

INTEGRAMS (i.e. Einstein's Riddle)

(Robert Meolic, 2003, 2013)

This is a small attempt to use BDDs for solving Einstein's Riddle. As stated in Wikipedia (http://en.wikipedia.org/wiki/Zebra_Puzzle) the original name for this popular logic puzzle is Zebra Puzzle.

1. The British person lives in the red house.
2. The Swede keeps dogs as pets.
3. The Dane drinks tea.
4. The green house is on the left of the white house.
5. The green homeowner drinks coffee.
6. The man who smokes Pall Mall keeps birds.
7. The owner of the yellow house smokes Dunhill.
8. The man living in the center house drinks milk.
9. The Norwegian lives in the first house.
10. The man who smokes Blend lives next to the one who keeps cats.
11. The man who keeps the horse lives next to the man who smokes Dunhill.
12. The man who smokes Bluemaster drinks beer.
13. The German smokes Prince.
14. The Norwegian lives next to the blue house.
15. The man who smokes Blend has a neighbor who drinks water.

Question: Who owns the fish?

File **integram.tcl** is a much smaller puzzle (in Slovene), just for testing the method. This test is, however, not complex enough to be trivially used for solving Einstein's Riddle.

```
outline "INTEGRAM"
outline "1. Andrej ali Borut je iz Maribora."
outline "2. Najvecji je iz Ljubljane, najmanjsi pa ni iz Celja."
outline "3. Ce je Andrej manjsi od Cene, potem ni iz Maribora."
outline "4. Cene je vecji od tistega iz Celja."
outline ""

# Oseba: Andrej (A), Borut (B), Cene (C)
# Mesto: Celje (ce), Maribor (mb), Ljubljana (lj)
# Velikost: majhen (s), vecji (l), najvecji (x)
#
# Kodiranje:
# Oseba: a,b,c
# Mesto: Maribor *1 *0, Ljubljana *1 0, Celje: 1 *0
# Velikost: majhen *3 *2, vecji *3 2, najvecji 3 *2
#
# Zapis z enacbami:
# 1.  Amb + Bmb
# 2a. (Ax -> Alj) * (Bx -> Blj) * (Cx -> Clj)
# 2b. (As -> !Ace) * (Bs -> !Bce) * (Cs -> !Cce)
# 3.  (As * Cl + As * Cx + Al * Cx) -> !Amb
# 4.  (Ace * As + Bce * Bs) * (Cl + Cx) + (Ace * Al + Bce * Bl) * Cx
# 4.  (Ace * (As*Cl + As*Cx + Al*Cx)) + (Bce * (Bs*Cl + Bs*Cx + Bl*Cx))
# 4.  !Cce * (Ace -> (As*Cl + As*Cx + Al*Cx)) * (Bce -> (Bs*Cl + Bs*Cx + Bl*Cx))
```

outline "Racunam..."

```
bdd_expr "(AND (NOT a1) (NOT a0) (NOT b1) b0 c1 (NOT c0))" P0
bdd_expr "(OR P0 (AND (NOT a1) (NOT a0) b1 (NOT b0) (NOT c1) c0))" P0
bdd_expr "(OR P0 (AND (NOT a1) a0 (NOT b1) (NOT b0) c1 (NOT c0)))" P0
bdd_expr "(OR P0 (AND (NOT a1) a0 b1 (NOT b0) (NOT c1) (NOT c0)))" P0
bdd_expr "(OR P0 (AND a1 (NOT a0) (NOT b1) (NOT b0) (NOT c1) c0))" P0
bdd_expr "(OR P0 (AND a1 (NOT a0) (NOT b1) b0 (NOT c1) (NOT c0)))" P0
```

```
bdd_expr "(AND (NOT a3) (NOT a2) (NOT b3) b2 c3 (NOT c2))" P1
bdd_expr "(OR P1 (AND (NOT a3) (NOT a2) b3 (NOT b2) (NOT c3) c2))" P1
bdd_expr "(OR P1 (AND (NOT a3) a2 (NOT b3) (NOT b2) c3 (NOT c2)))" P1
bdd_expr "(OR P1 (AND (NOT a3) a2 b3 (NOT b2) (NOT c3) (NOT c2)))" P1
bdd_expr "(OR P1 (AND a3 (NOT a2) (NOT b3) (NOT b2) (NOT c3) c2))" P1
bdd_expr "(OR P1 (AND a3 (NOT a2) (NOT b3) b2 (NOT c3) (NOT c2)))" P1
```

bdd_expr "(AND P0 P1)" SYSTEM

bdd_expr "(OR (AND (NOT a1) (NOT a0)) (AND (NOT b1) (NOT b0)))" F1

```
bdd_expr "(OR (NOT (AND a3 (NOT a2))) (AND (NOT a1) a0))" F2a
bdd_expr "(OR (NOT (AND b3 (NOT b2))) (AND (NOT b1) b0))" F2b
bdd_expr "(OR (NOT (AND c3 (NOT c2))) (AND (NOT c1) c0))" F2c
bdd_expr "(OR (NOT (AND (NOT a3) (NOT a2))) (NOT (AND a1 (NOT a0))))" F2d
bdd_expr "(OR (NOT (AND (NOT b3) (NOT b2))) (NOT (AND b1 (NOT b0))))" F2e
bdd_expr "(OR (NOT (AND (NOT c3) (NOT c2))) (NOT (AND c1 (NOT c0))))" F2f
bdd_expr "(AND F2a F2b F2c F2d F2e F2f)" F2
```

```
bdd_expr "(AND (NOT a3) (NOT a2) (NOT c3) c2)" F3a
bdd_expr "(AND (NOT a3) (NOT a2) c3 (NOT c2))" F3b
bdd_expr "(AND (NOT a3) a2 c3 (NOT c2))" F3c
bdd_expr "(OR (NOT (OR F3a F3b F3c)) (NOT (AND (NOT a1) (NOT a0))))" F3
```

```
bdd_expr "(AND a1 (NOT a0) (NOT a3) (NOT a2))" F4a
bdd_expr "(AND b1 (NOT b0) (NOT b3) (NOT b2))" F4b
bdd_expr "(AND a1 (NOT a0) (NOT a3) a2)" F4c
bdd_expr "(AND b1 (NOT b0) (NOT b3) b2)" F4d
bdd_expr "(AND (OR F4a F4b) (OR (AND (NOT c3) c2) (AND c3 (NOT c2))))" F4e
bdd_expr "(AND (OR F4c F4d) (AND c3 (NOT c2)))" F4f
bdd_expr "(OR F4e F4f)" F4
```

```
bdd_expr "(SYSTEM)" REZULTAT
bdd_expr "(AND F1 REZULTAT)" REZULTAT
bdd_expr "(AND F2 REZULTAT)" REZULTAT
bdd_expr "(AND F3 REZULTAT)" REZULTAT
bdd_expr "(AND F4 REZULTAT)" REZULTAT
```

outline "Rezultat"

bdd_out_function REZULTAT

REZULTAT:

+ a1 *a0 *b1 *b0 *c1 c0 *a3 a2 *b3 *b2 c3 *c2

#

Resitev: Ace * Bmb * Clj * Al * Bs * Cx

Dva alternativna izracuna za lastnost 4

bdd_expr "(AND (NOT a1) (NOT a0))" Amb

bdd_expr "(AND (NOT a1) a0)" Alj

```

bdd_expr "(AND a1 (NOT a0))" Ace
bdd_expr "(AND (NOT b1) (NOT b0))" Bmb
bdd_expr "(AND (NOT b1) b0)" B1j
bdd_expr "(AND b1 (NOT b0))" Bce
bdd_expr "(AND (NOT c1) (NOT c0))" Cmb
bdd_expr "(AND (NOT c1) c0)" C1j
bdd_expr "(AND c1 (NOT c0))" Cce

bdd_expr "(AND (NOT a3) (NOT a2))" As
bdd_expr "(AND (NOT a3) a2)" A1
bdd_expr "(AND a3 (NOT a2))" Ax
bdd_expr "(AND (NOT b3) (NOT b2))" Bs
bdd_expr "(AND (NOT b3) b2)" B1
bdd_expr "(AND b3 (NOT b2))" Bx
bdd_expr "(AND (NOT c3) (NOT c2))" Cs
bdd_expr "(AND (NOT c3) c2)" C1
bdd_expr "(AND c3 (NOT c2))" Cx

outline "Alternativni izracun za lastnost 4"
bdd_expr "(AND F4 SYSTEM)" F4SYSTEM

# verzija 1

bdd_expr "(AND Ace (OR (AND C1 As) (AND Cx As) (AND Cx A1)))" G4a
bdd_expr "(AND Bce (OR (AND C1 Bs) (AND Cx Bs) (AND Cx B1)))" G4b
bdd_expr "(OR G4a G4b)" G4
bdd_expr "(AND G4 SYSTEM)" G4SYSTEM

outline "(F4 == G4): [bdd_cmp F4 G4]"
outline "(F4SYSTEM == G4SYSTEM): [bdd_cmp F4SYSTEM G4SYSTEM]"

# verzija 2

bdd_expr "(NOT Cce)" H4a
bdd_expr "(OR (NOT Ace) (OR (AND C1 As) (AND Cx As) (AND Cx A1)))" H4b
bdd_expr "(OR (NOT Bce) (OR (AND C1 Bs) (AND Cx Bs) (AND Cx B1)))" H4c
bdd_expr "(AND H4a H4b H4c)" H4
bdd_expr "(AND H4 SYSTEM)" H4SYSTEM

outline "(F4 == H4): [bdd_cmp F4 H4]"
outline "(F4SYSTEM == H4SYSTEM): [bdd_cmp F4SYSTEM H4SYSTEM]"

```

Here is the log from EST:

Efficient Symbolic Tools, 2nd Edition, Copyright (C) 2003-2013 UM-FERI
This is free software, and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute EST; type "license" for details.

Running on i686 (Linux, 3.2.0-39-generic-pae) with tcl 8.5.11 and tk 8.5.11.

```

Initialization of GUI package... OK
Initialization of BDD package... OK
Initialization of Process_Algebra package... OK
Initialization of Versis package... OK
Initialization of Model checking package... OK
Initialization of Strucval package... OK
Initialization of CCS package... OK
Ready.

```

```

>cd "/home/meolic/est/est-2ed/data/integram"; source "integram.tcl"; cd "/home/meolic/est/est-2ed/data"

```

INTEGRAM

1. Andrej ali Borut je iz Maribora.
2. Najvecji je iz Ljubljane, najmanjsi pa ni iz Celja.
3. Ce je Andrej manjsi od Ceneta, potem ni iz Maribora.
4. Cene je vecji od tistega iz Celja.

Racunam...

Rezultat

+ a1 *a0 *b1 *b0 *c1 c0 *a3 a2 *b3 *b2 c3 *c2

Alternativni izracun za lastnost 4

(F4 == G4): TRUE

(F4SYSTEM == G4SYSTEM): TRUE

(F4 == H4): FALSE

(F4SYSTEM == H4SYSTEM): TRUE