

## INDEX FOR PART II

### A

<actuals> 94  
*any* 144  
*append* 138  
*array* 96  
ARRAY 113  
*array* 113, 114, 116, 118, 138  
<array-expression> 112, 138  
<array-initialization> 112, 123  
ASSERT 111  
<assert-expression> 110  
*assignment* 124  
<association> 88, 89  
<associations> 88, 89  
*a\_class* 76, 89, 90  
*a\_composite\_instance* 76  
*a\_integer* 74, 94, 98, 99, 106, 113, 114, 116,  
124, 126  
*a\_proc* 76, 87  
*a\_string* 75, 94, 98, 99  
*a\_undefined* 75, 85, 86, 87, 89, 90, 96, 98,  
99, 100, 101, 102, 106, 108, 111,  
112, 113, 116, 122, 123

### B

*bal* 144  
basic values (*BASIC-INSTANCE*) 74  
BEGIN 89, 90, 96, 98, 118, 119, 120  
*bind* 77, 84, 85, 86, 87, 89, 90, 96, 98, 110

*binding* 77, 84, 96, 98, 106, 118, 119, 122,  
123, 124, 126, 127  
*bits* 149  
<block> 55, 79, 85, 95, 101, 105, 111  
*break* 145

### C

CASE 103, 104  
<case-entry> 103, 104, 127  
<case-expression> 103, 104  
*center* 134  
CLASS 53, 76, 89, 90, 96, 98, 118, 119, 120  
<class-declaration> 52, 74, 88, 89, 95, 117,  
124  
*class\_decl* 76, 98, 99, 118, 119, 120, 128  
*close* 148  
comment 70  
comment symbol 70  
*compl* 149  
*COMPOSITE-INSTANCE* 53, 75  
*conj* 149  
CONST 86  
<constant> 94, 126  
<constant-declaration> 86, 111, 126  
<constant-expression> 86, 103, 126, 127  
<constant-initialization> 86, 127  
<constant-operator> 126  
*contents* 77, 128  
*copy* 150

**D**

*declarations* 83  
 DEFAULT 103, 104  
*delete* 139  
 <delimiter> 70  
 DENOTABLE-VALUE 74  
 denotational semantics 50, 71  
*dereference* 96, 100, 103, 104, 106, 110, 113,  
 114, 116, 118, 119, 121, 122, 123,  
 124, 126, 128  
*disj* 149  
 DO 105, 106  
 <dyadic-expression> 123, 124  
 <dyadic-operator> 92, 123, 124

**E**

ELIF 101, 102  
 ELSE 101, 102  
 END 89, 90, 96, 98, 118, 119, 120  
*env* 76  
 ENV 84, 85, 86, 87, 89, 90, 96, 98, 106,  
 108, 110, 111, 112, 118, 119, 122,  
 123, 124, 126, 127  
 ENVglobal 84, 96, 98  
 ENVIRONMENT 53, 77  
 environment 71, 77  
*equal* 128  
 ERROR 84, 86, 89, 90, 96, 98, 99, 103, 104,  
 106, 111, 112, 118, 119, 122, 123,  
 124, 126, 127  
 ESAC 103, 104  
 escape sequence 71  
*eval* 84, 86, 96, 98, 100, 101, 102, 103, 104,  
 105, 106, 108, 110, 111, 112, 113,  
 114, 116, 118, 121, 122, 123, 124,  
 126  
 evaluation process 54, 71, 78, 79  
*eval\_array\_init* 113, 114  
*eval\_call* 94, 96, 98, 110, 114, 116, 118, 119,  
 122, 123, 124, 126  
*eval\_field\_selection* 96, 106, 114, 116, 118,  
 119, 121, 122, 123, 124, 126  
*eval\_standard\_procedure* 96, 98, 99  
*eval\_table\_init* 116

*expand\_super\_class* 89, 90  
 <expression> 84, 93, 95, 101  
*extend* 77, 84, 85, 86, 96, 98

**F**

F 96, 98, 100, 101, 105, 108, 111, 112, 122,  
 123, 124, 126  
 FAILS 101  
 FETCH 118, 119, 120  
 <fetch-associations> 88, 89, 117  
 FI 101, 102  
 field selection 117, 123  
 <field-identifier> 88, 89, 117  
*file* 147  
*find* 145  
 FOR 106, 110  
 <for-expression> 105, 106, 135, 139, 141,  
 143  
 <formals> 92, 95  
 FR 96, 98, 100, 101, 102, 105, 108, 111,  
 112, 122, 123  
 FRETURN 100

**G**

*get* 148, 150

**H**

*has* 75, 119, 120  
*has\_binding* 77, 96, 122, 123, 124, 126  
*has\_field* 119, 120, 122, 123, 124, 126

**I**

<identifier> 52, 70, 74, 94, 95  
 <identifier-or-call> 94  
 IF 101, 102  
 <if-expression> 101  
 IN 106

*index* 96, 98, 111, 112, 114, 116, 118, 124,  
126, 135, 139, 142, 149  
INIT 89, 90, 96, 98, 113, 114, 116, 118, 119,  
120  
instance 88  
instance (creation of) 96  
*integer* 70  
**INTEGER** 74  
*integer* 130  
<integer-constant> 70, 94, 131  
*interval* 135, 141, 149  
*intval* 74, 98, 99, 128  
*is\_class* 77, 96, 98  
*is\_composite\_instance* 76  
*is\_instance* 96  
*is\_integer* 74, 98, 99, 127, 128  
*is\_loc* 77, 96, 106, 118, 119, 124, 126, 128  
*is\_proc* 76, 84, 96, 98, 122, 123, 124, 126  
*is\_string* 75, 98, 99, 110, 127, 128  
*is\_undefined* 75, 98, 99

**K**

keywords 70

**L**

*last* 139  
layout symbols 70  
*left* 135  
lexical units 70  
*lit* 145  
**LOCATION** 74

**M**

metalanguage 72  
*modify* 77, 106, 118, 119, 124, 126  
<monadic-expression> 122  
<monadic-operator> 92, 122  
*move* 146  
multiple assignment 123

**N**

*N* 84, 85, 86, 87, 89, 90, 94, 96, 98, 100,  
101, 102, 103, 104, 105, 106, 108,  
110, 111, 112, 113, 114, 116, 118,  
119, 121, 122, 123, 124, 126  
*names* 78, 84, 96, 98  
*name\_copy* 78, 96, 98, 110, 111, 112  
*new\_inner\_scope* 78, 96, 98, 110, 111, 112  
*new\_proc\_scope* 78, 84, 96, 98  
*next* 105, 135, 139, 141, 142, 149  
non-printable characters 71  
*NR* 96, 98, 100, 101, 102, 105, 108, 111,  
112, 122, 123

**O**

OD 105, 106  
OF 103, 104  
OP 87  
*operation* 75, 118, 119  
<operator-declaration> 87, 92, 122, 124  
<operator-symbol> 70, 71, 92, 122, 123  
<operator-symbol-declaration> 92

**P**

<parenthesized-expression> 111  
parse expression 78  
*partial\_state\_copy* 78, 108  
*pos* 146  
priority (of operators) 123  
<primary> 122  
PROC 87, 96, 98  
**PROCEDURE** 53, 76  
<procedure-declaration> 52, 74, 87, 95  
PROGRAM 84  
<program-declaration> 84  
*put* 148, 150

**R**

*real* 70, 132  
 <real-constant> 70, 94, 133  
*repl* 135  
*replace* 136  
*require\_constant\_expression* 86, 103, 104,  
 127  
*retrieve* 121, 136, 140, 143, 149  
 RETURN 100  
 <return-expression> 56, 81, 99, 108  
*reverse* 136  
*right* 136  
 ROF 110  
*rpos* 146  
*rtab* 147

**S**

*same\_as* 76, 128  
 SCAN 110  
 <scan-expression> 109  
*scan\_string* 109, 110, 143  
 scope 83  
 <select> 117, 123  
*self* 95, 96, 98  
 semantic domain 52, 74  
 <simple-constant-expression> 126  
 single quote character 71  
*size* 94, 96, 98, 99, 111, 112, 113, 116, 118,  
 119, 137, 140, 143, 149  
 <size-and-default> 112, 115  
*sort* 140  
*span* 147  
 STATE 74, 84, 85, 86, 96, 98, 106, 118,  
 119, 124, 126, 128  
*stop* 150  
 STORABLE-VALUE 74  
 STORE 118, 119, 120  
 <store-associations> 88, 89, 117  
*string* 71  
 STRING 75  
*string* 134  
 <string-constant> 70, 71, 94, 134  
*stringval* 75, 98, 99, 128  
*string\_equal* 89, 90, 118, 119, 120, 122, 123,

124, 126, 128, 129  
 <subclass-declaration> 88, 89  
*subject* 95, 96, 99, 109, 110  
 <subscript> 120, 123  
 subscription 120, 123  
*substr* 137  
*substring* 94, 129  
 SUCCEEDS 101  
 <summer-program> 83  
 syntax notation 69

**T**

*tab* 147  
*table* 115  
 TABLE 116  
*table* 142  
 <table-element> 115  
 <table-expression> 115  
 <table-initialization> 115  
 <test> 101, 104, 105  
*text* 96, 98, 118, 119, 120  
 THEN 101, 102  
 TRY 108  
 <try-expression> 107, 108, 148  
*type* 96, 98, 99, 150

**U**

UNDEFINED 75  
 <unit> 123  
 UNTIL 108  
*update* 114, 117, 121, 123, 141, 143, 149

**V**

VAR 85, 86  
 <variable-declaration> 83, 85, 111  
 <variable-initialization> 55, 79, 83, 85  
*varinit* 79, 83, 85  
 Vienna Definition Language 50, 71

**W**

WHILE 105  
<while-expression> 104, 105

**Y**

YRT 108