Лекция 3

Продукционные экспертные системы. В этих системах знания описываются правилами продукции: ЕСЛИ ТО (А → Б) А или анцидент включает в себя n условий связанных логической связкой И, а консиквент это 1 заключение.

Все понятия пишутся в словаре, и каждому понятию сопоставим вершину. Если a1 и a2 и an, то b. a1 – 1, a2 – 2, …, an – n. введем номера для вершин, это будут правила продукции.

Будем использовать в качестве модели правиле продукции двудольный граф И/ИЛИ, а вся база — это семантическая сеть, которая моделируется сетью гиперграфов.

Номера правил это одна доля вершин (обозначим квадратиками). А понятие это кружочки – это вершины графа. Вся база правил образует сеть, либо гиперграф, либо И/ИЛИ двудольный граф

СД

ВЕРШИНА

- Номер вершины (целые)
- Флаг

МОДУЛЬ ПРАВИЛА

- Номер правила
- Целевая вершина
- Массив или список входных вершин, которые связаны связкой И
- Их количество
- Метка правила (целое: закрыто, запрещено или открыто)

БАЗА ЗНАНИЙ описывается списком МОДУЛЕЙ ПРАВИЛ.
Ввести список вершин. Вершины и номера правил УНИКАЛЬНЫ.
Выделим диапазон вершин от 1 до 100 и от 100 до.
Система логического вывода будет представлять из себя класс поиска в графов.
Поля класса ПОИСК:

- Список модуля правил
- Списки открытых вершин и правил
- Списки закрытых вершин и правил
- Списки запрещенных вершин и правил
- Целевая вершина поиска
- Флаг решения (fy)
- Флаг нет решений (fn)
  // правило закрыто, если оно доказано
  // правило запрещено, если не можем доказать для текущих исходных правил

Конструктор класса ПОИСК передаем: текущий список базы знаний, целевую вершину поиска, массив или список исходных данных (исходных вершин), количество исходных данных (количество вершин). В конструкторе флаги устанавливаем в единицу, метки правил и вершин ставим в 0.

В список закрытых вершин, который является частью рабочей памяти… в конструкторе записываем заданные исходные вершины. В процессе поиска мы в этот список будем добавлять доказанные.

Основные методы класса, которые реализует метод поиска в ширину от данных к цели

Вход: целевая вершина и список исходных данных

Два основных метода:

1. Поиск в ширину (от данных). В нём выполняется цикла пока оба флага == 1
2. Родители или закрытые правила мы будем получать.

Рассмотрим второй метод, который формирует списки закрытых (доказанных) правил и списки закрытых (доказанных) вершин. В этом методе остальные списки не используются. В этом методе запускам цикл “пока не конец базы правил и флаг Y==1”. На каждом шаге выбираем первое текущее правила из списка правил при условии, что его метка == 0 (т.е. он не был просмотрен). Необходимо проверить проверяется ли множество его входных вершин закрытыми вершинами.

На первом шаге исходными данными... Если покрытие выполняется, то модуль считается доказанным (закрытым).

- Метка этого правила := 1.
- Само правило → список закрытых правил.
- Его выходную вершину → список закрытых вершин.
- Всем доказанным вершинам флаг := 1.

Выход из цикла либо по окончанию базы правил, либо Y==0.

> 💡 Если правило доказано, то проверим его вершину, не выставлен ли флаг цели, то его в 0

Метод возвращает количество найденных правил. Возможно три варианта

- 0 - нет правил
- не 0 - нашли решения
- …

Метод поиска

В цикле “пока оба флага 1” вызываем метод Родители. Если флаг решения == 0, то return. Иначе если количество найденных правил j == 0 ⇒ нет решения флаг N (?) сбрасываем в 0. На следующем флаге этого цикла (если мы не вышли) мы снова просматриваем список правил, но множество доказанных вершин вместе с исходными данными увеличилось и будут выполняться уже другие правила (возможно).

Для решения нужно получить дерево решения, то есть из множества доказанных правил выбрать те, которые оптимально идут от данных к цели.

Достоинство: простота. Недостаток: избыток правил.