

Languages-beta: OC-L-03-Names *

The P_LanCompS Project

OC-L-03-Names.cbs | PLAIN | PRETTY

OUTLINE

3 Names

- Naming objects
 - Infix operator precedence
 - Referring to named objects
-

Language "OCaml Light"

3 Names

Naming objects

```
Syntax VN : value-name ::= lowercase-ident
                        | '(' operator-name ')'
ON : operator-name ::= prefix-op | infix-op
PO : prefix-op ::= prefix-symbol

// infix-op ::= infix-symbol
//           | '*' | '+' | '-' | '-.' | '=' | '<' | '>' | '||' | '&' | '&&'
//           | '!=' | 'or' | ':=' | 'mod'
//           | 'land' | 'lor' | 'lxor' | 'lsl' | 'lsr' | 'asr'
```

*Suggestions for improvement: plancomps@gmail.com.
Reports of issues: <https://github.com/plancomps/CBS-beta/issues>.

Infix operator precedence

Syntax $IO : \text{infix-op} ::= \text{infix-op-1} \mid \text{infix-op-2} \mid \text{infix-op-3} \mid \text{infix-op-4}$
 $\mid \text{infix-op-5} \mid \text{infix-op-6} \mid \text{infix-op-7} \mid \text{infix-op-8}$

Lexis $IO-1 : \text{infix-op-1} ::= '**' \text{ operator-char}^* \mid 'lsl' \mid 'lsr' \mid 'asr'$
 $IO-2 : \text{infix-op-2} ::= '*'$
 $\mid '*' \text{ operator-char-not-asterisk operator-char}^*$
 $\mid ('/' \mid '\%') \text{ operator-char}^*$
 $\mid 'mod' \mid 'land' \mid 'lor' \mid 'lxor'$
 $IO-3 : \text{infix-op-3} ::= ('+' \mid '-') \text{ operator-char}^*$
 $IO-4 : \text{infix-op-4} ::= ('@' \mid '^') \text{ operator-char}^*$
 $IO-5 : \text{infix-op-5} ::= ('=' \mid '<' \mid '>' \mid '\$') \text{ operator-char}^*$
 $\mid '|' \text{ (operator-char-not-bar operator-char}^*)^?$
 $\mid '||' \text{ operator-char}^+$
 $\mid '&' \text{ operator-char-not-ampersand operator-char}^*$
 $\mid '&&' \text{ operator-char}^+$
 $\mid '!='$
 $IO-6 : \text{infix-op-6} ::= '&' \mid '&&'$
 $IO-7 : \text{infix-op-7} ::= 'or' \mid '||'$
 $IO-8 : \text{infix-op-8} ::= ':'$

Lexis $CN : \text{constr-name} ::= \text{capitalized-ident}$
 $TCN : \text{typeconstr-name} ::= \text{lowercase-ident}$
 $FN : \text{field-name} ::= \text{lowercase-ident}$
 $MN : \text{module-name} ::= \text{capitalized-ident}$

Referring to named objects

Syntax $VP : \text{value-path} ::= \text{value-name}$
 $CSTR : \text{constr} ::= \text{constr-name}$
 $TCSTR : \text{typeconstr} ::= \text{typeconstr-name}$
 $F : \text{field} ::= \text{field-name}$

Semantics $\text{value-name}[_ : \text{value-path}] : \Rightarrow \text{ids}$
Rule $\text{value-name}[LI] = "LI"$
Rule $\text{value-name}['(' PS ')'] = \text{string-append}("(" , "PS" , ")")$
Rule $\text{value-name}['(' IO-1 ')'] = \text{string-append}("(" , "IO-1" , ")")$
Rule $\text{value-name}['(' IO-2 ')'] = \text{string-append}("(" , "IO-2" , ")")$
Rule $\text{value-name}['(' IO-3 ')'] = \text{string-append}("(" , "IO-3" , ")")$
Rule $\text{value-name}['(' IO-4 ')'] = \text{string-append}("(" , "IO-4" , ")")$
Rule $\text{value-name}['(' IO-5 ')'] = \text{string-append}("(" , "IO-5" , ")")$
Rule $\text{value-name}['(' IO-6 ')'] = \text{string-append}("(" , "IO-6" , ")")$
Rule $\text{value-name}['(' IO-7 ')'] = \text{string-append}("(" , "IO-7" , ")")$
Rule $\text{value-name}['(' IO-8 ')'] = \text{string-append}("(" , "IO-8" , ")")$

Semantics $\text{constr-name}[_ : \text{constr}] : \Rightarrow \text{ids}$
Rule $\text{constr-name}[CN] = "CN"$

Semantics $\text{typeconstr-name}[_ : \text{typeconstr}] : \Rightarrow \text{ids}$

Rule $\text{typeconstr-name}[TCN] = "TCN"$

Semantics $\text{field-name}[_ : \text{field}] : \Rightarrow \text{ids}$

Rule $\text{field-name}[FN] = "FN"$