

Основные аспекты технологии создания методов автоматического построения онтологий

Найханова Л.В.

Кафедра систем информатики, Восточно-Сибирский государственный технологический университет, ул. Ключевская, д.40в, г. Улан-Удэ, 670013, Россия.

nlv@esstu.ru, obeke_nlv@mail.ru

***Аннотация.** Данная работа относится к области исследований "Систематизация знаний: Инженерия знаний. Создание онтологий предметных областей: технология создания, применения". Основная идея доклада заключается в применении методологии эволюционных вычислений и технологий автоматного программирования к созданию методов автоматического построения онтологий. К основным результатам работы относится разработка основных компонентов технологии создания методов автоматического построения онтологий, таких как модель генератора систем продукционных правил, модель генератора моделей преобразователя продукционных правил и автоматная модель аппарата активации.*

Ключевые слова: онтология, естественные языки, терминологический словарь, конструктивные знания предметной области, генетический алгоритм, автоматное программирование, генерация систем продукционных правил, генерация моделей преобразователей, аппарат активации продукционных правил.

1 Введение

В последние годы начали развиваться методы автоматического построения онтологий, базирующиеся на лингвистическом анализе текста [2-4,9,10]. Развитие этих методов стало возможным в силу наличия значительного опыта, накопленного в области естественно-языковой обработки текста. Конечно, появление таких методов позволит значительно ускорить процесс создания онтологий. Обычно методы автоматического извлечения знаний из текста разрабатываются в основном как программы традиционного типа, основанные на жестких алгоритмах. В интеллектуальных задачах должна существовать возможность на каждом шаге оценивать ситуацию и предпринимать соответствующие действия. Такую возможность обеспечивают правила. В связи с этим в работе методы лингвистического анализа, включая и методы извлечения знаний, имеют декларативное представление в виде систем продукционных правил. Применение продукционных правил обеспечивает следующие преимущества: простоту и модульность, удобство модификации, ясность, прозрачность, возможность постепенного наращивания, высокую степень общности правил обработки данных. Такой подход к представлению методов определил технологию их создания.

2 Основные аспекты технологии

Для решения задачи, связанной с разработкой компонентов технологии создания декларативных методов автоматического построения онтологий предлагается применять методологии эволюционных вычислений и технологии автоматного программирования.

2.1 Постановка задачи

Пусть заданы:

1) категориальный аппарат онтологии, в виде кортежа глобальных объектов:

$$KA = \langle t, a, s, e, p, q \rangle, \text{ где} \quad (1)$$

t – знак "понятие", a – знак "действие", s – знак "состояние", e – знак "событие", p – знак "свойство", q – знак "величины";

2) конструкция знака "понятие":

$$t = \langle D, P, A, C, S, T, M \rangle, \text{ где} \quad (2)$$

D – множество дефиниций понятия t , P – множество свойств понятия t , A – множество действий понятия t , C – множество понятий (терминов), имеющих количественные отношения с термином t , S – множество состояний понятия t , T – множество понятий, имеющих качественные отношения с термином t , M – множество метазнаков понятия t ;

3) аналогичным образом заданы конструкции знаков оставшихся глобальных объектов;

4) глобальные отношения: качественные QR_1 и количественные QR_2 , каждое из которых включает группы отношений;

5) текст терминологического словаря TD , релевантный модели предметной области M_{SA} , представленной в виде двойки:

$$M_{SA} = \langle T_k, K_k \rangle, \text{ где} \quad (3)$$

T_k – множество понятий; K_k – кортеж, описывающий множество конструкторов и их взаимосвязи.

Необходимо создать методы, предназначенные для распознавания в тексте TD глобальных отношений, глобальных объектов и элементов конструкций их знаков при следующих ограничениях:

1) степень покрытия конструкторов в порожденных правилах множества исходных конструкторов должна стремиться к единице;

2) результаты решения задач построения онтологий с применением порожденных систем продукций и результаты работы экспертов должны быть близки, а в идеальном случае – одинаковы.

2.2 Подход к решению

В работе предлагается решать данную задачу следующим образом. Каждый метод должен формироваться отдельно. Вначале, используя экспертные знания, нужно сформировать модель предметной области прикладной задачи, для решения которой создается метод. Применяя методологии эволюционных вычислений, на основе знаний о модели предметной области генерируются системы продукционных правил как декларативное представление метода. Так как эксперт рассуждает на естественном языке, то продукции будут иметь естественно-языковое представление, поэтому их нужно преобразовать в формальное представление. Для апробации подготовленного метода нужно доказать их достоверность в системе логического вывода.

2.2.1 Формирование модели предметной области

В работе [3] представлена упрощенная модель предметной области, принятая в теории экспертных систем. В решаемой задаче компонент K_k кортежа M_{SA} определяется тройкой:

$$K_k = \langle K_{kc}, K_{kv}, K_{kg} \rangle, \text{ где}$$

K_{kc} – множество конструкторов, K_{kv} – семейство множеств допустимых значений элементов конструкторов, K_{kg} – множество графов, определяющих допустимые взаимосвязи конструкторов.

Рассмотрим формирование модели M_{SA} на примере метода извлечения знаний из терминологических словарей о качественном отношении «Целое–часть». Прежде всего определим понятийное множество T_k . Для рассматриваемого метода

$$T_k = \{p, k, s, r, z, t, l, q, u, v, tr, tx, ty, h, x, c_1, c_2\}, \text{ где}$$

p – предложение, k – композиционное словосочетание (КСС), s – именное субстантивное словосочетание (ИСС), r – семантическое отношение (СемОтношение), z – заголовочный термин, t – термин, l – лексема, q – список, u – элемент списка (ЭлСписка), v – глагол, tr – терм-спутникR, tx – терм-спутникX, ty – терм-спутникY, h – признак, x – характеристика, c_1 – часть речи (ЧастьРечи), c_2 – падеж.

Прежде чем формировать второй компонент K_k модели M_{SA} рассмотрим модель рассуждения эксперта, которая заключается в следующем. В каждом методе есть объект исследования; объект исследования содержит некоторые компоненты; компоненты могут включать в себя

другие компоненты; все компоненты могут иметь характеристики; характеристики являются некоторыми свойствами; свойства имеют некоторые значения. Тогда в зависимости от ситуации, которая определяется значениями характеристик и взаимным расположением компонентов, т.е. их порядком, можно распознать в объекте анализа искомый элемент схемы знака глобального объекта.

Согласно модели рассуждения эксперта бинарные отношения определяются терминами «содержит», «имеет», «есть (является)», «эквивалентен». Тогда множество конструкторов K_{kc} можно задать в виде множества отношений:

$$K_{kc} = \{R_i | i=1 \div n\}, \text{ где } (4)$$

$R_i = \{(x,y) | x, y \in T_k\}$. Для заданной модели конструктивных знаний $n = 4$:

$$\begin{aligned} R_1 &= \{(x,y) | x \text{ содержит } y\}; & R_2 &= \{(x,y) | x \text{ имеет } y\}; \\ R_3 &= \{(x,y) | x \text{ есть } y\}; & R_4 &= \{(x,y) | x \text{ эквивалентен } y\}. \end{aligned}$$

Одно понятие может иметь различные отношения с другими понятиями. Поэтому введем семейство множеств понятий:

$$\Theta = \{T_j | T_j \in T_k, j = 1 \div m, \text{ возможно } T_i \cap T_j \neq \emptyset \text{ при } i \neq j\}.$$

Для метода распознавания семантического отношения «Целое-часть» $m = 10$:

$$\begin{aligned} T_1 &= \{\text{КСС, ИСС, СемОтношение, Термин, Лексема, Список}\}; \\ T_2 &= \{\text{КСС, список}\}; \\ T_3 &= \{\text{ИСС, ЭлементСписка}\}; \\ T_4 &= \{\text{Термин, Лексема, Терм-спутникX, Терм-спутникY}\}; \\ T_5 &= \{\text{Глагол, Терм-спутникR}\}; \\ T_6 &= \{\text{ИСС, Термин, Лексема, ЭлементСписка, Терм-спутникX, Терм-спутникY}\}; \\ T_7 &= \{\text{Падеж, ЧастьРечи}\}; \\ T_8 &= \{\text{ИСС, Термин, Лексема}\}; \\ T_9 &= \{\text{Глагол, Терм-спутникR, Терм-спутникX, Терм-спутникY, Признак, ЧастьРечи, Падеж}\}; \\ T_{10} &= \{\text{Падеж, ЧастьРечи, Индекс}\}. \end{aligned}$$

Перечислим элементы всех отношений $R_i, i = 1, \dots, n$:

$$\begin{aligned} r_{11} &= \text{"x содержит y", где } x \in \{\text{Предложение}\}, y \in T_1; \\ r_{12} &= \text{"x содержит y", где } x \in T_2, y \in T_3; \\ r_{13} &= \text{"x содержит y", где } x \in \{\text{ИСС}\}, y \in T_4; \\ r_{14} &= \text{"x содержит y", где } x \in \{\text{СемОтношение}\}, y \in T_5; \\ r_{21} &= \text{"x имеет y", где } x \in T_6, y \in \{\text{Характеристика}\}; \\ r_{22} &= \text{"x имеет y", где } x \in T_9, y \in \{\text{Значение}\}; \\ r_{23} &= \text{"x имеет y", где } x \in T_1, y \in \{\text{Индекс}\}; \\ r_{31} &= \text{"x есть y", где } x \in \{\text{Характеристика}\}, y \in T_7; \\ r_{32} &= \text{"x есть y", где } x \in \{\text{ЭлементСписка}\}, y \in T_8; \\ r_{41} &= \text{"x эквивалентен x", где } x \in T_{10}. \end{aligned}$$

Таким образом, создается некоторая иерархическая организация, которая хорошо согласуется с теоретическими основаниями когнитивной психологии. Согласно ей при мышлении используются не языковые конструкции как таковые, а их коды в форме некоторых абстракций, которые образуют иерархические структуры.

Пример взаимосвязи конструкторов представлен на рисунке 1. В корне графа всегда находится объект исследования. Вершины первого уровня графа содержат основные компоненты предложения. Верхние дуги графа в основном помечены глаголом «содержит», что показывает иерархическую вложенность понятий. Листья графа содержат константные значения или индекс. Из анализа существующих продукций, разработанных для данного метода, было выявлено, что первый уровень вершин V_1 графа могут составлять множество из одиннадцати альтернативных наборов компонентов: $V_1 = \{(k, r, q|s|l), (s, r, q|s|l), (z, r, q|s|t), (s|z, r, q, h)\}$.

Это множество определяет возможные ветки дерева: k, s, z, r, q, l, t, h . Приведем вариант построения ветки k :

$$\text{branch_k} = \begin{cases} s(tx(\text{value})), & s(z(x(c(\text{value})))) \\ s(x(c(\text{value}))), & s(z(x(c(\text{value})))) \text{, index} \\ s(tx(\text{value}, (x(c(\text{value}))))), & s(z(x(c(\text{value})))) \text{, index} \end{cases}$$

В ветке *branch_k* запись во второй строке соответствует изображению ветки *k* на рисунке 1. Это означает, что вершина *k* включает две ветки с вершинами *s*₁ и *s*₂, заканчивающимися листьями с константным значением, и ветку с вершиной *index*, которая является листом.

Таким образом, порождаемое продукционное правило может быть представлено в виде дерева, сгенерированного на основе конструктов, каждый из которых состоит из двух вершин и помеченной дуги. Допустимое соединение конструктов задается с помощью описаний веток дерева.

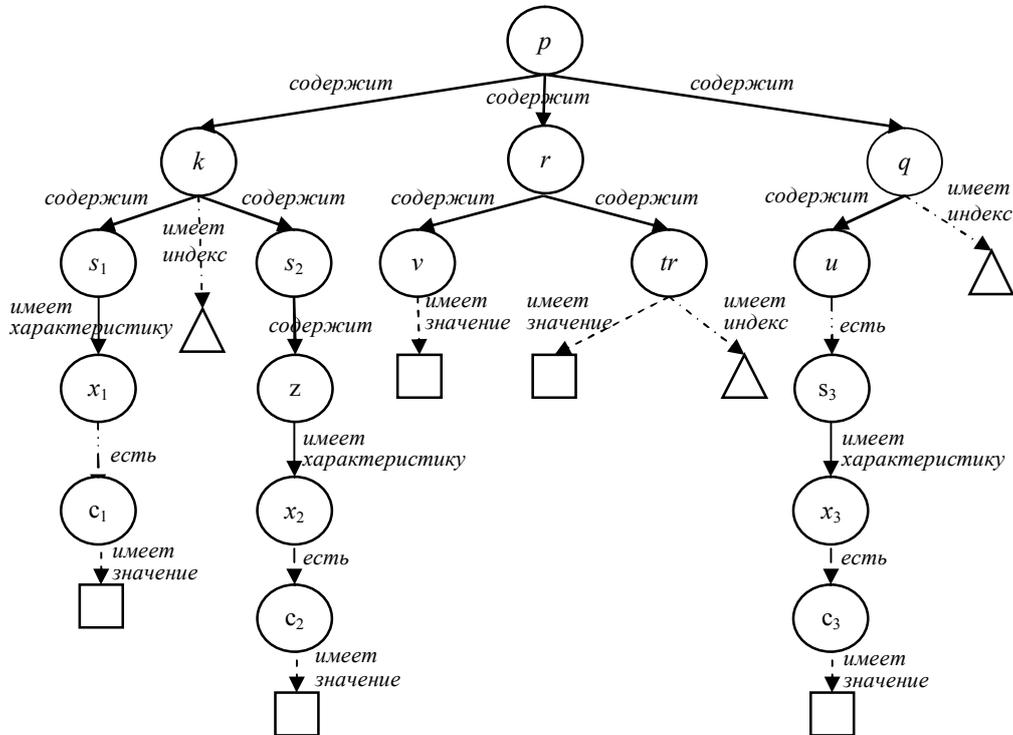


Рис.1. Пример графовой структуры, отражающей взаимосвязь конструктов метода извлечения знаний о семантическом отношении «Целое – часть».

Из вышеизложенного следует, что спецификация метода должна включать множество понятий T_k , используемых в методе; семейство множеств конструктов K_{kc} , определяющих связь между двумя понятиями; множество графов K_{kg} в виде множества альтернативных кортежей, компоненты которых составляют вершины первого уровня графа, и множества возможных вариантов построения веток. Этой информации достаточно для генерации ядер продукционных правил, входящих в систему продукций. В качестве средства описания спецификации метода использован язык XML.

2.2.2 Генерация систем продукций

Для генерации моделей решения задач построения онтологий построена модель генератора систем продукций с применением методологии эволюционных вычислений, а именно генетического алгоритма. Генетический алгоритм (генератор систем продукционных правил) настраивается на прикладную задачу посредством спецификации на XML-языке, которая содержит описание предметной области. Цель генератора систем продукционных правил заключается в построении множества правил R_1, \dots, R_n , таких, что они, являясь базовыми правилами метода, корректно решают прикладную задачу, например, задачу распознавания семантического отношения «Целое-часть».

Для каждого метода в автоматизированном режиме разрабатывается его спецификация, в которой описываются шесть основных элементов:

- объект исследования (*RESEARCH_OBJECT*);
- множество понятий (*SET_VERTEX*);
- множество конструктов (*SET_CONSTRUCTS*);
- альтернативные кортежи вершин первого уровня дерева (*LEVEL_1_TREE_TUPLE*);
- множество вариантов содержимого веток дерева (*SET_BRANCH_TREE*);
- множество значений понятий (*SET_VALUE_VERTEX*).

Объектом исследования *RESEARCH_OBJECT* может быть предложение, словосочетание, лексема или морфема. *SET_VERTEX* соответствует множеству T_k , *SET_CONSTRUCTS* – множеству K_{kc} , *LEVEL_1_TREE_TUPLE* – множеству V_1 , *SET_BRANCH_TREE* – множеству $\{\text{branch}_v | v \in V_1\}$, *SET_VALUE_VERTEX* – множеству значений понятий. Например, для методов распознавания семантических отношений объектом исследования является предложение p .

При разработке генетических алгоритмов использован единый подход. Они созданы по модифицированной схеме генетического поиска, предложенного в работе А.А. Кажарова и А.А. Рокотьянского, созданного на основе анализа работ Холанда, Гольдберга и Дэвиса. При формировании начальной популяции используется стратегия «дробовика», т.е. генерируется случайное множество решений, достаточно большое, но не исчерпывающее всех вариантов, что подходит для задач, в которых невозможен полный перебор, и отсутствует четкая зависимость качества решения от параметров решения. В качестве метода представления генов хромосомы использовано векторное представление, что объясняется структурой хромосомы, которая является достаточно сложной и состоит из отдельных и/или вложенных блоков (ДНК).

Структура хромосомы имеет вид дерева продукции. Молекула ДНК состоит из динамического числа генов и представляется в виде кортежа:

$$\langle \alpha_i \rightarrow \beta_1, lc, \beta_2, lc, \dots, lc, \beta_m \rangle, \quad \text{где}$$

$\alpha_i, \beta_1, \beta_2, \dots, \beta_m, lc$ – гены, $\max(m)=7$; $\alpha_i, \beta_j \in T_k$, T_k – входной алфавит, $|T_k|=k$; символ ‘ \rightarrow ’ обозначает глагол; lc – логическая связка «И» или «ИЛИ».

Например, если $\alpha_i = p, \beta_1 = k, \beta_2 = h$, то в графе молекулы ДНК две смежные ветки, исходящие из одной вершины, связаны между собой логической связкой «И» или «ИЛИ» (рис.2). Если глагол имеет отрицание (унарную логическую связку ‘ \neg ’), то связь обозначается стрелкой ‘ $\xrightarrow{\neg}$ ’.

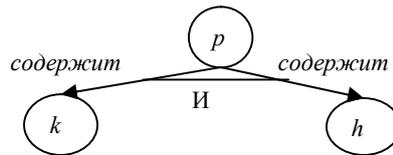


Рис. 2. Пример молекулы ДНК.

При разработке генетических алгоритмов были проведены большие исследования по выбору генетических операторов. Это объясняется тем, что эволюционные вычисления были применены для решения не совсем стандартных задач генетического поиска. Так, например, использованы как классические двух- и трехточечные операторы кроссинговера, так и разработанные операторы кроссинговера, относящиеся к универсальным (арифметический и генно-смешивающий). Из операторов мутации используются простая и модифицированная точечная мутации (α -мутация, β -мутация, A -мутация, $L1$ -мутация, *Branch*-мутация). Для селекции и отбора хромосом используются элитный отбор (Proportional Selection), модифицированный элитный отбор (Tournament Selection) и модифицированный пропорциональный отбор (Roulette Selection). С точки зрения развития генетических операторов в алгоритме применяются ряды генетических операторов. Исследования, проведенные в работе, показали, что комбинация операторов позволяет создавать лучшие решения с меньшими вычислительными затратами, и это подтверждают вычислительные эксперименты.

Особенностью генетических алгоритмов является то, что хромосома интерпретирована в виде дерева решений. Это позволяет использовать в качестве функции пригодности особи (Fitness-функции) модифицированную формулу Дайеса для расчета меры близости графов, основанную на коэффициенте Дайеса. Кроме того, в генетических алгоритмах использованы жесткие ограничения на их параметры, однако их значения можно изменять при настройке алгоритмов на решение конкретной задачи.

Пример сгенерированной хромосомы, полученной на 27 шаге эволюции (размер популяции – 2000 особей), приведен ниже. Графический аналог особи представлен на рисунке 3.

```

Chromosome 1811)
{Chromosome:
<DNA №0|Предложение|p|3 branch(s):[to №1][to №2][to №3]>
<DNA №1|ИСС|s2|1 branch(s):[to №4]>
<DNA №2|СемОтношение|r1|2 branch(s):[to №5][to №7]>
<DNA №3|Лексема|l1|2 branch(s):[to №8][to №11]>
<DNA №4|Индекс|(i1)|0 branch(s):>
  
```

```

<DNA №5|Глагол|v1|1 branch(s):[to №6]>
<DNA №6|Значение|["включает"]|0 branch(s):>
<DNA №7|Индекс|(i1+1)|0 branch(s):>
<DNA №8|Характеристика|x1|1 branch(s):[to №9]>
<DNA №9|ЧастьРечи|cpl|1 branch(s):[to №10]>
<DNA №10|Значение|["сущ"]|0 branch(s):>
<DNA №11|Индекс|(i1+2)|0 branch(s):>
}

```

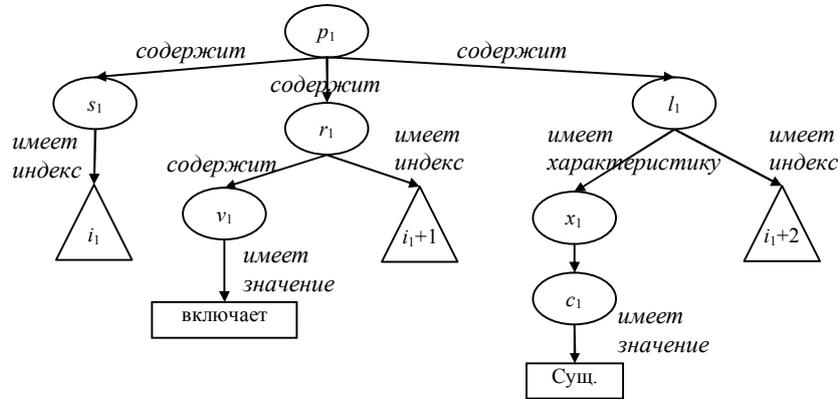


Рис.3. Графическое представление Chromosome 1811.

Первичная оценка достоверности сгенерированной системы продукций выполнена на основе подхода, описанного в работе А.А. Асанова «Генетический алгоритм построения экспертных решающих правил в задаче многокритериальной классификации». Для этого введены коэффициенты абсолютной ошибки E_{abs} и относительной ошибки E_{rel} :

$$E_{abs}(K_{kc}, \tilde{K}_{kc}) = |K_{kc} \setminus \tilde{K}_{kc}|$$

$$E_{rel}(K_{kc}, \tilde{K}_{kc}) = E_{abs}(K_{kc}, \tilde{K}_{kc}) / |K_{kc}|$$

где K_{kc} – множество исходных конструкторов метода, определенных по формуле (4); \tilde{K}_{kc} – множество конструкторов метода в порожденных продукциях.

К настоящему времени оба коэффициента имеют нулевые значения, т.е. в порожденных системах продукций наблюдается полное покрытие исходных конструкторов, что говорит о том, что достоверность порожденных продукций достаточно высока.

2.2.3 Преобразование продукционных правил

В связи с тем, что конструкторы описаны на естественном языке, то и порожденные продукционные правила также представлены на ограниченном подмножестве естественного языка. Поэтому для преобразования продукционных правил в формальный вид разработан конечный преобразователь. Так как для апробации методов требуется система логического вывода, в качестве которой избран линейный резолютивный вывод, то после преобразования продукционных правил в формулы логики предикатов первого порядка необходимо преобразовывать их во множество дизъюнктов. Кроме того, методы извлечения знаний могут быть описаны не только классическими продукциями, но и нечеткими, т.е. возникает задача преобразования нечетких продукций в нечеткие предикаты. Таким образом, имеем три прикладные задачи, для каждой из которых необходимо строить свой преобразователь. Это обусловило применения симбиоза технологий генетического и автоматного программирования при создании преобразователя. Генетический алгоритм (генератор моделей конечных преобразователей, рис.4) порождает требуемую модель преобразователя, а автоматное программирование позволяет получить его код.

Генератор модели автоматического преобразователя должен работать в тесном контакте с инструментальной системой UniMod [8], поддерживающей технологии автоматного программирования. Это объясняется тем, что функция пригодности особей сгенерированной популяции, прошедшей предварительный отбор, вычисляется на основе результатов, полученных после прогонки каждой особи в среде UniMod. Принципиальную схему работы генератора можно представить в виде следующей последовательности действий.

1. Осуществляется настройка на прикладную задачу, для решения которой следует сгенерировать модель преобразователя с помощью генетического алгоритма посредством считывания данных из конфигурационного файла “config.xml”.

2. Создается новая популяция размером, заданным в конфигурационном файле.
3. Выполняется тестирование каждой особи в среде UniMod на тестовых данных. Для прогонки особи в среде UniMod генерируется XML-описание модели автомата. По результатам тестирования вычисляется оценка (*Fitness*-функция) каждой особи популяции (модели преобразователя).
4. Проверяется значение *Fitness*-функции: если оно достигло своего максимального значения, заданного в файле “*config.xml*”, то осуществляется переход на пункт 6, иначе – на следующий пункт.
5. Генерируется новое поколение популяции с помощью генетических операторов мутации и скрещивания, затем управление передается на пункт 3.
6. Формируется исполняемый код преобразователя по лучшей модели *FST*, останов алгоритма.



Рис.4. Обобщенная схема модели генератора.

Симбиоз технологий позволяет оценивать результат преобразования каждой особи с эталонным результатом. Эта возможность позволила не задавать грамматики перевода в явном виде. В итоге было получено три модели преобразователя для перевода продукционных правил с естественно-языкового представления в формулы логики предикатов первого порядка, затем формулы логики предикатов первого порядка – во множество дизъюнктов, и нечеткие предикаты – в нечеткие продукции.

2.2.4 Апробация метода

Для апробации порожденных и преобразованных систем продукций разработана автоматная модель аппарата активации. Принципиально аппарат активации построен как модуль управления продукционными знаниями. Модуль управления разработан в среде Eclipse по технологии автоматного программирования. Автоматная модель модуля управления построена в соответствии с её принципиальной схемой, приведенной на рисунке 5.

Формирование входной ситуации и параметров метода. Все методы могут иметь множество входных ситуаций, каждая из которых представляется множеством дизъюнктов. К параметрам метода относятся наименование метода (аббревиатура метода), сфера применения *K* (например, *МА* – морфологический анализ), данные для проверки условия применимости продукции.



Рис. 5. Принципиальная схема модуля управления.

Выборка системы продукций. Выборка системы продукции происходит по наименованию метода и сфере применимости. Имя продукции состоит из аббревиатуры метода, номера подсистемы и номера продукции в подсистеме.

Выборка дизъюнкта для доказательства. Входная ситуация d_0 задается в виде множества дизъюнктов. Если $|d_0| > 1$, то последовательно доказываются все дизъюнкты.

Линейный резолютивный вывод. Машина логического вывода построена в виде системы линейного резолютивного вывода на основе применения автоматного и генетического программирования. В систему вводится входной дизъюнкт, который доказывается на активной системе продукций и базе фактов. Базы фактов, необходимые для работы текущего метода, выбираются одновременно с системой продукций. Если для всех входных дизъюнктов выведен пустой дизъюнкт, то гипотеза, заложенная в продукционном правиле, считается доказанной.

Поиск и выполнение метода, указанного в постдействии продукции. При успешном доказательстве одной из входных ситуаций выполняется вычислительная процедура. Все процедуры постдействия продукции реализованы на основе применения автоматного программирования в виде методов, записанных в библиотеке методов постдействий. Поиск необходимого метода осуществляется по идентификатору.

Разрешение конфликтного множества результатов. В данной работе не определяется конфликтный набор правил. Входная ситуация доказывается на всех правилах. Это означает, что срабатывают процедуры постдействия всех продукций, для которых при доказательстве получен пустой дизъюнкт. Если для одной ситуации имеют место несколько пустых дизъюнктов, то формируется конфликтное множество результатов работы метода, для разрешения которого предусмотрены либо классические системы продукций, либо нечеткие. В первом случае формируется новое множество входных ситуаций, которые доказываются на специальной системе продукций метода, предназначенной для разрешения конфликтного множества результатов. Во втором случае управление передается системе нечеткого логического вывода.

Нечеткий логический вывод. Система нечеткого логического вывода построена на основе применения методов нечеткого регулирования и автоматного программирования.

Весь описанный процесс повторяется до тех пор, пока не будет закончена естественно-языковая обработка данным методом. Например, при выполнении метода морфологического анализа завершение наступает после обработки всех лексем текста.

2.2.4 Вычислительные эксперименты

Были исследованы несколько словарных статей экономического терминологического словаря Е.Г. Багудиной. Предложения словарной статьи «Цикл»: «ЦИКЛЫ – регулярно повторяющиеся периоды в развитии природы и общества. Циклы могут быть природными, производственными, научно-техническими, экономическими, экологическими». При анализе этой словарной статьи автоматически заполняется знак-фрейм «Цикл». При анализе первого предложения распознается дефиниция и записывается в соответствующий слот, затем во втором предложении распознается семантическое отношение «Род-вид» (рис.6).

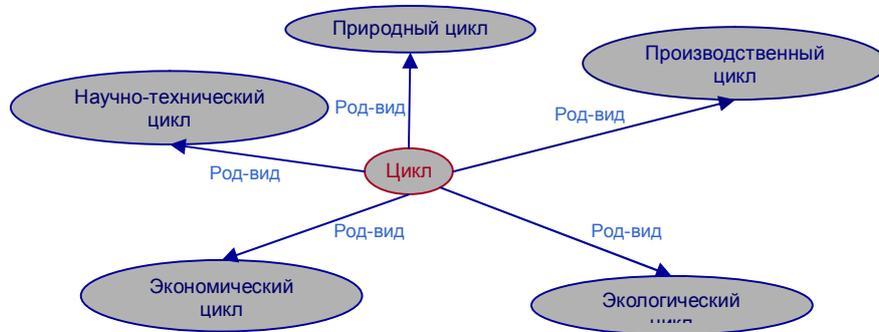


Рис.6. Фрагмент семантической сети понятия «Цикл».

Далее рассматривается вторая словарная статья «Экономические циклы», анализ которой позволяет достроить семантическую сеть понятия «Цикл», рис.7.

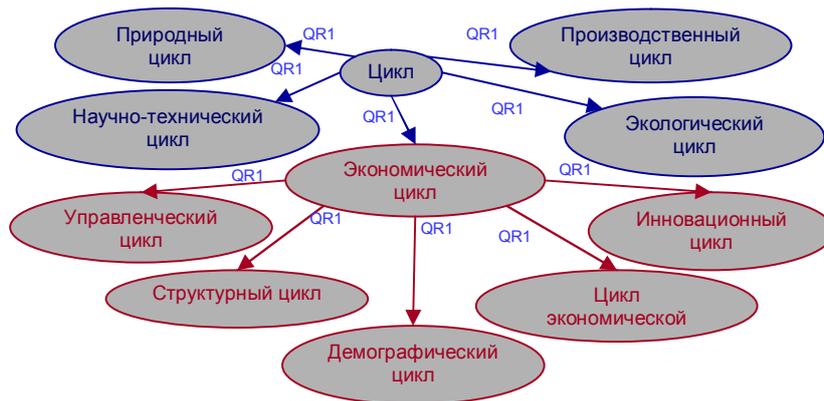


Рис.7. Достроенный фрагмент семантической сети понятия «Цикл».

Предложения словарной статьи «Экономические циклы»: «ЭКОНОМИЧЕСКИЕ ЦИКЛЫ – регулярно повторяющиеся периоды в развитии рыночной экономики. (2) Экономические циклы делятся на циклы экономической активности, демографические циклы, инновационные циклы, структурные циклы, управленческие циклы. (3) Цикл экономической активности состоит из четырех последовательных фаз: кризиса, депрессии, оживления и подъема. (4) Фаза кризиса характеризуется снижением объема производства до минимального уровня, ростом безработицы. (5) Фаза депрессии характеризуется медленными или нулевыми темпами роста производства, присущими застою состоянию экономики. (6) Фаза оживления характеризуется повышением темпов роста, ростом объемов производства, инвестиций и занятости. (7) Для фазы подъема характерен дальнейший рост объемов производства. (8) Все показатели соответственно превосходят максимум, достигнутый в предыдущем цикле».

3 Заключение

Суммируя вышеизложенное, можно сделать вывод о том, что симбиоз методологий эволюционных вычислений и автоматного программирования обеспечивает единую методологию алгоритмизации и программирования, позволяющую строить, читать, проверять, верифицировать алгоритмы и программы. Такой симбиоз используется в случаях, если возможно представить решение в виде модели автомата или преобразователя. Так, эти методологии применены при решении задач линейного резолютивного вывода, преобразования продукционных правил из естественно-языкового представления в формулы предикатов первого порядка, предикатов – во множество дизъюнктов, нечетких продукционных правил из

естественно-языкового представления – в нечеткие предикаты. Методологии эволюционных вычислений требуются при создании только моделей решения задачи, например, при генерации систем продукций. Если модель решения известна, и необходимо разработать программное обеспечение, то следует применять чистые технологии автоматного программирования. Так, например, они применяются при создании аппарата активации продукций и компонентов, входящих в него (нечеткий логический вывод, основанный на методах нечеткого регулирования и процедуры постдействия продукций). В настоящее время исследования компонентов технологии создания методов автоматического построения онтологий продолжаются. Необходимо отметить, что вычислительные эксперименты носят фрагментарный характер, однако уже сейчас можно отметить жизнеспособность разработанных компонентов.

4 Благодарности

Автор благодарен Анатолию Абрамовичу Шалыто, Татьяне Альбертовне Гавриловой и Николаю Григорьевичу Загоруйко за многочисленные полезные советы и поддержку. Автор благодарит своих коллег по кафедре за совместную работу при проведении исследований, а также за разнообразную помощь и поддержку.

Литература

- [1] Богатырев, М.Ю. Применение концептуальных графов в системах поддержки электронных библиотек [Электронный ресурс] / М.Ю. Богатырев, В.Е. Латов, И.А. Столбовская // Электронные библиотеки: перспективные методы и технологии, электронные коллекции - RCDL'2007: труды 9-й Всерос. науч. конф. – Переславль-Залесский, 2007. – Режим доступа: http://rcdl2007.pereslavl.ru/papers/paper_59_v2.pdf.
- [2] Ермаков, А.Е. Автоматизация онтологического инжиниринга в системах извлечения знаний из текста [Текст] / А.Е. Ермаков // Диалог: материалы ежегод. Междунар. конф., Бекасово, 4-8 июня 2008 г. – М.: РГГУ, 2008. – Вып. 7 (14): Компьютерная лингвистика и интеллектуальные технологии.– С. 154-159.
- [3] Загоруйко, Н.Г. Система ONTOGRID для автоматизации процессов построения онтологий предметных областей [Текст] / Н.Г. Загоруйко [и др.] // Автометрия. – , 2005. – Т.41. – № 5. – С. 13-25.
- [4] Загоруйко, Н. Г. Таксономия и распознавание в L-пространствах с использованием концептов [Текст] / Н.Г.Загоруйко, И.А.Борисова, И.Н.Сунина // Обнаружение эмпирических закономерностей. – Новосибирск: [б.и.], 1999. – Вып.166: Вычислительные системы. – С.24-34.
- [5] Кажаров, А.А. Исследование генетических алгоритмов на основе тестирующих функций де Йонга [Текст] / А.А. Кажаров, А.А. Рокотянский // Перспективные информационные технологии и интеллектуальные системы. Таганрог: [б.и.], 2007. – Вып. №4 (32): Интеллектуальные системы.– С. 18-32.
- [6] Найханова, Л.В. Технология создания методов автоматического построения онтологий с применением генетического и автоматного программирования: монография [Текст] /Л.В. Найханова. – Улан-Удэ: Изд-во БНЦ СО РАН, 2008. – 244 с.
- [7] Шалыто, А.А. Автоматно-ориентированное программирование [Текст] / А.А. Шалыто // Фундаментальные исследования в технических университетах: материалы IX Всерос. конф. по проблемам науки и высшей школы. – СПб.: Изд-во Политехн. ун-та, 2005. – С. 44-52.
- [8] Сайт UniMod [Электронный ресурс]. – Режим доступа: <http://unimod.sourceforge.net/>.
- [9] Lynn, S. Automatic Generation of Ontologies from Canonicalized Web Tables [Electronic resource] / S. Lynn, D.W. Embley. – Mode of access: <http://www.deg.byu.edu/papers/mogo.pdf>.
- [10] Völker, J. Automatic Evaluation of Ontologies [Electronic resource] / J. Völker, D. Vrandečić, Y. Sure/ – Mode of access: <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=0E2E84EECAF5209408A1B2210E2C305?doi=10.1.1.59.7177&rep=rep1&type=pdf>.