

Problem spełniania ograniczeń

Sprawozdanie

Marcin Polanowski

1. Wstęp

Problem spełniania ograniczeń to problem matematyczny zdefiniowany następująco:

1.1 Definicja problemu:

Zmienne: $\langle V_1, V_2, \dots, V_n \rangle$

Dziedziny: $\langle D_1, D_2, \dots, D_n \rangle$

Ograniczenia – predykaty na zmiennych np. $V_1 \neq V_2, V_3 > V_2$

Cel: Odnaleźć takie wartościowanie zmiennych, które spełniałyby wszystkie ograniczenia

1.2 Rodzaje problemu

Problem spełniania ograniczeń ma wiele odmian. My będziemy zajmować się dyskretną skończoną wersją (to znaczy taką, w której dziedziny są dyskretne i skończone). Można tu wspomnieć również o odmianie boolowskiej, która jest tożsąca z problemem spełnialności

1.3 Złożoność obliczeniowa

Pesymistyczna złożoność obliczeniowa to po prostu liczba wszystkich możliwych kombinacji czyli:
 $|D_1| * |D_2| * \dots * |D_n|$

2. Opis implementacji

2.1 Definicja pliku

Pierwszy wiersz zawiera nazwy n zmiennych oddzielone spacjami. Nazwy zmiennych są dowolnymi napisami bez spacji. Liczba zmiennych NIE jest podana i wynika z liczby napisów. Kolejne n wierszy zawiera dziedziny kolejnych zmiennych. Każdy wiersz zawiera dziedzinę jednej zmiennej, wartości oddzielone są spacjami. Elementy dziedziny są dowolnymi napisami (lub liczbami) bez spacji. Liczba elementów dziedziny NIE jest podana i wynika z liczby napisów. Pozostałe wiersze pliku zawierają ograniczenia zapisane w odwrotnej notacji polskiej (ONP). W każdym wierszu podane jest jedno ograniczenie. Liczba ograniczeń NIE jest podana i wynika bezpośrednio z liczby wierszy w pliku. Każde sformułowane ograniczenie musi wartościować się do pojedynczej wartości boolowskiej. Prawda (true) oznacza, że ograniczenie jest spełnione. Fałsz (false) oznacza, że ograniczenie nie jest spełnione.

2.1.1 Zaimplementowane operatory:

+, -, * - dodawanie, odejmowanie, mnożenie (tylko dla wartości liczbowych)

=, <> - równość, nierówność (dla dowolnych wartości)

>, < - operatory porównania wartości liczbowych

|| - wartość bezwzględna (tylko wartości liczbowe),

[] - ekstrakcja zadanego elementu z napisu (1 argument – napis, 2 argument – indeks),

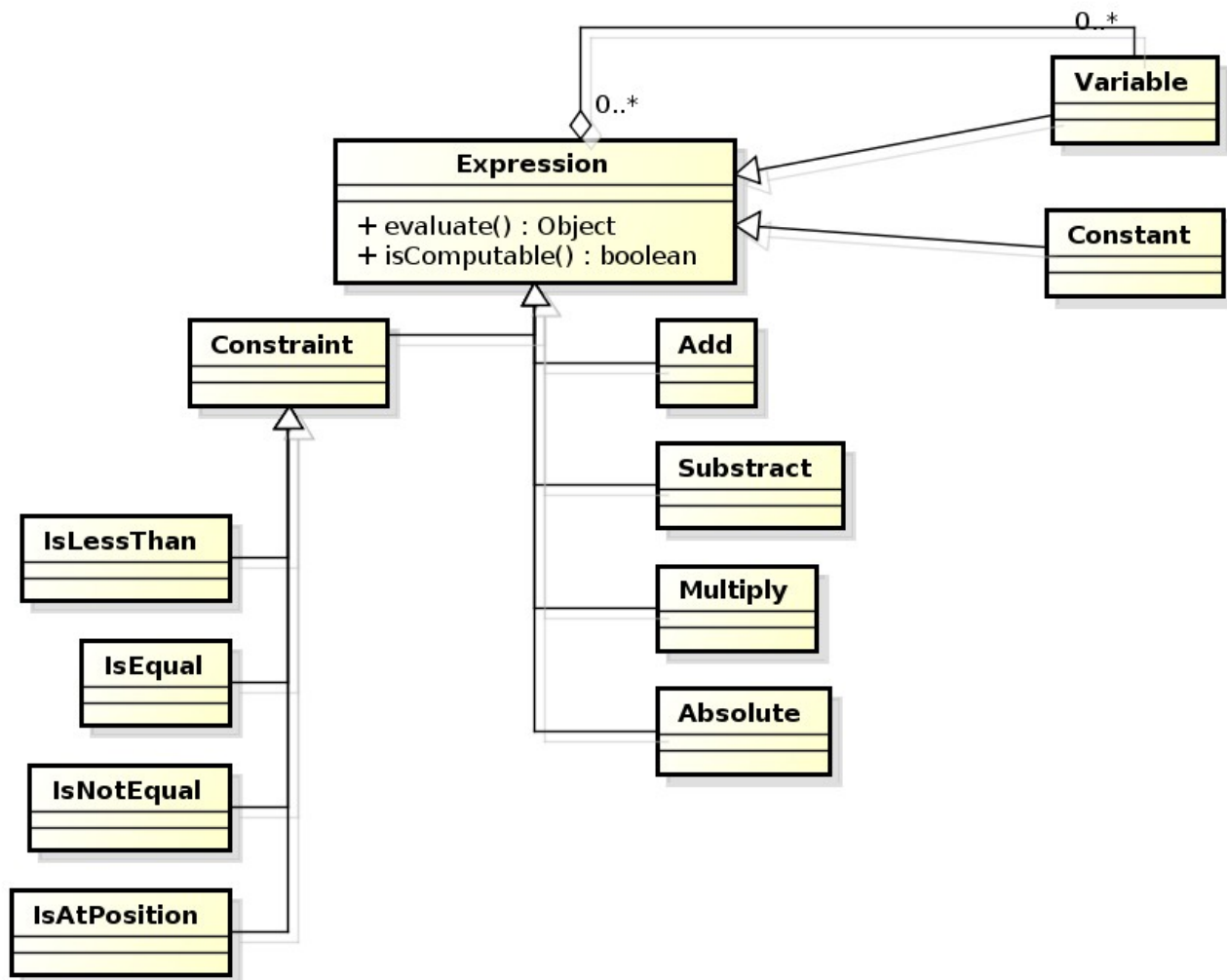
indeksowanie napisów od 0 (do zastosowania np. w krzyżówkach).

rozne - wszystkie różne, pierwszy argument – liczba n argumentów, potem n argumentów do porównania

2.2 Czytanie z pliku

Mechanizm czytania pliku jest napisany tak jak klasyczne implementacje parsera (z użyciem wzorca kompozyt). Nie będę go jednak opisywał w tym sprawozdaniu, gdyż jest to problem inżynierii oprogramowania, nie jest więc przedmiotem rozważań.

Warto jednak wspomnieć, że rozwiązanie to łączy się z reprezentacją ograniczeń oraz wyrażen matematycznych

[illegible]

a b - || 1 <>
 a c - || 2 <>
 a d - || 3 <>
 a e - || 4 <>
 a f - || 5 <>
 a g - || 6 <>
 a h - || 7 <>
 b c - || 1 <>
 b d - || 2 <>
 b e - || 3 <>
 b f - || 4 <>
 b g - || 5 <>
 b h - || 6 <>
 c d - || 1 <>
 c e - || 2 <>
 c f - || 3 <>
 c g - || 4 <>
 c h - || 5 <>
 d e - || 1 <>
 d f - || 2 <>
 d g - || 3 <>
 d h - || 4 <>
 e f - || 1 <>
 e g - || 2 <>
 e h - || 3 <>
 f g - || 1 <>
 f h - || 2 <>
 g h - || 1 <>

3.1.2 Wyniki

BackTracking:

Czas: 3855 ms

Liczba wartościowań: 24204

ForwardChecking:

Czas: 1506 ms

Liczba wartościowań: 9275

Liczba wartościowań przy propagacji w przód: 5126

Czas algorytmu ForwardChecking jest krótszy, ze względu na propagację informacji o ograniczeniach.

3.2 Sudoku

Sudoku - łamigłówka, której celem jest wypełnienie diagramu $N^2 \times N^2$ w taki sposób, aby w każdym wierszu, w każdej kolumnie i w każdym z N^2 pogrubionych kwadratów $N \times N$ (zwanych „blokami” lub „podkwadratami”) znalazło się po jednej cyfrze od 1 do N^2 .

3.1.1 Definicja pliku

(dla $N=2$)

c11 c12 c13 c14 c21 c22 c23 c24 c31 c32 c33 c34 c41 c42 c43 c44
 1 2 3 4

1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
4 c11 c12 c13 c14 rozne
4 c21 c22 c23 c24 rozne
4 c31 c32 c33 c34 rozne
4 c41 c42 c43 c44 rozne
4 c11 c21 c31 c41 rozne
4 c12 c22 c32 c42 rozne
4 c13 c23 c33 c43 rozne
4 c14 c24 c34 c44 rozne
4 c11 c12 c21 c22 rozne
4 c13 c14 c23 c24 rozne
4 c31 c32 c41 c42 rozne
4 c33 c34 c43 c44 rozne

3.2.2 Wyniki

BackTracking:

Czas: 195 ms

Liczba wartościowań: 208

ForwardChecking:

Czas: 90 ms

Liczba wartościowań: 208

Liczba wartościowań przy propagacji w przód: 1221

Czas algorytmu ForwardChecking jest krótszy mimo, że liczba głównych wartościowań jest taka sama, ze względu na uszczuplone przez propagację wprzód dziedziny

Może się jednak zdarzyć że algorytm ForwardChecking działa dłużej niż BackTracking. Dzieje się tak jeśli liczba wartościowań przy propagacji jest odpowiednio w stosunku do wartości usuniętych.

4. Perspektywy rozwoju

Do programu można dodać takie usprawnienia jak spójność łukowa, czy podział zadania ze względu na zmienne lub dziedzinę