CS 8803: Logic in Computer Science

Georgia Institute of Technology

School of Computer Science

Project-I Interim Report: Encoding Einstein's Riddle in CNF & Solving by DPLL Algorithm

Submitted by:

Prithwish Jana
PhD Student
School of Computer Science
Georgia Institute of Technology

Email: `pjana7@gatech.edu`

Course project report for CS 8803, taught by Prof. Suguman Bansal

Date: September 17, 2023

1 Problem Statement

The DPLL Procedure

The DPLL procedure consists of two rules:

- Unit-preference rule; select a unit clause c, assign to the proposition of c a truth value that satisfies c, simplify the input formula and call DPLL recursively.
- Splitting rule: select a proposition p, assign to it a truth value, simplify the input formula and call DPLL recursively. If this call fails to satisfy the input formula, assign another truth value to p, simplify the input formula and call DPLL recursively.

Note that this description leaves you with important implementation choices: how to choose propositions in the unit-preference rule and in the splitting rule and how to choose a truth value in the splitting rule. These choices can have a significant impact on the running time of DPLL. Note that even though DPLL is described as a recursive procedure, you need not implement it as such; you can gain significant efficiency by implementing it as an iterative procedure.

You should choose a heuristic for choosing propositions and truth values. To evaluate the effectiveness of your heuristic you should compare them to the *random-choice heuristic*, in which all choices are resolved randomly with a uniform distribution. You should also compare *your heuristic* to the *2-clause heuristic*, in which we choose propositions with maximum occurrences in 2-clauses, i.e., clauses with two literals, and break ties randomly. I expect you to find a heuristic that would be better than the 2-clause heuristic. *Make sure that all implementations agree on the answer (satisfiable or unsatisfiable) for all each input formula.*

You will evaluate your solver by using it to solve a puzzle and measuring its performance on random formulas.

The Puzzle

Supposedly, Albert Einstein wrote this riddle, and said 98% of the people could not solve it.

There are five houses in five different colors. In each house lives a man with a different nationality. The five owners drink a certain type of beverage, smoke a certain brand of cigar, and keep a certain pet. No owners have the same pet, smoke the same brand of cigar or drink the same beverage. The question is: "Who owns the fish?"

Hints:

- The Brit lives in the red house.
- The Swede keeps dogs as pets.
- The Dane drinks tea.
- The green house is on the left of the white house.
- The green house's owner drinks coffee.
- The person who smokes Pall Mall rears birds.
- The owner of the yellow house smokes Dunhill.
- The man living in the center house drinks milk.
- The Norwegian lives in the first house.
- The man who smokes Blends lives next to the one who keeps cats.
- The man who keeps the horse lives next to the man who smokes Dunhill.
- The owner who smokes Bluemasters drinks beer.
- The German smokes Prince.
- The Norwegian lives next to the blue house.
- The man who smokes Blends has a neighbor who drinks water.

In order to solve the puzzle, you have to make some assumptions:

1. The owner is the resident of each house.
2. One of the residents owns the fish.
3. The term neighbor in the last hint refers only to a directly adjacent neighbor.
4. The houses are on the same side of the street.
5. They are next to each other, and are ordered from left to right as you face them rather than standing in front of a house facing the street (i.e. facing the same direction as the house).

You need to solve this puzzle by expressing it first as a CNF formula such that the satisfying assignment of the formula corresponds to possible solutions of the puzzle.

Interim Report: Your interim report should include your encoding of the problem as a CNF formula and your solution of the puzzle, obtained using your implementation of DPLL. Include a link to the source code for your implementation of DPLL.

2 Defining the Propositions

From the fifteen hints given in the Einstein's riddle, it can be observed that there are five houses (on the same side of the street) and every house has an owner / resident. And there are some characteristics (or ***properties***) of these residents or their respective house, namely:

p0. **Color** of the resident's house:

- a0. *Red*
- a1. *Green*
- a2. *White*
- a3. *Blue*
- a4. *Yellow*

p1. **Nationality** of the resident:

- a0. *Brit*
- a1. *Swede*
- a2. *Dane*
- a3. *Norwegian*
- a4. *German*

p2. **Drink** preferred by the resident:

- a0. *Tea*
- a1. *Coffee*
- a2. *Water*
- a3. *Beer*
- a4. *Milk*

p3. **Cigarette** preferred by the resident:

- a0. *Prince*
- a1. *Blends*
- a2. *Pall Mall*
- a3. *Bluemasters*
- a4. *Dunhill*

p4. **Pet** of the resident:

- a0. *Dog*
- a1. *Cat*
- a2. *Bird*
- a3. *Horse*
- a4. *Fish*

So, there are five *properties* $\{p_0, p_1, p_2, p_3, p_4\}$ and five possible *assignments* $\{a_0, a_1, a_2, a_3, a_4\}$ for every *property*. Thus, for every owner or their respective house, there are $5 \times 5 = 25$ possible assignments. And because there are five houses $\{h_1, h_2, h_3, h_4, h_5\}$, we define $25 \times 5 = 125$ propositions. Essentially, we define a proposition $P_{h,p,a}$ as follows:

$$P_{i,j,k} = \begin{cases} T, & \text{if resident of house } h_i \text{ has property } p_j \text{ assigned to } a_k \\ F, & \text{otherwise} \end{cases} \tag{1}$$

where, $i \in [1, 2, 3, 4, 5]$ and $j, k \in [0, 1, 2, 3, 4]$.

For example, $P_{1,0,3} = T$ means the resident of house h_1 has his property p_0 i.e. color of his house assigned to a_3 i.e. Blue. $P_{2,4,1} = T$ means the resident of house h_2 has his property p_4 i.e. pet assigned to a_1 i.e. Cat.

For implementation purpose, we represent $P_{i,j,k}$ as $P_{(j\times25+k\times5+i)}$. This is a one-to-one mapping. For example, P_{94} represents $P_{4,3,3}$ because $94 = 25\times3+5\times3+4$. Thus, we have 125 propositions namely P_1 to P_{125} ($P_1 = P_{1,0,0}$ and $P_{125} = P_{5,4,4}$). We represent them as numbers 1 to 125 (with negative sign for NOT) in the subsequent sections, as per DIMACS CNF format.

3 Defining the Constraints

1. The property assignment (p_0, a_0) must be true for at least one of the house. Thus, it must be true that $P_{1,0,0} \vee P_{2,0,0} \vee P_{3,0,0} \vee P_{4,0,0} \vee P_{5,0,0}$ i.e. $P_1 \vee P_2 \vee P_3 \vee P_4 \vee P_5$. Similarly, this is true for all possible property assignments (p_j, a_k) for all $j, k \in [0, 1, 2, 3, 4]$. So, this gives rise to **25 constraints** as clauses.

   <u>EXAMPLE</u>: For property assignment (p_0, a_0), the clause in DIMACS CNF format is "1 2 3 4 5 0" (because, $P_{i,0,0} = P_{0\times25+0\times5+i}$).

   For property assignment (p_4, a_4), the clause in DIMACS CNF format is "121 122 123 124 125 0" (because, $P_{i,4,2} = P_{4\times25+4\times5+i}$).

2. For each property assignment (p_j, a_k), the proposition corresponding to only one of the houses must be true. So, for each of the possible property assignments (p_j, a_k) such that $j, k \in [0, 1, 2, 3, 4]$, it must be true that either $P_{n,j,k}$ is false or $P_{m,j,k}$ is false, where $m, n \in [1, 2, 3, 4, 5]$ and $m \neq n$ (m, n denote the house numbers). There are $5 \times 5 = 25$ property assignments (p_j, a_k) possible. And, $5C2 = 10$ unordered house pairs are possible. So, this gives rise to $\mathbf{25 \times 10 = 250}$ **constraints** as clauses.

   <u>EXAMPLE</u>: For property assignment (p_0, a_0), the ten clauses in DIMACS CNF format are:

   -1 -2 0 (because, $P_{m,0,0} = P_{0\times25+0\times5+m}$ and for $m = 1, n = 2$)
   -1 -3 0 (for $m = 1, n = 3$)
   -1 -4 0 (for $m = 1, n = 4$)
   -1 -5 0 (for $m = 1, n = 5$)

$-2\ \ -3\ \ 0$ (for $m = 2, n = 3$)
$-2\ \ -4\ \ 0$ (for $m = 2, n = 4$)
$-2\ \ -5\ \ 0$ (for $m = 2, n = 5$)
$-3\ \ -4\ \ 0$ (for $m = 3, n = 4$)
$-3\ \ -5\ \ 0$ (for $m = 3, n = 5$)
$-4\ \ -5\ \ 0$ (for $m = 4, n = 5$)

And similarly, there will be 10 clauses each for all the 25 property assignments.

3. For a house h_i where $i \in [1, 2, 3, 4, 5]$ and a property p_j where $j \in [0, 1, 2, 3, 4]$, the proposition corresponding to only one of the assignments must be true. So, for a fixed h_i and p_j, it must be true that either $P_{i,j,m}$ is false or $P_{i,j,n}$ is false, where $m, n \in [0, 1, 2, 3, 4]$ and $m \neq n$ (m, n denote the assignment numbers within a property). For each property (p_j), there are $5C2 = 10$ possibilities of assignment pairs m, n. So, there are $10 \times 5 = 50$ clauses per house. Because there are 5 houses, this gives rise to $\mathbf{50 \times 5 = 250}$ **constraints** as clauses.

   EXAMPLE: For house h_1 and property p_0, the ten clauses in DIMACS CNF format are:

   $-1\ \ -6\ \ 0$ (because, $P_{1,0,m} = P_{0\times25+m\times5+1}$ and for $m = 0, n = 1$)
   $-1\ \ -11\ \ 0$ (for $m = 0, n = 2$)
   $-1\ \ -16\ \ 0$ (for $m = 0, n = 3$)
   $-1\ \ -21\ \ 0$ (for $m = 0, n = 4$)
   $-6\ \ -11\ \ 0$ (for $m = 1, n = 2$)
   $-6\ \ -16\ \ 0$ (for $m = 1, n = 3$)
   $-6\ \ -21\ \ 0$ (for $m = 1, n = 4$)
   $-11\ \ -16\ \ 0$ (for $m = 2, n = 3$)
   $-11\ \ -21\ \ 0$ (for $m = 2, n = 4$)
   $-16\ \ -21\ \ 0$ (for $m = 3, n = 4$)

   And similarly, there will be 10 clauses each for all the 25 house-property pair.

4 Encoding the Hints as CNF

H1: "The Brit lives in the red house." We know that the proposition for resident of house h_i having property Nationality (p_1) as Brit (a_0) is $P_{i,1,0}$ i.e. P_{25+0+i}, where $i \in [1, 2, 3, 4, 5]$. And the proposition for resident of house h_i having property residence color (p_0) as Red (a_0) is $P_{i,0,0}$ i.e. P_{0+0+i}, where $i \in [1, 2, 3, 4, 5]$. This means that for any of the house h_i where $i \in [1, 2, 3, 4, 5]$, it must be true that $(\neg P_{25+i} \lor P_{0+i}) \land (P_{25+i} \lor \neg P_{0+i})$ which is in CNF. We call this as "isPairedWith" relationship, and will use it in the subsequent hints. The corresponding DIMACS CNF clauses are (by substituting $i \in [1, 2, 3, 4, 5]$):

```
-26 1 0
26 -1 0
```

```
-27 2 0
27 -2 0
-28 3 0
28 -3 0
-29 4 0
29 -4 0
-30 5 0
30 -5 0
```

H2: "The Swede keeps dogs as pets." We know that the proposition for resident of house h_i having property Nationality (p_1) as Swede (a_1) is $P_{i,1,1}$ i.e. P_{25+5+i}, where $i \in [1,2,3,4,5]$. And the proposition for resident of house h_i having property pet (p_4) as Dogs (a_0) is $P_{i,4,0}$ i.e. $P_{100+0+i}$, where $i \in [1,2,3,4,5]$. Using "isPairedWith" relationship, the corresponding DIMACS CNF clauses are (by substituting $i \in [1,2,3,4,5]$):

```
-31 101 0
31 -101 0
-32 102 0
32 -102 0
-33 103 0
33 -103 0
-34 104 0
34 -104 0
-35 105 0
35 -105 0
```

H3: "The Dane drinks tea." We know that the proposition for resident of house h_i having property Nationality (p_1) as Dane (a_2) is $P_{i,1,2}$ i.e. $P_{25+10+i}$, where $i \in [1,2,3,4,5]$. And the proposition for resident of house h_i having property drink (p_2) as tea (a_0) is $P_{i,2,0}$ i.e. P_{50+0+i}, where $i \in [1,2,3,4,5]$. Using "isPairedWith" relationship, the corresponding DIMACS CNF clauses are (by substituting $i \in [1,2,3,4,5]$):

```
-36 51 0
36 -51 0
-37 52 0
37 -52 0
-38 53 0
38 -53 0
-39 54 0
39 -54 0
-40 55 0
40 -55 0
```

H4: "The green house is on the left of the white house." We know that the proposition for resident of house h_i having property house color (p_0) as Green (a_1) is $P_{i,0,1}$ i.e. P_{0+5+i}, where $i \in [1,2,3,4,5]$. And the proposition for resident of house h_i having property house color (p_0) as White (a_2) is $P_{i,0,2}$ i.e. P_{0+10+i}, where $i \in [1,2,3,4,5]$. Representing this proposition in CNF means essentially NOT-ing (we know $\neg(a \wedge b) = (\neg a \vee \neg b)$) all those cases where the green house is to the right of the white house. The corresponding DIMACS CNF clauses are therefore:

```
-11 -10 0
-11 -9 0
-11 -8 0
-11 -7 0
-12 -10 0
-12 -9 0
-12 -8 0
-13 -10 0
-13 -9 0
-13 -6 0
-14 -10 0
-14 -7 0
-14 -6 0
-15 -8 0
-15 -7 0
-15 -6 0
```

H5: "The green house's owner drinks coffee." We know that the proposition for resident of house h_i having property house color (p_0) as Green (a_1) is $P_{i,0,1}$ i.e. P_{0+5+i}, where $i \in [1,2,3,4,5]$. And the proposition for resident of house h_i having property drink (p_2) as coffee (a_1) is $P_{i,2,1}$ i.e. P_{50+5+i}, where $i \in [1,2,3,4,5]$. Using "isPairedWith" relationship, the corresponding DIMACS CNF clauses are (by substituting $i \in [1,2,3,4,5]$):

```
-6 56 0
6 -56 0
-7 57 0
7 -57 0
-8 58 0
8 -58 0
-9 59 0
9 -59 0
-10 60 0
10 -60 0
```

H6: "The person who smokes Pall Mall rears birds." We know that the proposition for resident of house h_i having property cigarette (p_3) as Pall Mall (a_2) is $P_{i,3,2}$ i.e. $P_{75+10+i}$, where $i \in [1,2,3,4,5]$. And the proposition for resident of house h_i having property pet (p_4) as bird (a_2) is $P_{i,4,2}$ i.e. $P_{100+10+i}$, where $i \in [1,2,3,4,5]$. Using "isPairedWith" relationship, the corresponding DIMACS CNF clauses are (by substituting $i \in [1,2,3,4,5]$):

```
-86 111 0
86 -111 0
-87 112 0
87 -112 0
-88 113 0
88 -113 0
-89 114 0
89 -114 0
-90 115 0
90 -115 0
```

H7: "The owner of the yellow house smokes Dunhill." We know that the proposition for resident of house h_i having property house color (p_0) as Yellow (a_4) is $P_{i,0,4}$ i.e. P_{0+20+i}, where $i \in [1,2,3,4,5]$. And the proposition for resident of house h_i having property Cigarette (p_3) as Dunhill (a_4) is $P_{i,3,4}$ i.e. $P_{75+20+i}$, where $i \in [1,2,3,4,5]$. Using "isPairedWith" relationship, the corresponding DIMACS CNF clauses are (by substituting $i \in [1,2,3,4,5]$):

```
-21 96 0
21 -96 0
-22 97 0
22 -97 0
-23 98 0
23 -98 0
-24 99 0
24 -99 0
-25 100 0
25 -100 0
```

H8: "The man living in the center house drinks milk." The center house is h_3 because there are five houses h_i where $i \in [1,2,3,4,5]$. We know that the proposition for resident of house h_3 having property Drink (p_2) as Milk (a_4) is $P_{3,2,4}$ i.e. $P_{50+20+3}$. The corresponding DIMACS CNF clause is:

```
73 0
```

H9: "The Norwegian lives in the first house." The first house is h_1 because there are five houses h_i where $i \in [1,2,3,4,5]$. We know that the proposition for resident of house h_3 having property Nationality (p_1) as Norwegian (a_3) is $P_{1,1,3}$ i.e. $P_{25+15+1}$. The corresponding DIMACS CNF clause is:

```
41 0
```

H10: "The man who smokes Blends lives next to the one who keeps cats." We know that the proposition for resident of house h_i having property cigarette (p_3) as Blends (a_1) is $P_{i,3,1}$ i.e. P_{75+5+i}, where $i \in [1,2,3,4,5]$. And the proposition for resident of house h_i having property pet (p_4) as cat (a_1) is $P_{i,4,1}$ i.e. $P_{100+5+i}$, where $i \in [1,2,3,4,5]$. This can mean: the resident who smokes Blends lives in the first house and one who pets a cat lives in the second house. That is, $P_{75+5+1} \rightarrow P_{100+5+2}$ i.e. $\neg P_{81} \vee P_{107}$. Else, it can mean the resident who smokes Blends lives in the fifth house and one who pets a cat lives in the fourth house. That is, $P_{75+5+5} \rightarrow P_{100+5+4}$ i.e. $\neg P_{85} \vee P_{109}$. Else generically, it can indicate that the resident who smokes Blends lives in the i-th house and the one who pets a cat lives in the (i-1)-th or (i+1)-th house, for $i \in [2,3,4]$. This gives rise to clauses $\neg P_{80+2} \vee P_{105+(2-1)} \vee P_{105+(2+1)}$ i.e. $\neg P_{82} \vee P_{106} \vee P_{108}$, $\neg P_{83} \vee P_{107} \vee P_{109}$ and $\neg P_{84} \vee P_{108} \vee P_{110}$. This we'll regard as "isNeighbor" relationship henceforth. So, using "isNeighbor" relationship, the corresponding DIMACS CNF clauses are:

```
-81 107 0
-85 109 0
-82 106 108 0
-83 107 109 0
-84 108 110 0
```

H11: "The man who keeps the horse lives next to the man who smokes Dunhill." We know that the proposition for resident of house h_i having property pet (p_4) as Horse (a_3) is $P_{i,4,3}$ i.e. $P_{100+15+i}$, where $i \in [1,2,3,4,5]$. And the proposition for resident of house h_i having property Cigarette (p_3) as Dunhill (a_4) is $P_{i,3,4}$ i.e. $P_{75+20+i}$, where $i \in [1,2,3,4,5]$. Then by "isNeighbor" relationship, the corresponding DIMACS CNF clauses are:

```
-116 97 0
-120 99 0
-117 96 98 0
-118 97 99 0
-119 98 100 0
```

H12: "The owner who smokes Bluemasters drinks beer." We know that the proposition for resident of house h_i having property Cigarette (p_3) as Bluemasters (a_3) is $P_{i,3,3}$ i.e.

$P_{75+15+i}$, where $i \in [1,2,3,4,5]$. And the proposition for resident of house h_i having property Drink (p_2) as Beer (a_3) is $P_{i,2,3}$ i.e. $P_{50+15+i}$, where $i \in [1,2,3,4,5]$. Using "isPairedWith" relationship, the corresponding DIMACS CNF clauses are (by substituting $i \in [1,2,3,4,5]$):

```
-91 66 0
91 -66 0
-92 67 0
92 -67 0
-93 68 0
93 -68 0
-94 69 0
94 -69 0
-95 70 0
95 -70 0
```

H13: "The German smokes Prince." We know that the proposition for resident of house h_i having property Nationality (p_1) as German (a_4) is $P_{i,1,4}$ i.e. $P_{25+20+i}$, where $i \in [1,2,3,4,5]$. And the proposition for resident of house h_i having property Cigarette (p_3) as Prince (a_0) is $P_{i,3,0}$ i.e. P_{75+0+i}, where $i \in [1,2,3,4,5]$. Using "isPairedWith" relationship, the corresponding DIMACS CNF clauses are (by substituting $i \in [1,2,3,4,5]$):

```
-46 76 0
46 -76 0
-47 77 0
47 -77 0
-48 78 0
48 -78 0
-49 79 0
49 -79 0
-50 80 0
50 -80 0
```

H14: "The Norwegian lives next to the blue house." It is given in H9 that the Norwegian lives in the first house i.e. h_1. We know that the proposition for resident of house h_2 having property house color (p_0) as Blue (a_3) is $P_{2,0,3}$ i.e. P_{0+15+2}. The corresponding DIMACS CNF clause is:

```
17 0
```

H15: "The man who smokes Blends has a isNeighborOf who drinks water." We know that the proposition for resident of house h_i having property Cigarette (p_3) as Blends (a_1)

is $P_{i,3,1}$ i.e. P_{75+5+i}, where $i \in [1,2,3,4,5]$. And the proposition for resident of house h_i having property Drink (p_2) as Water (a_2) is $P_{i,2,2}$ i.e. $P_{50+10+i}$, where $i \in [1,2,3,4,5]$. Then by "isNeighbor" relationship, the corresponding DIMACS CNF clauses are:

```
-81 62 0
-85 64 0
-82 61 63 0
-83 62 64 0
-84 63 65 0
```

5 The overall DIMACS CNF Representation

```
p cnf 125 639
1 2 3 4 5 0
-1 -2 0
-1 -3 0
-2 -3 0
-1 -4 0
-2 -4 0
-3 -4 0
-1 -5 0
-2 -5 0
-3 -5 0
-4 -5 0
6 7 8 9 10 0
-6 -1 0
-6 -7 0
-7 -2 0
-6 -8 0
-7 -8 0
-8 -3 0
-6 -9 0
-7 -9 0
-8 -9 0
-9 -4 0
-6 -10 0
-7 -10 0
-8 -10 0
-9 -10 0
-10 -5 0
11 12 13 14 15 0
-11 -1 0
-11 -6 0
-11 -12 0
-12 -2 0
-12 -7 0
-11 -13 0
-12 -13 0
-13 -3 0
-13 -8 0
-11 -14 0
-12 -14 0
-13 -14 0
-14 -4 0
-14 -9 0
```

```
-11 -15 0
-12 -15 0
-13 -15 0
-14 -15 0
-15 -5 0
-15 -10 0
16 17 18 19 20 0
-16 -1 0
-16 -6 0
-16 -11 0
-16 -17 0
-17 -2 0
-17 -7 0
-17 -12 0
-16 -18 0
-17 -18 0
-18 -3 0
-18 -8 0
-18 -13 0
-16 -19 0
-17 -19 0
-18 -19 0
-19 -4 0
-19 -9 0
-19 -14 0
-16 -20 0
-17 -20 0
-18 -20 0
-19 -20 0
-20 -5 0
-20 -10 0
-20 -15 0
21 22 23 24 25 0
-21 -1 0
-21 -6 0
-21 -11 0
-21 -16 0
-21 -22 0
-22 -2 0
-22 -7 0
-22 -12 0
-22 -17 0
-21 -23 0
-22 -23 0
-23 -3 0
-23 -8 0
```

```
-23 -13 0
-23 -18 0
-21 -24 0
-22 -24 0
-23 -24 0
-24 -4 0
-24 -9 0
-24 -14 0
-24 -19 0
-21 -25 0
-22 -25 0
-23 -25 0
-24 -25 0
-25 -5 0
-25 -10 0
-25 -15 0
-25 -20 0
26 27 28 29 30 0
-26 -27 0
-26 -28 0
-27 -28 0
-26 -29 0
-27 -29 0
-28 -29 0
-26 -30 0
-27 -30 0
-28 -30 0
-29 -30 0
31 32 33 34 35 0
-31 -26 0
-31 -32 0
-32 -27 0
-31 -33 0
-32 -33 0
-33 -28 0
-31 -34 0
-32 -34 0
-33 -34 0
-34 -29 0
-31 -35 0
-32 -35 0
-33 -35 0
-34 -35 0
-35 -30 0
36 37 38 39 40 0
-36 -26 0
```

```
-36 -31 0
-36 -37 0
-37 -27 0
-37 -32 0
-36 -38 0
-37 -38 0
-38 -28 0
-38 -33 0
-36 -39 0
-37 -39 0
-38 -39 0
-39 -29 0
-39 -34 0
-36 -40 0
-37 -40 0
-38 -40 0
-39 -40 0
-40 -30 0
-40 -35 0
41 42 43 44 45 0
-41 -26 0
-41 -31 0
-41 -36 0
-41 -42 0
-42 -27 0
-42 -32 0
-42 -37 0
-41 -43 0
-42 -43 0
-43 -28 0
-43 -33 0
-43 -38 0
-41 -44 0
-42 -44 0
-43 -44 0
-44 -29 0
-44 -34 0
-44 -39 0
-41 -45 0
-42 -45 0
-43 -45 0
-44 -45 0
-45 -30 0
-45 -35 0
-45 -40 0
46 47 48 49 50 0
```

```
-46 -26 0
-46 -31 0
-46 -36 0
-46 -41 0
-46 -47 0
-47 -27 0
-47 -32 0
-47 -37 0
-47 -42 0
-46 -48 0
-47 -48 0
-48 -28 0
-48 -33 0
-48 -38 0
-48 -43 0
-46 -49 0
-47 -49 0
-48 -49 0
-49 -29 0
-49 -34 0
-49 -39 0
-49 -44 0
-46 -50 0
-47 -50 0
-48 -50 0
-49 -50 0
-50 -30 0
-50 -35 0
-50 -40 0
-50 -45 0
51 52 53 54 55 0
-51 -52 0
-51 -53 0
-52 -53 0
-51 -54 0
-52 -54 0
-53 -54 0
-51 -55 0
-52 -55 0
-53 -55 0
-54 -55 0
56 57 58 59 60 0
-56 -51 0
-56 -57 0
-57 -52 0
-56 -58 0
```

```
-57 -58 0
-58 -53 0
-56 -59 0
-57 -59 0
-58 -59 0
-59 -54 0
-56 -60 0
-57 -60 0
-58 -60 0
-59 -60 0
-60 -55 0
61 62 63 64 65 0
-61 -51 0
-61 -56 0
-61 -62 0
-62 -52 0
-62 -57 0
-61 -63 0
-62 -63 0
-63 -53 0
-63 -58 0
-61 -64 0
-62 -64 0
-63 -64 0
-64 -54 0
-64 -59 0
-61 -65 0
-62 -65 0
-63 -65 0
-64 -65 0
-65 -55 0
-65 -60 0
66 67 68 69 70 0
-66 -51 0
-66 -56 0
-66 -61 0
-66 -67 0
-67 -52 0
-67 -57 0
-67 -62 0
-66 -68 0
-67 -68 0
-68 -53 0
-68 -58 0
-68 -63 0
-66 -69 0
```

```
-67 -69 0
-68 -69 0
-69 -54 0
-69 -59 0
-69 -64 0
-66 -70 0
-67 -70 0
-68 -70 0
-69 -70 0
-70 -55 0
-70 -60 0
-70 -65 0
71 72 73 74 75 0
-71 -51 0
-71 -56 0
-71 -61 0
-71 -66 0
-71 -72 0
-72 -52 0
-72 -57 0
-72 -62 0
-72 -67 0
-71 -73 0
-72 -73 0
-73 -53 0
-73 -58 0
-73 -63 0
-73 -68 0
-71 -74 0
-72 -74 0
-73 -74 0
-74 -54 0
-74 -59 0
-74 -64 0
-74 -69 0
-71 -75 0
-72 -75 0
-73 -75 0
-74 -75 0
-75 -55 0
-75 -60 0
-75 -65 0
-75 -70 0
76 77 78 79 80 0
-76 -77 0
-76 -78 0
```

```
-77 -78 0
-76 -79 0
-77 -79 0
-78 -79 0
-76 -80 0
-77 -80 0
-78 -80 0
-79 -80 0
81 82 83 84 85 0
-81 -76 0
-81 -82 0
-82 -77 0
-81 -83 0
-82 -83 0
-83 -78 0
-81 -84 0
-82 -84 0
-83 -84 0
-84 -79 0
-81 -85 0
-82 -85 0
-83 -85 0
-84 -85 0
-85 -80 0
86 87 88 89 90 0
-86 -76 0
-86 -81 0
-86 -87 0
-87 -77 0
-87 -82 0
-86 -88 0
-87 -88 0
-88 -78 0
-88 -83 0
-86 -89 0
-87 -89 0
-88 -89 0
-89 -79 0
-89 -84 0
-86 -90 0
-87 -90 0
-88 -90 0
-89 -90 0
-90 -80 0
-90 -85 0
91 92 93 94 95 0
```

```
-91 -76 0
-91 -81 0
-91 -86 0
-91 -92 0
-92 -77 0
-92 -82 0
-92 -87 0
-91 -93 0
-92 -93 0
-93 -78 0
-93 -83 0
-93 -88 0
-91 -94 0
-92 -94 0
-93 -94 0
-94 -79 0
-94 -84 0
-94 -89 0
-91 -95 0
-92 -95 0
-93 -95 0
-94 -95 0
-95 -80 0
-95 -85 0
-95 -90 0
96 97 98 99 100 0
-96 -76 0
-96 -81 0
-96 -86 0
-96 -91 0
-96 -97 0
-97 -77 0
-97 -82 0
-97 -87 0
-97 -92 0
-96 -98 0
-97 -98 0
-98 -78 0
-98 -83 0
-98 -88 0
-98 -93 0
-96 -99 0
-97 -99 0
-98 -99 0
-99 -79 0
-99 -84 0
```

```
-99 -89 0
-99 -94 0
-96 -100 0
-97 -100 0
-98 -100 0
-99 -100 0
-100 -80 0
-100 -85 0
-100 -90 0
-100 -95 0
101 102 103 104 105 0
-101 -102 0
-101 -103 0
-102 -103 0
-101 -104 0
-102 -104 0
-103 -104 0
-101 -105 0
-102 -105 0
-103 -105 0
-104 -105 0
106 107 108 109 110 0
-106 -101 0
-106 -107 0
-107 -102 0
-106 -108 0
-107 -108 0
-108 -103 0
-106 -109 0
-107 -109 0
-108 -109 0
-109 -104 0
-106 -110 0
-107 -110 0
-108 -110 0
-109 -110 0
-110 -105 0
111 112 113 114 115 0
-111 -101 0
-111 -106 0
-111 -112 0
-112 -102 0
-112 -107 0
-111 -113 0
-112 -113 0
-113 -103 0
```

```
-113 -108 0
-111 -114 0
-112 -114 0
-113 -114 0
-114 -104 0
-114 -109 0
-111 -115 0
-112 -115 0
-113 -115 0
-114 -115 0
-115 -105 0
-115 -110 0
116 117 118 119 120 0
-116 -101 0
-116 -106 0
-116 -111 0
-116 -117 0
-117 -102 0
-117 -107 0
-117 -112 0
-116 -118 0
-117 -118 0
-118 -103 0
-118 -108 0
-118 -113 0
-116 -119 0
-117 -119 0
-118 -119 0
-119 -104 0
-119 -109 0
-119 -114 0
-116 -120 0
-117 -120 0
-118 -120 0
-119 -120 0
-120 -105 0
-120 -110 0
-120 -115 0
121 122 123 124 125 0
-121 -101 0
-121 -106 0
-121 -111 0
-121 -116 0
-121 -122 0
-122 -102 0
-122 -107 0
```

```
-122 -112 0
-122 -117 0
-121 -123 0
-122 -123 0
-123 -103 0
-123 -108 0
-123 -113 0
-123 -118 0
-121 -124 0
-122 -124 0
-123 -124 0
-124 -104 0
-124 -109 0
-124 -114 0
-124 -119 0
-121 -125 0
-122 -125 0
-123 -125 0
-124 -125 0
-125 -105 0
-125 -110 0
-125 -115 0
-125 -120 0
-26 1 0
26 -1 0
-27 2 0
27 -2 0
-28 3 0
28 -3 0
-29 4 0
29 -4 0
-30 5 0
30 -5 0
-31 101 0
31 -101 0
-32 102 0
32 -102 0
-33 103 0
33 -103 0
-34 104 0
34 -104 0
-35 105 0
35 -105 0
-36 51 0
36 -51 0
-37 52 0
```

```
37 -52 0
-38 53 0
38 -53 0
-39 54 0
39 -54 0
-40 55 0
40 -55 0
-6 56 0
6 -56 0
-7 57 0
7 -57 0
-8 58 0
8 -58 0
-9 59 0
9 -59 0
-10 60 0
10 -60 0
-86 111 0
86 -111 0
-87 112 0
87 -112 0
-88 113 0
88 -113 0
-89 114 0
89 -114 0
-90 115 0
90 -115 0
-21 96 0
21 -96 0
-22 97 0
22 -97 0
-23 98 0
23 -98 0
-24 99 0
24 -99 0
-25 100 0
25 -100 0
73 0
41 0
-81 107 0
-85 109 0
-82 106 108 0
-83 107 109 0
-84 108 110 0
-116 97 0
-120 99 0
```

```
-117 96 98 0
-118 97 99 0
-119 98 100 0
-91 66 0
91 -66 0
-92 67 0
92 -67 0
-93 68 0
93 -68 0
-94 69 0
94 -69 0
-95 70 0
95 -70 0
-46 76 0
46 -76 0
-47 77 0
47 -77 0
-48 78 0
48 -78 0
-49 79 0
49 -79 0
-50 80 0
50 -80 0
17 0
-81 62 0
-85 64 0
-82 61 63 0
-83 62 64 0
-84 63 65 0
-11 -10 0
-11 -9 0
-11 -8 0
-11 -7 0
-12 -10 0
-12 -9 0
-12 -8 0
-13 -10 0
-13 -9 0
-13 -6 0
-14 -10 0
-14 -7 0
-14 -6 0
-15 -8 0
-15 -7 0
-15 -6 0
```

6 Satisfying Assignment obtained by running the SAT Solver

In total, all the constraints and the hints added up to 639 clauses over 125 propositions.

On running the SAT solver (by random heuristic, 2-clause heuristic and own heuristic), the answer obtained is SAT i.e. the given CNF formula is satisfiable. Precisely, the output of my SAT solver and the satisfying assignment is found to be as follows:

```
SAT
v -1 -2 3 -4 -5 -6 -7 -8 9 -10 -11 -12 -13 -14 15 -16 17 -18 -19 -20 21 -22 -23
-24 -25 -26 -27 28 -29 -30 -31 -32 -33 -34 35 -36 37 -38 -39 -40 41 -42 -43 -44
-45 -46 -47 -48 49 -50 -51 52 -53 -54 -55 -56 -57 -58 59 -60 61 -62 -63 -64 -65
-66 -67 -68 -69 70 -71 -72 73 -74 -75 -76 -77 -78 79 -80 -81 82 -83 -84 -85 -86
-87 88 -89 -90 -91 -92 -93 -94 95 96 -97 -98 -99 -100 -101 -102 -103 -104 105 106
-107 -108 -109 -110 -111 -112 113 -114 -115 -116 117 -118 -119 -120 -121 -122
-123 124 -125 0
```

7 Solving the Puzzle using the Satisfying Assignment

The puzzle asks the question: "Who owns the fish?". We now try to find its answer using the satisfying assignment:

The proposition for resident of house h_i having property pet (p_4) as Fish (a_4) is $P_{i,4,4}$ i.e. $P_{100+20+i} = P_{120+i}$, where $i \in [1, 2, 3, 4, 5]$. From the satisfying assignment, we see that only proposition P_{124} is true among P_{121}, P_{122}, P_{123}, P_{124}, P_{125}. So, the person who owns the fish **resides in house $\mathbf{h_4}$ i.e. the fourth house.**

If we now want to probe the Nationality (p_1) of the owner of the fourth house (h_4), we can check propositions $P_{4,1,k}$ i.e. $P_{25+4+5k}$ i.e. P_{29+5k}. As k can take values $[0, 1, 2, 3, 4]$, we can check propositions P_{29}, P_{34}, P_{39}, P_{44} and P_{49}. Out of them, only P_{49} is true in the satisfying assignment. This means, for owner of house h_4, the property p_1 i.e. Nationality has assignment a_4 which is German. So, the person who owns the fish **resides in house $\mathbf{h_4}$ i.e. the fourth house** and is **German**.

Thus, the German (who lives in the fourth house) owns the fish.

8 Result & Time Performance for Different Heuristics Tested

We tested on the given two heuristics i.e. *random-choice heuristic* in which all choices are resolved randomly with a uniform distribution and 2-clause heuristic in which we choose propositions with maximum occurrences in 2-clauses and break ties randomly. We also tested with a third heuristic which is the 2-sided Jeroslow-Wang rule. For all these three, we calculated the time performance. The result is as follows:

```
(base) pjana@lawn solution_PJ % python DPLL_random_PJ.py
-------------
SAT Solver using Random-Choice Heuristic
-------------
SAT
v -1 -2 3 -4 -5 -6 -7 -8 9 -10 -11 -12 -13 -14 15 -16 17 -18 -19 -20 21 -22 -23
-24 -25 -26 -27 28 -29 -30 -31 -32 -33 -34 35 -36 37 -38 -39 -40 41 -42 -43 -44
-45 -46 -47 -48 49 -50 -51 52 -53 -54 -55 -56 -57 -58 59 -60 61 -62 -63 -64 -65
-66 -67 -68 -69 70 -71 -72 73 -74 -75 -76 -77 -78 79 -80 -81 82 -83 -84 -85 -86
-87 88 -89 -90 -91 -92 -93 -94 95 96 -97 -98 -99 -100 -101 -102 -103 -104 105 106
-107 -108 -109 -110 -111 -112 113 -114 -115 -116 117 -118 -119 -120 -121 -122
-123 124 -125 0
Execution Time: 0.05127096176147461 sec
(base) pjana@lawn solution_PJ %
```

```
(base) pjana@lawn solution_PJ % python DPLL_2clause_PJ.py
-------------
SAT Solver using 2-Clause Heuristic
-------------
SAT
v -1 -2 3 -4 -5 -6 -7 -8 9 -10 -11 -12 -13 -14 15 -16 17 -18 -19 -20 21 -22 -23
-24 -25 -26 -27 28 -29 -30 -31 -32 -33 -34 35 -36 37 -38 -39 -40 41 -42 -43 -44
-45 -46 -47 -48 49 -50 -51 52 -53 -54 -55 -56 -57 -58 59 -60 61 -62 -63 -64 -65
-66 -67 -68 -69 70 -71 -72 73 -74 -75 -76 -77 -78 79 -80 -81 82 -83 -84 -85 -86
-87 88 -89 -90 -91 -92 -93 -94 95 96 -97 -98 -99 -100 -101 -102 -103 -104 105 106
-107 -108 -109 -110 -111 -112 113 -114 -115 -116 117 -118 -119 -120 -121 -122
-123 124 -125 0
Execution Time: 0.01483917236328125 sec
(base) pjana@lawn solution_PJ %
```

```
(base) pjana@lawn solution_PJ % python DPLL_betterHeuristic.py
-------------
SAT Solver using Jeroslow-Wang rule Heuristic
-------------
SAT
v -1 -2 3 -4 -5 -6 -7 -8 9 -10 -11 -12 -13 -14 15 -16 17 -18 -19 -20 21 -22 -23
-24 -25 -26 -27 28 -29 -30 -31 -32 -33 -34 35 -36 37 -38 -39 -40 41 -42 -43 -44
-45 -46 -47 -48 49 -50 -51 52 -53 -54 -55 -56 -57 -58 59 -60 61 -62 -63 -64 -65
-66 -67 -68 -69 70 -71 -72 73 -74 -75 -76 -77 -78 79 -80 -81 82 -83 -84 -85 -86
-87 88 -89 -90 -91 -92 -93 -94 95 96 -97 -98 -99 -100 -101 -102 -103 -104 105 106
-107 -108 -109 -110 -111 -112 113 -114 -115 -116 117 -118 -119 -120 -121 -122
-123 124 -125 0
Execution Time: 0.026946067810058594 sec
(base) pjana@lawn solution_PJ %
```