Gramatica EBNF del lenguaje CPNlite que modela y ejecuta Colored Petri Nets

program → *declarations evaluations*

declarations → *declaration***;**{*declaration*}

declaration → **define** (**domain** *domain_def* | **trans** *transition_def* | **place** *place_def*
| arc *arc_def* | **init** *init_def*)

domain_def → *didentifier* = (*denum* | *dprod* | *dsetop*)

denum → **{** *dvalue* {,*dvalue*} **}**

dprod → *didentifier* **X** *didentifier*

dsetop → *didentifier setop didentifier*

setop → **U** | **∩** | **−**

dvalue → *string* | *integer*

string → *cap_letter*{*cap_letter*}

cap_letter → **A** | **B** | ...| **Z**

letter → **a** | **b** | ...| **z**

integer → *digit*{*digit*}

digit → **0** | **1** | ….| **9**

didentifier → *letter*{*digit* | *letter*}

transition_def → *tidentifier* [**guard** *guard_def*] {**,** *transition_def*}

tidentifier → *didentifier*

place_def → *pidentifier* **type** *didentifier*{**,** *place_def*}

pidentifier → *didentifier*

arc_def → (*input_arc* | *output_arc*) **var** *videntifier* [*arc_cond*]

input_arc → *pidentifier* **to** *tidentifier*

output_arc → *pidentifier* **from** *tidentifier*

videntifier → *didentifier*

arc_cond → *boolop extdvalue*

extdvalue → *exprvalue* | *exprvalue* **X** *exprvalue*

boolop → **=** | **!=** | **<** | **>**

exprvalue → *condexpr* | *mathexpr*

condexpr → **if (***var_cond***) then** *mathexpr* **else** *mathexpr*

mathexpr → *unary {mathop unary}*

mathop → **+** | **-** | ***** | **/** | **%**

unary → **(** **-** *primary* **)** | *primary*

primary → *integer* | *didentifier* | **(***mathexpr***)**

guard_def → *var_cond {relop guard_def}* | **(***guard_def***)**

var_cond → *videntifier arc_cond*

relop → **&&** | **||**

init_def → *place_assig {, place_assign}*

place_assign → *pidentifier* **= {***extdvalue{,extdvalue}***}**

evaluations → *evaluation***;***{evaluation}*

evaluation **->***list* | *query* | *fire*

list → *plist* | *tlist* | *alist* | *elist*

plist → **list places**

tlist → **list transitions**

alist → **list arcs**

elist → **list enabled**

query → *equery*

equery → **is_enabled** *tidentifier*

fire → *tfire* | *afire* | *ufire*

tfire → **fire** *tidentifier{,tidentifier}*

afire → **fire all** *[integer* **times***]*

ufire → **fire until** *trans_cond [* **limit** *integer]*

trans_cond → *tran_cond {relop tran_cond}*

tran_cond → *tidentifier [***not***]* **reach** *guard_def*

Ejemplo: Cena de filosofos

```
define domain phs = {0,1,2,3,4};

define place piensa type phs, come type phs, ten type phs;

define trans empieza guard i=p && d=(p+1)%5 , termina;

define arc piensa to empieza var p,
        ten to empieza var i,
        ten to empieza var d,
        come from empieza var p,
        come to termina var p,
        pienza from termina var p,
        ten from termina var i = p,
        ten from termina var d = (p+1)%5;

define init piensa={0,1,2,3,4}, ten={0,1,2,3,4};

list transitions;
```