

**INFO-F-403**

**Projet : Compilateur Perl**

**Auteurs :** Rodriguez Paul  
Vaccari Eric



# 1 Unités lexicales

## 1.1 Tableau

Nom	Regex
var	<code>[\$[a-zA-Z_][a-zA-Z0-9_]*</code>
identifier	<code>[a-zA-Z_][a-zA-Z0-9_]*</code>
integer	<code>[0-9]+</code>
float	<code>{integer}\.{integer}</code>
string	<code>'[^']*'</code>
space	<code>[\t\n ]</code>
comment	<code>#.*\n</code>
lbrace	<code>\{</code>
rbrace	<code>\}</code>
lpar	<code>\(</code>
rpar	<code>\)</code>
semicolon	<code>;</code>
call_mark	<code>&amp;</code>
plus	<code>\+</code>
minus	<code>\-</code>
times	<code>\*</code>
divide	<code>\/</code>
not	<code>!</code>
notletters	<code>not</code>
lazy_and	<code>&amp;&amp;</code>
lazy_or	<code>  </code>
equals	<code>==</code>
eq	<code>eq</code>
different	<code>!=</code>
ne	<code>ne</code>
lower	<code>&lt;</code>
lt	<code>lt</code>
greater	<code>&gt;</code>
gt	<code>gt</code>
lower_equals	<code>&lt;=</code>
le	<code>le</code>
greater_equals	<code>&gt;=</code>
ge	<code>ge</code>
comma	<code>,</code>
concat_mark	<code>\.</code>
assign_mark	<code>=</code>

Nom	Regex
sub	<code>sub</code>
if	<code>if</code>
else	<code>else</code>
elsif	<code>elsif</code>
unless	<code>unless</code>
return	<code>return</code>
defined	<code>defined</code>
int	<code>int</code>
length	<code>length</code>
print	<code>print</code>
scalar	<code>scalar</code>
substr	<code>substr</code>

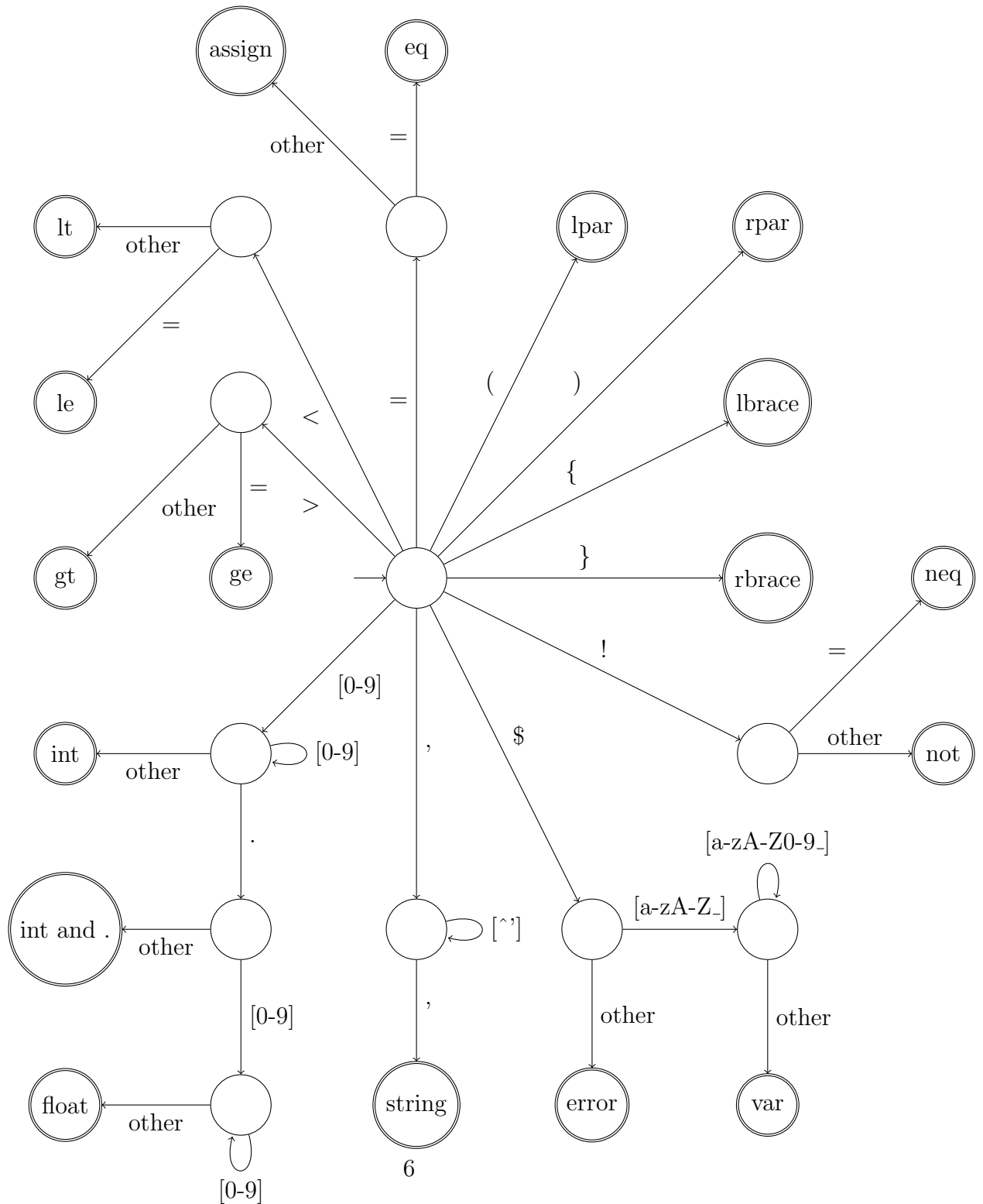
## 1.2 Remarques

La syntaxe complète de Perl concernant les noms de variables est beaucoup plus compliquée mais concerne des fonctionnalités (packages) hors du cadre de ce projet, ce pourquoi nous nous sommes limités aux règles les plus simples.



## 2 Graphe du DFA

### 2.1 Variables, comparateurs, blocs, littéraux



## 2.2 Remarques

Le graphe a été scindé pour améliorer la clarté de l'ensemble. Certains tokens sont identifiables dès que leur dernier caractère a été lu (par exemple les accolades), d'autres nécessitent la lecture du caractère suivant le dernier (par exemple, pour terminer un entier il faut lire autre chose qu'un chiffre). Dans ce deuxième cas, après avoir identifié le token la lecture du dernier caractère est annulée, il servira comme premier caractère du token suivant.