

SYSTEMY EKSPERTOWE CONTROLL-LOG/S I CLASS-LOG/S DLA POTRZEB AUTOMATYCZNEGO STEROWANIA I ROZPOZNAWANIA

Dariusz Gajewski
Jerzy Józefczyk
Mirosław Szczypiński

Instytut Sterowania i Techniki Systemów, Politechnika Wrocławska

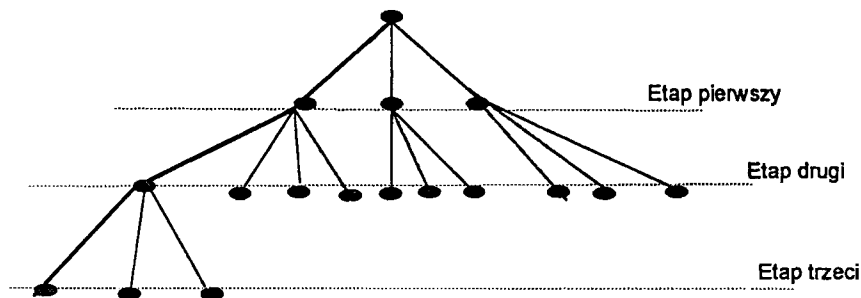
Streszczenie. W artykule przedstawiono systemy ekspertowe ogólnego przeznaczenia CONTROL-LOG/S i CLASS-LOG/S, które mogą mieć zastosowanie odpowiednio do zadań sterowania i rozpoznawania w przypadku kiedy opisy sterowanych/rozpoznawanych obiektów odbiegają od tradycyjnej postaci i przyjmują postać reprezentacji wiedzy. Prace nad rozwojem systemów prowadzone są w ramach projektu badawczego nr 8S50501504.

1. Wprowadzenie

Do problemu podejmowania decyzji na podstawie określonej bazy wiedzy, można podejść w dwojaki sposób. Pierwszy (najczęściej omawiany w literaturze) to sposób polegający na przeszukiwaniu przestrzeni rozwiązań za pomocą określonego algorytmu - zwanego metodą wnioskowania. W zależności od zastosowanego prawa wnioskowania (modus ponens lub modus tollens) wyróżnia się dwie metody: przeszukiwanie wprzód oraz przeszukiwanie wstecz. Obie metody posiadają wiele odmian w zależności od stosowanych reguł podejmowania decyzji na poszczególnych etapach wnioskowania (problem wyboru ścieżki). Modyfikacje polegają głównie na wyborze kolejności konkluzji dostępnych na danym etapie wnioskowania (np. metoda wprzód "pierszy wgląd" ang. depth first search).

Omawiane metody ze względu na zaimplementowane w nich reguły wnioskowania oraz związany z nimi sposób działania (wyciąganie wniosków z dostępnych na danym poziomie przesłanek) nazywane są często metodami dedukcyjnymi, w odróżnieniu od drugiej grupy algorytmów stosowanych do rozwiązywania problemów w systemach z bazą wiedzy zwanych metodami indukcyjnymi. Zastosowanie tych algorytmów wymaga opisanie problemu za pomocą zdań sformułowanych w języku logiki (poziom wartości logicznych) i stosowania dalej przekształceń algebraicznych. To właśnie różni opisane wyżej metody dedukcyjne od metod indukcyjnych. Zastąpiono w nich tradycyjnie stosowane reguły wnioskowania regułami

wykonywania działań i przekształceń w algebrze logiki. Opis problemu na poziomie wielkości logicznych pozbawiony jest części znaczeniowej (opisowej) co powoduje, iż dany opis może być stosowany do problemów z różnych obszarów życia, którymi rządzą te same reguły działania.



Rys.1. Przeszukiwanie metodą "wprzód pierwszy wgląd"

Dla systemów z reprezentacją wiedzy wyodrębnia się przeważnie dwie klasy problemów tzw. problem analizy oraz problem podejmowania decyzji (nazywany często problemem syntezy). Z problemem analizy najczęściej spotykamy się w systemach doradczych. Jest to problem odpowiedzi na pytanie czy dane zdanie sformułowane przez użytkownika jest prawdziwe przy danej reprezentacji wiedzy i systemie wnioskowania. Problem sterowania, który najczęściej pojawia się w systemach związanych z zarządzaniem, jest problemem odwrotnym czyli co powinno być spełnione, by dane zdanie było prawdziwe. W przypadku metod indukcyjnych (podejście logiczno-algebraiczne) oba problemy możemy sformułować w opisany poniżej sposób.

Oznaczmy przez :

$(\alpha_1, \alpha_2, \dots, \alpha_n) = \alpha$ - ciąg formuł elementarnych

$(a_1, a_2, \dots, a_n) = a$ - ciąg wartości logicznych formuł

$F_i(\alpha)$ - i -ty fakt w bazie wiedzy gdzie $i = 1, \dots, k$

$w[F_i(\alpha)] = F_i(a)$ - wartość logiczną i-tego faktu

W omawianych metodach zakładamy, że wszystkie fakty są prawdziwe czyli

$$F_i(a) = 1 \text{ dla } i = 1, \dots, k$$

Przy tych oznaczeniach problem analizy możemy sformułować następująco:

Dla danej bazy wiedzy określić wartość logiczną danego zdania $F_b(\alpha_b)$. Pytamy więc o wartość:

$$b = w[F_b(\alpha_b)]$$

Na poziomie logicznym b może przybrać trzy wartości 0 (odpowiedź, "nie" - zdanie F_b nie jest prawdziwe dla danej reprezentacji wiedzy), 1 (odpowiedź, "tak" - zdanie F_b jest prawdziwe) lub ? (odpowiedź, "nie wiem" - nie można określić wartości logicznej zdania F_b na podstawie danej reprezentacji wiedzy).

Problem wyznaczania decyzji możemy przedstawić w następujący sposób:

Dla danej reprezentacji wiedzy oraz postawionego przez użytkownika zdania F_b wyznaczyć najlepszą decyzję $F_c(\alpha_c)$ dla której prawdziwe jest zdanie:

$$F_c(\alpha_c) \Rightarrow F_b(\alpha_b)$$

Algorytm generacji rozwiązania dla zadania analizy polega na wygenerowaniu zbioru S_a wszystkich ciągów zero-jedynkowych o długości n , następnie wyróżnieniu podzbioru S_b o elementach spełniających warunek:

$$F_b(a)=1$$

a następnie porównaniu go ze zbiorem S spełniającym warunek (fakty w bazy wiedzy są prawdziwe):

$$F(a)=1$$

W zależności od wzajemnej relacji zbiorów S_b oraz S otrzymujemy właściwą odpowiedź:

$$S \subset S_b \text{ odpowiedź "tak",}$$

$$S \subset S_a - S_b \text{ odpowiedź "nie",}$$

$$S \not\subset S_a - S_b \cap S \not\subset S_b \text{ odpowiedź "nie wiem"}$$

Dla zadania syntezy generacja odpowiedzi jest podobna. Wygenerowany zbiór S dzielimy na dwa podzbiory S_1 wszystkich ciągów dla których prawdziwe jest zdanie $F_b(a)=1$ oraz jego dopełnienie do zbioru S zbiór S_2 . Oznaczmy przez S_{c1} zbiór części ciągów a_c ($a=(a_c, a_w, a_b)$) należących do S_1 oraz S_{c2} zbiór części ciągów a_c należących do S_2 . Forma zdania F_c jest określona przez zbiór $S_{c1} - S_{c2}$ w następujący sposób: każdemu ciągowi a_c należącemu do różnicy $S_{c1} - S_{c2}$ odpowiada jeden składnik alternatywy:

$$A_1 \vee A_2 \vee \dots \vee A_p$$

gdzie p określa licznosc zbioru $S_{c1} - S_{c2}$, natomiast A_i ma postac :

$$\overline{\alpha_{c1}} \wedge \overline{\alpha_{c2}} \wedge \dots \wedge \overline{\alpha_{ca}}$$

W ostatnim wyrazieniu s oznacza liczbe formu w ciagu a_c natomiast $\overline{\alpha_{ci}} = \alpha_{ci}$ gdy i -ty wyraz ciagu a_c , któremu A odpowiada, rowna sie 1 oraz $\overline{\alpha_{ci}} = \neg \alpha_{ci}$ gdy rowna sie 0.

Przedstawione wyzej metody generacji rozwiazania sa metodami ogolnym pozwalajacymi na rozwiazywanie dowolnych problemow opisanych na poziomie logicznym. Ze wzgledu jednak na czas generacji rozwiazania (uzalezniony glownie od czasu potrzebnego na wygenerowanie 2^n ciagow 0-1) w praktyce stosowane sa metody rekurekcyjne oparte na dekompozycji bazy wiedzy.

Dekompozycja bazy wiedzy polega na takim jej podziale, który wyodrębnia jedynie fakty związane z postawionym przez użytkownika pytaniem, odrzucając fakty z nim nie związane. Zasadę dekompozycji można przedstawić w następujący sposób:

Oznaczmy przez $a_0 = \overline{a_0}$ ciąg wartości formuł elementarnych występujących w zdaniu F_b . Znajdźmy wszystkie fakty, w których występują formuły elementarne $\overline{a_0}$. Oznaczmy koniunkcję tych faktów przez $\overline{F_1(a_0, a_1)}$, gdzie $\overline{a_1}$ oznacza podzbiór wszystkich pozostałych formuł elementarnych występujących w faktach $\overline{F_1}$. Kolejny etap to wybór ze zbioru pozostałych faktów tych, które zawierają formuły elementarne z podciagu $\overline{a_1}$. Ciąg tak wybranych faktów oznaczmy przez $\overline{F_2(a_1, a_2)}$. Procedurę tą powtarzamy aż do wyczerpania całego zbioru faktów. Jako wynik dekompozycji otrzymamy podział ciągu a na podciągi:

$$a = (\overline{a_0}, \overline{a_1}, \dots, \overline{a_N})$$

natomiast zbiór faktów zostanie podzielony na podzbiory:

$$\overline{F_j(a_{j-1}, a_j)}$$

Do tak zdekomponowanej bazy wiedzy możemy stosować zmodyfikowane (rekurekcyjne) metody generacji rozwiązania. W metodach tych zbiór S_0 zastępowany jest przez zbiór S_0 , zbiór S_b zbiorem $\overline{S_b}$, oraz zbiór S zbiorem $\overline{S_0}$. Przy czym zbiór $\overline{S_{N-1}}$ to zbiór ciągów $\overline{a_{N-1}}$, dla których spełniony jest warunek:

$$\overline{F_N(a_{N-1}, a_N)} = 1,$$

czyli chodzi o znalezienie takich ciągów $\overline{a_{N-1}}$, dla których w zbiorze $\overline{S_N}$ istnieje ciąg $\overline{a_N}$ taki że spełniony jest powyższy warunek. Procedurę znajdowania zbiorów $\overline{S_{j-1}}$ powtarzamy dla

$j=N, N-1, \dots, 0$. W rezultacie otrzymujemy zbiór $\overline{S}_0$, który podobnie jak w metodzie podstawowej porównujemy ze zbiorem $\overline{S}_b$ (zbiór ciągów $a_b = \overline{a}_b$, dla których $F_b(a_b)=1$). Otrzymujemy więc następujące relacje:

$$\begin{aligned}\overline{S}_0 \subset \overline{S}_b & \text{ odpowiedź "tak",} \\ \overline{S}_0 \subset S_0 - \overline{S}_b & \text{ odpowiedź "nie",} \\ \overline{S}_0 \not\subset S_0 - \overline{S}_b \wedge \overline{S}_0 \not\subset \overline{S}_b & \text{ odpowiedź "nie wiem"}\end{aligned}$$

Jak widać operacja określająca czas generacji odpowiedzi, czyli generowanie ciągów 0-1, jest w tej metodzie ograniczona na poszczególnych etapach do niezbędnego minimum. Dzięki temu możliwa jest analiza nawet dużych (zawierających wiele formuł elementarnych) baz wiedzy.

2. System CONTROL-LOG/S

System CONTROL-LOG/S jest systemem uniwersalnym. W szczególności może być zastosowany do zadań związanych ze sterowaniem obiektem opisanym przy pomocy reprezentacji wiedzy przedstawionej w poprzednim punkcie. System może wspomagać też prace projektowe. Zadania realizowane przez system należą do dwóch kategorii: zadania ogólne (analiza, synteza) oraz zadania związane z kompleksami operacji. W zakresie kompleksów operacji system pozwala na rozwiązanie prostego zadania szeregowania oraz wyznaczenie kolejności operacji. Zadania te realizowane mogą być jedną z trzech metod: bez dekompozycji, z dekompozycją w przód, z dekompozycją wstecz. Wybór metody zależy od użytkownika systemu. System opracowywany jest w środowisku graficznym X Window na komputer Sun SPARC Station 2. Jednym z istotnych aspektów konstrukcji systemu jest właśnie wygodny, graficzny sposób komunikacji z użytkownikiem.

Problemy należące do klasy zadań ogólnych formułowane są przy pomocy języka CLASS-LOG. Zadania związane z kompleksami operacji formułowane są poprzez odpowiedni wybór w menu i formy ekranowe. Poniżej przedstawiony jest przykład zadania rozwiązywanego przez system CONTROL-LOG/S: zapis formuł elementarnych (a), konstrukcja bazy wiedzy (b) i odpowiedni program w języku CONTROL-LOG (c).

Przykład 1

a. Formuły elementarne:

$\alpha_1 = "O_{12} \text{ jest wykonywana po } O_{11}"$, $\alpha_2 = "O_{22} \text{ po } O_{21}"$, $\alpha_3 = "O_{32} \text{ po } O_{31}"$, $\alpha_4 = "wystąpił stan S"$,
 $\alpha_5 = "temperatura \text{ jest mała}"$, $\alpha_6 = "ciśnienie \text{ jest wysokie}"$, $\alpha_7 = "potrzebny \text{ jest zasób } w"$,
 $\alpha_8 = "powinien \text{ zostać użyty } E_1"$, $\alpha_9 = "T < d"$, $\alpha_{10} = "wystąpił stan \bar{S}"$

b. Fakty:

1. $(\alpha_3 \vee \neg \alpha_1) \Rightarrow \alpha_4$,
2. $(\neg \alpha_1 \wedge \alpha_7) \vee \neg \alpha_3$,
3. $(\alpha_9 \wedge \alpha_1) \Rightarrow \alpha_2$,
4. $(\alpha_4 \wedge \neg \alpha_7) \vee \alpha_5$,
5. $\alpha_6 \Rightarrow (\alpha_4 \wedge \alpha_8)$,
6. $\alpha_2 \Rightarrow (\neg \alpha_4 \wedge \alpha_5)$,
7. $(\alpha_3 \wedge \alpha_2) \vee \alpha_{10}$.

Pytanie: Czy " $T < d$?"

c. Program:

facts

$\{a3\} \text{ or not } \{a1\} \Rightarrow \{a4\};$
 $(\text{not } \{a1\} \text{ and } \{a7\}) \text{ or not } \{a3\};$
 $(\{a9\} \text{ and } \{a1\}) \Rightarrow \{a2\};$
 $(\{a4\} \text{ and not } \{a7\}) \text{ or } \{a5\};$
 $\{a6\} \Rightarrow (\{a4\} \text{ and } \{a8\});$
 $\{a2\} \Rightarrow (\text{not } \{a4\} \text{ and } \{a6\});$
 $(\{a3\} \text{ and } \{a2\}) \text{ or } \{a10\};$

invars

$\{a3\}$

goal

$\{a9\}$

end

3. System CLASS-LOG/S

Wprowadźmy następujące oznaczenia potrzebne do sformułowania problemu:

j - indeks klasy, $j \in \{1, 2, \dots, M\} \stackrel{\Delta}{=} J$

$x \in X$ - obserwowane cechy,

$z \in Z$ - zmienne dodatkowe.

Reprezentacja wiedzy jest następująca RW :

$$RW = \langle (\alpha_x, \alpha_z, \alpha_j), F(\alpha_x, \alpha_z, \alpha_j) \rangle$$

gdzie zbiór wszystkich faktów RW :

$$F(\alpha_x, \alpha_z, \alpha_j) \stackrel{\Delta}{=} [F_1(\alpha_x, \alpha_z, \alpha_j), \dots, F_k(\alpha_x, \alpha_z, \alpha_j)]$$

Zakładamy, że wszystkie fakty są prawdziwe:

$$F = F_1 \wedge \dots \wedge F_k = 1$$

gdzie

$F_i(\alpha_x, \alpha_z, \alpha_j)$ - to fakt złożony z formuł $(\alpha_x, \alpha_z, \alpha_j)$ z użyciem: $\wedge, \vee, \neg, \Rightarrow$.

$\alpha_x = (\alpha_{x_1}, \dots, \alpha_{x_p})$ - ciąg formuł α_{x_i}

α_{x_i} - formuła elementarna,

α_z - ciąg formuł α_{z_i} ,

$\alpha_{z_i}(x, z)$ - formuła elementarna zbudowana z użyciem x, z ,

$\alpha_j = (\alpha_{j_1}, \dots, \alpha_{j_M})$ - ciąg formuł α_{j_i} ,

α_{j_i} - formuła elementarna "obiekt należy do klasy i "

Problem rozpoznawania: Dla danego α_z i α_j , znajdź F_j

$$F_j(\alpha_j) = \bigvee_{n=1}^s [(\alpha_{j_n} \text{ dla } i = j_n) \wedge \bigwedge_{i \neq j_n} \neg \alpha_{j_i}]$$

dla najmniejszego s

$$F(\alpha_x, \alpha_z, \alpha_j) \wedge F_x(\alpha_x) \Rightarrow F_j(\alpha_j)$$

Rozwiązanie :

$$D_j = \{j_1, \dots, j_s\} \subset J$$

System CLASS-LOG/S jest systemem uniwersalnym przeznaczonym do rozwiązywania zadań rozpoznawania, a w szczególności może on znaleźć zastosowanie w diagnostyce obiektów technicznych w przypadku kiedy opis obiektu odbiega od postaci tradycyjnej i przyjmuje postać reprezentacji wiedzy. CLASS-LOG/S jest systemem zintegrowanym w skład którego wchodzi edytor, kompilator i inne elementy powiązane przez graficzny interfejs z użytkownikiem. Podobnie jak system CONTROL-LOG/S system ten jest realizowany w środowisku graficznym X Window na komputer Sun SPARC Station 2. Zadania dla systemu formułuje się w języku CLASS-LOG dedykowanym dla zadań rozpoznawania. Poniżej pokazany jest przykład przejścia od opisu problemu do zapisu odpowiedniego programu: zapis formuł elementarnych (a), zapis faktów w języku logiki (b), program w języku CLASS-LOG (c).

Przykład 2

a. Formuły elementarne:

$\alpha_{x1} = "x \in D_{x1}"$, $\alpha_{x2} = "x \in D_{x2}"$, $\alpha_{zi} = "z \in D_{zi}"$, $i \in \overline{1,7}$, $\alpha_{z8} = "(x,z) \in D"$,

$\alpha_{j1} = "j = 1"$, $\alpha_{j2} = "j = 2"$

b. Fakty:

1. $(\alpha_{x1} \wedge \alpha_{x3}) \vee \neg \alpha_{x1} \Rightarrow \alpha_{z4}$,

2. $(\alpha_{x1} \wedge \alpha_{z7}) \vee \neg \alpha_{x3}$,

3. $(\alpha_{j1} \wedge \alpha_{x1}) \Rightarrow \alpha_{z2}$,

4. $(\alpha_{z4} \wedge \neg \alpha_{z7}) \dot{\vee} (\alpha_{z5})$,

5. $(\alpha_{x2} \wedge \alpha_{z6}) \Rightarrow (\alpha_{z4} \wedge \alpha_{z8})$,

6. $(\alpha_{x1} \wedge \alpha_{z2}) \Rightarrow (\alpha_{z4} \wedge \alpha_{z6})$,

7. $(\alpha_{z3} \wedge \alpha_{z2}) \vee (\alpha_{j2})$.

c. Program

facts

$((\{ax1\} \text{ and } \{az3\}) \text{ or not } \{az1\}) \Rightarrow \{az4\}$;

$((\{az1\} \text{ and } \{az7\}) \text{ or not } \{az3\})$;

$((\{aj1\} \text{ and } \{az7\}) \text{ or not } \{az3\})$;

$((\{aj1\} \text{ and } \{az1\}) \Rightarrow \{az2\})$;

$((\{az4\} \text{ and not } \{az7\}) \text{ or } \{az5\})$;

$((\{az4\} \text{ and not } \{az7\}) \text{ or } \{az5\})$;

$((\{ax2\} \text{ and } \{az6\}) \Rightarrow ((\{az4\} \text{ and } \{az8\}))$;

$((\{ax1\} \text{ and } \{az2\}) \Rightarrow ((\{az4\} \text{ and } \{az6\}))$;

$((\{az3\} \text{ and } \{az2\}) \text{ or } \{aj2\})$;

classes

$\{aj1\}$, $\{aj2\}$

features

$\{ax1\}$, $\{ax2\}$

end

3. Podsumowanie

Systemy ekspertowe CONTROL-LOG/S i CLASS-LOG/S są rozwinięciem wcześniej opracowanych wersji. Rozwinięcie polega na dodaniu nowych metod rozwiązywania zadań (metody z dekompozycją), nowych zadań możliwych do rozwiązywania przez systemy (np. kompleksy operacji) oraz usunięciu barier technicznych uniemożliwiających rozwiązywanie zadań o dużej liczbie formuł elementarnych i faktów (a co za tym idzie o dużej zajętości pamięci). Rozwinięcie polega też na zmianie interfejsu ze znakowego na graficzny i zmianie platformy sprzętowo-programowej. Docelowo systemy będą pracować w środowisku X Window i UNIX-a. W chwili obecnej w fazie testów znajduje się system CONTROL-LOG/S oraz w ich wyniku dokonywane są odpowiednie zmiany w jądrze obydwu systemów. W fazie oprogramowania są też ostateczne interfejsy graficzne. Wersja testowa dostępna jest w chwili obecnej dla systemów MS Windows, LINUX, SunOS 4.1.1. oraz SCO UNIX (z prostym interfejsem znakowym). W nowe wersje zostaną także wbudowane udogodnienia pozwalające na wygodne prowadzenie testów pozwalających na ocenę różnych aspektów wbudowanych w system metod rozwiązywania zadań.

Literatura

- [1] Bubnicki Z., Problems of Knowledge-based Control Systems, Proc. VII Int. Conf. on Syst. Eng., Las Vegas, s.501-509, 1990
- [2] Bubnicki Z., Knowledge-based Decision Making for Manufacturing Systems, Proc. VIII Int. Conf. on Syst. Eng., Coventry, 1991
- [3] Bubnicki Z., Introduction to Expert Systems, Warsaw, PWN, 1990 (in Polish)
- [4] Bubnicki Z., Foundations of Information Management Systems, Wrocław, WPWR, 1993 (in Polish)
- [5] Bubnicki Z., New Problems and Directions in the Area of Control and Identification for Knowledge-Based Systems, in Modelling Identification and Control, Zurich, Acta Press, 1993
- [6] Bubnicki Z., Knowledge-based Approach as a Generalisation of Pattern Recognition Problems and Methods, Systems Science, vol. 19, no.2, WTUP, Wrocław, 1993
- [7] Bubnicki Z.: Decomposition of a System Described by Logical Model, Cybernetic and System Research, London, New Jersey: World Scientific, 1992
- [8] D.Gajewski, M.Szczypiński, Systemy ekspertowe ogólnego przeznaczenia CONTROL-LOG i CLASS-LOG, II Krajowa Konferencja Naukowa "Inżynieria Wiedzy i Systemy Ekspertowe", Wrocław, 1993