CMU 17-614 Formal Methods

Project: Structural Modeling

Garlan & Kang **Due: October 10, 2022**

The purpose of this project is to give you experience in modeling complex structures that arise in software systems. In particular, you will use the object modeling techniques discussed in class to build a model of a small social network in Alloy. You will also model and analyze privacy aspects of this social network. There have been numerous examples of privacy violations in popular social networks due to flaws in privacy settings. Through this project, you will see how modeling and analysis of a system design could help identify potential flaws early in the development process.

A specification of the system is provided along with this project description in a file called Social Network Spec.pdf.

You should carry out this project in your assigned team. Make sure that everyone in the group contributes to the overall effort. Each team should submit a single write-up of the project, due at the beginning of class on the project due date. We have posted a template for a group project write-up under the LaTeX section of the course web site.

Submit a .zip file (named "project-teamN.zip", where N is your team number) containing (1) your project report and (2) all Alloy models (plus visualization theme files, if any) on Canvas before class on the due date. Include a README file indicating the main model file that encompasses your system.

Start early! This project is likely to take more time than you might expect.

Task 1 (70 points):

Your first task is to develop a formal model of the social network in Alloy based on the provided specification. Your model should include three major aspects of the system: (1) the structure of the social network, including users and their friendships, (2) operations for modifying user content, and (3) privacy settings that control access to those content.

Your model should include a predicate called `invariants` that captures the set of invariants that must hold over each state of the system. The predicate may take any number of arguments. In addition, include assertions to check that each operation preserves the invariants.

You should build your model in an incremental manner. Start by building a simple static model that describes the basic structure, and try generating sample instances as a sanity check to see whether your model is not missing any key invariants. Then, add operations one-by-one and check that they preserve those invariants. Finally, augment your model by adding privacy settings and their effects on the visibility of contents. For this process, we encourage you to use the module system in Alloy, separating your specification into multiple smaller files.

Be sure to document your model with comments.

Effective use of predicates for capturing and reusing common constraints will be one of the evaluation criteria.

Make use of the analysis capability in Alloy, including the unsat core, which can help identity inadvertent over-constraints in your model.

Answer the following questions in your project write-up. Where appropriate, your answers should include pointers to the part (e.g., line number) of your model that substantiates your answer.

1. Include an object model diagram for your system.
2. Describe any decisions that your team made in resolving ambiguities in the English document. Also describe alternatives you considered and rejected.
3. Describe any invariants that are not explicitly mentioned in the given document but were discovered during the modeling process.
4. Describe an invariant that was violated by an operation in your model and how it was fixed.
5. List the scope used for the analysis of invariant preservation. Why did you choose this scope, and why do you think it is sufficient?

Task 2 (20 Points):

For the second part of the project, you will use the model that you've constructed in Task 1 to analyze whether privacy provided by Nicebook is as strong as it claims to be. In particular, extend your model as follows:

1. Define an Alloy function called `canView`, which takes at least one parameter that represents some user in the social network, and returns the set of all content that can be viewed by that user. You may use any number of helper predicates or functions to define `canView`.
2. You will use `canView` to check whether it is possible for a user to access a piece of content against the intent of the owner. In particular, we say that a privacy violation has occurred if a user is able to view a piece of content without adhering to the privacy level that is assigned to the content by its owner. For example, suppose that Alice and Bob are not friends, and Alice has a photo that is assigned privacy setting Friends; if Bob is able to view the photo, this would be considered a privacy violation. Define an Alloy assertion called `NoPrivacyViolation` to check that no such violation is possible.
3. Does your model satisfy `NoPrivacyViolation`? If so, explain why. If not, include a counterexample that demonstrates a violation, and suggest a modification to the design of privacy settings in Nicebook. Describe any alternatives you considered and rejected, along with justifications for your decision.

Task 3 (10 points): Reflection

In this part of the report we would like you to reflect on your modeling experience.

1. What are the strengths and weaknesses of Alloy and its tool, the Alloy Analyzer?
2. Under what situations would you recommend its use (or not)? Why?
3. With respect to this notation, what is the single most-important future development that would be needed to make it more generally useful to practitioners?