

CnC Runtime Model with Monotonic State

MODULE CNC-SYNTAX

Runtime initialization commands

SYNTAX $CollID ::= Int$

SYNTAX $InitCommand ::= (item-coll CollID) \mid (step-coll CollID)$

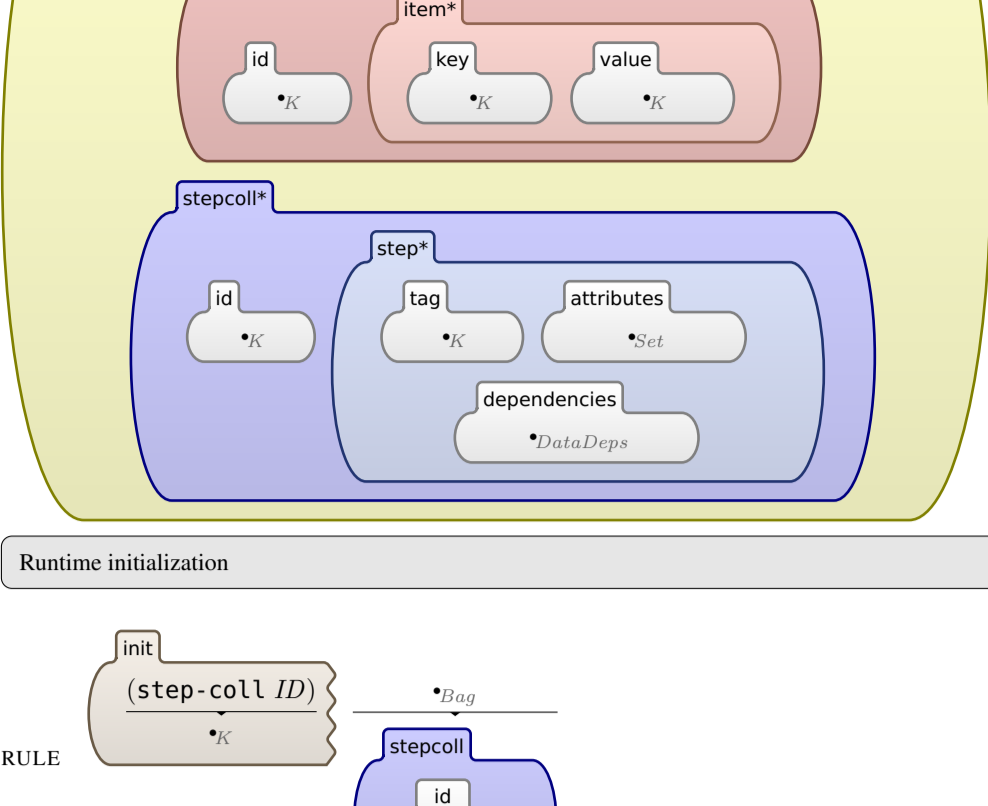
SYNTAX $InitSeq ::= List\{InitCommand, ""\}$

END MODULE

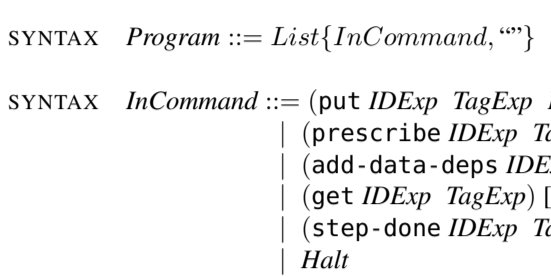
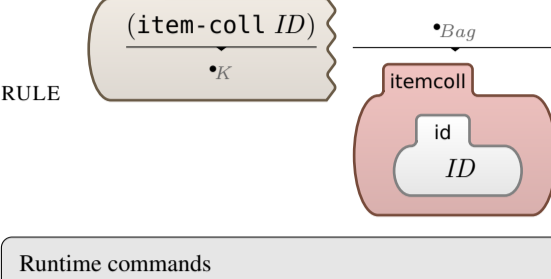
MODULE CNC

CnC runtime state representation

CONFIGURATION:



Runtime initialization



Runtime commands

SYNTAX $Program ::= List\{InCommand, ""\}$

SYNTAX $InCommand ::= (put IDExp TagExp DataExp) [seqstrict] \mid (prescribe IDExp TagExp) [seqstrict] \mid (add-data-deps IDExp TagExp DepsExp) [seqstrict] \mid (get IDExp TagExp) [seqstrict] \mid (step-done IDExp TagExp) [seqstrict] \mid Halt$

SYNTAX $OutCommand ::= (query-data-deps CollID Tag) \mid (run-step CollID Tag) \mid (give CollID Tag DataItem) \mid (finished) \mid (error-thrown)$

SYNTAX $ReadCmd ::= (read-cmd)$

SYNTAX $IDExp ::= CollID$

SYNTAX $DataItem ::= Int$

SYNTAX $DataExp ::= DataItem$

SYNTAX $DataItems ::= DataItems DataItem \mid DataItem$

SYNTAX $Tag ::= (DataItems)$

SYNTAX $TagExp ::= Tag$

SYNTAX $DepPair ::= IDExp TagExp [seqstrict]$

SYNTAX $DepPairExp ::= DepPair$

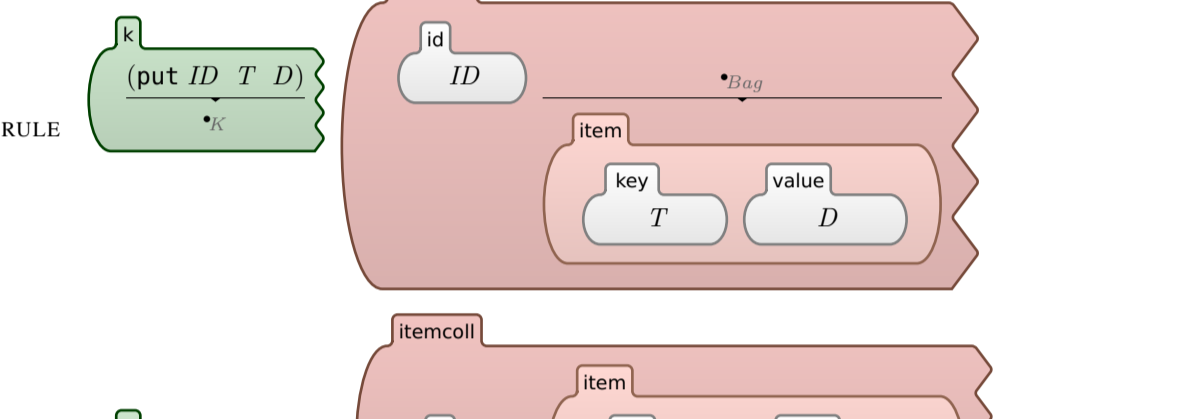
SYNTAX $DataDeps ::= List\{DepPairExp, ""\} [seqstrict]$

SYNTAX $DepsExp ::= DataDeps$

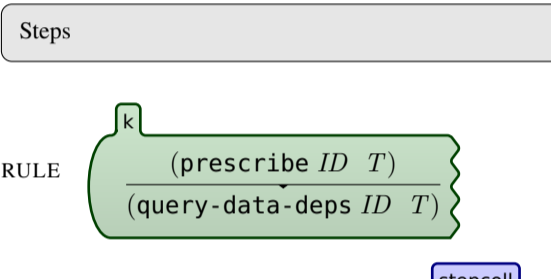
SYNTAX $Halt ::= halt \mid error-halt$

SYNTAX $Attribute ::= control-ready \mid data-ready \mid enabled \mid done$

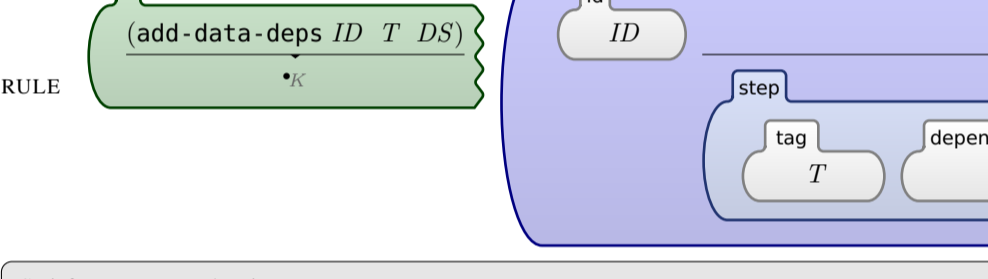
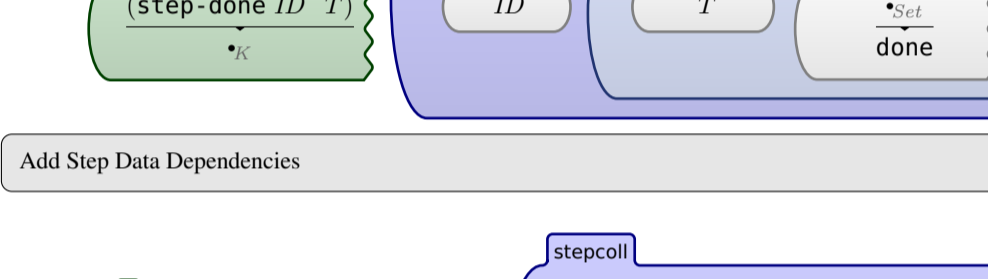
Dynamic single assignment



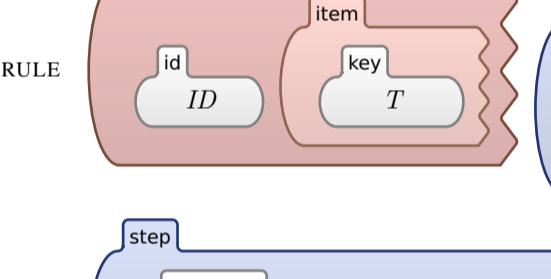
Read in calls to runtime



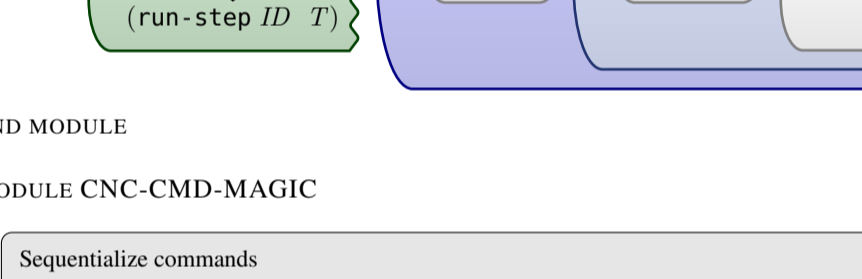
Data items



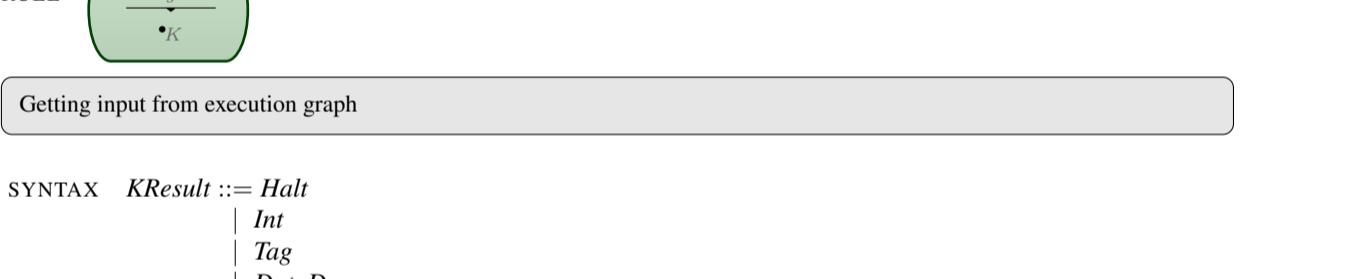
Steps



Satisfy Data Dependencies



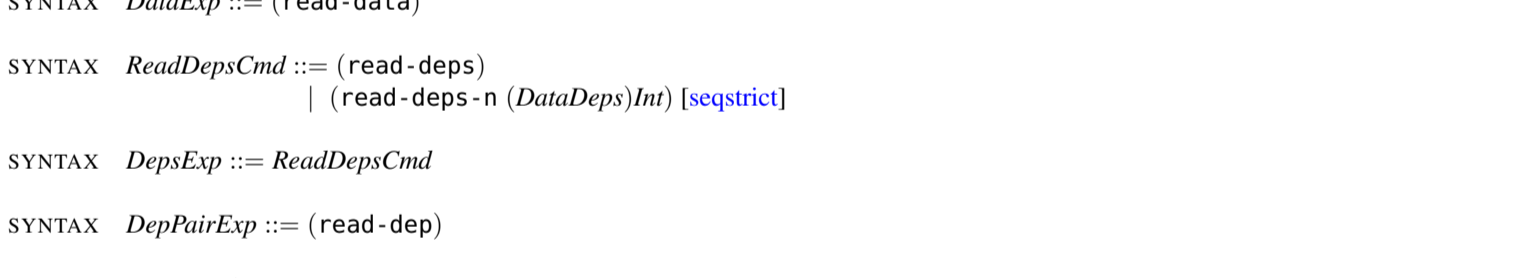
Run Step



END MODULE

MODULE CNC-CMD-MAGIC

Sequentialize commands



Getting input from execution graph

SYNTAX $KResult ::= Halt \mid Int \mid Tag \mid DataDeps \mid DepPairs$

SYNTAX $IDExp ::= (read-id)$

SYNTAX $TagExp ::= (read-tag) \mid (read-n Int) \mid (read-n Int Tag)$

SYNTAX $DataExp ::= (read-data)$

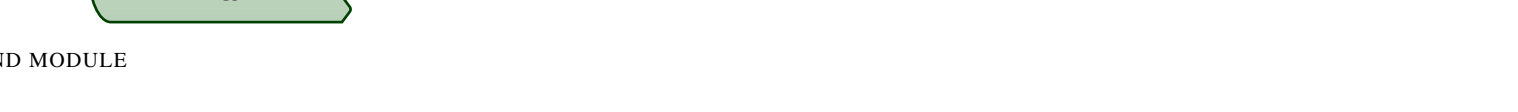
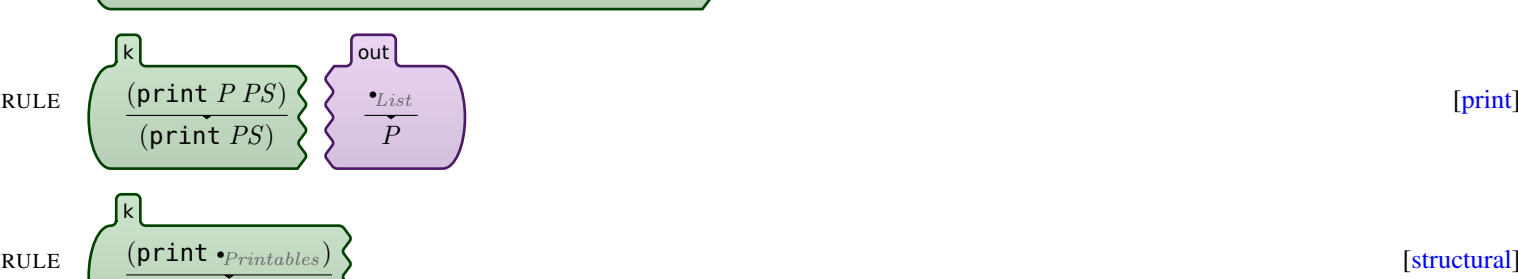
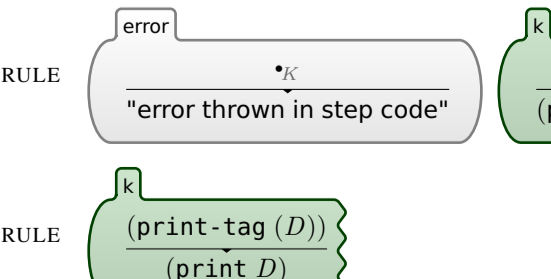
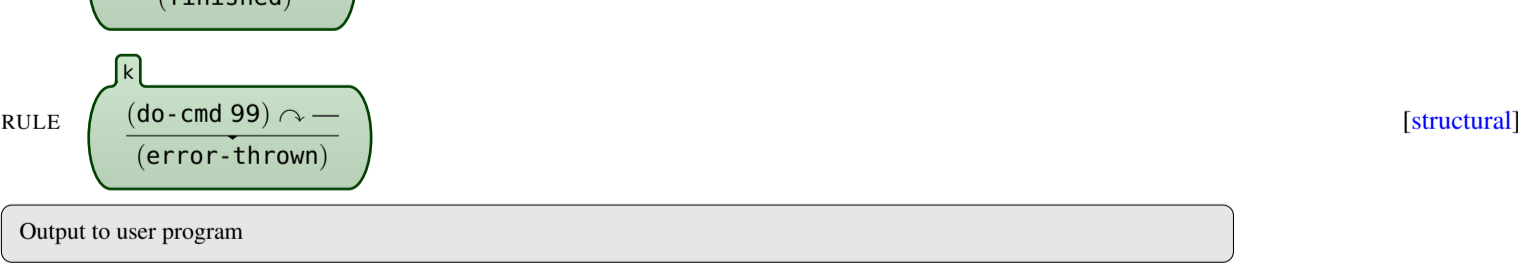
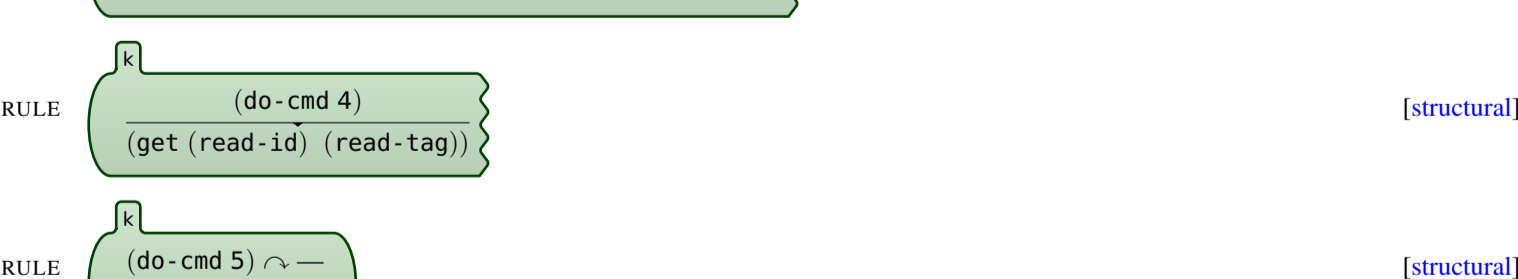
SYNTAX $ReadDepsCmd ::= (read-deps) \mid (read-deps-n (DataDeps) Int) [seqstrict]$

SYNTAX $DepsExp ::= ReadDepsCmd$

SYNTAX $DepPairExp ::= (read-dep)$

SYNTAX $CommandID ::= Int$

SYNTAX $InCommand ::= (do-cmd CommandID)$



Output to user program

SYNTAX $Printable ::= Tag \mid Int \mid String$

SYNTAX $Printables ::= List\{Printable, ""\}$

SYNTAX $WriteCommand ::= (print Printables) \mid (print-tag Tag)$

END MODULE