

# INFO-F-403 : Language theory and compiling

## Rapport projet partie 1

Simon Picard  
Arnaud Rosette

16 octobre 2014

### 1 Expressions régulières

Afin de décrire les mots (tokens) acceptés par le langage Iulius, nous utilisons les expressions régulières étendues.

Voici le tableau qui reprend l'ensemble des tokens du langage Iulius. Chaque ligne de celui-ci est composée du nom du token, le type dans l'enum `LexicalUnit` qui lui est associé et de l'expression régulière qui correspond à ce token.

Nom	Type	Expression régulière
Comment	/	#.*(\\r)?\\n
Circumflex (^)	POWER	\^
Percentage (%)	REMAINDER	%
Tilde (~)	BITWISE_NOT	~
Pipe ( )	BITWISE_OR	\
Dollar (\$)	BITWISE_XOR	\\$
Dubble greater (»)	ARITHMETIC_SHIFT_RIGHT	>>
Dubble lower («)	ARITHMETIC_SHIFT_LEFT	<<
Dubble equal (==)	EQUALITY	==
Exclamation equal (!=)	INEQUALITY	!=
Function	FUNCTION	function
Return	RETURN	return
Arrow right (->)	MAP_TO	->
Question mark (?)	TERNARY_IF	\?
Exclamation mark (!)	NEGATION	!
Colon ( : )	TERNARY_ELSE	:
Dubble ampersand (&&)	LAZY_AND	&&
Dubble pipe (  )	LAZY_OR	\ \
While	WHILE	while
For	FOR	for
Semicolon (;)	END_OF_INSTRUCTION	;
Println	PRINTLN	println
Const	CONST	const
Let	LET	let
Dubble colon ( :: )	TYPE_DEFINITION	::
Boolean (type)	BOOLEAN_TYPE	Bool
Real (type)	REAL_TYPE	FloatingPoint
Integer (type)	INTEGER_TYPE	Integer
Integer (cast)	INTEGER_CAST	int
Real (cast)	REAL_CAST	float
Read integer	READ_INTEGER	readint
Read real	READ_REAL	readfloat
Boolean (cast)	BOOLEAN_CAST	bool