Міністерство освіти і науки України
Дніпровський національний університет імені Олеся Гончара
Факультет прикладної математики
Кафедра комп'ютерних технологій

ЛАБОРАТОРНА РОБОТА №2

з курсу “Математичні основи інформатики”
Тема: “Автоматизація доведення програм, що містять розгалуження за допомогою ПЗ Simplify”
Варіант 25

Виконавець: Панасенко Єгор Сергійович
студент групи ПА-17-2

1. Постановка задачі

Написати специфікацію. Написати код програми.
Довести програму двома способами:

1. за допомогою теореми про розгалуження;
2. без застосування теореми про розгалуження.

у двох режимах:

1. Вручну у вигляді тексту.
2. В автоматизованому режимі за допомогою ПЗ simplify.

Індивідуальні завдання:

1. умови завдань отримати за номером у папці .\umovy
2. для кожної задачі навести доведення ”вручну” (на папері) та за допомогою Simplify.

Примітка. У варіантах завдань із номерами 04 та більше результат роботи програми - це

1. координати шахового поля - у випадку стверджувальної відповіді на запитання задачі;

2. координати шахового поля (0,0) - у випадку спростовної відповіді.

Індивідуальний варіант. Поле шахової доска визначається парою натуральних чисел, кожне з яких не перевищує восьми: перше число - номер вертикалі (при рахунку зліва направо), друге - номер горизонталі (при рахунку від низу до верху). Дано натуральні числа a, b, c, d, e, f, кожне з яких не перевищує восьми.

На поле (a, b) розташована біла фігура, на поле (c, d) - чорна. Визначити, чи може біла фігура піти на поле (e, f), не потрапивши при цьому під удар чорної фігури.

Розглянути наступні варіанти поєднань білої і чорної фігур: п) слон і кінь.

2. Хід виконання

2.1. Специфікація

Напишемо специфікацію.

Слон рухається тільки по діагоналі. Нехай маємо безкінечна дошка, у якому (x, y) - це будь яке поле на дошці описане цілими координатами $x,\ y \in \mathbb{Z}$. Нехай $(0, 0)$ - це центр дошки. Тоді рух слона з центру дошки можна описати за формулою $|x| = |y|$. Таким чином, щоб описати рух слона з поля (a, b), потрібно взяти (a, b) за центр дошки, тоді усі координати будуть $(x - a, y - b)$. Таким чином щоб перевірити, чи може рухатися слон з поля (a, b) у поле (e, f), потрібно перевірити, чи виконується рівність $|e-a| = |f-b|$. Перепишемо це мовою предикатів $abs(e-a) = abs(f-b)$ та розкриємо модулі $e-a = f-b \vee e-a = b-f$

Кінь завжди рухается на два поля вперед і на одне поле в сторону, якщо кінь рухається з центру дошки, то це можна описати так:

$$abs(x) = 1 \wedge abs(y) = 2 \vee abs(x) = 2 \wedge abs(y) = 1$$

Розкриємо модулі:

$$(x = 1 \vee x = -1) \wedge (y = 2 \vee y = -2) \vee (x = 2 \vee x = -2) \wedge (y = 1 \vee y = -1)$$

Тепер перевіримо, чи може кінь перейти з (c, d) до (e, f), взявши за центр поле (c, d):

$$\begin{aligned}(e - c = 1 \vee e - c = -1) \wedge (f - d = 2 \vee f - d = -2) \vee \\ \vee (e - c = 2 \vee e - c = -2) \wedge (f - d = 1 \vee f - d = -1)\end{aligned}$$

По задачі нам потрібно, щоб слон попав з (a, b) до (e, f), але не був під ударом коня, тобто кінь не міг попасти з (c, f) до (e, f), також слон не може попасти на (e, f), коли кінь вже стоїть на (e, f)

$$
\begin{aligned}
&\{Q: 1 \le a \wedge a \le 8 \wedge 1 \le b \wedge b \le 8 \wedge 1 \le c \wedge c \le 8\\
&\quad \wedge 1 \le d \wedge d \le 8 \wedge 1 \le e \wedge e \le 8 \wedge 1 \le f \wedge f \le 8\}\\
&\ S: \mathit{if}\\
&\qquad (e - a = f - b \vee e - a = b - f) \wedge \neg(e = c \wedge f = d) \wedge\\
&\qquad\quad \wedge \neg((e - c = 1 \vee e - c = -1) \wedge (f - d = 2 \vee f - d = -2) \vee\\
&\qquad\quad\ \ \vee (e - c = 2 \vee e - c = -2) \wedge (f - d = 1 \vee f - d = -1))\\
&\qquad\quad \rightarrow x, y := c, d\\
&\qquad |\neg((e - a = f - b \vee e - a = b - f) \wedge \neg(e = c \wedge f = d) \wedge\\
&\qquad |\quad \wedge \neg((e - c = 1 \vee e - c = -1) \wedge (f - d = 2 \vee f - d = -2) \vee\\
&\qquad |\qquad \vee (e - c = 2 \vee e - c = -2) \wedge (f - d = 1 \vee f - d = -1)))\\
&\qquad |\quad \rightarrow x, y := 0, 0\\
&\quad \mathit{fi}\\
&\{R: ((e - a = f - b \vee e - a = b - f) \wedge \neg(e = c \wedge f = d) \wedge\\
&\qquad \wedge \neg((e - c = 1 \vee e - c = -1) \wedge (f - d = 2 \vee f - d = -2) \vee\\
&\qquad\quad \vee (e - c = 2 \vee e - c = -2) \wedge (f - d = 1 \vee f - d = -1)) \wedge\\
&\qquad \wedge x = c \wedge y = d) \vee\\
&\quad\ \vee (\neg((e - a = f - b \vee e - a = b - f) \wedge \neg(e = c \wedge f = d) \wedge\\
&\qquad \wedge \neg((e - c = 1 \vee e - c = -1) \wedge (f - d = 2 \vee f - d = -2) \vee\\
&\qquad\quad \vee (e - c = 2 \vee e - c = -2) \wedge (f - d = 1 \vee f - d = -1))) \wedge\\
&\qquad \wedge x = 0 \wedge y = 0)\}
\end{aligned}
$$

Нехай

$$
\begin{aligned}
CH = &(e - a = f - b \vee e - a = b - f) \wedge \neg(e = c \wedge f = d) \wedge\\
&\quad \wedge \neg((e - c = 1 \vee e - c = -1) \wedge (f - d = 2 \vee f - d = -2) \vee\\
&\quad\ \ \vee (e - c = 2 \vee e - c = -2) \wedge (f - d = 1 \vee f - d = -1))
\end{aligned}
$$

Тоді специфікацію можна записати так:

$$
\begin{aligned}
&\{Q: 1 \le a \wedge a \le 8 \wedge 1 \le b \wedge b \le 8 \wedge 1 \le c \wedge c \le 8\\
&\quad \wedge 1 \le d \wedge d \le 8 \wedge 1 \le e \wedge e \le 8 \wedge 1 \le f \wedge f \le 8\}\\
&\ S: \mathit{if}\\
&\qquad CH \rightarrow x, y := c, d\\
&\qquad |\neg(CH) \rightarrow x, y := 0, 0\\
&\quad \mathit{fi}\\
&\{R: (CH \wedge x = c \wedge y = d) \vee (\neg(CH) \wedge x = 0 \wedge y = 0)\}
\end{aligned}
$$

2.2. Доведення

2.3. Доведення з теоремою про розгалуження

Наведемо теорему про розгалуження.

Теорема 2.1 (про розгалуження)**.** *Розглянемо команду IF. Нехай предикат Q задовільняє умовам:*

1. $Q \Rightarrow BB$
2. $Q \wedge B_i \Rightarrow wp(S_i, R)$ *для всіх i,* $1 \leq i \leq n$.

Тоді і тільки тоді $Q \Rightarrow wp(IF, R)$.

1. $Q \Rightarrow BB$

$$\begin{aligned}
&Q \Rightarrow B1 \vee B2, \text{ (за означенням IF)}\\
&Q \Rightarrow CH \vee \neg CH\\
&Q \Rightarrow TRUE, \text{ (закон виключення третього: } E1 \vee \neg E2 = T\text{)}\\
&\neg Q \vee TRUE, \text{ (закон імплікації)}\\
&TRUE, \text{ (закон спрощення } \vee\text{)}
\end{aligned}$$

Таким чином ми довели, що

$$Q \Rightarrow BB = TRUE \tag{1}$$

2. $(\forall i : 1 \leq i \leq 2 : Q \wedge Bi \Rightarrow wp(S_i, R))$

$$Q \wedge CH \Rightarrow wp(x, y := c, d, R) \wedge Q \wedge \neg CH \Rightarrow wp(x, y := 0, 0, R))$$

$$(Q \wedge CH \Rightarrow R^{x,y}_{c,d}) \wedge (Q \wedge \neg CH \Rightarrow R^{x,y}_{0,0}), \text{ (за озн. присвоювання)}$$

$$\begin{aligned}
&(Q \wedge CH \Rightarrow (CH \wedge c = c \wedge d = d) \vee (\neg(CH) \wedge c = 0 \wedge d = 0))\wedge\\
&\wedge (Q \wedge \neg CH \Rightarrow (CH \wedge 0 = c \wedge 0 = d) \vee (\neg(CH) \wedge 0 = 0 \wedge 0 = 0))
\end{aligned}$$

$$\begin{aligned}
&(Q \wedge CH \Rightarrow (CH \wedge TRUE \wedge TRUE) \vee (\neg(CH) \wedge c = 0 \wedge d = 0))\wedge\\
&\wedge (Q \wedge \neg CH \Rightarrow (CH \wedge 0 = c \wedge 0 = d) \vee (\neg(CH) \wedge TRUE \wedge TRUE))
\end{aligned}$$

$$\begin{aligned}
&(Q \wedge CH \Rightarrow CH \vee (\neg(CH) \wedge c = 0 \wedge d = 0))\wedge\\
&\wedge (Q \wedge \neg CH \Rightarrow (CH \wedge 0 = c \wedge 0 = d) \vee \neg CH)
\end{aligned}$$

$(\neg(Q \wedge CH) \vee CH \vee (\neg(CH) \wedge c = 0 \wedge d = 0)) \wedge$
$\wedge (\neg(Q \wedge \neg CH) \vee (CH \wedge 0 = c \wedge 0 = d) \vee \neg CH)$, (закон імплікації)

$(\neg Q \vee \neg CH \vee CH \vee (\neg(CH) \wedge c = 0 \wedge d = 0)) \wedge$
$\wedge (\neg Q \vee CH \vee (CH \wedge 0 = c \wedge 0 = d) \vee \neg CH)$, (закон де Моргана)

$(\neg Q \vee TRUE \vee (\neg(CH) \wedge c = 0 \wedge d = 0)) \wedge$
$\wedge (\neg Q \vee TRUE \vee (CH \wedge 0 = c \wedge 0 = d))$, (закон виключення третього)

$TRUE \wedge TRUE$, (закон спрощення $\vee$)

$TRUE$

Таким чином ми довели, що

$$(\forall i : 1 \leq i \leq 2 : Q \wedge Bi \Rightarrow wp(S_i, R)) = TRUE \qquad (2)$$

Ми довели, що для нашої задачі $Q \Rightarrow wp(S, R)$

2.4. Доведення без теореми про розгалуження

Визначення 2.1.

$$wp(IF, R) = (\exists i : 1 \leq i \leq n : B_i) \wedge (\forall i : 1 \leq i \leq n : B_i \Rightarrow wp(S_i, R))$$

Для того, щоб довести програму, потрібно довести, що $Q \Rightarrow wp(S, R)$ тавтологія.

Спочатку приведемо вираження до виду, який буде легко перевести на мову Simplify.

$Q \Rightarrow wp(IF, R)$

$Q \Rightarrow (\exists i : 1 \leq i \leq 2 : B_i) \wedge (\forall i : 1 \leq i \leq 2 : B_i \Rightarrow wp(S_i, R))$

$Q \Rightarrow (B1 \vee B2) \wedge (\forall i : 1 \leq i \leq 2 : B_i \Rightarrow wp(S_i, R))$

$Q \Rightarrow (B1 \vee B2) \wedge (B_1 \Rightarrow wp(S_1, R)) \wedge (B_2 \Rightarrow wp(S_2, R))$

$$Q \Rightarrow (CH \vee \neg CH) \wedge (CH \Rightarrow wp(x, y := c, d, R))$$
$$\wedge (\neg CH \Rightarrow wp(x, y := 0, 0, R))$$

$$Q \Rightarrow (CH \vee \neg CH) \wedge (CH \Rightarrow R^{x,y}_{c,d}) \wedge (\neg CH \Rightarrow R^{x,y}_{0,0})$$

$$Q \Rightarrow (CH \vee \neg CH) \wedge (CH \Rightarrow (CH \wedge c = c \wedge d = d) \vee (\neg(CH) \wedge c = 0 \wedge d = 0))$$
$$\wedge (\neg CH \Rightarrow (CH \wedge 0 = c \wedge 0 = d) \vee (\neg(CH) \wedge 0 = 0 \wedge 0 = 0))$$

Ми привели до простого виду, якщо підставити CH, то ми отримаємо предикат у якому тільки операції $\neg, \wedge, \vee, =, \Rightarrow$. Тепер можна доводити вручну.

$$Q \Rightarrow (CH \vee \neg CH) \wedge (CH \Rightarrow (CH \wedge TRUE \wedge TRUE) \vee (\neg(CH) \wedge c = 0 \wedge d = 0))$$
$$\wedge (\neg CH \Rightarrow (CH \wedge 0 = c \wedge 0 = d) \vee (\neg(CH) \wedge TRUE \wedge TRUE))$$

$Q \Rightarrow (CH \vee \neg CH) \wedge (CH \Rightarrow CH \vee (\neg(CH) \wedge c = 0 \wedge d = 0))$
$\wedge (\neg CH \Rightarrow (CH \wedge 0 = c \wedge 0 = d) \vee \neg CH)$, (закон спрощення $\wedge$)

$\neg Q \vee (CH \vee \neg CH) \wedge (\neg CH \vee CH \vee (\neg(CH) \wedge c = 0 \wedge d = 0))$
$\wedge (CH \vee (CH \wedge 0 = c \wedge 0 = d) \vee \neg CH)$, (закон імплікації)

$\neg Q \vee (TRUE) \wedge (TRUE \vee (\neg(CH) \wedge c = 0 \wedge d = 0))$
$\wedge (TRUE \vee (CH \wedge 0 = c \wedge 0 = d))$, (закон виключення третього)

$\neg Q \vee TRUE \wedge TRUE \wedge TRUE$, (закони спрощення $\vee$)

$\neg Q \vee TRUE$, (закони спрощення $\wedge$)

$TRUE$, (закони спрощення $\vee$)

2.5. Доведення за допомогою Simplify

Раніше ми привели предикат до зручної форми для переведення в Simplify:

$$Q \Rightarrow (CH \vee \neg CH) \wedge (CH \Rightarrow (CH \wedge c = c \wedge d = d) \vee (\neg(CH) \wedge c = 0 \wedge d = 0))$$
$$\wedge (\neg CH \Rightarrow (CH \wedge 0 = c \wedge 0 = d) \vee (\neg(CH) \wedge 0 = 0 \wedge 0 = 0))$$

Напишемо тепер мовою Simplify:

```
(IMPLIES Q (AND (OR CH (NOT CH))
  (IMPLIES CH (OR (AND CH (EQ c c) (EQ d d))
    (AND (NOT CH) (EQ c 0) (EQ d 0))))
  (IMPLIES (NOT CH) (OR (AND CH (EQ 0 c) (EQ 0 d))
    (AND (NOT CH) (EQ 0 0) (EQ 0 0))))))
```

Згадаймо чому дорівнює CH:

$$CH = (e - a = f - b \vee e - a = b - f) \wedge \neg(e = c \wedge f = d) \wedge$$
$$\wedge \neg((e - c = 1 \vee e - c = -1) \wedge (f - d = 2 \vee f - d = -2) \vee$$
$$\vee (e - c = 2 \vee e - c = -2) \wedge (f - d = 1 \vee f - d = -1))$$

Напишемо тепер CH мовою Simplify:

```
(AND (OR (EQ (- e a) (- f b)) (EQ (- e a) (- b f)))
  (NOT (AND (EQ e c) (EQ f d)))
  (NOT (OR (AND (OR (EQ (- e c) 1) (EQ (- e c) -1))
                (OR (EQ (- f d) 2) (EQ (- f d) -2)))
           (AND (OR (EQ (- e c) 2) (EQ (- e c) -2))
                (OR (EQ (- f d) 1) (EQ (- f d) -1))))))
```

Підставимо це у перший предикат:

```
(IMPLIES Q (AND (OR (AND (OR (EQ (- e a) (- f b)) (EQ (- e a) (- b f)))
(NOT (AND (EQ e c) (EQ f d))) (NOT (OR (AND (OR (EQ (- e c) 1) (EQ (- e c) -1))
(OR (EQ (- f d) 2) (EQ (- f d) -2))) (AND (OR (EQ (- e c) 2) (EQ (- e c) -2))
(OR (EQ (- f d) 1) (EQ (- f d) -1)))))) (NOT (AND (OR (EQ (- e a) (- f b))
(EQ (- e a) (- b f))) (NOT (AND (EQ e c) (EQ f d))) (NOT (OR (AND
(OR (EQ (- e c) 1) (EQ (- e c) -1)) (OR (EQ (- f d) 2) (EQ (- f d) -2)))
(AND (OR (EQ (- e c) 2) (EQ (- e c) -2)) (OR (EQ (- f d) 1) (EQ (- f d) -1))))))
   ))
(IMPLIES (AND (OR (EQ (- e a) (- f b)) (EQ (- e a) (- b f)))
(NOT (AND (EQ e c) (EQ f d))) (NOT (OR (AND (OR (EQ (- e c) 1) (EQ (- e c) -1))
(OR (EQ (- f d) 2) (EQ (- f d) -2))) (AND (OR (EQ (- e c) 2) (EQ (- e c) -2))
(OR (EQ (- f d) 1) (EQ (- f d) -1))))))
(OR (AND (AND (OR (EQ (- e a) (- f b)) (EQ (- e a) (- b f)))
(NOT (AND (EQ e c) (EQ f d))) (NOT (OR (AND (OR (EQ (- e c) 1) (EQ (- e c) -1))
(OR (EQ (- f d) 2)
(EQ (- f d) -2))) (AND (OR (EQ (- e c) 2) (EQ (- e c) -2))
(OR (EQ (- f d) 1) (EQ (- f d) -1))))))
(EQ c c) (EQ d d)) (AND (NOT (AND (OR (EQ (- e a) (- f b))
(EQ (- e a) (- b f))) (NOT (AND (EQ e c) (EQ f d)))
```

```
(NOT (OR (AND (OR (EQ (- e c) 1) (EQ (- e c) -1)) (OR (EQ (- f d) 2)
(EQ (- f d) -2))) (AND (OR (EQ (- e c) 2) (EQ (- e c) -2))
(OR (EQ (- f d) 1) (EQ (- f d) -1))))))) (EQ c 0) (EQ d 0))))
(IMPLIES (NOT (AND (OR (EQ (- e a) (- f b)) (EQ (- e a) (- b f)))
(NOT (AND (EQ e c) (EQ f d))) (NOT (OR (AND (OR (EQ (- e c) 1) (EQ (- e c) -1))
(OR (EQ (- f d) 2) (EQ (- f d) -2))) (AND (OR (EQ (- e c) 2)
(EQ (- e c) -2)) (OR (EQ (- f d) 1) (EQ (- f d) -1)))))))
(OR (AND (AND (OR (EQ (- e a) (- f b)) (EQ (- e a) (- b f)))
(NOT (AND (EQ e c) (EQ f d))) (NOT (OR (AND (OR (EQ (- e c) 1)
(EQ (- e c) -1)) (OR (EQ (- f d) 2) (EQ (- f d) -2)))
(AND (OR (EQ (- e c) 2) (EQ (- e c) -2))
(OR (EQ (- f d) 1) (EQ (- f d) -1)))))) (EQ 0 c) (EQ 0 d))
(AND (NOT (AND (OR (EQ (- e a) (- f b)) (EQ (- e a) (- b f)))
(NOT (AND (EQ e c) (EQ f d))) (NOT (OR (AND (OR (EQ (- e c) 1)
(EQ (- e c) -1)) (OR (EQ (- f d) 2) (EQ (- f d) -2)))
(AND (OR (EQ (- e c) 2) (EQ (- e c) -2)) (OR (EQ (- f d) 1)
(EQ (- f d) -1))))))) (EQ 0 0) (EQ 0 0))))))
```

Збережемо це у файл, та перевіримо це у Simplify. Отримаємо такий результат:

```
[egor@gauramid Simplify]$ ./Simplify-1.5.4.linux lab2.txt
1: Valid.
```