# Model Abstraction Using CTL Formulae

Yorgo Zoughby and Al-Abbass Khalil*

*American University of Beirut, Department of Computer Science, Beirut*

E-mail: ytz00@mail.aub.edu/aak103@mail.aub.edu

## Introduction

The purpose of this report is to give a description of the work done in adding the feature Abstraction by Set of Formulae to the tool Eshmun. This feature takes as input, a model and a set of CTL formulae, and gives as output an abstracted model according to the formulae. You may also modify (delete states/transitions by model repair) this abstracted model and concretize back to the original with those changes taken into effect.

The report structure will be as follows: **Method** where we discuss the idea we wish to implement, **Implementation** how the idea was actually implemented, **User Manual** an example of how to use this new feature in Eshmun and finally **Conclusion & Future Work**.

## Method

### Truth Table of Formulae

Given a model made up of states and a set of formulae. We build the truth table of formulae which consists of the states in the model along with the truth value of each state on each formula.

| States\Formulae | φ 1 | φ 2 | ..... | φ m |
|---|---|---|---|---|
| S0 | 0 | 0 | .. | 0 |
| S1 | 0 | 0 | .. | 0 |
| S2 | 1 | 0 | | 0 |
| . | . | . | . | . |
| . | . | . | . | . |
| . | . | . | . | . |
| Sn | 0 | 1 | . | 1 |

Figure 1: Truth Table

For example, in the figure above, state S0 does not satisfy $\phi 1$ & $\phi 2$, whereas S2 satisfies $\phi 1$ but not $\phi 2$. In order to fill this table, we will use Global Model Check, which returns the states which are true given a formula.

## Abstract Classes

Now that we have the truth table of formulae, we can find the unique classes which will represent our abstracted states. Each class has a unique id; for example (0,0,...,0) is the class of states that satisfy none of the formulae, whereas (1,1,...,1) is the class of states that satisfies all formulae.

The total number of possible classes is $2^m$, where m is the number of formulae. In the case where we have $2^m$ classes, this means that every possible class represents some states in the model. So there exists some states that satisfy none of the formulae, and there exists some states that satisfy all the formulae, and everything in between. Therefore, by going through the truth table we can derive the classes needed to represent the abstracted model.

## Abstraction

Now we know what the abstract classes are, and we know which states belong to those classes. We must create a kripke structure to represent this abstracted model. Our kripke structure is made up of abstract states that represent the junction of all the states that made up this abstract state.

### Concretization

In the abstract model, we can delete abstract states or transitions using model repair. When concretizing back to the original model, the states/transitions related to the deleted abstract states/transitions are also deleted.

### Model Check

While in the abstract model, we can model check any abstract model. Since this feature still a beta, the model checking might behave in an unexpected way.

### Model Repair

Model repair of the abstract structure also work. Repairs made will be propagated to the concretized model as well.

## Implementation

### Overview

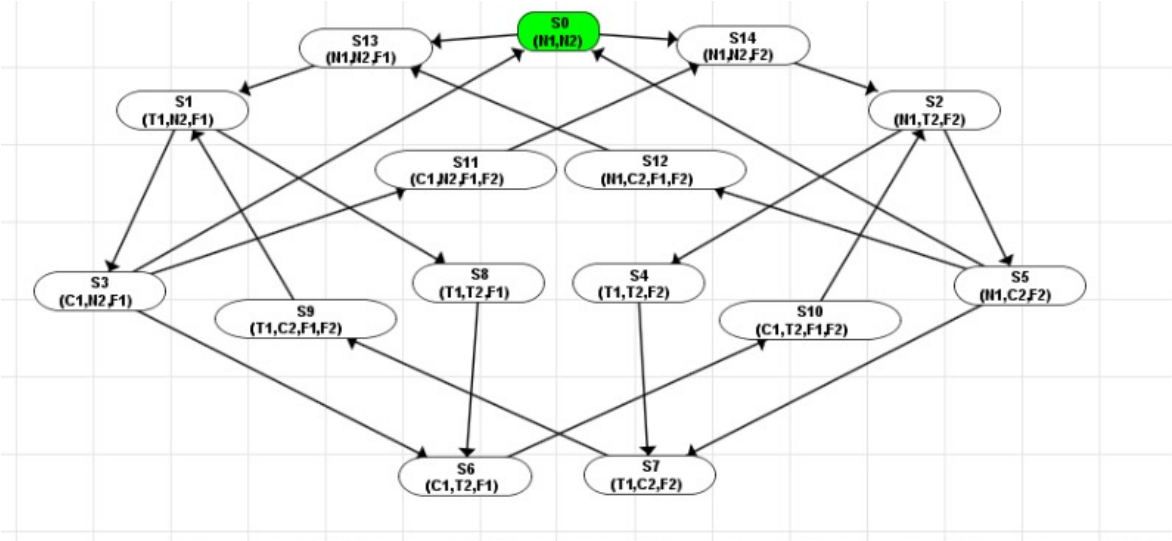
First, we created a button in Eshmun menu bar under the "tools" section called Abstract by set of CTL Formulae. This button will create and launch a new process that runs in parallel with Eshmun but will not affect the runtime behavior of Eshmun and wont cause any new bugs since it is an independent process that will handle only this new feature. When this button is pressed, a new UI interface will launch and will allow the user to input many CTL formulae (up to 10 formulae for now in the beta version). Every formula is then parsed by the CTL parser to make sure no wrong formulae was entered by mistake. Then, every formula is passed to the stateByStateModelCheck() function (Global Model Check) which will return the states that satisfy the checked formula. This allows us to create the truth table details above and then derive the distinct classes and their states. Finally, every set of

states that belong to the same abstract class are merged together (The procedure of merging the states here is very similar to the merging of states that is done in the abstraction with manual selection).

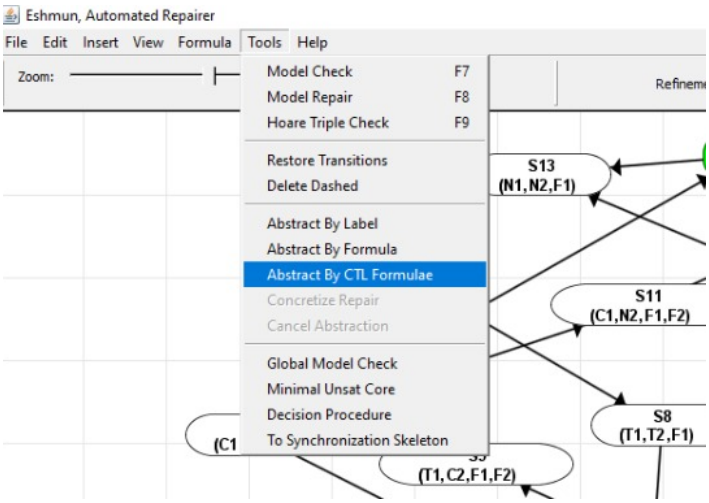
# User Manual

We demonstrate an example on how to use this feature: abstraction by set of formulae.

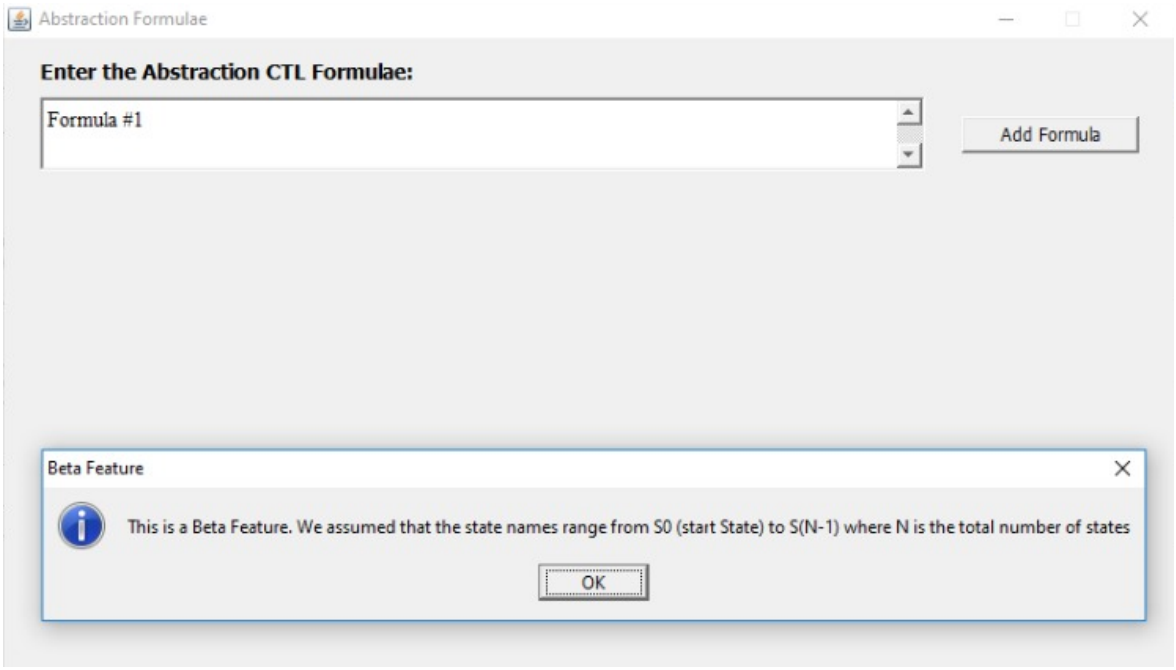
1. First, the example will be ran on mutex exclusion between 2 processes.



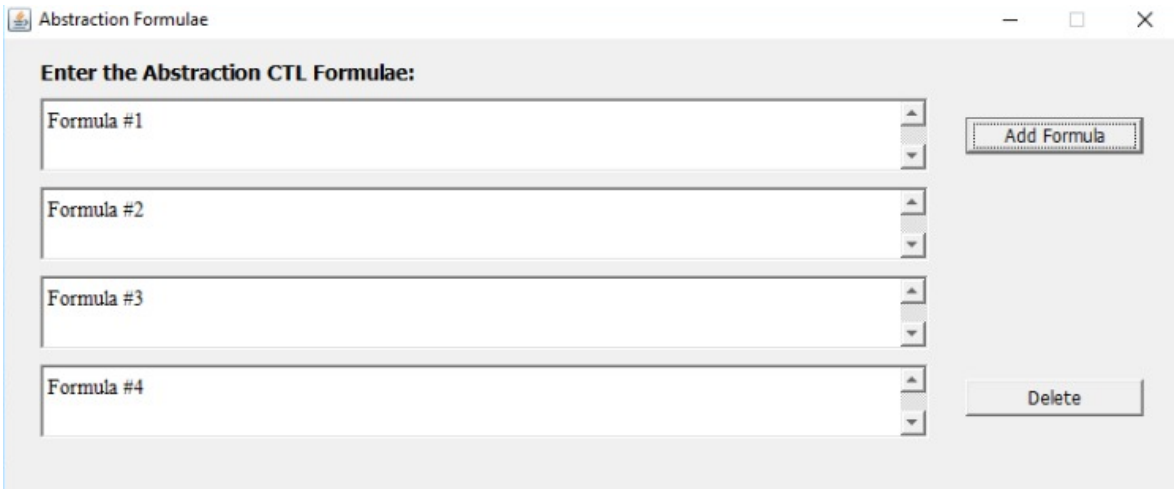

2. Choose Tools >Abstract By CTL Formulae



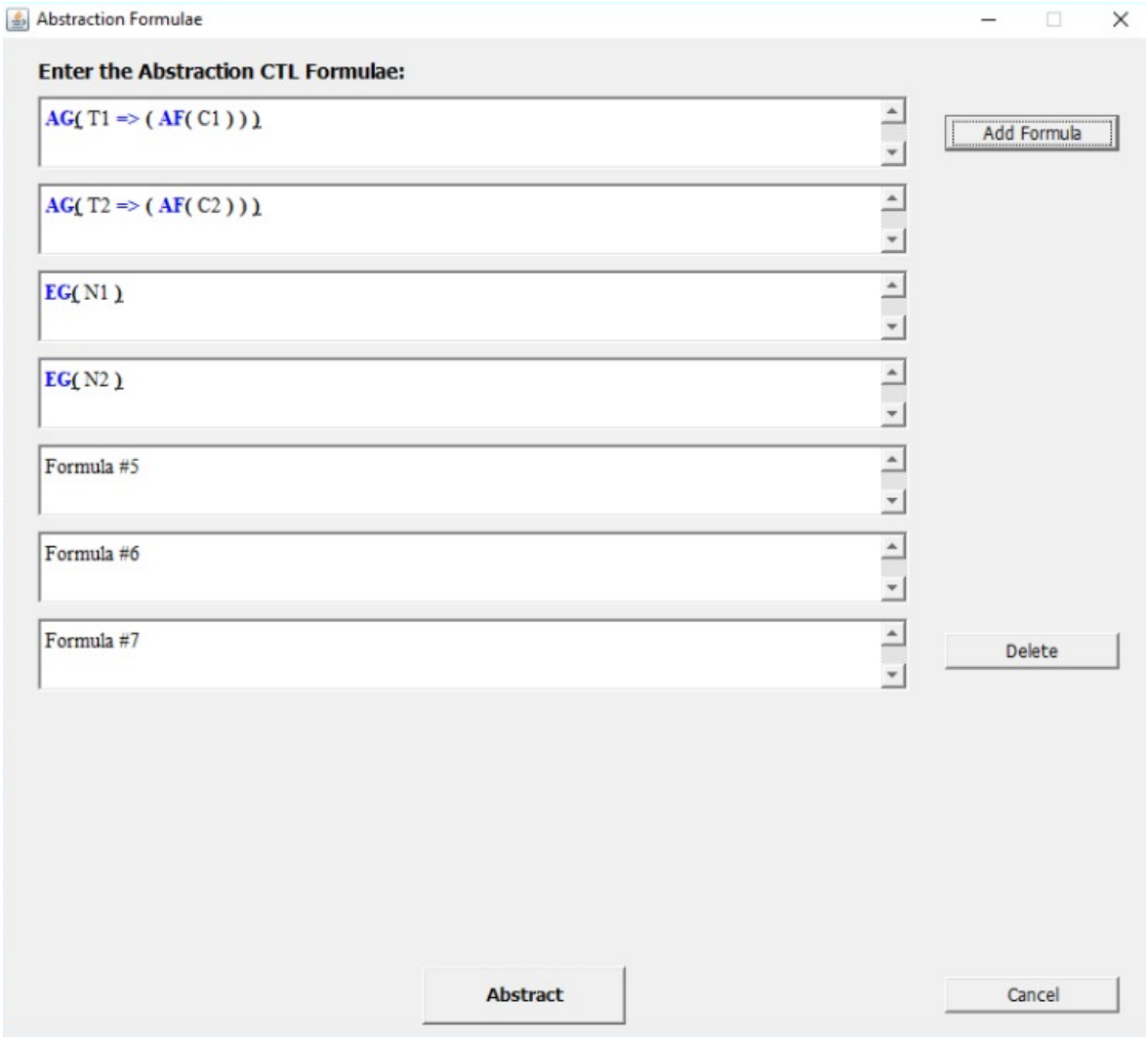
3. We assume that the states will be named from S0 to S(n-1) where n is the number of states



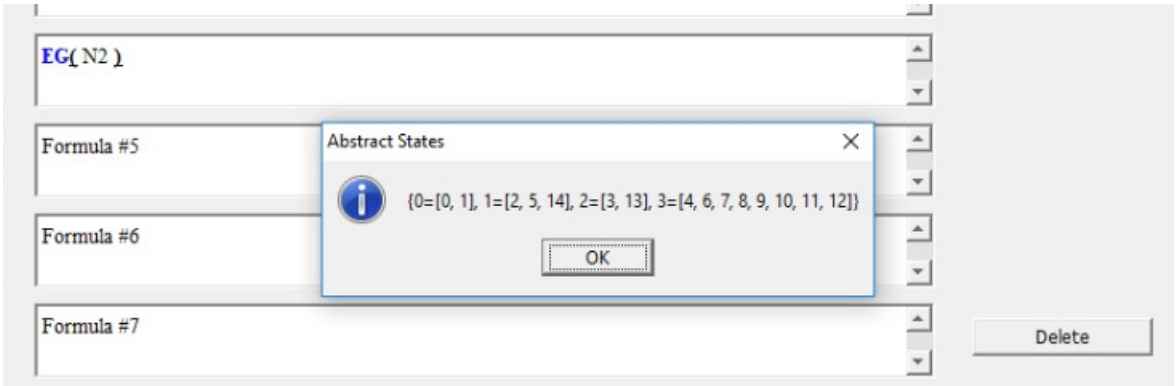

4. Add Formula adds as many formulas as required. Delete removes the last formula added.



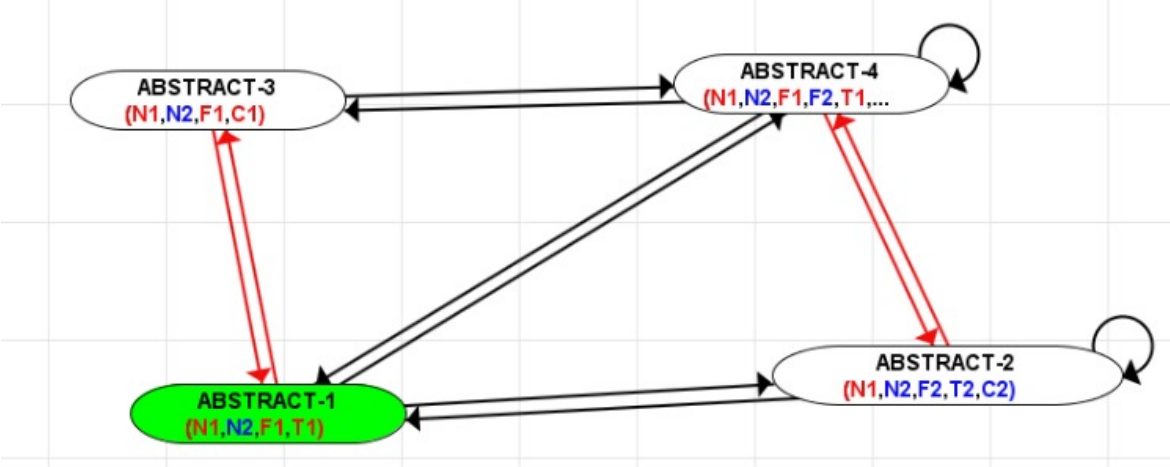
5. Unused textboxes will be ignored



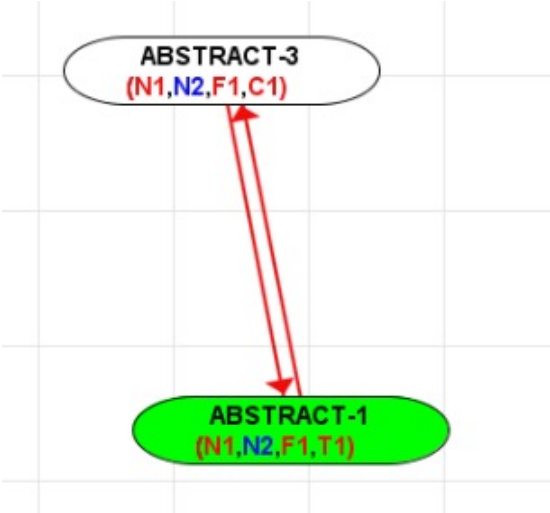

6. When you click Abstract, a textbox will appear showing the abstract classes and the states belonging to those classes.



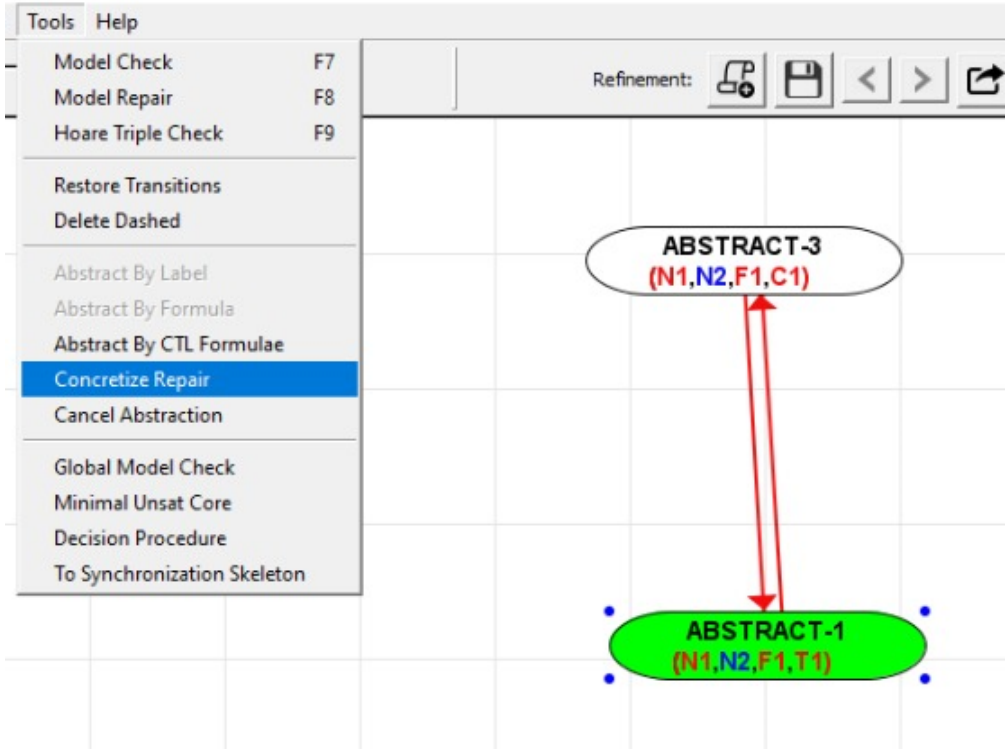
Also the abstract model is shown.



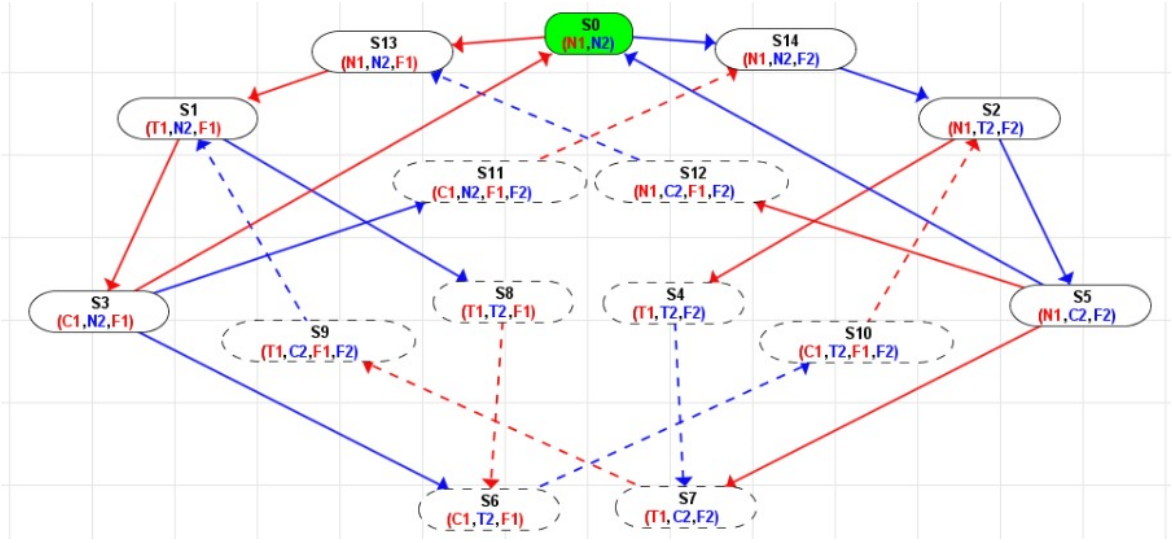

7. You can delete classes or transitions using model repair, for example here we delete ABSTRACT-2 and ABSTRACT-4



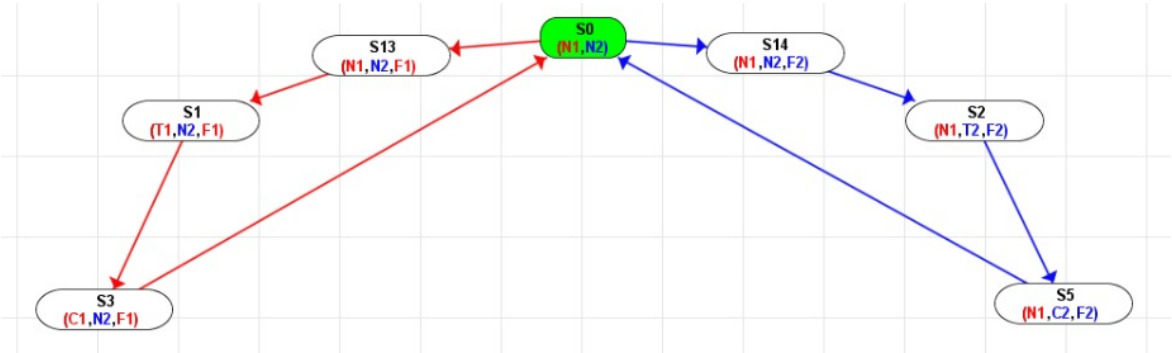
8. Choose Tools >Concretize Repair




We get the following:



9. Finally after removing the dashed we get this model.




# Conclusion & Future Work

1. Eshmun now has an abstraction by formulae feature. When abstracting, you can delete states/transitions using model repair in the abstracted model and concretize back to the original model where you can delete the states/transitions represented by the ones deleted in the abstracted model.

2. When displaying abstract classes, the disjunction of the labels of the states belonging to an abstract class are displayed comma separated. In order to solve this we could create a new state object that extends the current one and then update kripke parser to handle this new state architecture.

3. Model checking and model repair can be on the abstracted model. However, since we have to display the labels comma separated, so the results might not be accurate.

4. We had an assumption where the states have to be named from S0 to Sn-1 where n is the number of states. We hope to remove this constraint later on.

# Acknowledgement

We would like to thank Prof. Paul Attie and Mr. Chukri Soueidi for their guidance throughout the semester. This project would not have been possible without their support.