УДК 004.7, 681.55

О. В. Герман, кандидат технических наук, доцент (БГТУ);
О. И. Садовская, преподаватель (ГрГУ)

МЕТОД СИНТЕЗА ПОВЕДЕНИЯ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ

В статье рассмотрена формальная задача синтеза поведения интеллектуальной системы (например, робота) на основе системы логических формул с временным параметром, описывающих знания о системе. Излагается оригинальный алгоритм синтеза, основанный на использовании метода подстановок для переменных состояния системы. С формальной точки зрения суть алгоритма сводится к многократному решению систем булевских уравнений, соответствующих последовательным моментам времени $t = 0, 1, 2 ..., z$. Алгоритм ориентирован на конкретное значение z. Приводится сравнительная оценка с известными методами.

The paper considers a formal specification of a problem connected to intelligent system behavior synthesis (for example, that one represented by a robot). The procedure uses a system of logical formulas with time parameter defining the knowledge base. The original synthesis algorithm is presented based on the consequent substitutions for the system state variables. From a formal viewpoint the essence of the algorithm consists in multiple solution of the systems of Boolean equations representing its behavior at the steps $t = 0, 1, 2 ..., z$. The value of z is restricted. A comparative estimation of the suggested algorithm is given.

Введение. Под интеллектуальной системой понимаем систему, поведение которой строится на основе базы знаний. Будем рассматривать продукционную базу знаний, содержащую продукции вида

$$S_i \,\&\, u_j \to S_{t+1},$$

где S_i – текущее состояние системы; S_{t+1} – следующее состояние системы, u_j – применяемое управление. Для описания состояния системы можно использовать конечные векторы переменных состояния, причем мы рассматриваем здесь булевы векторы с двоичными переменными.

В качестве классического примера интеллектуальной задачи (или теста на интеллектуальность) часто используют известную задачу об обезьяне и банане, которая формулируется так.

Обезьяна находится в точке A, ящик – в точке B, банан – в точке C. Требуется определить порядок действий, которые обезьяна должна выполнить, чтобы получить банан. Введем следующие переменные состояния: m_A, m_C, m_B – двоичные переменные, определяющие местонахождение обезьяны; b_A, b_C, b_B – двоичные переменные, определяющие местонахождение ящика; on, cap – двоичные переменные, определяющие находится ли обезьяна на ящике ($on = 1$), вне ящика ($on = 0$), а также схвачен банан или нет (cap). Таким образом, состояние системы можно описать с помощью восьми булевских переменных. На самом деле достаточно шести переменных, так как две из приведенных булевых переменных зависимы.

Итак, исходное состояние системы можно записать так:

$$S_0 = < m_A = 1,\ b_B = 1,\ on = 0,\ cap = 0 > .$$

Задается также целевое (конечное) состояние вида:

$$S_{fin} = < cap = 1 > .$$

Заметим, что значения некоторых переменных в векторах состояний нам безразличны.

Теперь надо перейти к рассмотрению знаний. Для представления знаний, кроме переменных состояний, можно ввести управления (хотя это не обязательно, поскольку, зная два смежных во времени состояния, легко узнать, какое управление было реализовано). Такими управлениями будут:

u_A, u_C, u_B – перейти в точку A (C, B соответственно);

p_A, p_C, p_B – передвинуть ящик в точку A (C, B соответственно);

u – забраться на ящик;

g – схватить банан.

Итак, нам надо представить знания о поведении системы.

Знания. Для идентификации состояний нам необходимы правила (функции) для определения реакции.

1. Если обезьяна без банана и рядом с ящиком, то надо взобраться на ящик:

$$\overline{on}(t) \cdot m_C(t) \cdot b_C(t) \to on(t+1),$$
$$(\bar{m}_C(t) \vee \bar{b}_C(t)) \to \overline{on}(t+1),$$
$$on(t) \to on(t+1).$$

2. Если обезьяна на ящике, но банан не схвачен, то надо схватить банан:

$$on(t) \cdot \overline{cap(t)} \to cap(t+1),$$
$$\overline{on}(t) \to \overline{cap}(t+1).$$

3. Если обезьяна вне ящика, то нужно идти к ящику:

$$m_A(t)\cdot b_B(t)\to m_B(t+1),$$
$$m_C(t)\to \overline{m}_B(t+1),$$
$$m_B(t)\cdot b_B(t)\to \overline{m}_B(t+1).$$

4. Если обезьяна у ящика, но банан в другом месте, то нужно тащить ящик к банану:

$$m_B(t)\cdot b_B(t)\to m_C(t+1)\cdot b_C(t+1),$$
$$\overline{m}_B(t)\vee \overline{b}_B(t)\to \overline{m_C(t+1)\cdot b_C(t+1)}.$$

5. Наконец, нужны вспомогательные условия, что обезьяна может быть только в одном месте, как и ящик:

$$\overline{m}_A(t)\vee \overline{m}_B(t),$$
$$\overline{m}_A(t)\vee \overline{m}_C(t),$$
$$\overline{m}_B(t)\vee \overline{m}_C(t),$$
$$m_A(t)\vee m_B(t)\vee m_C(t),$$
$$b_B(t)\vee b_C(t),$$
$$\overline{b}_B(t)\vee \overline{b}_C(t),$$
$$b_B(t)\to \overline{on}(t).$$

6. Имеется еще целый ряд условий, который следует «схватить», но в интересах краткости мы их не рассматриваем. Об этих условиях будет сказано ниже в процессе рассмотрения работы метода.

Модель. Приведенные выше знания нужно трансформировать в систему логических дизъюнктов и получить подстановки для переменных, с помощью которых можно найти решение задачи. Прежде всего, получим эквивалентную систему дизъюнктов:

$$on(t)\vee \overline{m}_C(t)\vee \overline{b}_C(t)\vee on(t+1),$$
$$m_C(t)\vee \overline{on}(t+1),$$
$$b_C(t)\vee \overline{on}(t+1),$$
$$\overline{on}(t)\vee on(t+1),$$
$$\overline{on(t)}\vee cap(t)\vee cap(t+1),$$
$$on(t)\vee \overline{cap}(t+1),$$
$$\overline{m}_A\vee \overline{b}_B(t)\vee m_B(t+1),$$
$$\overline{m}_C(t)\vee \overline{m}_B(t+1),$$
$$\overline{m}_B(t)\vee \overline{b}_B(t)\vee \overline{m}_B(t+1),$$
$$\overline{m}_B(t)\vee \overline{b}_B(t)\vee m_C(t+1),$$
$$\overline{m}_B(t)\vee \overline{b}_B(t)\vee b_C(t+1),$$
$$m_B(t)\vee \overline{m}_C(t+1)\vee \overline{b}_C(t+1),$$
$$b_B(t)\vee \overline{m}_C(t+1)\vee \overline{b}_C(t+1),$$
$$\overline{m}_B(t)\vee \overline{b}_B(t)\vee b_C(t+1),$$
$$\overline{m}_A(t)\vee \overline{m}_B(t),$$
$$\overline{m}_A(t)\vee \overline{m}_C(t),$$
$$\overline{m}_B(t)\vee \overline{m}_C(t),$$
$$m_A(t)\vee m_B(t)\vee m_C(t),$$
$$b_B(t)\vee b_C(t),$$
$$\overline{b}_B(t)\vee \overline{b}_C(t),$$
$$\overline{b}_B(t)\vee \overline{on}(t).$$

Далее мы будем использовать метод подстановок, описанный в [1]. Суть этого метода заключается в получении выражений для переменных с временным параметром $(t + 1)$ через другие переменные. Согласно [1, 2] выразим, например,

$$on(t+1)=m_C(t)\cdot b_C(t)\,. \tag{1}$$

Правило получения этой подстановки таково: выпишем дизъюнкты, содержащие $\overline{on}(t+1)$:

$$\begin{aligned}&m_C(t)\vee \overline{on}(t+1),\\&b_C(t)\vee \overline{on}(t+1).\end{aligned} \tag{2}$$

Подстановка (1) получается непосредственно из (2) по схеме:

$$\begin{aligned}&\overline{\alpha}\vee F_1,\\&\overline{\alpha}\vee F_2,\\&\ldots\ldots\ldots\ldots\\&\overline{\alpha}\vee F_z,\end{aligned} \tag{3}$$

$$\alpha\equiv F_1\cdot F_2\cdot\ldots\cdot F_z.$$

Эта подстановка, как доказано в [1], сохраняет как минимум одно решение, если оно есть. Однако она может терять некоторые решения, что не позволяет использовать ее в общем случае для задачи синтеза управления. В связи с этим предлагается модифицировать подстановку (3) к виду

$$\begin{aligned}&\overline{\alpha}\vee F_1,\\&\overline{\alpha}\vee F_2,\\&\ldots\ldots\ldots\ldots\\&\overline{\alpha}\vee F_z,\end{aligned} \tag{4}$$

$$\alpha\equiv F_1\cdot F_2\cdot\ldots\cdot F_z\cdot\varepsilon,$$

где добавлена в качестве множителя новая формула ε. Можно доказать, что подстановка (4) не только не приводит к потере решений, но и не добавляет в систему новых решений. Теоретическое обоснование этого нами опускается. С учетом (4) перепишем (1) таким образом:

$$on(t+1) = m_C(t) \cdot b_C(t) \cdot \varepsilon_1(t). \qquad (5)$$

Теперь надо выполнить подстановку (5) в систему исходных дизъюнктов и получить новые подстановки. Опуская эти действия, получим окончательно:

$$cap(t+1) = on(t) \cdot \varepsilon_2(t),$$
$$m_B(t+1) = \bar{m}_C(t) \cdot (\bar{m}_B(t) \vee \bar{b}_B(t)) \cdot \varepsilon_3(t),$$
$$m_C(t+1) = (m_B(t) \vee \bar{b}_C(t+1)) \times$$
$$\times (b_B(t) \vee \bar{b}_C(t+1)) \cdot \varepsilon_4(t),$$
$$\bar{b}_C(t+1) = (\bar{m}_B(t) \vee \bar{b}_B(t)) \cdot \varepsilon_5(t).$$

Система трансформируется к следующему виду:

$$\begin{gathered} on(t) \vee \bar{m}_C(t) \vee \bar{b}_C(t), \\ \overline{on}(t) \vee \varepsilon_1(t), \\ \overline{on}(t) \vee cap(t) \vee \varepsilon_2(t), \\ \bar{m}_A(t) \vee \bar{b}_B(t) \vee \bar{m}_C(t), \\ \bar{m}_A(t) \vee \bar{b}_B(t) \vee \bar{m}_B(t), \\ \bar{m}_A(t) \vee \bar{b}_B(t) \vee \varepsilon_3(t), \\ \bar{m}_A(t) \vee \bar{m}_B(t), \\ \bar{m}_A(t) \vee \bar{m}_C(t), \\ \bar{m}_B(t) \vee \bar{m}_C(t), \\ m_A(t) \vee m_B(t) \vee m_C(t), \\ b_B(t) \vee b_C(t), \\ \bar{b}_B(t) \vee \bar{b}_C(t), \\ \bar{b}_B(t) \vee \overline{on}(t). \end{gathered} \qquad (6)$$

Мы имеем исходное и конечное состояния, систему подстановок и систему формул, ограничивающих множества достижимых состояний на каждом шаге.

Метод. Суть метода практически очевидна: система подстановок используется для порождения последовательности состояний

$$S_0 \to S(1) \to S(2) \to S(3) \to \ldots \to Sfin.$$

В каждом промежуточном состоянии проверяем, не достигнуто ли целевое условие, которое в нашем примере соответствует истинному значению формулы $cap(t+z)$. Исходное состояние таково:

$$S_0 = \begin{array}{l} m_A(0) = 1, \\ b_B(0) = 1, \\ on(0) = cap(0) = m_B(0) = m_C(0) = 0. \end{array}$$

Применение подстановок дает

$$b_C(1) = 0,$$
$$m_C(1) = \varepsilon_4(0),$$
$$m_B(1) = \bar{\varepsilon}_4(0) \cdot \varepsilon_3(0),$$
$$cap(1) = 0,$$
$$on(1) = \varepsilon_1(t) = 0,$$
$$b_B(1) = 1.$$

Эта система дает три возможных решения, из которых только одно имеет смысл:

$$b_C(1) = 0,$$
$$m_C(1) = 0,$$
$$m_B(1) = 1,$$
$$cap(1) = 0,$$
$$on(1) = \varepsilon_1(t) = 0,$$
$$b_B(1) = 1.$$

Опять, используя систему подстановок, получаем новое состояние:

$$\bar{b}_C(2) = 0,$$
$$b_C(2) = 1,$$
$$m_C(2) = \varepsilon_4(1),$$
$$m_B(2) = \varepsilon_3(1),$$
$$cap(2) = 0,$$
$$on(2) = 0.$$

После этого процесс выполняем по аналогии, пока не получим требуемого конечного состояния. В нашем примере решение будет найдено за четыре шага.

Заключение. Суть предложенного метода синтеза поведения интеллектуальной системы практически очевидна. В качестве сравнения можно использовать известный метод логического дифференциала, который требует находить все возможные допустимые решения исходной системы формул, что уже даже для небольших задач с десятком переменных становится проблемой. В предложенном методе требуется анализировать возможные продолжения из текущего состояния, поэтому его эффективность будет тем выше, чем меньше продолжений допускается в каждом текущем состоянии, т. е. чем более узким фронтом выполняется поиск решения. Во многих случаях это обстоятельство имеет место.

Литература

1. Герман, О. В. Синтез управляющего алгоритма в системе продукционных правил с временным параметром. / О. В. Герман, Д. В. Занько // Автоматика и телемеханика. – 2003. – № 5. – С. 41–52.

2. Герман, О. В. Неклассические логические исчисления / О. В. Герман. – Минск: БГУИР, 2012. – 125 с.

Поступила 04.03.2013