Available online at www.sciencedirect.com
SCIENCE DIRECT
ELSEVIER
Electronic Notes in Theoretical Computer Science 130 (2005) 3–21
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

An Abstract Equivalence Notion for Object Models

Rohit Gheyi[1,2] Tiago Massoni[3] Paulo Borba[4]

Informatics Center
Federal University of Pernambuco
Recife, Brazil

Abstract

Equivalence notions for object models are usually too concrete in the sense that they assume that the compared models are formed by elements with the same names. This is not adequate in several situations: during model refactoring, when using auxiliary model elements, or when the compared models comprise distinct but corresponding elements. So, in this paper, we propose a more abstract and language-independent equivalence notion for object models. It supports, as desired, abstraction from names and elements when comparing models. We use the PVS system to specify and prove properties of our notion. It is illustrated here by comparing simple models in Alloy, a formal object-oriented modeling language, but has also been applied for deriving a comprehensive set of algebraic laws for Alloy.

Keywords: equivalence notion, theorem proving, object models

1 Introduction

Comparing deliverables during the software development process is quite important. In fact, there are several approaches for comparing the behavior of programs. For instance, these are useful during maintenance, when we

[1] We would like to thank Augusto Sampaio, Daniel Jackson and members of the Software Design Group at MIT, Carlos Pombo, and members of the Software Productivity Group at UFPE, for their important comments. This work was supported by CNPq and CAPES (Brazilian research agencies).

[2] Email: rg@cin.ufpe.br
[3] Email: tlm@cin.ufpe.br
[4] Email: phmb@cin.ufpe.br

doi:10.1016/j.entcs.2005.03.002

may wish to replace a given component by a behaviorally equivalent, better structured one. It is similarly useful to compare design models, which can be expressed by object modeling languages such as Alloy [13] or UML class diagrams [1].

However, current equivalence notions for object models are usually too concrete. They assume that the compared models are formed by elements with the same names. This is not adequate in several situations. For example, model refactoring changes the structure of models, yet maintaining the previous semantics. Nevertheless, it is difficult to verify whether the resulting model preserves the semantics, especially when model elements are replaced by alternative structures. Furthermore, auxiliary model elements may be used, which should be ignored when calculating equivalences. Also, when the compared models comprise distinct but corresponding elements, we can easily find models that are intuitively equivalent but cannot be proved so based on most equivalence notions.

In this paper, we propose a more abstract equivalence notion for object models. It supports, as desired, abstraction from names and elements when comparing models. It is flexible enough for comparing only parts of object models, by relying on a chosen set of relevant elements— the alphabet— and a mapping from relevant elements to their equivalent counterparts in the corresponding model— the view. Furthermore, this equivalence notion is applicable to any object modeling language, as we regard an object model as an abstract description of object structure and inter-relationships. We encoded it in the Prototype Verification System (PVS) [18] specification language and formally derived several properties by using the PVS prover [19]. These proofs show the independence of the notion with respect to the underlying semantics of the object modeling language.

Here we illustrate our equivalence notion by comparing simple models, but it has also been applied for deriving a comprehensive set of algebraic laws for Alloy [10]. It is also useful for other applications of semantics-preserving model transformations. For instance, we used it in an atomization process [5], which transforms an Alloy model to improve the analysis performance of the Alloy Analyzer tool [14]. We show that an atomization transformation preserves the semantics of the model, by applying algebraic laws and the equivalence notion proposed here [10]. Moreover, this notion can be used to formally derive model refactorings that can be useful for introducing design patterns [7] into a model. Additionally, a flexible equivalence notion can be useful, for instance, when we are interested in verifying whether partial models are equivalent. This notion can also be valuable for comparing components' specifications. In case they are equivalent, one component can be replaced by another following