

# WST'2004 Competition/Exhibition of termination tools: format of input files

C. Marché and A. Rubio

March 12, 2004

## 1 Syntax of TRS input files

$spec ::= ( decl ) spec \mid \varepsilon$   
 $decl ::= VAR idlist \mid THEORY listofthdecl \mid RULES listofrules$   
 $\quad \mid STRATEGY strategydecl \mid id anylist$   
 $anylist ::= \varepsilon \mid id anylist \mid string anylist$   
 $\quad \mid ( anylist ) anylist \mid , anylist$   
 $idlist ::= \varepsilon \mid id idlist$   
 $listofthdecl ::= \varepsilon \mid ( thdecl ) listofthdecl$   
 $thdecl ::= id idlist \mid EQUATIONS eqlist$   
 $eqlist ::= \varepsilon \mid equation eqlist$   
 $equation ::= term == term$   
 $listofrules ::= \varepsilon \mid rule listofrules$   
 $rule ::= term \rightarrow term \mid term \rightarrow term \mid condlist$   
 $condlist ::= cond \mid cond , condlist$   
 $cond ::= term \rightarrow term \mid term \rightarrow <- term$   
 $term ::= id \mid id ( ) \mid id ( termlist )$   
 $termlist ::= term \mid term , termlist$   
 $strategydecl ::= INNERMOST \mid CONTEXTSENSITIVE csstratlist$   
 $csstratlist ::= \varepsilon \mid ( id intlist ) csstratlist$   
 $intlist ::= \varepsilon \mid int intlist$

$id$  are non-empty sequences of characters except space,  $'(, ')$ ,  $'"'$  and  $' , '$ ; and excluding special sequences  $'\rightarrow'$ ,  $'=='$ ,  $'\rightarrow<-'$ ,  $'|'$  and keywords CONTEXTSENSITIVE, EQUATIONS, INNERMOST, RULES, STRATEGY, THEORY and VAR.  
 $string$  are sequences of any characters between double quotes  
 $int$  are non-empty sequences of digits

## 2 Semantical conditions

- a symbol declared in a VAR section must not have been used in previous declarations, and is assumed to denote a variable in remaining declarations (in particular must not be applied to arguments)
- a symbol occurring in a RULES section which has not been used before is assumed to denote a function symbol, and must be used afterwards always with the same arity.

## 3 Syntax of SRS input files

```
spec ::= ( decl ) spec | ε  
decl ::= RULES listofrules | STRATEGY strategydecl | id anylist  
anylist ::= ε | id anylist | string anylist | ( anylist ) anylist | , anylist  
listofrules ::= ε | rule , listofrules  
rule ::= word -> word | word ->  
word ::= id | id word  
strategydecl ::= LEFTMOST | RIGHTMOST
```

*id* are non-empty sequences of characters except space, '(', ')', '"', and ', ' ; and excluding special sequence '->', and keywords LEFTMOST, RIGHTMOST, RULES, and STRATEGY.

*string* are sequences of any characters between double quotes