Helena 3.0
Syntax summary

Sami Evangelista - (Sami [dot] Evangelista [at] lipn.univ-paris13 [dot] fr)

November 29, 2017

1 Net specification language

Nets

⟨*net*⟩ ::= ⟨*net name*⟩
['**(**' ⟨*net parameter list*⟩ '**)**']
'**{**' (⟨*definition*⟩)* '**}**'
⟨*net name*⟩ ::= ⟨*name*⟩
⟨*definition*⟩ ::= ⟨*type*⟩
| ⟨*constant*⟩
| ⟨*function*⟩
| ⟨*place*⟩
| ⟨*transition*⟩
| ⟨*state proposition*⟩

Net parameters

⟨*net parameter list*⟩ ::= ⟨*net parameter*⟩
| ⟨*net parameter*⟩ '**,**' ⟨*net parameter list*⟩
⟨*net parameter*⟩ ::= ⟨*net parameter name*⟩ '**:=**' ⟨*number*⟩
⟨*net parameter name*⟩ ::= ⟨*name*⟩

Types and subtypes

⟨*type*⟩ ::= ⟨*type name*⟩ '**:**' ⟨*type definition*⟩ '**;**'
| ⟨*subtype*⟩
⟨*type name*⟩ ::= ⟨*name*⟩
⟨*type definition*⟩ ::= ⟨*range type*⟩
| ⟨*modulo type*⟩
| ⟨*enumeration type*⟩
| ⟨*vector type*⟩
| ⟨*struct type*⟩
| ⟨*list type*⟩
| ⟨*set type*⟩

Range type

⟨*range type*⟩ ::= ⟨*range*⟩
⟨*range*⟩ ::= '**range**' ⟨*expression*⟩ '**..**' ⟨*expression*⟩

Modulo type

⟨*modular type*⟩ ::= '**mod**' ⟨*expression*⟩

Enumeration type

⟨*enumeration type*⟩ ::= '**enum**' '**(**' ⟨*enumeration constant*⟩ ('**,**' ⟨*enumeration constant*⟩)* '**)**'
⟨*enumeration constant*⟩ ::= ⟨*name*⟩

Vector type

⟨*vector type*⟩ ::= '**vector**' '**[**' ⟨*index type list*⟩ '**]**' '**of**' ⟨*type name*⟩
⟨*index type list*⟩ ::= ⟨*type name*⟩ ('**,**' ⟨*type name*⟩)*

Structured type

⟨*struct type*⟩ ::= '**struct**' '**{**' (⟨*component*⟩)+ '**}**'
⟨*component*⟩ ::= ⟨*type name*⟩ ⟨*component name*⟩ '**;**'
⟨*component name*⟩ ::= ⟨*name*⟩

List type

⟨*list type*⟩ ::= '**list**' '**[**' ⟨*type name*⟩ '**]**' '**of**' ⟨*type name*⟩ '**with**' '**capacity**' ⟨*expression*⟩

Set type

⟨*set type*⟩ ::= '**set**' '**of**' ⟨*type name*⟩ '**with**' '**capacity**' ⟨*expression*⟩

Subtype

⟨*subtype*⟩ ::= ⟨*subtype name*⟩ '**:**' ⟨*parent name*⟩ [⟨*constraint*⟩]
⟨*subtype name*⟩ ::= ⟨*type name*⟩
⟨*parent name*⟩ ::= ⟨*type name*⟩
⟨*constraint*⟩ ::= ⟨*range*⟩

Constants

⟨*constant*⟩ ::= '**constant**' ⟨*type name*⟩ ⟨*constant name*⟩ '**:=**' ⟨*expression*⟩ '**;**'
⟨*constant name*⟩ ::= ⟨*name*⟩

Functions

⟨*function*⟩ ::= ⟨*function declaration*⟩
| ⟨*function body*⟩
⟨*function prototype*⟩ ::= '**function**' ⟨*function name*⟩
'**(**' ⟨*parameters specification*⟩ '**)**' '**->**' ⟨*type name*⟩
⟨*function declaration*⟩ ::= ⟨*function prototype*⟩ '**;**'
⟨*function body*⟩ ::= '**import**' ⟨*function prototype*⟩ '**;**'
| ⟨*function prototype*⟩ ⟨*statement*⟩
⟨*parameters specification*⟩ ::= [⟨*parameter specification*⟩ ('**,**' ⟨*parameter specification*⟩)*]
⟨*parameter specification*⟩ ::= ⟨*type name*⟩ ⟨*parameter name*⟩
⟨*function name*⟩ ::= ⟨*name*⟩
⟨*parameter name*⟩ ::= ⟨*name*⟩

Statements

⟨*statement*⟩ ::= ⟨*assignment*⟩
| ⟨*if statement*⟩
| ⟨*case statement*⟩
| ⟨*while statement*⟩
| ⟨*for statement*⟩
| ⟨*return statement*⟩
| ⟨*assert statement*⟩
| ⟨*block*⟩

Assignment

⟨*assignment*⟩ ::= ⟨*variable*⟩ '**:=**' ⟨*expression*⟩ '**;**'

If-then-else

⟨*if statement*⟩ ::= '**if**' '**(**' ⟨*expression*⟩ '**)**' ⟨*true statement*⟩ ['**else**' ⟨*false statement*⟩]
⟨*true statement*⟩ ::= ⟨*statement*⟩
⟨*false statement*⟩ ::= ⟨*statement*⟩

Case

⟨*case statement*⟩ ::= '**case**' '**(**' ⟨*expression*⟩ '**)**' '**{**' (⟨*case alternative*⟩)* [⟨*default alternative*⟩] '**}**'
⟨*case alternative*⟩ ::= ⟨*expression*⟩ '**:**' ⟨*statement*⟩
⟨*default alternative*⟩ ::= '**default**' '**:**' ⟨*statement*⟩

While

⟨*while statement*⟩ ::= '**while**' '**(**' ⟨*expression*⟩ '**)**' ⟨*statement*⟩

For loop

⟨*for statement*⟩ ::= '**for**' '**(**' ⟨*iteration scheme*⟩ '**)**' ⟨*statement*⟩
⟨*iteration scheme*⟩ ::= ⟨*iteration variable*⟩ ['**,**' ⟨*iteration variable*⟩]
⟨*iteration variable*⟩ ::= ⟨*variable name*⟩ '**in**' ⟨*type name*⟩ [⟨*range*⟩]
| ⟨*variable name*⟩ '**in**' ⟨*place name*⟩
| ⟨*variable name*⟩ '**in**' ⟨*expression*⟩

Return

⟨*return statement*⟩ ::= '**return**' ⟨*expression*⟩ '**;**'

Assertion

⟨*assert statement*⟩ ::= '**assert**' '**:**' ⟨*expression*⟩ '**;**'

Block

⟨*block*⟩ ::= '**{**' (⟨*declaration*⟩)* (⟨*statement*⟩)+ '**}**'
⟨*declaration*⟩ ::= ⟨*constant declaration*⟩
| ⟨*variable declaration*⟩
⟨*variable declaration*⟩ ::= ⟨*type name*⟩ ⟨*variable name*⟩ ['**:=**' ⟨*expression*⟩] '**;**'
⟨*variable name*⟩ ::= ⟨*name*⟩

Places

⟨*place*⟩ ::= '**place**' ⟨*place name*⟩ '**{**' ⟨*place domain*⟩ (⟨*place attribute*⟩)* '**}**'
⟨*place attribute*⟩ ::= ⟨*initial marking*⟩
| ⟨*capacity*⟩
| ⟨*place type*⟩

Domain

⟨*domain*⟩ ::= '**dom**' '**:**' ⟨*domain definition*⟩ '**;**'
⟨*domain definition*⟩ ::= '**epsilon**'
| ⟨*types product*⟩
⟨*types product*⟩ ::= ⟨*type name*⟩ ('*****' ⟨*type name*⟩)*

Initial marking

⟨*initial marking*⟩ ::= '**init**' '**:**' ⟨*marking*⟩ '**;**'
⟨*marking*⟩ ::= ⟨*arc label*⟩

Capacity

⟨*capacity*⟩ ::= '**capacity**' '**:**' ⟨*expression*⟩ '**;**'

Type

⟨*place type*⟩ ::= '**type**' '**:**' ⟨*place type name*⟩ '**;**'
⟨*place type name*⟩ ::= '**process**'
| '**local**'
| '**shared**'
| '**protected**'
| '**buffer**'
| '**ack**'

Transitions

⟨*transition*⟩ ::= '**transition**' ⟨*transition name*⟩
'**{**' ⟨*transition inputs*⟩
⟨*transition outputs*⟩
[⟨*transition inhibitors*⟩]
[⟨*transition free variables*⟩]
[⟨*transition bound variables*⟩]
(⟨*transition attribute*⟩)* '**}**'

⟨*transition name*⟩ ::= ⟨*name*⟩

⟨*transition inputs*⟩ ::= '**in**' '**{**' (⟨*arc*⟩)* '**}**'

⟨*transition outputs*⟩ ::= '**out**' '**{**' (⟨*arc*⟩)* '**}**'

⟨*transition inhibitors*⟩ ::= '**inhibit**' '**{**' (⟨*arc*⟩)* '**}**'

⟨*transition attribute*⟩ ::= ⟨*transition guard*⟩
| ⟨*transition priority*⟩
| ⟨*transition description*⟩
| ⟨*safe*⟩

Arcs

⟨*arc*⟩ ::= ⟨*place name*⟩ '**:**' ⟨*arc label*⟩ '**;**'

Free variables

⟨*transition free variables*⟩ ::= '**pick**' '**{**' (⟨*free variable*⟩)* '**}**'

⟨*free variable*⟩ ::= ⟨*free variable name*⟩ '**in**' ⟨*free variable domain*⟩

⟨*free variable domain*⟩ ::= ⟨*type name*⟩ [⟨*range*⟩]
| ⟨*expression*⟩

Bound variables

⟨*transition bound variables*⟩ ::= '**let**' '**{**' (⟨*transition bound variable*⟩)* '**}**'

⟨*transition bound variable*⟩ ::= ⟨*type name*⟩ ⟨*variable name*⟩ '**:=**' ⟨*expression*⟩ '**;**'

Guard

⟨*transition guard*⟩ ::= '**guard**' '**:**' ⟨*guard definition*⟩ '**;**'

⟨*guard definition*⟩ ::= ⟨*expression*⟩

Priority

⟨*transition priority*⟩ ::= '**priority**' '**:**' ⟨*expression*⟩ '**;**'

Safe attribute

⟨*safe*⟩ ::= '**safe**' '**;**'

Description

⟨*transition description*⟩ ::= '**description**' '**:**' ⟨*string*⟩ ['**,**' ⟨*non empty expression list*⟩] '**;**'

Expressions

⟨*expression*⟩ ::= '**(**' ⟨*expression*⟩ '**)**' | ⟨*numerical constant*⟩
| ⟨*enumeration constant*⟩ | ⟨*variable*⟩
| ⟨*predecessor-successor operation*⟩ | ⟨*integer operation*⟩
| ⟨*comparison operation*⟩ | ⟨*boolean operation*⟩
| ⟨*function call*⟩ | ⟨*cast*⟩
| ⟨*if-then-else*⟩ | ⟨*structure*⟩
| ⟨*structure component*⟩ | ⟨*structure assignment*⟩
| ⟨*vector*⟩ | ⟨*vector component*⟩
| ⟨*vector assignment*⟩ | ⟨*empty list*⟩
| ⟨*list*⟩ | ⟨*list component*⟩
| ⟨*list assignment*⟩ | ⟨*list slice*⟩
| ⟨*list concatenation*⟩ | ⟨*list membership*⟩
| ⟨*empty set*⟩ | ⟨*set*⟩
| ⟨*set membership*⟩ | ⟨*set operation*⟩
| ⟨*token component*⟩ | ⟨*attribute*⟩
| ⟨*iterator*⟩

⟨*expression list*⟩ ::= ε
| ⟨*non empty expression list*⟩

⟨*non empty expression list*⟩ ::= ⟨*expression*⟩ ('**,**' ⟨*expression*⟩)*

Numerical and enumeration constants

⟨*numerical constant*⟩ ::= ⟨*number*⟩
⟨*enumeration constant*⟩ ::= ⟨*name*⟩

Predecessor and successor operators

⟨*predecessor-successor operation*⟩ ::= '**pred**' ⟨*expression*⟩
| '**succ**' ⟨*expression*⟩

Integer arithmetic

⟨*integer operation*⟩ ::= ⟨*expression*⟩ '**+**' ⟨*expression*⟩
| ⟨*expression*⟩ '**−**' ⟨*expression*⟩
| ⟨*expression*⟩ '*****' ⟨*expression*⟩
| ⟨*expression*⟩ '**/**' ⟨*expression*⟩
| ⟨*expression*⟩ '**%**' ⟨*expression*⟩
| '**+**' ⟨*expression*⟩
| '**−**' ⟨*expression*⟩

Comparison operators

⟨*comparison operation*⟩ ::= ⟨*expression*⟩ '**=**' ⟨*expression*⟩
| ⟨*expression*⟩ '**!=**' ⟨*expression*⟩
| ⟨*expression*⟩ '**>**' ⟨*expression*⟩
| ⟨*expression*⟩ '**>=**' ⟨*expression*⟩
| ⟨*expression*⟩ '**<**' ⟨*expression*⟩
| ⟨*expression*⟩ '**<=**' ⟨*expression*⟩

Boolean logic

⟨*boolean operation*⟩ ::= ⟨*expression*⟩ '**or**' ⟨*expression*⟩
| ⟨*expression*⟩ '**and**' ⟨*expression*⟩
| '**not**' ⟨*expression*⟩

Variables

⟨*variable*⟩ ::= ⟨*variable name*⟩
| ⟨*structure component*⟩
| ⟨*vector component*⟩
| ⟨*list component*⟩

Structures

⟨*structure*⟩ ::= '`{`' ⟨*non empty expression list*⟩ '`}`'
⟨*structure component*⟩ ::= ⟨*variable*⟩ '`.`' ⟨*component name*⟩
⟨*structure assignment*⟩ ::= ⟨*expression*⟩ '`::`' '`(`' ⟨*component name*⟩ '`:=`' ⟨*expression*⟩ '`)`'

Vectors

⟨*vector*⟩ ::= '`[`' ⟨*non empty expression list*⟩ '`]`'
⟨*vector component*⟩ ::= ⟨*variable*⟩ '`[`' ⟨*non empty expression list*⟩ '`]`'
⟨*vector assignment*⟩ ::= ⟨*expression*⟩ '`::`' '`(`' '`[`' ⟨*non empty expression list*⟩ '`]`' '`:=`' ⟨*expression*⟩ '`)`'

Lists

⟨*empty list*⟩ ::= '`empty`'
⟨*list*⟩ ::= '`|`' ⟨*non empty expression list*⟩ '`|`'
⟨*list component*⟩ ::= ⟨*variable*⟩ '`[`' ⟨*expression*⟩ '`]`'
⟨*list assignment*⟩ ::= ⟨*expression*⟩ '`::`' '`(`' '`[`' ⟨*expression*⟩ '`]`' '`:=`' ⟨*expression*⟩ '`)`'
⟨*list slice*⟩ ::= ⟨*expression*⟩ '`[`' ⟨*expression*⟩ '`..`' ⟨*expression*⟩ '`]`'
⟨*list concatenation*⟩ ::= ⟨*expression*⟩ '`&`' ⟨*expression*⟩
⟨*list membership*⟩ ::= ⟨*expression*⟩ '`in`' ⟨*expression*⟩

Sets

⟨*empty set*⟩ ::= '`empty`'
⟨*set*⟩ ::= '`|`' ⟨*non empty expression list*⟩ '`|`'
⟨*set membership*⟩ ::= ⟨*expression*⟩ '`in`' ⟨*expression*⟩
⟨*set operation*⟩ ::= ⟨*expression*⟩ '`or`' ⟨*expression*⟩
| ⟨*expression*⟩ '`and`' ⟨*expression*⟩
| ⟨*expression*⟩ '`-`' ⟨*expression*⟩

Function call

⟨*function call*⟩ ::= ⟨*function name*⟩ '`(`' ⟨*expression list*⟩ '`)`'

Cast

⟨*cast*⟩ ::= ⟨*type name*⟩ '`(`' ⟨*expression*⟩ '`)`'

If-then-else

⟨*if-then-else*⟩ ::= ⟨*condition*⟩ '`?`' ⟨*true expression*⟩ '`:`' ⟨*false expression*⟩
⟨*condition*⟩ ::= ⟨*expression*⟩
⟨*true expression*⟩ ::= ⟨*expression*⟩
⟨*false expression*⟩ ::= ⟨*expression*⟩

Token component

⟨*token component*⟩ ::= ⟨*token*⟩ '`->`' ⟨*component number*⟩
⟨*token*⟩ ::= ⟨*variable name*⟩
⟨*component number*⟩ ::= ⟨*number*⟩

Attributes

⟨*attribute*⟩ ::= ⟨*type name*⟩ '`'`' ⟨*type attribute*⟩
| ⟨*place name*⟩ '`'`' ⟨*place attribute*⟩
| ⟨*expression*⟩ '`'`' ⟨*container attribute*⟩
| ⟨*expression*⟩ '`'`' ⟨*list attribute*⟩

⟨*type attribute*⟩ ::= '`first`' | '`last`' | '`card`'
⟨*place attribute*⟩ ::= '`card`' | '`mult`'
⟨*container attribute*⟩ ::= '`full`' | '`empty`' | '`capacity`'
| '`size`' | '`space`'
⟨*list attribute*⟩ ::= '`first`' | '`first_index`' | '`prefix`'
| '`last`' | '`last_index`' | '`suffix`'

Iterator

⟨*iterator*⟩ ::= ⟨*iterator type*⟩ '**(**' ⟨*iteration scheme*⟩ [⟨*iterator condition*⟩] [⟨*iterator expression*⟩] '**)**'
⟨*iterator type*⟩ ::= '**forall**' | '**exists**' | '**card**' | '**mult**' | '**min**' | '**max**' | '**sum**' | '**product**'
⟨*iterator condition*⟩ ::= '**|**' ⟨*expression*⟩
⟨*iterator expression*⟩ ::= '**:**' ⟨*expression*⟩

Arc labels

⟨*arc label*⟩ ::= ⟨*complex tuple*⟩ ('**+**' ⟨*complex tuple*⟩)*

Tuples

⟨*complex tuple*⟩ ::= [⟨*tuple for*⟩] [⟨*tuple guard*⟩] [⟨*tuple factor*⟩] ⟨*tuple*⟩
⟨*tuple for*⟩ ::= '**for**' '**(**' ⟨*iteration scheme*⟩ '**)**'
⟨*tuple guard*⟩ ::= '**if**' '**(**' ⟨*expression*⟩ '**)**'
⟨*tuple factor*⟩ ::= ⟨*expression*⟩ '***'
⟨*tuple*⟩ ::= '**<(**' ⟨*non empty expression list*⟩ '**)>**'
| '**epsilon**'

State propositions

⟨*state proposition*⟩ ::= '**proposition**' ⟨*state proposition name*⟩ '**:**' ⟨*expression*⟩ '**;**'
⟨*state proposition name*⟩ ::= ⟨*name*⟩

2 Property specification language

Properties

⟨*property specification*⟩ ::= (⟨*property*⟩)*
⟨*property*⟩ ::= ⟨*state property*⟩
| ⟨*temporal property*⟩

State properties

⟨*state property*⟩ ::= '**state**' '**property**' ⟨*property name*⟩ '**:**' ⟨*state property definition*⟩
⟨*property name*⟩ ::= ⟨*name*⟩
⟨*state property definition*⟩ ::= ⟨*reject clause*⟩ (⟨*accept clause*⟩)*
⟨*reject clause*⟩ ::= '**reject**' ⟨*predicate*⟩ '**;**'
⟨*accept clause*⟩ ::= '**accept**' ⟨*predicate*⟩ '**;**'
⟨*predicate*⟩ ::= '**deadlock**'
| ⟨*state proposition name*⟩

Temporal properties

⟨*temporal property*⟩ ::= '**ltl**' '**property**' ⟨*property name*⟩ '**:**' ⟨*temporal expression*⟩ '**;**'
⟨*temporal expression*⟩ ::= '**(**' ⟨*temporal expression*⟩ '**)**'
| '**true**'
| '**false**'
| ⟨*state proposition name*⟩
| '**not**' ⟨*temporal expression*⟩
| ⟨*temporal expression*⟩ '**or**' ⟨*temporal expression*⟩
| ⟨*temporal expression*⟩ '**and**' ⟨*temporal expression*⟩
| '**[]**' ⟨*temporal expression*⟩
| '**<>**' ⟨*temporal expression*⟩
| ⟨*temporal expression*⟩ '**until**' ⟨*temporal expression*⟩