

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 $IO-1 : \text{infix-op-1} ::= \text{'**'} \text{ operator-char}^* \mid \text{'lsl'} \mid \text{'lsr'} \mid \text{'asr'}$
 $IO-2 : \text{infix-op-2} ::= \text{'*'} \mid \text{'*'} \text{ operator-char-not-asterisk operator-char}^*$
 $\mid \text{'/'} \mid \text{'\%'}$ operator-char*
 $\mid \text{'mod'} \mid \text{'land'} \mid \text{'lor'} \mid \text{'lxor'}$

$IO-3 : \text{infix-op-3} ::= \text{'+'} \mid \text{'-'}$ operator-char*

$IO-4 : \text{infix-op-4} ::= \text{'@'} \mid \text{'^'}$ operator-char*

$IO-5 : \text{infix-op-5} ::= \text{'='} \mid \text{'<'} \mid \text{'>'} \mid \text{'\$'}$ operator-char*
 $\mid \text{'|'} \text{ (operator-char-not-bar operator-char}^*\text{)}^?$
 $\mid \text{'||'}$ operator-char⁺
 $\mid \text{'\&'}$ operator-char-not-ampersand operator-char*
 $\mid \text{'\&\&'}$ operator-char⁺
 $\mid \text{'!='}$

$IO-6 : \text{infix-op-6} ::= \text{'\&'}$ \mid '\&\&'

$IO-7 : \text{infix-op-7} ::= \text{'or'}$ \mid '||'

$IO-8 : \text{infix-op-8} ::= \text{':='}$

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

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

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

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