

CnC Checkpoint Model with Monotonic State

MODULE CNC-CHECKPOINT-SYNTAX

SYNTAX *InCommand* ::= (init)

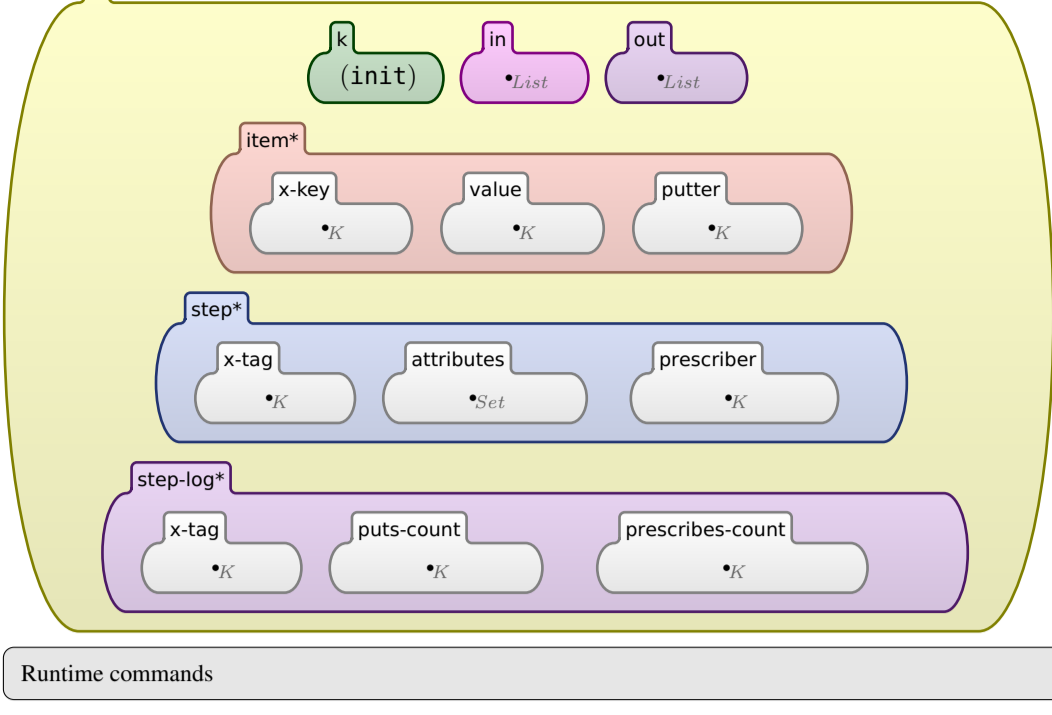
SYNTAX *Program* ::= *InCommand*

END MODULE

MODULE CNC-CHECKPOINT

CnC runtime state representation

CONFIGURATION:



Runtime commands

SYNTAX *CollID* ::= *Int*

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

SYNTAX *Program* ::= *List*{*InCommand*, ""}

SYNTAX *InCommand* ::= (init)
 | (put *DepExp DepExp DataExp*) [seqstrict]
 | (prescribe *DepExp DepExp*) [seqstrict]
 | (step-done *DepExp DataExp DataExp*) [seqstrict]
 | *Halt*

SYNTAX *DataItem* ::= *Int*

SYNTAX *Tag* ::= (*DataItems*)

SYNTAX *DataItems* ::= *DataItems DataItem*
 | *DataItem*

SYNTAX *IDExp* ::= *CollID*

SYNTAX *TagExp* ::= *Tag*

SYNTAX *DataExp* ::= *DataItem*

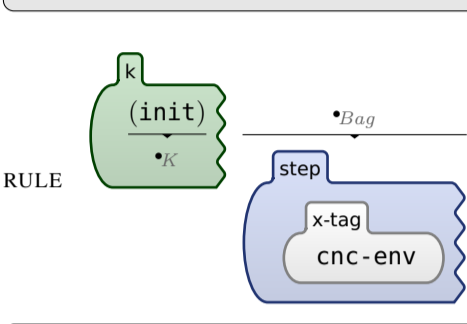
SYNTAX *EnvXTag* ::= cnc-env

SYNTAX *DepPair* ::= *EnvXTag*
 | *IDExp TagExp* [seqstrict]

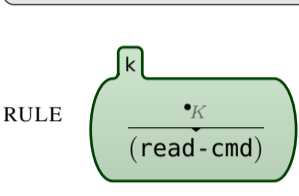
SYNTAX *Halt* ::= halt

SYNTAX *Attributes* ::= done

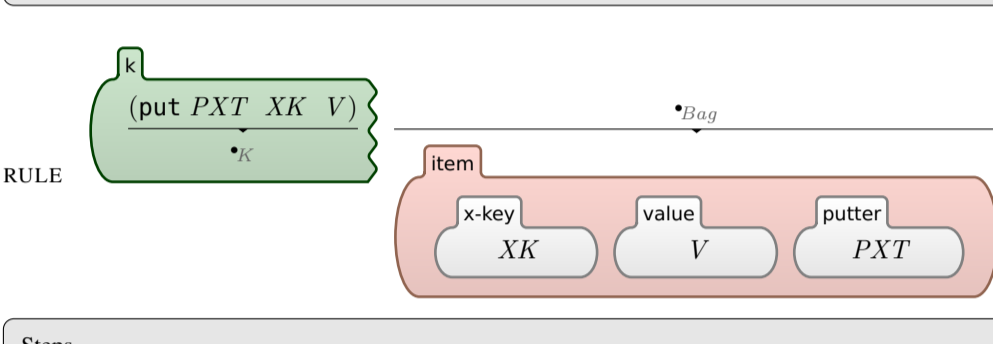
Init



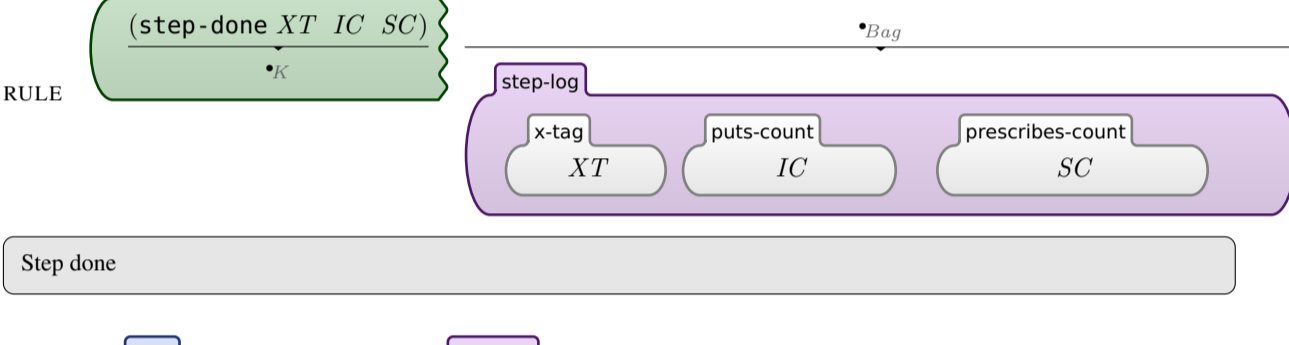
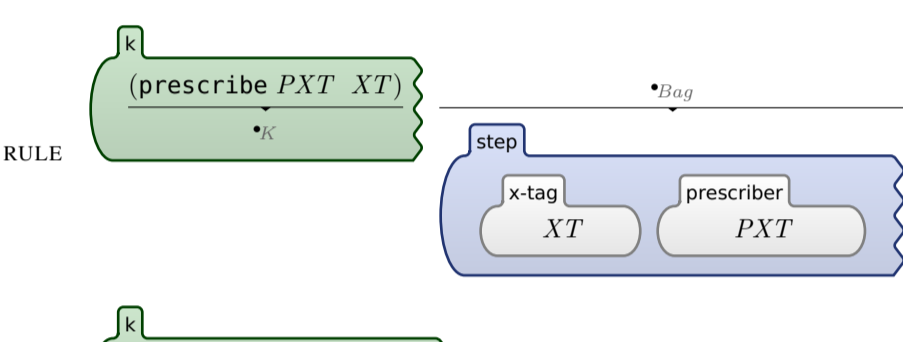
Read in calls to runtime



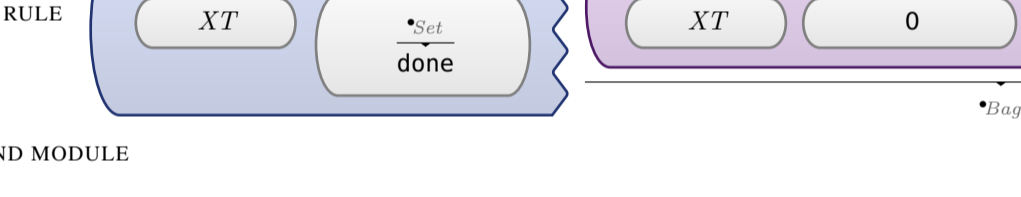
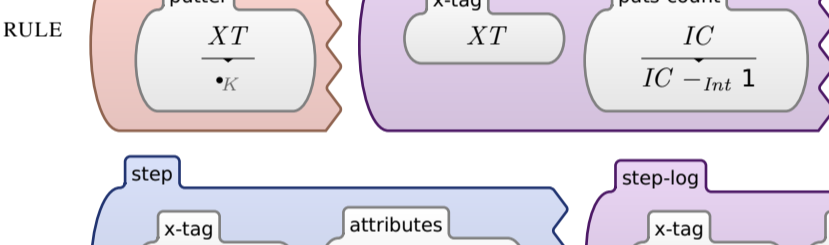
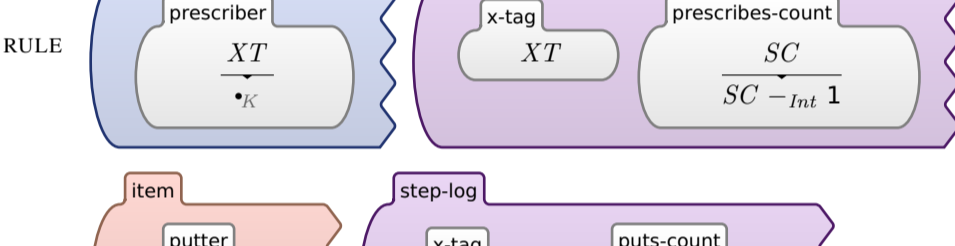
Data items



Steps



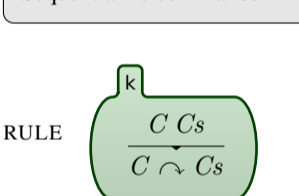
Step done



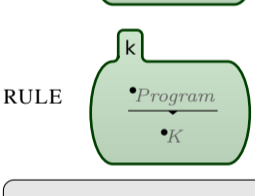
END MODULE

MODULE CNC-CHECKPOINT-CMD-MAGIC

Sequentialize commands



[structural]



Getting input from execution graph

SYNTAX *KResult* ::= *Halt*
 | *Int*
 | *Tag*
 | *DepPair*

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

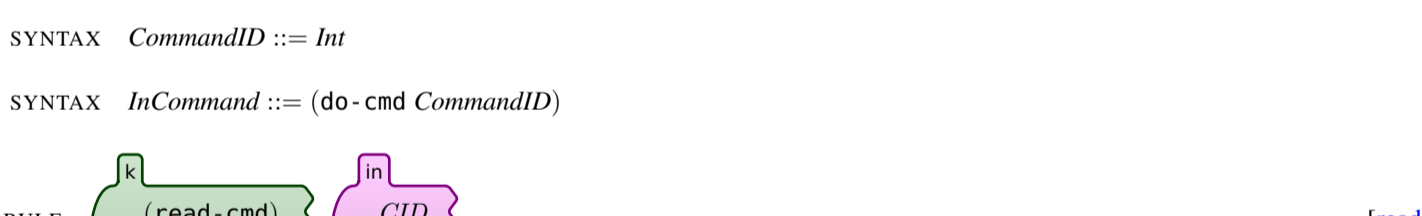
SYNTAX *TagExp* ::= (read-tag)
 | (read-n *Int*)
 | (read-n *Int Tag*)

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

SYNTAX *DepExp* ::= *DepPair*
 | (read-dep)

SYNTAX *CommandID* ::= *Int*

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



[read]



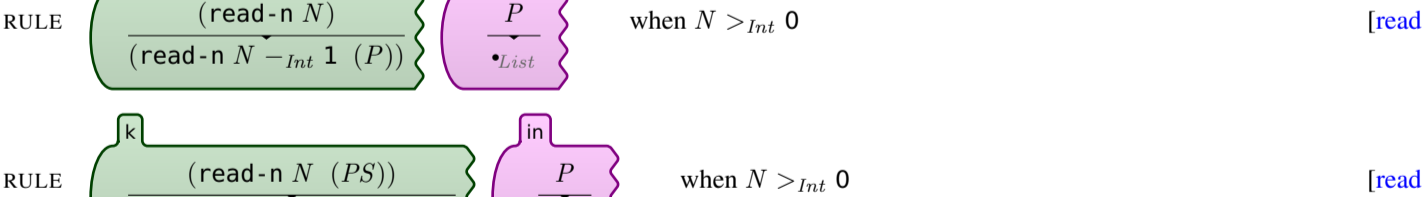
[read]



[read]



[read]



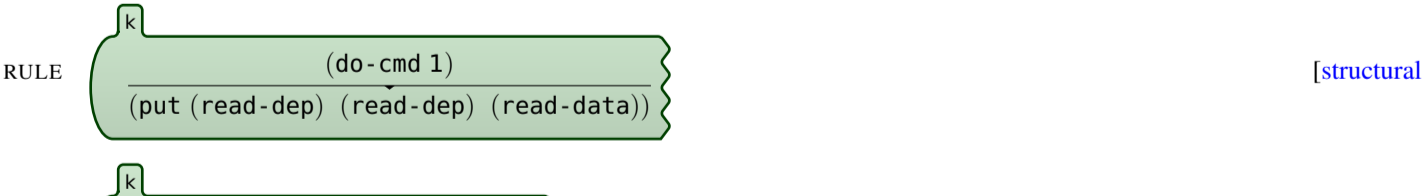
[read]



[read]



[structural]



[structural]



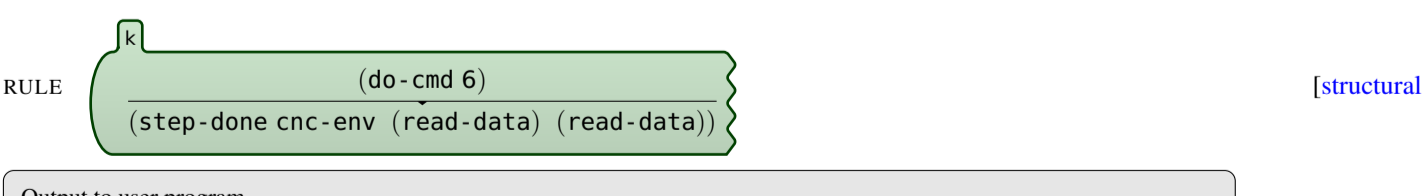
[structural]



[structural]



[structural]



[structural]



[structural]



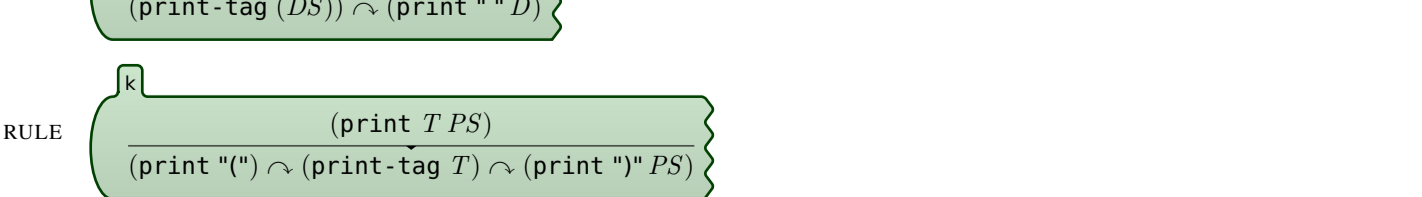
[structural]

Output to user program

SYNTAX *Printable* ::= *Tag*
 | *Int*
 | *String*

SYNTAX *Printables* ::= *List*{*Printable*, ""}

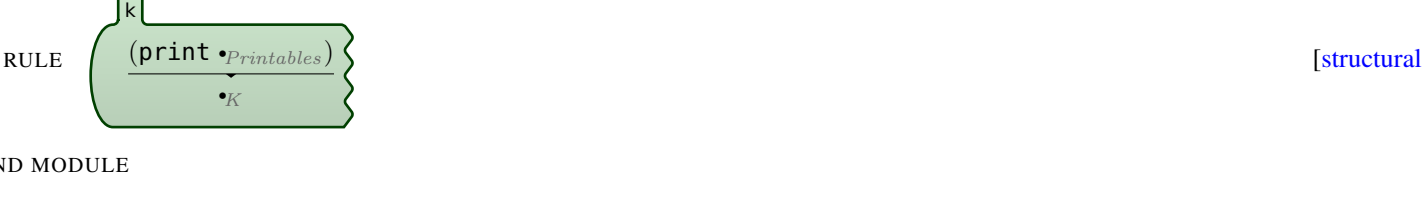
SYNTAX *WriteCommand* ::= (print *Printables*)
 | (print-tag *Tag*)



[structural]



[structural]



[print]

[structural]

END MODULE