

Atelier B

LOGIC-SOLVER

Syntaxe

version 1.2



ATELIER B
LOGIC-SOLVER Syntaxe
version 1.2

Document établi par CLEARSY.

Ce document est la propriété de CLEARSY et ne doit pas être copié, reproduit, dupliqué
totalement ou partiellement sans autorisation écrite.

Tous les noms des produits cités sont des marques déposées par leurs auteurs respectifs.

CLEARSY
Maintenance ATELIER B
Parc de la Duranne
320 avenue Archimède
Les Pléiades III - Bât.A
13857 Aix-en-Provence Cedex 3
France

Tél 33 (0)4 42 37 12 99
Fax 33 (0)4 42 37 12 71
email : maintenance.atelierb@clearsy.com

Table des matières

1	Langage de théorie	1
2	Langage de formules	3
3	Catégories lexicales	5
4	Caractères	7
5	Symboles réservés	9
6	Priorité, associativité des INFIXES	11
7	Symboles définis par l'utilisateur	13
8	Langage de tactique	17
9	Formule quantifiée	19

Chapitre 1

Langage de théorie

Catégorie Syntaxique	Définition
SYSTÈME	THÉORIE SYSTÈME & THÉORIE
THÉORIE	THEORY <i>nom_de_théorie</i> END THEORY <i>identificateur</i> IS CONTENU END
CONTENU	RÈGLE CONTENU ; RÈGLE
RÈGLE	CONSÉQUENT ANTÉCÉDENT => CONSÉQUENT
ANTÉCÉDENT	FORMULE ANTÉCÉDENT & FORMULE
CONSÉQUENT	FORMULE FORMULE == FORMULE

Avant de le rentrer dans le **Logic-Solver**, le code source d'un SYSTÈME est passé au travers du pré-processeur **cpp**, qui assure les fonctions suivantes :

- Suppression des commentaires introduits au moyen du parenthésage `/* */`.
- Inclusion de fichiers introduits au moyen de la commande `#include`.
- Élaboration des macros introduites au moyen de la commande `#define`.

Chapitre 2

Langage de formules

Catégorie Syntaxique	Définition
FORMULE	ATOMIQUE OPÉRATOIRE PROGRAMMÉE PARENTHESÉE FONCTIONELLE
ATOMIQUE	<i>joker</i> <i>nombre</i> <i>grand_nombre</i> <i>identificateur</i> <i>chaîne</i> <i>autre</i> <i>spécial</i>
OPÉRATOIRE	<i>opérateur_préfixe</i> FORMULE FORMULE <i>opérateur_infixe</i> FORMULE FORMULE <i>opérateur_postfixe</i>
PROGRAMMÉE	<i>mot_clé_ouvrant</i> PEIGNE <i>mot_clé_fermant</i>
PARENTHESÉE	(FORMULE) [FORMULE] { FORMULE }
FONCTIONELLE	FORMULE (FORMULE) FORMULE [FORMULE] [FORMULE] FORMULE
PEIGNE	FORMULE PEIGNE <i>mot_clé_infixe</i> FORMULE

Chapitre 3

Catégories lexicales

Catégorie Lexicale	Définition
<i>joker</i>	<i>LETTRE</i>
<i>nombre</i>	Suite de <i>CHIFFRES</i> formant un nombre naturel compris entre 0 et 2147483647.
<i>grand_nombre</i>	Suite de <i>CHIFFRES</i> formant un nombre naturel supérieur à 2147483647.
<i>identificateur</i>	Suite, de longueur au moins égale à 2, composée de <i>LETTRES</i> , de <i>CHIFFRES</i> , de <i>_</i> , et ayant au moins une <i>LETTRE</i> .
<i>chaine</i>	Suite de caractères situés entre " ". On peut trouver le caractère " dans la suite, à condition qu'il soit précédé du caractère \ .
<i>autre</i>	Suite composée de <i>LETTRES</i> , de <i>CHIFFRES</i> , de <i>_</i> , ou de <i>?</i> , et différente des précédentes.
<i>spécial</i>	La suite <i><></i> , ou la suite <i>{}</i> .
<i>opérateur_préfixe</i>	<i>PRÉFIXE</i> , ou <i>PRÉFIXE_RÉSERVÉ</i> .
<i>opérateur_infixe</i>	La suite <i>[]</i> , ou une suite d' <i>INFIXES</i> , ou <i>INFIXE_RÉSERVÉ</i> .
<i>opérateur_postfixe</i>	<i>POSTFIXE</i>

Remarque : le *LOGIC SOLVER* permet également d'attacher une caractéristique spéciale à un opérateur infixe, celle de *définisseur de label*. Si un opérateur *op* définit des labels à gauche, alors toute expression de la forme *f1 op f2* telle que *f1* soit un identificateur est traitée de manière spéciale : *f1* n'est pas considéré comme une occurrence libre de

l'identificateur. Il est considéré comme un label sans rapport avec d'éventuelles autres occurrences de *f1*, ce qui influe sur le calcul de substitutions et l'évaluation de certaines gardes. La caractéristique de définisseur de label est attachée à un opérateur en utilisant une table de symboles utilisateur.

Chapitre 4

Caractères

Caractères	Définition
<i>LETTRE</i>	a b c d e f g h i j k l m n o p q r s t u v w x y z A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
<i>CHIFFRE</i>	0 1 2 3 4 5 6 7 8 9
<i>PRÉFIXE</i>	! @ # %
<i>INFIXE</i>	& ; \$. / * , : ^ - + = \ < > ’
<i>POSTFIXE</i>	~

Chapitre 5

Symboles réservés

PRÉFIXE RÉSERVÉ

band bappend bcall bcall1 bcall2 bcatl bclean bcompile bconnect
bcrel bcrelr bcrer bctrule bdef1 bdef2 bdump bflat bfork bfresh
bftac bget bgetd bgetallhyp bgethyp bgetresult bgoal bguard bident
bidentb binhyp blemma blen blenf blent blident blood blvar bmark
bmask bmatch bmodr bnewv bnot bnum bpattern bpop bprintf bproved
breade breadf bresult brev brule bsearch bshell brecompact bresetcomp
bresult bsparemem bstatistics bstring bsubfrm btac btest bunify
bunmask bunproved bUpident bvr bwritef bwritem

INFIXE RÉSERVÉ

=> <= >= /= <=> := == blank2 evl bnmap bnlmap bsmap bsmap
spe

ATOMIQUE RÉSERVÉ

ARI CATL DED DEF FLAT GEN HYPEQL HYP LMAP MAP MODR NEWV RES REV
RULE SHELL SUB TACTIC WRITE bclose bfalse bhalt blank0? brecompact
bresetcomp bstatistics ghost?

Chapitre 6

Priorité, associativité des INFIXES

<i>INFIXE</i>	Priorité	Associativité	<i>INFIXE</i>	Priorité	Associativité
blank2	0	gauche	;	-7	gauche
evl	11	gauche	\$	11	gauche
spe	10	droite	.	10	droite
bnmap	0	gauche	/	3	gauche
bnlamp	0	gauche	*	3	gauche
bsmap	0	gauche	,	-1	gauche
bslmap	0	gauche	:	-4	gauche
=>	-6	gauche		-8	gauche
<=	0	gauche	^	0	gauche
>=	0	gauche	-	2	gauche
/=	0	gauche	+	2	gauche
<=>	-4	gauche	=	-4	gauche
:=	-2	gauche	\	0	gauche
==	-4	gauche	<	0	gauche
&	-5	gauche	>	0	gauche
'	5	gauche			

Chapitre 7

Symboles définis par l'utilisateur

L'utilisateur peut définir ses propres symboles qui viennent donc s'ajouter aux symboles pré-définis. Ces symboles sont déclarés dans un (ou plusieurs) fichier(s), annoncé dans le texte source d'un SYSTÈME au moyen du symbole `'`, situé en début de ligne, et immédiatement suivi par le nom du fichier en question. Les nouveaux symboles rentrent en vigueur après cette ligne. À noter que l'on ne peut pas redéfinir les symboles qui existent déjà. Chaque ligne du fichier de symboles se présente de la façon suivante :

symbole ARITÉ ÉDITION PRIORITÉ

L'ARITÉ peut prendre les valeurs suivantes :

ARITÉ	Commentaires
bin	<i>opérateur_infixe</i> <i>mot_clé_infixe</i>
unr	<i>opérateur_préfixe</i>
atm	<i>symbole_atomique</i>
opn	<i>mot_clé_ouvrant</i>

L'ÉDITION définit la façon dont on peut disposer des blancs ou des sauts de lignes de part et d'autre du symbole lors d'une édition. Elle peut prendre l'une des valeurs suivantes :

ÉDITION	Commentaires
void	symbole non imprimé
nrml	symbole imprimé tel que
blkb	symbole précédé d'un blanc
blka	symbole suivi d'un blanc
blba	symbole précédé et suivi d'un blanc
term	terminateur imprimé tel que
blbt	terminateur précédé d'un blanc
blat	terminateur suivi d'un blanc
bbat	terminateur précédé et suivi d'un blanc
keyw	mot_clé suivi d'un blanc
keyl	mot_clé suivi d'une ligne

La **PRIORITÉ** est un nombre entier.

Les lignes d'une table de symboles utilisateur peuvent aussi avoir la forme suivante :

label *op* x
ou bien

x *op* label

Il faut alors que *op* soit un opérateur binaire prédéfini ou défini avant dans la table. Ces commandes attachent une caractéristique de *définisseur de label* à l'opérateur *op*, le

label étant à gauche dans le premier cas et à droite dans le second. Un opérateur ne peut pas définir des label à la fois à gauche et à droite.

Si un opérateur *op* définit des labels à gauche, alors toute expression de la forme *f1 op f2* telle que *f1* soit un identificateur est traitée de manière spéciale : *f1* n'est pas considéré comme une occurrence libre de l'identificateur. Il est considéré comme un label sans rapport avec d'éventuelles autres occurrences de *f1*, ce qui influe sur le calcul de substitutions et l'évaluation de la garde de non liberté, des gardes **bfresh** et **blvar**, ainsi que l'opération GEN.

exemple :

Si **op** définit des labels à gauche, alors la formule

`[yy := 5] (yy * (yy op xx))`

est réécrite de la manière suivante :

`(5 * (yy op xx))`

car *yy* est considéré comme un label dans la deuxième partie de l'expression.

Chapitre 8

Langage de tactique

Catégorie Syntaxique	Définition
TACTIQUE	TACTIQUE ; TACTIQUE (TACTIQUE) ~ <i>Identificateur</i>

Chapitre 9

Formule quantifiée

Catégorie Syntaxique	Définition
Q_FORMULE	<i>Opérateur-préfixe</i> VARIABLE . FORMULE <i>Opérateur-préfixe</i> (VARIABLE) . FORMULE { VARIABLE FORMULE }
VARIABLE	VARIABLE_SIMPLE VARIABLE , VARIABLE_SIMPLE
VARIABLE_SIMPLE	<i>Identificateur</i> <i>Joker</i> Identificateur\$ <i>Index</i> Joker\$ <i>Index</i>
<i>Index</i>	Suite de <i>CHIFFRES</i> formant un nombre compris entre 0 et 9999