

УДК 681.3.06

Як ТЕПАНДИ

## ТЕСТИРОВАНИЕ ПРОДУКЦИОННЫХ СИСТЕМ ПО КРИТЕРИЮ ПОКРЫТИЯ ОПЕРАТОРОВ

(Представил Б. Тамм)

Экспертные системы (ЭС) стали широко доступными и используются в областях, где необходима повышенная надежность [1–3]. В связи с этим и повысились требования к качеству ЭС. Их качество зависит, во-первых, от уровня разработки методов и средств извлечения знаний, во-вторых, от качества инструментальных ЭС, и в-третьих, от развития методов и средств проверки, тестирования и оценивания ЭС. Методы и средства извлечения знаний, а также инструментальные ЭС усиленно разрабатываются. В то же время методы проверки, оценивания и тестирования ЭС изучены еще недостаточно и разработчики не могут гарантировать высокое качество ЭС [4–6]. Поэтому тестирование ЭС представляется актуальной задачей, при решении которой, однако, возникают проблемы.

В настоящей работе рассматриваются возможности тестирования ЭС на основе критерия покрытия операторов [7].

**Статический анализ ЭС.** Прежде чем приступать к тестированию ЭС, полезно проверить соответствие ее базы знаний (БЗ) некоторым формальным критериям. В частности, для этого разработаны методы проверки, с точки зрения того, чтобы в БЗ не было циклов, лишних правил, отсутствующих правил, противоречивых правил, включающих правил [8–10]. ЭС может предлагать ввод дополнительных условий в посылку правила, основываясь на сходстве между правилами [11].

Достоинство этих методов заключается в том, что они хорошо формализуются и поэтому сравнительно легко могут быть реализованы в виде программ. Кроме того, они не нуждаются в оценке эксперта. В то же время эти методы не обеспечивают достаточной проверки БЗ и некоторые из них, возможно, не будут использоваться в следующих поколениях ЭС. Например, БЗ может быть непротиворечивой, но все же включать содержательные ошибки. Тем не менее человек успешно работает, используя противоречивые знания. Такие же возможности должны предоставлять и ЭС.

**Тестирование на основе текста БЗ.** В традиционном программировании для отбора тестов широко используются критерии покрытия операторов и покрытия решений [7]. Мы можем переформулировать их для ЭС следующим образом [12]. Каждое предложение БЗ, отличное от правил, должно использоваться хотя бы один раз. По критерию покрытия операторов каждое правило должно быть активизировано по меньшей мере один раз, причем все условия этого правила должны быть удовлетворены. По критерию покрытия решений каждое правило должно быть активизировано как с истинной, так и с ложной посылкой.

Критерии комбинаторного покрытия решений и покрытия всех путей в программе [7] в большинстве случаев очень трудоемкие как для обычных программ, так и для ЭС.

Рассмотрим теперь генерацию тестов на основе текста БЗ, заданной в виде продукционной системы [13] специального вида, весьма распространенного в ЭС [12]. Введем для этого следующие определения.

**Определение.** *Продукционная система (ПС) — это пятерка  $P = (F, R, G, C, I)$ ,*

*где  $F$  — база данных, включающая факты о решаемой проблеме (предполагается, что существует конечное число разных фактов);*

*$R$  — база правил, включающая правила вида:*

*ЕСЛИ  $\{f_1$  и  $\{f_2$  и ... и  $\{f_K$ , ТО  $\{$ ,*

*где  $\{f_1, \dots, \{f_K, \{$  — факты. Мы будем говорить, что  $\{f_1, \dots, \{f_K$  — условия, а  $\{$  — следствие указанного правила. Будем также говорить, что указанное правило ведет к  $\{$ ;*

*$G$  — множество целей;*

*$C$  — множество ограничений, которым должны соответствовать факты в базе данных (в дальнейшем предполагается, что  $C$ , как минимум, налагает запрет на одновременное присутствие в базе данных факта и его отрицания);*

*$I$  — интерпретатор правил, реализующий процесс вывода (предполагается, что интерпретатор ведет поиск вывода от цели).*

**Определения.** Пусть  $P = (F, R, G, C, I)$  — продукционная система.

1. Система  $P$  называется **статически корректной**, если методы статического анализа уже применены, т. е. в  $P$  нет циклов, лишних правил, отсутствующих правил, противоречивых правил, включающих правил.

2. **И/ИЛИ-граф продукционной системы  $P$**  — это граф, образованный на основе правил из  $R$ . Если  $P$  не включает циклов, то мы будем говорить об И/ИЛИ-дереве.

3. **Правило  $r_2$  следует за правилом  $r_1$** , если интерпретатор при поиске вывода всегда рассматривает в первую очередь правило  $r_1$ , а затем —  $r_2$ .

4. **Ширина правила** — это число фактов, входящих в его условие.

**Определения.** Пусть  $P = (F, R, G, C, I)$  — продукционная система без циклов, а  $T$  — ее И/ИЛИ-дерево.

1. **Высотой правила ЕСЛИ ... ТО  $\{$**  называется высота вершины  $\{$  в И/ИЛИ-дереве ПС  $P$ .

2. **Формула, соответствующая факту  $\{$**  — это дизъюнкция формул, соответствующих ведущим к  $\{$  правилам. Если таких правил нет, то  $\{$  соответствует формула ( $\{$ ).

3. **Формула, соответствующая правилу  $r$**  — это конъюнкция формул, соответствующих условиям правила  $r$ .

4.  $P$  — ПС с **разделенными условиями**, если условия любых двух правил из  $R$  не включают одинаковых фактов.

Методы генерации тестов должны быть эффективными, чтобы применять их на практике. Здесь возникают, по меньшей мере, две проблемы: определение сложности процесса генерации тестов и определение максимального числа тестов. Так как один и тот же набор тестов может использоваться многократно, а выполнение теста и предсказание правильных результатов — процессы трудоемкие, то вторая задача представляется более важной. Ниже мы проанализируем некоторые алгоритмы исходя из указанных проблем.

В статически корректной системе существует тривиальный метод генерации тестовых данных, удовлетворяющих критерию покрытия опе-

раторов (при условии, что эти данные существуют), а именно — достаточно генерировать всевозможные комбинации фактов в базе данных. Тогда при сделанных выше предложениях, каждое правило в БЗ будет активизироваться с истинной предпосылкой по крайней мере один раз. Количество сгенерированных таким образом тестов растет экспоненциально от числа возможных фактов. Хотя время генерации одного теста здесь постоянное, суммарное время генерации тестов растет также экспоненциально от числа фактов.

Ниже приводится более эффективный алгоритм.

#### А л г о р и т м Б.

1. Пусть Тестируемые-Правила и Нетестируемые-Правила — пустые множества.

2. Для каждой цели  $g$  повторять шаги 2.1—2.3.

2.1. Запустить интерпретатор  $I$ , чтобы достичь цели  $g$ .

2.2. Добавить использованные на предыдущем шаге правила к множеству Тестируемые-Правила.

2.3. Отпечатать полученный тестовой пример, цель  $g$  и использованные правила.

3. Пока  $R \langle \rangle$  Тестируемые-Правила  $\cup$  Нетестируемые-Правила, повторять шаги 3.1—3.5.

3.1. Выбрать правило  $r$  из множества  $R \setminus (\text{Тестируемые-Правила} \cup \text{Нетестируемые-Правила})$ .

3.2. Построить доказательство  $Pr$  некоторой цели  $g$  для интерпретатора  $I$ , включающее выбранное правило. При этом факты, необходимые для построения  $Pr$ , должны удовлетворять ограничениям  $S$ . Если удалось найти такое доказательство, то идти к 3.4.

3.3. Если нужное доказательство и нужная цель не существуют, то выбранное правило не может быть протестировано по критерию покрытия операторов. Добавить  $r$  к множеству Нетестируемые-Правила. Идти к 3.

3.4. Добавить использованные в доказательстве  $Pr$  правила к множеству Тестируемые-Правила.

3.5. Отпечатать полученный тестовой пример, цель  $g$  и использованные правила.

4. Если множество Нетестируемые-Правила непусто, то напечатать его.

*Предложение 1. Алгоритм Б распознает существование набора тестовых примеров для тестирования  $P$  по критерию покрытия операторов. Если такой набор существует, то алгоритм Б генерирует его, причем число примеров в наборе не превышает числа правил в  $R$ .*

При сделанных предположениях приведенный алгоритм гарантирует достижение всех целей и активизирование каждого тестируемого правила с истинной предпосылкой. На самом деле, первая часть алгоритма — достижение целей — включена лишь по соображениям эффективности реализации. Каждый тест включает, по крайней мере, одно новое правило, поэтому число тестов не превышает числа правил в БЗ. В реальных ситуациях число тестов меньше, так как многие тесты включают больше чем одно новое правило. В то же время число тестов, сгенерированных приведенным алгоритмом, не обязательно должно быть минимальным.

Исследуем теперь сложность проблемы генерации тестов.

*Предложение 2. Пусть  $P = (F, R, G, S, I)$  — продукционная система. Следующие задачи NP-полны:*

1. Существует ли тест для данного правила из  $R$ ?

2. Пусть  $P$  — статически корректная система. Сгенерировать тест для данного правила из  $R$ .

3. Сгенерировать набор тестовых данных, удовлетворяющий критерию покрытия операторов для данной (статически корректной) ПС.

Предложение 2 показывает, что в общем случае маловероятно существование эффективных по времени алгоритмов для генерации набора тестов, удовлетворяющего критерию покрытия операторов. Следовательно, желательно найти ограниченные классы продукционных систем, которые можно было бы практически использовать, эффективно распознать и для которых существовал бы эффективный алгоритм генерации тестов.

Сначала узнаем, существует ли возможность ограничивать сложность правил таким образом, чтобы генерация тестов оказалась простой. Следующее предложение показывает, что для правил весьма простого вида такое ограничение существует, но достаточно уже небольших ослаблений, чтобы генерация тестов стала экспоненциальной.

Предложение 3. Если высота И/ИЛИ-дерева продукционной системы  $P$  равна 2, то задача генерации тестового набора, удовлетворяющего критерию покрытия операторов, требует полиномиального времени. Та же задача NP-полна для ПС, включающих правила максимальной ширины 3, а среди них и хотя бы одно правило высоты 4.

Следующее полезное ограничение ПС дается в предложении 4.

Предложение 4. Пусть  $P$  — продукционная система с разделенными условиями. Тогда можно генерировать набор тестов, соответствующий критерию покрытия операторов, за полиномиальное время.

**Заключение.** Необходимость в тестировании и оценке ЭС в настоящее время начинает только ощущаться. Поэтому соответствующие методы и средства разработаны еще недостаточно. Большинство методов тестирования и оценки БЗ находится пока еще на неформальном уровне. В то же время существует богатый опыт тестирования и оценки обычных программ. Ввиду многих отличий последних от ЭС, этот опыт нельзя использовать непосредственно — требуется большая работа, чтобы приспособить имеющиеся методы к ЭС. В настоящей статье сделаны некоторые шаги в этом направлении.

Автор благодарен руководству ИТК САН г. Братиславы и Базовой лаборатории ИТК САН за предоставление вычислительных ресурсов и за создание благоприятных условий работы.

#### ЛИТЕРАТУРА

1. Buchanan, B. G. // Expert Systems, 1986, 3, 32—51.
2. Anderson, D., Ortis, C. // IEEE Expert, 1987, 2, 71—79.
3. Schindler, M. // Electronic Design, 1984, 32, 106—146.
4. Построение экспертных систем (Под ред. Ф. Хейеса-Рота, Д. Уотермана. Д. Лелана). М., Мир, 1987.
5. Ramamoorthy, C. V., Shekar, S., Garg, V. // Computer, 1987, 20, 30—37.
6. O'Keefe, R. M., Balci, O., Smith, E. P. // IEEE Expert, 1987, 2, 81—90.
7. Майерс, Г. Искусство тестирования программ. М., Мир, 1982.
8. Nguyen, T. A. // Proc. 3rd Conf. Artif. Intell. Appl., Kissimee, Fla, Febr. 23—27, 1987. Washington, D. C., 1987, 4—8.
9. Cragun, B. J., Stendel, H. J. // Int. J. Man—Mach. Stud., 1987, 26, 633—648.
10. Suwa, M., Scott, A. C., Shortliffe, E. H. // Rule-Based Expert Systems (eds. B. G. Buchanan, E. H. Shortliffe). London, Addison-Wesley, 1984, 159—170.
11. Davis, R., Lenat, D. B. Knowledge-Based Systems in Artificial Intelligence. New York, Mebraw-Hill, 1982.
12. Tepandi, J. // Proc. Int. Conf. Algorithms'89, 1989, 310—312.
13. Pospelov, D. // Artif. Intell. and Inf.-Contr. Syst. Rob., Proc. 4th Int. Conf. Smolenice, Amsterdam, 1987, 93—97.

## PRODUKTSIOONISÜSTEEMI TESTIMINE VASTAVALT LAUSEADEKVAATSUSE KRITERIUMILE

Ekspertsüsteeme kasutatakse vastutusrikaste ülesannete lahendamisel, seepärast peab nende töökindlus olema piisav. Üks töökindluse parendamise vahendeid on testimine. Artiklis on uuritud produktsioonisüsteemide testimist (produktsioonisüsteem on ekspertsüsteemi liik) vastavalt lauseadekvaatsuse kriteeriumile. Viimane nõuab üldjuhul, et programmi iga lause oleks testimise käigus vähemalt korra töötanud. Kriteerium on ümber sõnastatud produktsioonisüsteemide tarvis ning pakutud testandmete genereerimise põhimõtted ja algoritmid. On näidatud, et üldjuhul osutub testandmete genereerimise ülesanne NP-täielikuks. Eraldi on uuritud kahte produktsioonisüsteemide klassi, mille puhul see ülesanne on polünomiaalse keerukusega.

Jaak TEPANDI

## TESTING A PRODUCTION SYSTEM ACCORDING TO THE STATEMENT ADEQUACY CRITERION

Principles and methods for testing a production system according to the statement adequacy criterion are proposed. The complexity of the algorithms is characterized, two classes of production systems with polynomial test generation complexity are investigated.