

## 1 Wstęp

Projekt wykonany w prologu

Program zawiera główne 2 predykaty jeden służy do znajdowania zredukowanych kwadratów łacińskich o zadanym wymiarze a drugi dla zadanego zredukowanego kwadratu sprawdza czy istnieje dla niego jakiś kwadrat łaciński, którego pierwszy wiersz jest taki sam jak wiersz zredukowanego kwadratu.

## 2 Znajdowanie kwadratów zredukowanych

Do znajdowania kwadratów zredukowanych służy predykat `red_matrix(L,Q)`, gdzie `L` jest listą która zawiera symbole, które pojawiają się w pierwszym wierszu i w pierwszej kolumnie uzgodnionego kwadratu łacińskiego `Q` standardowo `L=[1,...,n]`.

Opis działania algorytmu

Uzupełniamy macierz `Q` tworząc system różnych reprezentantów (`srr`) dla kolejnych wierszy macierzy `Q` ale w pierwszej kolumnie ustalamy odpowiednie elementy z listy `L`.

## 3 Szukanie kwadratów ortogonalnych do danego

Do znajdowania kwadratów zredukowanych służy predykat `ort_matrix(Q,P)` gdzie `Q` dana macierz zredukowana a `P` poszukiwana macierz ortogonalna do `Q`, której pierwszy wiersz jest równy `L`.

Opis działania algorytmu

Uzupełniamy macierz `P` tworząc system różnych reprezentantów (`srr`) dla kolejnych wierszy macierzy `P` ale od razu odrzucamy także możliwości, w których para  $(n,m)$  występuje dla różnych indeksów w macierzy `Q` i otrzymanej części macierzy `P`.

## 4 System różnych reprezentantów

Generowanie `srr` jest zaimplementowane za pomocą rekurencji to znaczy dla ciągu zbiorów  $A_1, \dots, A_n$  bierzemy dowolny  $x \in A_1$  i szukamy `srr` dla ciągu zbiorów  $A_2, \dots, A_n$  taki, że  $x$  nie jest w nim wykorzystywany. Szukamy także `srr` dla ciągu zbiorów  $A_1 - \{x\}, \dots, A_n$  i łączymy oba przypadki.

## 5 Znalezienie dwóch ortogonalnych

Do znalezienia kwadratów ortogonalnych można użyć predykatu `find_ort(L,Q,P)` gdzie `L` jest listą której symboli będą tworzone kwadraty łacińskie a `Q` i `P` są znalezionymi kwadratami łacińskimi. Jeśli nie będzie żadnych takich kwadratów to predykat zwróci `false`. Przykładowe wywołanie tego predykatu to `find_ort([1,2,3,4],Q,P)`