

Witold BRANDYS
Politechnika Śląska

KILKA PRZYKŁADÓW ZASTOSOWAŃ ALGORYTMÓW TEORII GRAFÓW W ELASTYCZNYCH SYSTEMACH PRODUKCYJNYCH

Streszczenie: Celem pracy jest przedstawienie wybranych sposobów reprezentacji wiedzy za pomocą teorii grafów. Praca przedstawia również elementy teorii grafów, przegląd aktualnie stosowanych metod reprezentacji wiedzy oraz niektóre zastosowania teorii grafów w metodach sztucznej inteligencji. Praktycznej realizacji dokonano za pomocą typowych algorytmów teorii grafów, które zaczerpnięto z literatury podanej w bibliografii. W pracy położono nacisk na jej charakter dydaktyczny, co pozwala na jej zastosowanie w laboratorium.

SOME EXAMPLES OF GRAPH ALGORITHMS IN FLEXIBLE MANUFACTURING

Summary: The aim of the paper is to discuss some yields of knowledge representation using the graph theory. The paper illustrates also fundamentals of the graph theory, review of actual knowledge representation methods and some applications of the graph theory being used in Artificial Intelligence methods. The practical implementation has been made with typical graph theory algorithms, basing on references included in bibliography. In the work main issue has been directed to its educational character and its use in students' laboratory.

ANWENDUNGSBEISPIELE DER ALGORITHMEN DER GRAPHEN-THEORIE IN DEN FLEXIBLEN PRODUKTIONSSYSTEMEN

Zusammenfassung: In der Arbeit werden die ausgewählten Methoden der Wissenrepräsentation mit Hilfe der Graphentheorie vorgestellt. Die Einführung in die Graphentheorie, Grundlagen, Methoden und Erfahrungen der Wissenrepräsentation sowie manche Anwendungen der Künstlichen Intelligenz werden diskutiert. Praktische Realisierung wurde mit Hilfe der typischen, bekannten aus Literatur, Algorithmen verwirklichen. Die Reswitate der Arbeit werden aus Rücksicht auf didaktischen Charakter, im studentischen Labor angewendet.

1. Zarys współczesnych metod reprezentacji wiedzy

Reprezentacja wiedzy stanowi jedno z ważniejszych zagadnień sztucznej inteligencji. Teoretyczne opracowania dotyczące reprezentacji wiedzy są stosunkowo nowe i przedstawiają

zwykle alternatywne sposoby pamiętania i interpretacji świata rzeczywistego w pamięci komputera. Blok reprezentacji wiedzy stanowi element, bez którego nie istnieje rozwiązanie zadania inteligentnego, tj. takiego, którego dokładnej metody rozwiązania nie znamy. Problem reprezentacji wiedzy polega na znalezieniu takiej metody symbolicznego kodowania wiedzy, aby wiernie odzwierciedlić rzeczywiste informacje i aby umożliwić komputerowi skuteczne manipulowanie tymi informacjami. Blok reprezentacji wiedzy dzieli się na blok interpretacji, blok uczenia się, inteligentną bazę wiedzy i blok generowania rozwiązań. Z kolei inteligentna baza wiedzy składa się z bazy wiedzy zawierającej wiadomości o danej dziedzinie przedmiotowej oraz bazy danych zawierających konkretne dane ilościowe i faktograficzne.

Baza wiedzy jest zbiorem definicji, faktów, pojęć i relacji pomiędzy nimi oraz reguł wnioskowania. Wiedza przekazana przez specjalistów, musi być poselekcjonowana i zorganizowana w taki sposób, aby można było ją przechowywać w pamięci komputera i za pomocą tegoż komputera efektywnie z niej korzystać. Proces organizowania wiedzy wymaga odpowiedniej metody reprezentacji wiedzy, zakodowania wiedzy zgodnie z przyjętym formatem oraz weryfikacji bazy wiedzy i mechanizmu wnioskowania.

Baza wiedzy dzieli się na: bazę tekstową - zawierającą objaśnienia i fakty, bazę reguł - zawierającą reguły deklaratywne i reguły formalne, bazę numeryczną - zawierającą procedury i modele matematyczne oraz bazę zdrowego rozsądku - zawierającą reguły specjalne. Każda baza wiedzy powinna przy tym zawierać w sobie model świata oraz wiedzę o celach.

Istnieją dwa rodzaje wiedzy:

- intencjonalna - opisująca abstrakcyjne obiekty, zdarzenia oraz związki występujące między obiektami,
- ekstencjonalna - zawierająca dane charakteryzujące konkretne obiekty, ich stany, wartości parametrów w określonych chwilach czasu.

Wiedza przedstawiana w komputerze jest wiedzą semiotyczną, czyli opartą na systemie znaków. Każdy znak zawarty w bazie wiedzy posiada zarówno swój intencjonal, jak i ekstencjonal. W zależności od użytego systemu znakowego, wiedza reprezentowana w komputerze może przyjmować różne formy, zawsze jednak system semiotyczny, według którego zostanie ona opisana, zawierać będzie przynajmniej jedną z trzech składowych: syntaktyczną, semantyczną lub pragmatyczną. Składowa syntaktyczna jest to wewnętrzne rozwinięcie systemu znakowego. Opisuje ona zasady budowy i przetwarzania złożonych wyrażeń. Wiedza syntaktyczna nie zależy przy tym od sensu i treści używanych pojęć. Składowa semantyczna wyraża zależności pomiędzy znakami i ich intencjonalami, nadaje znakom sens i znaczenie. Wiedza semantyczna zawiera informacje bezpośrednio powiązane ze znaczeniem i sensem opisywanych obiektów i zjawisk. Składowa pragmatyczna wyznacza znaki z punktu widzenia konkretnej sfery ich zastosowań lub podmiotu wykorzystującego dany system znakowy. Wiedza pragmatyczna charakteryzuje obiekty i zjawiska z punktu widzenia

rozwiązywanego zadania. Uwzględnia ona także ograniczenia nałożone na rozwiązanie zadania. Uwzględniając powyższe składowe otrzymujemy trzy podstawowe modele reprezentacji wiedzy. Dla zagadnień sztucznej inteligencji charakterystyczne jest istnienie modeli semantycznych i pragmatycznych nie spotykanych w innych dziedzinach badań związanych z modelami wiedzy.

Podział na konkretne reprezentacje wiedzy używane współcześnie, szczególnie w zagadnieniach sztucznej inteligencji, przedstawiono poniżej. Większość reprezentacji wiedzy omówiono skrótowo w dalszej części tego rozdziału, natomiast te rodzaje reprezentacji wiedzy, które opisane są przy wykorzystaniu teorii grafów, omówiono w rozdziale drugim.

Współczesne reprezentacje wiedzy:

1. Lingwistyczna (relacyjna) - opisana za pomocą stwierdzeń i reguł (jeżeli ... to ...), dopuszcza nieścisłość i niepewność. Jej opisem zajmuje się logika matematyczna. Dokładnie określone reguły wnioskowania mają zwykle postać: *jeżeli* ($P_1 \& P_2 \& \dots P_n$) *to* ($Q_1 \& Q_2 \& \dots Q_n$), gdzie $P_1 \dots P_n$ reprezentują warunki i zwykle definiują pewien wzorzec lub wymóg, który musi być spełniony, aby można było wykorzystać daną regułę, $Q_1 \dots Q_n$ reprezentują konsekwencje i zwykle określają akcję, która ma być wykonana lub konkluzję, która staje się faktem, w przypadku gdy warunki zostały spełnione [6].

2. Logiczna (rachunek predykatów, warunki Horna) - jest bardziej precyzyjna od wiedzy lingwistycznej. Jej opisem zajmuje się rachunek predykatów, a jej struktura zbliżona jest do generalnej postaci reguł wnioskowania z pewnymi dodatkowymi restrykcjami wynikającymi z zasad rachunku predykatów. Między innymi warunki muszą być w postaci koniunktywnej oraz dopuszczają tylko jedną konsekwencję [6].

3. Trójki: { obiekt, atrybut, wartość (O, A, W) } - tego typu reprezentacja wiedzy jest wykorzystywana w wielu innych metodach reprezentacji wiedzy, np. za pomocą grafu. Atrybuty są generalnymi cechami obiektu, takimi jak np. rozmiar, kształt, czy kolor i definiują stan obiektu w danym momencie [6].

4. Grafowa (grafy skierowane, drzewa decyzyjne, grafy *and / or*, drzewa binarne, sieci Petri) - jej opisem zajmuje się teoria grafów.

5. Sieci semantyczne (grafowe przedstawienie rachunku predykatów) - jej opisem zajmuje się teoria grafów.

6. Gry (macierze, drzewa decyzyjne, metody optymalizacji) - jej opisem zajmuje się teoria gier. Ten opis wiedzy znajduje swoje zastosowanie np. w problemach podejmowania decyzji. Występują w nim decydenci, którzy zmuszeni są do podejmowania konfliktowych decyzji, przy czym możliwe jest także założenie o kooperacji między nimi.

7. Sieci neuronowe - system neuronowy charakteryzuje się równoległym przetwarzaniem danych, zdolnością adaptacji i uczenia się, umiejętnością rekonstrukcji niepełnej lub częściowo przekłamaną informacją oraz tolerancją błędów. Porównując własną odpowiedź z prawidłową,

sieć neuronowa zmienia wartości wag na swoich wewnętrznych połączeniach. Jeżeli odpowiedź sieci jest prawidłowa, to wagi odpowiednich wiązań są zwiększane, w przeciwnym przypadku następuje ich zmniejszenie. Wiedza w sieciach neuronowych zapisana jest w wagach ich połączeń, które po zakończeniu procesu uczenia się nie są modyfikowane. Neurosystemy posiadają warstwę wejściową oraz wyjściową. Pomiedzy nimi znajduje się jedna lub wiele warstw (określanych jako ukryte) zawierających wagi. "Inteligencja" systemu neuronowego, umożliwiająca porównywanie bodźców zewnętrznych z wyuczonymi wzorcami, tkwi właśnie w połączeniach międzykomórkowych oraz ich wagach [9].

8. Listy (zbiory) - są to systemy reprezentujące wiedzę skonstruowane tak, aby jak najlepiej informację można było zapisać w pamięci komputera. Oparte są na adresach tej pamięci tak, że adresy reprezentują przejścia między kolejnymi elementami listy elementów zapisywanej w pamięci komputera [2].

9. Listy i struktury rozmyte - zbudowane analogicznie do list opisanych powyżej, różnią się tym, że przejście do kolejnego elementu listy następuje z pewnym prawdopodobieństwem ustalonym podczas konstruowania struktury [2].

10. Ramy wiedzy (scenariusze) - są one bardziej ogólną i elastyczną formą opisu za pomocą rekordu, czyli formalnego zapisu struktury danych. Pola rekordu odpowiadają atrybutom obiektu i zwykle zawierają wartości atrybutów, wskaźniki do innych ram, zbiory reguł lub odwołania do procedur, za pomocą których można uzyskać wartości atrybutów [6]. Ramy mogą być połączone ze sobą i zorganizowane w hierarchię zgodnie z zasadą "od ogółu do szczegółu". Koncepcja ram opiera się na dwóch podstawowych pojęciach: grupowania danych i procedur w jednostki zwane obiektami lub ramami - enkapsulacja, oraz dziedziczenia cech, parametrów i procedur przez ramy niższe w hierarchii - dziedziczenie. Ten sposób reprezentacji wiedzy jest wykorzystywany np. w informatyce do programowania zorientowanego obiektowo.

2. Podstawowe pojęcia teorii grafów

Grafem nazywamy taką parę (W, K) , że W jest niepustym zbiorem elementów, zaś K relacją dwuargumentową na elementach tego zbioru. Ze względu na interpretację geometryczną, najczęściej określa się W jako zbiór wierzchołków, zwykle reprezentujących obiekty fizyczne oraz K jako zbiór krawędzi, które zwykle reprezentują relacje zachodzące między tymi obiektami fizycznymi. Zakłada się przy tym, że obiekty fizyczne reprezentowane przez graf są obiektami dyskretnymi, a także nie są ważne wewnętrzne właściwości obiektów, a relacje zachodzące między nimi. Jeżeli w relacji zachodzącej między obiektami ważne jest jej skierowanie, mówimy o grafie skierowanym, a krawędź reprezentującą daną relację nazywamy łukiem. Jeśli skierowanie relacji nie ma znaczenia, mówimy o grafie nieskierowanym, a jego krawędzie nazywamy gałęziami.

Podgrafem danego grafu nazywamy taki graf, w którym występują tylko niektóre z wierzchołków danego grafu, wraz z krawędziami łączącymi te wierzchołki.

Grafem częściowym danego grafu nazywamy taki graf, w którym występują tylko niektóre krawędzie danego grafu.

Krawędzie (wierzchołki) nazywamy przyległymi, jeśli mają wspólny wierzchołek (krawędź) i są różne. Krawędź, która łączy się z wierzchołkiem, nazywa się incydentną z tym wierzchołkiem.

Drogą nazywa się taki ciąg łuków, że koniec jednego jest początkiem drugiego. Zamknięta, skończona droga, której początek pokrywa się z końcem, nazywa się konturem. Długość drogi jest określona liczbą łuków wchodzących w jej skład. Kontur o długości równej jeden nazywa się pętlą.

Łłańcuchem nazywa się ciąg gałęzi takich, że jeden z granicznych wierzchołków jednej gałęzi jest granicznym wierzchołkiem kolejnej gałęzi. Skończony łańcuch, w którym rozpoczynając jego przejście od pewnego wierzchołka kończymy go w tym samym, nazywa się cyklem.

Jeżeli dowolne wierzchołki grafu można połączyć cyklem, to graf taki nazywa się grafem spójnym, a jeżeli można je połączyć drogą, to grafem silnie spójnym. Spójne podgrafy danego grafu są jego składowymi spójnościami.

Graf spójny bez konturów, który posiada wierzchołek (korzeń), do którego nie dochodzi żadna droga, zaś do każdego z pozostałych wierzchołków dochodzi jedna i tylko jedna droga, nazywa się prądrzewem (dendrytem).

Jeśli w grafie nie rozpatrujemy orientacji, to drzewem nazywa się graf spójny bez cykli, posiadający przynajmniej dwa wierzchołki. W grafie spójnym można zawsze znaleźć graf częściowy rozpinający dany graf między dwoma zadanymi wierzchołkami.

Grafy skierowane

Są one prawdopodobnie jednym z najpopularniejszych sposobów reprezentacji wiedzy i bardzo często wykorzystywane są w innych metodach jej reprezentacji. Węzły zwykle reprezentują informacje o obiekcie, jego stanie lub przedstawiają rozwiązania danego problemu. Łuki reprezentują zwykle przy tym instrukcje bądź działania zachodzące między węzłami.

Sieci semantyczne

Sieci semantyczne reprezentują relacje pomiędzy elementami dziedziny (węzłami sieci) poprzez definiowanie łuków. Zarówno węzły, jak i łuki są etykietowane wybranymi nazwami. Węzły reprezentują zwykle obiekty fizyczne (np. osoby, zwierzęta, przedmioty) lub obiekty koncepcyjne (np. czynności, wydarzenia, pojęcia abstrakcyjne) oraz deskryptory (np. cechy charakterystyczne obiektów, dodatkowe informacje o obiektach). Łuki łączą najczęściej obiekty i ich deskryptory, reprezentując różnego rodzaju relacje. Dwie najczęściej spotykane

relacje, to: "jest" - reprezentuje przynależność do pewnej klasy lub posiadanie konkretnej cechy oraz "ma" - identyfikuje węzły, które są cechami (częściami) innych węzłów [6].

Nazwy niektórych łuków są definicyjne np. "Akcelerometr *mierzy* przyspieszenie", inne reprezentują zasady heurystyczne np. "Akcelerometr *może być* uszkodzony".

Główną zaletą sieci semantycznych jest ich elastyczność - nie ma tu praktycznie żadnych restrykcji co do liczby węzłów i łuków czy też ich etykietowania, oraz dziedziczenie cech nadrzędnej klasy obiektów przez podrzędne elementy tej klasy.

Grafy *and / or*

Grafy *and / or* (znane również jako grafy biologiczne) są podobne do grafów skierowanych, ale dzięki swojej specyfice są używane do reprezentacji procesów równoległych lub w strategiach poszukiwań. Węzły reprezentują tutaj stany występujące w procesie, a łuki sterowania lub połączenia między nimi. Zarówno łuki wychodzące (wyjścia), jak i wchodzące (wejścia) do węzła muszą być powiązane operacją sumy logicznej lub iloczynu logicznego [2].

Sieci Petri [2]

Sieci Petri reprezentują, podobnie jak grafy *and / or*, równoległe przetwarzanie informacji, a w szczególności nadają się do reprezentacji dynamicznych zmian parametrów procesów zachodzących w obiekcie. W sieci Petri występują węzły reprezentujące stany (i wartości tych stanów) oraz węzły reprezentujące przejścia między stanami. Obydwa rodzaje węzłów są etykietowane. Łuki nie są etykietowane i przedstawiają połączenia między stanami poprzez węzły przejść. Dla każdej sieci Petri definiuje się stan początkowy. Przejście między stanami może być uruchomione, gdy wagi wszystkich stanów, których łuki skierowane są do węzła przejścia, są większe lub równe jedności. Następuje wtedy przeniesienie wagi o wartości jeden od węzła startowego do węzłów wyznaczonych za pomocą łuków. Stan końcowy zostaje osiągnięty, gdy żadne z przejść nie może być uruchomione.

3. Wybrane algorytmy teorii grafów zastosowane w pracy

3.1. Wprowadzenie

Realizację programową algorytmów przeprowadzono za pomocą języka Turbo Pascal firmy Borland w wersji 6.0. Programy oparte są bezpośrednio na algorytmach zawartych w literaturze dotyczącej teorii grafów, wymienionej w bibliografii. Do tych algorytmów dodano własne procedury, realizujące operacje odczytu danych wprowadzanych przez użytkownika z klawiatury, oraz odczytu i zapisu danych z gotowego pliku z danymi. Założeniem przy tworzeniu programów była jak najprostsza obsługa programu, który służyć ma przedstawieniu niektórych algorytmów teorii grafów. Do programów dołączono również moduł realizujący operację graficznej reprezentacji wyników na ekranie komputera, skonstruowany tak, aby możliwe to było za pomocą dowolnej karty graficznej [8]. Ze względu

na charakter poglądowy programu, a także ze względu na czytelność, do prezentacji graficznej dopuszczono jedynie grafy o najwyżej dziesięciu wierzchołkach. Programowo zrealizowano następujące algorytmy teorii grafów: wyznaczanie wszystkich dróg w grafie, istnienie konturów w grafie zorientowanym, wyznaczanie konturów w grafie zorientowanym, drzewo częściowe i baza cykli grafu spójnego, las częściowy i baza cykli grafu, wyznaczanie najkrótszej drogi między dwoma wierzchołkami grafu, wyznaczanie drzewa najkrótszych połączeń (dendrytu) w grafie. Poniżej opisano szczegółowo dwa algorytmy teorii grafów, które wydają się być najbardziej reprezentatywne dla poruszanego problemu.

3.2. Algorytm najkrótszej drogi między dwoma węzłami sieci

Jest to bardzo częsty problem, spotykany w wielu zagadnieniach związanych ze strategią przeszukiwania grafu (sieci). Algorytm ma więc szerokie zastosowanie w wielu zagadnieniach.

Dane:

n - liczba wierzchołków danego grafu

$A[1..n, 1..n]$ - macierz incydencji danego grafu

a - numer wierzchołka początkowego

b - numer wierzchołka końcowego

Wyniki:

Wyniki obrazowane są graficznie jako droga wykreślona pomiędzy dwoma wybranymi wierzchołkami grafu.

Dodatkowo podawana jest tablica, zawierająca numery węzłów tworzących tę drogę oraz długość tej drogi.

Użyta metoda:

Każdemu wierzchołkowi x przypisuje się indeks równy długości z a do x . W tym celu:

1. Wierzchołek a oznaczamy indeksem 0.
2. Jeżeli wszystkie wierzchołki oznaczone indeksem m tworzą zbiór $X(m)$ znany, to oznaczamy indeksem $m + 1$ zbiór wierzchołków $X(m + 1)$ taki, że x należy do zbioru $X(m)$ i x nie należy do zbioru $X(k)$, dla $k \leq m$.
3. Jeżeli wierzchołek b zostaje oznaczony, to zatrzymujemy się. Jeżeli b należy do zbioru $X(m)$, to rozważmy wierzchołki b_1, b_2, \dots takie, że:

b_1 należy do $X(n - 1)$ i istnieje łuk b_1, b

b_2 należy do $X(n - 1)$ i istnieje łuk b_2, b

.....

$a = b_n$ należy do $X(0)$ i istnieje łuk $b_n - 1, b_n$

Droga $\{ a = b_n, b_n - 1, \dots, b_1, b \}$ daje rozwiązanie.

3.3. Algorytm drzewa najkrótszych połączeń

Algorytm ten wyznacza minimalne drzewo częściowe, tj. drzewo częściowe o najmniejszej sumie długości krawędzi (tzw. dendryt).

Dane:

n - liczba wierzchołków danego grafu

$A[1..n, 1..n]$ - macierz incydencji danego grafu

M - największa liczba typu 'real' (długość nieistniejącej krawędzi)

s - korzeń minimalnego drzewa częściowego

Wyniki:

Wyniki przedstawiane są graficznie, tj. rysowany jest dendryt. Oprócz tego wypisywana jest tablica zawierająca długości krawędzi tego dendrytu. W przypadku grafów niespójnych, jako wynik otrzymujemy las częściowy.

Użyta metoda:

Algorytm ten jest realizacją metody Dijkstry.

Niech każdemu węzłowi będzie przyporządkowana para etykiet q_i, p_i , gdzie $i = (1, 2, \dots, n)$.

1. $q_i = \mu$, $p_i = 0$ ($i = 1, 2, \dots, n$), $X_0 = X \setminus \{s\}$, $r = s$.

Wykonać $n - 1$ razy kroki 2 i 3:

2. Dla każdego i należącego do X_0 : jeśli $q_i > d(r, i)$, to

$q_i = d(r, i)$ i $p_i = r$.

3. Niech $q_t = \min q_i$ (i należy do X_0), wtedy $X_0 = X_0 \setminus \{t\}$ i $r = t$.

3.4. Inne użyteczne algorytmy teorii grafów

Strategia wnioskowania w przód

W tego rodzaju przeszukiwaniach poruszamy się od faktów (symptomów), w kierunku konkluzji (diagnozy). W przypadku systemów opartych o regułę wnioskowania *jeżeli...to...* wnioskowanie w przód polega na kojarzeniu warunku z posiadanymi faktami i poszerzaniu zbioru znanych faktów przez dołączanie do nich kolejnych konkluzji [4]. Tego typu strategia jest bardzo łatwa do zaprogramowania i jest często stosowana przy reprezentacji za pomocą grafu *and / or*. Przykład przedstawiono na rysunku 1.

Fakty: a, b, c, d, e

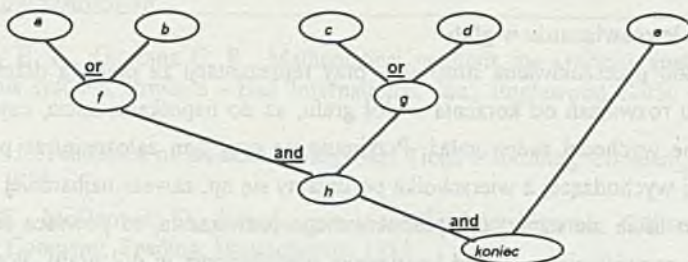
Reguły:

jeżeli a lub b to f

jeżeli c lub d to g

jeżeli f i g to h

jeżeli e i h to koniec



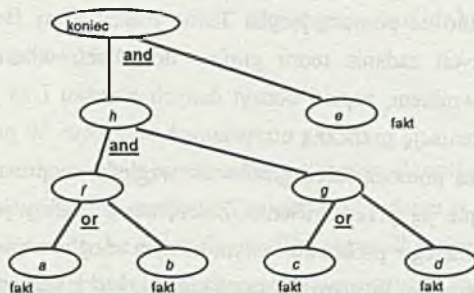
Rys. 1. Przykład rozumowania w przód

Fig. 1. Example of forward chaining

Strategia wnioskowania w tył

Jest to strategia wnioskowania przeciwna do wyżej opisanej. Stosowana jest również bardzo często przy reprezentacji wiedzy za pomocą grafu *and* / *or*. W przypadku tego rodzaju strategii zaczyna się od hipotezy i poszukuje faktów, które mogłyby ją potwierdzić lub obalić. Rozumowanie wstecz zwykle programowane jest jako rekursja, a jego główną zaletą jest większa naturalność [4]. Przykład przedstawiono na rysunku 2.

Hipoteza: rozwiązaniem jest *koniec*



Rys. 2. Przykład rozumowania wstecz

Fig. 2. Example of back chaining

Poszukiwanie rozwiązania wszerek

Jest to sposób przeszukiwania stosowany przy reprezentacji za pomocą drzewa. Oparty jest na przeglądaniu rozwiązań poprzez przechodzenie po kolei przez kolejne warstwy węzłów (węzły leżące w jednakowej odległości od korzenia), tj. na sprawdzaniu wszystkich odgałęzień danego wierzchołka. Jeżeli nie odnaleziono rozwiązania, to przechodzi się do najbliższego wierzchołka leżącego w tej samej odległości od korzenia, co dany i tam kontynuuje przeszukiwanie. Jeżeli przeszukane zostały wszystkie wierzchołki danej warstwy, a rozwiązania nie odnaleziono, to przechodzi się np. do najbardziej skrajnie lewego wierzchołka nowej warstwy i stamtąd prowadzi dalsze przeszukiwanie grafu, dopóty dopóki nie zostanie odnalezione poszukiwane rozwiązanie [3].

Poszukiwanie rozwiązania w głąb

Jest to sposób przeszukiwania stosowany przy reprezentacji za pomocą drzewa. Polega na przeglądaniu rozwiązań od korzenia w dół grafu, aż do napotkania liścia, czyli wierzchołka, z którego nie wychodzi żadna gałąź. Przyjmuje się przy tym założenie, że przy więcej niż jednej gałęzi wychodzącej z wierzchołka poruszamy się np. zawsze najbardziej w lewo. Jeżeli w drodze do liścia nie napotkano poszukiwanego rozwiązania, to powraca się od liścia do najbliższego rozwidlenia i stamtąd kontynuuje poszukiwania w dół grafu. Postępując w ten sposób natrafiamy w końcu na rozwiązanie [3].

4. Podsumowanie

Tematem pracy są wybrane algorytmy teorii grafów stosowane w metodach inżynierii wiedzy. Praca przedstawia również podstawowe pojęcia teorii grafów, przegląd aktualnie stosowanych metod reprezentacji wiedzy oraz niektóre zastosowania teorii grafów w metodach sztucznej inteligencji. Dokonano realizacji praktycznej za pomocą typowych algorytmów teorii grafów mających szerokie zastosowanie w metodach reprezentacji wiedzy, a także w innych dziedzinach. Algorytmy te zaczerpnięto z literatury podanej w bibliografii, a ich realizacji dokonano za pomocą języka Turbo Pascal firmy Borland w wersji 6.0. Do algorytmów realizujących zadania teorii grafów dołączono własne procedury realizujące komunikację z użytkownikiem, zapis i odczyt danych z dysku i na dysk, a także procedury przedstawiające reprezentację graficzną otrzymanych wyników. W pracy zdecydowano się na reprezentację wiedzy za pomocą teorii grafów ze względu na prostotę reprezentacji wiedzy i łatwe, wręcz intuicyjne jej przedstawienie. Zaletą teorii grafów jest możliwość szybkiego zrozumienia przedstawianego problemu, a tym samym możliwość odnalezienia rozwiązania w krótszym czasie. Praca jest ilustrowana przykładami, które częściowo zaczerpnięte zostały z literatury wymienionej w bibliografii. W opogramowaniu położono nacisk na charakter dydaktyczny (wyniki działania algorytmów przedstawiane są graficznie, program posiada procedury realizujące szeroką komunikację z użytkownikiem), co pozwala na jego zastosowanie w laboratorium. Ponieważ budowa programu ma charakter otwarty, możliwe jest dołączanie kolejnych użytecznych algorytmów teorii grafów (na przykład wymienionych w punkcie 3.4.). Możliwe jest również dalsze ulepszanie komunikacji z użytkownikiem, np. zastosowanie animacji graficznej.

LITERATURA

- [1] Deo N., Teoria grafów i jej zastosowania w technice i informatyce, PWN, Warszawa 1980.

- [2] Dougherty E. R., Giardina C. R., *Mathematical methods for artificial intelligence and autonomous systems*, Prentice - Hall International, Inc., Englewood Cliffs, New Jersey 1988.
- [3] Nilsson N. J., *Principles of artificial intelligence*, Tioga Publishing Company, Palo Alto, California 1985.
- [4] Charniak E., McDermott D., *Introduction to artificial intelligence*, Addison - Wesley Publishing Company, Reading, Massachusetts 1985.
- [5] Świerniak A., *Podjęmowanie decyzji w sytuacjach konfliktowych. Podstawy teoretyczne*, Politechnika Śląska, skrypt uczelniany nr 1420, Gliwice 1988.
- [6] Chwiałkowska E., *Sztuczna inteligencja w systemach eksperckich*, Z.N.I. "Mikom", Warszawa 1991.
- [7] Kucharczyk J., Sysło M., *Algorytmy optymalizacji w języku ALGOL 60*, PWN, Warszawa 1975.
- [8] Brandys W., *Przebadac wybrane algorytmy teorii grafow pod katem ich zastosowania do reprezentacji wiedzy*, praca dyplomowa, Instytut Automatyki Politechniki Śląskiej, Gliwice 1993.
- [9] A. Chabiński "Odkrywanie mózgu", *Chip* 5 (1993), pp.16 - 19.

Recenzent: Prof. dr hab. inż. Bernard Baron

Wpłynęło do Redakcji do 30.04.1994 r.

Abstract

The main objective of this work is to present some chosen algorithms of the graph theory being used in knowledge engineering methods. The aim of the paper is to discuss some yields of knowledge representation using the graph theory. The paper illustrates also fundamentals of the graph theory, review of actual knowledge representation methods and some applications of the graph theory being used in Artificial Intelligence methods. The practical implementation has been made with typical graph theory algorithms being often used at knowledge representation methods as well as in the other areas. Basing on references included in bibliography they have been realised use of the simple program written in Borland Turbo Pascal language version 6.0. Algorithms that realise the graph theory tasks have been connected with original procedures which perform communication with user, read and write operation from and to disc, and graphic procedures illustrating results on the screen. The principle of the work was not complementary review of all graph theory algorithms, but only a chosen group of them, mainly because of broad range of the subject. The paper has been illustrated by many examples, which partly are original. In the work main issue has been directed to its educational character and its use in students' laboratory. The open structure of the program allows to its build up, that means including new algorithms and better communication with user.