

Continuation Semantics of Logic Programs with Disciplined Exception Handling

This paper presents a disciplined exception mechanism based on continuations. Continuations provide mechanisms for labels and jumps. We present a mechanism to preserve a Prolog program continuation to be invoked at a later point. This mechanism is used in the design of exception signals, handlers, and exception continuations for logic programs. An exception continuation specifies how a computation should continue (e.g. resume, undo, alternate, or terminate) after an exception is raised. The Z notation is used to specify the continuation semantics of logic programs with exceptions. Based on the semantic functions, an interpreter written in Prolog is derived to justify our approach.