



Wrocław University of Technology

Sterowanie wnioskowaniem w ES

Halina Kwaśnicka

Katedra Inteligencji Obliczeniowej

Politechnika Wrocławska

Halina.kwasnicka@pwr.edu.pl

Strategie sterowania (wg. Nilsson'a)

1. Czy system produkcji jest deterministyczny?
W ogólności ***program jest niedeterministyczny, kiedy dopuszcza więcej niż jedno obliczenie*** (wnioskowanie kilkoma ścieżkami do celu równocześnie)
2. Jak porządek, w którym reguły i fakty są uporządkowane w bazie wiedzy, testowane i używane, wpływa na wyniki i efektywność algorytmu?
 - Nilsson, 1980, strategie sterowania:
 - **'Irrevocable' (nieodwołalna, bez pamięci)** - wybiera i stosuje 'stosowalne' reguły bez zabezpieczenia przed ponownym rozpatrzeniem w późniejszym punkcie sekwencji wnioskowania
 - **'Tentative' (czasowa)** - wybiera i stosuje reguły, ale zabezpiecza przed ponownym ich rozpatrywaniem (np. backtracking w PROLOGu)

Wnioskowanie - przeszukiwanie

- System produkcji z FCh, BCh lub kombinowanym paradygmatem wnioskowania - maszyna wnioskująca szuka jednej lub więcej ścieżek do celu
- Wnioskowanie można traktować jak *proces poszukiwań w przestrzeni stanów problemu*
- IE bada *dużą liczbę redundancyjnych, nie dających sukcesu ścieżek w tej przestrzeni*

Zależność od kolejności reguł

Systemy produkcji:

- **commutative** (zamienne, niezależne od kolejności reguł)
- **noncommutative** (niezamienne, zależne od kolejności reguł)
- Reguła stosowalna w kontekście bazy danych (faktów) D - kto pamięta?

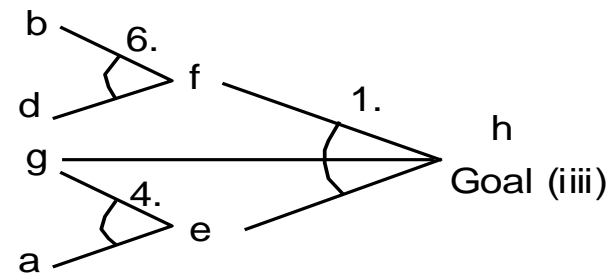
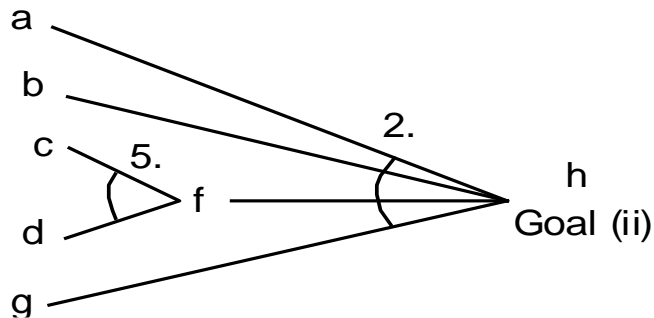
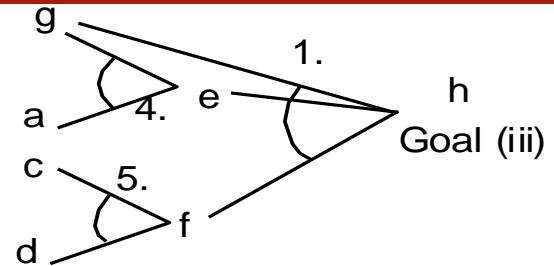
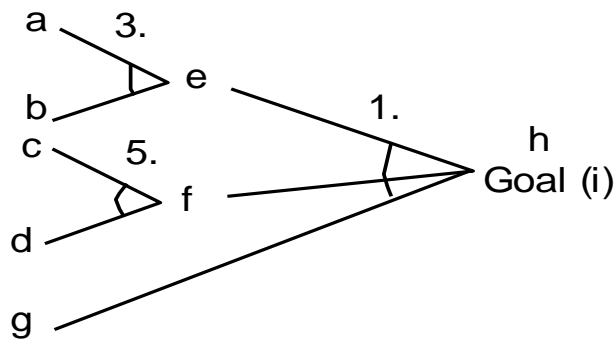
Trzy własności systemów "zamiennych" - za Nilsson'em:

1. Każda reguła stosowalna w D , jest też stosowalna w D' osiągniętej z D przez stosowanie stosowalnych reguł
2. Jeśli cel jest osiągalny w D , to jest też osiągalny w każdej D' produkowanej przez stosowalne reguły w D
3. D' generowana przez zapalone stosowalne reguły w D jest niezmienna dla permutacji w zapalonej sekwencji

Przy 'zamiennych' systemach wystarczy:

- stosować jedną permutację stosowalnych reguł
- stosować 'irrevocable' strategię (nieodwołalna, bez pamięci)
- Przykład
- rules
 - r1: h :- e, f, g. r2: h :- a, b, f, g. r3: e :- a, b.
 - r4: e :- g, a. r5: f :- c, d. r6: f :- b, d.
- facts: a. b. c. d. g. goal: h?
- Stosowalne reguły: r3,r4,r5,r6

Alternatywne wnioskowania



Możliwe sekwencje wnioskowań:

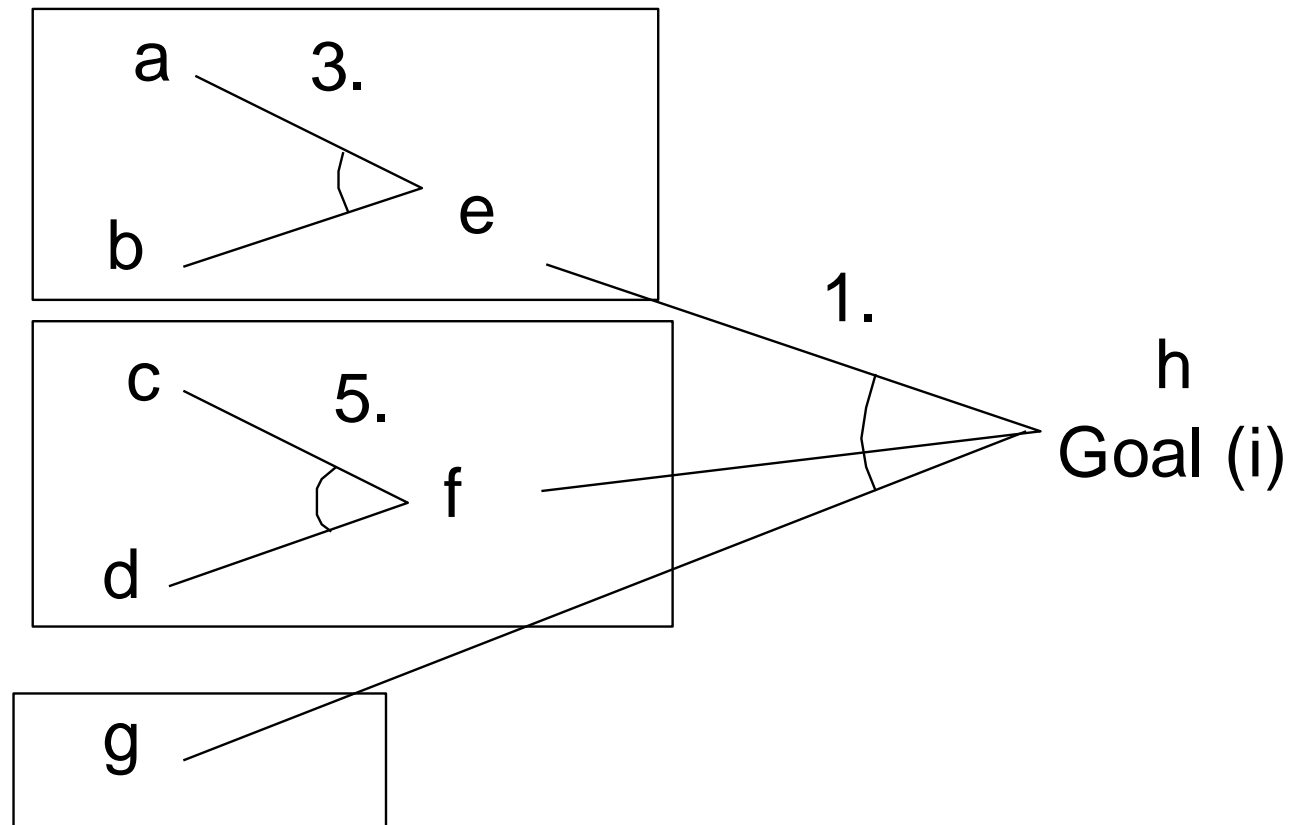
$r3, r5, r1; r5, r2; r4, r5, r1; r6, r4, r1$

Równoważne: $r5, r3, r1; r5, r2; r5, r4, r1; r4, r6, r1$

Dekomponowalność systemów produkcji (SP)

- SP można zdekomponować, gdy jego baza danych i cel są dekomponowalne
- Baza danych jest dekomponowalna, jeśli można ją podzielić na rozłączne zbiory, które mogą być niezależnie przetwarzane
- cel jest dekomponowalny na podcele, jeśli można je wnioskować w oparciu o częściowe bazy danych (podsystemy)
- Cel jest spełniany przez pewną logiczną funkcję podceli, z których każdy może być rezultatem niezależnego procesu wnioskowania w podzielonej bazie danych

Przykład dekompozycji systemu produkcji



‘Conflict resolution’ w SP - występowanie problemu

Wnioskowanie w skrócie:

1. Znajdź wszystkie stosowalne reguły
2. Z tego zbioru wybierz „dobrą regułę” lub podzbiór „dobrych reguł” do zapalenia
3. Zapal tę regułę (lub podzbiór dobrych reguł) i przejdź do p.1.

Pytanie: co to jest ‘dobra reguła’?

*Conflict resolution is thus the choosing of
"good" rules*

Dobre reguły - przykładowe kryteria

Def: Reguła A jest bardziej **specjalistyczna** niż B, jeśli poprzedniki A są nadzbiorem poprzedników B

Wybierz regułę:

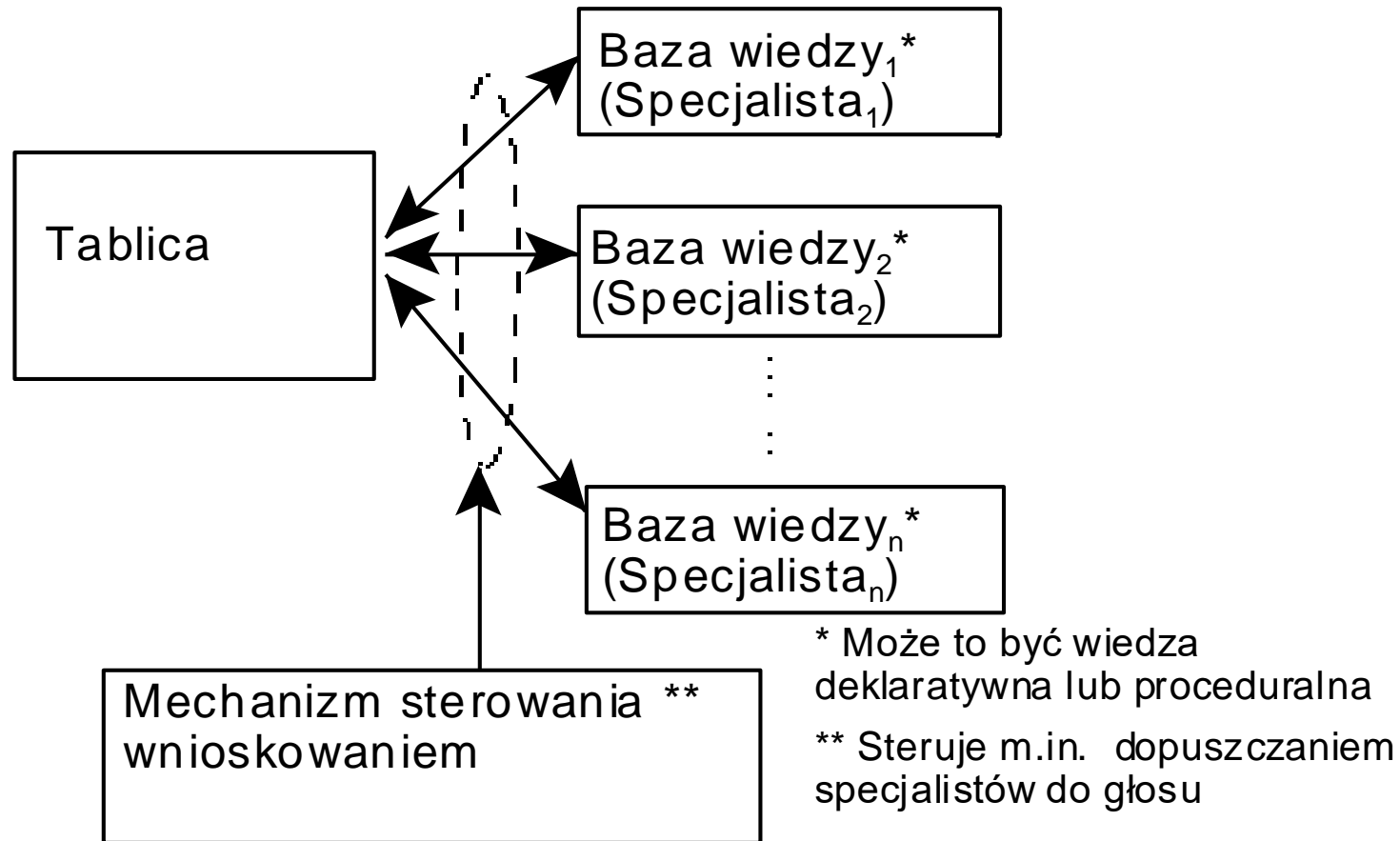
1. bardziej specjalistyczną w aktualnej sytuacji (specificity-ordered conflict resolution)
2. której zapalenie przesuwa bazę faktów (stan systemu) najbliżej do celu (wymagana miara odległości od stanu celowego)
3. o największej liczbie konsekwencji (założenie 'more is better')
4. wg 'a priori' ustawionym porządkiem
5. najczęściej ostatnio używaną
6. najrzadziej ostatnio używaną
7. z najmniejszą (największą) liczbą zmiennych



Wrocław University of Technology

Systemy Typu Tablica
(*Blackboard systems*)

System typu tablica - definicja



skupianie uwagi na tej części wiedzy, która jest ważna dla rozwiązywanego problemu

Rola przewodniczącego (moderatora)

- która wiedza źródłowa jest aktualnie aktywna (przy „tablicy”)
- jak długo dany specjalista ‘jest przy głosie’
- jakie wartości wag powinny być przypisane do wyrażeń produkowanych przez każdego specjalistę

System sterujący korzysta z metawiedzy

Sekwencja działania w STT

1. postawienie problemu (na tablicy pojawia się problem do rozwiązania)
2. specjaliści wskazują mechanizmowi sterującemu ich potencjalny wkład w problem, bazując na aktualnym stanie problemu
3. mechanizm sterujący określa następnego 'eksperta' Jeśli o 'dojście do tablicy' prosi więcej specjalistów, to mechanizm sterujący określa ich kolejność
4. Specjaliści kolejno zmieniają zawartość tablicy wnioskując mając dostępną całą wiedzę na tablicy
5. Mechanizm sterujący określa, czy problem jest rozwiązany (lub jest nierozwiązywalny), jeśli tak, to p.6. Jeśli nie, to wraca do p.1.
6. system zgłasza rozwiązanie i gotowość rozwiązywania następnego problemu

Weryfikacja bazy wiedzy - kompletność i zgodność

- **REDUNDANCY** (nadmiarowość)

dwie reguły są równoważne - spełniają te same warunki i produkują te same konkluzje: $p(X) \wedge q(X) \rightarrow r(X)$ i $p(Y) \wedge q(Y) \rightarrow r(Y)$

- **CONFLICTS** (sprzeczność)

dwie lub więcej reguł jest stosowalnych, lecz produkują sprzeczne konkluzje: $p(X) \rightarrow r(X)$ i $p(Y) \rightarrow \neg r(Y)$

- **SUBSUMED or SUBORDINATE RULES** (podporządkowane reguły)

dwie lub więcej reguł produkują te same konsekwencje, lecz jedna zawiera dodatkowe warunki: $p(X) \wedge q(X) \rightarrow r(X)$ i $q(Y) \rightarrow r(Y)$ (kto odniesie to do 'conflict resolution'?)

Weryfikacja bazy wiedzy - kompletność i zgodność, cd.

- **UNNECESSARY CONDITIONS** (zbędne warunki)
dwie reguły: $p(X) \cap q(Y) \rightarrow r(Z)$ i $p(X) \cap \neg q(Y) \rightarrow r(Z)$, wynik:
 $p(X) \rightarrow r(Z)$
- **UNRECHABLE CONDITION** (nieosiągalne warunki)
istniejące reguły nigdy nie mogą być zapalone, bo: (1)nie ma w bazie odpowiedniego podzbioru faktów, (2)nie ma reguł produkujących potrzebne fakty
- **CIRCULAR RULES** (zapętłone reguły)
sieć wnioskowania ma cykl: $p(X) \rightarrow q(X)$; $q(Y) \rightarrow r(Y)$;
 $r(Z) \rightarrow p(Z)$