



Artificial Intelligence

Laboratory activity

Name: Stoian Iulia Tia, Săsăran Andrei
Group: 30231

Email:

stoianiuliata@gmail.com
sasaranandreipaul@gmail.com

Teaching Assistant:

Alexandru Ghiuruțan
alexandru.ghiurutan95@gmail.com



Contents

1 A1: Search	5
1.1 8 Puzzle - State Space Searches	5
2 A2: Logics	6
2.1 Who owns the crocodile? - Easy Zebra Puzzle	6
2.1.1 Clues	6
2.1.2 Human Thinking Solution	7
2.1.3 Mathematical Solution	7
2.1.4 Mace4 Results	8
2.1.5 Prover9 Results	10
2.2 Ancient Civilizations - Medium Zebra Puzzle	11
2.2.1 Clues	11
2.2.2 Human Thinking Solution	12
2.2.3 Mathematical Solution	12
2.2.4 Mace4 Results	13
2.2.5 Prover9 Results	15
2.3 Persian Rugs - Hard Zebra Puzzle	16
2.3.1 Clues	16
2.3.2 Human Thinking Solution	17
2.3.3 Mathematical Solution	17
2.3.4 Mace4 Results	18
2.3.5 Prover9 Results	20
2.4 Travel Agency - Hard Zebra Puzzle	21
2.4.1 Clues	21
2.4.2 Human Thinking Solution	22
2.4.3 Mathematical Solution	22
2.4.4 Mace4 Results	23
2.4.5 Prover9 Results	25
2.5 Secret Agents - Very Hard Zebra Puzzle	26
2.5.1 Clues	26
2.5.2 Human Thinking Solution	27
2.5.3 Mathematical Solution	27
2.5.4 Mace4 Results	28
2.5.5 Prover9 Results	30
2.6 Who likes dolphins? - Easy Zebra Puzzle	31
2.6.1 Clues	31
2.6.2 Human Thinking Solution	31
2.6.3 Mathematical Solution	31
2.6.4 Mace4 Results	32
2.6.5 Prover9 Results	33

2.7	Hardware Store - Medium Zebra Puzzle	34
2.7.1	Clues	34
2.7.2	Human Thinking Solution	35
2.7.3	Mathematical Solution	35
2.7.4	Mace4 Results	36
2.7.5	Prover9 Results	38
2.8	Black Friday - Hard Zebra Puzzle	39
2.8.1	Clues	39
2.8.2	Human Thinking Solution	40
2.8.3	Mathematical Solution	40
2.8.4	Mace4 Results	41
2.8.5	Prover9 Results	43
2.9	Home Remodeling - Hard Zebra Puzzle	44
2.9.1	Clues	44
2.9.2	Human Thinking Solution	45
2.9.3	Mathematical Solution	45
2.9.4	Mace4 Results	46
2.9.5	Prover9 Results	48
3	A3: Planning	49
A	Your original code	51
A.1	Stoian Iulia Tia's code:	51
A.1.1	Search	51
A.1.2	Logics	55
A.2	Săsăran Andrei Paul's code:	76
A.2.1	Search	76
A.2.2	Logics	84

Table 1: Lab scheduling

Activity	Deadline
<i>Searching agents, Linux, Latex, Python, Pacman</i>	W_1
<i>Uninformed search</i>	W_2
<i>Informed Search</i>	W_3
<i>Adversarial search</i>	W_4
<i>Propositional logic</i>	W_5
<i>First order logic</i>	W_6
<i>Inference in first order logic</i>	W_7
<i>Knowledge representation in first order logic</i>	W_8
<i>Classical planning</i>	W_9
<i>Contingent, conformant and probabilistic planning</i>	W_{10}
<i>Multi-agent planing</i>	W_{11}
<i>Modelling planning domains</i>	W_{12}
<i>Planning with event calculus</i>	W_{14}

Lab organisation.

1. Laboratory work is 25% from the final grade.
2. There are three deliverables in total: 1. Search, 2. Logic, 3. Planning.
3. Before each deadline, you have to send your work (latex documentation/code) at moodle.cs.utcluj.ro
4. We use Linux and Latex
5. Plagiarism: Don't be a cheater! Cheating affects your colleagues, scholarships and a lot more.

Chapter 1

A1: Search

1.1 8 Puzzle - State Space Searches

Problema 8 Puzzle consta intr-un puzzle compus din (3 x 3 - 1) celule, numerotate de la 1 la 8. Cea de a 9-a celula care defineste puzzle-ul este o celula "goala" care este folosita de catre solver pentru a modifica componenția puzzle-ului, prin mutarea unei celule adiacente in pozitia acesteia. Goal state-ul este atins atunci cand celula goala se afla pe prima pozitie (coltul din stanga sus), iar fiecare celula numerotata asezata in ordine crescatoare in continuarea acesteia, cu prioritate de la stanga la dreapta. Pentru a rezolva aceasta problema am utilizat doua tipuri de algoritmi de cautare neinformate (Breadth-First Search si Depth-First Search) si doua tipuri de algoritmi de cautare informate (Greedy Search si A* Search, fiecare utilizat pe rand cu una dintre euristicile Manhattan Distance si Euclidian Distance).

Fiecare forma in care pot fi asezate celulele puzzle-lui este un state space. Succesorii reprezinta state space-urile care rezulta prin realizarea unei mutari posibile (o mutare legala, care sa respecte cadrul puzzle-ului) a celulei goale, adica state space-ul curent are la randul lui ale state space-uri succesoare. In functie de algoritmul de cautare ales, aceste state space-uri vor fi parcurse pana la gasirea solutiei optime.

Proiectul este bazat pe modelul MVC (Model-View-Controller), avand o interfata prin intermediul careia utilizatorul poate avea acces la functionalitatatile existente. Utilizatorul poate vedea efectiv celulele puzzle-ului, poate selecta sa se afiseze cum arata goal state-ul, poate apasa pe butonul "shuffle" care amesteca celulele intre ele si poate apasa pe butonul "solve" care va afisa in consola un string reprezentat prin concatenarea literelor N, E, S, W, reprezentand directiile in care a fost mutata celula goala.

Chapter 2

A2: Logics

2.1 Who owns the crocodile? - Easy Zebra Puzzle

Five girls are sitting in a row. Each girl has a favorite color, a chocolate bar, a pet, a hobby, and a place to go on holiday. Determine the name of the girl who owns the crocodile.

2.1.1 Clues

1. Jo likes the Wispa Bites.
2. The person with the Hamster likes Swimming.
3. Hannah eats Dairy Milk.
4. Jessica is on the left of Georgina.
5. Lucy is the first on the left.
6. The first person on the right likes Swimming.
7. The person who eats Milky Bars owns a Horse.
8. The person in the middle eats Dairy Milk.
9. Jessica likes Green.
10. The person on the left of the middle wants to go to Tobago.
11. The person who wants to go to Maldives likes Lilac.
12. The person who likes Wispa Bites sits next to the person who wants to go to Florida.
13. The person who likes Pink wants to go to Florida.
14. The person who sits first on the left likes lilac.
15. The girl that likes Blue owns a Puppy.
16. The person who likes Skiing sits next to the person who has a Hamster.
17. The girl on the right of the girl who likes Tennis likes Horse riding.
18. The girl next to the girl who likes Milky bars likes Boost.
19. The girl who likes Purple wants to go to Canada.
20. The girl who likes Crunchies owns a rabbit.
21. The girl who likes Skiing sits next to the girl who play Ten-Pin bowlings.
22. Jessica wants to go to Australia.

2.1.2 Human Thinking Solution

1. At the fifth position is the girl that likes swimming and has a hamster. (**Clue 6 and Clue 2**)
2. The girl at the third position is called Hannah and she eats Dairy Milk. (**Clue 8 and Clue 3**)
3. The girl at the first position is called Lucy. She likes lilac and wants to go to Maldives. (**Clue 5, Clue 14 and Clue 11**)
4. **Step 1** → The fourth girl likes skiing and the third girl plays ten-pin bowling. (**Clue 16 and Clue 21**)
5. **Step 4** → The first girl likes Tennis and the second girl likes horse riding. (**Clue 17**)
6. **Step 2 and Step 3** → The fourth girl can only be Jessica and the fifth girl can only be Georgina. (**Clue 4**)
7. **Step 6** → The second girl can only be Jo and she likes the Wispa Bites. (**Clue 1**)
8. **Step 7** → The third girl wants to go to Florida and she likes pink. (**Clue 12 and Clue 13**)
9. The second girl wants to go to Tobago. (**Clue 10**)
10. **Step 6** → The fourth girl wants to go to Australia and she likes green. (**Clue 22 and Clue 9**)
11. **Step 3, Step 7, Step 8 and Step 9** → The fifth girl wants to go to Canada and she likes purple. (**Clue 19**)
12. **Step 3, Step 8, Step 10 and Step 11** → The second girl likes blue and she owns a puppy. (**Clue 15**)
13. **Step 1, Step 2, Step 7 and Step 12** → The fourth girl can only own a horse and she likes Milky Bars. The fifth girl likes Boost. (**Clue 18 and Clue 7**)
14. **Step 13** → The first girl can only like Crunchies and she owns a rabbit. (**Clue 20**)
15. **Step 11** → The third girl owns the crocodile.

2.1.3 Mathematical Solution

	Girl #1	Girl #2	Girl #3	Girl #4	Girl #5
Color	Lilac	Blue	Pink	Green	Purple
Name	Lucy	Jo	Hannah	Jessica	Georgina
Chocolate	Crunchies	Wispa Bites	Dairy Milk	Milky Bars	Boost
Pet	Rabbit	Puppy	Crocodile	Horse	Hamster
Hobby	Tennis	Horse Riding	Bowling	Skiing	Swimming
Holiday	Maldives	Tobago	Florida	Australia	Canada

2.1.4 Mace4 Results

This puzzle was solved using predicates.

Australia	0	0	0	1	0
Blue	0	1	0	0	0
Boost	0	0	0	0	1
Bowling	0	0	1	0	0
Canada	0	0	0	0	1
Crocodile	0	0	1	0	0
Crunchies	1	0	0	0	0
Dairy Milk	0	0	1	0	0
Florida	0	0	1	0	0
Georgina	0	0	0	0	1
Girl1	1	0	0	0	0
Girl2	0	1	0	0	0
Girl3	0	0	1	0	0
Girl4	0	0	0	1	0
Girl5	0	0	0	0	1
Green	0	0	0	1	0
Hamster	0	0	0	0	1
Hannah	0	0	1	0	0
Horse	0	0	0	1	0
Jessica	0	0	0	1	0
Jo	0	1	0	0	0
Lilac	1	0	0	0	0
Lucy	1	0	0	0	0
Maldives	1	0	0	0	0
Milky Bars	0	0	0	1	0
Pink	0	0	1	0	0
Puppy	0	1	0	0	0

Purple		0	0	0	0	1
Rabbit		1	0	0	0	0
Riding		0	1	0	0	0
Skiing		0	0	0	1	0
Swimming		0	0	0	0	1
Tennis		1	0	0	0	0
Tobago		0	1	0	0	0
Wispa Bites		0	1	0	0	0

differentFrom		0	1	2	3	4
0		0	1	1	1	1
1		1	0	1	1	1
2		1	1	0	1	1
3		1	1	1	0	1
4		1	1	1	1	0

directlyright		0	1	2	3	4
0		0	1	0	0	0
1		0	0	1	0	0
2		0	0	0	1	0
3		0	0	0	0	1
4		0	0	0	0	0

nextto		0	1	2	3	4
0		0	1	0	0	0
1		1	0	1	0	0
2		0	1	0	1	0
3		0	0	1	0	1
4		0	0	0	1	0

2.1.5 Prover9 Results

Proof from prover9, proving line 2619 by using lines 502, 2615 and 2616.

```
502 -Florida(g3) | -Florida(g4). [resolve(174,c,77,a)].  
2615 Florida(g3). [resolve(2605,b,2486,a),merge(b)].  
2616 Florida(g4). [back_unit_del(2609),unit_del(b,2615)].  
2619 $F. [back_unit_del(502),unit_del(a,2615),unit_del(b,2616)].
```

Back-unit deletion at line 502 \implies Copy clause 502.

$$P \text{ or } Q \text{ and } \neg Q \implies P$$

$$\begin{aligned} Q &= \neg \text{Florida}(g4) \\ \neg Q &= \text{Florida}(g4) \\ \implies P &= \text{Florida}(g3) \end{aligned}$$

Unit-deletion at line 2615 \implies The first literal **a** has been removed because it was an instance of the negation clause 2615 (which is a unit clause).

Unit-deletion at line 2616 \implies The second literal **b** has been removed because it was an instance of the negation clause 2616 (which is a unit clause).

$$Q \text{ and } \neg Q \text{ or } R \implies R$$

$$\begin{aligned} Q &= \text{Florida}(g3) \\ \neg Q &= \neg \text{Florida}(g3) \\ \implies R &= \text{Florida}(g4) \end{aligned}$$

2.2 Ancient Civilizations - Medium Zebra Puzzle

Five girls are side by side talking about ancient civilizations. Each girl is passionate about a certain ancient civilization and knows a lot of things regarding a specific subject of it. Follow the clues to figure out which girl likes the Aztecs.

2.2.1 Clues

1. At the fourth position is the girl drinking Strawberry juice.
2. The girl drinking Strawberry juice is somewhere between the girl who likes Art and the girl that is drinking Apple juice, in that order.
3. Riley is next to the girl who likes the Roman civilization.
4. The girl wearing the Green shirt is drinking Apple juice.
5. At the fourth position is the girl that is interested in Religion.
6. The 12 years old girl is exactly to the right of the 11 years old girl.
7. At one of the ends is the girl who likes the Egyptian civilization.
8. The girl wearing the Black shirt is somewhere between the girl that likes Science and the girl wearing the Red shirt, in that order.
9. The girl wearing the Red shirt is somewhere to the left of the girl wearing the White shirt.
10. The girl drinking Cranberry juice likes the Greek civilization.
11. Ella is at the third position.
12. The 12 years old girl is next to the girl that is interested in Architecture.
13. Riley is exactly to the right of the girl that admires the Roman civilization.
14. The girl drinking Cranberry juice is somewhere between the 13 years old girl and the girl drinking Grapefruit juice, in that order.
15. At one of the ends is the girl that likes Science.
16. At one of the ends is the girl who loves Architecture.
17. Lily is exactly to the right of the girl who likes Economy.
18. Makayla is 10.
19. The girl interested in the Aztec civilization is somewhere between the girl interested in the Roman civilization and the girl interested in the Chinese civilization, in that order.
20. The youngest girl is at one of the ends.
21. At the second position is the girl that likes Art.

2.2.2 Human Thinking Solution

1. The second girl likes art, the fourth girl drinks strawberry juice, while the fifth girl drinks apple juice and has a green shirt. (**Clue 1, Clue 2, Clue 4 and Clue 21**)
2. The first girl likes science and the fifth girl likes architecture. (**Clue 8, Clue 15 and Clue 16**)
3. **Step 1 and Step 2** → The second girl is wearing a black shirt, the third girl is wearing a red shirt and the fourth girl is wearing a white shirt. (**Clue 8**)
4. **Step 3** → The first girl is wearing a blue shirt.
5. **Step 1 and Step 2** → The third girl is interested in Economy and the fourth girl is interested in religion. (**Clue 5**)
6. **Step 5** → The fourth girl is called Lily. (**Clue 17**)
7. **Step 6** → The first girl likes the Roman civilization, the second girl is called Riley and the third girl is called Ella. (**Clue 11 and Clue 13**)
8. **Step 7** → The fifth girl likes the Egyptian civilization. (**Clue 7**)
9. **Step 2** → The fourth girl is 12 years old. (**Clue 12**)
10. **Step 9** → The third girl is 11 years old. (**Clue 6**)
11. **Step 1** → The first girl is 13 years old, the second girl drinks cranberry juice and the third girl drinks grapefruit juice. (**Clue 14**)
12. **Step 11** → The first girl can only drink orange juice.
13. **Step 11** → The fifth girl is called Makayla and she is 10 years old. (**Clue 18 and Clue 20**)
14. **Step 13** → The first girl can only be named Julia and the second girl can only be 14 years old.
15. **Step 11** → The second girl likes the Greek civilization. (**Clue 10**)
16. **Step 7, Step 8, Step 15** → The third girl likes the Aztec civilization and the fourth girl likes the Chinese civilization. (**Clue 19**)

2.2.3 Mathematical Solution

	Girl #1	Girl #2	Girl #3	Girl #4	Girl #5
Shirt	Blue	Black	Red	White	Green
Name	Julia	Riley	Ella	Lily	Makayla
Civilization	Roman	Greek	Aztec	Chinese	Egyptian
Subject	Science	Art	Economy	Religion	Architecture
Age	13 years	14 years	11 years	12 years	10 years old
Juice	Orange	Cranberry	Grapefruit	Strawberry	Apple

2.2.4 Mace4 Results

This puzzle was solved using predicates.

Apple	0	0	0	0	1
Architecture	0	0	0	0	1
Art	0	1	0	0	0
Aztec	0	0	1	0	0
Black	0	1	0	0	0
Blue	1	0	0	0	0
Chinese	0	0	0	1	0
Cranberry	0	1	0	0	0
Economy	0	0	1	0	0
Egyptian	0	0	0	1	0
Eleven	0	0	1	0	0
Ella	0	0	1	0	0
Fourteen	0	1	0	0	0
Girl1	1	0	0	0	0
Girl2	0	1	0	0	0
Girl3	0	0	1	0	0
Girl4	0	0	0	1	0
Girl5	0	0	0	0	1
Grapefruit	0	0	1	0	0
Greek	0	1	0	0	0
Green	0	0	0	0	1
Julia	1	0	0	0	0
Lily	0	0	0	1	0
Makayla	0	0	0	0	1
Orange	1	0	0	0	0
Red	0	0	0	1	0
Religion	0	0	1	0	0
Riley	0	1	0	0	0

Roman		0	0	0	0	1
Science		1	0	0	0	0
Strawberry		0	1	0	0	0
Ten		0	0	0	1	0
Thirteen		0	0	0	0	1
Twelve		1	0	0	0	0
White		0	1	0	0	0
extremity		1	0	0	0	1

differentFrom		0	1	2	3	4
0		0	1	1	1	1
1		1	0	1	1	1
2		1	1	0	1	1
3		1	1	1	0	1
4		1	1	1	1	0

directlyright		0	1	2	3	4
0		0	1	0	0	0
1		0	0	1	0	0
2		0	0	0	1	0
3		0	0	0	0	1
4		0	0	0	0	0

nextto		0	1	2	3	4
0		0	1	0	0	0
1		1	0	1	0	0
2		0	1	0	1	0
3		0	0	1	0	1
4		0	0	0	1	0

inbetween		0	1	2	3	4
0		0	0	1	1	1
1		0	0	0	1	1
2		0	0	0	0	1
3		0	0	0	0	0
4		0	0	0	0	0

right	0	1	2	3	4
0	0	1	1	1	1
1	0	0	1	1	1
2	0	0	0	1	1
3	0	0	0	0	1
4	0	0	0	0	0

2.2.5 Prover9 Results

Proof from prover9, proving line 2776 by using lines 2775 and 160.

```
160 -inbetween(g2,g2).  [assumption].
2775 inbetween(g2,x).  [back_unit_del(582),unit_del(a,2774)].
2776 $F.  [resolve(2775,a,160,a)].
```

Resolve at line 2776 \Rightarrow resolve the first literal of clause 2775 with the first literal of clause 160.

2.3 Persian Rugs - Hard Zebra Puzzle

Five men, who just bought Persian rugs, are side by side. Each one bought a very specific Persian rug for his home. Try to find out which are the characteristics from Timothy's new rug.

2.3.1 Clues

1. The man who bought the 400 dollars rug is exactly to the left of the man who bought the rug with Birds.
2. Bryan is next to the client that got the Runner rug.
3. The Red rug's owner is somewhere to the left of the man who bought a rug for his Hall.
4. The buyer who got the most expensive rug is next to the buyer who got the Round rug.
5. The Yellow rug's owner is somewhere between the Oval rug's owner and the man that spent 800 dollars, in that order.
6. The man that bought the Red rug is exactly to the left of the man that bought a rug for his Office.
7. Garrett is next to the customer who spent 600 dollars.
8. Shane is somewhere to the right of the Orange rug's owner.
9. The cheapest rug has Stars on it.
10. At the fifth position is the client who bought the Rectangular rug.
11. The man who bought the 600 dollars rug is exactly to the left of the man who bought the rug with Roses.
12. Casey is somewhere to the right of the customer that got the Yellow rug.
13. The buyer who purchased the Round rug is exactly to the left of the man who spent 800 dollars.
14. The man that bought the cheapest rug is exactly to the left of the man that acquired the Square rug.
15. At the third position is the man that got a rug for his Bedroom.
16. The customer that bought the rug with Stars is exactly to the right of the customer that bought the Green rug.
17. Shane is next to the man who acquired the Round rug.
18. The man who got the rug with Birds is somewhere between the man who got the Yellow rug and the man who got the rug with Lotuses, in that order.
19. The customer that spent 800 dollars is somewhere to the right of the Red rug's owner.
20. The client that bought a rug for his Living room is exactly to the right of the client that bought a rug for his Library.

2.3.2 Human Thinking Solution

1. The man that bought the green rug, the man that bought the 400 dollars rug that has stars, and the man that bought the square rug that has birds are next to each other, in this order. **(Clue 9, Clue 10 and Clue 11)**
2. The man that bought the rug for his office can only be directly to the left of the man that bought the rug for his hall. **(Clue 15, Clue 20, Clue 6 and Clue 3)**
3. **Step 2** → The red rug is the third, the man that bought the rug for his office is the fourth and the man that bought the rug for his hall is the fifth. **(Clue 6)**
4. **Step 3** → The man that bought the rug for his living room is the first and the man that bought the rug for his library is the second. **(Clue 20 and Clue 15)**
5. The man that spent 1.200 dollars on the rug is directly to the left of the man that bought the round rug. **(Clue 13 and Clue 4)**
6. **Step 2 and Step 5** → The third man bought the 1.200 dollars red rug, the fourth man bought the round rug and the fifth man bought the 800 dollars rug. **(Clue 19)**
7. **Step 1 and Step 6** → The first man bought the green rug, the second man bought the 400 dollars rug that has stars and the third man bought the square rug that has birds.
8. **Step 7** → The second man bought the yellow rug. **(Clue 18)**
9. **Step 8** → The first man bought the oval rug. **(Clue 5)**
10. **Step 9** → The second man bought the runner rug.
11. **Step 6 and Step 7** → The fourth man spent 600 dollars on the rug and the fifth man bought the rug that has roses. **(Clue 11)**
12. **Step 11** → The first man bought the 1.000 dollars rug.
13. **Step 7 and Step 11** → The fourth man bought the rug that has lotuses. **(Clue 18)**
14. **Step 13** → The first man bought the rug that has diamonds.
15. **Step 7, Step 8 and Step 3** → The fourth man bought the orange rug and the fifth man is called Shane. **(Clue 8)**
16. **Step 15** → The fifth man bought the blue rug.
17. **Step 15** → The third man is called Garrett. **(Clue 7)**
18. **Step 15 and Step 17** → The fourth man is called Casey. **(Clue 12)**
19. **Step 15 and Step 10** → The first man is called Bryan. **(Clue 2)**
20. **Step 19** → The second man is called Timothy.

2.3.3 Mathematical Solution

	Client #1	Client #2	Client #3	Client #4	Client #5
Color	Green	Yellow	Red	Orange	Blue
Name	Bryan	Timothy	Garrett	Casey	Shane
Type	Oval	Runner	Square	Round	Rectangular
Symbols	Diamonds	Stars	Birds	Lotuses	Roses
Price	1000 dollars	400 dollars	1200 dollars	600 dollars	800 dollars
Place	Library	Living Room	Bedroom	Office	Hall

2.3.4 Mace4 Results

This puzzle was solved using predicates.

Bedroom	0	0	1	0	0
Birds	0	0	1	0	0
Blue	0	0	0	0	1
Bryan	1	0	0	0	0
Casey	0	0	0	1	0
Diamonds	1	0	0	0	0
EightHundred	0	0	0	0	1
FourHundred	0	1	0	0	0
Garrett	0	0	1	0	0
Green	1	0	0	0	0
Hall	0	0	0	0	1
Library	1	0	0	0	0
LivingRoom	0	1	0	0	0
Lotuses	0	0	0	1	0
Man1	1	0	0	0	0
Man2	0	1	0	0	0
Man3	0	0	1	0	0
Man4	0	0	0	1	0
Man5	0	0	0	0	1
Office	0	0	0	1	0
OneThousan	1	0	0	0	0
Orange	0	0	0	1	0
Oval	1	0	0	0	0
Rectangular	0	0	0	0	1
Red	0	0	1	0	0
Roses	0	0	0	0	1
Round	0	0	0	1	0
Runner	0	1	0	0	0

Shane		0	0	0	0	1
SixHundred		0	0	0	1	0
Square		0	0	1	0	0
Stars		0	1	0	0	0
Timothy		0	1	0	0	0
TwelveHundred		0	0	1	0	0
Yellow		0	1	0	0	0
extremity		1	0	0	0	1

differentFrom	0	1	2	3	4
0	0	1	1	1	1
1	1	0	1	1	1
2	1	1	0	1	1
3	1	1	1	0	1
4	1	1	1	1	0

directlyright	0	1	2	3	4
0	0	1	0	0	0
1	0	0	1	0	0
2	0	0	0	1	0
3	0	0	0	0	1
4	0	0	0	0	0

nextto	0	1	2	3	4
0	0	1	0	0	0
1	1	0	1	0	0
2	0	1	0	1	0
3	0	0	1	0	1
4	0	0	0	1	0

right	0	1	2	3	4
0	0	1	1	1	1
1	0	0	1	1	1
2	0	0	0	1	1
3	0	0	0	0	1
4	0	0	0	0	0

2.3.5 Prover9 Results

Proof from prover9, proving line 11474 by using lines 11471, 10628 and 126.

```
126 -right(d2,d1). [assumption].  
10628 EightHundred(d1). [resolve(10619,b,10600,b),merge(c),unit_del(b,131)].  
11471 -EightHundred(x) | right(d2,x). [resolve(11364,a,191,b)].  
11474 $F. [resolve(11471,a,10628,a),unit_del(a,126)].
```

Resolve at line 11474 \Rightarrow resolve the first literal of clause 11471 with the first literal of clause 10628.

Unit-deletion at line 126 \Rightarrow The first literal **a** has been removed because it was an instance of the negation clause 126 (which is a unit clause).

2.4 Travel Agency - Hard Zebra Puzzle

Five women are side by side booking a trip in a travel agency. Each one is traveling to a different country. Follow the clues to discover where each one is going.

2.4.1 Clues

1. The Singer is at the third position.
2. The woman traveling for 20 days is somewhere between the woman who is going to Peru and the owner of the Blue purse, in that order.
3. Ana is exactly to the left of the Biologist.
4. The 32 years old is going to see the Sahara.
5. The owner of the White purse is exactly to the right of the woman traveling to visit Machu Picchu.
6. Glenda is somewhere to the right of the woman who has the Green purse.
7. The person wearing the White purse is somewhere between the 30 years old woman and the owner of the Blue purse, in that order.
8. The 24 years old woman is going to visit an Aztec pyramid.
9. The woman wearing the White purse is somewhere to the left of the youngest woman.
10. The traveler going to Italy is exactly to the right of the woman traveling for 20 days.
11. The person who is going to travel for 25 days has the Red purse.
12. The Judge is in the first position.
13. The Nurse is exactly to the right of the woman who is going to travel for 20 days.
14. The Hostess is somewhere between Lara and the woman who has the Blue purse, in that order.
15. In the second position is the woman that is going to travel for 15 days.
16. Rose has the Green purse.
17. The woman who is traveling for less than a week is exactly to the left of the 32 years old woman.
18. The person traveling for 5 days is 28.
19. The Blue purse owner is somewhere between the 30 years old woman and the owner of the Yellow purse, in that order.

2.4.2 Human Thinking Solution

1. The third woman is traveling for 20 days, the fourth woman has the blue purse and the fifth woman has the yellow purse. (**Clue 15, Clue 2 and Clue 19**)
2. The first woman is called Lara, the second woman is a hostess. (**Clue 12, Clue 1 and Clue 14**)
3. The fourth woman is a nurse. (**Clue 13 and Clue 1**)
4. **Step 3** → The fifth woman can only be a biologist.
5. **Step 1 ans Step 2** → The first woman is traveling for 25 days and has a red purse. (**Clue 11, Clue 1 and Clue 15**)
6. **Step 1 and Step 5** → The fourth woman is 28 years old and is traveling for 5 days, while the fifth woman is 32 years old and is traveling to Egypt. (**Clue 17, Clue 18, Clue 4 and Clue 15**)
7. **Step 6** → The fifth woman can only be travelling for 10 days.
8. **Step 4** → The fourth woman is called Ana. (**Clue 3**)
9. **Step 1 and Step 6** → The first woman is 30 years old, the second woman has the white purse and the third woman is 24 years old. (**Clue 7 and Clue 9**)
10. **Step 9** → The third woman can only have the green purse.
11. **Step 10** → The third woman is called Rose. (**Clue 14**)
12. **Step 9** → The second woman can only be 26 years old.
13. **Step 9** → The first woman is going to Peru. (**Clue 5**)
14. **Step 9** → The third woman is going to Mexico. (**Clue 8**)
15. **Step 1** → The fourth woman is travelling to Italy. (**Clue 10**)
16. **Step 15** → The second woman can only be travelling to China.
17. **Step 10 and Step 8** → The fifth woman is called Glenda. (**Clue 6**)
18. **Step 17** → The second woman can only be called Jessie.

2.4.3 Mathematical Solution

	Woman #1	Woman #2	Woman #3	Woman #4	Woman #5
Purse	Red	White	Green	Blue	Yellow
Name	Lara	Jessie	Rose	Ana	Glenda
Age	30 years	26 years	24 years	28 years	32 years
Profession	Judge	Hostess	Singer	Nurse	Biologist
Country	Peru	China	Mexico	Italy	Egypt
Duration	25 days	15 days	20 days	5 days	10 days

2.4.4 Mace4 Results

This puzzle was solved using predicates.

Ana	0	0	0	1	0
Biologist	0	0	0	0	1
Blue	0	0	0	1	0
China	0	1	0	0	0
Egypt	0	0	0	0	1
FifteenDays	0	1	0	0	0
FiveDays	0	0	0	1	0
Glenda	0	0	0	0	1
Green	0	0	1	0	0
Hostess	0	1	0	0	0
Italy	0	0	0	1	0
Jessie	0	1	0	0	0
Judge	1	0	0	0	0
Lara	1	0	0	0	0
Mexico	0	0	1	0	0
Nurse	0	0	0	1	0
Peru	1	0	0	0	0
Red	1	0	0	0	0
Rose	0	0	1	0	0
Singer	0	0	1	0	0
TenDays	0	0	0	0	1
Thirty	1	0	0	0	0
ThirtyTwo	0	0	0	0	1
TwentyDays	0	0	1	0	0
TwentyEight	0	0	0	1	0
TwentyFiveDays	1	0	0	0	0
TwentyFour	0	0	1	0	0
TwentySix	0	1	0	0	0

White		0	1	0	0	0
Woman1		1	0	0	0	0
Woman2		0	1	0	0	0
Woman3		0	0	1	0	0
Woman4		0	0	0	1	0
Woman5		0	0	0	0	1
Yellow		0	0	0	0	1
extremity		1	0	0	0	1

differentFrom		0	1	2	3	4
0		0	1	1	1	1
1		1	0	1	1	1
2		1	1	0	1	1
3		1	1	1	0	1
4		1	1	1	1	0

directlyright		0	1	2	3	4
0		0	1	0	0	0
1		0	0	1	0	0
2		0	0	0	1	0
3		0	0	0	0	1
4		0	0	0	0	0

nextto		0	1	2	3	4
0		0	1	0	0	0
1		1	0	1	0	0
2		0	1	0	1	0
3		0	0	1	0	1
4		0	0	0	1	0

right		0	1	2	3	4
0		0	1	1	1	1
1		0	0	1	1	1
2		0	0	0	1	1
3		0	0	0	0	1
4		0	0	0	0	0

2.4.5 Prover9 Results

Proof from prover9, proving line 14449 by using lines 14448, 12412 and 130.

```
130 -right(d1,d1). [assumption].  
12412 Blue(d1). [resolve(12411,b,12044,b),merge(b)].  
14448 -Blue(x) | right(d1,x). [resolve(14338,a,13504,a)].  
14449 $F. [resolve(14448,a,12412,a),unit_del(a,130)].
```

Resolve at line 14448 \Rightarrow resolve the first literal of clause 14448 with the first literal of clause 12412.

Unit-deletion at line 130 \Rightarrow The first literal **a** has been removed because it was an instance of the negation clause 130 (which is a unit clause).

2.5 Secret Agents - Very Hard Zebra Puzzle

Five spies are side by side in a briefing room getting instructions for their next missions. Each spy has an exclusive accessory, a special skill and is going to travel to a determined country. Which one is a martial arts expert?

2.5.1 Clues

1. Austin is next to the agent wearing the Black tie.
2. The master of Disguise is exactly to the right of the agent that has a spy Umbrella.
3. The 35 years old agent is going to a mission on Tripoli.
4. James is the youngest secret agent.
5. The agent that is going to Australia is next to the agent that is specialized in Parkour.
6. James is exactly to the right of the agent that has a special Clock.
7. The spy that has an unique Umbrella is between the 40 years old agent and Austin, in that order.
8. Stan is next to the agent that is going to Asia.
9. Sterling is at one of the ends.
10. The man wearing the Red tie is 40 years old.
11. The spy that is going to South America is exactly to the left of the 45 years old spy.
12. Jason is exactly to the left of Austin.
13. The Driver expert is next to the 30 years old man.
14. The 35 years old agent is next to the agent that is going to Sydney.
15. The agent with advanced knowledge of Hacking is exactly to the left of the 35 years old man.
16. The spy wearing the Purple tie is next to the Geek spy.
17. Austin is 30.
18. The agent that has a special Phone is exactly to the left of the agent that is going to Africa.
19. The agent wearing a spy Ring is somewhere to the right of the agent wearing the Purple tie.
20. At the second position is the spy wearing the Green tie.
21. The spy that is going to Australia is exactly to the right of the 30 years old spy.

2.5.2 Human Thinking Solution

1. The first agent is 40 years old, the second agent is called Jason and has a unique umbrella, the third agent is called Austin and he is 30 years old, the fourth agent has advanced knowledge in computer hacking and is going to Australia, the fifth agent is 35 years old and is going to Libya. (**Clue 12, Clue 17, Clue 21, Clue 3, Clue 14, Clue 15 and Clue 7**)
2. **Step 1** → The fourth agent is wearing the black tie. (**Clue 20 and Clue 1**)
3. **Step 1** → The third agent is a master of disguise. (**Clue 2**)
4. **Step 1** → The fourth agent is called James and he is 25 years old. (**Clue 4**)
5. **Step 4** → The second agent can only be 45 years old.
6. **Step 1 and Step 3** → The fifth agent is specialized in parkour. (**Clue 5**)
7. **Step 4** → The third agent has a special clock. (**Clue 6**)
8. **Step 1 and Step 4** → The first agent is called Stan and the second agent is going to Russia. (**Clue 8**)
9. **Step 8** → The fifth agent is called Sterling. (**Clue 9**)
10. **Step 1** → The first agent is wearing the red tie. (**Clue 10**)
11. **Step 5** → The first agent is going to Brazil. (**Clue 11**)
12. **Step 1** → The second agent is an expert driver. (**Clue 13**)
13. **Step 12** → The first agent can only be specialized in martial arts.
14. **Step 11** → The third agent can only be going to Germany.
15. **Step 1** → The fourth agent has a special phone. (**Clue 18**)
16. **Step 1, Step 7, Step 15, Step 10, Step 20 and Step 2** → The third agent is wearing the purple tie and the fifth agent is wearing the spy ring. (**Clue 19**)
17. **Step 16** → The first agent can only have a special pen and the fifth agent can only be wearing the blue tie.

2.5.3 Mathematical Solution

	Agent #1	Agent #2	Agent #3	Agent #4	Agent #5
Tie	Red	Green	Purple	Black	Blue
Name	Stan	Jason	Austin	James	Sterling
Country	Brazil	Russia	Germany	Australia	Libya
Accessory	Pen	Umbrella	Clock	Phone	Ring
Skill	Martial Arts	Driving	Disguise	Computer Hacking	Parkour
Age	40 years	45 years	30 years	25 years	35 years

2.5.4 Mace4 Results

This puzzle was solved using predicates.

Agent1		1	0	0	0	0
Agent2		0	1	0	0	0
Agent3		0	0	1	0	0
Agent4		0	0	0	1	0
Agent5		0	0	0	0	1
Austin		0	0	1	0	0
Australia		0	0	0	1	0
Black		0	0	0	1	0
Blue		0	0	0	0	1
Brazil		1	0	0	0	0
Clock		0	0	1	0	0
ComputerHacking		0	0	0	1	0
Disguise		0	0	1	0	0
Driving		0	1	0	0	0
Forty		1	0	0	0	0
FortyFive		0	1	0	0	0
Germany		0	0	1	0	0
Green		0	1	0	0	0
James		0	0	0	1	0
Jason		0	1	0	0	0
Libya		0	0	0	0	1
MartialArts		1	0	0	0	0
Parkour		0	0	0	0	1
Pen		1	0	0	0	0
Phone		0	0	0	1	0
Purple		0	0	1	0	0
Red		1	0	0	0	0
Ring		0	0	0	0	1

Russia		0	1	0	0	0
Stan		1	0	0	0	0
Sterling		0	0	0	0	1
Thirty		0	0	1	0	0
ThirtyFive		0	0	0	0	1
TwentyFive		0	0	0	1	0
Umbrella		0	1	0	0	0
extremity		1	0	0	0	1

differentFrom		0	1	2	3	4
0		0	1	1	1	1
1		1	0	1	1	1
2		1	1	0	1	1
3		1	1	1	0	1
4		1	1	1	1	0

directlyright		0	1	2	3	4
0		0	1	0	0	0
1		0	0	1	0	0
2		0	0	0	1	0
3		0	0	0	0	1
4		0	0	0	0	0

nextto		0	1	2	3	4
0		0	1	0	0	0
1		1	0	1	0	0
2		0	1	0	1	0
3		0	0	1	0	1
4		0	0	0	1	0

right		0	1	2	3	4
0		0	1	1	1	1
1		0	0	1	1	1
2		0	0	0	1	1
3		0	0	0	0	1
4		0	0	0	0	0

2.5.5 Prover9 Results

Proof from prover9, proving line 24218 by using lines 24217, 18365 and 99.

```
99 -directlyright(d3,d1). [assumption].  
18365 Austin(d1). [resolve(18364,b,17649,b),merge(b)].  
24217 -Austin(x) | directlyright(d3,x). [resolve(24176,a,187,a)].  
24218 $F. [resolve(24217,a,18365,a),unit_del(a,99)].
```

Resolve at line 24218 \Rightarrow resolve the first literal of clause 24217 with the first literal of clause 18365.

Unit-deletion at line 99 \Rightarrow The first literal **a** has been removed because it was an instance of the negation clause 99 (which is a unit clause).

2.6 Who likes dolphins? - Easy Zebra Puzzle

There are four houses in a line. Each house has a number, a color and an owner. Each owner has a nationality, a favourite animal and sport. Find out whose favourite animals are dolphins.

2.6.1 Clues

1. The second house is Black.
2. The British lives in the first house.
3. There are two houses between the person who likes Bowling and the person who likes Swimming.
4. The American live directly to the left of the person who likes Turtles.
5. There are two houses between the person who likes Horses and the person who likes Butterflies on the right.
6. The person who likes Bowling lives somewhere to the right of the person who likes Tennis.
7. There is one house between the Irish and the person who likes Handball on the left.
8. There is one house between the person who likes Horses and the Red house on the right.
9. There is one house between the person who likes Handball and the White house on the right.

2.6.2 Human Thinking Solution

1. The person who likes Horses lives in the first house and the person who likes Butterflies lives in the last house. (**Clue 5**)
2. The person who likes Tennis lives in the second or third house and the person who likes Bowling lives in the first or last house. The person who likes Handball lives in the second or third house. (**Clue 3**)
3. **Step 2** → The person who likes Bowling lives in the last house and the person who likes Swimming lives in the first house. (**Clue 6 and Clue 3**)
4. **Step 2** → The person who likes Handball lives in the second house and the person who lives in the last house is Irish. The American person lives in the second or third house. (**Clue 7**)
5. **Step 4** → The American Person lives in the second house and the person who likes Turtles lives in the third house. (**Clue 4**)
6. **Step 5** → The person who likes dolphins lives in the second house.

2.6.3 Mathematical Solution

	House #1	House #2	House #3	House #4
Color	Blue	Black	Red	White
Nationality	British	American	Canadian	Irish
Animal	Horses	Dolphins	Turtles	Butterflies
Sport	Swimming	Handball	Tennis	Bowling

2.6.4 Mace4 Results

This puzzle was solved using predicates.

American	0	1	0	0
Black	0	1	0	0
Blue	1	0	0	0
Bowling	1	0	0	0
British	1	0	0	0
Butterflies	0	0	0	1
Canadian	0	0	1	0
Dolphins	0	1	0	0
Handball	0	1	0	0
Horses	1	0	0	0
HouseFour	0	0	0	1
HouseOne	1	0	0	0
HouseThree	0	0	1	0
HouseTwo	0	1	0	0
Irish	0	0	0	1
Red	0	1	0	0
Swimming	0	0	1	0
Tennis	0	0	1	0
Turtles	0	0	1	0
White	0	0	0	1

differentFrom	0	1	2	3
0	0	1	1	1
1	1	0	1	1
2	1	1	0	1
3	1	1	1	0

inbetween	0	1	2	3
0	0	0	0	1
1	0	0	0	0
2	0	0	0	0
3	1	0	0	0

middleneighbour	0	1	2	3
0	0	0	1	0
1	0	0	0	1
2	1	0	0	0
3	0	1	0	0

right	0	1	2	3
0	0	1	1	1
1	0	0	1	1
2	0	0	0	1
3	0	0	0	0

rightneighbour	0	1	2	3
0	0	1	0	0
1	0	0	1	0
2	0	0	0	1
3	0	0	0	0

2.6.5 Prover9 Results

Proof from prover9, proving line 514 by using lines 513, 490 and 107.

```

107 -inbetween(h4,h2).  [assumption].
490 Bowling(h4).  [resolve(470,b,476,b),merge(c),unit_del(b,106)].
513 -Bowling(x) | inbetween(x,h2).  [resolve(506,a,137,b)].
514 $F.  [resolve(513,a,490,a),unit_del(a,107)].

```

Resolve at line 514 \Rightarrow resolve the first literal of clause 513 with the first literal of clause 490.

Unit-deletion at line 107 \Rightarrow The first literal **a** has been removed because it was an instance of the negation clause 107 (which is a unit clause).

2.7 Hardware Store - Medium Zebra Puzzle

Five men are next to each other at a hardware store. Each one is buying a specific tool and getting a discount. Solve this logic problem to discover these information.

2.7.1 Clues

1. The man wearing the Yellow shirt is somewhere to the left of the Mechanic.
2. Billy is buying a tool with 20 percent off.
3. The guy buying a tool with 15 percent discount is at one of the ends.
4. The 45 years old man is somewhere between the Plumber and the 35 years old man, in that order.
5. The one buying a tool with the biggest discount is next to the 50 years old man.
6. Dennis is wearing the Green shirt.
7. The 50 years old man is in the middle.
8. Larry is next to the man that is buying a tool with the smallest discount.
9. The one buying a Hammer is somewhere between the man buying the Wrench and the man buying the Screwdriver, in that order.
10. The Technician is at the third position.
11. The Plumber is next to the man wearing the Green shirt.
12. The oldest man is buying a Drill.
13. The guy wearing the White shirt is somewhere to the right of the man wearing the Blue shirt.
14. The man buying a Hammer is somewhere between the one who is getting 15 percent discount and the one buying a Saw, in that order.
15. George is next to the Electrician.
16. The man buying tools with 5 percent and 25 percent off are next to each other.
17. Philip is buying a Saw.
18. At the second position is the man buying a Drill.
19. The man wearing the Yellow shirt is next to the one buying a Saw.
20. The guy getting the best discount is somewhere between the 55 years old man and the man buying a tool with 10 percent off, in that order.
21. The man wearing the Red shirt is next to the man buying a tool with 15

2.7.2 Human Thinking Solution

1. The second man is 55 years old. (**Clue 18 and Clue 12**)
2. **Step 1** → The fourth man is 45 years old and the fifth person is 35 years old. (**Clue 4 and Clue 7**)
3. **Step 2** → The first man is 40 years old.
4. The first man has a 15 percent discount. (**Clue 3 and Clue 14**)
5. **Step 4** → The second man is wearing the red shirt. (**Clue 21**)
6. **Step 2** → The first man is called Dennis and he is wearing the green shirt, while the second man is a plumber. (**Clue 10, Clue 6 and Clue 4**)
7. **Step 10 and Step 6** → The fourth man is wearing the yellow shirt and the fifth man is a mechanic. (**Clue 1**)
8. **Step 5, Step 6 and Step 7** → The third man is wearing the blue shirt and the fifth man is wearing the white shirt. (**Clue 13**)
9. **Step 7 and Step 4** → The fifth man is called Philip and he is buying a saw. (**Clue 19, Clue 17 and Clue 14**)
10. The fourth man has a 25 percent discount. (**Clue 7, Clue 16 and Clue 20**)
11. **Step 1 and Step 10** → The fifth man has a 10 percent discount. (**Clue 20**)
12. **Step 10 and Step 11** → The third man has a 5 percent discount. (**Clue 16**)
13. **Step 12** → The second man can only have a 20 percent discount.
14. **Step 13** → The second man is called Billy. **Clue 2**
15. **Step 14 and Step 2** → The fourth man is called Larry. (**Clue 8**)
16. **Step 15** → The third man can only be called George.
17. **Step 16 and Step 6** → The fourth man is an electrician. (**Clue 15**)
18. **Step 17** → The first man can only be an engineer.
19. **Step 9** → The first man is buying a wrench, the second man is buying a hammer and the fourth man is buying a screwdriver. (**Clue 9, Clue 14 and Clue 18**)

2.7.3 Mathematical Solution

	Man #1	Man #2	Man #3	Man #4	Man #5
Shirt	Green	Red	Blue	Yellow	White
Name	Dennis	Billy	George	Larry	Philip
Tool	Wrench	Drill	Hammer	Screwdriver	Saw
Discount	15%	20%	5%	25%	10%
Profession	Engineer	Plumber	Technician	Electrician	Mechanic
Age	40 years	55 years	50 years	45 years	35 years

2.7.4 Mace4 Results

This puzzle was solved using predicates.

Billy		0	1	0	0	0
Blue		0	0	1	0	0
Dennis		1	0	0	0	0
Drill		0	1	0	0	0
Electrician		0	0	0	1	0
Engineer		1	0	0	0	0
FifteenPercent		1	0	0	0	0
Fifty		0	0	1	0	0
FiftyFive		0	1	0	0	0
FivePercent		0	0	1	0	0
Forty		1	0	0	0	0
FortyFive		0	0	0	1	0
George		0	0	1	0	0
Green		1	0	0	0	0
Hammer		0	0	1	0	0
Larry		0	0	0	1	0
Man1		1	0	0	0	0
Man2		0	1	0	0	0
Man3		0	0	1	0	0
Man4		0	0	0	1	0
Man5		0	0	0	0	1
Mechanic		0	0	0	0	1
Philip		0	0	0	0	1

Plumber		0	1	0	0	0
Red		0	1	0	0	0
Saw		0	0	0	0	1
Screwdriver		0	0	0	1	0
Technician		0	0	1	0	0
TenPercent		0	0	0	0	1
ThirtyFive		0	0	0	0	1
TwentyFivePercent		0	0	0	1	0
TwentyPercent		0	1	0	0	0
White		0	0	0	0	1
Wrench		1	0	0	0	0
Yellow		0	0	0	1	0
extremity		1	0	0	0	1

differentFrom		0	1	2	3	4
0		0	1	1	1	1
1		1	0	1	1	1
2		1	1	0	1	1
3		1	1	1	0	1
4		1	1	1	1	0

directlyright		0	1	2	3	4
0		0	1	0	0	0
1		0	0	1	0	0
2		0	0	0	1	0
3		0	0	0	0	1
4		0	0	0	0	0

inbetween		0	1	2	3	4
0		0	0	1	1	1
1		0	0	0	1	1
2		0	0	0	0	1
3		0	0	0	0	0
4		0	0	0	0	0

nextto	0	1	2	3	4
0	0	1	0	0	0
1	1	0	1	0	0
2	0	1	0	1	0
3	0	0	1	0	1
4	0	0	0	1	0

right	0	1	2	3	4
0	0	1	1	1	1
1	0	0	1	1	1
2	0	0	0	1	1
3	0	0	0	0	1
4	0	0	0	0	0

2.7.5 Prover9 Results

Proof from prover9, proving line 1022 by using lines 413, 1018 and 1019.

```
413 -Hammer(m3) | -Hammer(m4).  [resolve(211,c,75,a)].  
1018 Hammer(m3).  [resolve(1015,b,999,a),merge(b)].  
1019 Hammer(m4).  [back_unit_del(849),unit_del(b,1018)].  
1022 $F.  [back_unit_del(413),unit_del(a,1018),unit_del(b,1019)].
```

Back-unit deletion at line 413 \Rightarrow Copy clause 413.

P or Q and $\neg Q \rightarrow P$

$Q = \neg \text{Hammer}(m4)$
 $\neg Q = \text{Hammer}(m4)$
 $\rightarrow P = \neg \text{Hammer}(m3)$

Unit-deletion at line 1018 \Rightarrow The first literal **a** has been removed because it was an instance of the negation clause 1018 (which is a unit clause).

Unit-deletion at line 1019 \Rightarrow The second literal **b** has been removed because it was an instance of the negation clause 1019 (which is a unit clause).

Q and $\neg Q$ or $R \rightarrow R$

$Q = \text{Hammer}(m3)$
 $\neg Q = \neg \text{Hammer}(m3)$
 $\rightarrow R = \text{Hammer}(m4)$

2.8 Black Friday - Hard Zebra Puzzle

Five friends are side by side drinking juice and talking about the deals they got during the Black Friday sales. Follow the clues to find out who bought the laptop.

2.8.1 Clues

1. The man drinking the Orange juice is exactly to the right of the man who got the 70 percent discount.
2. Keith is 45 years old.
3. The man who bought the TV is exactly to the left of the man wearing the Red shirt.
4. At the third position is the man who got the 50 percent discount.
5. Keith is next to the man wearing the White shirt.
6. The 25-year-old man is somewhere between the 35-year-old man and the 40-year-old man, in that order.
7. The man drinking Apple juice bought the Smartphone.
8. The 30-year-old man is exactly to the left of the man that bought the Beard trimmer.
9. Sean is the youngest.
10. The man that got the 40 percent discount is exactly to the right of the man who bought the Beard trimmer.
11. Keith is next to the 35-year-old man.
12. Eugene is 40 years old.
13. Sean is wearing the Black shirt.
14. At the fourth position is the man who got the biggest discount.
15. Dustin got 60 percent off.
16. The man drinking the Lemon juice is exactly to the right of the man drinking the Grape juice.
17. Keith bought a Game console.
18. The man who got the 80 percent discount is exactly to the left of the man who is wearing the Blue shirt.
19. The man drinking Grape juice bought the Beard trimmer.
20. The man wearing the Black shirt is somewhere to the right of Keith.
21. The man that bought the Smartphone is next to the man wearing the Black shirt.

2.8.2 Human Thinking Solution

1. The fifth man is wearing the blue shirt. (**Clue 18**)
2. The first man has a 10 percent discount, the second man drinks orange juice, the fourth man bought the beard trimmer and is drinking grape juice and the fifth man has a 40 percent discount and is drinking lemon juice. (**Clue 4, Clue 14, Clue 19, Clue 10, Clue 16 and Clue 1**)
3. **Step 2** → The second man can only have the 60 percent discount.
4. **Step 3** → The second man is called Dustin. (**Clue 15**)
5. **Step 2 and Step 4** → The third man bought a smartphone and is drinking apple juice, the fourth man is called Sean, he is 25 years old and is wearing the black shirt and the fifth man is called Eugene and he is 40 years old. (**Clue 21, Clue 7, Clue 9, Clue 13, Clue 16 and Clue 12**)
6. **Step 4, Step 5 and Step 2** → The first man is called Keith, he is 45 years old and he bought a game console. (**Clue 17 and Clue 20**)
7. **Step 6** → The third man can only be named Hank.
8. **Step 6** → The second man is 35 years old. (**Clue 11**)
9. **Step 8** → The third man can only be 30 years old.
10. **Step 6, Step 5 and Step 2** → The second man bought the TV and the third man is wearing the red shirt. (**Clue 3**)
11. **Step 5** → The first man can only be drinking cranberry juice.
12. **Step 10** → The fifth man can only be buying the laptop.
13. **Step 6** → The second man can only be wearing the white shirt. (**Clue 5**)
14. **Step 13** → The first man can only be wearing the green shirt.

2.8.3 Mathematical Solution

	Man #1	Man #2	Man #3	Man #4	Man #5
Shirt	Green	White	Red	Black	Blue
Name	Keith	Dustin	Hank	Sean	Eugene
Deal	Game Console	TV	Smartphone	Beard Trimmer	Laptop
Discount	70%	60%	50%	80%	40%
Age	45 years	35 years	30 years	25 years	40 years
Juice	Cranberry	Orange	Apple	Grape	Lemon

2.8.4 Mace4 Results

This puzzle was solved using predicates.

Apple	0	0	1	0	0
BeardTrimmer	0	0	0	1	0
Black	0	0	0	1	0
Blue	0	0	0	0	1
Cranberry	1	0	0	0	0
Dustin	0	1	0	0	0
EightyPercent	0	0	0	1	0
Eugene	0	0	0	0	1
FiftyPercent	0	0	1	0	0
Forty	0	0	0	0	1
FortyFive	1	0	0	0	0
FortyPercent	0	0	0	0	1
GameConsole	1	0	0	0	0
Grape	0	0	0	1	0
Green	1	0	0	0	0
Hank	0	0	1	0	0
Keith	1	0	0	0	0
Laptop	0	0	0	0	1
Lemon	0	0	0	0	1
Man1	1	0	0	0	0
Man2	0	1	0	0	0
Man3	0	0	1	0	0
Man4	0	0	0	1	0
Man5	0	0	0	0	1
Orange	0	1	0	0	0
Red	0	0	1	0	0
Sean	0	0	0	1	0
SeventyPercent	1	0	0	0	0

SixtyPercent		0	1	0	0	0
Smartphone		0	0	1	0	0
TV		0	1	0	0	0
Thirty		0	0	1	0	0
ThirtyFive		0	1	0	0	0
TwentyFive		0	0	0	1	0
White		0	1	0	0	0
extremity		1	0	0	0	1

differentFrom		0	1	2	3	4
0		0	1	1	1	1
1		1	0	1	1	1
2		1	1	0	1	1
3		1	1	1	0	1
4		1	1	1	1	0

directlyright		0	1	2	3	4
0		0	1	0	0	0
1		0	0	1	0	0
2		0	0	0	1	0
3		0	0	0	0	1
4		0	0	0	0	0

inbetween		0	1	2	3	4
0		0	0	1	1	1
1		0	0	0	1	1
2		0	0	0	0	1
3		0	0	0	0	0
4		0	0	0	0	0

nextto		0	1	2	3	4
0		0	1	0	0	0
1		1	0	1	0	0
2		0	1	0	1	0
3		0	0	1	0	1
4		0	0	0	1	0

right	0	1	2	3	4
0	0	1	1	1	1
1	0	0	1	1	1
2	0	0	0	1	1
3	0	0	0	0	1
4	0	0	0	0	0

2.8.5 Prover9 Results

Proof from prover9, proving line 11693 by using lines 11692, 11352 and 87.

```
87 -directlyright(m2,m2). [assumption].
11352 BeardTrimmer(m2). [resolve(11350,b,11337,b),merge(b)].
11692 -BeardTrimmer(x) | directlyright(x,m2). [resolve(11515,a,233,a)].
11693 $F. [resolve(11692,a,11352,a),unit_del(a,87)].
```

Resolve at line 11693 \Rightarrow resolve the first literal of clause 11692 with the first literal of clause 11352.

Unit-deletion at line 87 \Rightarrow The first literal **a** has been removed because it was an instance of the negation clause 87 (which is a unit clause).

2.9 Home Remodeling - Hard Zebra Puzzle

Five couples are side by side buying things to remodel their houses. Each couple chose a room to remodel. Figure out how much they are willing to spend and where they got the inspiration for the remodeling.

2.9.1 Clues

1. The couple guiding the Black cart is somewhere between the couple inspired by a Website and the couple guiding the Red cart, in that order.
2. Amelia has a 15,000 dollars budget.
3. The partners who will remodel their Bedroom is next to the couple who will remodel their Living room.
4. Cassie is at one of the ends.
5. Benny is next to the partners steering the Black shopping cart.
6. At the first position is the couple who has the biggest budget.
7. The couple inspired by a Magazine is exactly to the right of the couple that will remodel their Living room.
8. Jeanne is immediately before the partners that are going to remodel their Bathroom.
9. Toni was inspired by a interior design Exhibition.
10. The couple guiding the Green cart is going to remodel their Bathroom.
11. At the fifth position is the couple who got an inspiration from a Friend's house.
12. Benny is next to Sylvia and her husband.
13. Vince is exactly to the right of the couple that will remodel their Kitchen.
14. Myles and his wife will remodel their Bathroom.
15. The couple inspired by a TV show is somewhere between the couple that has the smallest budget and the couple that has a 9,000 dollars budget, in that order.
16. The partners steering the the Red cart is somewhere to the left of the couple steering the White cart.
17. The couple that will remodel their Kitchen is immediately before the couple that will remodel their Bedroom.
18. The partners that have a 12,000 dollars budget is somewhere to the right of the partners guiding the Green cart.
19. Darrel is exactly to the right of the couple who has a 9,000 dollars budget.

2.9.2 Human Thinking Solution

1. At the first position, the couple's budget is 15.000 dollars and the wife is Amelia. (**Clue 6 and Clue 2**)
2. The couples that are remodeling the kitchen, bedroom and living room are right next to each other, exactly in this order. (**Clue 17 and Clue 3**)
3. The couple that is remodeling the bathroom cannot be positioned at neither of the ends. (**Clue 8 and Clue 18**)
4. **Step 2 and Step 3** → The couple that is remodeling the bathroom is either the second, or the fourth.
5. **Step 1 and Step 4** → The couple that is remodeling the bathroom can only be the fourth.
6. **Step 2 and Step 5** → The couples that are remodeling the kitchen, bedroom and living room are the first, second and third.
7. **Step 6** → The couple that is remodeling the dining room is the fifth.
8. **Step 7** → Jeanne is carrying is remodeling her bathroom, the couple that is remodeling their bathroom are carrying a green cart and the couple that is remodeling the dining room have a budget of 12.000 dollars. (**Clue 8, Clue 10 and Clue 18**)
9. At the fourth position, the couple was inspired by a magazine, while at the fifth position they were inspired by a friend's house. (**Clue 7 and Clue 11**)
10. **Step 1** → Kassie can only be at the fifth position. (**Clue 4**)
11. **Step 9** → Toni can only be at the second position and her inspiration was an exhibition. (**Clue 9**)
12. **Step 9, Step 8 and Step 11** → The couple inspired by a website can only be at the first position. (**Clue 1**)
13. **Step 12** → The couple inspired by a TV show can only be at the third position.
14. The couples that are carrying the black, red and white carts need to be positioned in this order, but not necessarily next to each other. (**Clue 1 and Clue 16**)
15. **Step 7, Step 15** → The couples that are carrying the black, red and white carts are the second, third and the fifth.
16. **Step 16** → The couple that is carrying the blue cart can only be the first.
17. The couple that has a budget of 3.000 dollars is the second, while the one that has a budget of 9.000 dollars is the fourth. (**Clue 15**)
18. **Step 17** → The couple that has a budget of 6.000 dollars can only be the third.
19. **Step 9** → Myles and Sylvia is at the fourth position. (**Clue 14**)
20. **Step 19** → Benny is the third, while Darrel is the fifth. (**Clue 19**)
21. Vince is the second. **Clue 13**
22. **Step 21** → Stephen is the first.

2.9.3 Mathematical Solution

	Customer #1	Customer #2	Customer #3	Customer #4	Customer #5
Shirt	Blue	Black	Red	Green	White
Name	Stephen	Vince	Benny	Myles	Darrel
Product	Amelia	Toni	Jeanne	Sylvia	Kassie
Discount	Kitchen	Bedroom	Living Room	Bathroom	Dining Room
Delivery	15.000 \$	3.000 \$	6.000 \$	9.000 \$	12.000 \$
Age	Website	Exhibition	TV show	Magazine	Friend's house

2.9.4 Mace4 Results

This puzzle was solved using predicates.

Amelia	1	0	0	0	0
Bathroom	0	0	0	1	0
Bedroom	0	1	0	0	0
Benny	0	0	1	0	0
Black	0	1	0	0	0
Blue	1	0	0	0	0
Couple1	1	0	0	0	0
Couple2	0	1	0	0	0
Couple3	0	0	1	0	0
Couple4	0	0	0	1	0
Couple5	0	0	0	0	1
Darrel	0	0	0	0	1
DiningRoom	0	0	0	0	1
Exhibition	0	1	0	0	0
FifteenThousand	1	0	0	0	0
FriendsHouse	0	0	0	0	1
Green	0	0	0	1	0
Jeanne	0	0	1	0	0
Kassie	0	0	0	0	1
Kitchen	1	0	0	0	0
LivingRoom	0	0	1	0	0
Magazine	0	0	0	1	0
Myles	0	0	0	1	0
NineThousand	0	0	0	1	0
Red	0	0	1	0	0
SixThousand	0	0	1	0	0
Stephen	1	0	0	0	0
Sylvia	0	0	1	0	0

TVshow		0	0	1	0	0
ThreeThousand		0	1	0	0	0
Toni		0	1	0	0	0
TwelveThousand		0	1	0	0	0
Vince		0	1	0	0	0
Website		1	0	0	0	0
White		0	0	0	0	1
extremity		1	0	0	0	1

differentFrom		0	1	2	3	4
0		0	1	1	1	1
1		1	0	1	1	1
2		1	1	0	1	1
3		1	1	1	0	1
4		1	1	1	1	0

directlyright		0	1	2	3	4
0		0	1	0	0	0
1		0	0	1	0	0
2		0	0	0	1	0
3		0	0	0	0	1
4		0	0	0	0	0

nextto		0	1	2	3	4
0		0	1	0	0	0
1		1	0	1	0	0
2		0	1	0	1	0
3		0	0	1	0	1
4		0	0	0	1	0

right		0	1	2	3	4
0		0	1	1	1	1
1		0	0	1	1	1
2		0	0	0	1	1
3		0	0	0	0	1
4		0	0	0	0	0

2.9.5 Prover9 Results

Proof from prover9, proving line 11693 by using lines 11692, 11352 and 87.

```
14881 Bathroom(d4) | Bathroom(d2) | Bathroom(d3). [back_unit_del(11605),  
unit_del(d,14880)].  
14882 -Bathroom(d2). [back_unit_del(493),unit_del(a,14880)].  
14883 -Bathroom(d3). [back_unit_del(492),unit_del(a,14880)].  
14884 -Bathroom(d4). [back_unit_del(491),unit_del(a,14880)].  
14886 $F. [back_unit_del(14881),unit_del(a,14884),unit_del(b,14882),  
unit_del(c,14883)].
```

Back-unit deletion at line 14881 \Rightarrow Copy clause 14881.

Unit-deletion at line 14884 \Rightarrow The first literal **a** has been removed because it was an instance of the negation clause 14884 (which is a unit clause).

Unit-deletion at line 14882 \Rightarrow The second literal **b** has been removed because it was an instance of the negation clause 14882 (which is a unit clause).

Unit-deletion at line 14883 \Rightarrow The third literal **c** has been removed because it was an instance of the negation clause 14883 (which is a unit clause).

Chapter 3

A3: Planning

Bibliography

Appendix A

Your original code

A.1 Stoian Iulia Tia's code:

A.1.1 Search

Code starts from the next page.

```

1 import display
2 import structures
3 from Queue import *
4 from copy import deepcopy
5
6 def isLegalMove(tempx, tempy):
7     if tempx < 0 or tempx > 2:
8         return False
9
10    if tempy < 0 or tempy > 2:
11        return False
12
13    return True
14
15 def switchMove(index):
16     switcher = {
17         0: "N",
18         1: "E",
19         2: "S",
20         3: "W"
21     }
22
23     return switcher.get(index)
24
25 class PuzzleModel:
26
27     def __init__(self, puzzle):
28         self.puzzle = puzzle
29         self.startState = structures.Node(puzzle) #puzzle = array de inturi
30
31     def solveFunction(self):
32         print("solving shit")
33
34         #print("Considering BFS: ", self.BFS())
35         #print("Considering DFS: ", self.IDDFS())
36         #print("Considering Greedy with Manhattan Distance Heuristic",
37         #      self.greedySearch(0))
38         #print("Considering Greedy with Euclidian Distance Heuristic",
39         #      self.greedySearch(1))
40         #print("Considering A* with Manhattan Distance Heuristic", self.aStarSearch(0))
41         #print("Considering A* with Euclidian Distance Heuristic", self.aStarSearch(1))
42
43     def isGoalState(self, currentState):
44         goalState = [0, 1, 2, 3, 4, 5, 6, 7, 8]
45
46         #print(currentState.puzzle)
47         #print(goalState)
48
49         for i in range(0, 9):
50             if (currentState.puzzle[i] != goalState[i]):
51                 #print("The current state is not the goal state.")
52                 return False
53
54         #print("The current state is the goal state.")
55         return True
56
57     def getSuccessors(self, currentState):
58         x = y = tile = 0
59
60         for i in range(0, 9):
61             if currentState.puzzle[i] == 0:
62                 x = i / 3
63                 y = i % 3
64                 tile = i
65                 break
66
67         dx = [-1, 0, 1, 0]

```

```

66         dy = [0, 1, 0, -1]
67
68     successors = Queue()
69     for i in range(0, 4):
70         tempx = x + dx[i]
71         tempy = y + dy[i]
72
73     if isLegalMove(tempx, tempy) == True:
74         newPuzzle = deepcopy(currentState.puzzle)
75         newTile = tempx * 3 + tempy % 3
76
77         newPuzzle[tile] = newPuzzle[newTile]
78         newPuzzle[newTile] = 0
79
80         action = switchMove(i)
81         #print(action)
82
83         newState = structures.Node(newPuzzle, currentState, action)
84         successors.put(newState)
85
86     return successors
87
88 def BFS(self):
89
90     frontier = Queue()
91     steps = list()
92
93     frontier.put(self.startState)
94     while True:
95         if frontier.empty():
96             return None # nu gaseste solutie daca vreodata frontiera ajunge goala
97
98         currentState = frontier.get()
99         if self.isGoalState(currentState):
100             return currentState.actions
101
102         elif currentState not in steps:
103             steps.append(currentState)
104             successors = self.getSuccessors(currentState)
105             while not successors.empty():
106                 frontier.put(successors.get())
107
108     # Iterative Deepening Depth-First Search Algorithm
109     def IDDFS(self):
110         depth = 0
111         currentState = None
112         while currentState == None:
113             currentState = self.DLS(depth)
114             depth = depth + 1
115
116         return currentState.actions
117
118     # Depth Limited Search ALgorithm
119     def DLS(self, depth):
120         frontier = LifoQueue()
121         frontier.put(self.startState)
122
123         while True:
124             if frontier.empty():
125                 return None
126
127             currentState = frontier.get()
128             if self.isGoalState(currentState):
129                 return currentState
130
131             elif currentState.depth != depth:
132                 successors = self.getSuccessors(currentState)

```

```

133             while not successors.empty():
134                 frontier.put(successors.get())
135
136             # nu merge
137             def greedySearch(self, type):
138                 currentState = self.startState
139
140                 frontier = PriorityQueue()
141                 steps = list()
142
143                 frontier.put((currentState.heuristic(type), currentState))
144
145                 while True:
146                     if frontier.empty():
147                         return None
148
149                     currentState = frontier.get()[1]
150
151                     print(currentState.puzzle)
152
153                     if self.isGoalState(currentState):
154                         print(currentState.actions)
155                         return currentState
156
157                     elif currentState not in steps:
158                         steps.append(currentState)
159                         successors = self.getSuccessors(currentState)
160
161                         while not successors.empty():
162                             successor = successors.get()
163                             frontier.put((successor.heuristic(type), successor))
164
165             def aStarSearch(self, type):
166                 currentState = self.startState
167                 frontier = PriorityQueue()
168                 frontier.put((currentState.heuristic(type), currentState))
169                 steps = list()
170
171                 while True:
172                     if frontier.empty():
173                         return None
174
175                     currentState = frontier.get()[1]
176                     if self.isGoalState(currentState):
177                         print(currentState.actions)
178                         return currentState
179
180                     elif currentState not in steps:
181                         steps.append(currentState)
182                         successors = self.getSuccessors(currentState)
183
184                         while not successors.empty():
185                             successor = successors.get()
186                             frontier.put((successor.heuristic(type) + successor.depth,
successor))

```

A.1.2 Logics

Who owns the crocodile? (page 1 of 4)

```
1  if(Mace4).
2  assign(domain_size, 5).
3  end_if.
4
5  if(Prover9).
6  formulas(goals).
7
8  Girl3(g3) -> Crocodile(g3).
9
10 end_of_list.
11 end_if.
12
13 % Avoid using sentence names starting with u, v, w, x, y, z.
14
15 % Girls' order in row
16     % Girl1: first girl
17     % Girl2: second girl
18     % Girl3: third girl
19     % Girl4: fourth girl
20     % Girl5: fifth girl
21
22 % Girls' names
23     % Georgina, Hannah, Jessica, Jo, Lucy
24
25 % Favourite Colour
26     % Blue, Green, Lilac, Pink, Purple
27
28 % Favourite Chocolate
29     % Boost, Crunchies, DairyMilk: Dairy Milk, MilkyBars: Milky Bars, WispaBites:
      Wispa Bites
30
31 % Favourite Pet
32     % Crocodile, Hamster, Horse, Puppy, Rabbit
33
34 % Favourite Hobby
35     % Riding: Horse riding, Skiing, Bowling: Ten-Pin bowling, Tennis
36
37 % Favourite Holiday
38     % Australia, Canada, Florida, Maldives, Tobago
39
40 formulas(assumptions).
41
42 % The 5 girls differ
43     differentFrom(g1, g2).
44     differentFrom(g1, g3).
45     differentFrom(g1, g4).
46     differentFrom(g1, g5).
47     differentFrom(g2, g3).
48     differentFrom(g2, g4).
49     differentFrom(g2, g5).
50     differentFrom(g3, g4).
51     differentFrom(g3, g5).
52     differentFrom(g4, g5).
53
54 % The "differentFrom" relation is symmetrical
55     differentFrom(x, y) -> differentFrom(y, x).
56
57     g1 = 0.
58     g2 = 1.
59     g3 = 2.
60     g4 = 3.
61     g5 = 4.
62
63     Girl1(0).
64     Girl2(1).
65     Girl3(2).
66     Girl4(3).
67     Girl5(4).
68
```

Who owns the crocodile? (page 2 of 4)

```

69  % Relation directlyright(x, y) that affirms that "girl y" is directly to the right of
70   "girl x"
71      directlyright(g1, g2).
72      directlyright(g2, g3).
73      directlyright(g3, g4).
74      directlyright(g4, g5).
75
76      -directlyright(g1, g1).
77      -directlyright(g2, g2).
78      -directlyright(g3, g3).
79      -directlyright(g4, g4).
80      -directlyright(g5, g5).
81      -directlyright(g1, g3).
82      -directlyright(g2, g4).
83      -directlyright(g3, g5).
84      -directlyright(g3, g1).
85      -directlyright(g4, g2).
86      -directlyright(g5, g3).
87      -directlyright(g2, g1).
88      -directlyright(g3, g2).
89      -directlyright(g4, g3).
90      -directlyright(g5, g4).
91      -directlyright(g1, g4).
92      -directlyright(g2, g5).
93      -directlyright(g4, g1).
94      -directlyright(g5, g2).
95      -directlyright(g1, g5).
96      -directlyright(g5, g1).
97
98  % Relation nextto(x, y) that affirms that "girl y" is next to "girl x"
99      nextto(g1, g2).
100     nextto(g2, g1).
101     nextto(g2, g3).
102     nextto(g3, g2).
103     nextto(g3, g4).
104     nextto(g4, g3).
105     nextto(g4, g5).
106
107    -nextto(g1, g1).
108    -nextto(g2, g2).
109    -nextto(g3, g3).
110    -nextto(g4, g4).
111    -nextto(g5, g5).
112    -nextto(g1, g3).
113    -nextto(g2, g4).
114    -nextto(g3, g5).
115    -nextto(g3, g1).
116    -nextto(g4, g2).
117    -nextto(g5, g3).
118    -nextto(g1, g4).
119    -nextto(g2, g5).
120    -nextto(g4, g1).
121    -nextto(g5, g2).
122    -nextto(g1, g5).
123    -nextto(g5, g1).
124
125  % Each girl is different and has a name, favourite colour, chocolate, pet, hobby and
126  % holiday
127      Girl1(x) | Girl2(x) | Girl3(x) | Girl4(x) | Girl5(x).
128      Georgina(x) | Hannah(x) | Jessica(x) | Jo(x) | Lucy(x).
129      Blue(x) | Green(x) | Lilac(x) | Pink(x) | Purple(x).
130      Boost(x) | Crunchies(x) | DairyMilk(x) | MilkyBars(x) | WispaBites(x).
131      Crocodile(x) | Hamster(x) | Horse(x) | Puppy(x) | Rabbit(x).
132      Riding(x) | Skiing(x) | Swimming(x) | Bowling(x) | Tennis(x).
133      Australia(x) | Canada(x) | Florida(x) | Maldives(x) | Tobago(x).
134
135  % Each property applies at most to one girl.
           Georgina(x) & Georgina(y) -> -differentFrom(x, y).

```

Who owns the crocodile? (page 3 of 4)

```
136    Hannah(x) & Hannah(y) -> -differentFrom(x, y).
137    Jessica(x) & Jessica(y) -> -differentFrom(x, y).
138    Jo(x) & Jo(y) -> -differentFrom(x, y).
139    Lucy(x) & Lucy(y) -> -differentFrom(x, y).
140
141    Blue(x) & Blue(y) -> -differentFrom(x, y).
142    Green(x) & Green(y) -> -differentFrom(x, y).
143    Lilac(x) & Lilac(y) -> -differentFrom(x, y).
144    Pink(x) & Pink(y) -> -differentFrom(x, y).
145    Purple(x) & Purple(y) -> -differentFrom(x, y).
146
147    Boost(x) & Boost(y) -> -differentFrom(x, y).
148    Crunchies(x) & Crunchies(y) -> -differentFrom(x, y).
149    DairyMilk(x) & DairyMilk(y) -> -differentFrom(x, y).
150    MilkyBars(x) & MilkyBars(y) -> -differentFrom(x, y).
151    WispaBites(x) & WispaBites(y) -> -differentFrom(x, y).
152
153    Crocodile(x) & Crocodile(y) -> -differentFrom(x, y).
154    Hamster(x) & Hamster(y) -> -differentFrom(x, y).
155    Horse(x) & Horse(y) -> -differentFrom(x, y).
156    Puppy(x) & Puppy(y) -> -differentFrom(x, y).
157    Rabbit(x) & Rabbit(y) -> -differentFrom(x, y).
158
159    Riding(x) & Riding(y) -> -differentFrom(x, y).
160    Skiing(x) & Skiing(y) -> -differentFrom(x, y).
161    Swimming(x) & Swimming(y) -> -differentFrom(x, y).
162    Bowling(x) & Bowling(y) -> -differentFrom(x, y).
163    Tennis(x) & Tennis(y) -> -differentFrom(x, y).
164
165    Australia(x) & Australia(y) -> -differentFrom(x, y).
166    Canada(x) & Canada(y) -> -differentFrom(x, y).
167    Florida(x) & Florida(y) -> -differentFrom(x, y).
168    Maldives(x) & Maldives(y) -> -differentFrom(x, y).
169    Tobago(x) & Tobago(y) -> -differentFrom(x, y).
170
171 %Clues
172
173    % 1. Jo likes the Wispa Bites.
174    Jo(x) -> WispaBites(x).
175
176    % 2. The person with the Hamster likes Swimming.
177    Hamster(x) -> Swimming(x).
178
179    % 3. Hannah eats Dairy Milk.
180    Hannah(x) -> DairyMilk(x).
181
182    % 4. Jessica is on the left of Georgina.
183    Jessica(x) & Georgina(y) -> directlyright(x, y).
184
185    % 5. Lucy is the first on the left.
186    Girl1(x) -> Lucy(x).
187
188    % 6. The first person on the right likes Swimming.
189    Girl5(x) -> Swimming(x).
190
191    % 7. The person who eats Milky Bars owns a Horse.
192    MilkyBars(x) -> Horse(x).
193
194    % 8. The person in the middle eats Dairy Milk.
195    Girl3(x) -> DairyMilk(x).
196
197    % 9. Jessica likes Green.
198    Jessica(x) -> Green(x).
199
200    % 10. The person on the left of the middle wants to go to Tobago.
201    Girl2(x) -> Tobago(x).
202
203    % 11. The person who wants to go to Maldives likes Lilac.
204    Maldives(x) -> Lilac(x).
```

Who owns the crocodile? (page 4 of 4)

```
205
206      % 12. The person who likes Wispa Bites sits next to the person who wants to go to
207      Florida.
208      WispaBites(x) & Florida(y) -> nextto(y, x).
209
210      % 13. The person who likes Pink wants to go to Florida.
211      Pink(x) -> Florida(x).
212
213      % 14. The person who sits first on the left likes lilac.
214      Girl1(x) -> Lilac(x).
215
216      % 15. The girl that likes Blue owns a Puppy.
217      Blue(x) -> Puppy(x).
218
219      % 16. The person who likes Skiing sits next to the person who has a Hamster.
220      Skiing(x) & Hamster(y) -> nextto(y, x).
221
222      % 17. The girl on the right of the girl who likes Tennis likes Horse riding.
223      Tennis(x) & Riding(y) -> directlyright(x, y).
224
225      % 18. The girl next to the girl who likes Milky bars likes Boost.
226      MilkyBars(x) & Boost(y) -> nextto(x, y).
227
228      % 19. The girl who likes Purple wants to go to Canada.
229      Purple(x) -> Canada(x).
230
231      % 20. The girl who likes Crunchies owns a rabbit.
232      Crunchies(x) -> Rabbit(x).
233
234      % 21. The girl who likes Skiing sits next to the girl who play Ten-Pin bowlings.
235      Skiing(x) & Bowling(y) -> nextto(y, x).
236
237      % 22. Jessica wants to go to Australia.
238      Jessica(x) -> Australia(x).
239      end_of_list.
240
```

Ancient Civilizations (page 1 of 5)

```
1  if(Mace4).
2  assign(domain_size, 5).
3  end_if.
4
5  if(Prover9).
6  formulas(goals).
7
8  Gir11(g1) -> Blue(g1).
9
10 end_of_list.
11 end_if.
12
13 % Avoid using sentence names starting with u, v, w, x, y, z.
14
15 % Girls' order in row
16     % Gir11: first girl
17     % Gir12: second girl
18     % Gir13: third girl
19     % Gir14: fourth girl
20     % Gir15: fifth girl
21
22 % Girls' names
23     % Ella, Julia, Lily, Makayla, Riley
24
25 % Shirt
26     % Black, Blue, Green, Red, White
27
28 % Civilization
29     % Aztec, Chinese, Egyptian, Greek, Roman
30
31 % Subject
32     % Architecture, Art, Economy, Religion, Science
33
34 % Age
35     % Ten, Eleven, Twelve, Thirteen, Fourteen
36
37 % Juice
38     % Apple, Cranberry, Grapefruit, Orange, Strawberry
39
40 formulas(assumptions).
41
42 % The 5 girls differ
43     differentFrom(g1, g2).
44     differentFrom(g1, g3).
45     differentFrom(g1, g4).
46     differentFrom(g1, g5).
47     differentFrom(g2, g3).
48     differentFrom(g2, g4).
49     differentFrom(g2, g5).
50     differentFrom(g3, g4).
51     differentFrom(g3, g5).
52     differentFrom(g4, g5).
53
54 % The "differentFrom" relation is symmetrical
55     differentFrom(x, y) -> differentFrom(y, x).
56
57     g1 = 0.
58     g2 = 1.
59     g3 = 2.
60     g4 = 3.
61     g5 = 4.
62
63     Gir11(0).
64     Gir12(1).
65     Gir13(2).
66     Gir14(3).
67     Gir15(4).
68
69 % Relation directlyright(x, y) that affirms that "girl y" is directly to the right of
```

Ancient Civilizations (page 2 of 5)

```
"girl x"
70    directlyright(g1, g2).
71    directlyright(g2, g3).
72    directlyright(g3, g4).
73    directlyright(g4, g5).
74
75    -directlyright(g1, g1).
76    -directlyright(g2, g2).
77    -directlyright(g3, g3).
78    -directlyright(g4, g4).
79    -directlyright(g5, g5).
80    -directlyright(g1, g3).
81    -directlyright(g2, g4).
82    -directlyright(g3, g5).
83    -directlyright(g3, g1).
84    -directlyright(g4, g2).
85    -directlyright(g5, g3).
86    -directlyright(g2, g1).
87    -directlyright(g3, g2).
88    -directlyright(g4, g3).
89    -directlyright(g5, g4).
90    -directlyright(g1, g4).
91    -directlyright(g2, g5).
92    -directlyright(g4, g1).
93    -directlyright(g5, g2).
94    -directlyright(g1, g5).
95    -directlyright(g5, g1).
96
97 % Relation nextto(x, y) that affirms that "girl y" is next to "girl x"
98    nextto(g1, g2).
99    nextto(g2, g1).
100   nextto(g2, g3).
101   nextto(g3, g2).
102   nextto(g3, g4).
103   nextto(g4, g3).
104   nextto(g4, g5).
105   nextto(g5, g4).
106
107  -nextto(g1, g1).
108  -nextto(g2, g2).
109  -nextto(g3, g3).
110  -nextto(g4, g4).
111  -nextto(g5, g5).
112  -nextto(g1, g3).
113  -nextto(g2, g4).
114  -nextto(g3, g5).
115  -nextto(g3, g1).
116  -nextto(g4, g2).
117  -nextto(g5, g3).
118  -nextto(g1, g4).
119  -nextto(g2, g5).
120  -nextto(g4, g1).
121  -nextto(g5, g2).
122  -nextto(g1, g5).
123  -nextto(g5, g1).
124
125 % Relation extremity(x) that affirms that "girl x" is at one of the ends
126    extremity(g1).
127    extremity(g5).
128
129    -extremity(g2).
130    -extremity(g3).
131    -extremity(g4).
132
133 % Relation inbetween(x, y) that affirms that there is a girl somewhere in between "girl
134 x" and "girl y", exactly in that order
135    inbetween(g1, g3).
136    inbetween(g2, g4).
137    inbetween(g3, g5).
```

Ancient Civilizations (page 3 of 5)

```
137      inbetween(g1, g4).
138      inbetween(g2, g5).
139      inbetween(g1, g5).
140
141      -inbetween(g1, g2).
142      -inbetween(g2, g1).
143      -inbetween(g2, g3).
144      -inbetween(g3, g2).
145      -inbetween(g3, g4).
146      -inbetween(g4, g3).
147      -inbetween(g4, g5).
148      -inbetween(g5, g4).
149      -inbetween(g1, g1).
150      -inbetween(g2, g2).
151      -inbetween(g3, g3).
152      -inbetween(g4, g4).
153      -inbetween(g5, g5).
154      -inbetween(g3, g1).
155      -inbetween(g4, g2).
156      -inbetween(g5, g3).
157      -inbetween(g4, g1).
158      -inbetween(g5, g2).
159      -inbetween(g5, g1).
160
161 % Relation right(x, y) that affirms that "girl y" is somewhere to the right of "girl x"
162      right(g1, g3).
163      right(g2, g4).
164      right(g3, g5).
165      right(g1, g4).
166      right(g2, g5).
167      right(g1, g5).
168      right(g1, g2).
169      right(g4, g5).
170      right(g2, g3).
171      right(g3, g4).
172
173      -right(g2, g1).
174      -right(g3, g2).
175      -right(g4, g3).
176      -right(g5, g4).
177      -right(g1, g1).
178      -right(g2, g2).
179      -right(g3, g3).
180      -right(g4, g4).
181      -right(g5, g5).
182      -right(g3, g1).
183      -right(g4, g2).
184      -right(g5, g3).
185      -right(g4, g1).
186      -right(g5, g2).
187      -right(g5, g1).
188
189 % Each girl has different characteristics
190      Girl1(x) | Girl2(x) | Girl3(x) | Girl4(x) | Girl5(x).
191      Ella(x) | Julia(x) | Lily(x) | Makayla(x) | Riley(x).
192      Black(x) | Blue(x) | Green(x) | Red(x) | White(x).
193      Aztec(x) | Chinese(x) | Egyptian(x) | Greek(x) | Roman(x).
194      Architecture(x) | Art(x) | Economy(x) | Religion(x) | Science(x).
195      Ten(x) | Eleven(x) | Twelve(x) | Thirteen(x) | Fourteen(x).
196      Apple(x) | Cranberry(x) | Grapefruit(x) | Orange(x) | Strawberry(x).
197
198 % Each property applies at most to one girl
199      Ella(x) & Ella(y) -> -differentFrom(x, y).
200      Julia(x) & Julia(y) -> -differentFrom(x, y).
201      Lily(x) & Lily(y) -> -differentFrom(x, y).
202      Makayla(x) & Makayla(y) -> -differentFrom(x, y).
203      Riley(x) & Riley(y) -> -differentFrom(x, y).
204
205      Black(x) & Black(y) -> -differentFrom(x, y).
```

Ancient Civilizations (page 4 of 5)

```
206    Blue(x) & Blue(y) -> -differentFrom(x, y).
207    Green(x) & Green(y) -> -differentFrom(x, y).
208    Red(x) & Red(y) -> -differentFrom(x, y).
209    White(x) & White(y) -> -differentFrom(x, y).
210
211    Aztec(x) & Aztec(y) -> -differentFrom(x, y).
212    Chinese(x) & Chinese(y) -> -differentFrom(x, y).
213    Egyptian(x) & Egyptian(y) -> -differentFrom(x, y).
214    Greek(x) & Greek(y) -> -differentFrom(x, y).
215    Roman(x) & Roman(y) -> -differentFrom(x, y).
216
217    Architecture(x) & Architecture(y) -> -differentFrom(x, y).
218    Art(x) & Art(y) -> -differentFrom(x, y).
219    Economy(x) & Economy(y) -> -differentFrom(x, y).
220    Religion(x) & Religion(y) -> -differentFrom(x, y).
221    Science(x) & Science(y) -> -differentFrom(x, y).
222
223    Ten(x) & Ten(y) -> -differentFrom(x, y).
224    Eleven(x) & Eleven(y) -> -differentFrom(x, y).
225    Twelve(x) & Twelve(y) -> -differentFrom(x, y).
226    Thirteen(x) & Thirteen(y) -> -differentFrom(x, y).
227    Fourteen(x) & Fourteen(y) -> -differentFrom(x, y).
228
229    Apple(x) & Apple(y) -> -differentFrom(x, y).
230    Cranberry(x) & Cranberry(y) -> -differentFrom(x, y).
231    Grapefruit(x) & Grapefruit(y) -> -differentFrom(x, y).
232    Orange(x) & Orange(y) -> -differentFrom(x, y).
233    Strawberry(x) & Strawberry(y) -> -differentFrom(x, y).
234
235 %Clues
236
237    % 1. At the fourth position is the girl drinking Strawberry juice.
238    Girl4(x) -> Strawberry(x).
239
240    % 2. The girl drinking Strawberry juice is somewhere between the girl who likes Art
241    and the girl that is drinking Apple juice, in that order.
242    Art(x) & Apple(x) -> inbetween(x, y).
243    Strawberry(x) & Art(y) -> right(y, x).
244    Strawberry(x) & Apple(y) -> right(x, y).
245
246    % 3. Riley is next to the girl who likes the Roman civilization.
247    Riley(x) & Roman(y) -> nextto(y, x).
248
249    % 4. The girl wearing the Green shirt is drinking Apple juice.
250    Green(x) -> Apple(x).
251
252    % 5. At the fourth position is the girl that is interested in Religion.
253    Girl4(x) -> Religion(x).
254
255    % 6. The 12 years old girl is exactly to the right of the 11 years old girl.
256    Twelve(x) & Eleven(y) -> directlyright(y, x).
257
258    % 7. At one of the ends is the girl who likes the Egyptian civilization.
259    Egyptian(x) -> extremity(x).
260
261    % 8. The girl wearing the Black shirt is somewhere between the girl that likes
262    Science and the girl wearing the Red shirt, in that order.
263    Science(x) & Red(y) -> inbetween(x, y).
264    Black(x) & Science(y) -> right(y, x).
265    Black(x) & Red(y) -> right(x, y).
266
267    % 9. The girl wearing the Red shirt is somewhere to the left of the girl wearing
268    the White shirt.
269    Red(x) & White(y) -> right(x, y).
270
271    % 10. The girl drinking Cranberry juice likes the Greek civilization.
272    Cranberry(x) -> Greek(x).
273
274    % 11. Ella is at the third position.
```

Ancient Civilizations (page 5 of 5)

```
272     Girl3(x) -> Ella(x).
273
274     % 12. The 12 years old girl is next to the girl that is interested in Architecture.
275     Twelve(x) & Architecture(y) -> nextto(y, x).
276
277     % 13. Riley is exactly to the right of the girl that admires the Roman civilization.
278     Riley(x) & Roman(y) -> directlyright(y, x).
279
280     % 14. The girl drinking Cranberry juice is somewhere between the 13 years old girl
281     and the girl drinking Grapefruit juice, in that order.
282     Thirteen(x) & Grapefruit(y) -> inbetween(x, y).
283     Cranberry(x) & Thirteen(y) -> right(y, x).
284     Cranberry(x) & Grapefruit(y) -> right(x, y).
285
286     % 15. At one of the ends is the girl that likes Science.
287     Science(x) -> extremity(x).
288
289     % 16. At one of the ends is the girl who loves Architecture.
290     Architecture(x) -> extremity(x).
291
292     % 17. Lily is exactly to the right of the girl who likes Economy.
293     Lily(x) & Economy(y) -> directlyright(y, x).
294
295     % 18. Makayla is 10.
296     Makayla(x) -> Ten(x).
297
298     % 19. The girl interested in the Aztec civilization is somewhere between the girl
299     interested in the Roman civilization and the girl interested in the Chinese
300     civilization, in that order
301     Roman(x) & Chinese(y) -> inbetween(x, y).
302     Aztec(x) & Roman(y) -> right(y, x).
303     Aztec(x) & Chinese(y) -> right(x, y).
304
305     % 20. The youngest girl is at one of the ends.
306     Ten(x) -> extremity(x).
307
308     end_of_list.
309
```

Travel Agency (page 1 of 4)

```
1  if(Mace4).
2  assign(domain_size, 5).
3  end_if.
4
5  if(Prover9).
6  formulas(goals).
7
8  Woman4(d4) -> Italy(d4).
9
10 end_of_list.
11 end_if.
12
13 % Avoid using sentence names starting with u, v, w, x, y, z.
14
15 % Womans' order in row
16     % Woman1: first woman
17     % Woman2: second woman
18     % Woman3: third woman
19     % Woman4: fourth woman
20     % Woman5: fifth woman
21
22 % Womans' names
23     % Ana, Glenda, Jessie, Lara, Rose
24
25 % Shirt
26     % Blue, Green, Red, White, Yellow
27
28 % Age
29     % TwentyFour, TwentySix, TwentyEight, Thirty, ThirtyTwo
30
31 % Profession
32     % Biologist, Hostess, Judge, Nurse, Singer
33
34 % Country
35     % China, Egypt, Italy, Mexico, Peru
36
37 % Duration
38     % FiveDays, TenDays, FifteenDays, TwentyDays, TwentyFiveDays
39
40 formulas(assumptions).
41
42 % The 5 womans differ
43     differentFrom(d1, d2).
44     differentFrom(d1, d3).
45     differentFrom(d1, d4).
46     differentFrom(d1, d5).
47     differentFrom(d2, d3).
48     differentFrom(d2, d4).
49     differentFrom(d2, d5).
50     differentFrom(d3, d4).
51     differentFrom(d3, d5).
52     differentFrom(d4, d5).
53
54 % The "differentFrom" relation is symmetrical
55     differentFrom(x, y) -> differentFrom(y, x).
56
57     d1 = 0.
58     d2 = 1.
59     d3 = 2.
60     d4 = 3.
61     d5 = 4.
62
63     Woman1(0).
64     Woman2(1).
65     Woman3(2).
66     Woman4(3).
67     Woman5(4).
68
69 % Relation directlyright(x, y) that affirms that "woman y" is directly to the right of
```

Travel Agency (page 2 of 4)

```
"woman x"
70    directlyright(d1, d2).
71    directlyright(d2, d3).
72    directlyright(d3, d4).
73    directlyright(d4, d5).
74
75    -directlyright(d1, d1).
76    -directlyright(d2, d2).
77    -directlyright(d3, d3).
78    -directlyright(d4, d4).
79    -directlyright(d5, d5).
80    -directlyright(d1, d3).
81    -directlyright(d2, d4).
82    -directlyright(d3, d5).
83    -directlyright(d3, d1).
84    -directlyright(d4, d2).
85    -directlyright(d5, d3).
86    -directlyright(d2, d1).
87    -directlyright(d3, d2).
88    -directlyright(d4, d3).
89    -directlyright(d5, d4).
90    -directlyright(d1, d4).
91    -directlyright(d2, d5).
92    -directlyright(d4, d1).
93    -directlyright(d5, d2).
94    -directlyright(d1, d5).
95    -directlyright(d5, d1).
96
97 % Relation nextto(x, y) that affirms that "woman y" is next to "woman x"
98    directlyright(x, y) | directlyright(y, x) <-> nextto(x, y).
99
100 % Relation extremity(x) that affirms that "woman x" is at One of the ends.
101    extremity(d1).
102    extremity(d5).
103
104    -extremity(d2).
105    -extremity(d3).
106    -extremity(d4).
107
108 % Relation right(x, y) that affirms that "woman y" is somewhere to the right of "woman x"
109    right(d1, d3).
110    right(d2, d4).
111    right(d3, d5).
112    right(d1, d4).
113    right(d2, d5).
114    right(d1, d5).
115    right(d1, d2).
116    right(d4, d5).
117    right(d2, d3).
118    right(d3, d4).
119
120    -right(d2, d1).
121    -right(d3, d2).
122    -right(d4, d3).
123    -right(d5, d4).
124    -right(d1, d1).
125    -right(d2, d2).
126    -right(d3, d3).
127    -right(d4, d4).
128    -right(d5, d5).
129    -right(d3, d1).
130    -right(d4, d2).
131    -right(d5, d3).
132    -right(d4, d1).
133    -right(d5, d2).
134    -right(d5, d1).
135
136 % Each woman has different characteristics
137    Woman1(x) | Woman2(x) | Woman3(x) | Woman4(x) | Woman5(x).
```

Travel Agency (page 3 of 4)

```

138     Ana(x) | Glenda(x) | Jessie(x) | Lara(x) | Rose(x).
139     Blue(x) | Green(x) | Red(x) | White(x) | Yellow(x).
140     TwentyFour(x) | TwentySix(x) | TwentyEight(x) | Thirty(x) | ThirtyTwo(x).
141     Biologist(x) | Hostess(x) | Judge(x) | Nurse(x) | Singer(x).
142     China(x) | Egypt(x) | Italy(x) | Mexico(x) | Peru(x).
143     FiveDays(x) | TenDays(x) | FifteenDays(x) | TwentyDays(x) | TwentyFiveDays(x).
144
145     % Each property applies at most to one woman
146     Ana(x) & Ana(y) -> -differentFrom(x, y).
147     Glenda(x) & Glenda(y) -> -differentFrom(x, y).
148     Jessie(x) & Jessie(y) -> -differentFrom(x, y).
149     Lara(x) & Lara(y) -> -differentFrom(x, y).
150     Rose(x) & Rose(y) -> -differentFrom(x, y).
151
152     Blue(x) & Blue(y) -> -differentFrom(x, y).
153     Green(x) & Green(y) -> -differentFrom(x, y).
154     Red(x) & Red(y) -> -differentFrom(x, y).
155     White(x) & White(y) -> -differentFrom(x, y).
156     Yellow(x) & Yellow(y) -> -differentFrom(x, y).
157
158     TwentyFour(x) & TwentyFour(y) -> -differentFrom(x, y).
159     TwentySix(x) & TwentySix(y) -> -differentFrom(x, y).
160     TwentyEight(x) & TwentyEight(y) -> -differentFrom(x, y).
161     Thirty(x) & Thirty(y) -> -differentFrom(x, y).
162     ThirtyTwo(x) & ThirtyTwo(y) -> -differentFrom(x, y).
163
164     Biologist(x) & Biologist(y) -> -differentFrom(x, y).
165     Hostess(x) & Hostess(y) -> -differentFrom(x, y).
166     Judge(x) & Judge(y) -> -differentFrom(x, y).
167     Nurse(x) & Nurse(y) -> -differentFrom(x, y).
168     Singer(x) & Singer(y) -> -differentFrom(x, y).
169
170     China(x) & China(y) -> -differentFrom(x, y).
171     Egypt(x) & Egypt(y) -> -differentFrom(x, y).
172     Italy(x) & Italy(y) -> -differentFrom(x, y).
173     Mexico(x) & Mexico(y) -> -differentFrom(x, y).
174     Peru(x) & Peru(y) -> -differentFrom(x, y).
175
176     FiveDays(x) & FiveDays(y) -> -differentFrom(x, y).
177     TenDays(x) & TenDays(y) -> -differentFrom(x, y).
178     FifteenDays(x) & FifteenDays(y) -> -differentFrom(x, y).
179     TwentyDays(x) & TwentyDays(y) -> -differentFrom(x, y).
180     TwentyFiveDays(x) & TwentyFiveDays(y) -> -differentFrom(x, y).
181
182     %Clues
183
184     % 1. The Singer is at the third position.
185     Woman3(x) -> Singer(x).
186
187     % 2. The woman traveling for 20 days is somewhere between the woman who is going to
188     Peru and the owner of the Blue purse, in that order.
189     Peru(x) & TwentyDays(y) & Blue(z) -> right(x, y) & right(y, z).
190
191     % 3. Ana is exactly to the left of the Biologist.
192     Ana(x) & Biologist(y) -> directlyright(x, y).
193
194     % 4. The 32 years old is going to see the Sahara.
195     ThirtyTwo(x) -> Egypt(x).
196
197     % 5. The owner of the White purse is exactly to the right of the woman traveling to
198     visit Machu Picchu.
199     Peru(x) & White(y) -> directlyright(x, y).
200
201     % 6. Glenda is somewhere to the right of the woman who has the Green purse.
202     Green(x) & Glenda(y) -> right(x, y).
203     % 7. The person wearing the White purse is somewhere between the 30 years old woman
204     and the owner of the Blue purse, in that order.
205     Thirty(x) & White(y) & Blue(z) -> right(x, y) & right(y, z).

```

Travel Agency (page 4 of 4)

```
204
205 % 8. The 24 years old woman is going to visit an Aztec pyramid.
206 TwentyFour(x) -> Mexico(x).
207
208 % 9. The woman wearing the White purse is somewhere to the left of the youngest
209 woman.
210 White(x) & TwentyFour(y) -> right(x, y).
211
212 % 10. The traveler going to Italy is exactly to the right of the woman traveling
213 for 20 days.
214 TwentyDays(x) & Italy(y) -> directlyright(x, y).
215
216 % 11. The person who is going to travel for 25 days has the Red purse.
217 TwentyFiveDays(x) -> Red(x).
218
219 % 12. The Judge is in the first position.
220 Woman1(x) -> Judge(x).
221
222 % 13. The Nurse is exactly to the right of the woman who is going to travel for 20
223 days.
224 TwentyDays(x) & Nurse(y) -> directlyright(x, y).
225
226 % 14. The Hostess is somewhere between Lara and the woman who has the Blue purse,
227 in that order.
228 Lara(x) & Hostess(y) & Blue(z) -> right(x, y) & right(y, z).
229
230 % 15. In the second position is the woman that is going to travel for 15 days.
231 Woman2(x) -> FifteenDays(x).
232
233 % 16. Rose has the Green purse.
234 Rose(x) -> Green(x).
235
236 % 17. The woman who is traveling for less than a week is exactly to the left of the
237 32 years old woman.
238 FiveDays(x) & ThirtyTwo(y) -> directlyright(x, y).
239
240 % 18. The person traveling for 5 days is 28.
241 FiveDays(x) -> TwentyEight(x).
242
243 % 19. The Blue purse owner is somewhere between the 30 years old woman and the
244 owner of the Yellow purse, in that order.
245 Thirty(x) & Blue(y) & Yellow(z) -> right(x, y) & right(y, z).
246
247 end_of_list.
```

Persian Rugs (page 1 of 4)

```
1  if(Mace4).
2  assign(domain_size, 5).
3  end_if.
4
5  if(Prover9).
6  formulas(goals).
7
8  Man4(d4) -> Round(d4).
9
10 end_of_list.
11 end_if.
12
13 % Avoid using sentence names starting with u, v, w, x, y, z.
14
15 % Men's order in row
16     % Man1: first man
17     % Man2: second man
18     % Man3: third man
19     % Man4: fourth man
20     % Man5: fifth man
21
22 % Men's names
23     % Bryan, Casey, Garrett, Shane, Timothy
24
25 % Colour
26     % Blue, Green, Orange, Red, Yellow
27
28 % Type
29     % Oval, Rectangular, Round, Runner, Square
30
31 % Symbols
32     % Birds, Diamonds, Lotuses, Roses, Stars
33
34 % Price
35     % FourHundred, SixHundred, EightHundred, OneThousand, TwelveHundred
36
37 % Place
38     % Bedroom, Hall, Library, LivingRoom, Office
39
40 formulas(assumptions).
41
42 % The 5 men differ
43     differentFrom(d1, d2).
44     differentFrom(d1, d3).
45     differentFrom(d1, d4).
46     differentFrom(d1, d5).
47     differentFrom(d2, d3).
48     differentFrom(d2, d4).
49     differentFrom(d2, d5).
50     differentFrom(d3, d4).
51     differentFrom(d3, d5).
52     differentFrom(d4, d5).
53
54 % The "differentFrom" relation is symmetrical
55     differentFrom(x, y) -> differentFrom(y, x).
56
57     d1 = 0.
58     d2 = 1.
59     d3 = 2.
60     d4 = 3.
61     d5 = 4.
62
63     Man1(0).
64     Man2(1).
65     Man3(2).
66     Man4(3).
67     Man5(4).
68
69 % Relation directlyright(x, y) that affirms that "man y" is directly to the right of
```

Persian Rugs (page 2 of 4)

```
"man x"
70    directlyright(d1, d2).
71    directlyright(d2, d3).
72    directlyright(d3, d4).
73    directlyright(d4, d5).
74
75    -directlyright(d1, d1).
76    -directlyright(d2, d2).
77    -directlyright(d3, d3).
78    -directlyright(d4, d4).
79    -directlyright(d5, d5).
80    -directlyright(d1, d3).
81    -directlyright(d2, d4).
82    -directlyright(d3, d5).
83    -directlyright(d3, d1).
84    -directlyright(d4, d2).
85    -directlyright(d5, d3).
86    -directlyright(d2, d1).
87    -directlyright(d3, d2).
88    -directlyright(d4, d3).
89    -directlyright(d5, d4).
90    -directlyright(d1, d4).
91    -directlyright(d2, d5).
92    -directlyright(d4, d1).
93    -directlyright(d5, d2).
94    -directlyright(d1, d5).
95    -directlyright(d5, d1).
96
97 % Relation nextto(x, y) that affirms that "man y" is next to "man x"
98    directlyright(x, y) | directlyright(y, x) <-> nextto(x, y).
99
100 % Relation extremity(x) that affirms that "man x" is at One of the ends.
101    extremity(d1).
102    extremity(d5).
103
104    -extremity(d2).
105    -extremity(d3).
106    -extremity(d4).
107
108 % Relation right(x, y) that affirms that "man y" is somewhere to the right of "man x"
109    right(d1, d3).
110    right(d2, d4).
111    right(d3, d5).
112    right(d1, d4).
113    right(d2, d5).
114    right(d1, d5).
115    right(d1, d2).
116    right(d4, d5).
117    right(d2, d3).
118    right(d3, d4).
119
120    -right(d2, d1).
121    -right(d3, d2).
122    -right(d4, d3).
123    -right(d5, d4).
124    -right(d1, d1).
125    -right(d2, d2).
126    -right(d3, d3).
127    -right(d4, d4).
128    -right(d5, d5).
129    -right(d3, d1).
130    -right(d4, d2).
131    -right(d5, d3).
132    -right(d4, d1).
133    -right(d5, d2).
134    -right(d5, d1).
135
136 % % Each man has different characteristics
137    Man1(x) | Man2(x) | Man3(x) | Man4(x) | Man5(x).
```

Persian Rugs (page 3 of 4)

```

138      Bryan(x) | Casey(x) | Garrett(x) | Shane(x) | Timothy(x).
139      Blue(x) | Green(x) | Orange(x) | Red(x) | Yellow(x).
140      Oval(x) | Rectangular(x) | Round(x) | Runner(x) | Square(x).
141      Birds(x) | Diamonds(x) | Lotuses(x) | Roses(x) | Stars(x).
142      FourHundred(x) | SixHundred(x) | EightHundred(x) | OneThousand(x) | TwelveHundred(x).
143      Bedroom(x) | Hall(x) | Library(x) | LivingRoom(x) | Office(x).
144
145      % Each property applies at most to one man
146      Bryan(x) & Bryan(y) -> -differentFrom(x, y).
147      Casey(x) & Casey(y) -> -differentFrom(x, y).
148      Garrett(x) & Garrett(y) -> -differentFrom(x, y).
149      Shane(x) & Shane(y) -> -differentFrom(x, y).
150      Timothy(x) & Timothy(y) -> -differentFrom(x, y).
151
152      Blue(x) & Blue(y) -> -differentFrom(x, y).
153      Green(x) & Green(y) -> -differentFrom(x, y).
154      Orange(x) & Orange(y) -> -differentFrom(x, y).
155      Red(x) & Red(y) -> -differentFrom(x, y).
156      Yellow(x) & Yellow(y) -> -differentFrom(x, y).
157
158      Oval(x) & Oval(y) -> -differentFrom(x, y).
159      Rectangular(x) & Rectangular(y) -> -differentFrom(x, y).
160      Round(x) & Round(y) -> -differentFrom(x, y).
161      Runner(x) & Runner(y) -> -differentFrom(x, y).
162      Square(x) & Square(y) -> -differentFrom(x, y).
163
164      Birds(x) & Birds(y) -> -differentFrom(x, y).
165      Diamonds(x) & Diamonds(y) -> -differentFrom(x, y).
166      Lotuses(x) & Lotuses(y) -> -differentFrom(x, y).
167      Roses(x) & Roses(y) -> -differentFrom(x, y).
168      Stars(x) & Stars(y) -> -differentFrom(x, y).
169
170      FourHundred(x) & FourHundred(y) -> -differentFrom(x, y).
171      SixHundred(x) & SixHundred(y) -> -differentFrom(x, y).
172      EightHundred(x) & EightHundred(y) -> -differentFrom(x, y).
173      OneThousand(x) & OneThousand(y) -> -differentFrom(x, y).
174      TwelveHundred(x) & TwelveHundred(y) -> -differentFrom(x, y).
175
176      Bedroom(x) & Bedroom(y) -> -differentFrom(x, y).
177      Hall(x) & Hall(y) -> -differentFrom(x, y).
178      Library(x) & Library(y) -> -differentFrom(x, y).
179      LivingRoom(x) & LivingRoom(y) -> -differentFrom(x, y).
180      Office(x) & Office(y) -> -differentFrom(x, y).
181
182      %Clues
183
184      % 1. The man who bought the $400 rug is exactly to the left of the man who bought
185      % the rug with Birds.
186      FourHundred(x) & Birds(y) -> directlyright(x, y).
187
188      % 2. Bryan is next to the client that got the Runner rug.
189      Bryan(x) & Runner(y) -> nextto(y, x).
190
191      % 3. The Red rug's owner is somewhere to the left of the man who bought a rug for
192      % his Hall.
193      Red(x) & Hall(y) -> right(x, y).
194
195      % 4. The buyer who got the most expensive rug is next to the buyer who got the
196      % Round rug.
197      TwelveHundred(x) & Round(y) -> nextto(y, x).
198
199      % 5. The Yellow rug's owner is somewhere between the Oval rug's owner and the man
200      % that spent $800, in that order.
201      Oval(x) & Yellow(y) & EightHundred(z) -> right(x, y) & right(y, z).

```

Persian Rugs (page 4 of 4)

```
202     % 7. Garrett is next to the customer who spent $600.  
203     Garrett(x) & SixHundred(y) -> nextto(y, x).  
204  
205     % 8. Shane is somewhere to the right of the Orange rug's owner.  
206     Shane(x) & Orange(y) -> right(y, x).  
207  
208     % 9. The cheapest rug has Stars on it.  
209     FourHundred(x) -> Stars(x).  
210  
211     % 10. At the fifth position is the client who bought the Rectangular rug.  
212     Man5(x) -> Rectangular(x).  
213  
214     % 11. The man who bought the $600 rug is exactly to the left of the man who bought  
215     the rug with Roses.  
216     SixHundred(x) & Roses(y) -> directlyright(x, y).  
217  
218     % 12. Casey is somewhere to the right of the customer that got the Yellow rug.  
219     Casey(x) & Yellow(y) -> right(y, x).  
220  
221     % 13. The buyer who purchased the Round rug is exactly to the left of the man who  
222     spent $800.  
223     Round(x) & EightHundred(y) -> directlyright(x, y).  
224  
225     % 14. The man that bought the cheapest rug is exactly to the left of the man that  
226     acquired the Square rug.  
227     FourHundred(x) & Square(y) -> directlyright(x, y).  
228  
229     % 15. At the third position is the man that got a rug for his Bedroom.  
230     Man3(x) -> Bedroom(x).  
231  
232     % 16. The customer that bought the rug with Stars is exactly to the right of the  
233     customer that bought the Green rug.  
234     Stars(x) & Green(y) -> directlyright(y, x).  
235  
236     % 17. Shane is next to the man who acquired the Round rug.  
237     Shane(x) & Round(y) -> nextto(y, x).  
238  
239     % 18. The man who got the rug with Birds is somewhere between the man who got the  
240     Yellow rug and the man who got the rug with Lotuses, in that order.  
241     Yellow(x) & Birds(y) & Lotuses(z) -> right(x, y) & right(y, z).  
242  
243     % 19. The customer that spent $800 is somewhere to the right of the Red rug's owner.  
244     EightHundred(x) & Red(y) -> right(y, x).  
245  
246     end_of_list.
```

Secret Agents (page 1 of 4)

```
1  if(Mace4).
2  assign(domain_size, 5).
3  end_if.
4
5  if(Prover9).
6  formulas(goals).
7
8  Agent4(d4) -> James(d4).
9
10 end_of_list.
11 end_if.
12
13 % Avoid using sentence names starting with u, v, w, x, y, z.
14
15 % Agents' order in row
16     % Agent1: first agent
17     % Agent2: second agent
18     % Agent3: third agent
19     % Agent4: fourth agent
20     % Agent5: fifth agent
21
22 % Agents' names
23     % Austin, James, Jason, Stan, Sterling
24
25 % Tie
26     % Black, Blue, Green, Purple, Red
27
28 % Country
29     % Australia, Brazil, Germany, Libya, Russia
30
31 % Accessory
32     % Clock, Pen, Phone, Ring, Umbrella
33
34 % Skill
35     % ComputerHacking, Disguise, Driving, MartialArts, Parkour
36
37 % Age
38     % TwentyFive, Thirty, ThirtyFive, Forty, FortyFive
39
40 formulas(assumptions).
41
42 % The 5 agents differ
43     differentFrom(d1, d2).
44     differentFrom(d1, d3).
45     differentFrom(d1, d4).
46     differentFrom(d1, d5).
47     differentFrom(d2, d3).
48     differentFrom(d2, d4).
49     differentFrom(d2, d5).
50     differentFrom(d3, d4).
51     differentFrom(d3, d5).
52     differentFrom(d4, d5).
53
54 % The "differentFrom" relation is symmetrical
55     differentFrom(x, y) -> differentFrom(y, x).
56
57     d1 = 0.
58     d2 = 1.
59     d3 = 2.
60     d4 = 3.
61     d5 = 4.
62
63     Agent1(0).
64     Agent2(1).
65     Agent3(2).
66     Agent4(3).
67     Agent5(4).
68
69 % Relation directlyright(x, y) that affirms that "agent y" is directly to the right of
```

Secret Agents (page 2 of 4)

```
"agent x".
70    directlyright(d1, d2).
71    directlyright(d2, d3).
72    directlyright(d3, d4).
73    directlyright(d4, d5).
74
75    -directlyright(d1, d1).
76    -directlyright(d2, d2).
77    -directlyright(d3, d3).
78    -directlyright(d4, d4).
79    -directlyright(d5, d5).
80    -directlyright(d1, d3).
81    -directlyright(d2, d4).
82    -directlyright(d3, d5).
83    -directlyright(d3, d1).
84    -directlyright(d4, d2).
85    -directlyright(d5, d3).
86    -directlyright(d2, d1).
87    -directlyright(d3, d2).
88    -directlyright(d4, d3).
89    -directlyright(d5, d4).
90    -directlyright(d1, d4).
91    -directlyright(d2, d5).
92    -directlyright(d4, d1).
93    -directlyright(d5, d2).
94    -directlyright(d1, d5).
95    -directlyright(d5, d1).
96
97 % Relation nextto(x, y) that affirms that "agent y" is next to "agent x".
98    directlyright(x, y) | directlyright(y, x) <-> nextto(x, y).
99
100 % Relation extremity(x) that affirms that "agent x" is at One of the ends.
101    extremity(d1).
102    extremity(d5).
103
104    -extremity(d2).
105    -extremity(d3).
106    -extremity(d4).
107
108 % Relation right(x, y) that affirms that "agent y" is somewhere to the right of "agent
x".
109    right(d1, d3).
110    right(d2, d4).
111    right(d3, d5).
112    right(d1, d4).
113    right(d2, d5).
114    right(d1, d5).
115    right(d1, d2).
116    right(d4, d5).
117    right(d2, d3).
118    right(d3, d4).
119
120    -right(d2, d1).
121    -right(d3, d2).
122    -right(d4, d3).
123    -right(d5, d4).
124    -right(d1, d1).
125    -right(d2, d2).
126    -right(d3, d3).
127    -right(d4, d4).
128    -right(d5, d5).
129    -right(d3, d1).
130    -right(d4, d2).
131    -right(d5, d3).
132    -right(d4, d1).
133    -right(d5, d2).
134    -right(d5, d1).
135
136 % Each agent has different characteristics
```

Secret Agents (page 3 of 4)

```
137 Agent1(x) | Agent2(x) | Agent3(x) | Agent4(x) | Agent5(x).
138 Austin(x) | James(x) | Jason(x) | Stan(x) | Sterling(x).
139 Black(x) | Blue(x) | Green(x) | Purple(x) | Red(x).
140 Australia(x) | Brazil(x) | Germany(x) | Libya(x) | Russia(x).
141 Clock(x) | Pen(x) | Phone(x) | Ring(x) | Umbrella(x).
142 ComputerHacking(x) | Disguise(x) | Driving(x) | MartialArts(x) | Parkour(x).
143 TwentyFive(x) | Thirty(x) | ThirtyFive(x) | Forty(x) | FortyFive(x).
144
145 % Each property applies at most to one agent
146 Austin(x) & Austin(y) -> -differentFrom(x, y).
147 James(x) & James(y) -> -differentFrom(x, y).
148 Jason(x) & Jason(y) -> -differentFrom(x, y).
149 Stan(x) & Stan(y) -> -differentFrom(x, y).
150 Sterling(x) & Sterling(y) -> -differentFrom(x, y).
151
152 Black(x) & Black(y) -> -differentFrom(x, y).
153 Blue(x) & Blue(y) -> -differentFrom(x, y).
154 Green(x) & Green(y) -> -differentFrom(x, y).
155 Purple(x) & Purple(y) -> -differentFrom(x, y).
156 Red(x) & Red(y) -> -differentFrom(x, y).
157
158 Australia(x) & Australia(y) -> -differentFrom(x, y).
159 Brazil(x) & Brazil(y) -> -differentFrom(x, y).
160 Germany(x) & Germany(y) -> -differentFrom(x, y).
161 Libya(x) & Libya(y) -> -differentFrom(x, y).
162 Russia(x) & Russia(y) -> -differentFrom(x, y).
163
164 Clock(x) & Clock(y) -> -differentFrom(x, y).
165 Pen(x) & Pen(y) -> -differentFrom(x, y).
166 Phone(x) & Phone(y) -> -differentFrom(x, y).
167 Ring(x) & Ring(y) -> -differentFrom(x, y).
168 Umbrella(x) & Umbrella(y) -> -differentFrom(x, y).
169
170 ComputerHacking(x) & ComputerHacking(y) -> -differentFrom(x, y).
171 Disguise(x) & Disguise(y) -> -differentFrom(x, y).
172 Driving(x) & Driving(y) -> -differentFrom(x, y).
173 MartialArts(x) & MartialArts(y) -> -differentFrom(x, y).
174 Parkour(x) & Parkour(y) -> -differentFrom(x, y).
175
176 TwentyFive(x) & TwentyFive(y) -> -differentFrom(x, y).
177 Thirty(x) & Thirty(y) -> -differentFrom(x, y).
178 ThirtyFive(x) & ThirtyFive(y) -> -differentFrom(x, y).
179 Forty(x) & Forty(y) -> -differentFrom(x, y).
180 FortyFive(x) & FortyFive(y) -> -differentFrom(x, y).
181
182 %Clues
183
184 %1. Austin is next to the agent wearing the Black tie.
185 Austin(x) & Black(y) -> nextto(y, x).
186
187 %2. The master of Disguise is exactly to the right of the agent that has a spy
188 Umbrella.
189 Disguise(x) & Umbrella(y) -> directlyright(y, x).
190
191 %3. The 35 years old agent is going to a mission on Tripoli.
192 ThirtyFive(x) -> Libya(x).
193
194 %4. James is the youngest secret agent.
195 James(x) -> TwentyFive(x).
196
197 %5. The agent that is going to Australia is next to the agent that is specialized
198 in Parkour.
199 Australia(x) & Parkour(y) -> nextto(y, x).
200
201 %6. James is exactly to the right of the agent that has a special Clock.
202 James(x) & Clock(y) -> directlyright(y, x).
203
204 %7. The spy that has an unique Umbrella is between the 40 years old agent and
205 Austin, in that order.
```

Secret Agents (page 4 of 4)

```
203     Forty(x) & Umbrella(y) & Austin(z) -> right(x, y) & right(y, z).
204
205     %8. Stan is next to the agent that is going to Asia.
206     Stan(x) & Russia(y) -> nextto(y, x).
207
208     %9. Sterling is at one of the ends.
209     Sterling(x) -> extremity(x).
210
211     %10. The man wearing the Red tie is 40 years old.
212     Red(x) -> Forty(x).
213
214     %11. The spy that is going to South America is exactly to the left of the 45 years
215     old spy.
216     Brazil(x) & FortyFive(y) -> directlyright(x, y).
217
218     %12. Jason is exactly to the left of Austin.
219     Jason(x) & Austin(y) -> directlyright(x, y).
220
221     %13. The Driver expert is next to the 30 years old man.
222     Driving(x) & Thirty(y) -> nextto(y, x).
223
224     %14. The 35 years old agent is next to the agent that is going to Sydney.
225     ThirtyFive(x) & Australia(y) -> nextto(y, x).
226
227     %15. The agent with advanced knowledge of Hacking is exactly to the left of the 35
228     years old man.
229     ComputerHacking(x) & ThirtyFive(y) -> directlyright(x, y).
230
231     %16. The spy wearing the Purple tie is next to the Geek spy.
232     Purple(x) & ComputerHacking(y) -> nextto(y, x).
233
234     %17. Austin is 30.
235     Austin(x) -> Thirty(x).
236
237     %18. The agent that has a special Phone is exactly to the left of the agent that is
238     going to Africa.
239     Phone(x) & Libya(y) -> directlyright(x, y).
240
241     %19. The agent wearing a spy Ring is somewhere to the right of the agent wearing
242     the Purple tie.
243     Ring(x) & Purple(y) -> right(y, x).
244
245     %20. At the second position is the spy wearing the Green tie.
246     Agent2(x) -> Green(x).
247
248     end_of_list.
```

A.2 Săsăran Andrei Paul's code:

A.2.1 Search

Code starts from the next page.

```

1  from Tkinter import *
2  import tkFont
3  import os
4  import sys
5  from model import *
6
7
8  from random import randint
9
10 def main():
11     root = Tk()
12     app = PuzzleWindow(root)
13
14 class PuzzleWindow:
15
16     def __init__(self,master):
17         self.master = master
18         self.master.title("8 PUZZLE")
19         self.master.geometry("220x400+900+200")
20         self.master.resizable(0, 0)
21         self.master.config(bg="#5DADE2")
22         self.frame = Frame(self.master, background="#D6EAF8")
23         self.frame.pack()
24
25         self.goalWindow = Toplevel()
26         self.goalWindow.geometry("250x300+1123+200")
27         rootPath = os.path.dirname(os.path.abspath("goalState.gif"))
28         rootPath = rootPath[:-4]
29         rootPath = rootPath + "goalState.gif"
30         goalImage = PhotoImage(file=rootPath)
31         goalLabel = Label(self.goalWindow, image=goalImage)
32         goalLabel.image = goalImage
33         goalLabel.pack()
34         self.goalWindow.overrideredirect(1)
35         self.goalWindow.deiconify()
36
37
38
39
40
41         shuffleFrame = LabelFrame(self.frame, width=300, height=50, background="#5DADE2")
42         puzzleFrame = LabelFrame(self.frame, width=300, height=300, background="#2874A6")
43         solutionFrame = LabelFrame(self.frame, width=300, height=120,
44         background="#AED6F1")
45         shuffleFrame.pack()
46         puzzleFrame.pack()
47         solutionFrame.pack()
48
49         # SHUFFLE FRAME
50         shuffleButton = Button(shuffleFrame, text="SHUFFLE", command =
51         self.shuffleFunction, bd=4, bg="#7FB3D5",width=10, height=1, font=tkFont.BOLD
52         ).pack(side=LEFT)
53
54         self.showGoal = IntVar(value=1)
55         showGoalButton = Checkbutton(shuffleFrame, text="show GOAL",
56         variable=self.showGoal, command = self.showGoalFunction, bd = 1, bg="#7FB3D5",
57         width=10, height=1).pack(side=RIGHT)
58
59
60         # PUZZLE FRAME
61         self.puzzleCells = []
62         self.emptyCell = self.puzzleCells[0]
63         noOfCells = 0
64
65         # initialize puzzle
66         for i in range (3):
67             for j in range (3):

```

```

63             cell = Label(puzzleFrame, text =
64                 str(noOfCells), borderwidth=2, relief="groove", width=5, height=3,
65                 font=tkFont.BOLD, bg="#AED6F1")
66             cell.grid(row = i, column = j)
67             self.puzzleCells.append(cell)
68             noOfCells = noOfCells + 1
69             self.puzzleCells[0].config(text = "")
70             self.emptyCellIndexGlobal = 0
71             self.inputPuzzle = []
72
73             # SOLUTION FRAME
74             solutionLabel = Label(solutionFrame, text = "SOLUTION:", font = tkFont.NORMAL,
75                 bg="#D4E6F1").pack(side=LEFT)
76             solveButton = Button(solutionFrame, text="SOLVE", command =
77                 self.solveFunction, bd=4, bg="#7FB3D5", width=10, height=1, font=tkFont.BOLD
78                 ).pack(side=RIGHT)
79
80
81             self.master.mainloop()
82
83             # SHUFFLE FRAME
84             def showGoalFunction (self):
85                 choose = self.showGoal.get()
86
87                 if choose:
88                     # publici fereastra
89                     self.goalWindow.deiconify()
90
91                 else:
92                     # ascunzi fereastra
93                     self.goalWindow.withdraw()
94
95             def shuffleFunction(self):
96                 #reset the inputPuzzle
97                 self.inputPuzzle = []
98
99                 emptyCellIndex = self.emptyCellIndexGlobal
100                permutations = randint(10,30)
101                for i in range (0, permutations):
102                    successors = self.getSuccessors(emptyCellIndex) # get successors
103                    randomSuccessorIndex = successors[randint(0, len(successors)) - 1] # pick a
104                    random successor
105
106
107                    currentLabel = self.puzzleCells[emptyCellIndex]
108                    successorLabel = self.puzzleCells[randomSuccessorIndex]
109                    successorValue = successorLabel['text']
110
111                    currentLabel.configure(text=successorValue)
112                    successorLabel.configure(text=' ')
113
114                    emptyCellIndex = randomSuccessorIndex
115                    self.emptyCellIndexGlobal = emptyCellIndex
116
117
118             # PUZZLE FRAME
119
120             def getSuccessors(self, cellIndex):
121                 #cellIndex = self.puzzleCells.index(cell) # position of cell in puzzleCells list
122                 row = cellIndex / 3 # position of cell in context
123                 col = cellIndex % 3 # of matrix
124
125                 successors = []
126
127                 if (row > 0): # not on the first row

```

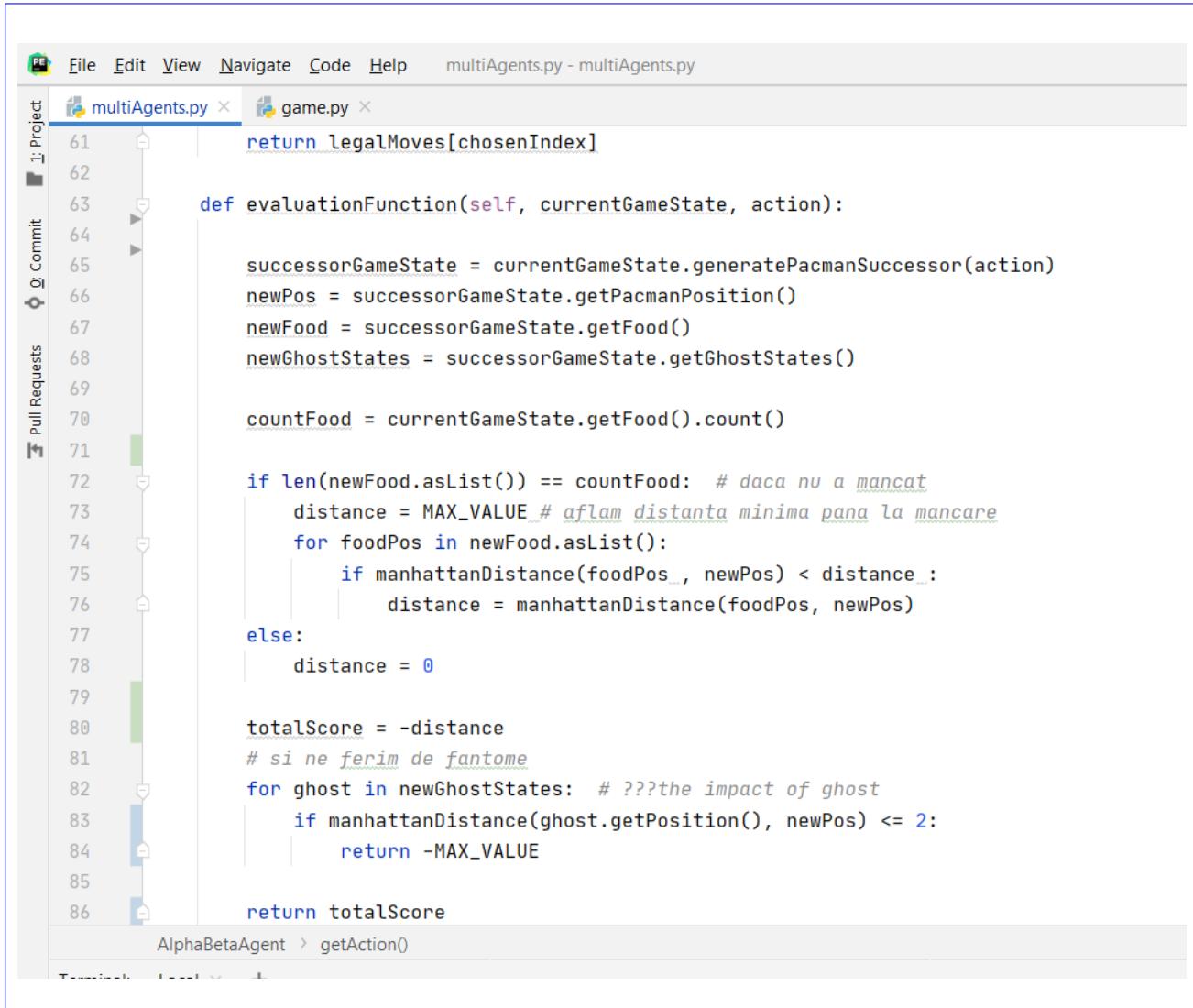
```

124         successors.append(cellIndex - 3) # add the NORTH successor
125     if (col < 2): # not on the right col
126         successors.append(cellIndex + 1) # add the EAST successor
127     if (row < 2): # not on the last row
128         successors.append(cellIndex + 3) # add the SOUTH successor
129     if (col > 0): # not on the left col
130         successors.append(cellIndex - 1) # add the WEST successor
131
132     return successors
133
134
135
136 def solveFunction(self):
137     self.transferPuzzle()
138     #print ("before")
139     #print self.inputPuzzle
140     self.model = PuzzleModel(self.inputPuzzle)
141     #print ("after")
142     #print self.inputPuzzle
143
144     self.model.solveFunction()
145
146
147 def transferPuzzle (self):
148     count = 0
149     for i in range(0,3):
150         for j in range(0,3):
151             labelText = self.puzzleCells[count].cget("text")
152             if (labelText == ''):
153                 self.inputPuzzle.append(0)
154             else:
155                 self.inputPuzzle.append(int(labelText))
156             count = count + 1
157
158
159
160
161 if __name__ == '__main__':
162     main()
163
164
165

```

BONUS

Reflex Agent: (page 1 of 1)



The screenshot shows a code editor interface with a Python file named `multiAgents.py` open. The code implements a `ReflexAgent` class that overrides the `getAction()` method. The implementation uses a `maximize` function to choose the best action based on the maximum score from all legal moves. The score is calculated by evaluating the successor game state after each action, taking into account the distance to food and the impact of ghosts.

```
File Edit View Navigate Code Help multiAgents.py - multiAgents.py
Project 1: Project
multiAgents.py × game.py ×
61     return legalMoves[chosenIndex]
62
63 def evaluationFunction(self, currentGameState, action):
64
65     successorGameState = currentGameState.generatePacmanSuccessor(action)
66     newPos = successorGameState.getPacmanPosition()
67     newFood = successorGameState.getFood()
68     newGhostStates = successorGameState.getGhostStates()
69
70     countFood = currentGameState.getFood().count()
71
72     if len(newFood.asList()) == countFood: # daca nu a mancat
73         distance = MAX_VALUE # aflam distanta minima pana la mancare
74         for foodPos in newFood.asList():
75             if manhattanDistance(foodPos, newPos) < distance :
76                 distance = manhattanDistance(foodPos, newPos)
77         else:
78             distance = 0
79
80     totalScore = -distance
81     # si ne ferim de fantome
82     for ghost in newGhostStates: # ???the impact of ghost
83         if manhattanDistance(ghost.getPosition(), newPos) <= 2:
84             return -MAX_VALUE
85
86     return totalScore
AlphaBetaAgent > getAction()
```

Minimax Agent: (page 1 of 3)

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, View, Navigate, Code, Help.
- Project Tab:** multiAgents.py (selected), game.py.
- Code Area:** Displays Python code for a Minimax Agent. The code includes functions for getting actions, performing minimax search, calculating max values, and generating successors. It uses comments to explain its logic, such as calculating s-a calculations and handling the first agent.
- Terminal:** Local

```
124
125 def getAction(self, gameState):
126     |
127     def hMinimax (gameState, currentDepth, agentCount):
128
129         if agentCount == gameState.getNumAgents(): #daca s-a calculat
130             # o mutare a fiecarui agent => creste depth
131             agentCount = 0 # se reinince cu pacman
132             currentDepth = currentDepth + 1
133             if currentDepth == self.depth or gameState.isWin() or gameState.isLose():
134                 return self.evaluationFunction(gameState) # evaluam functia
135             elif agentCount == 0: # pacman e MAX
136                 return maxValue(gameState, currentDepth, agentCount)
137             else:
138                 return minValue(gameState, currentDepth, agentCount)
139
140         def maxValue (gameState, currentDepth, agentCount):
141             maxAction = [-maxsize, ""] # in order to find action of
142             # max value, we store a pair of [maxValue, actionOfMaxValue]
143             legalActions = gameState.getLegalActions(agentCount)
144             if not legalActions:
145                 return self.evaluationFunction(gameState)
146
147             for action in legalActions:
148                 nextState = gameState.generateSuccessor(agentCount, action)
149                 value = hMinimax(nextState, currentDepth, agentCount + 1)
MinimaxAgent > getAction()
Terminal: Local
```

Minimax Agent: (page 2 of 3)

```
File Edit View Navigate Code Help multiAgents.py - multiAgents.py
multiAgents.py game.py
160
161
162     def minValue (gameState, currentDepth, agentCount):
163         minAction = [maxsize, ""] # in order to find action of
164         # min value, we store a pair of [minValue, actionOfMinValue]
165         legalActions = gameState.getLegalActions(agentCount)
166         if not legalActions:
167             return self.evaluationFunction(gameState)
168
169         for action in legalActions:
170             nextState = gameState.generateSuccessor(agentCount, action)
171             value = hMinimax(nextState, currentDepth, agentCount + 1)
172             # trecem la urmatorul agent
173
174             if type(value) is list:
175                 minValue = value[0]
176             else:
177                 minValue = value
178
179             if minValue < minAction[0]:
180                 minAction = [minValue, action] # change the result
181
182         return minAction # return just the action as a result
183
184         resultMove = hMinimax(gameState, 0, 0)[1] #pacman muta primu
185         return resultMove
MinimaxAgent > getAction()
Terminal: Local +
```

Your grades are NOT yet registered. To register your grades, make sure to follow your instructor's guidelines to receive credit on your project.

```
D:\AI\github\InteligentaArtificiala\LABORATOARE\LAB1\multiagent>
```

Git Problems Terminal Run Python Console

System clipboard is unavailable (today 4:28 PM)

Minimax Agent: (page 3 of 3)

```
File Edit View Navigate Code Help multiAgents.py - multiAgents.py
multiAgents.py game.py
1: Project
136     return maxValue(gameState, currentDepth, agentCount)
137
138     else:
139         return minValue(gameState, currentDepth, agentCount)
140
141 def maxValue (gameState, currentDepth, agentCount):
142     maxAction = [-maxsize, ""] # in order to find action of
143     # max value, we store a pair of [maxValue, actionOfMaxValue]
144     legalActions = gameState.getLegalActions(agentCount)
145     if not legalActions:
146         return self.evaluationFunction(gameState)
147
148     for action in legalActions:
149         nextState = gameState.generateSuccessor(agentCount, action)
150         value = hMinimax(nextState, currentDepth, agentCount + 1)
151
152         if type(value) is list:
153             testValue = value[0]
154         else:
155             testValue = value
156
157         if testValue > maxAction[0]:
158             maxAction = [testValue, action] # change the result
159
160
161 return maxAction # return just the action + value
MinimaxAgent > getAction()
Terminal: Local +
```

Your grades are NOT yet registered. To register your grades, make sure to follow your instructor's guidelines to receive credit on your project.

```
D:\AI\github\InteligentaArtificiala\LABORATOARE\LAB1\multiagent>
```

Git Problems Terminal Run Python Console

System clipboard is unavailable (today 4:28 PM)

A.2.2 Logics

Who likes dolphins? (page 1 of 4)

```
1  if(Mace4).
2  assign(domain_size, 4).
3  end_if.
4
5  if(Prover9).
6  formulas(goals).
7
8  HouseThree(h3) -> Canadian(h3).
9
10 end_of_list.
11 end_if.
12
13 % Avoid using sentence names starting with u, v, w, x, y, z.
14
15 % House number
16   % HouseOne: you live in house one
17   % HouseTwo: you live in house two
18   % HouseThree: you live in house three
19   % HouseFour: you live in house four
20
21 % Nationality
22   % American: you are american
23   % British: you are british
24   % Canadian: you are canadian
25   % Irish: you are irish
26
27 % Animal
28   % Butterflies: you are the person with the butterflies
29   % Dolphins: you are the person with dolphins
30   % Horses: you are the person with the horses
31   % Turtles: you are the person with the turtles
32
33 % Sport
34   % Bowling: you play bowling
35   % Handball: you play handball
36   % Swimming: you play swimming
37   % Tennis: you play tennis
38
39 % Colour
40   % Black: the house is black
41   % Blue: the house is blue
42   % Red: the house is red
43   % White: the house is white
44
45 formulas(assumptions).
46
47 % The three houses differ
48   differentFrom(h1, h2).
49   differentFrom(h1, h3).
50   differentFrom(h1, h4).
51   differentFrom(h2, h3).
52   differentFrom(h2, h4).
53   differentFrom(h3, h4).
54
55 % The "differentFrom" relation is symmetrical
56   differentFrom(x, y) -> differentFrom(y, x).
57
58 % Each house is assigned an order number, with reference to their position
59   h1 = 0.
60   h2 = 1.
61   h3 = 2.
62   h4 = 3.
63
64   HouseOne(0).
65   HouseTwo(1).
66   HouseThree(2).
67   HouseFour(3).
68
69 % Relation right(x, y) that affirms that "house y" is somewhere to the right of "house x"
```

Who likes dolphins? (page 2 of 4)

```
70      right(h1, h2).
71      right(h1, h3).
72      right(h1, h4).
73      right(h2, h3).
74      right(h2, h4).
75      right(h3, h4).
76
77      -right(h1, h1).
78      -right(h2, h2).
79      -right(h3, h3).
80      -right(h4, h4).
81      -right(h3, h1).
82      -right(h4, h2).
83      -right(h2, h1).
84      -right(h3, h2).
85      -right(h4, h3).
86      -right(h4, h1).
87
88
89 % Relation rightneighbour(x, y) that affirms that "house y" is directly to the right of
90 % "house x"
91      rightneighbour(h1, h2).
92      rightneighbour(h2, h3).
93      rightneighbour(h3, h4).
94
95      -rightneighbour(h1, h1).
96      -rightneighbour(h2, h2).
97      -rightneighbour(h3, h3).
98      -rightneighbour(h4, h4).
99      -rightneighbour(h1, h3).
100     -rightneighbour(h2, h4).
101     -rightneighbour(h3, h1).
102     -rightneighbour(h4, h2).
103     -rightneighbour(h2, h1).
104     -rightneighbour(h3, h2).
105     -rightneighbour(h4, h3).
106     -rightneighbour(h4, h1).
107     -rightneighbour(h1, h4).
108
109    % Relation middleneighbour(x, y) that affirms that there is one house between "house x"
110    % and "house y"
111    middleneighbour(h1, h3).
112    middleneighbour(h3, h1).
113    middleneighbour(h2, h4).
114    middleneighbour(h4, h2).
115
116    -middleneighbour(h1, h2).
117    -middleneighbour(h2, h3).
118    -middleneighbour(h3, h4).
119    -middleneighbour(h1, h1).
120    -middleneighbour(h2, h3).
121    -middleneighbour(h3, h3).
122    -middleneighbour(h4, h4).
123    -middleneighbour(h2, h1).
124    -middleneighbour(h3, h2).
125    -middleneighbour(h4, h3).
126    -middleneighbour(h4, h1).
127
128    % Relation inbetween(x, y) that affirms that there are two houses between "house x" and
129    % "house y"
130    inbetween(h4, h1).
131    inbetween(h1, h4).
132
133    -inbetween(h1, h3).
134    -inbetween(h3, h1).
135    -inbetween(h2, h4).
136    -inbetween(h4, h2).
137    -inbetween(h1, h2).
```

Who likes dolphins? (page 3 of 4)

```
136      -inbetween(h2, h3).
137      -inbetween(h3, h4).
138      -inbetween(h1, h1).
139      -inbetween(h2, h3).
140      -inbetween(h3, h3).
141      -inbetween(h4, h4).
142      -inbetween(h2, h1).
143      -inbetween(h3, h2).
144      -inbetween(h4, h3).
145
146 % Each house is different and has a colour, nationality, animal and sport
147 HouseOne(x) | HouseTwo(x) | HouseThree(x) | HouseFour(x).
148 Black(x) | Blue(x) | Red(x) | White(x).
149 American(x) | British(x) | Canadian(x) | Irish(x).
150 Butterflies(x) | Dolphins(x) | Horses(x) | Turtles(x).
151 Bowling(x) | Handball(x) | Swimming(x) | Tennis(x).
152
153 % Each property applies at most to one house
154 Black(x) & Black(y) -> -differentFrom(x, y).
155 Blue(x) & Blue(y) -> -differentFrom(x, y).
156 Red(x) & Red(y) -> -differentFrom(x, y).
157 White(x) & White(y) -> -differentFrom(x, y).
158
159 American(x) & American(y) -> -differentFrom(x, y).
160 British(x) & British(y) -> -differentFrom(x, y).
161 Canadian(x) & Canadian(y) -> -differentFrom(x, y).
162 Irish(x) & Irish(y) -> -differentFrom(x, y).
163
164 Butterflies(x) & Butterflies(y) -> -differentFrom(x, y).
165 Dolphins(x) & Dolphins(y) -> -differentFrom(x, y).
166 Horses(x) & Horses(y) -> -differentFrom(x, y).
167 Turtles(x) & Turtles(y) -> -differentFrom(x, y).
168
169 Bowling(x) & Bowling(y) -> -differentFrom(x, y).
170 Handball(x) & Handball(y) -> -differentFrom(x, y).
171 Swimming(x) & Swimming(y) -> -differentFrom(x, y).
172 Tennis(x) & Tennis(y) -> -differentFrom(x, y).
173
174 %Clues
175
176 % 1. The second house is Black.
177 HouseTwo(x) -> Black(x).
178
179 % 2. The British lives in the first house.
180 HouseOne(x) -> British(x).
181
182 % 3. There are two houses between the person who likes Bowling and the person who
183 likes Swimming.
184 Bowling(x) & Swimming(y) -> inbetween(x, y).
185
186 % 4. The American live directly to the left of the person who likes Turtles.
187 American(x) & Turtles(y) -> rightneighbour(x, y).
188
189 % 5. There are two houses between the person who likes Horses and the person who
190 likes Butterflies on the right.
191 Horses(x) & Butterflies(y) -> inbetween(x, y).
192 HouseFour(x) -> Butterflies(x).
193
194 % 6. The person who likes Bowling lives somewhere to the right of the person who
195 likes Tennis.
196 Bowling(x) & Tennis(y) -> right(y, x).
197
198 % 7. There is one house between the Irish and the person who likes Handball on the
199 left.
200 Irish(x) & Handball(y) -> middleneighbour(x, y).
201 Irish(x) & Handball(y) -> right(y, x).
202
203 % 8. There is one house between the person who likes Horses and the Red house on
204 the right.
```

Who likes dolphins? (page 4 of 4)

```
200      Horses(x) & Red(y) -> middleneighbour(x, y).
201      Horses(x) & Red(y) -> right(x, y).
202
203      % 9. There is one house between the person who likes Handball and the White house
204      % on the right.
205      Handball(x) & White(y) -> middleneighbour(x, y).
206      Handball(x) & White(y) -> right(x, y).
207      end_of_list.
208
209
210
```

Hardware Store (page 1 of 5)

```
1  if(Mace4).
2  assign(domain_size, 5).
3  end_if.
4
5  if(Prover9).
6  formulas(goals).
7
8  Man1(m1) -> Wrench(m1).
9
10 end_of_list.
11 end_if.
12
13 % Avoid using sentence names starting with u, v, w, x, y, z.
14
15 % Men's order in row
16     % Man1: first man
17     % Man2: second man
18     % Man3: third man
19     % Man4: fourth man
20     % Man5: fifth man
21
22 % Men's names
23     % Billy, Dennis, George, Larry, Philip
24
25 % Shirt
26     % Blue, Green, Red, White, Yellow
27
28 % Tool
29     % Drill, Hammer, Saw, Screwdriver, Wrench
30
31 % Discount
32     % FivePercent, TenPercent, FifteenPercent, TwentyPercent, TwentyFivePercent
33
34 % Profession
35     % Electrician, Engineer, Mechanic, Plumber, Technician
36
37 % Age
38     % ThirtyFive, Forty, FortyFive, Fifty, FiftyFive
39
40 formulas(assumptions).
41
42 % The 5 men differ
43     differentFrom(m1, m2).
44     differentFrom(m1, m3).
45     differentFrom(m1, m4).
46     differentFrom(m1, m5).
47     differentFrom(m2, m3).
48     differentFrom(m2, m4).
49     differentFrom(m2, m5).
50     differentFrom(m3, m4).
51     differentFrom(m3, m5).
52     differentFrom(m4, m5).
53
54 % The "differentFrom" relation is symmetrical
55     differentFrom(x, y) -> differentFrom(y, x).
56
57     m1 = 0.
58     m2 = 1.
59     m3 = 2.
60     m4 = 3.
61     m5 = 4.
62
63     Man1(0).
64     Man2(1).
65     Man3(2).
66     Man4(3).
67     Man5(4).
68
69 % Relation directlyright(x, y) that affirms that "man y" is directly to the right of
```

Hardware Store (page 2 of 5)

```
"man x"
70    directlyright(m1, m2).
71    directlyright(m2, m3).
72    directlyright(m3, m4).
73    directlyright(m4, m5).
74
75    -directlyright(m1, m1).
76    -directlyright(m2, m2).
77    -directlyright(m3, m3).
78    -directlyright(m4, m4).
79    -directlyright(m5, m5).
80    -directlyright(m1, m3).
81    -directlyright(m2, m4).
82    -directlyright(m3, m5).
83    -directlyright(m3, m1).
84    -directlyright(m4, m2).
85    -directlyright(m5, m3).
86    -directlyright(m2, m1).
87    -directlyright(m3, m2).
88    -directlyright(m4, m3).
89    -directlyright(m5, m4).
90    -directlyright(m1, m4).
91    -directlyright(m2, m5).
92    -directlyright(m4, m1).
93    -directlyright(m5, m2).
94    -directlyright(m1, m5).
95    -directlyright(m5, m1).
96
97 % Relation nextto(x, y) that affirms that "man y" is next to "man x"
98    nextto(m1, m2).
99    nextto(m2, m1).
100   nextto(m2, m3).
101   nextto(m3, m2).
102   nextto(m3, m4).
103   nextto(m4, m3).
104   nextto(m4, m5).
105   nextto(m5, m4).
106
107  -nextto(m1, m1).
108  -nextto(m2, m2).
109  -nextto(m3, m3).
110  -nextto(m4, m4).
111  -nextto(m5, m5).
112  -nextto(m1, m3).
113  -nextto(m2, m4).
114  -nextto(m3, m5).
115  -nextto(m3, m1).
116  -nextto(m4, m2).
117  -nextto(m5, m3).
118  -nextto(m1, m4).
119  -nextto(m2, m5).
120  -nextto(m4, m1).
121  -nextto(m5, m2).
122  -nextto(m1, m5).
123  -nextto(m5, m1).
124
125 % Relation extremity(x) that affirms that "man x" is at one of the ends
126    extremity(m1).
127    extremity(m5).
128
129    -extremity(m2).
130    -extremity(m3).
131    -extremity(m4).
132
133 % Relation inbetween(x, y) that affirms that there is a man somewhere in between "man
134 x" and "man y", exactly in that order
135    inbetween(m1, m3).
136    inbetween(m2, m4).
137    inbetween(m3, m5).
```

Hardware Store (page 3 of 5)

```
137     inbetween(m1, m4).  
138     inbetween(m2, m5).  
139     inbetween(m1, m5).  
140  
141     -inbetween(m1, m2).  
142     -inbetween(m2, m1).  
143     -inbetween(m2, m3).  
144     -inbetween(m3, m2).  
145     -inbetween(m3, m4).  
146     -inbetween(m4, m3).  
147     -inbetween(m4, m5).  
148     -inbetween(m5, m4).  
149     -inbetween(m1, m1).  
150     -inbetween(m2, m2).  
151     -inbetween(m3, m3).  
152     -inbetween(m4, m4).  
153     -inbetween(m5, m5).  
154     -inbetween(m3, m1).  
155     -inbetween(m4, m2).  
156     -inbetween(m5, m3).  
157     -inbetween(m4, m1).  
158     -inbetween(m5, m2).  
159     -inbetween(m5, m1).  
160  
161 % Relation right(x, y) that affirms that "man y" is somewhere to the right of "man x"  
162     right(m1, m3).  
163     right(m2, m4).  
164     right(m3, m5).  
165     right(m1, m4).  
166     right(m2, m5).  
167     right(m1, m5).  
168     right(m1, m2).  
169     right(m4, m5).  
170     right(m2, m3).  
171     right(m3, m4).  
172  
173     -right(m2, m1).  
174     -right(m3, m2).  
175     -right(m4, m3).  
176     -right(m5, m4).  
177     -right(m1, m1).  
178     -right(m2, m2).  
179     -right(m3, m3).  
180     -right(m4, m4).  
181     -right(m5, m5).  
182     -right(m3, m1).  
183     -right(m4, m2).  
184     -right(m5, m3).  
185     -right(m4, m1).  
186     -right(m5, m2).  
187     -right(m5, m1).  
188  
189 % Each man has different characteristics  
190     Man1(x) | Man2(x) | Man3(x) | Man4(x) | Man5(x).  
191     Billy(x) | Dennis(x) | George(x) | Larry(x) | Philip(x).  
192     Blue(x) | Green(x) | Red(x) | White(x) | Yellow(x).  
193     Drill(x) | Hammer(x) | Saw(x) | Screwdriver(x) | Wrench(x).  
194     FivePercent(x) | TenPercent(x) | FifteenPercent(x) | TwentyPercent(x) |  
195     TwentyFivePercent(x).  
196     Electrician(x) | Engineer(x) | Mechanic(x) | Plumber(x) | Technician(x).  
197     ThirtyFive(x) | Forty(x) | FortyFive(x) | Fifty(x) | FiftyFive(x).  
198  
199 % Each property applies at most to one man  
200     Billy(x) & Billy(y) -> -differentFrom(x, y).  
201     Dennis(x) & Dennis(y) -> -differentFrom(x, y).  
202     George(x) & George(y) -> -differentFrom(x, y).  
203     Larry(x) & Larry(y) -> -differentFrom(x, y).  
204     Philip(x) & Philip(y) -> -differentFrom(x, y).
```

Hardware Store (page 4 of 5)

```
205     Blue(x) & Blue(y) -> -differentFrom(x, y).
206     Green(x) & Green(y) -> -differentFrom(x, y).
207     Red(x) & Red(y) -> -differentFrom(x, y).
208     White(x) & White(y) -> -differentFrom(x, y).
209     Yellow(x) & Yellow(y) -> -differentFrom(x, y).
210
211     Drill(x) & Drill(y) -> -differentFrom(x, y).
212     Hammer(x) & Hammer(y) -> -differentFrom(x, y).
213     Saw(x) & Saw(y) -> -differentFrom(x, y).
214     Screwdriver(x) & Screwdriver(y) -> -differentFrom(x, y).
215     Wrench(x) & Wrench(y) -> -differentFrom(x, y).
216
217     FivePercent(x) & FivePercent(y) -> -differentFrom(x, y).
218     TenPercent(x) & TenPercent(y) -> -differentFrom(x, y).
219     FifteenPercent(x) & FifteenPercent(y) -> -differentFrom(x, y).
220     TwentyPercent(x) & TwentyPercent(y) -> -differentFrom(x, y).
221     TwentyFivePercent(x) & TwentyFivePercent(y) -> -differentFrom(x, y).
222
223     Electrician(x) & Electrician(y) -> -differentFrom(x, y).
224     Engineer(x) & Engineer(y) -> -differentFrom(x, y).
225     Mechanic(x) & Mechanic(y) -> -differentFrom(x, y).
226     Plumber(x) & Plumber(y) -> -differentFrom(x, y).
227     Technician(x) & Technician(y) -> -differentFrom(x, y).
228
229     ThirtyFive(x) & ThirtyFive(y) -> -differentFrom(x, y).
230     Forty(x) & Forty(y) -> -differentFrom(x, y).
231     FortyFive(x) & FortyFive(y) -> -differentFrom(x, y).
232     Fifty(x) & Fifty(y) -> -differentFrom(x, y).
233     FiftyFive(x) & FiftyFive(y) -> -differentFrom(x, y).
234
235 %Clues
236
237     % 1. The man wearing the Yellow shirt is somewhere to the left of the Mechanic.
238     Yellow(x) & Mechanic(y) -> right(x, y).
239
240     % 2. Billy is buying a tool with 20% off.
241     Billy(x) -> TwentyPercent(x).
242
243     % 3. The guy buying a tool with 15% discount is at one of the ends.
244     FifteenPercent(x) -> extremity(x).
245
246     % 4. The 45 years old man is somewhere between the Plumber and the 35 years old
247     man, in that order.
248     Plumber(x) & ThirtyFive(y) -> inbetween(x, y).
249     FortyFive(x) & Plumber(y) -> right(y, x).
250     FortyFive(x) & ThirtyFive(y) -> right(x, y).
251
252     % 5. The one buying a tool with the biggest discount is next to the 50 years old man.
253     TwentyFivePercent(x) & Fifty(y) -> nextto(y, x).
254
255     % 6. Dennis is wearing the Green shirt.
256     Dennis(x) -> Green(x).
257
258     % 7. The 50 years old man is in the middle.
259     Man3(x) -> Fifty(x).
260
261     % 8. Larry is next to the man that is buying a tool with the smallest discount.
262     Larry(x) & TenPercent(y) -> nextto(y, x).
263
264     % 9. The one buying a Hammer is somewhere between the man buying the Wrench and the
265     man buying the Screwdriver, in that order.
266     Wrench(x) & Screwdriver(y) -> inbetween(x, y).
267     Hammer(x) & Wrench(y) -> right(y, x).
268     Hammer(x) & Screwdriver(y) -> right(x, y).
269
270     % 10. The Technician is at the third position.
271     Man3(x) -> Technician(x).
272
273     % 11. The Plumber is next to the man wearing the Green shirt.
```

Hardware Store (page 5 of 5)

```
272     Plumber(x) & Green(y) -> nextto(y, x).
273
274     % 12. The oldest man is buying a Drill.
275     FiftyFive(x) -> Drill(x).
276
277     % 13. The guy wearing the White shirt is somewhere to the right of the man wearing
278     the Blue shirt.
279     White(x) & Blue(y) -> right(y, x).
280
281     % 14. The man buying a Hammer is somewhere between the one who is getting 15%
282     discount and the one buying a Saw, in that order.
283     FifteenPercent(x) & Saw(y) -> inbetween(x, y).
284     Hammer(x) & FifteenPercent(y) -> right(y, x).
285     Hammer(x) & Saw(y) -> right(x, y).
286
287     % 15. George is next to the Electrician.
288     George(x) & Electrician(y) -> nextto(y, x).
289
290     % 16. The man buying tools with 5% and 25% off are next to each other.
291     FivePercent(x) & TwentyFivePercent(y) -> nextto(x, y).
292     FivePercent(x) & TwentyFivePercent(y) -> nextto(y, x).
293
294     % 17. Philip is buying a Saw.
295     Philip(x) -> Saw(x).
296
297     % 18. At the second position is the man buying a Drill.
298     Man2(x) -> Drill(x).
299
300     % 19. The man wearing the Yellow shirt is next to the one buying a Saw.
301     Yellow(x) & Saw(y) -> nextto(y, x).
302
303     % 20. The guy getting the best discount is somewhere between the 55 years old man
304     and the man buying a tool with 10% off, in that order.
305     FiftyFive(x) & TenPercent(y) -> inbetween(x, y).
306     TwentyFivePercent(x) & FiftyFive(y) -> right(y, x).
307     TwentyFivePercent(x) & TenPercent(y) -> right(x, y).
308
309     % 21. The man wearing the Red shirt is next to the man buying a tool with 15%
310     discount.
311     Red(x) & FifteenPercent(y) -> nextto(y, x).
312
313     end_of_list.
```

Black Friday (page 1 of 5)

```
1  if(Mace4).
2  assign(domain_size, 5).
3  end_if.
4
5  if(Prover9).
6  formulas(goals).
7
8  Man3(m3) -> Hank(m3).
9
10 end_of_list.
11 end_if.
12
13 % Avoid using sentence names starting with u, v, w, x, y, z.
14
15 % Men's order in row
16     % Man1: first man
17     % Man2: second man
18     % Man3: third man
19     % Man4: fourth man
20     % Man5: fifth man
21
22 % Men's names
23     % Dustin, Eugene, Hank, Keith, Sean
24
25 % Shirt
26     % Black, Blue, Green, Red, White
27
28 % Deal
29     % BeardTrimmer, GameConsole, Laptop, Smartphone, TV
30
31 % Discount
32     % FortyPercent, FiftyPercent, SixtyPercent, SeventyPercent, EightyPercent
33
34 % Age
35     % TwentyFive, Thirty, ThirtyFive, Forty, FortyFive
36
37 % Juice
38     % Apple, Cranberry, Grape, Lemon, Orange
39
40 formulas(assumptions).
41
42 % The 5 men differ
43     differentFrom(m1, m2).
44     differentFrom(m1, m3).
45     differentFrom(m1, m4).
46     differentFrom(m1, m5).
47     differentFrom(m2, m3).
48     differentFrom(m2, m4).
49     differentFrom(m2, m5).
50     differentFrom(m3, m4).
51     differentFrom(m3, m5).
52     differentFrom(m4, m5).
53
54 % The "differentFrom" relation is symmetrical
55     differentFrom(x, y) -> differentFrom(y, x).
56
57     m1 = 0.
58     m2 = 1.
59     m3 = 2.
60     m4 = 3.
61     m5 = 4.
62
63     Man1(0).
64     Man2(1).
65     Man3(2).
66     Man4(3).
67     Man5(4).
68
69 % Relation directlyright(x, y) that affirms that "man y" is directly to the right of
```

Black Friday (page 2 of 5)

```
"man x"
70    directlyright(m1, m2).
71    directlyright(m2, m3).
72    directlyright(m3, m4).
73    directlyright(m4, m5).
74
75    -directlyright(m1, m1).
76    -directlyright(m2, m2).
77    -directlyright(m3, m3).
78    -directlyright(m4, m4).
79    -directlyright(m5, m5).
80    -directlyright(m1, m3).
81    -directlyright(m2, m4).
82    -directlyright(m3, m5).
83    -directlyright(m3, m1).
84    -directlyright(m4, m2).
85    -directlyright(m5, m3).
86    -directlyright(m2, m1).
87    -directlyright(m3, m2).
88    -directlyright(m4, m3).
89    -directlyright(m5, m4).
90    -directlyright(m1, m4).
91    -directlyright(m2, m5).
92    -directlyright(m4, m1).
93    -directlyright(m5, m2).
94    -directlyright(m1, m5).
95    -directlyright(m5, m1).
96
97 % Relation nextto(x, y) that affirms that "man y" is next to "man x"
98    nextto(m1, m2).
99    nextto(m2, m1).
100   nextto(m2, m3).
101   nextto(m3, m2).
102   nextto(m3, m4).
103   nextto(m4, m3).
104   nextto(m4, m5).
105   nextto(m5, m4).
106
107  -nextto(m1, m1).
108  -nextto(m2, m2).
109  -nextto(m3, m3).
110  -nextto(m4, m4).
111  -nextto(m5, m5).
112  -nextto(m1, m3).
113  -nextto(m2, m4).
114  -nextto(m3, m5).
115  -nextto(m3, m1).
116  -nextto(m4, m2).
117  -nextto(m5, m3).
118  -nextto(m1, m4).
119  -nextto(m2, m5).
120  -nextto(m4, m1).
121  -nextto(m5, m2).
122  -nextto(m1, m5).
123  -nextto(m5, m1).
124
125 % Relation extremity(x) that affirms that "man x" is at one of the ends
126    extremity(m1).
127    extremity(m5).
128
129    -extremity(m2).
130    -extremity(m3).
131    -extremity(m4).
132
133 % Relation inbetween(x, y) that affirms that there is a man somewhere in between "man
134 x" and "man y", exactly in that order
135    inbetween(m1, m3).
136    inbetween(m2, m4).
137    inbetween(m3, m5).
```

Black Friday (page 3 of 5)

```
137     inbetween(m1, m4).
138     inbetween(m2, m5).
139     inbetween(m1, m5).
140
141     -inbetween(m1, m2).
142     -inbetween(m2, m1).
143     -inbetween(m2, m3).
144     -inbetween(m3, m2).
145     -inbetween(m3, m4).
146     -inbetween(m4, m3).
147     -inbetween(m4, m5).
148     -inbetween(m5, m4).
149     -inbetween(m1, m1).
150     -inbetween(m2, m2).
151     -inbetween(m3, m3).
152     -inbetween(m4, m4).
153     -inbetween(m5, m5).
154     -inbetween(m3, m1).
155     -inbetween(m4, m2).
156     -inbetween(m5, m3).
157     -inbetween(m4, m1).
158     -inbetween(m5, m2).
159     -inbetween(m5, m1).
160
161 % Relation right(x, y) that affirms that "man y" is somewhere to the right of "man x"
162     right(m1, m3).
163     right(m2, m4).
164     right(m3, m5).
165     right(m1, m4).
166     right(m2, m5).
167     right(m1, m5).
168     right(m1, m2).
169     right(m4, m5).
170     right(m2, m3).
171     right(m3, m4).
172
173     -right(m2, m1).
174     -right(m3, m2).
175     -right(m4, m3).
176     -right(m5, m4).
177     -right(m1, m1).
178     -right(m2, m2).
179     -right(m3, m3).
180     -right(m4, m4).
181     -right(m5, m5).
182     -right(m3, m1).
183     -right(m4, m2).
184     -right(m5, m3).
185     -right(m4, m1).
186     -right(m5, m2).
187     -right(m5, m1).
188
189 % Each man has different characteristics
190     Man1(x) | Man2(x) | Man3(x) | Man4(x) | Man5(x).
191     Dustin(x) | Eugene(x) | Hank(x) | Keith(x) | Sean(x).
192     Black(x) | Blue(x) | Green(x) | Red(x) | White(x).
193     BeardTrimmer(x) | GameConsole(x) | Laptop(x) | Smartphone(x) | TV(x).
194     FortyPercent(x) | FiftyPercent(x) | SixtyPercent(x) | SeventyPercent(x) |
195     EightyPercent(x).
196     Apple(x) | Cranberry(x) | Grape(x) | Lemon(x) | Orange(x).
197     TwentyFive(x) | Thirty(x) | ThirtyFive(x) | Forty(x) | FortyFive(x).
198
199 % Each property applies at most to one man
200     Dustin(x) & Dustin(y) -> -differentFrom(x, y).
201     Eugene(x) & Eugene(y) -> -differentFrom(x, y).
202     Hank(x) & Hank(y) -> -differentFrom(x, y).
203     Keith(x) & Keith(y) -> -differentFrom(x, y).
204     Sean(x) & Sean(y) -> -differentFrom(x, y).
```

Black Friday (page 4 of 5)

```
205    Black(x) & Black(y) -> -differentFrom(x, y).
206    Blue(x) & Blue(y) -> -differentFrom(x, y).
207    Green(x) & Green(y) -> -differentFrom(x, y).
208    Red(x) & Red(y) -> -differentFrom(x, y).
209    White(x) & White(y) -> -differentFrom(x, y).
210
211    BeardTrimmer(x) & BeardTrimmer(y) -> -differentFrom(x, y).
212    GameConsole(x) & GameConsole(y) -> -differentFrom(x, y).
213    Laptop(x) & Laptop(y) -> -differentFrom(x, y).
214    Smartphone(x) & Smartphone(y) -> -differentFrom(x, y).
215    TV(x) & TV(y) -> -differentFrom(x, y).
216
217    FortyPercent(x) & FortyPercent(y) -> -differentFrom(x, y).
218    FiftyPercent(x) & FiftyPercent(y) -> -differentFrom(x, y).
219    SixtyPercent(x) & SixtyPercent(y) -> -differentFrom(x, y).
220    SeventyPercent(x) & SeventyPercent(y) -> -differentFrom(x, y).
221    EightyPercent(x) & EightyPercent(y) -> -differentFrom(x, y).
222
223    Apple(x) & Apple(y) -> -differentFrom(x, y).
224    Cranberry(x) & Cranberry(y) -> -differentFrom(x, y).
225    Grape(x) & Grape(y) -> -differentFrom(x, y).
226    Lemon(x) & Lemon(y) -> -differentFrom(x, y).
227    Orange(x) & Orange(y) -> -differentFrom(x, y).
228
229    TwentyFive(x) & TwentyFive(y) -> -differentFrom(x, y).
230    Thirty(x) & Thirty(y) -> -differentFrom(x, y).
231    ThirtyFive(x) & ThirtyFive(y) -> -differentFrom(x, y).
232    Forty(x) & Forty(y) -> -differentFrom(x, y).
233    FortyFive(x) & FortyFive(y) -> -differentFrom(x, y).
234
235 %Clues
236
237    % 1. The man drinking the Orange juice is exactly to the right of the man who got
238    % the 70% discount.
239    Orange(x) & SeventyPercent(y) -> directlyright(y, x).
240
241    % 2. Keith is 45 years old.
242    Keith(x) -> FortyFive(x).
243
244    % 3. The man who bought the TV is exactly to the left of the man wearing the Red
245    % shirt.
246    TV(x) & Red(y) -> directlyright(x, y).
247
248    % 4. At the third position is the man who got the 50% discount.
249    Man3(x) -> FiftyPercent(x).
250
251    % 5. Keith is next to the man wearing the White shirt.
252    Keith(x) & White(y) -> nextto(y, x).
253
254    % 6. The 25-year-old man is somewhere between the 35-year-old man and the
255    % 40-year-old man, in that order.
256    ThirtyFive(x) & Forty(y) -> inbetween(x, y).
257    TwentyFive(x) & ThirtyFive(y) -> right(y, x).
258    TwentyFive(x) & Forty(y) -> right(x, y).
259
260    % 7. The man drinking Apple juice bought the Smartphone.
261    Apple(x) -> Smartphone(x).
262
263    % 8. The 30-year-old man is exactly to the left of the man that bought the Beard
264    % trimer.
265    Thirty(x) & BeardTrimmer(y) -> directlyright(x, y).
266
267    % 9. Sean is the youngest.
268    Sean(x) -> TwentyFive(x).
```

Black Friday (page 5 of 5)

```
269      % 11. Keith is next to the 35-year-old man.  
270      Keith(x) & ThirtyFive(y) -> nextto(y, x).  
271  
272      % 12. Eugene is 40 years old.  
273      Eugene(x) -> Forty(x).  
274  
275      % 13. Sean is wearing the Black shirt.  
276      Sean(x) -> Black(x).  
277  
278      % 14. At the fourth position is the man who got the biggest discount.  
279      Man4(x) -> EightyPercent(x).  
280  
281      % 15. Dustin got 60% off.  
282      Dustin(x) -> SixtyPercent(x).  
283  
284      % 16. The man drinking the Lemon juice is exactly to the right of the man drinking  
the Grape juice.  
285      Lemon(x) & Grape(y) -> directlyright(y, x).  
286  
287      % 17. Keith bought a Game console.  
288      Keith(x) -> GameConsole(x).  
289  
290      % 18. The man who got the 80% discount is exactly to the left of the man who is  
wearing the Blue shirt.  
291      EightyPercent(x) & Blue(y) -> directlyright(x, y).  
292  
293      % 19. The man drinking Grape juice bought the Beard trimmer.  
294      Grape(x) -> BeardTrimmer(x).  
295  
296      % 20. The man wearing the Black shirt is somewhere to the right of Keith.  
297      Black(x) & Keith(y) -> right(y, x).  
298  
299      % 21. The man that bought the Smartphone is next to the man wearing the Black shirt.  
300      Smartphone(x) & Black(y) -> nextto(y, x).  
301  
302      end_of_list.  
303
```

Home Remodeling (page 1 of 4)

```
1  if(Mace4).
2  assign(domain_size, 5).
3  end_if.
4
5  if(Prover9).
6  formulas(goals).
7
8  Couple1(d1) -> Stephen(d1).
9
10 end_of_list.
11 end_if.
12
13 % Avoid using sentence names starting with u, v, w, x, y, z
14
15 % Couple's order in row
16     % Couple1: first man
17     % Couple2: second man
18     % Couple3: third man
19     % Couple4: fourth man
20     % Couple5: fifth man
21
22 % Carts' colors
23     % Black, Blue, Green, Red, White
24
25 % Husbands
26     % Benny, Darrel, Myles, Stephen, Vince
27
28 % Wifes
29     % Amelia, Jeanne, Kassie, Sylvia, Toni
30
31 % Room
32     % Bathroom, Bedroom, DiningRoom, Kitchen, LivingRoom
33
34 % Budget
35     % ThreeThousand, SixThousand, NineThousand, TwelveThousand, FifteenThousand
36
37 % Inspiration
38     % Exhibition, FriendsHouse, Magazine, TVshow, Website
39
40 formulas(assumptions).
41
42 % The 5 couples differ
43     differentFrom(d1, d2).
44     differentFrom(d1, d3).
45     differentFrom(d1, d4).
46     differentFrom(d1, d5).
47     differentFrom(d2, d3).
48     differentFrom(d2, d4).
49     differentFrom(d2, d5).
50     differentFrom(d3, d4).
51     differentFrom(d3, d5).
52     differentFrom(d4, d5).
53
54 % The "differentFrom" relation is symmetrical
55     differentFrom(x, y) -> differentFrom(y, x).
56
57     d1 = 0.
58     d2 = 1.
59     d3 = 2.
60     d4 = 3.
61     d5 = 4.
62
63     Couple1(0).
64     Couple2(1).
65     Couple3(2).
66     Couple4(3).
67     Couple5(4).
68
69 % Relation directlyright(x, y) that affirms that "couple y" is directly to the right of
```

Home Remodeling (page 2 of 4)

```
70      "couple x"
71          directlyright(d1, d2).
72          directlyright(d2, d3).
73          directlyright(d3, d4).
74          directlyright(d4, d5).
75
76          -directlyright(d1, d1).
77          -directlyright(d2, d2).
78          -directlyright(d3, d3).
79          -directlyright(d4, d4).
80          -directlyright(d5, d5).
81          -directlyright(d1, d3).
82          -directlyright(d2, d4).
83          -directlyright(d3, d5).
84          -directlyright(d3, d1).
85          -directlyright(d4, d2).
86          -directlyright(d5, d3).
87          -directlyright(d2, d1).
88          -directlyright(d3, d2).
89          -directlyright(d4, d3).
90          -directlyright(d5, d4).
91          -directlyright(d1, d4).
92          -directlyright(d2, d5).
93          -directlyright(d4, d1).
94          -directlyright(d5, d2).
95          -directlyright(d1, d5).
96          -directlyright(d5, d1).
97
98 % Relation nextto(x, y) that affirms that "couple y" is next to "couple x"
99      directlyright(x, y) | directlyright(y, x) <-> nextto(x, y).
100
101 % Relation extremity(x) that affirms that "couple x" is at One of the ends.
102      extremity(d1).
103      extremity(d5).
104
105      -extremity(d2).
106      -extremity(d3).
107      -extremity(d4).
108
109 % Relation right(x, y) that affirms that "couple y" is somewhere to the right of
110 "couple x"
111      right(d1, d3).
112      right(d2, d4).
113      right(d3, d5).
114      right(d1, d4).
115      right(d2, d5).
116      right(d1, d2).
117      right(d4, d5).
118      right(d2, d3).
119      right(d3, d4).
120
121      -right(d2, d1).
122      -right(d3, d2).
123      -right(d4, d3).
124      -right(d5, d4).
125      -right(d1, d1).
126      -right(d2, d2).
127      -right(d3, d3).
128      -right(d4, d4).
129      -right(d5, d5).
130      -right(d3, d1).
131      -right(d4, d2).
132      -right(d5, d3).
133      -right(d4, d1).
134      -right(d5, d2).
135      -right(d5, d1).
136
137 % % Each couple has different characteristics
```

Home Remodeling (page 3 of 4)

```
137 Couple1(x) | Couple2(x) | Couple3(x) | Couple4(x) | Couple5(x) .
138 Benny(x) | Darrel(x) | Myles(x) | Stephen(x) | Vince(x) .
139 Black(x) | Blue(x) | Green(x) | Red(x) | White(x) .
140 Amelia(x) | Jeanne(x) | Kassie(x) | Sylvia(x) | Toni(x) .
141 ThreeThousand(x) | SixThousand(x) | NineThousand(x) | TwelveThousand(x) |
142 FifteenThousand(x) .
143 Exhibition(x) | FriendsHouse(x) | Magazine(x) | TVshow(x) | Website(x) .
144 Bathroom(x) | Bedroom(x) | DiningRoom(x) | Kitchen(x) | LivingRoom(x) .
145 % Each property applies at most to one couple
146 Benny(x) & Benny(y) -> -differentFrom(x, y) .
147 Darrel(x) & Darrel(y) -> -differentFrom(x, y) .
148 Myles(x) & Myles(y) -> -differentFrom(x, y) .
149 Stephen(x) & Stephen(y) -> -differentFrom(x, y) .
150 Vince(x) & Vince(y) -> -differentFrom(x, y) .
151
152 Black(x) & Black(y) -> -differentFrom(x, y) .
153 Blue(x) & Blue(y) -> -differentFrom(x, y) .
154 Green(x) & Green(y) -> -differentFrom(x, y) .
155 Red(x) & Red(y) -> -differentFrom(x, y) .
156 White(x) & White(y) -> -differentFrom(x, y) .
157
158 Amelia(x) & Amelia(y) -> -differentFrom(x, y) .
159 Jeanne(x) & Jeanne(y) -> -differentFrom(x, y) .
160 Kassie(x) & Kassie(y) -> -differentFrom(x, y) .
161 Sylvia(x) & Sylvia(y) -> -differentFrom(x, y) .
162 Toni(x) & Toni(y) -> -differentFrom(x, y) .
163
164 ThreeThousand(x) & ThreeThousand(y) -> -differentFrom(x, y) .
165 SixThousand(x) & SixThousand(y) -> -differentFrom(x, y) .
166 NineThousand(x) & NineThousand(y) -> -differentFrom(x, y) .
167 TwelveThousand(x) & TwelveThousand(y) -> -differentFrom(x, y) .
168 FifteenThousand(x) & FifteenThousand(y) -> -differentFrom(x, y) .
169
170 Exhibition(x) & Exhibition(y) -> -differentFrom(x, y) .
171 FriendsHouse(x) & FriendsHouse(y) -> -differentFrom(x, y) .
172 Magazine(x) & Magazine(y) -> -differentFrom(x, y) .
173 TVshow(x) & TVshow(y) -> -differentFrom(x, y) .
174 Website(x) & Website(y) -> -differentFrom(x, y) .
175
176 Bathroom(x) & Bathroom(y) -> -differentFrom(x, y) .
177 Bedroom(x) & Bedroom(y) -> -differentFrom(x, y) .
178 DiningRoom(x) & DiningRoom(y) -> -differentFrom(x, y) .
179 Kitchen(x) & Kitchen(y) -> -differentFrom(x, y) .
180 LivingRoom(x) & LivingRoom(y) -> -differentFrom(x, y) .
181
182 %Clues
183
184 % 1. The couple guiding the Black cart is somewhere between the couple inspired by
185 a Website and the couple guiding the Red cart, in that order.
186 Website(x) & Black(y) & Red(z) -> right(x, y) & right(y, z) .
187
188 % 2. Amelia has a $ 15,000 budget.
189 Amelia(x) -> FifteenThousand(x) .
190
191 % 3. The partners who will remodel their Bedroom is next to the couple who will
192 remodel their Living room.
193 Bedroom(x) & LivingRoom(y) -> nextto(y, x) .
194
195 % 4. Kassie is at one of the ends.
196 Kassie(x) -> extremity(x) .
197
198 % 5. Benny is next to the partners steering the Black shopping cart.
199 Benny(x) & Black(y) -> nextto(y, x) .
200
201 % 6. At the first position is the couple who has the biggest budget.
202 Couple1(x) -> FifteenThousand(x) .
203
204 % 7. The couple inspired by a Magazine is exactly to the right of the couple that
```

Home Remodeling (page 4 of 4)

```
will remodel their Living room.  
203 Magazine(x) & LivingRoom(y) -> directlyright(y, x).  
204  
205 % 8. Jeanne is immediately before the partners that are going to remodel their  
Bathroom.  
206 Jeanne(x) & Bathroom(y) -> directlyright(x, y).  
207  
208 % 9. Toni was inspired by a interior design Exhibition.  
209 Toni(x) -> Exhibition(x).  
210  
211 % 10. The couple guiding the Green cart is going to remodel their Bathroom.  
212 Green(x) -> Bathroom(x).  
213  
214 % 11. At the fifth position is the couple who got an inspiration from a Friend's  
house.  
215 Couple5(x) -> FriendsHouse(x).  
216  
217 % 12. Benny is next to Sylvia and her husband.  
218 Benny(x) & Sylvia(y) -> nextto(y, x).  
219  
220 % 13. Vince is exactly to the right of the couple that will remodel their Kitchen.  
221 Vince(x) & Kitchen(y) -> directlyright(y, x).  
222  
223 % 14. Myles and his wife will remodel their Bathroom.  
224 Myles(x) -> Bathroom(x).  
225  
226 % 15. The couple inspired by a TV show is somewhere between the couple that has the  
smallest budget and the couple that has a $ 9,000 budget, in that order.  
227 ThreeThousand(x) & TVshow(y) & NineThousand(z) -> right(x, y) & right(y, z).  
228  
229 % 16. The partners steering the the Red cart is somewhere to the left of the couple  
steering the White cart.  
230 Red(x) & White(y) -> right(x, y).  
231  
232 % 17. The couple that will remodel their Kitchen is immediately before the couple  
that will remodel their Bedroom.  
233 Kitchen(x) & Bedroom(y) -> directlyright(x, y).  
234  
235 % 18. The partners that have a $ 12,000 budget is somewhere to the right of the  
partners guiding the Green cart.  
236 TwelveThousand(x) & Green(y) -> right(y, x).  
237  
238 % 19. Darrel is exactly to the right of the couple who has a $ 9,000 budget.  
239 Darrel(x) & NineThousand(y) -> directlyright(y, x).  
240  
241  
242 end_of_list.  
243
```

Intelligent Systems Group

