To be presented at the 7th International Symposium on Computer and Information Science 1992

Transformational Derivation in the Programming Logic TK

Martin C. Henson,
Department of Computer Science, University of Essex, Colchester, Essex, ENGLAND.
Department of Computer Science, University of Otago, Dunedin, NEW ZEALAND.
hensm@uk.ac.sx

1 Abstract

We examine transformational programming techniques within the programming logic TK. In particular we investigate the transformational technique known as type simulation. The internalisation of this technique within TK is illustrated with respect to a derivation of the α–β-pruning algorithm from a specification of mini-maxing.

2 Introduction

In this paper we investigate transformational programming techniques within the programming logic TK. This theory has been elaborated in detail in [HeT88] and some aspects of program development within TK are explored in [Hen89a]. Our point of departure here is the transformational technique known as type simulation [Hen88] which is a generalisation of a idea due to Wand [Wan80]. We must, for reasons of space, refer the reader to the references for an detailed exposition of the programming logic TK. For the same reason most of the technical results of Section 4 are stated here without detailed proof. The interested reader may like to consult [Hen89c] for these. In Section 3 we describe the rules for inductive types which are central to our formalisation of transformations. The proofs of these propositions can be found in [Hen92]. In Section 4.1 we provide an introduction to transformation via type simulations and their formalisation in TK. Section 4.2 elaborates a more detailed account involving mutual induction and the derivation of the α-β-pruning algorithm.

3 Positive induction

The crucial rules of TK concerning inductive types are as follows:

$$\frac{z \in A}{z \in \Xi(A, \lambda X.B)} \ (\Xi\text{-}intro(i)) \qquad\qquad \frac{z \in B(\Xi(A, \lambda X.B))}{z \in \Xi(A, \lambda X.B)} \ (\Xi\text{–}intro(ii))$$

$$\frac{A \subseteq T \qquad B(T) \subseteq T}{\Xi(A, \lambda X.B) \subseteq T} \ (\Xi\text{-}elim)$$

where X must occur positively in B. Ξ-elim specialises to the following useful induction rule: