

Libreverse - parserGenerator

Stephen Torri
torrisa@auburn.edu

December 9, 2005

[*CPU*, *Table*, *Token*, *Rule*, *NIL*, *Bit*]

InternalData

$cpu : CPU$
 $tableList : \mathbb{P} Table$
 $tokenList : \mathbb{P} Token$
 $ruleList : \mathbb{P} Rule$

INIT

$cpu = NIL$
 $tableList = \emptyset$
 $tokenList = \emptyset$
 $ruleList = \emptyset$

setCPU

Δcpu
 $newCPU? : CPU$
 $newCPU? \neq cpu$
 $cpu' = newCPU?$

addTable

$\Delta(tableList)$
 $table? : Table$
 $table? \notin tableList$
 $tableList' = tableList \cup \{table?\}$

addToken

$\Delta(tokenList)$
 $token? : Token$
 $token? \notin tokenList$
 $tokenList' = tokenList \cup \{token?\}$

addRule

$\Delta(ruleList)$
 $rule? : Rule$
 $rule? \notin ruleList$
 $ruleList' = ruleList \cup \{rule?\}$

<i>CPU</i>	
<i>registers</i> : \mathbb{P} <i>Register</i>	
<i>INIT</i>	
<i>registers</i> = \emptyset	
<i>addRegister</i>	
<i>reg?</i> : <i>Register</i>	
<i>reg?</i> \notin <i>registers</i>	
<i>registers'</i> = <i>registers</i> \cup { <i>reg?</i> }	

Register

$bits : \text{seq } Bit$
 $indexes : String \rightarrow \mathbb{N}$
 $maxLength : \mathbb{N}$
 $name : String$
 $endian : Endian$

$\#bits == maxLength$
 $endian ::= little \mid big$

INIT

$name? : String$
 $index? : \mathbb{N}$

$bits = \text{seq } Bits$
 $index? > 0$
 $indexes = name? \rightarrow index?$
 $maxLength = index?$
 $endian = little$

addSubIndex

$\Delta indexes$
 $name? : String$
 $index? : \mathbb{N}$

$name? \notin \text{dom } indexes$
 $0 \leq index? \leq maxLength$
 $indexes' = indexes \cup \{name? \rightarrow index?\}$

getBitSequence

$name? : String$
 $result! : \text{seq } Bits$

$name? \in \text{dom } indexes$
 $((endian = little) \wedge (result! = \text{reverseSubSeq}(bits, indexes(name?)))) \vee$
 $((endian = big) \wedge (result! = \text{SubSeq}(bits, indexes(name?))))$
 $\#result! = indexes(name?)$

$\text{reverseSubSeq} : (\text{seq } Bits \times \mathbb{N}) \rightarrow \text{seq } Bits$