



Artificial Intelligence

Deductive Reasoning

Task Report 5

Harun GÜLEÇ

Białystok, 2017

Task 1

At this task our main object is understanding deductive reasoning and deduce some information on data set. Firstly, I loaded data from file to program. Then, I defined rules for father, mother, husband, wife, son, daughter, brother and sister. By using these rules, I defined complex rules for uncle, aunt, grandfather and grandmother. Rules are shown below.

```
father(A) <- parent(A,B) & person(A,male) .
mother(A) <- parent(A,B) & person(A,female) .
husband(A) <- married(A,B) & person(A,male) .
husband(A) <- married(B,A) & person(A,male) .
wife(A) <- married(A,B) & person(A,female) .
wife(A) <- married(B,A) & person(A,female) .
son(A) <- parent(B,A) & person(A,male) .
daughter(A) <- parent(B,A) & person(A,female) .
brother(A,B) <- parent(C,A) & parent(C,B) & A \= B & person(A,male) .
sister(A,B) <- parent(C,A) & parent(C,B) & A \= B & person(A,female) .
uncle(A) <- brother(A,C) & parent(C,B) .
aunt(A) <- sister(A,C) & parent(C,B) .
grandfather(A,B) <- parent(C,B) & parent(A,C) & person(A,male) .
grandmother(A,B) <- parent(C,B) & parent(A,C) & person(A,female) .
```

After that , I chose 8 person and check some sentences.

N1 → person(beatrice,female).

N2 → person(harry,male).

N3 → person(gail,female).

N4 → person(terry,male).

N5 → person(terri,female).

N6 → person(melvin,male).

N7 → person(maria,female).

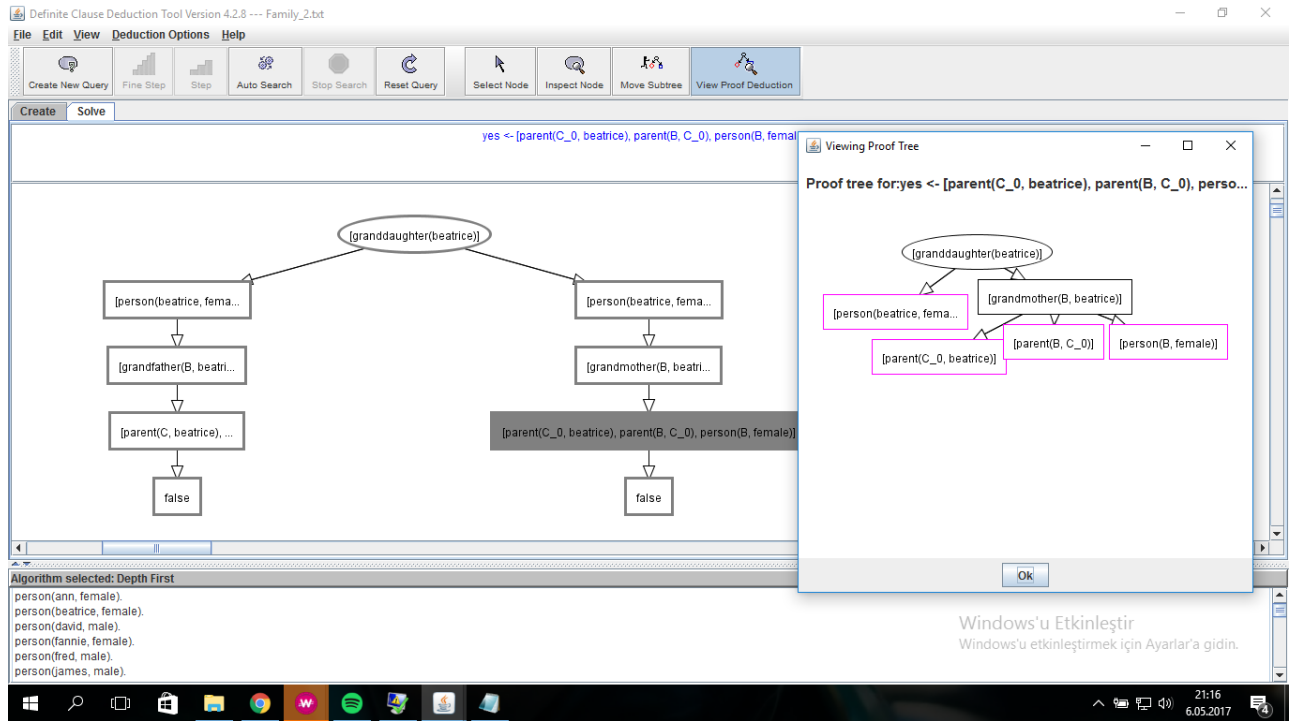
N8 → person(ann,female).

- *N1 is granddaughter.*

New Rule:

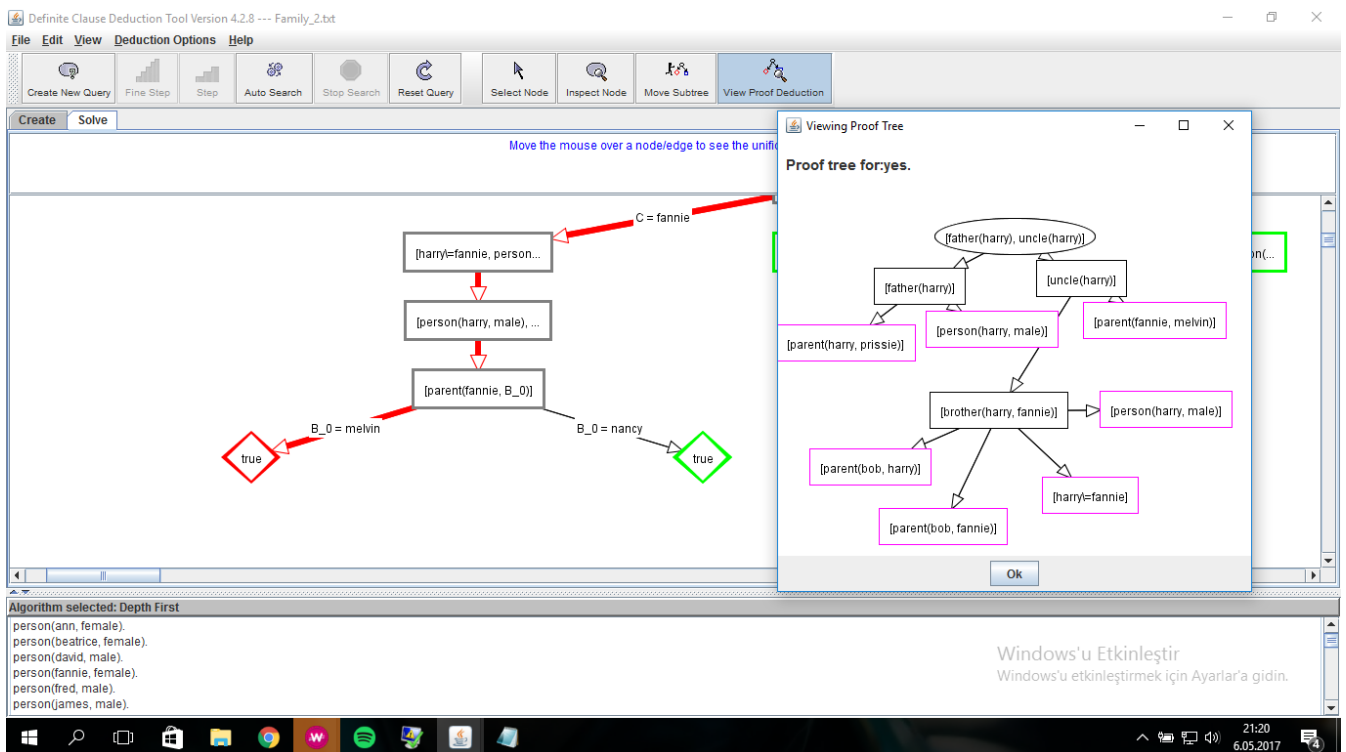
granddaughter(A) <- person(A, female) & grandfather(B, A).

granddaughter(A) <- person(A, female) & grandmother(B, A).



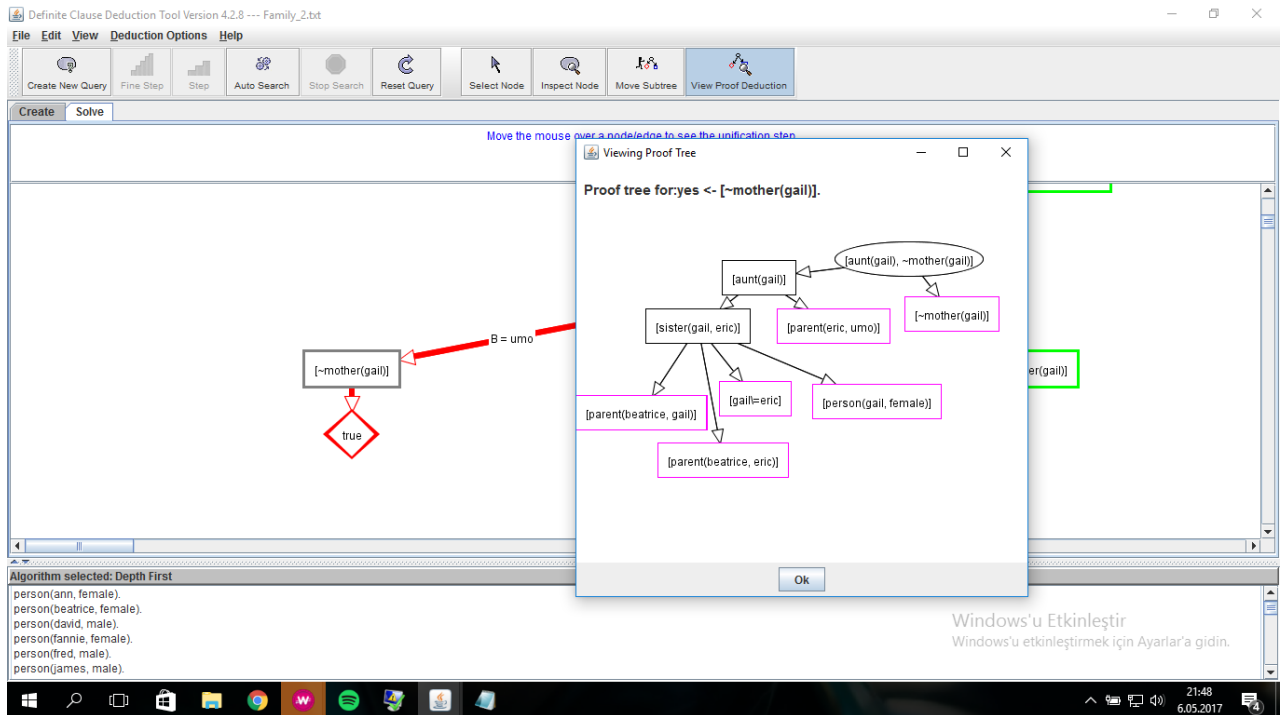
As we can see on picture, Beatrice is not granddaughter.

- *N2 is both a father and an uncle.*



According to result, Harry is both father and uncle.

- *N3 is an aunt, but she is not a mother.*



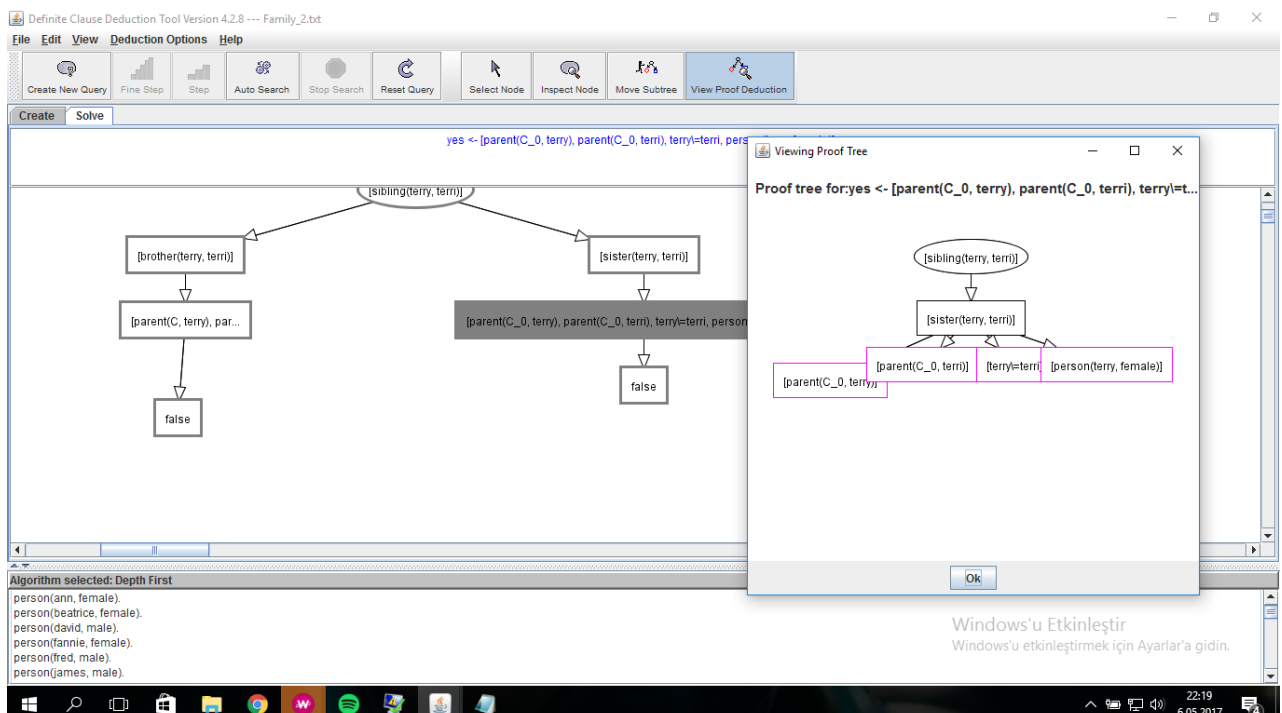
It is true because brother of gail (eric) has a child and she is not a mother.

- *N4 i N5 are siblings.*

New Rule:

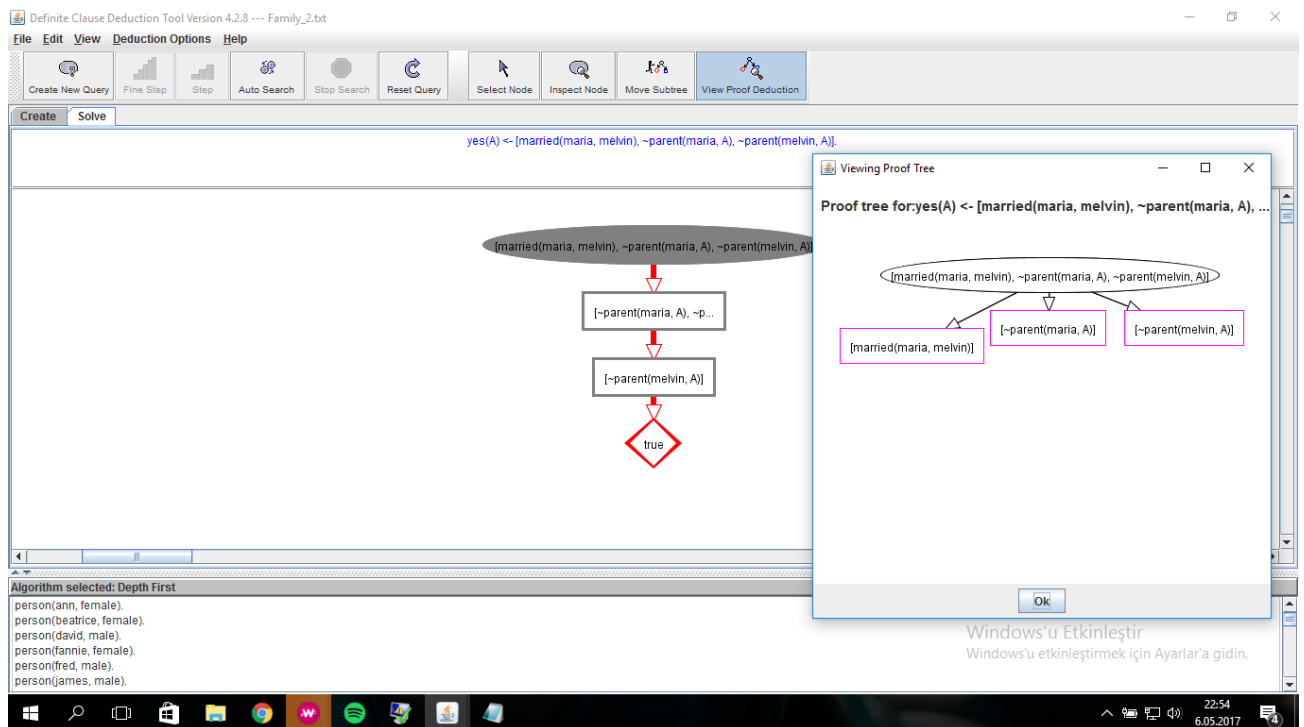
$sibling(A,B) \leftarrow -brother(A,B).$

$sibling(A,B) \leftarrow -sister(A,B).$



It is false because terri and terry are not brother or sister.

- *N6 i N7 are married, but they do not have children.*



It is true clearly because Maria and Melvin are married but they don't have a child.

- *N8 is married or her/his sister or brother is married.*

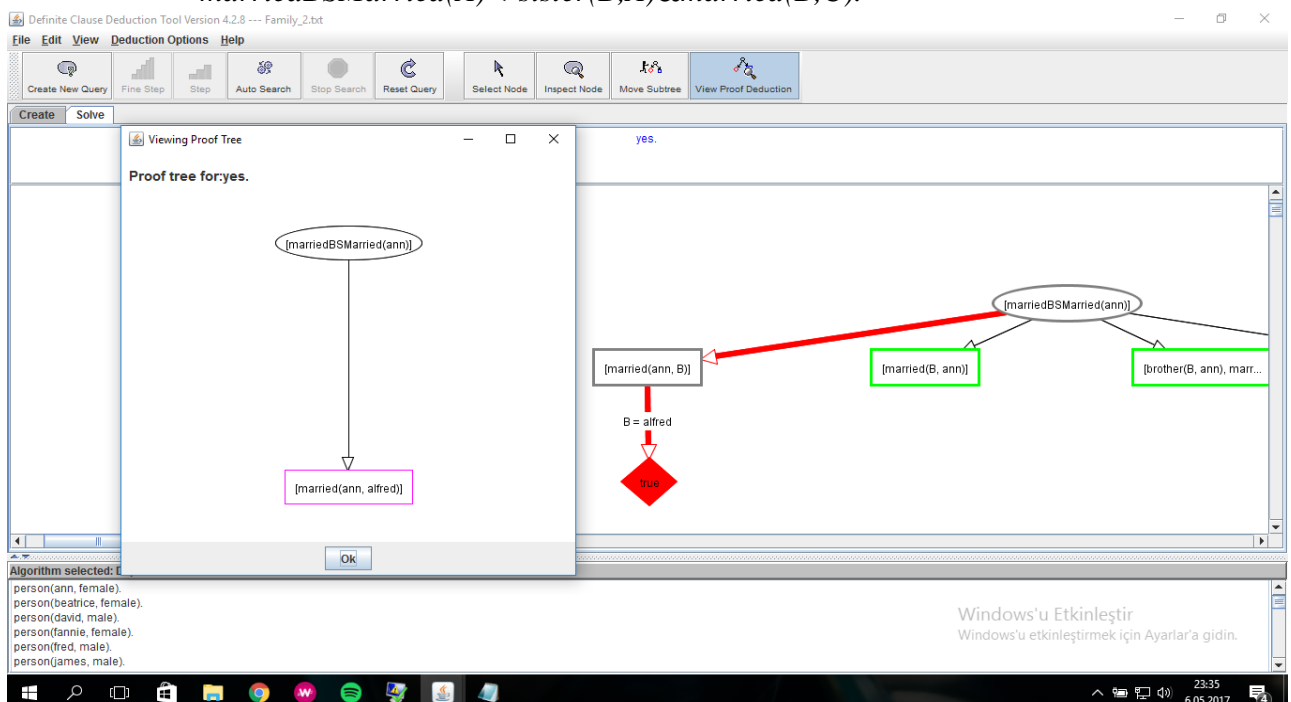
New Rule:

marriedBSMarried(A) <- married(A, B).

marriedBSMarried(A) <- married(B, A).

marriedBSMarried(A) <- brother(B, A) & married(C, B).

marriedBSMarried(A) <- sister(B, A) & married(B, C).



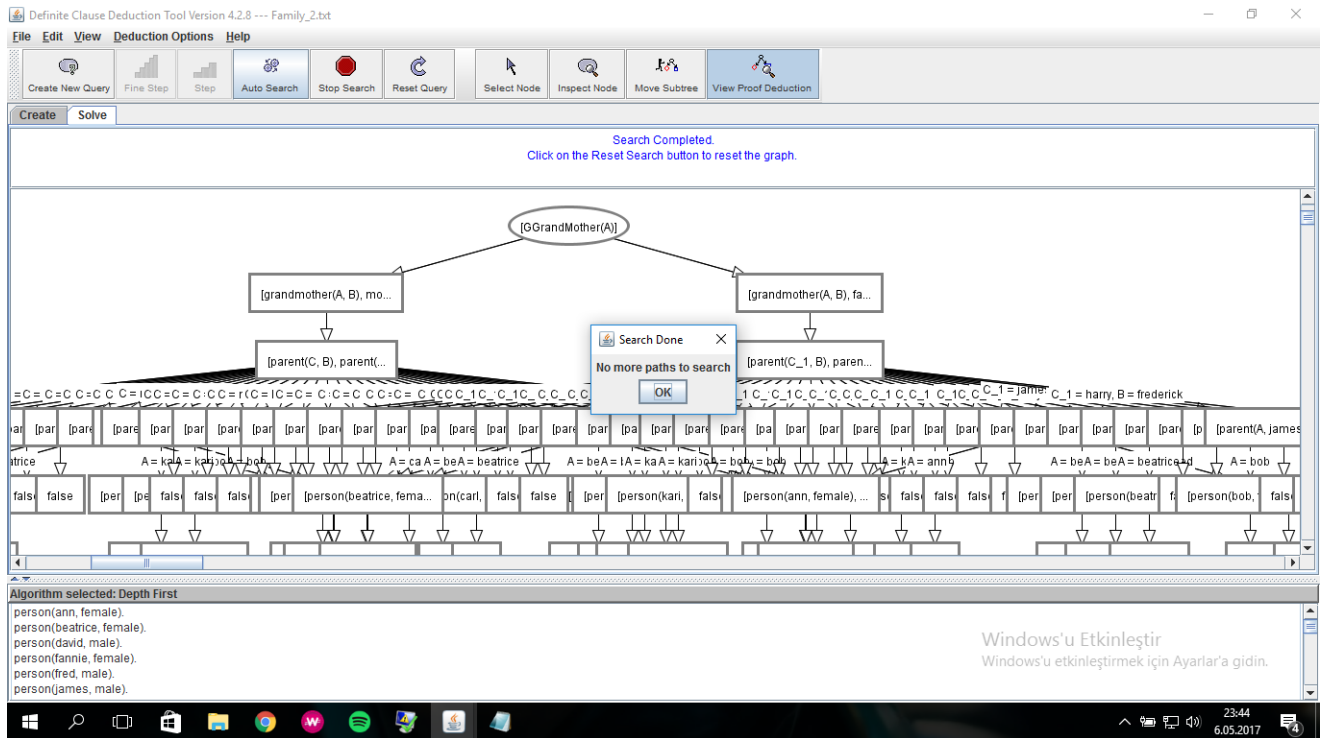
True, because Ann is married with Alfred.

- *There exists a person who is a grand-grandmother (a woman who is a mother of somebody's grandmother/grandfather).*

New Rule:

$GGrandMother(A) \leftarrow \text{grandmother}(A, B) \& \text{mother}(B).$

$GGrandMother(A) \leftarrow \text{grandmother}(A, B) \& \text{father}(B).$

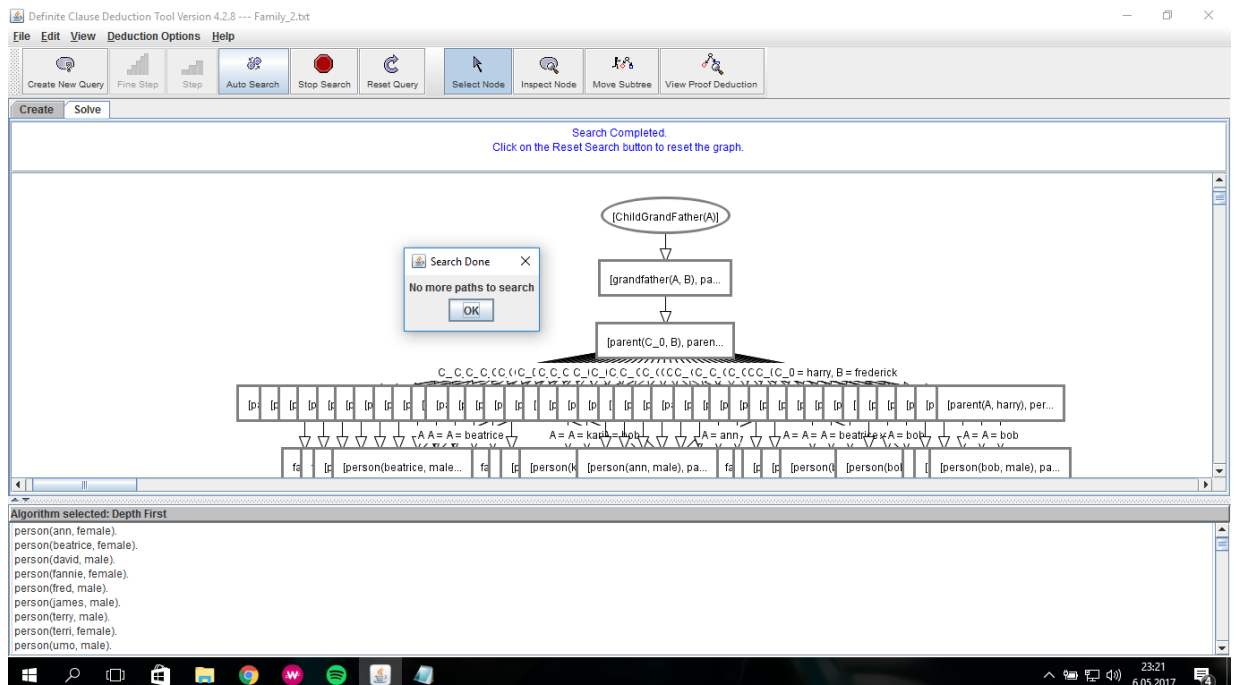


Nobody matched this condition.

- *There exists an only child who is a grandfather.*

New Rule:

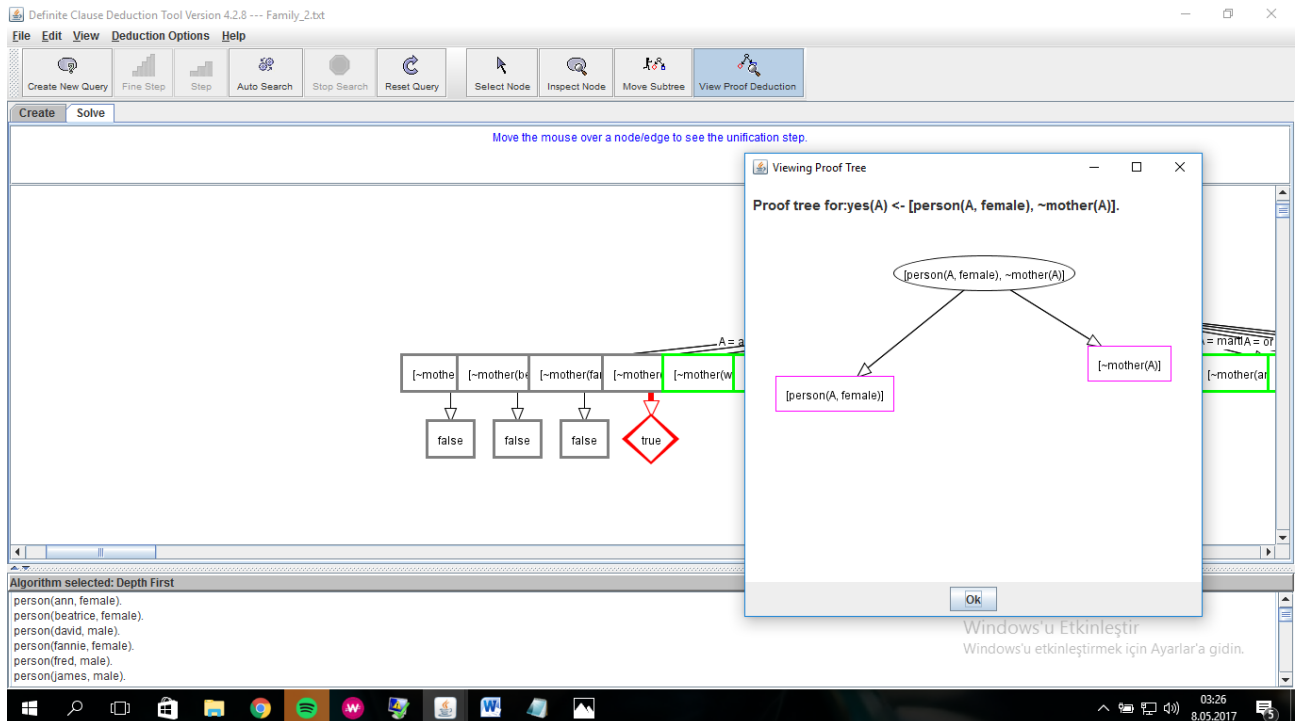
$ChildGrandFather(A) \leftarrow \text{grandfather}(A, B) \& \text{parent}(C, A).$



Nobody matched this condition.

- *All women are mothers.*

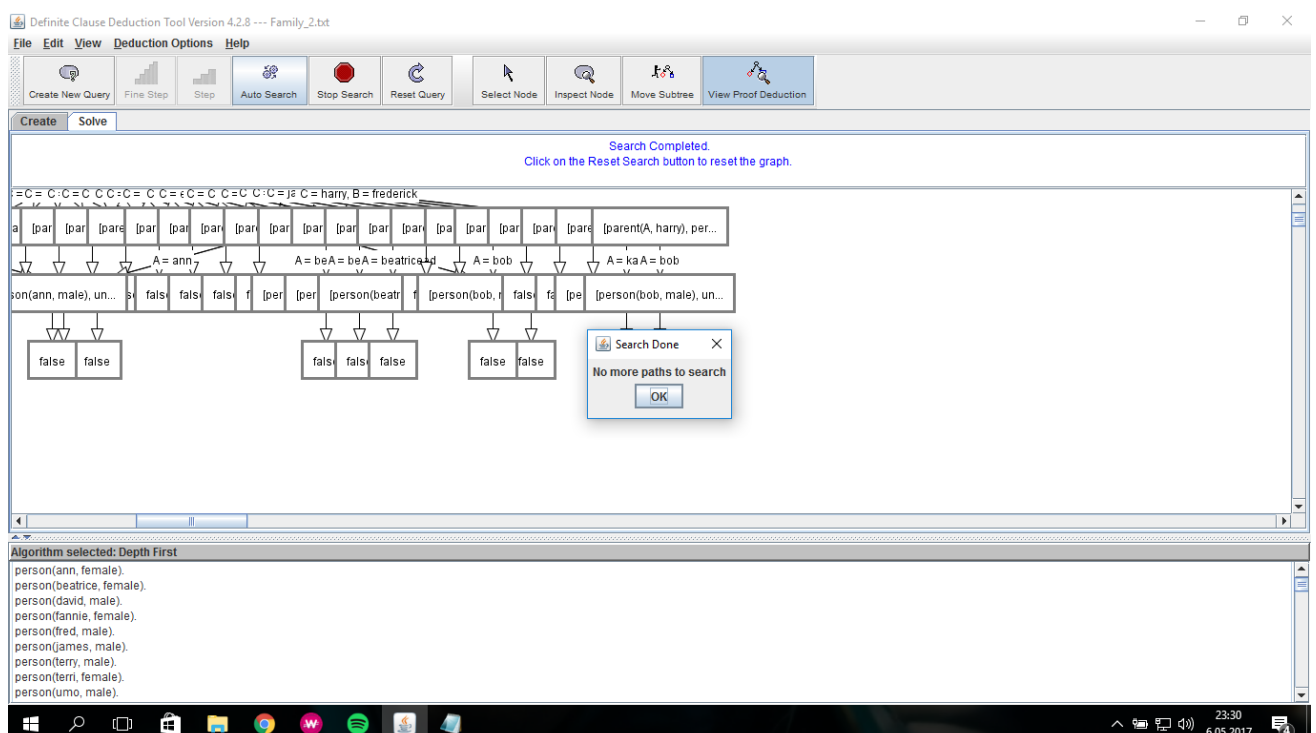
If we find one exception for this sentence we can say it is not true.



As we can see, Terri is a woman but she is not mother. So our sentence is not true.

- *Nobody is all three: a grandfather, a father, and an uncle.*

If we find one person against to this sentence. We can say it is not true.



We can find any example against to this sentence so it is true.

Task 2

At this task we have to propose a knowledge base consisting of at least three relations.

My proposition is about company. Person specifies employees and their department. Department specifies departments and salaries which taken by employees in this department. Manager specifies the managers of company.

```
person(harun, security) .
person(zahid, ir) .
person(mert, ir) .
person(ozan, marketing) .
person(hatice, security) .
person(sumeyye, marketing) .
person(gizem, mobile) .
person(ali, finance) .
department(security, 3750) .
department(ir, 3000) .
department(marketing, 2000) .
department(mobile, -1) .
department(finance, 2750) .
manager(harun) .
manager(mert) .
manager(ozan) .
manager(gizem) .
manager(ali) .
```

Then, we have to construct at least five rules.

```
managerOfDept(A, B) <- person(A, B) & manager(A) .
salaryOfPerson(A, B) <- person(B, C) & department(C, A) .
higherthan3000(A) <- person(A, B) & department(B, C) & C > 3000 .
projectbase(A) <- person(A, B) & department(B, C) & C < 0 .
managerOfPerson(A, B) <- manager(A) & person(A, C) & person(B, C) .
```

managerOfDept(A, B) → Find the Manager of Department.

salaryOfPerson(A, B) → Find the Salary of Person.

higherthan3000(A) → Find a person whose salary higher than 3000.

projectbase(A) → Find the person who works project based.

managerOfPerson(A, B) → Find the Manager of Person.

Facts that satisfying the rules are shown below.

Definite Clause Deduction Tool Version 4.2.8 --- proposedData_company.txt

File Edit View Deduction Options Help

Create New Query Fine Step Step Auto Search Stop Search Reset Query Select Node Inspect Node Move Subtree View Proof Deduction

Create Solve

Goal node reached!
Click again to find next goal node or Reset Query to start query over.

Goal Node Reached

yes(A) - [managerOfDept(A, security)].
yes(A) - [person(A, security), manager(A)].
yes(harun) - [manager(harun)].
yes(harun).

OK

manager(harun)

A = harun

[person(A, security), ...]

A = hatice

[manager(hatice)]

true

Algorithm selected: Depth First

person(harun, security).
person(zahid, ir).
person(mert, ir).
person(ozan, marketing).
person(hatice, security).
person(sumeyye, marketing).
person(gizem, mobile).
person(ali, finance).
department(security, 3750).

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

Definite Clause Deduction Tool Version 4.2.8 --- proposedData_company.txt

File Edit View Deduction Options Help

Create New Query Fine Step Step Auto Search Stop Search Reset Query Select Node Inspect Node Move Subtree View Proof Deduction

Create Solve

Goal node reached!
Click again to find next goal node or Reset Query to start query over.

Goal Node Reached

yes(A) - [salaryOfPerson(A, harun)].
yes(A) - [person(harun, C), department(C, A)].
yes(A) - [department(security, A)].
yes(3750).

OK

[salaryOfPerson(A, harun)]

[person(harun, C), dep...]

C = security

[department(security, A)]

A = 3750

true

Algorithm selected: Depth First

person(harun, security).
person(zahid, ir).
person(mert, ir).
person(ozan, marketing).
person(hatice, security).
person(sumeyye, marketing).
person(gizem, mobile).
person(ali, finance).
department(security, 3750).

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

Definite Clause Deduction Tool Version 4.2.8 --- proposedData_company.txt

File Edit View Deduction Options Help

Create New Query Fine Step Step Auto Search Stop Search Reset Query Select Node Inspect Node Move Subtree View Proof Deduction

Create Solve

Goal node reached!
Click again to find next goal node or Reset Query to start query over.

Goal Node Reached

yes - [higherthan3000(hatice)].
yes - [person(hatice, B), department(B, C), C > 3000].
yes - [department(security, C), C > 3000].
yes - [3750 > 3000].
yes.

OK

[person(hatice, B), de...]
B = security
[department(security, ...]
C = 3750
[3750 > 3000]
true

Algorithm selected: Depth First

person(harun, security).
person(zahid, ir).
person(mert, ir).
person(ozan, marketing).
person(hatice, security).
person(sumeyye, marketing).
person(gizem, mobile).
person(ali, finance).
department(security, 3750).

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

00:52
8.05.2017

Definite Clause Deduction Tool Version 4.2.8 --- proposedData_company.txt

File Edit View Deduction Options Help

Create New Query Fine Step Step Auto Search Stop Search Reset Query Select Node Inspect Node Move Subtree View Proof Deduction

Create Solve

Goal node reached!
Click again to find next goal node or Reset Query to start query over.

Goal Node Reached

yes - [projectBased(gizem)].
yes - [person(gizem, B), department(B, C), C0].
yes - [department(mobile, C), C0].
yes - [-10].
yes.

OK

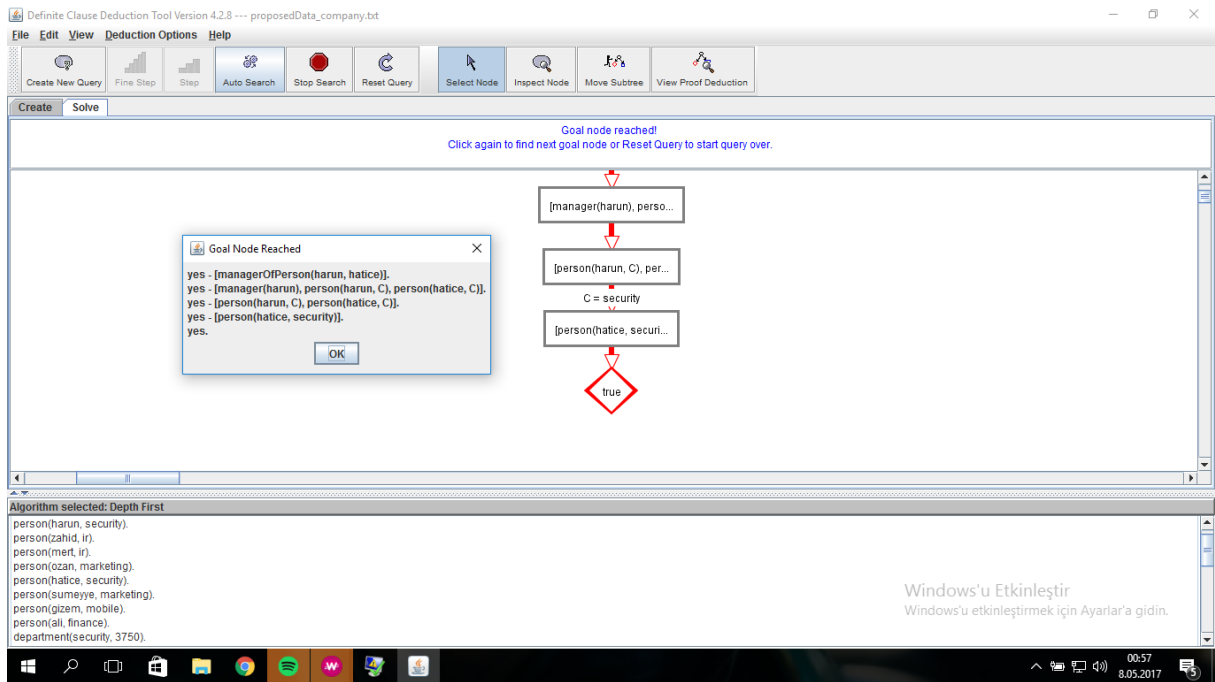
[person(gizem, B), dep...]
B = mobile
[department(mobile, C)...]
C = -1
[-1 < 0]
true

Algorithm selected: Depth First

person(harun, security).
person(zahid, ir).
person(mert, ir).
person(ozan, marketing).
person(hatice, security).
person(sumeyye, marketing).
person(gizem, mobile).
person(ali, finance).
department(security, 3750).

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

00:56
8.05.2017



After that, we have to construct new rules defining three new relations using previous rules.

```
onlyManager(A) <- manager(A) & ~managerOfPerson(A, B) .
higherthan3000Manager(A) <- higherthan3000(A) & manager(A) .
projectBasedManager(A) <- projectBased(A) & manager(A) .
```

onlyManager(A) → Find the departments that have only manager. No employee.

higherthan3000Manager(A) → Find a manager whose salary higher than 3000.

projectBasedManager(A) → Find a manager who works project based.

Facts that satisfying the rules are shown below.

Definite Clause Deduction Tool Version 4.2.8 --- proposedData_company.txt

File Edit View Deduction Options Help

Create New Query Fine Step Step Auto Search Stop Search Reset Query Select Node Inspect Node Move Subtree View Proof Deduction

Create Solve

Goal node reached!
Click again to find next goal node or Reset Query to start query over.

Goal Node Reached

yes - [onlyManager(gizem)].
yes - [manager(gizem), ~managerOfPerson(gizem, B)].
yes - [-managerOfPerson(gizem, B)].
yes.

OK

[onlyManager(gizem)]

[manager(gizem), ~mane...]

[-managerOfPerson(gize...]

true

Algorithm selected: Depth First

person(harun, security).
person(zahid, ir).
person(mert, ir).
person(ozan, marketing).
person(halice, security).
person(sumeyye, marketing).

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

00:58
8.05.2017

Definite Clause Deduction Tool Version 4.2.8 --- proposedData_company.txt

File Edit View Deduction Options Help

Create New Query Fine Step Step Auto Search Stop Search Reset Query Select Node Inspect Node Move Subtree View Proof Deduction

Create Solve

Goal node reached!
Click again to find next goal node or Reset Query to start query over.

Goal Node Reached

yes - [higherthan3000Manager(harun)].
yes - [higherthan3000(harun), manager(harun)].
yes - [person(harun, B), department(B, C), C>3000, manager(harun)].
yes - [department(security, C), C>3000, manager(harun)].
yes - [3750>3000, manager(harun)].
yes - [manager(harun)].
yes.

OK

B = security

[department(security, ...]

C = 3750

[3750>3000, manager(ha...]

[manager(harun)]

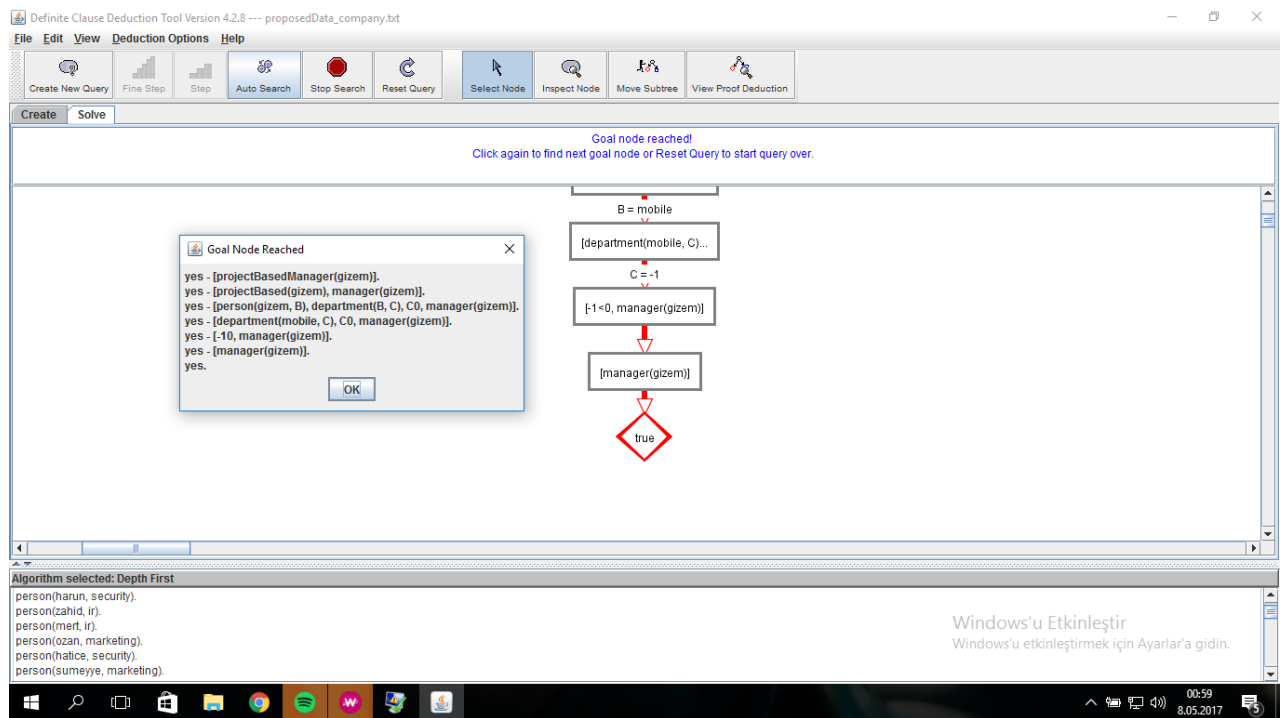
true

Algorithm selected: Depth First

person(harun, security).
person(zahid, ir).
person(mert, ir).
person(ozan, marketing).
person(halice, security).
person(sumeyye, marketing).

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

00:58
8.05.2017



My knowledge base is useful to decide who have they high salary? Who is the manager of department? Who is the manager of a person? Or which departments have only manager? Is there any project based manager?

Conclusion

Deductive reasoning is so efficient way to getting information by using relations. It can use in data mining and in artificial intelligence to understanding complex relations. For example, when we defined that we like science-fiction, and also when we defined that “The Hitchhiker’s Guide to the Galaxy” is a science-fiction book in suggestion system. It can suggest us this book.