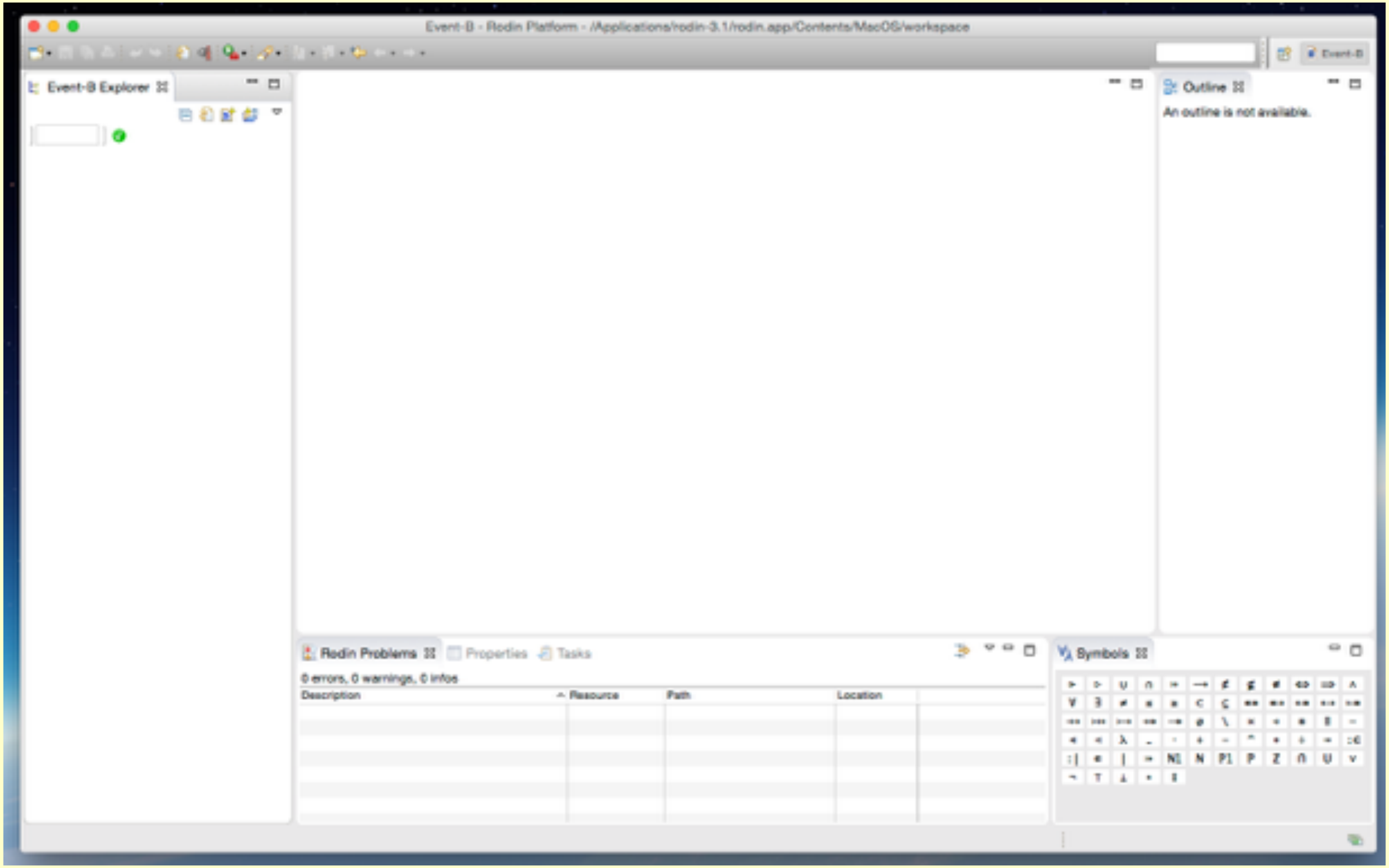
Event-B (Using RODIN): First Example

/~gibson/Teaching/CSC4504/Event-B-FirstExample.pdf

A CLEAN WORKSPACE

After installing RODIN and checking for updates you should have a clean workspace, which looks something like this:

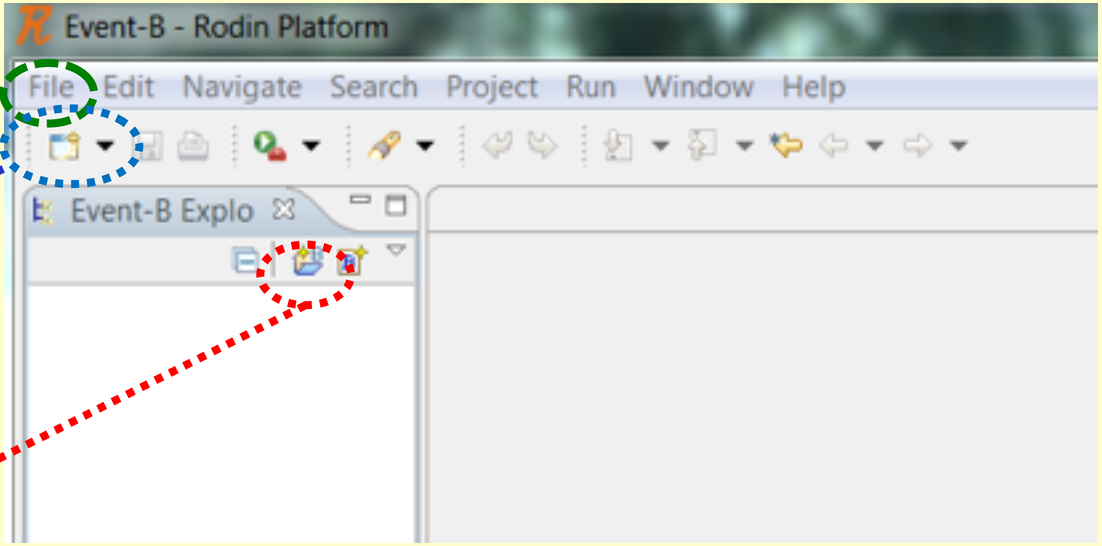


A NEW PROJECT

Now we wish to create a new project; and there are multiple ways of doing this (as there always is in RODIN – and Eclipse):

New Event-B project

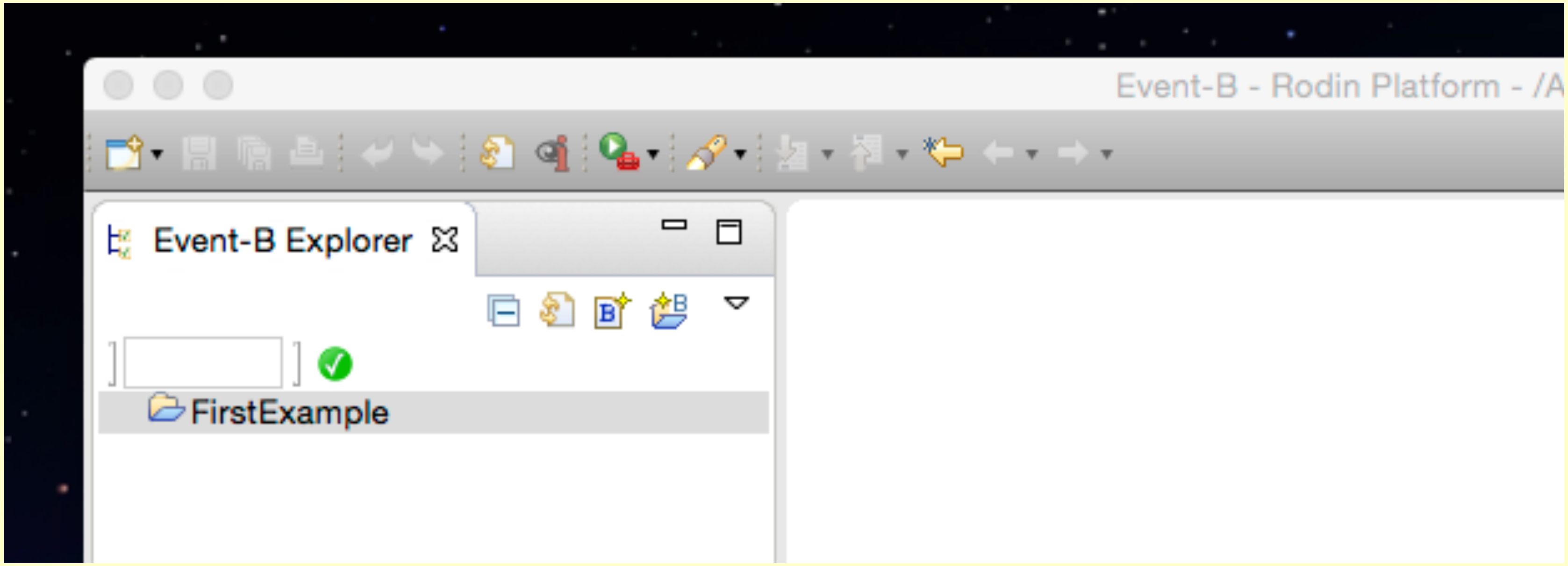
New Event-B project



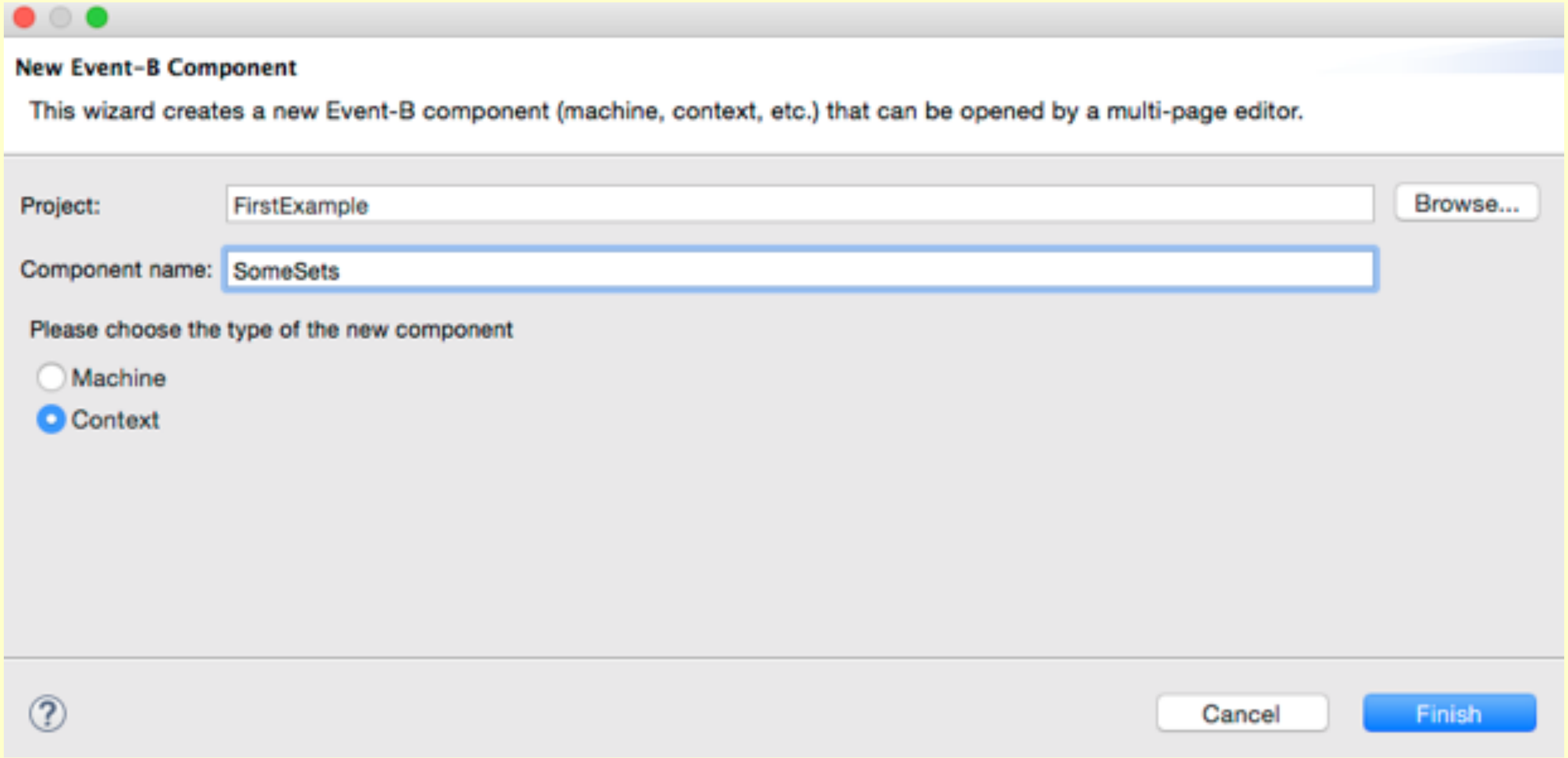
Here: using shortcut

TO DO - make a new Event-B project called “FirstExample”

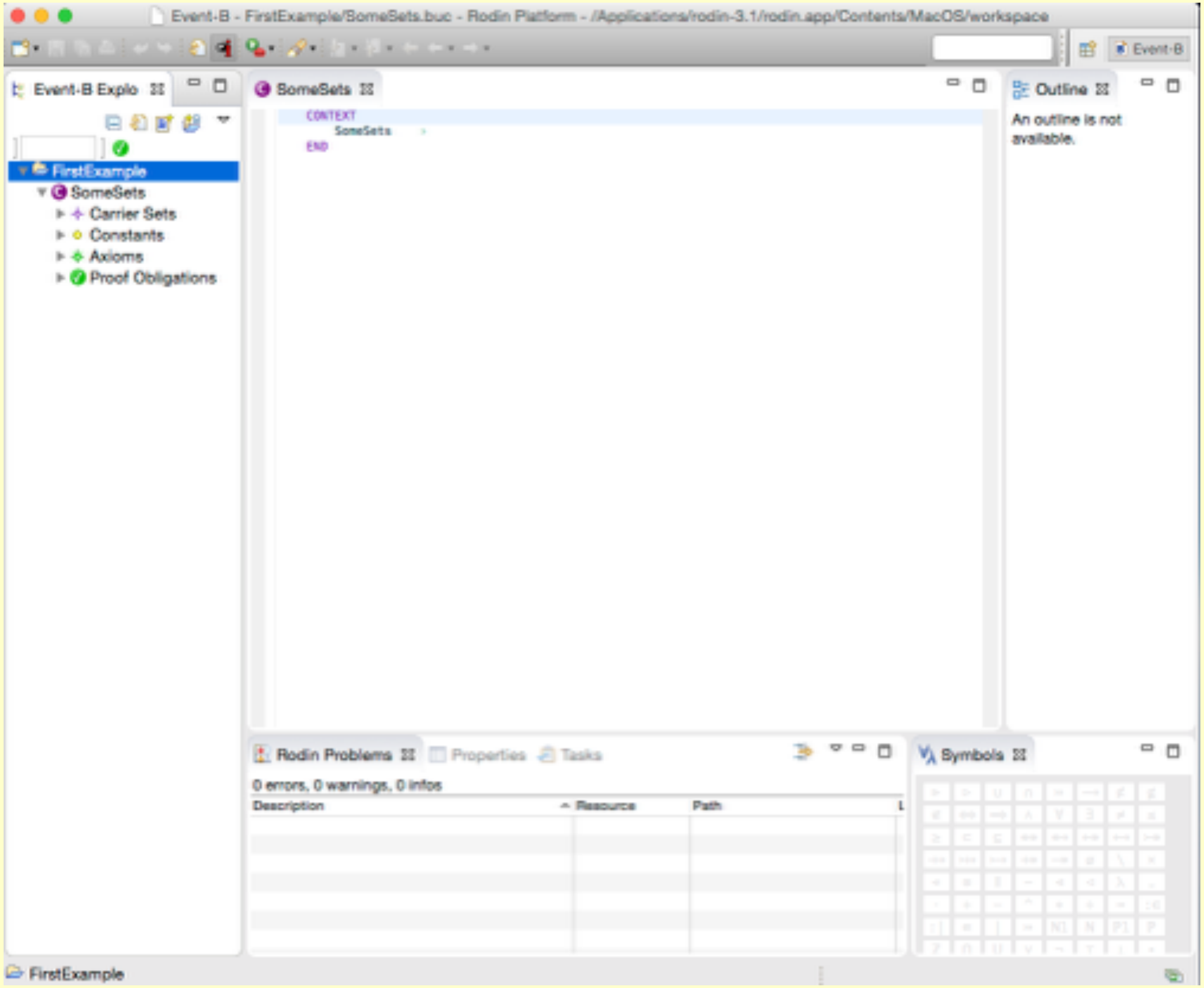
A NEW PROJECT



Now add an Event-B Context component – `SomeSets` - to our project (in order to do some mathematics using set notation)

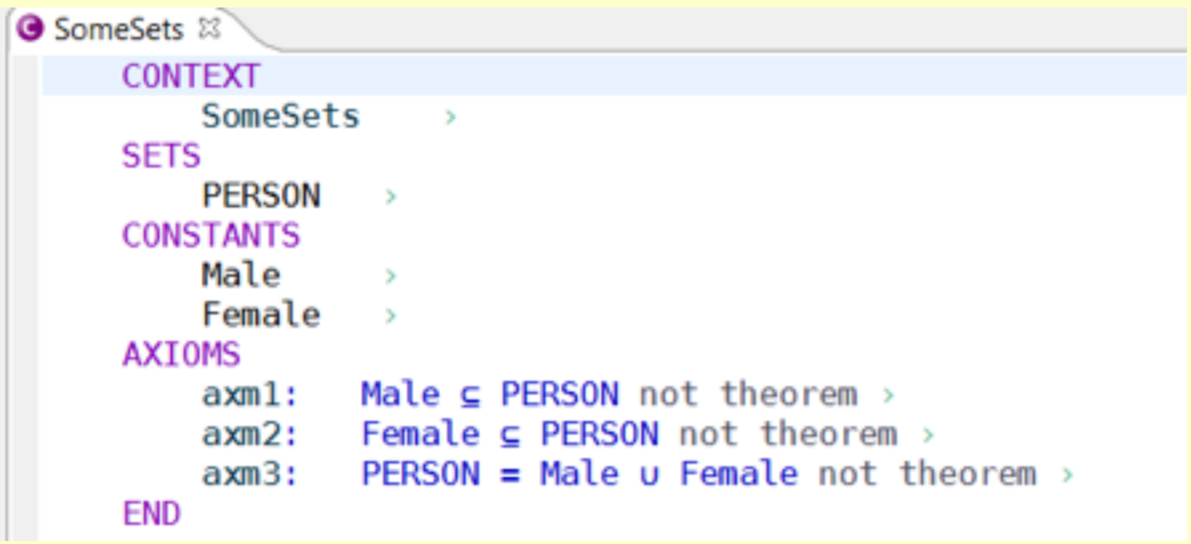


Check that the `SomeSets` context is empty



Now, try and write the following context specification … how intuitive is the RODIN user interface for beginners?

Use the wizards



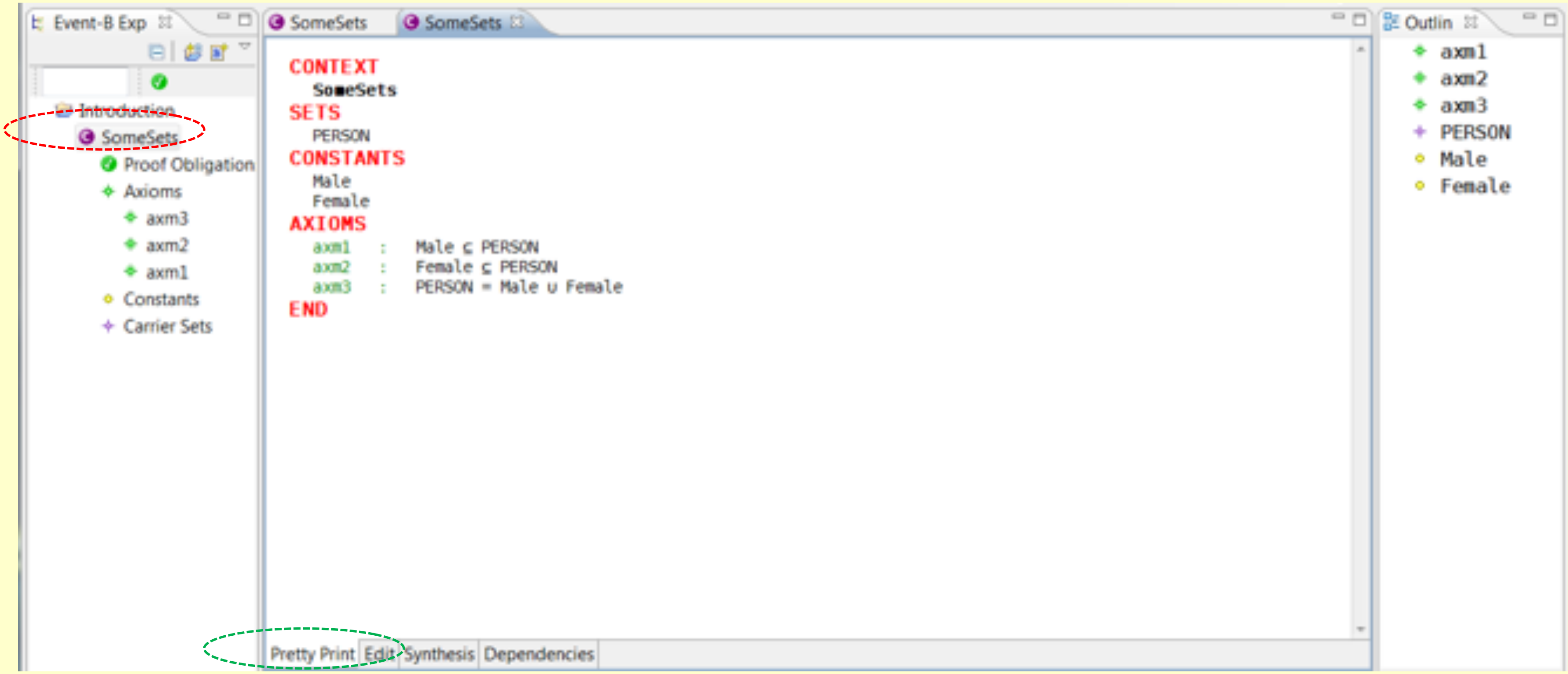
You may need some help with editing:

http://wiki.event-b.org/images/Summary.pdf

Can you explain what it means (in English)?

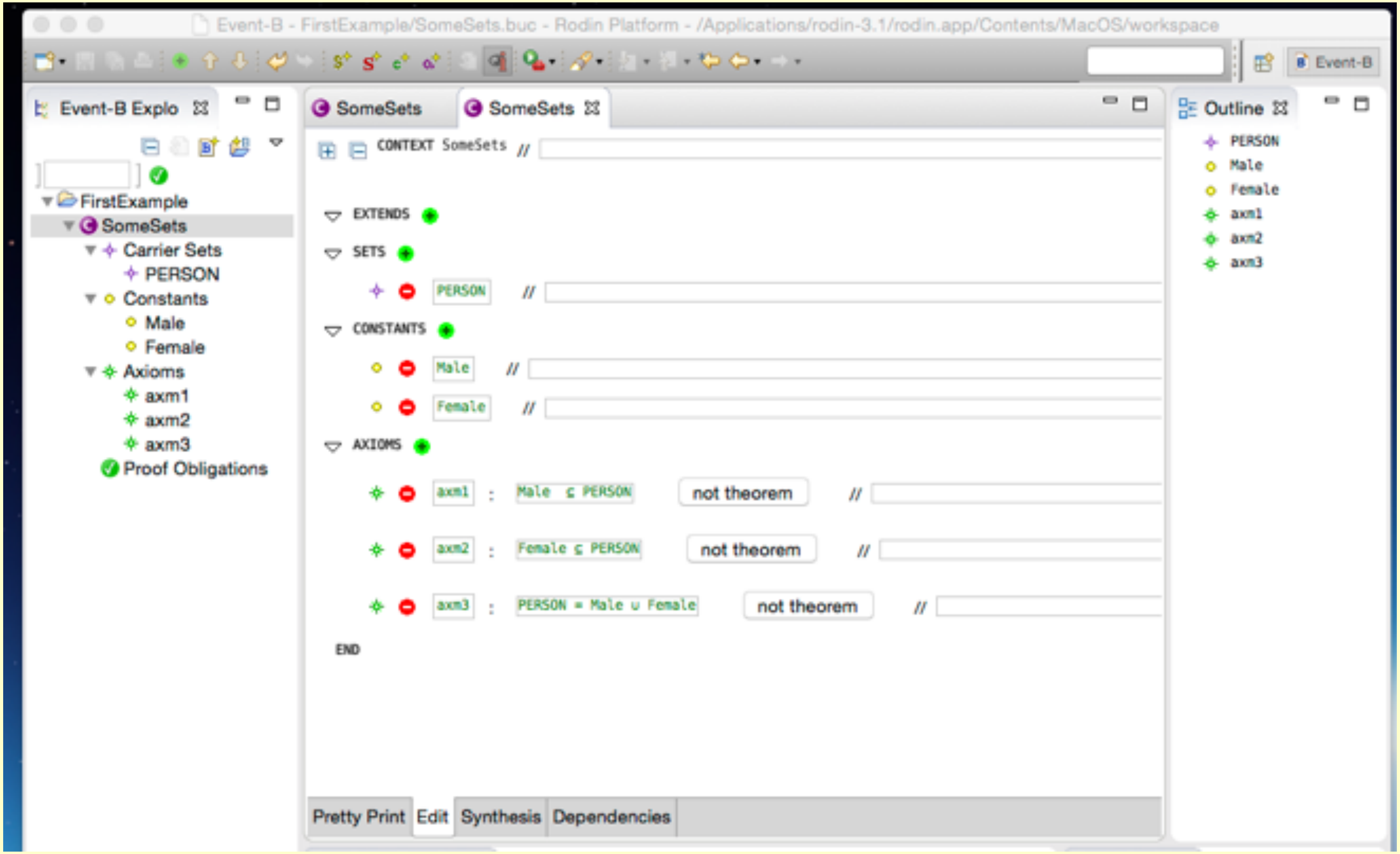
There is a second kind of editor/view that is popular

Right click `SomeSets` and select Open With Event-B context editor



This is the pretty print view of the Event-B context editor

Use the edit view to add English **comments** documenting the specification



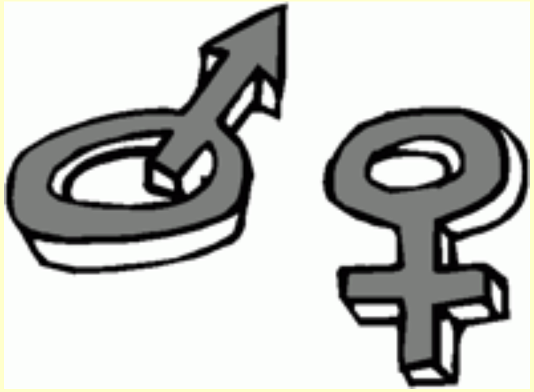
Use the editor view to add English **comments** documenting the specification

```
CONTEXT
   SomeSets
SETS
   PERSON
CONSTANTS
   Male
   Female
AXIOMS
   axm1   :   Male ⊆ PERSON        //   All males are persons
   axm2   :   Female ⊆ PERSON       //   All females are persons
   axm3   :   PERSON = Male ∪ Female      //   All persons are male or female
END
```

```
CONTEXT
      SomeSets      ›
SETS
      PERSON   ›
CONSTANTS
      Male     ›
      Female   ›
AXIOMS
      axm1:   Male ⊆ PERSON not theorem › All males are persons
      axm2:   Female ⊆ PERSON not theorem ›All females are persons
      axm3:   PERSON = Male ∪ Female not theorem ›All persons are male or female
END
```

NOTE: Both the different editor views get updated

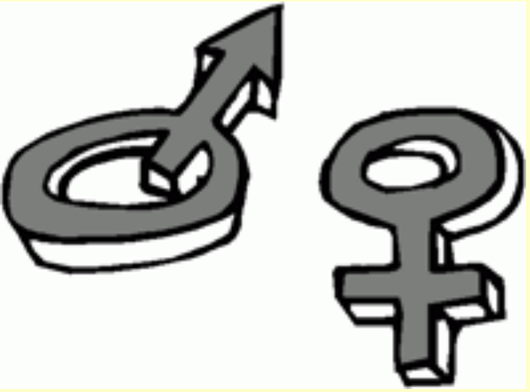
But, our specification is incomplete: it allows male and female sets to overlap



Now, can you specify that if you are male then you cannot be female

There are lots of different axioms that you could write.

Can you find one that you think is correct?



$\text{Male} \cap \text{Female} = \varnothing$

$\text{Female} = \text{PERSON} \setminus \text{Male}$

$\text{Male} = \text{PERSON} \setminus \text{Female}$

QUESTION: Are these axioms equivalent?

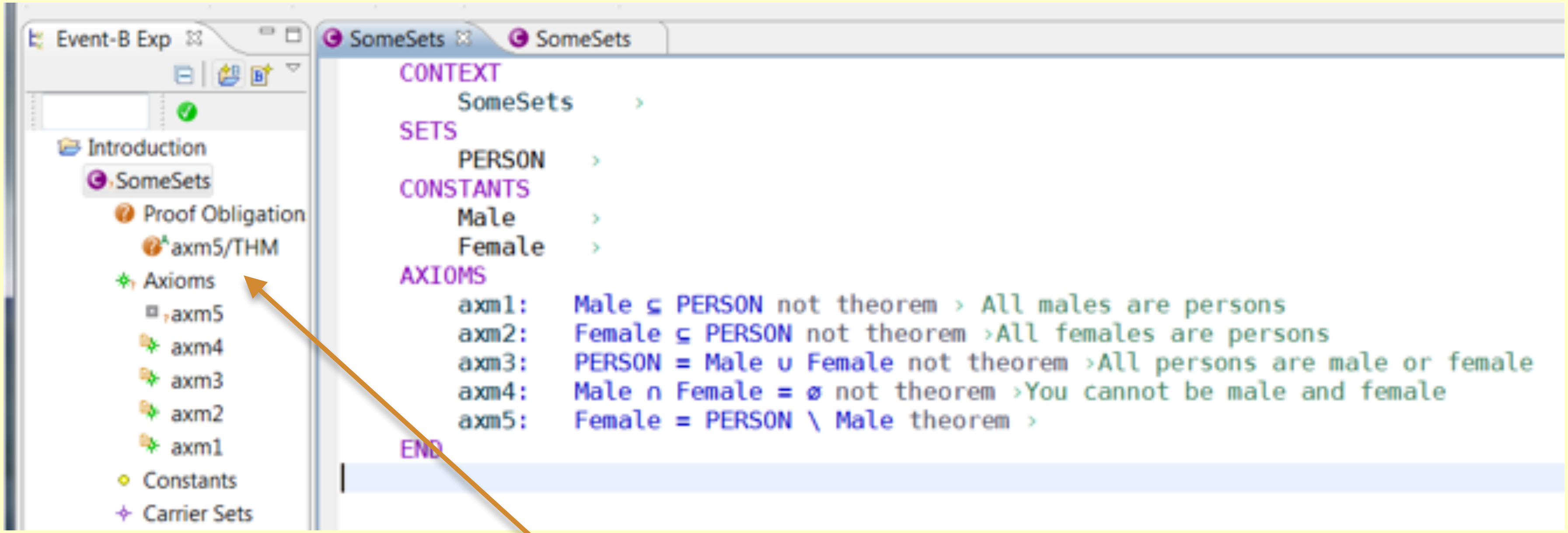
Let's see using RODIN:
Does $\text{Male} \cap \text{Female} = \varnothing$ imply $Female = PERSON \setminus Male$

```
CONTEXT
   SomeSets
SETS
   PERSON
CONSTANTS
   Male
   Female
AXIOMS
   axm1   :   Male ⊆ PERSON          //   All males are persons
   axm2   :   Female ⊆ PERSON        //   All females are persons
   axm3   :   PERSON = Male ∪ Female      //   All persons are male or female
   axm4   :   Male ∩ Female = ∅      //   You cannot be male and female
   axm5   :   Female = PERSON \ Male
END
```

Add Male $\cap$ Female $= \varnothing$ as an axiom (<u>non theorem</u>)

Add *Female = PERSON* $\setminus$ *Male* as a <u>theorem</u>

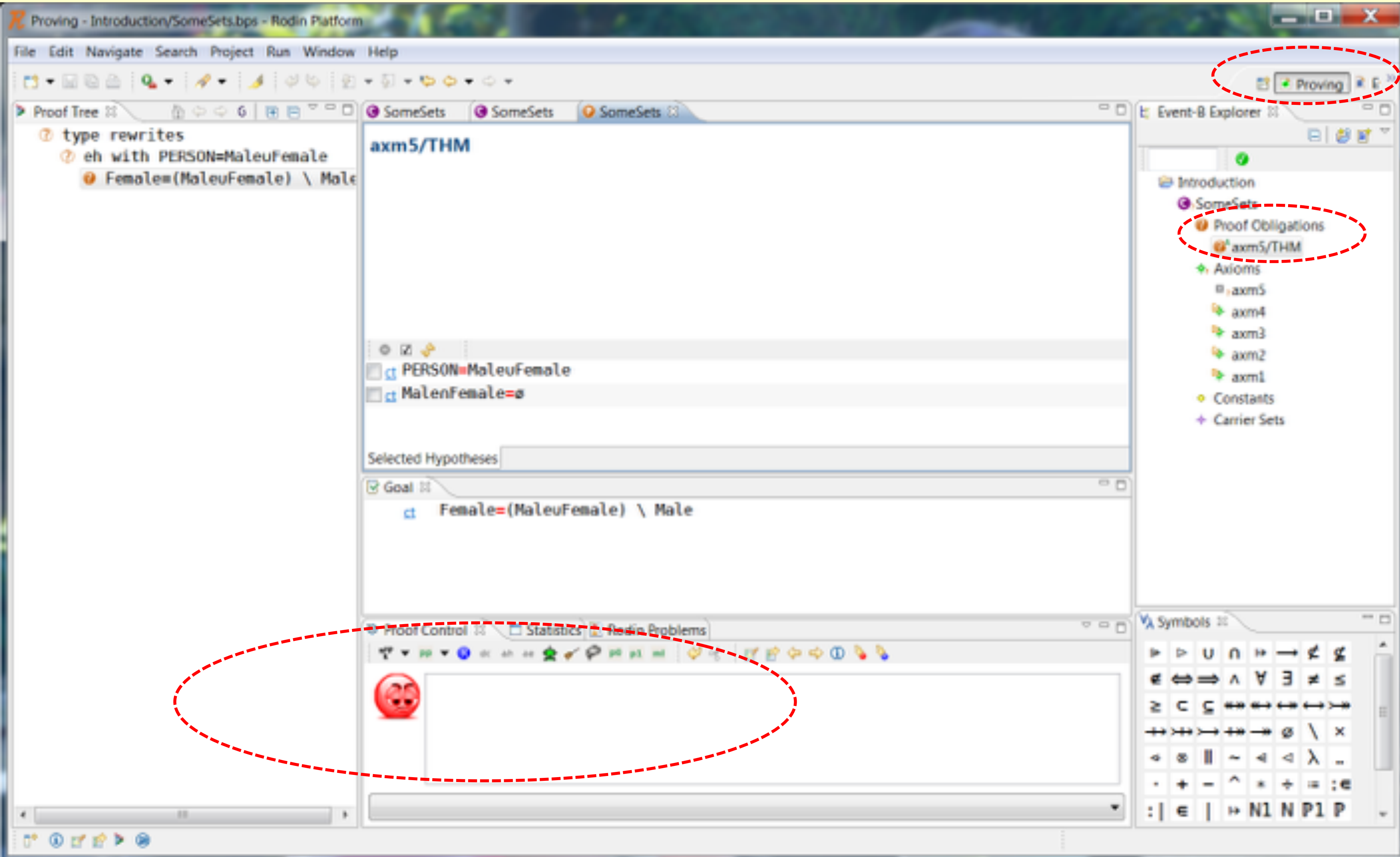
QUESTION: What do you notice about the `SomeSets` context properties ?



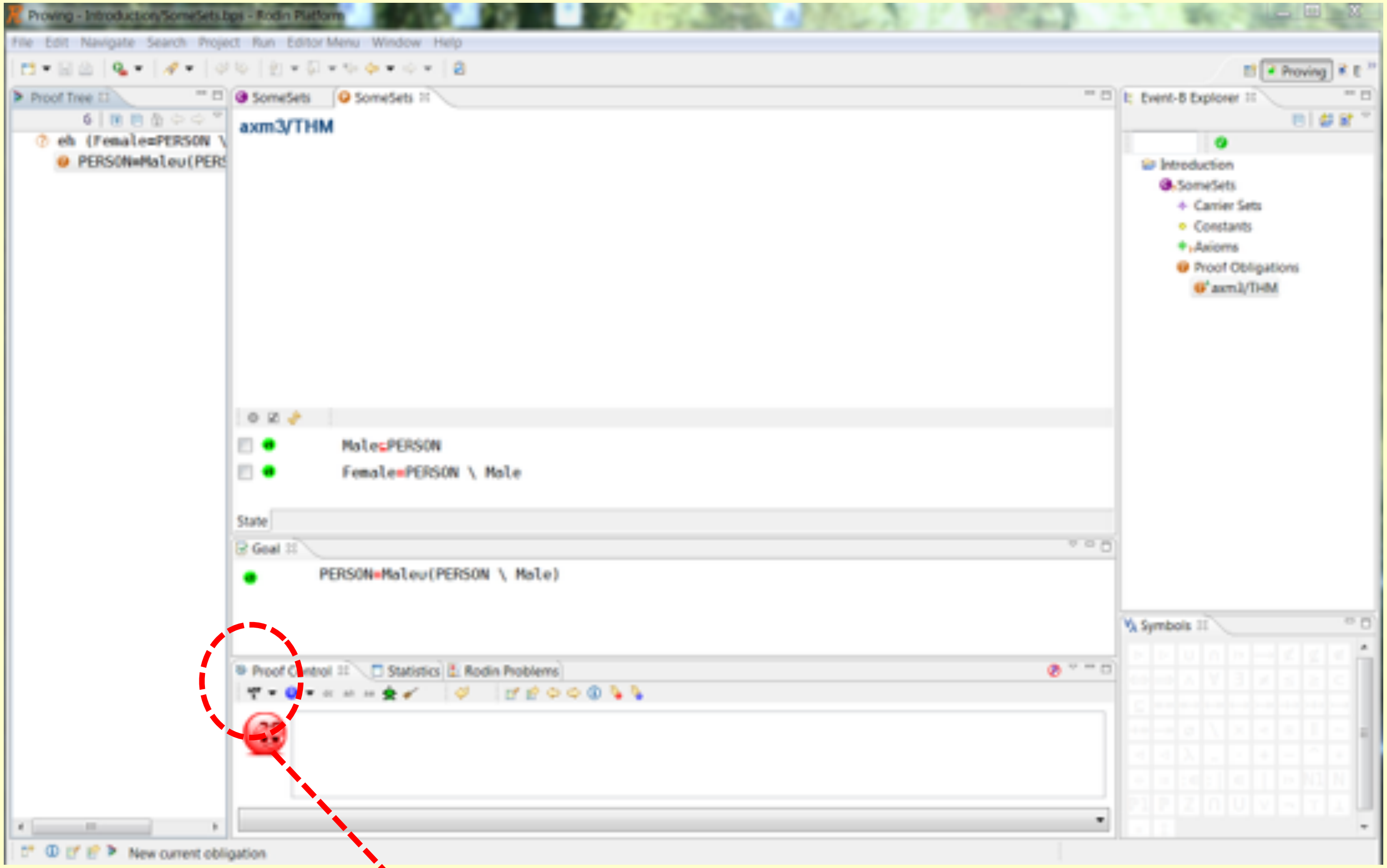
We may have some proof obligations to discharge with the prover

If you have a theorem that is not proved you need to launch the prover view in order to discharge it.

We can open the Proving perspective to try to prove axm5

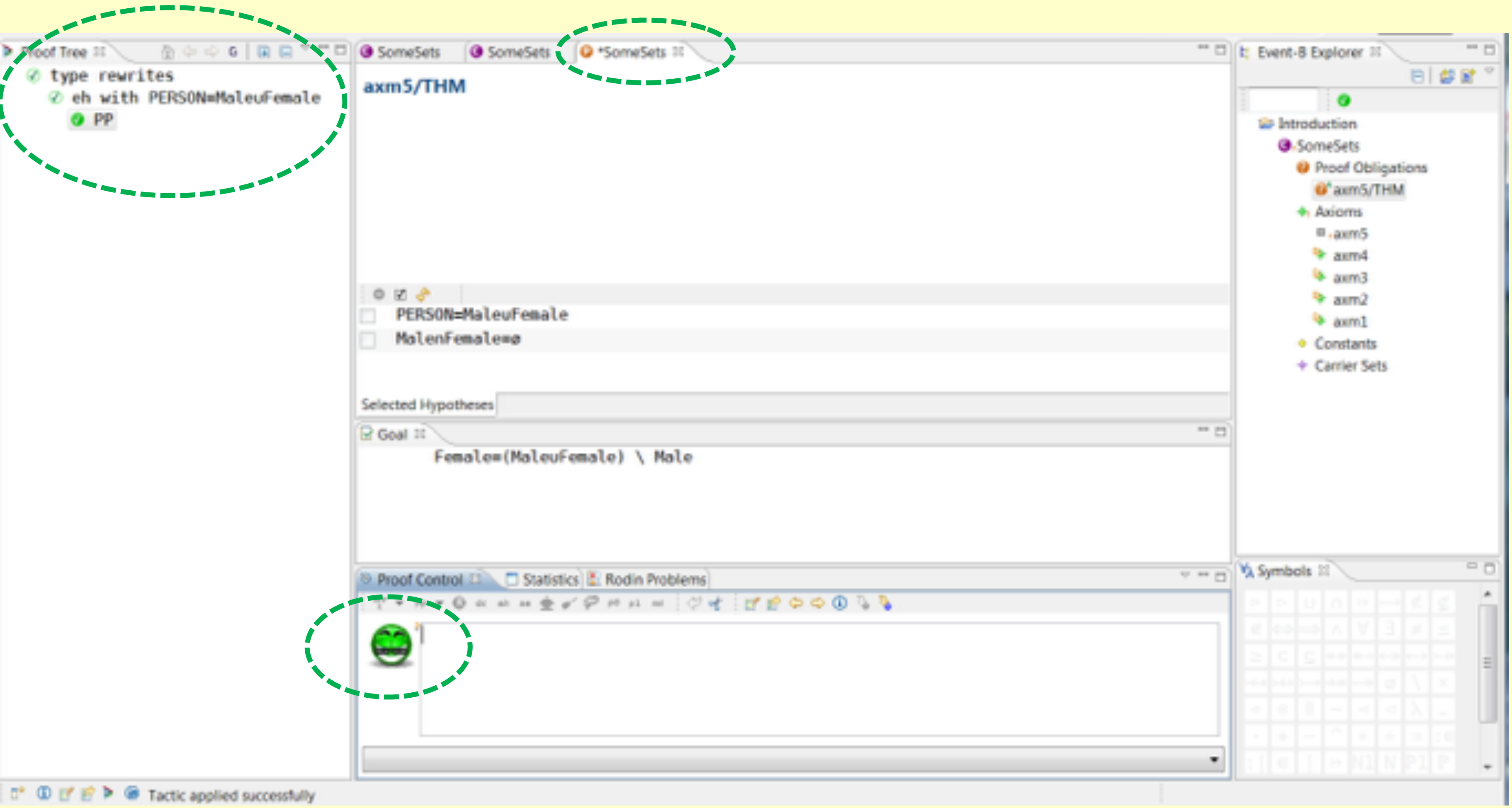


This shouldnt be so hard for a prover to Prove automatically



Launch PP (with all hypotheses)

Don’t forget to save the proof



Can you use RODIN to check equivalence of these axioms?

1. $Male \cap Female = \varnothing$

2. $Female = PERSON \setminus Male$

3. $Male = PERSON \setminus Female$

We have already proven that 1 => 2

Question: what else do we need to prove?

Let’s add some more sets to our context.

Imagine that we wish to build a context that models family relations

The types of concepts that we need are:

- Mother,
- Father,
- Parent,
- Child,
- Brother,
- Sister,
- Sibling,
- Ancestor,
- Descendant

TO DO: See how many of these you can model using just sets

Parents – the set of people who are mothers or fathers is pretty easy, eg:

```
CONTEXT
  SomeSets

SETS
  PERSON

CONSTANTS
  Male
  Female
  Mothers
  Fathers
  Parents

AXIOMS
  axm1   :   Male ⊆ PERSON      //    All males are persons
  axm2   :   Female = PERSON \ Male      //    Any person who is not male is female
  axm3   :   PERSON = Male ∪ Female      //    A person is either male or female
  axm4   :   Mothers ⊆ Female
  axm5   :   Fathers ⊆ Male
  axm6   :   Parents = Mothers ∪ Fathers

END
```

But, it is not clear how to model the other concepts (if it is indeed possible) without introducing relations between sets

Let us return to the family in RODIN to try and model the relationships

```
CONTEXT
  SomeSets

SETS
  PERSON

CONSTANTS
  Male
  Female
  Mothers
  Fathers
  Parents
  MotherOf
  FatherOf
  ParentOf

AXIOMS
  axm1  :  Male ⊆ PERSON        //  All males are persons
  axm2  :  Female = PERSON \ Male     //  Any person who is not male is female
  axm3  :  PERSON = Male ∪ Female     //  A person is either male or female
  axm4  :  Mothers ⊆ Female
  axm5  :  Fathers ⊆ Male
  axm6  :  Parents = Mothers ∪ Fathers
  axm7  :  MotherOf ∈ PERSON → Mothers    //  All persons have a single Mother
  axm8  :  FatherOf ∈ PERSON → Fathers    //  All persons have a single Father
  axm9  :  ParentOf = MotherOf ∪ FatherOf    //  Your parent is either:
                                             //  your mother or your father

END
```