ответ должен быть: " $?allowed-values crlf)
8 (printout t ?question " ")
9 (bind ?answer (read))
10 (if (lexemep ?answer)
11 then (bind ?answer (lowercase ?answer))))
12 ?answer)
```

4.3 Функция YesOrNo

Упрощает работу с бинарными вопросами ("да"/"нет") и возвращает булево значение (TRUE или FALSE). Варианты ответов (да, д, нет, н) поддерживают как полную, так и сокращённую форму. Используется внутри продукционных правил для принятия решений. Ознакомиться с кодом можно в «[Листинг 3](#)».

Листинг 3. Функция YesOrNo

```
1 (deffunction YesOrNo (?question)
2   (bind ?response (ask-question ?question да нет д н))
3   (if (or (eq ?response да) (eq ?response д))
4       then TRUE
5       else FALSE))
```

4.4 Шаблон status

Хранит текущее состояние рассуждений системы — факты о ситуации, полученные на основе ответов пользователя. Все ключевые промежуточные и финальные условия записываются в виде фактов status.

```
1 (deftemplate status (slot name))
```

4.5 Продукционные правила

Правила реализуют переходы между вершинами дерева решений. Каждое правило: проверяет текущий факт status, задаёт пользователю вопрос через YesOrNo, в зависимости от ответа добавляет новый факт status.

4.5.1 Запуск программы

Это правило инициализирует работу экспертной системы. Оно срабатывает только в том случае, если в рабочей памяти ещё не существует факта (status (name начато)). По сути создаётся первая точка входа в бинарное дерево принятия решений.

Листинг 4. Функция запуска

```
1 (defrule start
2   (not (status (name начато)))
3   =>
4   (assert (status (name начато))))
5 }
```

4.5.2 Узел дерева

```
1 (defrule regulated-crossing
2 (status (name начато))
3 =>
4 (if (YesOrNo "Перекрёсток регулируемый? (да/нет):")
5 then (assert (status (name регулируемый)))
6 else (assert (status (name нерегулируемый)))))
```

Данное правило реализует первый логический узел дерева решений. Оно срабатывает после того, как был установлен факт (status (name начато)), то есть после старта системы. При "да" → добавляется факт (status (name регулируемый)). При "нет" → добавляется факт (status (name нерегулируемый)).

Аналогично реализованы остальные узлы.

4.5.3 Финальное правило

Данное правило завершает одну из логических ветвей дерева решений. Оно формулирует конечный вывод — рекомендацию водителю, которая соответствует правилам дорожного движения и введенным условиям.

```
1 (defrule result27
2 (status (name главная_не_направо))
3 =>
4 (printout t crlf "Уступите дорогу встречным транспортным средст
   вам." crlf))
```

5 Результаты работы

Чтобы запустить программу, нужно загрузить файл, с помощью (load). При выполнении команды reset текущий список фактов CLIPS очищается, а затем в него добавляются все факты, заданные конструкторами deffacts. И, наконец, функция run запускает механизм логического вывода и приводит нашу программу в движение.

```
CLIPS> (load "...")
TRUE
CLIPS> (reset)
CLIPS> (run)
Перекрёсток регулируемый? (да/нет):
```

Рис. 4. Запуск программы

После запуска программы выводится первый вопрос, находящийся в корне бинарного дерева. Пользователь с помощью ответов да/нет или сокращенного варианта д/н отвечает на все последующие вопросы и приходит к одному из листов. Ниже приведён пример работы программы:

```
Перекрёсток регулируемый? (да/нет): да
Есть регулировщик? (да/нет): нет
Вам на светофоре горит зелёный сигнал? (да/нет): да
Вы хотите продолжить движение прямо? (да/нет): да
Дорожные знаки разрешают движение прямо в этой полосе? (да/нет): да
На перекрёстке затор? (да/нет): нет
Есть ли справа трамвай? (да/нет): нет
Проезд разрешен.
```

Рис. 5. Пример работы программы

Пример с сокращенными ответами:

Перекрёсток регулируемый? (да/нет): н

Можете ли Вы определить, на какой по приоритету дороге находитесь?: д

Вы на главной? (да/нет): д

Главная дорога меняет направление? (да/нет): н

Вы хотите продолжить движение прямо? (да/нет): д

Есть ли трамвай попутного направления, поворачивающий направо/встречного направления, поворачивающий налево? (да/нет): н

Продолжайте движение прямо.

Рис. 6. Пример работы программы

В случае некорректного ввода, программа предлагает ввести ответ ещё раз:

Вы на главной? (да/нет): аа

Ответ должен быть: (да нет д н)

Вы на главной? (да/нет):

Рис. 7. Некорректный ввод

Заключение

В ходе выполнения курсовой работы была разработана экспертная система с продукционной моделью представления знаний в области правил проезда перекрёстков, основанная на анализе дорожной ситуации и требований ПДД. Система позволяет на основе последовательных вопросов определить порядок действий водителя при проезде как регулируемых, так и нерегулируемых перекрёстков.

Было построено бинарное дерево решений, содержащее 26 узлов и 27 листьев. Структура дерева включает более 4 ярусов.

Достоинства:

- вопросы охватывают основные дорожные ситуации, предусмотренные ПДД;
- система выдает чёткие текстовые рекомендации, понятные пользователю;
- высокая скорость работы и выводов при минимальных ресурсах.

Недостатки:

- дерево решений частично несбалансировано;
- добавление новых ситуаций требует пересмотра структуры дерева или написания новых правил;
- система не учитывает все нестандартные случаи (например, круговое движение, погодные условия и т.д.).

Масштабирование: Система может быть расширена за счёт добавления новых правил, охватывающих более сложные ситуации (круговое движение, количество полос, скорость).

Курсовая работа выполнена в среде разработки CLIPS IDE версии 6.4.2.

Список использованной литературы

- [1] Секция «Телематика». — Текст : электронный // tema.spbstu.ru : [сайт]. — URL: <https://tema.spbstu.ru/tgraph/> (дата обращения: 07.05.2024).
- [2] CLIPS Rule Based Programming Language - Browse /CLIPS/6.4.1 at SourceForge.net // sourceforge.net URL: <https://sourceforge.net/projects/clipsrules/files/CLIPS/6.4.1/> (дата обращения: 20.04.2024).
- [3] Частиков А. П., Гаврилова Т. А., Белов Д. Л. Разработка экспертных систем. Среда CLIPS. — Санкт-Петербург: БХВ-Петербург, 2003. — 393 с.