|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

**Лабораторная работа №17**

**по курсу “Функциональное и логическое программирование”**

1. **по теме** “**Формирование эффективных программ на Prolog**”

|  |  |
| --- | --- |
| Студент: | Уласик Е.А. |
| Группа: | ИУ7-61 |
| Преподаватель: | Толпинская Н.Б. |

*2020 г.*

**Цель работы –** изучить способы организации эффективных программ на Prolog, особенности использования системных предикатов и порядок выполнения программ с их использованием.

**Задачи работы**: приобрести навыки эффективного описания предметной области с использованием фактов и правил. Изучить возможность использования системных предикатов в программе на Prolog, принципы и особенности порядка работы в этом случае. Способ формирования и изменения резольвенты в этом случае и порядок формирования ответа.

**Задание:** Ответить на вопросы:

1. Какое первое состояние резольвенты?
2. В каком случае система запускает алгоритм унификации? (т.е. Как эту необходимость на формальном уровне распознает система?)
3. Каково назначение использования алгоритма унификации?
4. Каков результат работы алгоритма унификации?
5. В каких пределах программы переменные уникальны?
6. Как применяется подстановка, полученная с помощью алгоритма унификации?
7. Как изменяется резольвента?
8. В каких случаях запускается механизм отката?

**В одной программе написать правила, позволяющие найти**

1. Максимум из двух чисел **а)** без использования отсечения,

**в)** с использованием отсечения;

1. Максимум из трех чисел **а)** без использования отсечения,

**в)** с использованием отсечения;

Убедиться в правильности результатов**.**

**Для каждого случая пункта 2 обосновать необходимость всех условий тела.**

**Для одного** из вариантов **ВОПРОСА** и каждого варианта **задания 2 составить таблицу**, отражающую конкретный порядок работы системы:

Т.к. резольвента хранится в виде стека, то состояние резольвенты требуется отображать в столбик: вершина – сверху! Новый шаг надо начинать с нового состояния резольвенты!

**Вопрос:…..**

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Состояние резольвенты, и вывод: дальнейшие действия (почему?) | Для каких термов запускается алгоритм унификации: Т1=Т2 и каков **результат** (и подстановка) | Дальнейшие действия: прямой ход или откат (почему и к чему приводит?) |
| 1… | … | … | Комментарий, вывод… |
| … | … | … | … |

Вопросы

1. **Какое первое состояние резольвенты?**

Первое состояние резольвенты – это вопрос.

1. **В каком случае система запускает алгоритм унификации? (Как эту необходимость на формальном уровне распознает система?)**

Алгоритм унификации запускается автоматически, если есть что доказывать. Система распознаёт это, если резольвента не пуста и есть неотмеченные знания в базе знаний.

1. **Каковы назначение и результат использования алгоритма унификации?**

Алгоритм унификации нужен для сопоставления двух термов. Это сопоставление может завершиться получением одного из двух результатов – успехом или тупиковой ситуацией, то есть неудачей.

1. **Каков результат работы алгоритма унификации?**

Алгоритм унификации в сопоставлении двух термов. Это сопоставление может завершиться получением одного из двух результатов – успехом или тупиковой ситуацией, то есть неудачей.

1. **В каких пределах программы переменные уникальны?**

Именованные переменные уникальны в рамках одного предложения, анонимные уникальны все.

1. **Как применяется подстановка, полученная с помощью алгоритма унификации?**

Подстановка применяется к подцели резольвенты, конкретизируя значения переменных.

1. **Как меняется резольвента?**

Резольвента изменяется с использованием редукции, то есть замены подцели телом правила, с которым унифицируется данная подцель, и механизма отката. В ходе отката резольвента возвращается в предыдущее состояние.

1. **В каких случаях запускается механизм отката?**

Механизм отката запускается в случае, если в ходе поиска доказательства подцели программа зашла в тупиковую ситуацию, и, если резольвента не пуста и решение найдено, но в базе знаний остались не отмеченные предложения.

Листинг программы

domains

num = integer.

predicates

**max2**(num, num, num).

**max2\_2**(num, num, num).

**max3**(num, num, num, num).

**max3\_2**(num, num, num, num).

clauses

**max2**(A, B, A) :- A >= B.

**max2**(A, B, B) :- B > A.

**max2\_2**(A, B, A) :- A >= B, !.

**max2\_2**(**\_**, B, B).

**max3**(A, B, C, A) :- A >= B, A >= C.

**max3**(A, B, C, B) :- B >= C, B >= A.

**max3**(A, B, C, C) :- C >= A, C >= B.

**max3\_2**(A, B, C, A) :- A >= B, A >= C, !.

**max3\_2**(**\_**, B, C, B) :- B >= C, !.

**max3\_2**(**\_**, **\_**, C, C).

goal

% max2(3, 2, RES).

% max2\_2(3, 6, RES).

% max3(3, -12, 26, RES).

**max3**(**1**, **2**, **2**, RES).

Обоснования

В правилах **max3** не используется отсечение. Это приводит к тому, что после нахождения максимального числа, начнётся поиск альтернативных решений, поэтому необходимо каждое число проверять на максимальность в каждом правиле.

В правилах **max3\_2** используется отсечение. Это позволяет отбросить поиск альтернативных решений. Если очередное число не является максимальным, то проверять его далее не имеет смысла. Последнее не проверенное число заносится в результат без проверок.

Результат работы

1. Вопрос: max2(2, 3, RES).

Ответ: RES=3

2. Вопрос: max2(3, 2, RES).

Ответ: RES=3

3. Вопрос: max2\_2(6, 3, RES).

Ответ: RES=6

4. Вопрос: max2\_2(3, 6, RES).

Ответ : RES=6

5. Вопрос: max3(26, -12, 3, RES).

Ответ: RES=26

6. Вопрос: max3(-12, 26, 3, RES).

Ответ: RES=26

7. Вопрос: max3(3, 26, -12, RES).

Ответ: RES=26

8. Вопрос: max3(3, -12, 26, RES).

Ответ: RES=26

9. Вопрос: max3\_2(4, 10, 24, RES).

Ответ: RES=24

10. Вопрос: max3\_2(4, 24, 10, RES).

Ответ: RES=24

11. Вопрос: max3\_2(10, 4, 24, RES).

Ответ: RES=24

12. Вопрос: max3\_2(24, 4, 10, RES).

Ответ: RES=24

Таблица

Вопрос: max3(-12, 26, 3, RES).

Сравнение в таблице осуществляется между подцелью из резольвенты и каждого знания из базы знаний.

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Состояние резольвенты, и вывод: дальнейшие действия (почему?) | Для каких термов запускается алгоритм унификации: Т1=Т2 и каков **результат** (и подстановка) | Дальнейшие действия: прямой ход или откат (почему и к чему приводит?) |
| 1 | Резольвента:  max3(-12, 26, 3, RES).  Дальнейшие действия:  унификация, так как резольвента не пуста. | Сравниваются max3(-12, 26, 3, RES) и max3(A, B, C, A). Термы унифицируемые. Подстановка:  {A = -12,  B = 26,  C = 3,  RES = -12}. | Прямой ход. Термы унифицируемы. Производится редукция. К резольвенте применяется подстановка. |
| 2 | Резольвента:  -12 >= 26,  -12 >= 3.  Дальнейшие действия:  унификация, так как резольвента не пуста | Сравниваются -12 и 26  на >=(-12, 26).  Результат: “нет” | Откат к предыдущему состоянию. |
| 3 | Резольвента:  max3(-12, 26, 3, RES).  Дальнейшие действия:  унификация, так как резольвента не пуста. | Сравниваются max3(-12, 26, 3, RES) и max3(A, B, C, B)). Термы унифицируемые. Подстановка:  {A = -12,  B = 26,  C = 3,  RES = 26}. | Прямой ход. Термы унифицируемы. Производится редукция. К резольвенте применяется подстановка. |
| 4 | Резольвента:  26 >= 3,  26 >= -12.  Дальнейшие действия:  унификация, так как резольвента не пуста | Сравниваются 26 и 3  на >=(26, 3).  Результат: “да” | Прямой ход. Переход к следующей подцели. |
| 5 | Резольвента:  26 >= -12.  Дальнейшие действия:  унификация, так как резольвента не пуста | Сравниваются 26 и -12  на >=(26, -12).  Результат: “да” | Прямой ход. Переход к следующей подцели. |
| 6 | Резольвента пуста. Все переменные связаны. |  | Вывод результата. Начинается поиск альтернативного решения, откат к предыдущему состоянию резольвенты. |
| 7 | Резольвента:  max3(-12, 26, 3, RES).  Дальнейшие действия:  унификация, так как резольвента не пуста | Сравниваются max3(-12, 26, 3, RES) и max3(A, B, C, C). Термы унифицируемые. Подстановка:  {A = -12,  B = 26,  C = 3,  RES = 3}. | Прямой ход. Термы унифицируемы. Производится редукция. К резольвенте применяется подстановка. |
| 8 | Резольвента:  3 >= -12,  3 >= 26.  Дальнейшие действия:  унификация, так как резольвента не пуста | Сравниваются 3 и -12  на >=(3, -12).  Результат: “нет” | Прямой ход. Переход к следующей подцели. |
| 9 | Резольвента:  3 >= 26.  Дальнейшие действия:  унификация, так как резольвента не пуста | Сравниваются 3 и 26  на >=(3, 26).  Результат: “нет” | Откат к предыдущему состоянию. |
| 10 | Резольвента:  max3(-12, 26, 3, RES).  Дальнейшие действия:  унификация, так как резольвента не пуста | Унификация подцели из резольвенты и ещё неотмеченных предложений из базы знаний. | Нет успешных унификаций. Завершение работы программы. |

Вопрос: max3\_2(10, 24, 4, RES)

|  |  |  |  |
| --- | --- | --- | --- |
| № шага | Состояние резольвенты, и вывод: дальнейшие действия (почему?) | Для каких термов запускается алгоритм унификации: Т1=Т2 и каков **результат** (и подстановка) | Дальнейшие действия: прямой ход или откат (почему и к чему приводит?) |
| 1 | Резольвента: max3\_2(10, 24, 4, RES)  Дальнейшие действия:  унификация, так как резольвента не пуста. | Сравниваются max3\_2(10, 24, 4, RES) и max3\_2(A, B, C, A) Термы унифицируемые. Подстановка:  {A = 10,  B = 24,  C = 4,  RES = 10}. | Прямой ход. Термы унифицируемы. Производится редукция. К резольвенте применяется подстановка. |
| 2 | Резольвента:  10 >= 24,  10 >= 4.  Дальнейшие действия:  унификация, так как резольвента не пуста | Сравниваются 10 и 24  на >=(10, 24).  Результат: “нет” | Откат к предыдущему состоянию. |
| 3 | Резольвента: max3\_2(10, 24, 4, RES)  Дальнейшие действия:  унификация, так как резольвента не пуста. | Сравниваются max3\_2(10, 24, 4, RES) и max3\_2(\_, B, C, B) Термы унифицируемые. Подстановка:  {\_ = 10,  B = 24,  C = 4,  RES = 24}. | Прямой ход. Термы унифицируемы. Производится редукция. К резольвенте применяется подстановка. |
| 4 | Резольвента:  24 >= 4,  !.  Дальнейшие действия:  унификация, так как резольвента не пуста | Сравниваются 4 и 24  на >=(24, 4).  Результат: “да” | Прямой ход. Переход к следующей подцели. |
| 5 | Резольвента:  !  Дальнейшие действия:  отсечение |  | Вывод результата. Произошло отсечение. Завершение выполнения программы. |

Вывод

Были изучены способы организации эффективных программ на Prolog, особенности использования системных предикатов и порядок выполнения программ с их использованием.

Используя отсечения, можно сократить количество сравнений и, таким образом, сократить время выполнения программы.

Ответы на замечания от 30.05

№15.

Замечание:

1. **Какова семантика (смысл) предложений раздела DOMAINS? Когда, где и с какой целью используется это описание?**

Предложения в разделе DOMAINS используются для определения доменов. Раздел доменов предназначен для описания структур. Структур чего???

Ответ: вариантных доменов.

Замечание:

На первом шаге работы происходит попытка унификации вопроса и первого предложения базы знаний Вопрос и предложение могут иметь разную структуру!!!

Ответ: На первом шаге работы происходит попытка унификации вопроса и первого терма, у которого функтор и арность совпадают с функтором и арностью вопроса.

Исправленный листинг программы:

(след. страница)

Часть 1. Продолжение на следующей странице

domains

surname, phone, city, street, brand, colour, bank, account, model, name = symbol.

house, flat, price = integer.

address = **adr**(city, street, house, flat).

sum = real.

property = **auto**(name, price);

**building**(name, price);

**land**(name, price);

**seaTransport**(name, price).

predicates

**phonebook**(surname, phone, address).

**contributor**(surname, bank, account, sum).

**owner**(surname, property).

**ownership**(surname, name, price).

**getInfoBySurnameAndCity**(surname, city, street, bank, phone).

**knowledgeProperty**(property, name, price).

clauses

**knowledgeProperty**(**auto**(Name, Price), PropertyName, PropertyPrice) :- PropertyName=Name, PropertyPrice=Price.

**knowledgeProperty**(**building**(Name, Price), PropertyName, PropertyPrice) :- PropertyName=Name, PropertyPrice=Price.

**knowledgeProperty**(**land**(Name, Price), PropertyName, PropertyPrice) :- PropertyName=Name, PropertyPrice=Price.

**knowledgeProperty**(**seaTransport**(Name, Price), PropertyName, PropertyPrice) :- PropertyName=Name, PropertyPrice=Price.

Часть 2.

**phonebook**(ivanov, "26121", **adr**(moscow, baumanskaya, **14**, **1**)).

**phonebook**(smirnov, "63173", **adr**(moscow, kutuzovskaya, **52**, **41**)).

**phonebook**(petrov, "55112", **adr**(saratov, pervaya, **3**, **4**)).

**phonebook**(ivanov, "26141", **adr**(moscow, baumanskaya, **14**, **1**)).

**contributor**(ivanov, narodniy, "12356123", **320042**).

**contributor**(smirnov, russkiy, "56191332", **452144**).

**contributor**(petrov, perviy, "15126522", **521242**).

**contributor**(petrov, russkiy, "42125312", **523621**).

**owner**(smirnov, **auto**(opel, **120000**)).

**owner**(petrov, **land**(kubinka, **134124**)).

**owner**(petrov, **building**(cottege, **4124124**)).

**owner**(ivanov, **seaTransport**(sublue, **735224**)).

**owner**(ivanov, **auto**(volkswagen, **893456**)).

**owner**(smirnov, **land**(vnukovo, **520000**)).

**getInfoBySurnameAndCity**(Surname, City, Street, Bank, Phone) :-

**phonebook**(Surname, Phone, **adr**(City, Street, **\_**, **\_**)),

**contributor**(Surname, Bank, **\_**, **\_**).

**ownership**(Surname, Name, Price) :- **owner**(Surname, Property), **knowledgeProperty**(Property, Name, Price).

goal

% zadanir 1 ownership(petrov, Name, \_).

% zadanie 2 ownership(ivanov, Name, Price).

% ownership(petrov, Name, Price).

№16.

Исправленный листинг программы:

domains

name = symbol.

predicates

**mother**(name, name).

**father**(name, name).

**ancestor**(name, name, name, name, name).

clauses

**father**(nikita, vladislav).

**father**(vladislav, alexander).

**father**(olya, petya).

**father**(alexey, ivan).

**father**(ivan, denis).

**father**(vasilisa, andrey).

**mother**(nikita, olya).

**mother**(vladislav, marya).

**mother**(olya, tanya).

**mother**(alexey, vasilisa).

**mother**(ivan, angelina).

**mother**(vasilisa, polina).

**ancestor**(Name, MaternalGrandmother, MaternalGrandfather, PaternalGrandmother, PaternalGrandfather) :-

**mother**(Name, Mother), **mother**(Mother, MaternalGrandmother), **father**(Mother, MaternalGrandfather),

**father**(Name, Father), **mother**(Father, PaternalGrandmother), **father**(Father, PaternalGrandfather).

goal

**ancestor**(alexey, MaternalGrandmother, MaternalGrandfather, PaternalGrandmother, PaternalGrandfather).