

УДК 004.8

СОБЧАК А.П., к.т.н., доцент,  
СОЛДАТЕНКО М.О., студентка (Национальный аэрокосмический  
университет им. Н.Е. Жуковского «ХАИ»)

## Применение нечеткой логики при построении искусственного интеллекта

*В статье реализована актуальная научно-прикладная задача - разработан программный метод формирования синтаксической части электронного компилятора и приведена временная диаграмма работы устройства для получения нечеткого логического вывода.*

**Ключевые слова:** искусственный интеллект, нечеткая логика, синтаксический анализатор, мехатронные системы.

### Введение

В современном мире проблема создания искусственного интеллекта (ИИ) поднимается все чаще. Не пройдет и века, как судьбу человечества будет вершить какая-нибудь сложно организованная машина. В целом понятие "искусственный интеллект" весьма расплывчато. Если вкратце высказать общую мысль человечества по созданию ИИ, то это простое копирование человекоподобной линии поведения на искусственно созданном объекте для уменьшения затрат и времени человека.

Рассматривается проблема построения искусственного интеллекта с применением нечеткой логики. На основе анализа факторов, обуславливающих развитие технологий в области изготовления интеллектуальных мехатронных систем формулируется актуальная научно-прикладная задача разработки метода формирования синтаксической части электронного компилятора для получения нечеткого логического вывода.

Таким образом, **целью данной статьи** является описание интеллектуальной системы с нечеткой логикой, в которой необходимая для выполнения управляющая программа создается именно устройством, а не человеком.

### Анализ факторов, обуславливающих тенденции развития искусственного интеллекта

ИИ изначально задумывался как прообраз человека и создание его сопряжено с неизвестностью, а для получения результата ИИ использует только определенную базу известных данных. Ему несвойственен эксперимент.

Ключевым фактором, определяющим сегодня развитие ИИ, считается темп роста вычислительной мощности компьютеров, так как принципы работы человеческой психики по-прежнему остаются неясными (на доступном для моделирования уровне детализации). Но рост производительности современных компьютеров в сочетании с повышением качества алгоритмов периодически делает возможным применение различных научных методов на практике.

Современные разработки, связанные с искусственным интеллектом, неспособны к самокопированию (размножению). На современном этапе развития кибернетики и электроники абсолютно самостоятельное самокопирование роботов невозможно, необходимо хотя бы частичное (часто значительное) вмешательство человека.

Проблемы в разработке ИИ прослеживаются и на уровне формирования образов и образной памяти. Поскольку образы в мышлении человека взаимопроникают друг в друга, то формирование образных цепочек у людей не представляет сложности — оно ассоциативно.

То есть именно проблемы и недостатки ИИ, связанные с нехваткой компьютерных (вычислительной мощности, емкости оперативной и внешней памяти) и людских (наукоемкая разработка интеллектуального ПО требует привлечения ведущих специалистов из разных областей знания и организации долгосрочных исследовательских проектов) ресурсов, подталкивают ученых на их решение, а так же являются основными факторами обуславливающими развитие ИИ.

**Основная часть****Реализация нечеткой логики**

Обработка нечеткой информации и нечеткий вывод давно применяются в различных интеллектуальных системах, однако наиболее широкое распространение нечеткие системы получили в области управления [1].

Общая схема обработки нечеткой информации выглядит следующим образом. Точные исходные данные с датчиков, контролирующих управляющий процесс, переводятся в значения лингвистических переменных в специальном блоке, получившем название “фазификатор”. Далее реализуются процедуры нечеткого вывода на множестве продукционных правил, составляющих базу знаний системы управления, в результате чего формируются выходные лингвистические значения, которые переводятся в точные значения результатов вычислений в блоке “дефазификатор”. На выходе последнего формируются управляющие воздействия, подаваемые на исполнительные механизмы. Эта концептуальная схема лежит в основе так называемого нечеткого контроллера, используемого в интеллектуальных системах обработки неопределенной информации, в частности, в “интеллектуальных” системах управления [2].

Компилятор языка высокого уровня — это программа, которая переводит исходный текст программы в эквивалентную ей объектную программу. Объектная программа формируется на объектном языке, который, как правило, является некоторым машинным языком. Особого внимания заслуживают алгоритмы синтеза МА и МП, поскольку на их основе могут быть разработаны аппаратные компиляторы, обладающие огромными преимуществами по сравнению с программными компиляторами. Залогом их успешной реализации служат достижения в технологии проектирования БИС с высокой степенью интеграции. До настоящего времени в отечественной практике такие компиляторы не применялись. Основным тормозом их развития является теория формальных грамматик, в рамках которой они не могли быть созданы в силу ориентации грамматик на программную реализацию. Для решения проблемы синтеза аппаратных компиляторов требуется создание принципиально нового математического аппарата, в качестве которого можно использовать АРЕКС и Р-СИИА. На базе АРЕКС и Р-СИИА возможна полная формализация основных этапов проектирования аппаратных систем компиляции, функционирующих как в режиме интерпретации, так и в режиме

трансляции с организацией лексической и синтаксической обработки языков, а также генерации объектного кода.

Режим интерпретации означает, что генерируемые компилятором структурные команды выполняются процессором сразу, не дожидаясь окончания процесса компиляции в целом.

При программной реализации интерпретаторы работают медленнее трансляторов, так как в первом случае процессор вынужден простаивать и дожидаться генерации очередной команды. Работу интерпретаторов можно ускорить за счет их аппаратной реализации. При этом быстродействие будет на 2–3 порядка выше, чем у программных интерпретаторов. На основании этого произойдет практически полная компенсация недостатков программных интерпретаторов. Снабдив подобные интерпретаторы средствами отладки, тестирования и распараллеливания программ, получим быстродействующую языковую машину, которая будет работать намного эффективнее, чем современные ЭВМ.

Для аппаратной реализации научных проектов применяются языки описания аппаратуры VHDL и AHDL. Язык AHDL полностью интегрируется с системой автоматизированного проектирования ПЛИС MAX+PLUS II. Файлы описания аппаратуры, написанные на языке AHDL, имеют расширение \*.TDF (Text design file), а файлы на языке VHDL имеют расширение \*.VHD. Для создания VHD- либо TDF-файла можно использовать как текстовый редактор системы MAX+PLUS II, так и любой другой. Проект, выполненный в виде VHD- или TDF-файла, компилируется, отлаживается и используется для формирования файла программирования или загрузки ПЛИС фирмы Altera.

Возьмем одну из частей модели электронного компилятора: синтаксический анализатор. Синтаксический анализатор выполняет работу по расчленению исходной программы на составные части, формированию ее внутреннего представления и занесению информации в таблицу, символов и другие таблицы. В данном случае его работа основана на том, что после знака в математической последовательности могут следовать либо знак, либо закрывающая скобка; после знака – открывающая скобка либо число; после открывающей скобки – только число и после закрывающей скобки может следовать только знак действия.

Временные диаграммы работы синтаксического анализатора представлены на рис. 1.

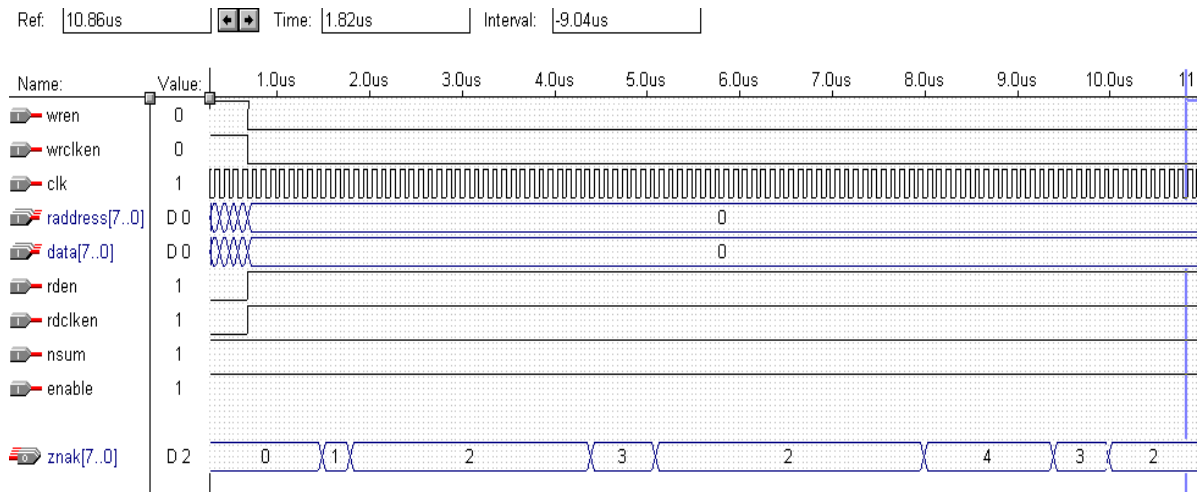


Рис. 1. Временные диаграммы работы синтаксического анализатора:

wren – сигнал разрешения записи в память, wrclken – сигнал разрешения тактирования при записи в память, wraddress[7..0] – адреса ячеек памяти, data[7..0] – записываемые данные, rden – сигнал разрешения чтения из памяти, rdclken – сигнал разрешения тактирования при чтении из памяти, nsum – сигнал разрешения тактирования сумматора, enable – сигнал разрешения счета счетчика, znak[7..0] – сгенерированная строка.

В данном случае в память была записана следующая строка: (1+2)×3. Посмотрев на выходной сигнал znak[7..0], можно сказать, что первой в строке была открывающая скобка(1), затем число(2), арифметической действие(3), число(2), закрывающая скобка(4), арифметическое действие(3) и число(2), то есть, можно сказать, что исходная компилируемая строка записана корректно.

Генератор строки реализован на языке VHDL:

```
entity multiplex1 is
Port
(
    clk : in bit;
    rezult : in integer range 0 to 255;
    nomer : in integer range 0 to 255;
    znak : out integer range 0 to 255
);
end multiplex1;
architecture m of multiplex1 is
begin
process (clk)
    variable znak1 : integer range 0 to 255;
begin
if (clk'event and clk = '0') then
    if rezult = 0 then
        if nomer = 14 then
            znak1 := 1;
        elsif nomer = 15 then
            znak1 := 4;
        elsif nomer < 10 then
            znak1 := 2;
        elsif nomer < 14 and nomer > 9 then
            znak1 := 3;
        end if;
    end if;
end if;
end if;
end if;
end process;
end m;
end multiplex1;
```

На основе нечетких моделей происходит формализация процесса принятия решений. Основной способ описания нечеткого множества заключается в определении степени принадлежности его элементов числом интервала [0,1] с двумя крайними значениями - 1 - "элемент принадлежит множеству"; 0 - "элемент не принадлежит множеству".

В отличие от булевой логики, в теории нечетких множеств используется большое количество логических систем, которые позволяют представить различные варианты человеческой интуиции в приближенных рассуждениях. Именно это обстоятельство представляет самую большую трудность при использовании нечетких множеств, поскольку градации человеческого интеллекта в принятии решений в оценке текущей ситуации варьируются в диапазоне от полного пессимизма до полного оптимизма.

Основные операции над нечеткими высказываниями определяются выбранной логической системой. Операции над нечеткими множествами вычисляются различными способами и зависят от решаемой задачи. Основной формой описания взаимосвязи характеристик объектов являются правила, то есть причинно-следственные нечеткие отношения.

Нечеткое отношение - это “мера” или степень, с которой объекты окружающего мира находятся в данном соотношении друг с другом. Далее рассматривается лишь бинарное нечеткое отношение, связывающее друг с другом две характеристики объектов.

Основной операцией, является операция нечеткого логического вывода. Алгоритмы логического вывода нечетких правил отличаются от классической булевой логики и зависят от множества критериев, трактующих различное использование многозначных логических систем.

Концептуальной основой формализации правил условного логического вывода является правило отделения (*modus ponens*), гласящее:

ЕСЛИ ( $\alpha \rightarrow \beta$ ) истинно И  $\alpha$  истинно ТО  $\beta$  истинно.

Для анализа эффективности правил используются критерии для нечеткого условного логического вывода. Смысл данных критериев заключается в том, что они дают возможность проверить, насколько то или иное правило нечеткого условного вывода удовлетворяет человеческой интуиции при приближенных рассуждениях.

#### **Примеры внедрения нечеткой логики при построении мехатронных систем**

Нечёткая логика является обобщением классической логики и теории множеств. Примером нечеткой логики является автоматика [4]. Слово “автомат” в переводе с греческого означает “самодельствующий”.

Автоматика - отрасль науки и техники, охватывающая теорию и принципы построения систем управления, действующих без непосредственного участия человека; в узком смысле — совокупность методов и технических средств, исключающих участие человека при выполнении операций конкретного процесса.

Развитие автоматки делится на три этапа:

1. Жесткая логика – явление, связанное с реализацией микропрограмм в виде сети элементов "И", "ИЛИ", "НЕ", связанных с регистром памяти RG.

2. Гибкая логика.

3. ПЛИС технологии - это новая элементная база, представляющая собой интегральные схемы, которым свойственна гибкость заказанных больших интегральных схем (БИС) и явление “жесткой логики”.

Сегодня продолжается внедрение логики в прикладные области и программы. Программ глобального масштаба, способных хоть в какой-то мере соответствовать реальному человеку, вести процесс разумного мышления и общения, пока нет и в ближайшем времени не предвидится (слишком много существует преград и неразрешимых проблем). Сегодня компьютер выполняет только точные указания, которые ему даст человек. При написании

любого приложения программист пользуется языком высокого уровня, затем программа-транслятор переводит это приложение на машинный язык директив, который и понимает процессор компьютера [5].

Тенденции развития проявятся в будущем: будут разработаны нанороботы, которые способны ремонтировать клетки. Снабжённые полным описанием человеческого тела с точностью до атома они смогут вернуть даже очень старого человека в то состояние, в котором он был в молодости.

От операций на органах мы перейдём к операциям на молекулах и станем практически бессмертными. Типичное медицинское наноустройство будет представлять собой робота микронного (мкм) размера, собранного из наночастей. Эти части будут варьироваться от 1 до 100 нм (1 нм = 10<sup>-9</sup> м), и будут должны составлять работоспособную машину, размерами около 0.5-3 мкм (1 мкм = 10<sup>-6</sup> м) в диаметре. Три микрона – максимальный размер для медицинских нанороботов кровотока, т.к. это минимальный размер капилляров.

Невозможно сказать сейчас, как будет выглядеть универсальный наноробот, но уже определена основная цель: создать искусственный интеллект в человеческом облики для того чтобы робот мог сам принимать решения в той или иной ситуации.

---

#### **Выводы**

В результате в данной статье рассмотрено построение искусственного интеллекта с применением нечеткой логики, на основе программной реализации устройства - синтаксической части электронного компилятора, а также переход от теории к аппаратной реализации объекта; приведены примеры внедрения нечеткой логики при построении искусственного интеллекта и представлены дальнейшие действия и перспективы на будущее, которое ученые пытаются приоткрыть уже сегодня.

Применение разработанного принципиально нового поколения модуля, в котором осуществлена интеграция всех трех компонент – электромеханической, электронной и компьютерной дает возможность облегчить решение и анализ сложных интеллектуальных задач, стабильно снизить стоимость аппаратных средств для предприятия и ускорить процесс производства.

Литература

1. Собчак А.П., Марченко А.М. Разработка цифровых устройств на базе программируемых логических интегральных схем - Інформаційно-керуючі системи на залізничному транспорті.-2002.-№6;
  2. Собчак А.П., Марченко А.М. Пліс-технології в системах управління ШПП - Збірник наукових праць “Удосконалення систем та комплексів електричної тяги”,– Харків: ХарДАЗТ. - 2003. – Вип.55. –С.40-46.;
  3. <http://www.nestor.minsk.by/kg/2005/20/kg52017.html>;
  4. Смирнов Н.Д. Аппаратная реализация устройства сжатия управляющей информации на программируемых логических интегральных схемах - Радіоелектронні і комп’ютерні системи, - 2005.-№3(11).- С.34-39;
  5. Ходарев К.В. Модель аппаратной реализации алгоритма Рутисхаузера - Радіоелектронні і комп’ютерні системи, -2005.-№4(12).- С.96-102.
- Собчак А.П., Солдатенко М.О. Застосування нечіткої логіки при побудові штучного інтелекту.** У статті реалізована актуальна науково-прикладна задача - розроблено програмний метод формування синтаксичної частини електронного компілятора і наведена тимчасова діаграма роботи пристрою для отримання нечіткого логічного виводу.  
**Ключові слова:** штучний інтелект, нечітка логіка, синтаксичний аналізатор, мехатронні системи.
- 
- Sobchak A.P., Soldatenko M.O. Application of fuzzy logic in constructing of artificial intellect.** An urgent scientific and applied problem has been implemented in the article – namely a software approach to the formation of syntax part of electronic compiler has been developed and a timing chart of a device operation to obtain fuzzy inference has been presented.  
**Key words:** artificial intelligence, fuzzy logic, parser, mechatronic systems.

Рецензент д.т.н., професор Латкин М.А. (НАУ «ХАІ»)

*Поступила 15.05.2013 г.*