Summary: Circus specifications define both data and behavioural aspects of systems using a combination of Z and CSP constructs. Previously, a denotational semantics has been given to Circus; however, a shallow embedding of Circus in Z, in which the mapping from Circus constructs to their semantic representation as a Z specification, with yet another language being used as a meta-language, was not useful for proving properties like the refinement laws that justify the distinguishing development technique associated with Circus. This work presents a final reference for the Circus denotational semantics based on Hoare and He’s Unifying Theories of Programming (UTP); as such, it allows the proof of meta-theorems about Circus including the refinement laws in which we are interested. Its correspondence with the CSP semantics is illustrated with some examples. We also discuss the library of lemmas and theorems used in the proofs of the refinement laws. Finally, we give an account of the mechanisation of the Circus semantics and of the mechanical proofs of the refinement laws.

MSC:
68Q65 Abstract data types; algebraic specification
68Q55 Semantics in the theory of computing

Keywords:
relational model; concurrency; refinement calculus; theorem proving

Software:
ProofPower; Circus; Z; ZRC; Z/EVES

Full Text: DOI

References:


This reference list is based on information provided by the publisher or from digital mathematics libraries. Its items are heuristically matched to zbMATH identifiers and may contain data conversion errors. It attempts to reflect the references listed in the original
paper as accurately as possible without claiming the completeness or perfect precision of the matching.