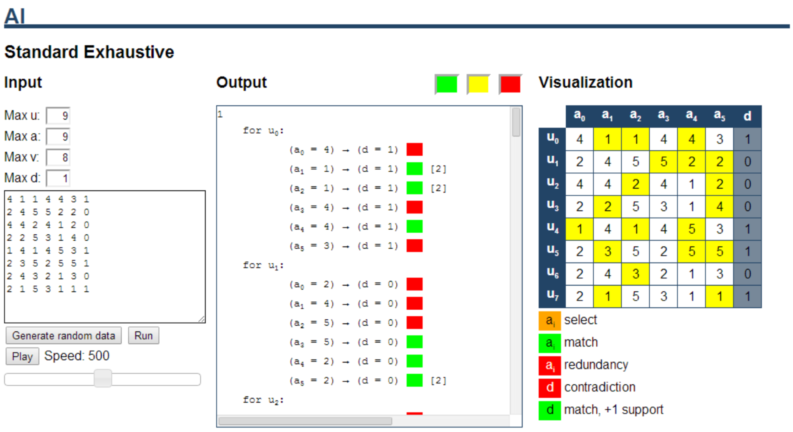
Ćwiczenie 1

Algorytmy wyliczania reguł decyzyjnych



Zadanie do wykonania

1) Tworzymy na pulpicie katalog w formacie Imię_nazwisko, w którym umieszczamy wszystkie pliki związane z ćwiczeniem.

2) Czytamy teorię, analizujemy przykłady, w razie problemu ze zrozumieniem wyliczamy reguły na kartce.

3) Generujemy system decyzyjny za pomocą programu ds_generator.exe.

4) Implementujemy algorytmy w wybranym języku programowania, wyliczając następujące reguły:

• reguły pokrywające obiekty (covering),

• reguły wyczerpujące (exhaustive) metodą bazującą na macierzy nieodróżnialności,

• reguły LEM2.

5) Przykładowy program demonstracyjny, ułatwiający pracę, w języku C++, znajdziemy na stronie http://wmii.uwm.edu.pl/~artem w zakładce Dydaktyka/Sztuczna Inteligencja.

Teoria do ćwiczeń z przykładami

Sposoby zapisu deskryptora

$(a = a(v))$
$(a = v)$
$(a, a(v))$
(a, v)
Znaczenie: $(a = a(v))$, atrybut a ma wartość v

System Informacyjny: (U, A) U - zbiór obiektów;
A - zbiór atrybutów warunkowych;

Przykład: (U, A), $U = \{ob_1, ob_2, ob_3\}$, $A = \{a_1, a_2, a_3\}$

	a_1	a_2	a_3
ob_1	1	2	3
ob_2	3	2	5
ob_3	10	2	17

System Decyzyjny: (U, A, d) U - zbiór obiektów;
A - zbiór atrybutów warunkowych;
d - atrybut decyzyjny
$d \notin A$

Przykład System decyzyjny zapisujemy jako (U, A, d), przyjmijmy,
$U = \{ob_1, ob_2, ob_3\}$
$A = \{a_1, a_2, a_3\}$
$d \in D = \{1, 2\}$
Przykładowy system decyzyjny zgodny z opisem powyżej, może wygladać następująco,

	a_1	a_2	a_3	d
ob_1	7	2	3	1
ob_2	3	3	5	2
ob_3	10	45	4	1

Zdefiniujmy reguły decyzyjne wzajemnie niesprzeczne,

$(a_1 = 1) \Rightarrow (d = 1)$

$(a_1 = 2) \wedge (a_2 = 7) \Rightarrow (d = 1)$

$(pogoda = słoneczna) \wedge (żona = w\ pracy) \wedge (czas = wolny) \Rightarrow (decyzja = park)$

$(pogoda = słoneczna) \wedge (żona = w\ domu) \wedge (czas = wolny) \Rightarrow (decyzja = dom)$

Reguły decyzyjne wzajemnie sprzeczne

$(a_1 = 1) \Rightarrow (d = 1)$
$(a_1 = 1) \Rightarrow (d = 2)$

$(pogoda = słoneczna) \wedge (żona = w\ pracy) \wedge (czas = wolny) \Rightarrow (decyzja = park)$
$(pogoda = słoneczna) \wedge (żona = w\ pracy) \wedge (czas = wolny) \Rightarrow (decyzja = dom)$

Reguła z alternatywną decyzją

$(pogoda = słoneczna) \wedge (żona = w\ pracy) \wedge (czas = wolny) \Rightarrow (decyzja = park) \vee (decyzja = dom)$

Algorytm z rodziny sekwencyjnie pokrywających (sequential covering)

Idea algorytmu pokrywającego obiekty

Szukamy w obiektach systemu decyzyjnego, począwszy od pierwszego, a skończywszy na ostatnim reguł długości jeden, które są niesprzeczne. Po znalezieniu reguły niesprzecznej, dany obiekt wyrzucamy z rozważań, pamiętając o tym, że dalej bierze udział w sprawdzaniu sprzeczności i może wspierać inne reguły.

Jeżeli po przeszukaniu wszystkich obiektów, pozostają obiekty nie wyrzucone z rozważań, szukamy w nich kombinacji niesprzecznej długości dwa i postępujemy analogicznie jak w przypadku reguł pierwszego rzędu. Wyszukiwanie reguł niesprzecznych jest kontynuowane do momentu wyeliminowania wszystkich obiektów niesprzecznych. Jeżeli w systemie pojawią się obiekty, które są sprzeczne na wszystkich deskryptorach, nie kreujemy z nich reguł.

Przykładowe wyliczanie reguł pokrywających obiekty:

Dany mamy system decyzyjny (U, A, d), gdzie $U = o_1, o_2, ..., o_7, o_8$, $A = a_1, a_2, ..., a_6$
d – atrybut decyzyjny

	a1	a2	a3	a4	a5	a6	d
o1	1	1	1	1	3	1	1
o2	1	1	1	1	3	2	1
o3	1	1	1	3	2	1	0
o4	1	1	1	3	3	2	1
o5	1	1	2	1	2	1	0
o6	1	1	2	1	2	2	1
o7	1	1	2	2	3	1	0
o8	1	1	2	2	4	1	1

Rząd I:

z o_1 brak
z o_2 $(a_6 = 2) \Rightarrow (d = 1)[3]$, wyrzucamy z rozważań obiekty o_2, o_4, o_6.
z o_3 brak
z o_5 brak
z o_7 brak
z o_8 $(a_5 = 4) \Rightarrow (d = 1)$, wyrzucamy z rozważań obiekt o_8.

Rząd II:

z o_1 $(a_3 = 1) \wedge (a_4 = 1) \Rightarrow (d = 1)[2]$, wyrzucamy z rozważań obiekt o_1.
z o_3 $(a_3 = 1) \wedge (a_5 = 2) \Rightarrow (d = 0)$, wyrzucamy z rozważań obiekt o_3.
z o_5 $(a_5 = 2) \wedge (a_6 = 1) \Rightarrow (d = 0)[2]$, wyrzucamy z rozważań obiekt o_5.
z o_7 $(a_3 = 2) \wedge (a_5 = 3) \Rightarrow (d = 0)$, wyrzucamy z rozważań obiekt o_7.

Reguły wyczerpujące (exhaustive) - z użyciem macierzy nieodróżnialności:

Dany mamy system decyzyjny (U, A, d), gdzie $U = o1, o2, .., o8$, $A = a_1, a_2, .., a_6$
d – atrybut decyzyjny

	a1	a2	a3	a4	a5	a6	d
o1	1	1	1	1	3	1	1
o2	1	1	1	1	3	2	1
o3	1	1	1	3	2	1	0
o4	1	1	1	3	3	2	1
o5	1	1	2	1	2	1	0
o6	1	1	2	1	2	2	1
o7	1	1	2	2	3	1	0
o8	1	1	2	2	4	1	1

Tworzymy dla niego macierz nieodróżnialności $\mu_A = [c_{ij}]_{8\times 8}$, gdzie $i, j = 1, 2, ..., 8$. Dla obiektów $x_1, \ldots, x_8$ zostawiamy w kolumnach tylko kratki o współrzędnych, które odpowiadają obiektom o innych decyzjach, gdzie

$$c_{ij} = \{a \in A : \; a(x_i) = a(x_j)\},$$

	o1	o2	o3	o4	o5	o6	o7	o8
o1			a1 a2 a3 a6		a1 a2 a4 a6		a1 a2 a5 a6	
o2			a1 a2 a3		a1 a2 a4		a1 a2 a5	
o3	a1 a2 a3 a6	a1 a2 a3		a1 a2 a3 a4		a1 a2 a5		a1 a2 a6
o4			a1 a2 a3 a4		a1 a2		a1 a2 a5	
o5	a1 a2 a4 a6	a1 a2 a4		a1 a2		a1 a2 a3 a4 a5		a1 a2 a3 a6
o6			a1 a2 a5		a1 a2 a3 a4 a5		a1 a2 a3	
o7	a1 a2 a5 a6	a1 a2 a5		a1 a2 a5		a1 a2 a3		a1 a2 a3 a4 a6
o8			a1 a2 a6		a1 a2 a3 a6		a1 a2 a3 a4 a6	

Tabela1: Macierz nieodróżnialności dla (U, A, d)

Komputerowi wystarczy tablica postaci:

	o1	o2	o3	o4	o5	o6	o7
o1							
o2							
o3	a1 a2 a3 a6	a1 a2 a3					
o4			a1 a2 a3 a4				
o5	a1 a2 a4 a6	a1 a2 a4		a1 a2			
o6			a1 a2 a5		a1 a2 a3 a4 a5		
o7	a1 a2 a5 a6	a1 a2 a5		a1 a2 a5		a1 a2 a3	
o8			a1 a2 a6		a1 a2 a3 a6		a1 a2 a3 a4 a6

Tabela2: Macierz nieodróżnialności trójkątna dla (U, A, d)

Wyliczania reguły za pomocą Tabeli 1. Rozważana Tabela, pozwala, na szybkie wyszukiwanie niesprzecznych reguł, wystarczy znaleźć deskryptor, lub grupę deskryptorów, które nie występują w danej kolumnie.

Zacznijmy od reguł I rzędu (rząd oznacza liczbę deskryptorów warunkowych, długość reguły):
dla $o2$ mamy
$\underline{(a_6 = 2) => (d = 1)}$
dla $o4$ mamy
$\underline{(a_6 = 2) => (d = 1)}$
dla $o6$ mamy
$\underline{(a_6 = 2) => (d = 1)}$
dla o8 mamy
$(a_5 = 4) => (d = 1)$
pozostałe kolumny nie generują żadnej reguły.
Trzy podkreślone reguły zapisujemy jako regułę z supportem 3 (support oznacza liczbę obiektów systemu decyzyjnego do których pasuje reguła), czyli grupa reguł rzędu I jest postaci:
$(a_6 = 2) => (d = 1)[3]$
$(a_5 = 4) => (d = 1)$
W kolejnym etapie modyfikujemy Tabelę 1 , zaznaczając deskryptory, z których nie będziemy mogli zbudować reguł wyższych rzędów. (czyli deskryptory: $(a_6 = 2), (a_5 = 4)$) Otrzymujemy:

	o1	o2	o3	o4	o5	o6	o7	o8
o1			a1 a2 a3 a6		a1 a2 a4 a6		a1 a2 a5 a6	
o2			a1 a2 a3		a1 a2 a4		a1 a2 a5	
o3	a1 a2 a3 a6	a1 a2 a3		a1 a2 a3 a4		a1 a2 a5		a1 a2 a6
o4			a1 a2 a3 a4		a1 a2		a1 a2 a5	
o5	a1 a2 a4 a6	a1 a2 a4		a1 a2		a1 a2 a3 a4 a5		a1 a2 a3 a6
o6			a1 a2 a5		a1 a2 a3 a4 a5		a1 a2 a3	
o7	a1 a2 a5 a6	a1 a2 a5		a1 a2 a5		a1 a2 a3		a1 a2 a3 a4 a6
o8			a1 a2 a6		a1 a2 a3 a6		a1 a2 a3 a4 a6	
		pomiń a6		pomiń a6		pomiń a6		pomiń a5

Tabela3

Teraz wybieramy reguły II rzędu, (szukamy kombinacji bez powtórzeń długości 2, które nie występują w danej kolumnie, pamiętając o pominięciu kombinacji zawierających reguły niższego rzędu)
z $o1$ mamy

$(a_3 = 1)$ & $(a_4 = 1) => (d = 1)$
$(a_3 = 1)$ & $(a_5 = 3) => (d = 1)$
$(a_4 = 1)$ & $(a_5 = 3) => (d = 1)$
z $o2$ mamy
$(a_3 = 1)$ & $(a_4 = 1) => (d = 1)$
$(a_3 = 1)$ & $(a_5 = 3) => (d = 1)$
$(a_4 = 1)$ & $(a_5 = 3) => (d = 1)$
z $o3$ mamy
$(a_3 = 1)$ & $(a_5 = 2) => (d = 0)$
$(a_4 = 3)$ & $(a_5 = 2) => (d = 0)$
$(a_4 = 3)$ & $(a_6 = 1) => (d = 0)$
$(a_5 = 2)$ & $(a_6 = 1) => (d = 0)$
z $o4$ mamy
$(a_3 = 1)$ & $(a_5 = 3) => (d = 1)$
$(a_4 = 3)$ & $(a_5 = 3) => (d = 1)$
z $o5$ mamy
$(a_5 = 2)$ & $(a_6 = 1) => (d = 0)$
z $o7$ mamy
$(a_3 = 2)$ & $(a_5 = 3) => (d = 0)$
$(a_4 = 2)$ & $(a_5 = 3) => (d = 0)$
Teraz reguły identyczne zapisujemy jako pojedyncze z odpowiednim supportem, otrzymujemy reguły postaci:
$(a_3 = 1)$ & $(a_4 = 1) => (d = 1)[2]$
$(a_3 = 1)$ & $(a_5 = 3) => (d = 1)[3]$
$(a_4 = 1)$ & $(a_5 = 3) => (d = 1)[2]$
$(a_3 = 1)$ & $(a_5 = 2) => (d = 0)$
$(a_4 = 3)$ & $(a_5 = 2) => (d = 0)$
$(a_4 = 3)$ & $(a_6 = 1) => (d = 0)$
$(a_5 = 2)$ & $(a_6 = 1) => (d = 0)[2]$
$(a_4 = 3)$ & $(a_5 = 3) => (d = 1)$
$(a_3 = 2)$ & $(a_5 = 3) => (d = 0)$
$(a_4 = 2)$ & $(a_5 = 3) => (d = 0)$

Następnie korzystając z Tabeli 3, wypisujemy kombinacje długości 3, które będą kandydatami na reguły:
z $o1$ mamy
$(a_1 = 1)$ & $(a_3 = 1)$ & $(a_4 = 1) => (d = 1)$
$(a_1 = 1)$ & $(a_3 = 1)$ & $(a_5 = 3) => (d = 1)$
$(a_1 = 1)$ & $(a_4 = 1)$ & $(a_5 = 3) => (d = 1)$
$(a_2 = 1)$ & $(a_3 = 1)$ & $(a_4 = 1) => (d = 1)$
$(a_2 = 1)$ & $(a_3 = 1)$ & $(a_5 = 3) => (d = 1)$
$(a_2 = 1)$ & $(4 = 1)$ & $(a_5 = 1) => (d = 1)$
$(a_3 = 1)$ & $(a_4 = 1)$ & $(a_5 = 3) => (d = 1)$
$(a_3 = 1)$ & $(a_4 = 1)$ & $(a_6 = 1) => (d = 1)$
$(a_3 = 1)$ & $(a_5 = 3)$ & $(a_6 = 1) => (d = 1)$
$(a_4 = 1)$ & $(a_5 = 3)$ & $(a_6 = 1) => (d = 1)$
z $o2$ mamy
$(a_1 = 1)$ & $(a_3 = 1)$ & $(a_4 = 1) => (d = 1)$

$(a_1 = 1)$ & $(a_3 = 1)$ & $(a_5 = 3)$ => $(d = 1)$
$(a_1 = 1)$ & $(a_4 = 1)$ & $(a_5 = 3)$ => $(d = 1)$
$(a_2 = 1)$ & $(a_3 = 1)$ & $(a_4 = 1)$ => $(d = 1)$
$(a_2 = 1)$ & $(a_3 = 1)$ & $(a_5 = 3)$ => $(d = 1)$
$(a_2 = 1)$ & $(a_4 = 1)$ & $(a_5 = 3)$ => $(d = 1)$
$(a_3 = 1)$ & $(a_4 = 1)$ & $(a_5 = 3)$ => $(d = 1)$
z o3 mamy
$(a_1 = 1)$ & $(a_3 = 1)$ & $(a_5 = 2)$ => $(d = 0)$
$(a_1 = 1)$ & $(a_4 = 3)$ & $(a_5 = 2)$ => $(d = 0)$
$(a_1 = 1)$ & $(a_4 = 3)$ & $(a_6 = 1)$ => $(d = 0)$
$(a_1 = 1)$ & $(a_5 = 2)$ & $(a_6 = 1)$ => $(d = 0)$
$(a_2 = 1)$ & $(a_3 = 1)$ & $(a_5 = 2)$ => $(d = 0)$
$(a_2 = 1)$ & $(a_4 = 3)$ & $(a_5 = 2)$ => $(d = 0)$
$(a_2 = 1)$ & $(a_4 = 3)$ & $(a_6 = 1)$ => $(d = 0)$
$(a_2 = 1)$ & $(a_5 = 2)$ & $(a_6 = 1)$ => $(d = 0)$
$(a_3 = 1)$ & $(a_4 = 3)$ & $(a_5 = 2)$ => $(d = 0)$
$(a_3 = 1)$ & $(a_4 = 3)$ & $(a_6 = 1)$ => $(d = 0)$
$(a_4 = 3)$ & $(a_5 = 2)$ & $(a_6 = 1)$ => $(d = 0)$
z $o4$ mamy
$(a_1 = 1)$ & $(a_3 = 1)$ & $(a_5 = 3)$ => $(d = 1)$
$(a_1 = 1)$ & $(a_4 = 3)$ & $(a_5 = 3)$ => $(d = 1)$
$(a_2 = 1)$ & $(a_3 = 1)$ & $(a_5 = 3)$ => $(d = 1)$
$(a_2 = 1)$ & $(a_4 = 3)$ & $(a_5 = 3)$ => $(d = 1)$
$(a_3 = 1)$ & $(a_4 = 3)$ & $(a_5 = 3)$ => $(d = 1)$
z $o5$ mamy
$(a_1 = 1)$ & $(a_5 = 2)$ & $(a_6 = 1)$ => $(d = 0)$
$(a_2 = 1)$ & $(a_5 = 2)$ & $(a_6 = 1)$ => $(d = 0)$
$(a_3 = 2)$ & $(a_4 = 1)$ & $(a_6 = 1)$ => $(d = 0)$
$(a_3 = 2)$ & $(a_5 = 2)$ & $(a_6 = 1)$ => $(d = 0)$
$(a_4 = 1)$ & $(a_5 = 2)$ & $(a_6 = 1)$ => $(d = 0)$
z o7 mamy
$(a_1 = 1)$ & $(a_3 = 2)$ & $(a_5 = 3)$ => $(d = 0)$
$(a_1 = 1)$ & $(a_4 = 2)$ & $(a_5 = 3)$ => $(d = 0)$
$(a_2 = 1)$ & $(a_3 = 2)$ & $(a_5 = 3)$ => $(d = 0)$
$(a_2 = 1)$ & $(a_4 = 2)$ & $(a_5 = 3)$ => $(d = 0)$
$(a_3 = 2)$ & $(a_4 = 2)$ & $(a_5 = 3)$ => $(d = 0)$
$(a_3 = 2)$ & $(a_5 = 3)$ & $(a_6 = 1)$ => $(d = 0)$
$(a_4 = 2)$ & $(a_5 = 3)$ & $(a_6 = 1)$ => $(d = 0)$

rozważane kombinacje na pewno nie są sprzeczne, stąd teraz eliminujemy, tych kandydatów, którzy zawierają reguły niższego rzędu: zaznaczmy podkreśleniem, zawieranie reguł niższego rzędu:

z $o1$ mamy
$(a_1 = 1)$ & $\underline{(a_3 = 1) \,\&\, (a_4 = 1)}$ => $(d = 1)$
$(a_1 = 1)$ & $\underline{(a_3 = 1) \,\&\, (a_5 = 3)}$ => $(d = 1)$
$(a_1 = 1)$ & $\underline{(a_4 = 1) \,\&\, (a_5 = 3)}$ => $(d = 1)$
$(a_2 = 1)$ & $\underline{(a_3 = 1) \,\&\, (a_4 = 1)}$ => $(d = 1)$

$(a_2 = 1)\ \&\ (a_3 = 1)\ \&\ (a_5 = 3) => (d = 1)$
$(a_2 = 1)\ \&\ (a_4 = 1)\ \&\ (a_5 = 3) => (d = 1)$
$(a_3 = 1)\ \&\ (a_4 = 1)\ \&\ (a_5 = 3) => (d = 1)$
$(a_3 = 1)\ \&\ (a_4 = 1)\ \&\ (a_6 = 1) => (d = 1)$
$(a_3 = 1)\ \&\ (a_5 = 3)\ \&\ (a_6 = 1) => (d = 1)$
$(a_4 = 1)\ \&\ (a_5 = 3)\ \&\ (a_6 = 1) => (d = 1)$
z $o2$ mamy
$(a_1 = 1)\ \&\ (a_3 = 1)\ \&\ (a_4 = 1) => (d = 1)$
$(a_1 = 1)\ \&\ (a_3 = 1)\ \&\ (a_5 = 3) => (d = 1)$
$(a_1 = 1)\ \&\ (a_4 = 1)\ \&\ (a_5 = 3) => (d = 1)$
$(a_2 = 1)\ \&\ (a_3 = 1)\ \&\ (a_4 = 1) => (d = 1)$
$(a_2 = 1)\ \&\ (a_3 = 1)\ \&\ (a_5 = 3) => (d = 1)$
$(a_2 = 1)\ \&\ (a_4 = 1)\ \&\ (a_5 = 3) => (d = 1)$
$(a_3 = 1)\ \&\ (a_4 = 1)\ \&\ (a_5 = 3) => (d = 1)$
z o3 mamy
$(a_1 = 1)\ \&\ (a_3 = 1)\ \&\ (a_5 = 2) => (d = 0)$
$(a_1 = 1)\ \&\ (a_4 = 3)\ \&\ (a_5 = 2) => (d = 0)$
$(a_1 = 1)\ \&\ (a_4 = 3)\ \&\ (a_6 = 1) => (d = 0)$
$(a_1 = 1)\ \&\ (a_5 = 2)\ \&\ (a_6 = 1) => (d = 0)$
$(a_2 = 1)\ \&\ (a_3 = 1)\ \&\ (a_5 = 2) => (d = 0)$
$(a_2 = 1)\ \&\ (a_4 = 3)\ \&\ (a_5 = 2) => (d = 0)$
$(a_2 = 1)\ \&\ (a_4 = 3)\ \&\ (a_6 = 1) => (d = 0)$
$(a_2 = 1)\ \&\ (a_5 = 2)\ \&\ (a_6 = 1) => (d = 0)$
$(a_3 = 1)\ \&\ (a_4 = 3)\ \&\ (a_5 = 2) => (d = 0)$
$(a_3 = 1)\ \&\ (a_4 = 3)\ \&\ (a_6 = 1) => (d = 0)$
$(a_4 = 3)\ \&\ (a_5 = 2)\ \&\ (a_6 = 1) => (d = 0)$
z $o4$ mamy
$(a_1 = 1)\ \&\ (a_3 = 1)\ \&\ (a_5 = 3) => (d = 1)$
$(a_1 = 1)\ \&\ (a_4 = 3)\ \&\ (a_5 = 3) => (d = 1)$
$(a_2 = 1)\ \&\ (a_3 = 1)\ \&\ (a_5 = 3) => (d = 1)$
$(a_2 = 1)\ \&\ (a_4 = 3)\ \&\ (a_5 = 3) => (d = 1)$
$(a_3 = 1)\ \&\ (a_4 = 3)\ \&\ (a_5 = 3) => (d = 1)$
z $o5$ mamy
$(a_1 = 1)\ \&\ (a_5 = 2)\ \&\ (a_6 = 1) => (d = 0)$
$(a_2 = 1)\ \&\ (a_5 = 2)\ \&\ (a_6 = 1) => (d = 0)$
$(a_3 = 2)\ \&\ (a_4 = 1)\ \&\ (a_6 = 1) => (d = 0)$ ta jest prawidłowa
$(a_3 = 2)\ \&\ (a_5 = 2)\ \&\ (a_6 = 1) => (d = 0)$
$(a_4 = 1)\ \&\ (a_5 = 2)\ \&\ (a_6 = 1) => (d = 0)$
z o7 mamy
$(a_1 = 1)\ \&\ (a_3 = 2)\ \&\ (a_5 = 3) => (d = 0)$
$(a_1 = 1)\ \&\ (a_4 = 2)\ \&\ (a_5 = 3) => (d = 0)$
$(a_2 = 1)\ \&\ (a_3 = 2)\ \&\ (a_5 = 3) => (d = 0)$
$(a_2 = 1)\ \&\ (a_4 = 2)\ \&\ (a_5 = 3) => (d = 0)$
$(a_3 = 2)\ \&\ (a_4 = 2)\ \&\ (a_5 = 3) => (d = 0)$
$(a_3 = 2)\ \&\ (a_5 = 3)\ \&\ (a_6 = 1) => (d = 0)$
$(a_4 = 2)\ \&\ (a_5 = 3)\ \&\ (a_6 = 1) => (d = 0)$

Mamy tylko jedną regułę trzeciego rzędu postaci,

$(a_3 = 2)$ & $(a_4 = 1)$ & $(a_6 = 1) => (d = 0)$

Kolejne rzędy reguł wyliczamy analogicznie. W naszym przypadku nie ma potrzeby sprawdzać, czy istnieją reguły rzędu cztery, otrzymanie tylko jednej reguły rzędu III, może być tu warunkiem stopu.
Ostatecznie nasz zbiór reguł exhaustive wyliczony zmodyfikowaną macierzą nieodróżnialności jest postaci:
I
$(a_6 = 2) => (d = 1)[3]$
$(a_5 = 4) => (d = 1)$
II
$(a_3 = 1)$ & $(a_4 = 1) => (d = 1)[2]$
$(a_3 = 1)$ & $(a_5 = 3) => (d = 1)[3]$
$(a_4 = 1)$ & $(a_5 = 3) => (d = 1)[2]$
$(a_3 = 1)$ & $(a_5 = 2) => (d = 0)$
$(a_4 = 3)$ & $(a_5 = 2) => (d = 0)$
$(a_4 = 3)$ & $(a_6 = 1) => (d = 0)$
$(a_5 = 2)$ & $(a_6 = 1) => (d = 0)[2]$
$(a_4 = 3)$ & $(a_5 = 3) => (d = 1)$
$(a_3 = 2)$ & $(a_5 = 3) => (d = 0)$
$(a_4 = 2)$ & $(a_5 = 3) => (d = 0)$
III
$(a_3 = 2)$ & $(a_4 = 1)$ & $(a_6 = 1) => (d = 0)$

Przykładowe wyliczanie reguł LEM2 (Learn from Examples by Modules):

Dany mamy system decyzyjny (U, A, d), gdzie $U = o1, o2, .., o7$, $A = a_1, a_2, .., a_5$
d – atrybut decyzyjny

	a_1	a_2	a_3	a_4	a_5	d
o_1	2	6	1	2	3	1
o_2	1	1	1	3	2	1
o_3	2	1	1	2	3	1
o_4	4	1	3	1	2	1
o_5	3	5	2	1	3	2
o_6	3	1	3	1	1	2
o_7	1	1	1	3	1	2

Idea algorytmu
Algorytm polega na tworzeniu pierwszej reguły przez sekwencyjny wybór "najlepszego" elementarnego warunku, przy zachowaniu ustalonych kryteriów. Przykłady treningowe pokryte przez regułę są usuwane z rozważań. Proces tworzenia reguł jest powtarzany iteracyjnie do momentu, gdy pozostają jakieś niepokryte obiekty w systemie treningowym.

Wszelkie konflikty rozwiązywane są hierarchicznie (wybierana jest wartość pierwsza od góry z lewej strony)

W praktyce wygląda to tak:
Patrzymy na koncept 1 (koncept jest synonimem klasy decyzyjnej), szukając deskryptora, który występuje najczęściej:
W wybranym koncepcie najczęściej występuje deskryptor

$(a_2 = 1) \rightharpoondown$ powstaje z obiektów o_2, o_3, o_4

Nie tworzy jednak reguły ponieważ w koncepcie 2 mamy sprzeczność.
Skupiając uwagę na obiektach do których pasuje $(a_2 = 1)$, czyli o_2, o_3, o_4, szukam kolejnego najlepszego deskryptora, z największym pokryciem w klasie 1. Tym deskryptorem jest $(a_3 = 1) \rightharpoondown$ powstaje z obiektów o_2, o_3, dokładam go do pierwszego deskryptora i tworzę koniunkcję:

$(a_2 = 1) \wedge (a_3 = 1)$, jednak powstała koniunkcja dalej jest sprzeczna,

Z faktu, że powyższa reguła powstała z obiektów o_2, o_3, szukam w nich kolejnego najbardziej licznego deskryptora, tym razem jest nim $(a_1 = 1) \rightharpoondown$ powstaje z obiektu o_2, dokładam znaleziony deskryptor do budowanej reguły:

$(a_2 = 1) \wedge (a_3 = 1) \wedge (a_1 = 1)$, sprzeczność nie została usunięta, stąd wybieramy kolejny deskryptor z obiektu o_2, dostajemy $(a_4 = 3)$, dołączamy do naszej reguły:

$(a_2 = 1) \wedge (a_3 = 1) \wedge (a_1 = 1) \wedge (a_4 = 3)$, koniunkcja jest wciąż sprzeczna, dodajemy do niej kolejny deskryptor postaci $(a_5 = 2)$, dostajemy:

$(a_2 = 1) \wedge (a_3 = 1) \wedge (a_1 = 1) \wedge (a_4 = 3) \wedge (a_5 = 2)$, ta kombinacja jest niesprzeczna, tworzymy z niej regułę postaci:

$(a_2 = 1) \wedge (a_3 = 1) \wedge (a_1 = 1) \wedge (a_4 = 3) \wedge (a_5 = 2) \Rightarrow (d = 1)$

W koncepcie 1 powstała już reguła z obiektu o_2, stąd przy szukaniu kolejnej skupiamy uwagę na o_1, o_3, o_4, najczęstszym deskryptorem jest $(a_1 = 2)$, który pasuje do obiektów o_1, o_3, ten deskryptor nie jest sprzeczny, stąd powstaje reguła:

$(a_1 = 2) \Rightarrow (d = 1)[2]$, reguła ma support 2, ponieważ pasuje do dwóch obiektów, o_1 i o_3.

w koncepcie 1, został nam do rozważenia tylko obiekt o_4, z którego powstaje niesprzeczna reguła:

$(a_1 = 4) \Rightarrow (d = 1)$

Następnie tworzymy reguły z konceptu 2:

Czyli rozważamy obiekty o_5, o_6, o_7. najbardziej licznym deskryptorem i pierwszym z brzegu jest $(a_1 = 3)$, do tego jest niesprzeczny, stąd tworzy regułę:

$(a_1 = 3) \Rightarrow (d = 2)[2]$, pokrywa obiekty o_5, o_6.

Ostatecznie tworzymy regułę z obiektu o_7:
Widzimy, że deskryptory:

$(a_1 = 1)$, $(a_2 = 1)$, $(a_3 = 1)$, $(a_4 = 3)$ tworzą sprzeczność, dopiero dołożenie deskryptora $(a_5 = 1)$, likwiduje sprzeczność i powstaje reguła:

$(a_1 = 1) \wedge (a_2 = 1) \wedge (a_3 = 1) \wedge (a_4 = 3) \wedge (a_5 = 1) \Rightarrow (d = 2)$

W przypadku gry sprzeczność występuje na wszystkich $card\{A\}$ deskryptorach warunkowych, tworzymy regułę, która ma alternatywne decyzje. Takie obiekty należą do brzegu systemu decyzyjnego.

Nasze reguły LEM2 ostatecznie mają postać:

rule1 $(a_2 = 1) \wedge (a_3 = 1) \wedge (a_1 = 1) \wedge (a_4 = 3) \wedge (a_5 = 2) \Rightarrow (d = 1)$
rule2 $(a_1 = 2) \Rightarrow (d = 1)[2]$
rule3 $(a_1 = 4) \Rightarrow (d = 1)$
rule4 $(a_1 = 3) \Rightarrow (d = 2)[2]$
rule5 $(a_1 = 1) \wedge (a_2 = 1) \wedge (a_3 = 1) \wedge (a_4 = 3) \wedge (a_5 = 1) \Rightarrow (d = 2)$