MaskD specification language

Luciano Putruele

Departamento de Computación, FCEFQyN, Universidad Nacional de Río Cuarto,
Río Cuarto, Córdoba, Argentina, lputruele@gmail.com

Abstract. This document describes the grammar of the MaskD Guarded Language

1 Introduction

In this document we present the grammar of the design language for the MaskD tool.

2 Grammar

The MaskD language is a simple guarded language:

⟨*specification*⟩ ::= ⟨*globals*⟩ ';' ⟨*process_list*⟩ ⟨*main_program*⟩ ';'

⟨*globals*⟩ ::= ⟨*global_decl*⟩
| ⟨*globals*⟩ '**;**' ⟨*global_decl*⟩

⟨*global_decl*⟩ ::= global ID : ⟨*type*⟩

⟨*process_list*⟩ ::= ⟨*process*⟩
| ⟨*process_list*⟩ '**;**' ⟨*process*⟩

⟨*process*⟩ ::= process ID ⟨*params*⟩ ' { ' ⟨*decl_list*⟩ ';' ⟨*initial_cond*⟩
';' ⟨*branch_list*⟩ ';' '}'

⟨*initial_cond*⟩ ::= initial '**:**' ⟨*expr*⟩

⟨*params*⟩ ::= ⟨*param*⟩
| ⟨*params*⟩ '**,**' ⟨*param*⟩

⟨*param*⟩ ::= ID : ⟨*type*⟩

⟨*decl_list*⟩ ::= ⟨*decl*⟩
| ⟨*decl_list*⟩ ‘`;`’ ⟨*decl*⟩

⟨*decl*⟩ ::= ⟨*vbles_decl*⟩ ‘`:`’ ⟨*type*⟩

⟨*vbles_decl*⟩ ::= ID
| ⟨*vbles_decl*⟩ ‘`,`’ ID

⟨*type*⟩ ::= bool

⟨*branch_list*⟩ ::= ⟨*branch*⟩
| ⟨*branch_list*⟩ ‘`;`’ ⟨*branch*⟩

⟨*branch*⟩ ::= ⟨*label*⟩ ⟨*mode*⟩ ⟨*expr*⟩ ‘`->`’ ⟨*assig_list*⟩
| ⟨*label*⟩ ⟨*expr*⟩ ‘`->`’ ⟨*assig_list*⟩
| ⟨*mode*⟩ ⟨*expr*⟩ ‘`->`’ ⟨*assig_list*⟩
| ⟨*expr*⟩ ‘`->`’ ⟨*assig_list*⟩

⟨*label*⟩ ::= [ID]

⟨*mode*⟩ ::= internal
| faulty

⟨*assign_list*⟩ ::= ⟨*assign*⟩
| ⟨*assign_list*⟩ ‘`,`’ ⟨*assign*⟩

⟨*assign*⟩ ::= ID ‘`=`’ ⟨*expr*⟩

⟨*expr*⟩ ::= ⟨*disjunction*⟩

⟨*disjunction*⟩ ::= ⟨*conjunction*⟩
| ⟨*disjunction*⟩ ‘`||`’ ⟨*conjunction*⟩

⟨*conjuntion*⟩ ::= ⟨*comparation*⟩
| ⟨*conjuntion*⟩ ‘`&&`’ ⟨*comparation*⟩

⟨*comparation*⟩ ::= ⟨*factor*⟩
| ⟨*factor*⟩ ‘`==`’ ⟨*factor*⟩

⟨*factor*⟩ ::= ⟨*primary*⟩
| '!' ⟨*factor*⟩

⟨*primary*⟩ ::= true
| false
| ID
| '(' ⟨*expr*⟩ ')'

⟨*main_program*⟩ ::= main '(' ')' '{' ⟨*body_main*⟩ '}'

⟨*body_main*⟩ ::= ⟨*process_decl*⟩ ⟨*process_inv*⟩ ';'

⟨*process_decl*⟩ ::= ⟨*proc*⟩
| ⟨*process_decl*⟩ ';' ⟨*proc*⟩

⟨*proc*⟩ ::= ⟨*vbles_decl*⟩ ':' ⟨*type_proc*⟩

⟨*type_proc*⟩ ::= ID

⟨*process_inv*⟩ ::= ⟨*inv*⟩
| ⟨*process_inv*⟩ ';' ⟨*inv*⟩

⟨*inv*⟩ ::= run ID'(' ')'
| run ID '(' ⟨*ids*⟩ ')'

⟨*ids*⟩ ::= ID
| ⟨*ids*⟩ ',' ID