

**Модели представления
знаний
в интеллектуальных
системах**

- **Представление знаний** - структурирование знаний с целью формализации процессов решения задач в определенной проблемной области.
- **Модель представления знаний** - формализм, предназначенный для отображения статических и динамических свойств предметной области.
- В искусственном интеллекте основными моделями представления знаний являются:
 - **продукционные системы,**
 - **семантические сети (когнитивные модели),**

Требования к моделям знаний

- общность (универсальность);
- наглядность представления знаний;
- однородность;
- реализация в модели свойства активности знаний;
- открытость;
- возможность отражения структурных отношений объектов предметной области;
- наличие механизма «проецирования» знаний на систему семантических шкал;
- возможность оперирования нечеткими знаниями;
- использование многоуровневых представлений (данные, модели, метамодел, метаметамодели и т.д.).

Общесистемные принципы к моделям

Принцип системности указывает на то, что при проведении какого-либо анализа, процесса диагностирования должны учитываться не только взаимосвязи между этапами анализа, но и связи с внешней средой.

Принцип многообразия указывает на то, что процесс анализа необходимо исследовать на основе различных подходов и методов (например, методов «мягких вычислений», методологии когнитивного и нечеткого когнитивного моделирования) и не ограничиваться оценками, полученными на основе только одного метода.

Принцип многокритериальности. Согласно данному принципу, какая-либо оценка может быть оптимизирована по многим показателям (научно-технические, производственные, маркетинговые и др.).

Принцип декомпозиции. Согласно данному принципу, процесс диагностирования можно разделить на множество этапов и связей между ними. Например, корректировка модели может рассматриваться как совокупность взаимосвязанных этапов (изменение структуры модели и изменение значений факторов).

Принцип интеграции (композиции) заключается в возможности объединения различными правилами и способами множества этапов процесса диагностирования с помощью связей в единый процесс и в выявлении системных свойств и функций вновь образованного процесса. Использование данного принципа при проведении диагностирования позволяет получить результаты, которые не могут быть получены с помощью отдельных этапов.

На принципах декомпозиции и композиции основаны анализ и синтез сложных систем.

Принцип эквивалентности путей достижения цели заключается в том, что существуют несколько эквивалентных путей для достижения конечного результата, которые отличаются друг от друга используемыми исходными данными, методами и алгоритмами. Например, оценка может быть получена с использованием нечетко-интервального и нечетко-множественного методов.

Принцип системной готовности. Согласно данному принципу, для проведения диагностирования необходимо наличие всех этапов его процесса и окружающей среды. Нарушение данного принципа может привести либо к нарушению сроков создания проекта, либо к его провалу.

Принцип итеративности заключается в том, что проведение диагностирования является процессом, предполагающим определенную последовательность операций, использования методов. Его успех зависит от того, как руководитель проекта выберет эти итерации, и как будем их комбинировать.

Когнитивное моделирование

В работе [Гореловой Г.В] под *когнитивным моделированием* понимается инструмент исследователя для решения набора системных задач (идентификации объекта, сценарного анализа, анализа путей и циклов когнитивной карты, анализа связности и сложности, анализа устойчивости, прогнозирования, принятия решений, задач реализации, наблюдаемости, управляемости, оптимизации, композиции-декомпозиции, теории катастроф, адаптируемости, самоорганизации), что позволяет не только анализировать различные аспекты сложной системы, но и уточнять когнитивные модели.

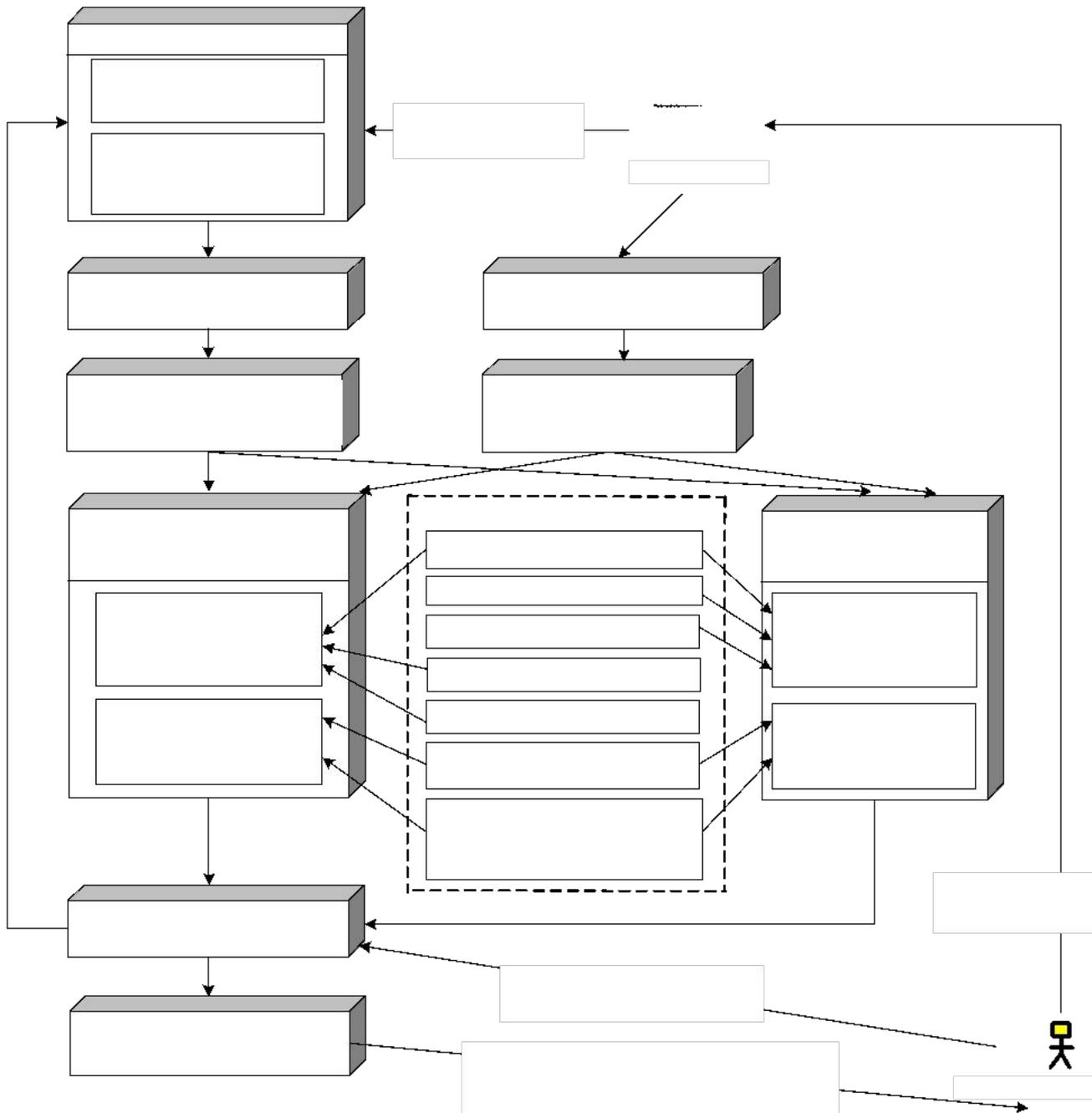
В работе [Авдеева З.К.] под *когнитивным моделированием* понимается исследование функционирования и развития слабоструктурированных систем (СС) и ситуаций посредством построения модели СС (ситуации) на основе когнитивной карты.

В работе [Васильева В.И.] под *когнитивным моделированием* понимается моделирование некоторой предметной области в виде когнитивной карты, объектами которой являются понятия данной предметной области (концепты) и связи между ними, выраженные в соотношениях влияния.

Под *методологией когнитивного моделирования* понимается инструмент для моделирования процесса анализа реализуемости проектов, включающий методы, методики, алгоритмы, предназначенных для решения взаимосвязанных системных задач (поиск циклов когнитивной модели, поиск собственных чисел, топологический анализ, задание управляющих воздействий, установление текущих значений показателей и начальных импульсов), что позволяет анализировать процесс анализа реализуемости проектов, а также уточнять четкие когнитивные модели (ЧКМ).

Под *методологией нечеткого когнитивного моделирования* понимается инструмент для моделирования процесса анализа реализуемости проектов, включающий методы, методики, алгоритмы, предназначенных для решения взаимосвязанных системных задач (топологический анализ, взаимовлияние факторов, обучение нечеткой когнитивной модели (НКМ), задание управляющих воздействий, установление текущих значений показателей и начальных импульсов), что позволяет анализировать процесс анализа реализуемости проектов, а также уточнять НКМ.

**Обобщенная
схема
методологии
когнитивного
и нечеткого
когнитивного
моделирования**



Под четкой когнитивной моделью понимается когнитивная карта (знаковый ориентированный граф), которая получается путем структуризации знаний руководителя, исполнителей проекта на основе теоретических представлений, исходных данных, применения различных экспертных методов

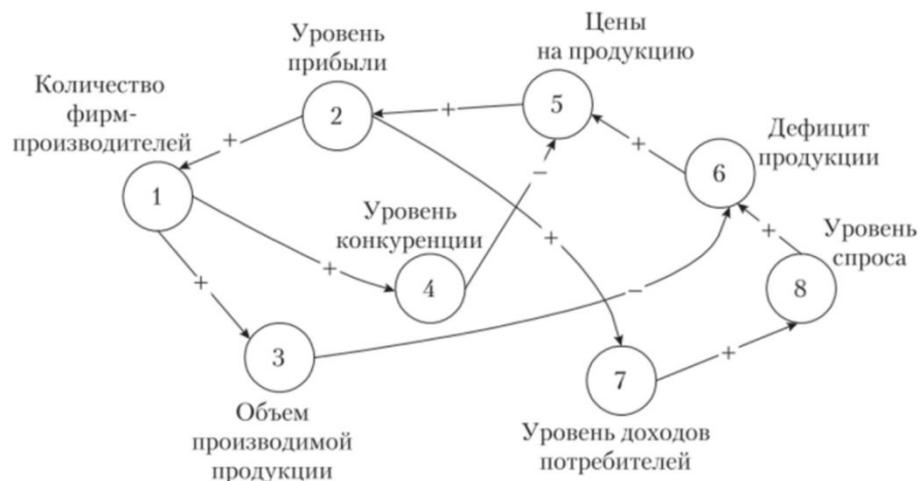
$$G = \langle V, E \rangle,$$

где $V = \{v_i\}$ – множество вершин, $i = \overline{1, h}$, h – количество вершин;

E – бинарное отношение на V (дуги (связи) между вершинами v_i и v_j).

Элементы e_{ij} , $e_{ij} \in E$ ($i, j = \overline{1, h}$) характеризуют направление и силу влияния между вершинами v_i и v_j , $e_{ij} = e(v_i, v_j)$.

Когнитивная модель G может быть представлена матрицей отношений E_G , которая представляет собой квадратную матрицу $E_G = (e_{ij})_{h \times h}$, причем элемент e_{ij} матрицы E_G , стоящий на пересечении i -й строки и j -го столбца, может принимать значения «1», «-1» либо «0».



Для установления связей определена шкала для оценки характера и силы связей между вершинами, где вербальному описанию связи присваивается число

Оценка связей между вершинами

Вербальное описание	Численное значение
Увеличение/увеличение Уменьшение/уменьшение	1
Увеличение/уменьшение Уменьшение/увеличение	-1
Отсутствует	0

Символ «1» обозначает положительную связь между вершинами v_i и v_j , т.е. увеличение (уменьшение) влияния вершины v_i вызывает увеличение (уменьшение) в вершине v_j ; символ «-1» означает отрицательную связь между v_i и v_j , т.е. увеличение (уменьшение) влияния вершины v_i вызывает уменьшение (увеличение) в вершине v_j ; символ «0» означает, что влияние v_i на v_j отсутствует.

1) Анализ структурной устойчивости и устойчивости по возмущению, по начальному значению.

Если в ЧКМ/НКМ отсутствует нечетное число отрицательных циклов, то модель структурно не устойчива. Это означает, что однажды возросшее негативное влияние со стороны каких-либо факторов, если ничего не предпринимать, будет продолжать неограниченно нарастать.

!!! Четный цикл (положительная обратная связь) имеет положительное произведение знаков всех входящих в него дуг, нечетный (отрицательная обратная связь) – отрицательное.

Если критерий устойчивости в теории управления $|M| < 1$, где $|M|$ – максимальное по модулю собственное число (корень характеристического уравнения матрицы отношений ЧКМ/НКМ), модель устойчива по возмущению, по начальному значению.

Если же $|M| > 1$, модель не устойчива ни по возмущению, ни по начальному значению. Это означает, что ЧКМ/НКМ выйдет из равновесного состояния при любом возмущении со стороны.

Пример 4.2. Построить ФП некоторого нечеткого множества \tilde{A} , носитель которого состоит из трех элементов $\{x_1, x_2, x_3\}$, если методом Саати получена следующая матрица парных сравнений:

$$A = \begin{pmatrix} 1 & 1/2 & 1/4 \\ 2 & 1 & 1/2 \\ 4 & 2 & 1 \end{pmatrix}.$$

► Для нахождения собственных значений λ_i матрицы A составим характеристическое уравнение:

$$\det(A - \lambda E) = 0,$$

где E – единичная матрица той же размерности, что и A ; $\det(X)$ – определитель (детерминант) матрицы X .

Имеем:

$$\det(A - \lambda E) = \begin{vmatrix} 1 - \lambda & 1/2 & 1/4 \\ 2 & 1 - \lambda & 1/2 \\ 4 & 2 & 1 - \lambda \end{vmatrix} = 0.$$

Раскрывая определитель, например, по правилу треугольников, получаем:

$$(1 - \lambda)^3 + 2 \cdot 2 \cdot \frac{1}{4} + 4 \cdot \frac{1}{2} \cdot \frac{1}{2} - 4 \cdot (1 - \lambda) \cdot \frac{1}{4} - 2 \cdot \frac{1}{2} \cdot (1 - \lambda) \cdot 2 = 0;$$

$$(1 - \lambda)^3 - (1 - \lambda) - 2(1 - \lambda) + 2 = 0;$$

$$(1 - \lambda)^3 - 3(1 - \lambda) + 2 = 0;$$

$$(1 - \lambda)^3 + 3\lambda - 1 = 0;$$

$$1 - 3\lambda + 3\lambda^2 - \lambda^3 + 3\lambda - 1 = 0;$$

$$3\lambda^2 - \lambda^3 = 0;$$

$$\lambda^2(3 - \lambda) = 0.$$

Получено два корня: один простой $\lambda_1=3$, а другой $\lambda_2=0$ имеет кратность 2. Максимальное собственное значение $\lambda_{\max}=3$.

Под *нечеткой когнитивной моделью* понимается нечеткая когнитивная карта, в которой вершины представляют факторы, а ребра – нечеткие причинно-следственные связи между факторами

$$G_{\text{неч}} = \langle V, W \rangle,$$

где $V = \{v_i\}$ – множество вершин, $v_i \in V$, $i = \overline{1, h}$, h – количество вершин;

$X = \{x_{v_i}\}$ – множество параметров вершин, $i = \overline{1, h}$. При этом каждой вершине ставится один параметр;

W – нечеткие причинно-следственные связи между вершинами. Элементы w_{ij} , $w_{ij} \in W$ характеризуют направление и силу влияния между вершинами v_i и v_j

$$w_{ij} = w(v_i, v_j),$$

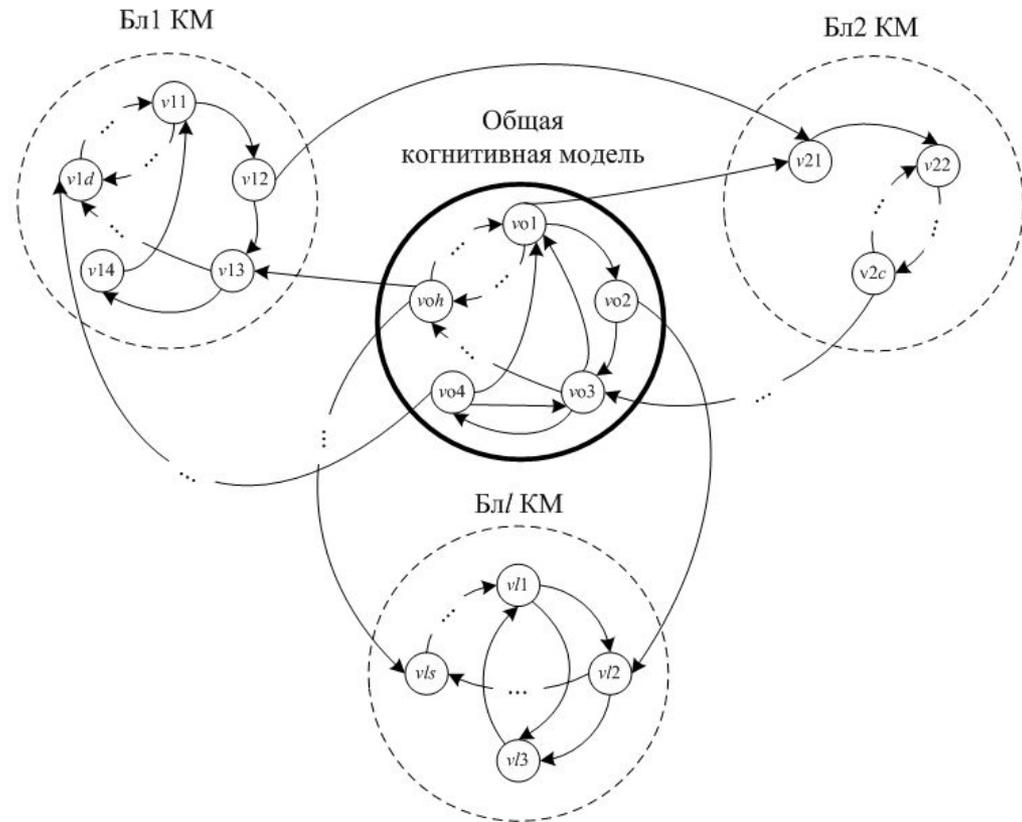
где w – показатель интенсивности влияния. Элементы w_{ij} обладают следующими свойствами [179]:

- 1) w_{ij} принимает значения из интервала $w_{ij} \in [-1, 1]$, т.е. $-1 \leq w_{ij} \leq 1$;
- 2) $w_{ij} = 0$, если влияние v_i на v_j , отсутствует;
- 3) $0 < w_{ij} \leq 1$ при положительном влиянии v_i на v_j , т.е. увеличение значения вершины v_i приводит к увеличению значения вершины v_j ;
- 4) $-1 \leq w_{ij} < 0$ при отрицательном влиянии v_i на v_j , т.е. увеличение значения вершины v_i приводит к уменьшению значения вершины v_j .

Отметим, что в общем случае, пока работа не ведется с ЧКМ и/или НКМ как с математической моделью, оперируем понятием «факторы». Как только началась работа с ЧКМ и/или НКМ, оперируем понятием «вершина».

Существует два подхода к построению ЧКМ и/или НКМ [35, 52]:

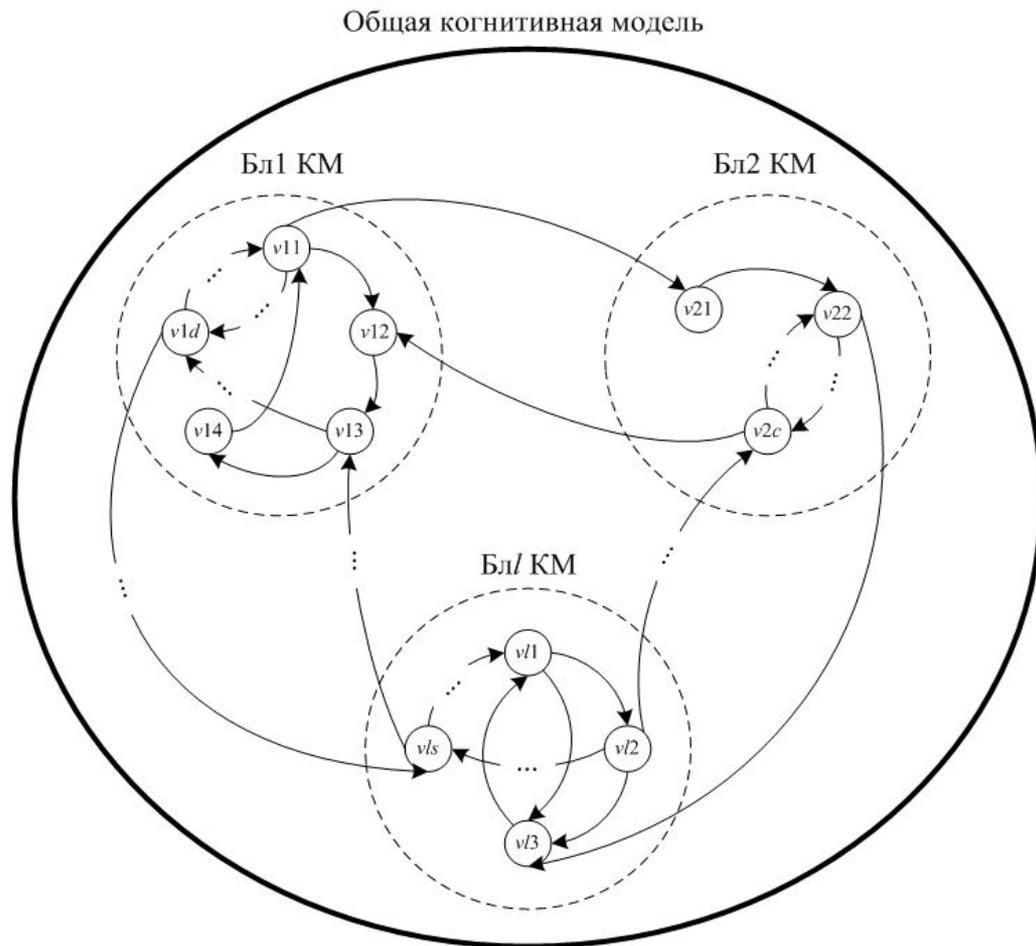
1) «Сверху». На начальном этапе строится общая ЧКМ и/или НКМ, которая в дальнейшем достраивается с помощью отдельных блоков когнитивной модели (в структуру включаются как качественные факторы, так и количественные)



Подход «сверху» к построению четкой и/или нечеткой когнитивной модели

2) «Снизу». На начальном этапе сначала строятся отдельные блоки когнитивной модели, а затем происходит их объединение в общую четкую когнитивную модель.

Здесь Бл1 КМ, Бл2 КМ, ..., Бл l КМ – отдельные блоки ЧКМ и/или НКМ ($r = \overline{1, l}$), которые достраиваются до общей ЧКМ и/или НКМ; $v_{o1}, v_{o2}, \dots, v_{oh}$ – вершины общей ЧКМ и/или НКМ.



Подход «снизу» к построению четкой и/или нечеткой когнитивной модели

2) Метод анализа импульсных процессов .

При исследовании динамики поведения НКМ решаются следующие задачи:

- выделяются управляющие вершины, на которые подается внешнее воздействие;
- выделяются целевые вершины, состояние которых оценивается в процессе проведения вычислительного эксперимента;
- строятся графики импульсных процессов с четырьмя вершинами для лучшего восприятия и понимания графика (по оси абсцисс отмечены шаги импульсного процесса t , по оси ординат – изменение значений вершин в процентах, %). На графиках отображается такое количество шагов t , которое отражает тенденции изменений значений факторов.

Неоднородная когнитивная модель $G_{\text{неод}}$ может быть представлена матрицей отношений (смежности) $W_{G_{\text{неод}}}$, которая представляет собой квадратную матрицу $W_{G_{\text{неод}}} = (w_{ij})_{h \times h}$.

$$W_{G_{\text{неод}}} = \begin{matrix} & \begin{matrix} 1 & 2 & \dots & h \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ \dots \\ h \end{matrix} & \begin{pmatrix} 0 & w_{12} & \dots & w_{1h} \\ w_{21} & 0 & \dots & w_{2h} \\ \dots & \dots & \dots & \dots \\ w_{h1} & w_{h2} & \dots & 0 \end{pmatrix} \end{matrix} \quad (1)$$

Импульсный процесс в НКМ задается следующим образом:

$$x_{v_i}(t+1) = x_{v_i}(t) + \sum_{j=1}^h w_{j,i} p_j(t), \quad (2)$$

где $x_{v_i}(t+1)$, $x_{v_i}(t)$ – значения вершины v_i на шаге $t+1$ и t соответственно, где $t = 1, 2, 3, \dots$; $p_j(t)$ – изменение в вершине v_j на шаге t , $p_j(t) = x_{v_j}(t) - x_{v_j}(t-1)$; w_{ij} – вес ребра между вершинами v_i и v_j .

Согласно формуле (2), если, например, имеется ребро с весом w между вершинами v_i и v_j и значение вершины v_j возрастает (убывает) на шаге t на некоторое число q , то значение вершины v_i на шаге $t+1$ возрастает (убывает) на величину qw .

Отметим, что при проведении импульсного процесса, дополнительно вводятся различающиеся понятия исходное $x_{v_i}^{(исх)}$ и начальное $x_{v_i}(0)$ значения в каждой вершине v_i , которые связаны между собой следующим соотношением:

$$x_{v_i}(0) = x_{v_i}^{(исх)} + p_j(0),$$

где $p_j(0)$ – начальный импульс вершины v_i (изменение на шаге $t = 0$).

При этом предполагается, что если в вершине v_j возник импульс $p_j(t)$ на шаге t , то он передается за единицу.

Для рассмотрения изменения значений вершин НКМ введем обозначения:

1) $x_v^{(исх)} = (x_{v_1}^{(исх)}, x_{v_2}^{(исх)} = f(x_{v_s}^{(исх)}), \dots, x_{v_h}^{(исх)})$ – вектор исходных значений вершин, где некоторые значения вершин вычисляются с помощью функций), где значение вершины $x_{v_2}^{(исх)}$ вычисляется с помощью функции, которая зависит от одного аргумента ($x_{v_s}^{(исх)}$ – исходное значение вершины s);

2) $x_v(t) = (x_{v_1}(t), x_{v_2}(t), \dots, x_{v_h}(t))$ – вектор значений вершин на шаге $t > 0$;

3) $p(0) = (p_1(0), p_2(0), \dots, p_h(0))$ – вектор начальных импульсов, при $t = 0$;

4) $p(t) = (p_1(t), p_2(t), \dots, p_h(t))$ – вектор импульсов на шаге $t > 0$.

5) $W_{Гнеод}$ – матрица смежности НКМ размером $h \times h$.

С учетом выше рассмотренных обозначений изменения значений вершин НКМ в ходе развития импульсного процесса представляются следующим выражением:

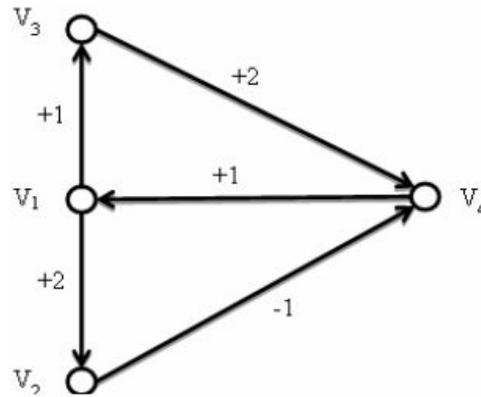
$$x_v(t) = x_v^{(исх)} + (I + W_{Гнеод} + W_{Гнеод}^2 + \dots + W_{Гнеод}^t)^T p(0), \quad (3)$$

$$p(t) = (W_{Гнеод}^T)^t p(0) \quad (4)$$

где I – единичная матрица размером $h \times h$.

Таким образом, если известен вектор начальных импульсов $p(0)$ и матрица смежности $W_{Гнеод}$, то по формуле (4) можно вычислить вектор импульсов во всех вершинах НКМ на любом шаге t . Если известен начальный вектор значений вершин, то по формуле (3) можно вычислить вектор значений всех вершин НКМ на любом шаге t и тем самым полностью решить задачу прогноза.

Пусть дана система в виде некоторого множества влияющих друг на друга вершин V_i , где $i = 1, 2, \dots, n$ (рис.). Взаимодействие описывается дугами, направленными от влияющей вершины к вершине, на которую оказывается влияние. Вес дуги — число, отражающее степень и направление влияния вершин: если дуга положительна, то влияние позитивное, если вес дуги отрицательный, то негативное.



Система взаимосвязанных вершин в графовом представлении

Пусть в исходном состоянии значения в вершинах графа заданы некоторым вектором начального состояния $V(n)$. В нашем примере (рис. 2) этот вектор нулевой, т. е. $V_i(n) = 0$. Пусть вследствие действия некоторого внешнего фактора в i -й вершине графа возник импульс. Этот импульс называется начальным и описывается вектором $P(0)$ величин, на которые увеличивается на данном шаге вес соответствующих вершин графа. В данном случае импульс был подан в первую вершину и был сформирован вектор начального импульса: $P(0) = (1, 0, 0, 0)$.

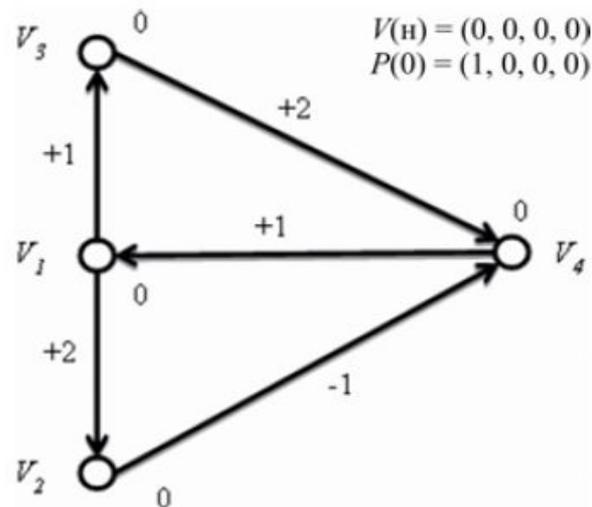


Рис. 2. Система в начальном состоянии

После введения начального импульса система переходит в нулевое состояние $V(0)$ (рис. 3, а) по схеме связей, описывающих взаимное влияние вершин. На следующей итерации система перейдет в состояние $V(1)$ (рис. 3, б) и т. д.

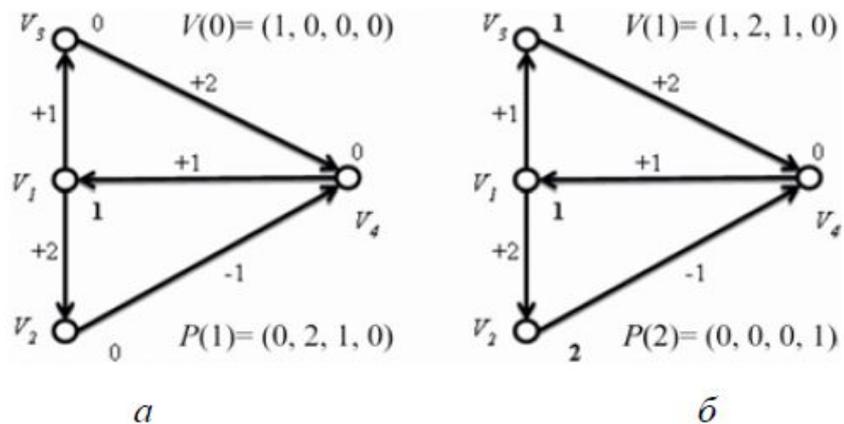


Рис. 3. Состояния вершин графа в моменты времени: а — при $t = 0$; б — при $t = 1$

Состояние вершин орграфа в любой момент времени определяется по формуле

$$V(t) = V(\mathbf{n}) + \left(\sum_{i=0}^t A^i \right) P(0),$$

где A — матрица смежности орграфа.

Предполагается, что возведение матрицы в нулевую степень переводит ее в единичную. Таким образом, для того чтобы определить состояние системы в любой момент времени, достаточно знать ее начальное состояние и начальный импульс внешнего воздействия. Вектор импульса определяется как разность вектора конечного и вектора предыдущего состояний:

$$P(t) = V(t) - V(t-1).$$

3) **топологический анализ структуры НКМ**, который позволит главному инженеру правильно выбрать целевые и управляющие вершины модели; рассчитать системные показатели НКМ с использованием методологии нечеткого когнитивного моделирования, которые позволят увидеть, какие вершины и как сильно они влияют на НКМ .

Оценка характера и силы связей между верши

представленными в виде вербальных описаний

Вербальное описание	Численное значение
Отсутствует	0
Очень слабо усиливает	[0,1, 0,3]
Очень слабо ослабляет	[- 0,1, - 0,3]
Слабо усиливает	[0,31, 0,5]
Слабо ослабляет	[- 0,31, - 0,5]
Умеренно усиливает	[0,51, 0,7]
Умеренно ослабляет	[- 0,51, - 0,7]
Сильно усиливает	[0,71, 0,9]
Сильно ослабляет	[- 0,71, - 0,9]
Очень сильно усиливает	[0,91, 1]
Очень сильно ослабляет	[- 0,91, -1]

Для определения взаимовлияния вершин друг на друга необходимо перейти от матрицы отношений $W_{\text{Гнеч}}$ с нечеткими связями к матрице, состоящей из положительных связей $R_{\text{Гнеч}} = (r_{ij})_{2h \times 2h}$, элементы которой определяются из матрицы $W_{\text{Гнеч}} = (w_{ij})_{h \times h}$ путем замены

$$\begin{cases} \text{если } w_{ij} > 0, \text{ то } r_{2i-1, 2j-1} = w_{ij}, r_{2i, 2j} = w_{ij}, \\ \text{если } w_{ij} < 0, \text{ то } r_{2i-1, 2j} = -w_{ij}, r_{2i, 2j-1} = -w_{ij}. \end{cases}$$

Остальные значения принимают нулевые значения.

Затем осуществляется транзитивное замыкание матрицы $\overline{R^{(m)}}$:
 $\overline{R^{(m)}} = R \vee R^2 \vee \dots \vee R^{(m)}$ следующим образом:

$$\overline{R^{(m)}} = (r_{ij}^{(m)}), \quad i, j = \overline{1; h}, \quad m = \overline{1; h}, \quad h - \text{количество вершин,}$$

$$r_{ij}^{(m)} = \max(r_{ij}^{(m-1)}; r_{im}^{(m-1)} \cdot r_{mj}^{(m-1)}),$$

если $m < h$, то построение матрицы $R^{(m)}$ продолжается;

если $m = h$, то построение матрицы $R^{(m)}$ заканчивается.

Матрица транзитивного замыкания позволяет снизить чувствительность результатов к входным воздействиям, а также учесть все имеющиеся в НКМ опосредованные (неявные) взаимовлияния вершин друг на друга. Поскольку размерность данной матрицы может быть велика, то данный результат предлагается оформить в виде матрицы взаимовлияния (4.17), состоящей из положительно-отрицательных пар $Z = ((z_{ij}, \bar{z}_{ij}))_{h \times h}$ (z_{ij} характеризует силу положительного влияния i -го фактора на j -й, \bar{z}_{ij} – силу отрицательного влияния),

$$Z = \begin{matrix} & \begin{matrix} 1 & 2 & \dots & h \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ \dots \\ h \end{matrix} & \left(\begin{array}{cccc} (z_{11}, \bar{z}_{11}) & (z_{12}, \bar{z}_{12}) & \dots & (z_{1h}, \bar{z}_{1h}) \\ (z_{21}, \bar{z}_{21}) & (z_{22}, \bar{z}_{22}) & \dots & (z_{2h}, \bar{z}_{2h}) \\ \dots & \dots & \dots & \dots \\ (z_{h1}, \bar{z}_{h1}) & (z_{h2}, \bar{z}_{h2}) & \dots & (z_{hh}, \bar{z}_{hh}) \end{array} \right), \end{matrix}$$

образованных по правилу:

$$\begin{cases} z_{ij} = \max(r_{2i-1,2j-1}, r_{2i,2j}), \\ \bar{z}_{ij} = -\max(r_{2i-1,2j}, r_{2i,2j-1}). \end{cases}$$

На основе матрицы Z рассчитываются системные показатели НКМ позволяющие выделять управляемые вершины, оказывающие наибольшее влияние на целевые вершины и оценивать степень этого влияния:

- Влияние i -й вершины на j -ю

$$p_{ij} = \text{sign}(z_{ij} + \bar{z}_{ij}) \max(|z_{ij}|, |\bar{z}_{ij}|), \quad z_{ij} \neq -\bar{z}_{ij}.$$

- Влияние i -й вершины на НКМ

$$\vec{P}_i = \frac{1}{n} \sum_{j=1}^n p_{ij}.$$

- Влияние НКМ на j -ю вершину

$$\bar{P}_j = \frac{1}{n} \sum_{i=1}^n p_{ij}.$$

- Консонанс влияния i -й вершины на j -ю

$$c_{ij} = \frac{|z_{ij} + \bar{z}_{ij}|}{|z_{ij}| + |\bar{z}_{ij}|}.$$

Отметим, что *консонанс* – функция, учитывающая положительное и отрицательное влияния вершины на вершину, т.е. *консонанс* выражает меру доверия к знаку (положительному и отрицательному влиянию) и силе воздействия. Консонанс определяет, насколько согласованно присутствие факторов в НКМ. Причем, чем выше консонанс, тем убедительнее мнение о знаке влияния.

- Консонанс влияния i -й вершины на НКМ

$$\bar{C}_i = \frac{1}{n} \sum_{j=1}^n c_{ij}.$$

- Консонанс влияния НКМ на j -ю вершину

$$\bar{C}_j = \frac{1}{n} \sum_{i=1}^n c_{ij}.$$

- Диссонанс влияния i -й вершины на НКМ. Диссонанс выражает меру недоверия к результату и может появиться по различным причинам, например, из-за логического несоответствия или из-за несоответствия прошлого опыта настоящей ситуации.

$$D_i = \frac{1}{n} \sum_{j=1}^n d_{ij}.$$

- Диссонанс влияния НКМ на j -ю вершину

$$D_j = \frac{1}{n} \sum_{i=1}^n d_{ij}.$$

Отметим, что консонанс и диссонанс являются противоположными, поэтому $\overline{C}_i + \overline{D}_i = 1$, $\overline{C}_j + \overline{D}_j = 1$. На данном этапе принимается решение о корректировке / не корректировке НКМ.

Фреймовая модель

Фреймовая модель представления знаний – это модель, в которой структура знаний предметной области формализуется в виде совокупности взаимосвязанных фреймов, описывающих объекты, а свойства этих объектов и факты, относящиеся к ним, описываются в структурных элементах фрейма.

Фреймом (англ. frame – рамка, каркас) называется структура данных для представления некоторого концептуального объекта.

Информация, относящаяся к фрейму, содержится в составляющих его слотах.

Слот (англ. slot – щель, прорезь) может быть терминальным (листом иерархии) или представлять собой фрейм нижнего уровня.

Фрейм имеет имя, служащее для идентификации описываемого им понятия, и содержит ряд описаний – слотов, с помощью которых определяются основные структурные элементы этого понятия. Слот может содержать не только конкретное значение, но и имя процедуры, вычисляющей это значение по заданному алгоритму.

Фреймовые системы подразделяются на *статические* и *динамические*. Динамические допускают изменение фреймов в процессе решения задачи.

Фрейм можно представить:

а) в виде цепочки:

Фрейм: <слот 1>, <слот 2>, ... , <слот N>

или более подробно, как список свойств:

Имя ФРЕЙМА: <имя 1-го слота: значение 1-го слота>, <имя 2-го слота: значение 2-го слота>, <...>, <имя N-го слота: значение N-го слота>.

б) в виде таблицы:

Имя фрейма	
Имя слота 1	Значение слота 1
Имя слота 2	Значение слота 2
Имя слота N	Значение слота N

$N: \{ \langle S_1, V_1, P_1 \rangle, \dots, \langle S_k, V_k, P_k \rangle, \dots, \langle S_n, V_n, P_n \rangle \}$

- N – имя фрейма
- S_k – имя слота
- V_k – значение слота
- P_k – процедура

Лекция	
Предмет	Физика
Лектор	Иванов И.И.
Аудитория	109

Пример фрейма **РУКОВОДИТЕЛЬ**

Имя слота	Значение слота	Тип значения слота
Имя	Иванов И.И.	Строка символов
Дата_рождения	01.01.1965	Дата
Возраст	Age(дата, дата_рождения)	Процедура
Специальность	Юрист	Строка символов
Отдел	Отдел кадров	Строка символов
Зарплата	50000	Число

Типы фреймов

1) Фрейм-прототип – это фрейм, содержащий знания о самом понятии и хранящийся в БЗ.

Битва	
Субъект (кто?)	X1
Объект (с кем?)	X2
Место (где?)	X3
Время (когда?)	X4
Результат	X5

2) Фрейм-экземпляр - это фрейм, содержащий конкретное описание понятия или знания. Создаются для отображения реальных фактических ситуаций на основе поступающих данных

Куликовская битва	
Субъект (кто?)	Князь Дмитрий
Объект (с кем?)	Хан Мамай
Место (где?)	Куликово поле
Время (когда?)	Утром в сентябре 1380 года
Результат	Победа князя Дмитрия

Модель фрейма является достаточно универсальной, поскольку в состав фрейма могут входить слоты с именами действий, то фреймы годятся для представления как декларативных, так и процедурных знаний.

Фрейм является *простым*, если он не содержит в себе других фреймов. *Сложный* (составной) фрейм содержит в себе два и более фрейма, и по существу представляет сеть фреймов.

Фрейм позволяет отобразить все многообразие знаний о мире через:

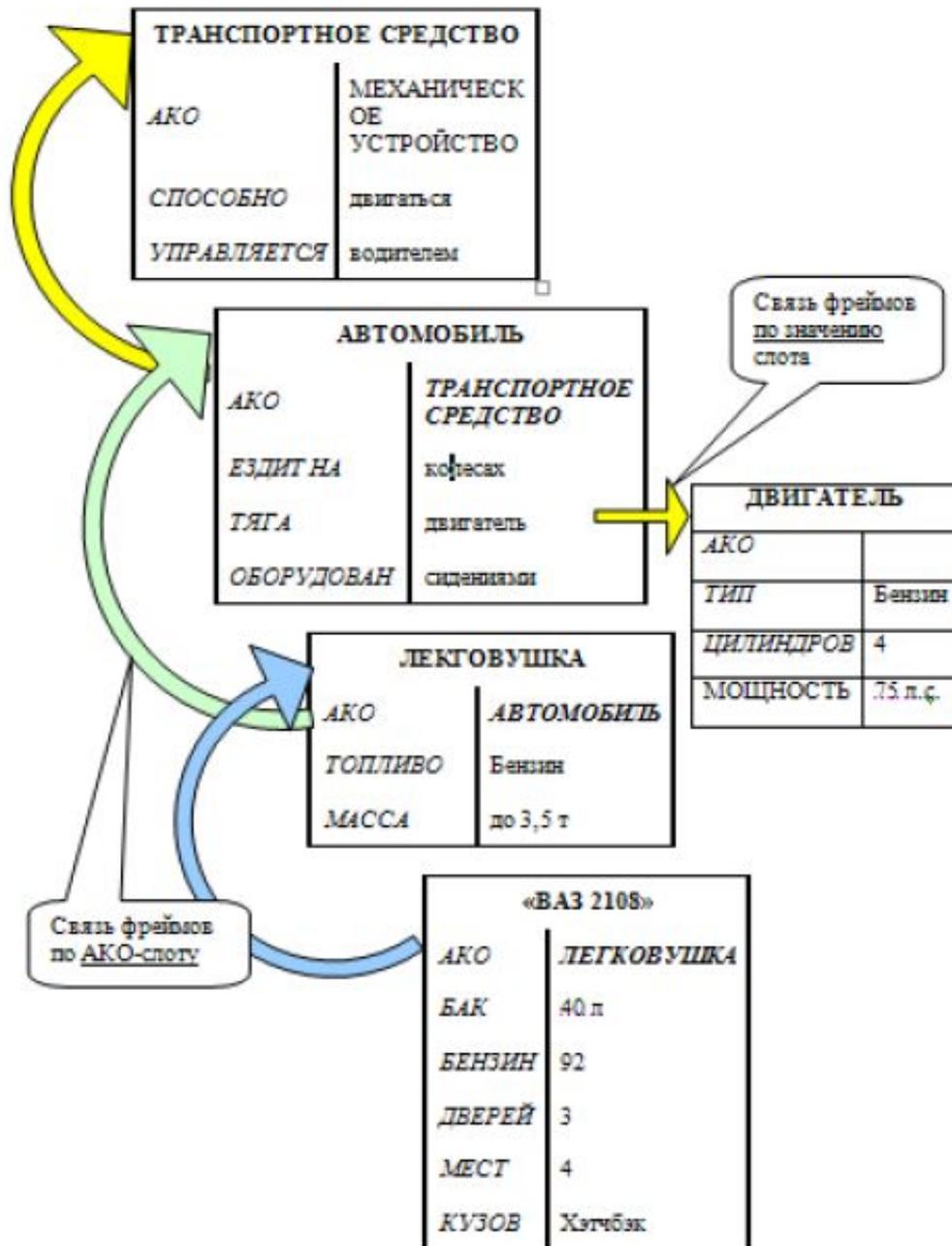
- фреймы – *структуры, использующиеся для обозначения объектов и понятий (заем, залог, вексель);*
- фреймы – *роли (менеджер, кассир, клиент);*
- фреймы – *сценарии (банкротство, собрание акционеров, празднование именин);*
- фреймы – *ситуации (тревога, авария, рабочий режим устройства) и др.*

Важнейшим свойством теории фреймов является *заимствование* из теории семантических сетей — так называемое наследование свойств. Во фреймах наследование происходит по *АКО-связям* (A-Kind-Of= это), которые связывают фреймы с фреймами, находящимися на уровень выше в иерархии, откуда неявно наследуются (переносятся) значения слотов. Слот АКО указывает на фрейм более высокого уровня иерархии, откуда неявно наследуются, то есть переносятся, значения аналогичных слотов.

Понятие «Лектор».



В данном случае представлено одно звено иерархии (ЧЕЛОВЕК-ЛЕКТОР). Здесь фрейм "ЧЕЛОВЕК" является обобщающим для фрейма "ЛЕКТОР". Таким образом, фрейм "ЛЕКТОР" наследует от фрейма "ЧЕЛОВЕК" значение слота "УМЕЕТ" (а также других слотов, не показанных в примере). Цепочка наследования может быть продолжена вплоть до, например, фрейма "ЖИВОЕ СУЩЕСТВО".



Связь фреймов осуществляется по значению слота. Такая структура позволяет систематизировать большой объем информации, оставляя ее при этом максимально удобной для использования. Кроме того, система (сеть) фреймов способна отражать концептуальную основу организации памяти человека.

ПРИМЕР ПОСТРОЕНИЯ ФРЕЙМОВОЙ МОДЕЛИ

Для построения фреймовой модели представления знаний необходимо выполнить следующие шаги:

1. Определить абстрактные объекты и понятия предметной области, необходимые для решения поставленной задачи. Оформить их в виде фреймов-прототипов (фреймов-объектов, фреймов-ролей).

2. Задать конкретные объекты предметной области. Оформить их в виде фреймов-экземпляров (фреймов-объектов, фреймов-ролей).

3. Определить набор возможных ситуаций. Оформить их в виде фреймов-ситуаций (прототипы). Если существуют прецеденты по ситуациям в предметной области, добавить фреймы-экземпляры и/или фреймы-ситуации.

4. Описать динамику развития ситуаций через набор сцен. Оформить их в виде фреймов-сценариев.

5. Добавить фреймы-объекты сценариев и сцен, которые отражают данные конкретной задачи.

к **достоинствам** фреймовой модели знаний относятся:

- Внутренняя интерпретация и наличие внутренней структуры связей;
 - Возможность использования предположений и ожиданий;
 - Механизм наследования свойств;
 - Универсальность за счет существования не только фреймов для обозначения объектов и понятий, но и фреймов - событий, фреймов - ситуаций, фреймов - ролей, фреймов - сценариев и т. п.;
 - Возможность легкого перехода к сетевой модели.
-
- **Основным недостатком** фреймовой модели представления знаний является отсутствие строгой формализации.

Логическая модель представления знаний

Логическая модель представляет собой формальную систему — некоторое логическое исчисление как правило, исчисление предикатов первого порядка, когда предметная область или задача описывается в виде набора аксиом.

Все знания о предметной области описываются в виде формул этого исчисления или правил вывода. Описание в виде формул дает возможность представить декларативные знания, а правила вывода — процедурные знания.

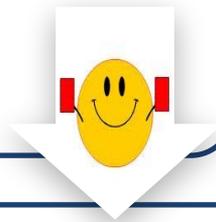
Предикат в исчислении предикатов специальный знак, отражающий определенное отношение между конечным множеством сущностей - аргументов. Предикат первого порядка имеет два состояния - истина и ложь.

Элементы логики высказывания

Обозначение	Название
\neg	отрицание, логическое НЕ
\wedge	конъюнкция, логическое И, логическое умножение
\vee	дизъюнкция, логическое ИЛИ, логическое сложение
\supset или \rightarrow	импликация, ЕСЛИ ... ТО...
\equiv или \leftrightarrow	эквивалентность
$\neg \equiv$	неравнозначность

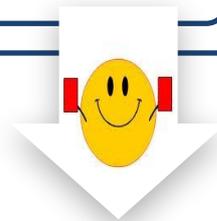
При работе с логическими моделями необходимо соблюдать следующие правила:

Порядок аргументов должен всегда задаваться в соответствии с интерпретацией предикатов принятой в данной предметной области. Программист принимает решение о фиксированном порядке аргументов и соблюдает его от начала до конца.



Предикат может иметь произвольное число аргументов

Отдельные высказывания, состоящие из предиката и связанных с ним аргументов, могут объединяться в сложные высказывания с помощью логических связок: *И* (*AND*, *и*), *ИЛИ* (*or*, *или*), *НЕ* (*not*, \sim), \rightarrow - импликация используемые для формулирования правил по форме: *ЕСЛИ... ТО...*

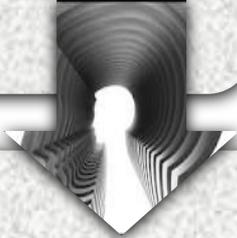


Первому предикату соответствует *имя отношения*, а термину аргументы-*объекты*.

1) Название предиката –
является

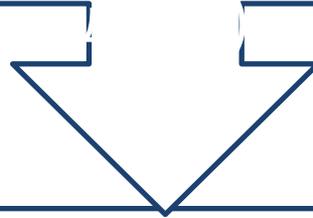


Является (Смит,
специалист по ИТ) \cap
читает (Смит, литература)



*Смит является
специалистом по ИТ и
читает литературу.*

Примеры:



Рассмотрим в качестве примера знание: "Когда температура в печи достигает 120° и прошло менее 30 минут с момента включения печи, давление не может превосходить критическое. Если с момента включения печи прошло более 30 мин, то необходимо открыть вентиль №2".

Логическая модель представления этого знания имеет вид:

$$P(p = 120) \wedge T(t < 30) \rightarrow (D < D_{кр});$$

$$P(p = 120) \wedge T(t > 30) \Rightarrow F(\text{№2}).$$

В этой записи использованы символы логического следования (\Rightarrow).

$P(p = 120)$ — предикат, становящийся истинным, когда температура достигает 120° ;

$T(t < 30)$ — предикат, остающийся истинным в течение 30 мин с начала процесса;

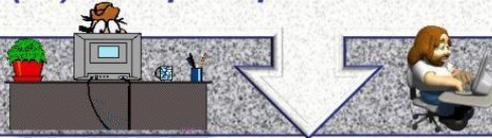
$T(t > 30)$ — предикат, становящийся истинным по истечении 30 мин с начала процесса;

$D < D_{кр}$ — утверждение о том, что давление ниже критического;

$F(\text{№2})$ — команда открыть вентиль №2.

логические связки И (\wedge),
импликации (\rightarrow)
логического следования (\Rightarrow).

1. $\forall(x)$ (специалист по ИТ
(X) \rightarrow программист (X))



*Все специалисты по ИТ
являются программистами*

2. $\exists(x)$ (специалист по ИТ
(X) \rightarrow хорошие программисты (X))



*Некоторые специалисты по ИТ
– хорошие программисты.*

3. $\forall(x) \exists(y)$ (служащий
(X) \rightarrow руководитель (Y, X))



*У каждого служащего
есть руководитель*

4. $\exists(y) \forall(x)$ (служащий
(X) \rightarrow руководитель (Y, X))



*Существует некоторое лицо,
которое руководит всеми.*

Языки представления знаний логического типа широко использовались на ранних стадиях развития интеллектуальных систем, но вскоре были вытеснены (или, во всяком случае, сильно потеснены) языками других типов. Объясняется это громоздкостью записей, опирающихся на классические логические исчисления. При формировании таких записей легко допустить ошибки, а поиск их очень сложен. Отсутствие наглядности, удобочитаемости (особенно для тех, чья деятельность не связана с точными науками) затрудняло распространение языков такого типа.

Исчисление предикатов 1-го порядка в промышленных экспертных системах практически не используется. Эта логическая модель применима в основном в исследовательских «игрушечных» системах, так как предъявляет очень высокие требования и ограничения к предметной области.

Продукционные правила

Продукционные правила

В этой модели знания представляются в виде предложений типа:

***Если** (условие), **то** (действие).*

Под **условием** (антецедентом) понимается некоторое предложение-образец, по которому осуществляется поиск в базе знаний, а под **действием, заключением** (консеквентом) - действия, выполняемые при успешном исходе поиска (они могут быть промежуточными, выступающими далее как условия, и терминальными или целевыми, завершающими работу системы).

Множество продукционных правил формируется на основе анализа опыта человека – специалиста в некоторой области деятельности.

Пример такого продукционного правила:

если Сигнал_датчика_давления = «красный» **то** ситуация = ТРЕВОГА.

Продукционные правила представляются в виде **если** <условие>, **то** <заключение> (действие)

В продукционных системах в качестве условия (часто – множество условий) выступает образец для поиска, а заключением будет вывод о подтверждении некоей гипотезы, факта.

Пример такого продукционного правила:

если Сигнал_датчика_давления = «красный» **то** ситуация = ТРЕВОГА.

В экспертных системах часто используются правила, в которых посылкой является описание ситуации, а заключением – действия, которые необходимо предпринять в данной ситуации.

Пример продукционных правил

Правило 1:

Если топливо поступает в двигатель и двигатель вращается,
то проблема в свечах зажигания.

Правило 2:

Если двигатель не вращается и фары не горят,
то проблема в аккумуляторе или проводке.

Правило 3:

Если двигатель не вращается и фары горят,
то проблема в стартере.

Правило 4:

Если в баке есть топливо и топливо поступает в карбюратор,
то топливо поступает в двигатель.

Вывод на продукциях

Процедура логического вывода в системах, основанных на продукционных моделях, связана с работой специальной программы, управляющей перебором и проверкой правил. Такая программа называется *интерпретатором*, или *машиной вывода*.

Множество продукционных правил образует *базу знаний* содержащую все допустимые зависимости между фактами предметной области и хранящуюся в долговременной памяти.

Также необходимо использовать *рабочую память* – область для хранения истинных фактов, описывающих текущее состояние предметной области. Такая база фактов может пополняться новыми доказанными фактами в процессе работы с продукционными правилами.

Стратегии вывода на продукциях

Механизм логического вывода обеспечивает формирование заключений, воспринимая вводимые факты как истинные, отыскивая правила, в состав которых входят введенные факты, и проверяя возможность выполнения этих правил.

Механизм логического вывода выполняет функции поиска в базе правил, последовательного выполнения операций над знаниями и получения заключений. Существует два способа вывода на продукциях – прямой и обратный вывод.

Прямой вывод: от имеющихся данных – к поиску цели.

Обратный вывод: от поставленной цели – к данным, необходимым для подтверждения цели.

Продукционные системы делят на два типа — с **прямыми** и **обратными** выводами.

- При прямом выводе рассуждение ведется от данных к гипотезам.
- При обратном производится поиск доказательства или опровержения некоторой гипотезы (от цели к данным).

Часто используются комбинации прямой и обратной цепи рассуждений

Существуют два типа выполнения систем продукций: прямой и обратный.

$A \Rightarrow B$ ("ЕСЛИ A , ТО B ")

- **Прямой вывод** называется также выводом, управляемым данными, или нисходящим. В таких системах поиск идет от исходных данных (фактов) к заключениям. Т.е. проверяются условия A , включающие известные факты, и активизируются те продукции, для которых A истинно. После этого в рабочую память заносятся промежуточные заключения B' , которые в дальнейшем выступают как дополнительные факты для A' и так до тех пор, пока не будет получено итоговое заключение B .
- **Обратный вывод** называется также выводом, управляемым целями, или восходящим. В таких системах выдвигается некоторая гипотеза B , а затем идет поиск промежуточных фактов A' , подтверждающих эту гипотезу. После этого в рабочую память заносятся промежуточные факты A' , которые в дальнейшем выступают как промежуточные гипотезы (заключения) B' . Если принятая гипотеза B приводит к известным фактам A , то она считается итоговым заключением.

- **Прямой вывод** рекомендуется использовать в следующих случаях:
 - - все или большинство исходных данных заданы в постановке задачи (например, в экспертной системе PROSPECTOR);
 - - существует большое количество потенциальных целей, но мало способов использования фактов (например, в экспертной системе DENDRAL эффективно использовались факт, что для любого органического соединения существует чрезвычайно большое число возможных структур, однако данные масс-спектрографа позволяют оставить лишь небольшое количество таких комбинаций);
 - - сформировать цель или гипотезу очень трудно (например, в экспертной системе DENDRAL изначально мало информации о возможной структуре соединения).
- **Обратный вывод** рекомендуется использовать в следующих случаях:
 - - цель поиска или гипотеза явно присутствует в постановке задачи или может быть легко сформулирована (например, при доказательстве математических теорем);
 - - имеется большое количество правил, которые на основе полученных фактов продуцируют всевозрастающее число заключений и целей. Своевременный отбор целей позволяет отсеять множество возможных ветвей, что делает процесс поиска более эффективным;
 - - исходные данные не приводятся в задаче, но подразумевается, что они должны быть известны решателю. Например, выбирается предварительный медицинский диагноз (гипотеза), а потом под него подбираются симптомы (факты).

Например, имеется фрагмент базы знаний из двух правил:

71: Если "отдых - летом" и "человек - активный", то "ехать в горы",

72: Если "любит солнце", то "отдых летом",

Пусть в систему поступили данные :
"человек активный" и "любит "солнце"

Прямой вывод

Необходимо, исходя из данных, получить ответ.

1-й проход.

- Шаг 1. Правило П1, не работает (не хватает данных "**отдых - летом**").
- Шаг 2. Правило П2, работает, в базу поступает факт "**отдых - летом**".

2-й проход.

- Шаг 3. Правило П1, срабатывает, активируется цель "**ехать в горы**", которая и выступает как совет, который дает ЭС.

Обратный вывод

Необходимо подтвердить выбранную цель при помощи имеющихся правил и данных.

1-й проход.

● Шаг 1. Цель - "**ехать в горы**":

пробуем П1 - данных, "**отдых - летом**" нет, они становятся новой целью, и ищется правило, где она в правой части.

● Шаг 2. Цель "**отдых - летом**":

правило П2 подтверждает цель и активирует ее.

2-й проход.

● Шаг 3. Пробуем П1, подтверждается искомая цель.

Прямой вывод

Мы видим, что в процессе прямого вывода последовательно выводятся новые факты, начиная с уже известных.

Однако отсутствие связи между фактами может привести к обрыву процедуры и конечный результат не всегда может быть получен.

Кроме того, экспертная система может иметь сотни продукционных правил. Используя прямой вывод можно выполнить множество правил и получить факты, истинные сами по себе, но не имеющие никакого отношения к цели.

В случае, когда надо установить один, конкретный факт, обратный вывод может оказаться предпочтительнее.

Обратный вывод

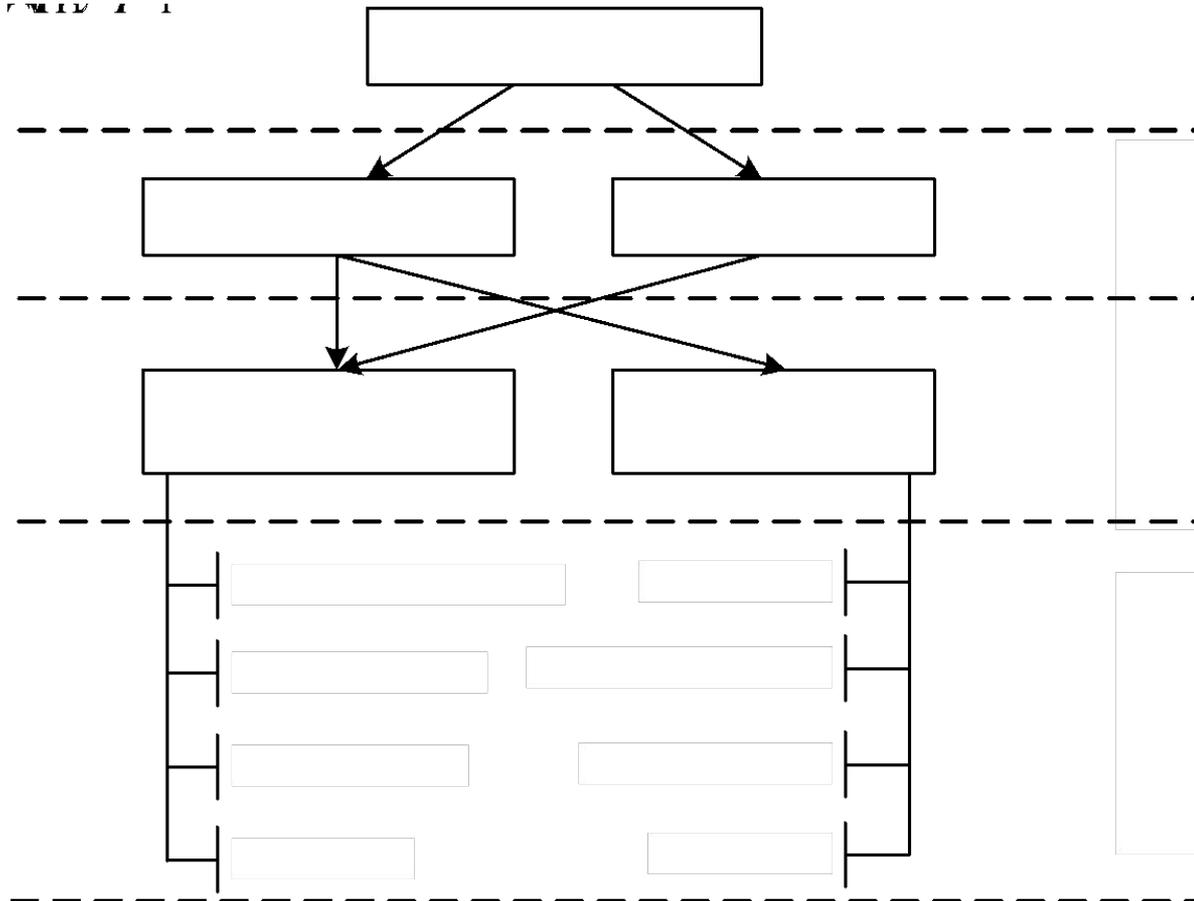
Обратным выводам (обратной цепочке рассуждений) соответствует движение от цели (факта, который требуется установить) к посылкам. В обратном механизме логического вывода работа начинается от поставленной цели. Если цель согласуется с заключением продукции, то посылка (условие) принимается за подцель и делается попытка подтверждения истинности этого факта. Процесс повторяется до тех пор, пока не будут просмотрены все правила, имеющие в качестве заключения требуемый факт.

Продукционная модель чаще всего применяется в промышленных экспертных системах. Она привлекает разработчиков своей наглядностью, высокой модульностью, легкостью внесения дополнений и изменений и простотой механизма логического вывода.

Классификация структурных ошибок в БЗ

Под *ошибкой* будет пониматься нарушение, наблюдаемое в БЗ, содержащей СПП, выявляемое в процессе статического анализа, порождающее проблемы невозможности получить достоверные выводы в определенные моменты времени

Рис. 1.1



Если ошибки в БЗ появляются при сопоставлении отдельных СПП, то осуществляется *статический анализ*, если в цепочке СПП – *динамический анализ*.

Для проведения статического анализа рекомендуется использовать ориентированный граф (вершинам соответствуют входные и выходные переменные, а ребрам – связи между переменными), позволяющий наглядно представить БЗ.

Определение 1. Смешанные продукционные правила представляют собой модифицированную модель нечетких продукционных правил [8, 9].

$$R_k : IF \ x_1 = A_{k1} \ AND \ x_2 = \tilde{A}_{k2} \ AND \ \dots \ AND \ x_n = \tilde{A}_{kn}([x_{knmin}, x_{knmax}]), \ THEN \ y = B_k \ [Rung_k],$$

где x_i – входные переменные, $A_i, i = 1, \dots, n$; A_i – четкие значения переменных; $\tilde{A}_i = \{x_i, \mu_{\tilde{A}_i}(x_i)\}$ – нечеткие значения переменных, $\mu_{\tilde{A}_i}(x_i)$ – функция принадлежности; $\tilde{A}_i([x_{imin}, x_{imax}])$ – значения переменных представлены в виде интервала, где x_{imin}, x_{imax} – минимальное и максимальное значение i -й переменной; которые могут принимать только числовые значения; y – выходная переменная, B – четкое и/или нечеткое значение выходной переменной; $Rung$ – значимость (оценка) правила, $Rung \in \{1, 2, \dots, c\}$, c – количество СПП, $c \in [0, 1]$.

Определение 2. Смешанные продукционные правила называются избыточными, если два правила полностью дублируют друг друга.

Таким образом, БЗ будет являться избыточной, если будут выполняться следующие условия:

1) Отсутствуют полные дубликаты СПП, т.е. в БЗ отсутствуют правила с одинаковыми предусловиями и заключениями. Для наглядности правил представим их в виде ориентированного графа, где вершинам соответствуют входные и выходные переменные, а ребрам – связи между переменными (рисунок 1):

$$R_1: IF x_1 = A_{11} AND x_2 = \tilde{A}_{12} AND \dots AND x_n = \tilde{A}_{1n} ([x_{1nmin}, x_{1nmax}]), THEN y = B_1,$$

$$R_2: IF x_1 = A_{11} AND x_2 = \tilde{A}_{12} AND \dots AND x_n = \tilde{A}_{1n} ([x_{1nmin}, x_{1nmax}]), THEN y = B_1.$$

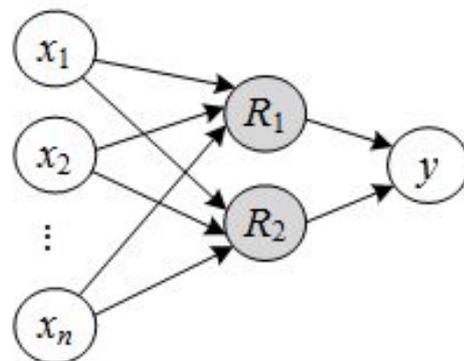


Рисунок 1. Полные дубликаты СПП

Для решения данной проблемы необходимо удалить все СПП дублирования, кроме одного.

2) Отсутствуют неполные дубликаты (рисунок 2):

$$R_1: \text{IF } x_1 = A_{11} \text{ AND } x_2 = \tilde{A}_{12}, \text{ THEN } y = B_1.$$

$$R_2: x_2 = \tilde{A}_{12} \text{ AND } \dots \text{ AND } x_n = \tilde{A}_{2n} ([x_{2n\min}, x_{2n\max}]), \text{ THEN } y = B_1.$$

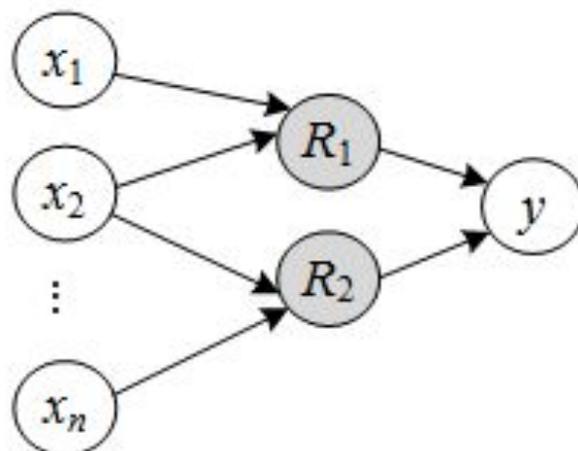


Рисунок 2. Неполные дубликаты СПП

3) Отсутствует включение предусловий (inclusive duplicates), т.е. в БЗ не существует правил с одинаковыми заключениями, таких, что предусловие некоторого правила является подмножеством предусловия другого правила (рисунок 3):

$$R_1 : IF x_1 = A_{11} AND x_2 = \tilde{A}_{12}, THEN y = B_1,$$

$$R_2 : IF x_1 = A_{11} AND x_2 = \tilde{A}_{12} AND \dots AND x_n = \tilde{A}_{2n}([x_{2nmin}, x_{2nmax}]), THEN y = B_1$$

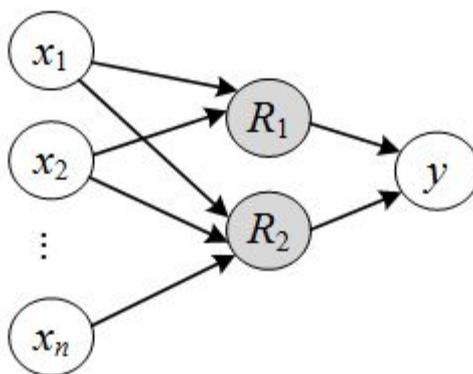


Рисунок 3. Включение предусловий

Отметим, что единого решения для решения данных проблем, а именно исправления условий 2 и 3 не существует, поэтому конкретное решение о способе улучшения БЗ и исправлении структурных ошибок принимает эксперт в каждом конкретном случае.

Определение 3. Смешанные продукционные правила называются циклическими, если цепочка правил в наборе циклических правил образует цикл (рисунок 4), т.е. последовательность правил, в которой предусловия всех правил, кроме последнего, содержатся в заключение следующего правила в последовательности, а предусловия правила содержатся в заключение первого правила последовательности.

$$R_1 : IF x_1 = A_{11} AND x_2 = \tilde{A}_{12}, THEN y = B_1,$$

$$R_2 : IF x_3 = B_1 AND x_4 = \tilde{A}_{23}, THEN y = B_2,$$

$$R_3 : IF x_5 = B_2 AND x_6 = \tilde{A}_{32}, THEN y = \tilde{A}_{12}.$$

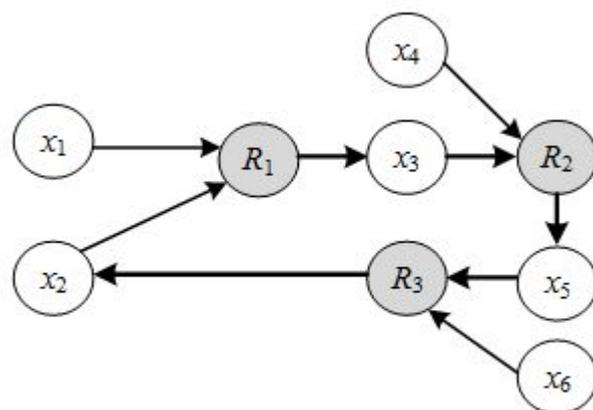


Рисунок 4. Циклические СПП

Таким образом, в БЗ отсутствуют циклические правила, если выполняются следующие условия :

1) Отсутствует прямой цикл, т.е. в БЗ отсутствуют правила, в которых в предусловии и заключении присутствуют термы одной лингвистической переменной (ЛП).

2) Отсутствует не прямой цикл, т.е. в БЗ отсутствуют правила, при которых ЛП определяют ЛП, от которых они зависят.

Смешанные продукционные правила называются правилами недостижимого вывода, если заключение не является целью и не фигурирует в качестве условия ни в одном другом СПП, хранящемся в БЗ.

Определение 4. Смешанные продукционные правила называются противоречивыми, если они имеют одинаковые предусловия (одинаковые значения входных переменных), но различные заключения (различные значения выходных переменных) (рисунок 5):

$$R_1 : IF x_1 = A_{11} AND x_2 = \tilde{A}_{12} AND \dots AND x_n = \tilde{A}_{1n}([x_{1nmin}, x_{1nmax}]), THEN y = B_1,$$

$$R_2 : IF x_1 = A_{11} AND x_2 = \tilde{A}_{12} AND \dots AND x_n = \tilde{A}_{1n}([x_{1nmin}, x_{1nmax}]), THEN y = B_2.$$

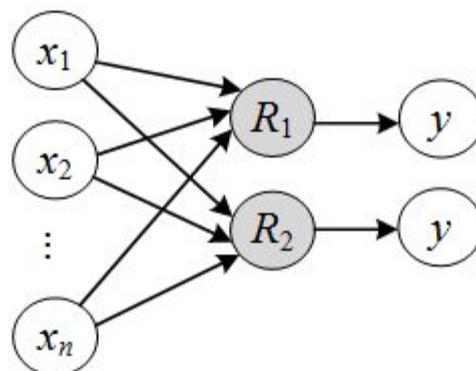


Рисунок 5. Противоречивые СПП

Из рисунка 5 видно, что правила R_1 и R_2 являются противоречивыми, поскольку одно и то же предусловие (A_{11} и $\tilde{A}_{12}, \dots, \tilde{A}_{1n}([x_{1nmin}, x_{1nmax}])$) активизирует разные выходные переменные (термы, значения B_1 и B_2).

Определение 5. База знаний является неполной, если в ней отсутствуют СПП, приводящие к определенным значениям целевых параметров [11].

В качестве ошибки неполноты могут выступать изолированные входные и выходные переменные, т.е. отсутствуют правила, ведущие к ним (рисунок 6).

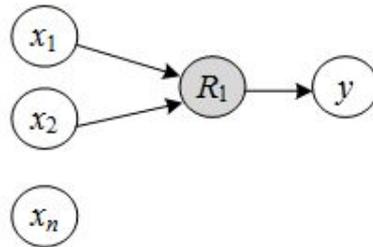


Рисунок 6. Изолированные переменные

Из рисунка видно, что переменная x_n является изолированной. Для решения данной проблемы эксперту предлагается выяснить следующее: если переменная является важной входной либо выходной переменной, то необходимо добавить правило, иначе переменная может быть удалена из БЗ.

Определение 6. База знаний является связной, если для любой пары смежных СПП значение выходной ЛПП тоже смежное, т.е. предусловия этих правил отличаются только одним из n подусловием и в этих подусловий используется одна и та же ЛПП с различными значениями:

$$R_1: IF \ x_1 = A_{11} \ AND \ x_2 = \tilde{A}_{12} \ AND \dots \ AND \ x_n = \tilde{A}_{1n} \ ([x_{1nmin}, x_{1nmax}]), \ THEN \ y = B_1,$$

$$R_2: IF \ x_1 = A_{21} \ AND \ x_2 = \tilde{A}_{22} \ AND \dots \ x_n = \tilde{A}_{2n} \ ([x_{2nmin}, x_{2nmax}]), \ THEN \ y = B_2.$$

Приведенные СПП являются смежными, поскольку значения A_{11} и A_{21} входной переменной x_1 смежные.

Существует три основные причины неполноты БЗ:

- 1) в процессе приобретения знаний персонал, возможно, непреднамеренно оставили пробелы в БЗ, не заметив этого;
- 2) поведение персонала часто основано на эвристических, неполных и неопределенных знаниях, которые переносятся в интеллектуальную систему;
- 3) персонал может сбиться с пути, поскольку БЗ становится все больше и становится неразрешимой.

Оценивание значимости СПП осуществляется на основе длины условий правила. Смешанное продукционное правило, включающее большое количество входных переменных, получает ранг 1, далее 2 и т. д. (чем больше ранг СПП, тем оно полезнее для принятия решений).

Поскольку значимость правила должна находиться в пределах $[0, 1]$, то применяется правило нормирования, тогда в этом случае, чем меньше значение ранга, тем оно более значимое.

Преимущества продукций

Основные достоинства систем продукционных:

- Простота создания и понимания, отдельных правил.
- Простота понимания и модифицирования знаний.
- Простота программной реализации механизма логического вывода.

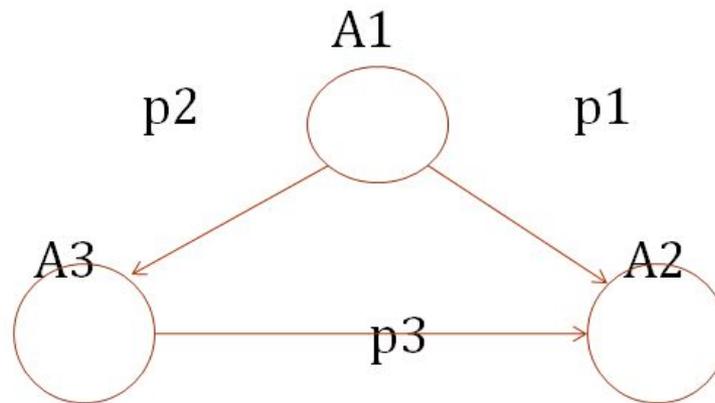
Недостатки:

1. Неясность взаимоотношений правил.
2. Сложность оценки целостного образа знаний.
3. Весьма низкая эффективность обработки.
4. Существенные отличия от человеческих систем знаний.
5. Отсутствие гибкости в логическом выводе.

Семантическая сеть

Семантическая сеть – это модель, в которой структура знаний предметной области формализуется в виде ориентированного графа вершины которого – понятия, а дуги – отношения между ними.

Например, «*программист сел за компьютер и отладил программу*». Объектами являются: программист (A1), компьютер (A2), программа (A3). Объекты связаны отношениями: сел за компьютер (p1), отладил (p2), загружена в компьютер программа (p3).



Самыми распространенными являются следующие типы отношений:

- **быть** элементом класса, то есть объект входит в состав данного класса (ВАЗ 2106 является автомобилем);
- **иметь** свойства, то есть задаются свойства объектов (жираф имеет длинную шею);
- **иметь** значение, то есть задается значение свойств объектов (человек может иметь двух братьев);
- **является** следствием, то есть отражается причинно-следственная связь (астеническое состояние является следствием перенесенного простудного заболевания).

Более детально можно выделить следующие отношения:

- связи типа "часть-целое" ("класс-подкласс", "элемент-множество" и т.п.);
- функциональные связи (определяемые обычно глаголами "производит", "влияет" ...);
- количественные (больше, меньше, равно...);
- пространственные (далеко от, близко от, за, под, над...);
- временные (раньше, позже, в течение...);
- атрибутивные связи (*иметь свойство*, *иметь значение*...);
- логические связи (и, или, не);
- отношения «близости»;
- отношения «сходства-различия»;
- отношения «причина-следствие» и др.

Классификация семантических сетей:

Можно предложить несколько классификаций семантических сетей, связанных с типами отношений между понятиями.

По количеству типов отношений:

Однородные (с единственным типом отношений).

Неоднородные (с различными типами отношений).

По типам отношений:

Бинарные (в которых отношения связывают два объекта).

N-арные (в которых есть специальные отношения, связывающие более двух понятий).

Пример: «Иванов имеет личный автомобиль «Волга» красного цвета с мощностью двигателя 75 л.с.»



Пример. Имеется фраза: «Автомобили Нива и Волга движутся навстречу друг к другу по направлению к городу Томску». Требуется построить фрагмент семантической сети для этой фразы.

Решение. Фрагмент семантической сети для заданной фразы



Фрагмент семантической сети

ПОСТРОЕНИЯ СЕМАНТИЧЕСКОЙ СЕТИ

Для построения СС необходимо выполнить следующие шаги:

1. Определить абстрактные объекты и понятия предметной области, необходимые для решения поставленной задачи. Оформить их в виде вершин.

2. Задать свойства для выделенных вершин, оформив их в виде вершин, связанных с исходными вершинами атрибутивными отношениями.

3. Задать связи между вершинами, используя функциональные, пространственные, количественные, логические, временные, атрибутивные отношения, а также отношения типа «являться наследником» (АКО) и «являться частью» (ISA).

4. Добавить конкретные объекты и понятия, описывающие решаемую задачу. Оформить их в виде вершин, связанных с уже существующими отношениями типа «являться экземпляром», «есть».

5. Проверить правильность установленных отношений (вершины и отношения при правильном построении образуют предложение).

Пример. Построить семантическую модель представления знаний в предметной области «Больница».

Решение.

1. Ключевые понятия данной предметной области – врач, пациенты, болезнь. У пациентов больницы есть общие характеристики, поэтому целесообразно выделить общее абстрактное понятие – человек. Метод постановки диагноза – анализ симптомов заболевания.

Исходя из этого, вершины графа будут следующими: «Врач», «Человек», «Симптом», «Болезнь», «Диагноз».

2. У этих объектов есть определенные свойства и атрибуты. Например, симптоматика включает ряд показателей. Поэтому в граф следует добавить, например, вершины «Пульс тахикардия», «Боль постоянная», «Тошнота», «Напряжение мышц брюшной стенки», «Рвота», «Боль в животе», «Температура 37,2 – 37,6». Одни и те же симптомы сопутствуют различным заболеваниям. Поэтому в граф добавим, например, вершины «Острый аппендицит», «Тромбоз сосудов», «Острая кишечная непроходимость», «Перфорация язвы желудка», «Острый холецистит», «Острый панкреатит», «Острый аднексит».

3. Определим для имеющихся вершин отношения и их типы.

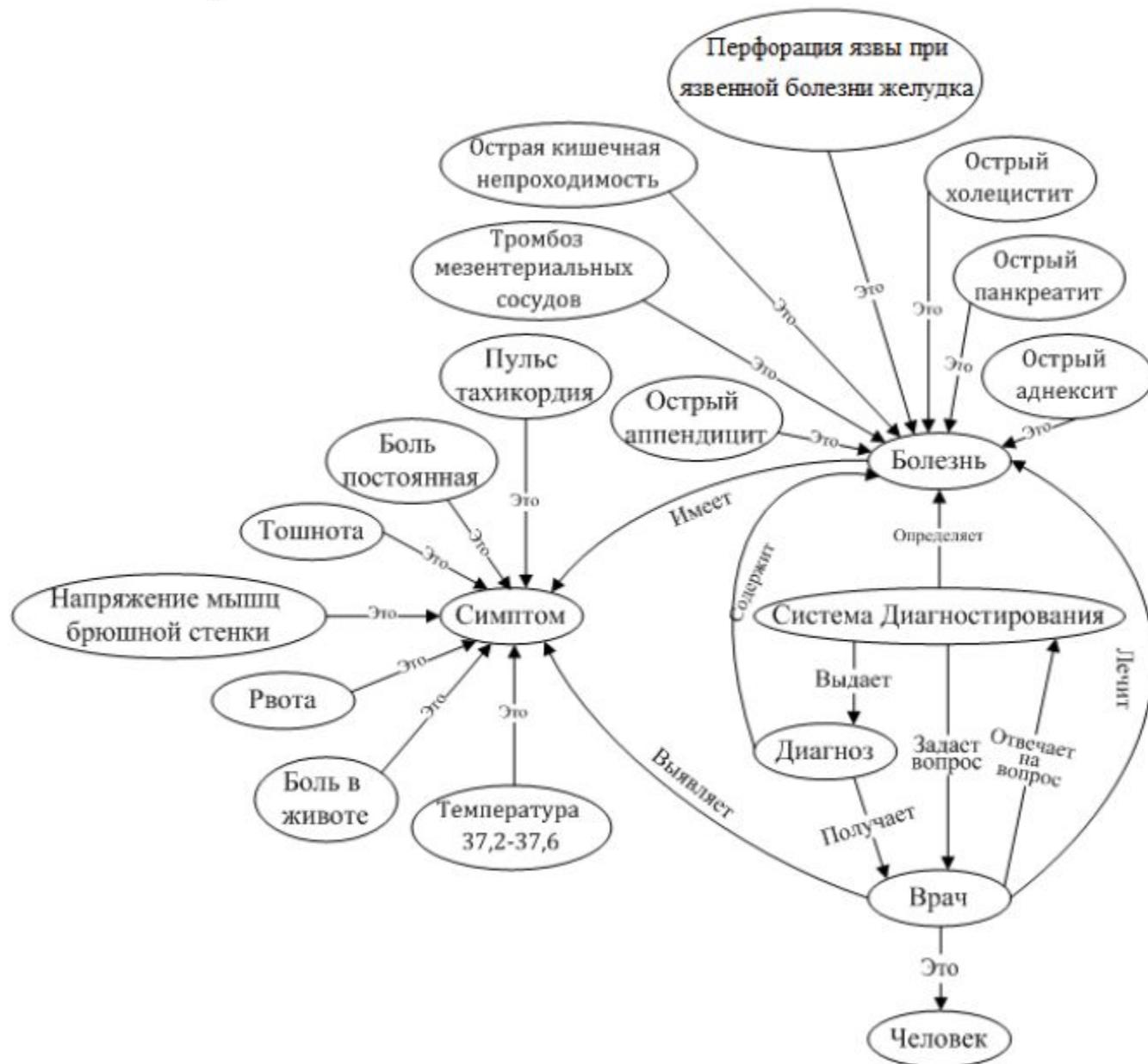
4. Добавим знание о конкретных фактах решаемой задачи. Пусть на прием к врачу гастроэнтерологу пришел пациент с жалобой на боли в животе. Врач проводит осмотр и опрос пациента с целью выявления других симптомов его заболевания. Определив круг возможных заболеваний пациента, врач выдает пациенту рекомендации и, возможно, назначает дополнительные исследования для уточнения диагноза.

Исходя из этого, добавим в граф отношения «Задаёт вопрос», «Отвечает», «Выдает», «Получает». Итоговая семантическая сеть представлена на рис. 10.

5. Осуществим проверку установленных отношений. Например, возьмем вершину «Боль в животе» и пройдем по установленным отношениям. Получаем следующую информацию: боль в животе является частью симптомов, присущих болезни, которая и является диагнозом пациента.

Для получения ответа на вопрос «Пациент болен острым панкреатитом?», необходимо найти соответствующий участок сети и, используя отношения, получить результат. Из запроса понятно, что необходимо найти следующие вершины: «Пациент», «Врач», «Острый панкреатит». Часть СС, находящаяся между этими вершинами, содержит ответ, а именно: «Острый панкреатит» может стать диагнозом заболевания пациента, если выявятся соответствующие симптомы.

Семантическая сеть предметной области "Больница"



Достоинства семантических сетей:

- в семантических сетях существует возможность представлять знания более естественным и структурированным образом, чем в других формализмах.
- семантические сети более других соответствуют современным представлениям об организации долговременной памяти человека.

Недостатки семантических сетей:

- нет специальных средств, позволяющих определить временные зависимости, поэтому временные значения и события трактуются как обычные понятия.
- сложность организации процедуры поиска вывода на семантической сети.