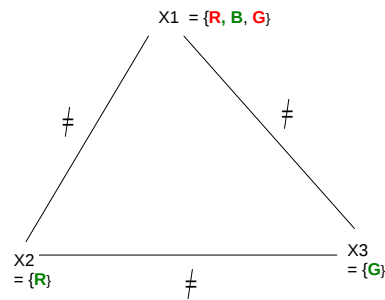


1 Ćwiczenie 3: CSP

Proszę zobaczyć folder o nazwie csp-program, aby mieć wyobrażenie o tym, jak można przedstawić i rozwiązać CSP. Czerpiąc z przykładowego programu napisz program do poniższych zadań.

1. Napisz swój program reprezentujący następujący CSP, który ma zbiór zmiennych $\{X_1, X_2, X_3\}$; na obrazku pokazana jest domena każdej zmiennej i wszystkie ograniczenia między zmiennymi.



2. Załóżmy, że przy przypisywaniu wartości do zmiennych jako pierwszy wybierany jest X_2 . Napisz program w taki sposób, aby przy wyborze kolejnej zmiennej przestrzegał zasady minimalnych pozostałych wartości (Zobacz MRV :strona 13 z wykładu 5).
3. Napisz program, aby znaleźć rozwiązanie CSP.

2 Znajdowanie reguły z systemu decyzyjnego

Poniższy link pokazuje jedną metodę sekwencyjnego algorytmu pokrywania.

Link do przeczytania: <http://cs.bc.edu/~alvarez/ML/covering.html>

Za pomocą pomysłu na napisanie programu do znajdowania reguł z systemu decyzyjnego wykonaj następujące zadanie:

2.1 Zdania do zrobienia

- (4) Generuj system decyzyjny za pomocą programu ds generator.exe. W przypadku problemów z generatorem można rozważyć dowolny z systemów decyzyjnych z pierwszego zestawu ćwiczeń. Napisz program, który znajdzie regułę z danego systemu decyzyjnego zgodnie z następującą metodą pokrywania sekwencyjnego.

2.2 Sequential covering: Idea algorytmu pokrywającego obiekty

Shukamy w obiektach systemu decyzyjnego, poczynawszy od pierwszego, a skończywszy na ostatnim regułę 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.

Jeśli 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. Jeśli w systemie pojawiają się obiekty, które są sprzeczne na wszystkich deskryptorach, nie kreujemy z nich reguł.

Example 1.

	a_1	a_2	a_3	a_4	a_5	a_6	d
o_1	1	1	1	1	3	1	1
o_2	1	1	1	1	3	2	1
o_3	1	1	1	3	2	1	0
o_4	1	1	1	3	3	2	1
o_5	1	1	2	1	2	1	0
o_6	1	1	2	1	2	2	1
o_7	1	1	2	2	3	1	0
o_8	1	1	2	2	4	1	1

Reguły: rząd-1: o_1 brak reguły

o_2 : $(a_6 = 2) \Rightarrow (d = 1)[3]$, wyrzucamy z rozważań obiekty o_2, o_4, o_6 .

o_3 brak reguły

o_5 brak reguły

o_7 brak reguły

o_8 : $(a_5 = 4) \Rightarrow (d = 1)[1]$, wyrzucamy z rozważań obiekt o_8 .

Reguły: rząd-2: o_1 : $(a_3 = 1) \wedge (a_4 = 1) \Rightarrow (d = 1)[2]$, wyrzucamy z rozważań obiekt o_1 .

o_3 : $(a_3 = 1) \wedge (a_5 = 2) \Rightarrow (d = 0)[1]$, wyrzucamy z rozważań obiekt o_3 .

o_5 : $(a_5 = 2) \wedge (a_6 = 1) \Rightarrow (d = 0)[2]$, wyrzucamy z rozważań obiekt o_5 .

o_7 : $(a_3 = 2) \wedge (a_5 = 3) \Rightarrow (d = 0)[2]$, wyrzucamy z rozważań obiekt o_7 .