## RE(Regular Expression)

WIDTH = [1-9][0-9]\*PRECISION = \.[0-9]\*

ASC = [^%]\*

% = %[\-0# +]\*{WIDTH}?{PRECISION}?%

 $X = %([\-#]*|[0#]*)?{WIDTH}?{PRECISION}?X$ 

 $c = %-*{WIDTH}?c$ 

 $d = %([\-]*|[\-+]*|[0]*|[0+]*)?{WIDTH}?{PRECISION}?d$ 

 $i = %([\-]*|[\-+]*|[0]*|[0+]*)?{WIDTH}?{PRECISION}?i$ 

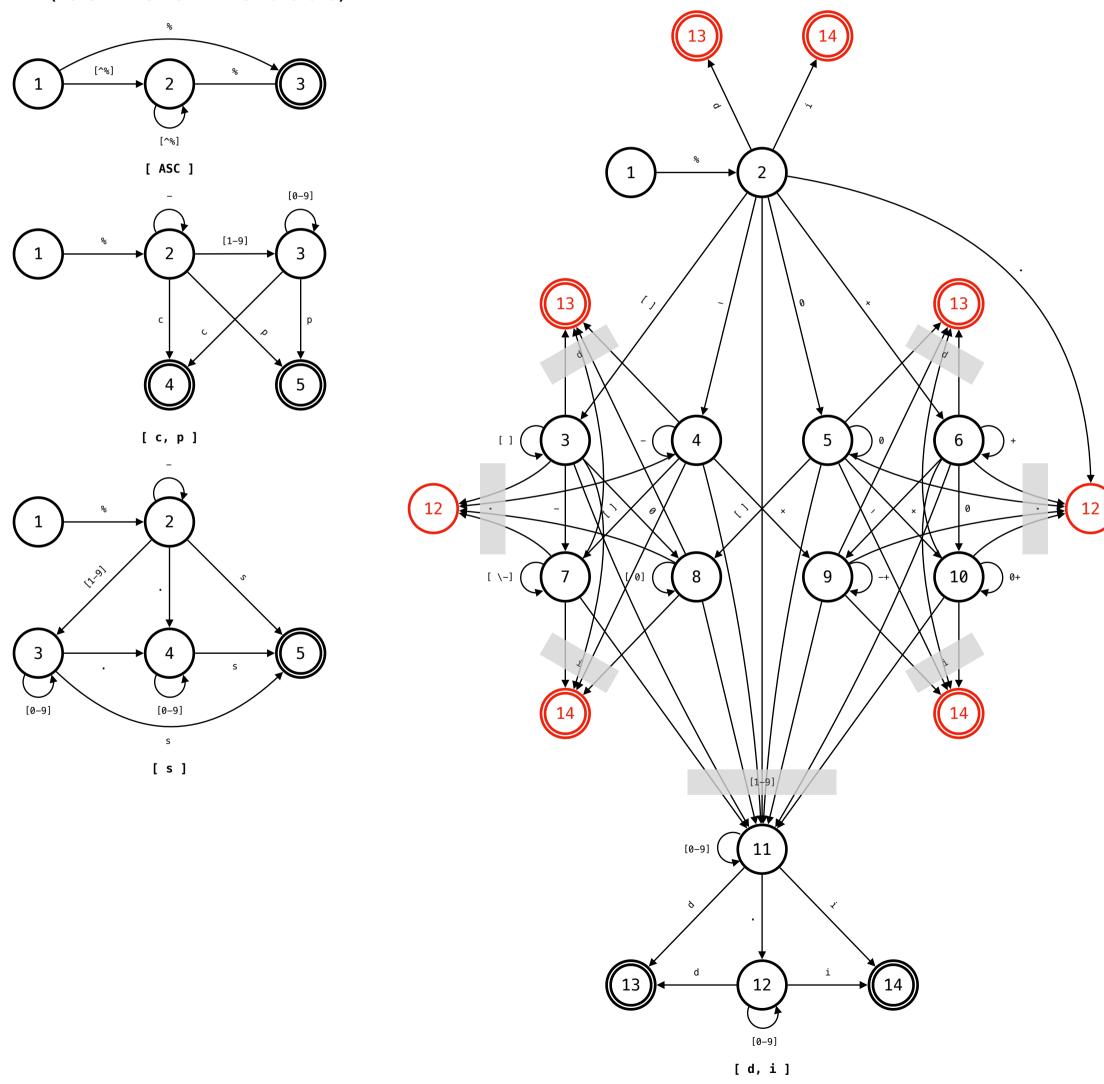
p = %-\*{WIDTH}?p

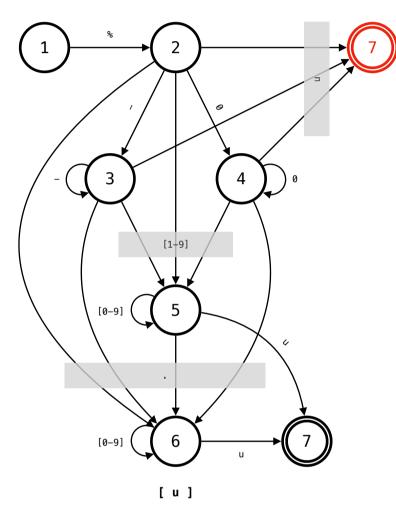
s = %-\*{WIDTH}?{PRECISION}?s

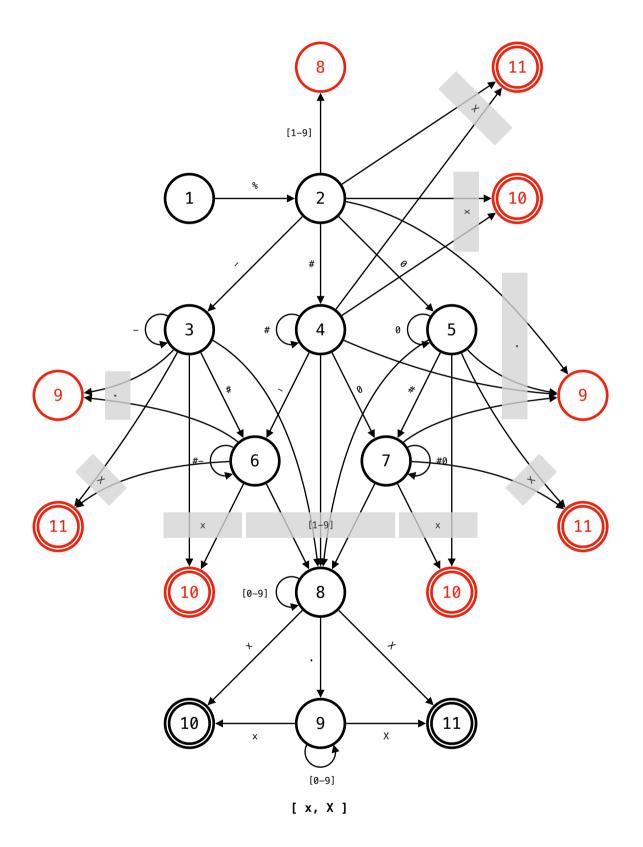
 $u = %(-*|0*)?{WIDTH}?{PRECISION}?u$ 

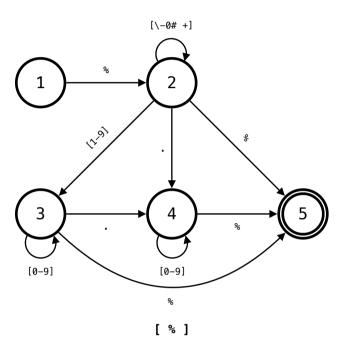
 $x = %([\-\#]*|[0\#]*)?{WIDTH}?{PRECISION}?x$ 

# DFA(Deterministic Finite Automata)









## **RE(Regular Expression)**

WIDTH = [1-9][0-9]\*PRECISION = \.[0-9]\*

ASC = [^%]\*

% = %[\-0# +]\*{WIDTH}?{PRECISION}?%

 $X = %([\-\#]*|[0\#]*)?{WIDTH}?{PRECISION}?X$ 

 $c = %-*{WIDTH}?c$ 

 $d = %([\-]*|[\-+]*|[0]*|[0+]*)?{WIDTH}?{PRECISION}?d$ 

 $i = %([\-]*|[\-+]*|[0]*|[0+]*)?{WIDTH}?{PRECISION}?i$ 

 $p = %-*{WIDTH}?p$ 

s = %-\*{WIDTH}?{PRECISION}?s

 $u = %(-*|0*)?\{WIDTH\}?\{PRECISION\}?u$ 

 $x = %([\-\#]*|[0\#]*)?{WIDTH}?{PRECISION}?x$ 



## ECS(Equivalence Classes)

	0	1	2	3	4	5	6	7	8	9
0	0	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	2	1	1	3	1	4	1	1
4	1	1	1	5	1	6	7	1	8	9
5	9	9	9	9	9	9	9	9	1	1
6	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	10	1
9	1	1	1	1	1	1	1	1	1	11
10	12	1	1	1	1	13	1	1	1	1
11	1	1	14	1	1	15	1	16	1	1
12	17	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1	1	1
16	1	1	1	1	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	1
18	1	1	1	1	1	1	1	1	1	1
19	1	1	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1	1	1
21	1	1	1	1	1	1	1	1	1	1
22	1	1	1	1	1	1	1	1	1	1
23	1	1	1	1	1	1	1	1	1	1
24	1	1	1	1	1	1	1	1	1	1
25	1	1	1	1	1	1				

## **State Transition Table**

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
				#	%	+	_	•	0		X	С	d	i	р	S	u	x
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	2	3	3	3	4	3	3	3	3	3	3	3	3	3	3	3	3	3
2	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
3	2	3	3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	3
4	_	_	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
5	_	_	5	21	7	21	22	23	24	25	_	_	15	16	_	_	_	_
6	_	_	21	6	7	21	26	27	28	29	13	_	_	_	_	_	_	20
7	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
8	_	_	21	_	_	8	30	_	31	_	_	_	_	_	_	_	_	_
9	_	_	22	26	_	30	_	_	21	_	_	_	_	_	_	_	_	_
10	_	_	_	_	7	_	_	_	10	10	13	_	15	16	_	18	19	20
11	-	_	24	28	7	31	21	32	11	33	13	_	15	16	_	_	19	20
12	-	_	-	_	-	_	_	10	12	12	-	14	_	-	17	_	_	_
13	_	_	_	_	_	_	_	-	_	_	_	_	_	_	_	_	_	-
14	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
15	_	_	_	_	_	_	_	_	_	_	_	_	_	-	_	_	_	_
16	_	_	_	_	-	_	_	_	_	_	_	_	_	-	_	_	_	_
17	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
18	-	_	-	_	-	_	_	_	_	_	-	_	_	-	_	_	_	_
19	-	_	-	-	-	-	_	-	-	_	-	_	_	-	_	_	-	-
20	_	_	-	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
21	_	_	21	21	7	21	21	34	21	35	_	_	_	_	_	_	_	_
22	_	_	22	_	-	_	-	_	21	_	_	_	_	-	_	_	_	_
23	_	_	_	_	7	_	_	_	23	23	_	_	15	16	_	_	_	-
24	_	_	24	_	-	_	21	_	_	_	_	_	_	-	_	_	_	_
25	_	_	_	_	_	_	_	23	25	25	_	_	_	_	_	_	_	_
26	_	_	_	26	_	_	_	_	21	_	_	_	_	_	_	_	_	_
27	_	_	_	_	7	_	_	_	27	27	13	_	_	_	_	_	_	20
28	_	_	_	28	_	_	21	_	_	_	_	_	_	_	_	_	_	_
29	_	_	_	_	_	_	_	27	29	29	_	_	_	_	_	_	_	_
30	_	_	_	_	_	30	_	_	21	_	_	_	_	_	_	_	_	_
31	_	_	_	_	_	31	21	_	_	_	_	_	_	_	_	_	_	_
32	_	_	_	_	7	_	_	_	32	32	13	_	15	16	_	_	19	20
33	_	_	_	_	_	_	_	32	33	33	_	_	_	_	_	_	_	_
34	_	_	_	_	7	_	_	_	34	34	_	_	_	_	_	_	_	_
35	_	_	-	-	-	-	_	34	35	35	-	_	_	_	_	_	_	_

## Base Table

	0	1	2	3	4	5	6	7	8	9
0	0	18	0	36	52	68	80	0	90	97
1	99	115	126	0	0	0	0	0	0	0
2	0	139	102	145	124	143	98	151	88	155
3	131	73	161	172	178	176				

## Check Table

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	1	1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	3	3	3	3
4	3	3	3	3	3	3	3	3	3	3
5	3	3	3	3	4	4	4	4	4	4
6	4	4	4	4	4	4	4	4	4	4
7	5	5	5	5	5	5	5	5	31	31
8	5	5	6	6	6	6	6	6	6	6
9	6	28	8	_	28	8	8	6	8	9
10	9	26	9	10	22	9	26	10	10	10
11	22	10	10	_	10	10	10	11	11	11
12	11	11	11	11	11	11	24	11	11	_
13	24	11	11	12	12	12	30	12	_	30
14	12	21	21	21	21	21	21	21	21	23
15	25	25	25	23	23	27	_	23	23	27
16	27	27	29	29	29	32	_	_	27	32
17	32	32	_	32	32	_	_	32	32	33
18	33	33	34	35	35	35	34	34	_	-
19	_	_	_	_	_	_				

CSR Compressed Sparse Row

# Next Table

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	2	3
2	3	3	4	3	3	3	3	3	3	3
3	3	3	3	3	3	3	2	3	3	3
4	2	3	3	3	3	3	3	3	3	3
5	3	3	3	3	5	6	7	8	9	10
6	11	12	13	14	15	16	17	18	19	20
7	5	21	7	21	22	23	24	25	31	21
8	15	16	21	6	7	21	26	27	28	29
9	13	28	21	_	21	8	30	20	31	22
10	26	26	30	7	22	21	21	10	10	13
11	21	15	16	_	18	19	20	24	28	7
12	31	21	32	11	33	13	24	15	16	_
13	21	19	20	10	12	12	30	14	_	21
14	17	21	21	7	21	21	34	21	35	7
15	23	25	25	23	23	7	_	15	16	27
16	27	13	27	29	29	7	_	_	20	32
17	32	13	_	15	16	_	_	19	20	32
18	33	33	7	34	35	35	34	34	_	_
19	_	_	_	_	_	_				

#### ECS(Equivalence Classes)

#### Base Table

 0
 1
 2
 3
 4
 5
 6
 7
 8
 9

 0
 0
 18
 0
 36
 52
 68
 80
 0
 90
 97

 1
 99
 115
 126
 0
 0
 0
 0
 0
 0
 0

 2
 0
 139
 102
 145
 124
 143
 98
 151
 88
 155

 3
 131
 73
 161
 172
 178
 176

#### Check Table

#### Def Table

#### Next Table

# MECS Meta Equivalence Classes

#### ECS(Equivalence Classes)-1

0 1 2 3 4 5 6 7 8 9

0 0 1 1 1 1 1 1 1 1 1 1 1

2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 **3** | 1 | 1 | 2 | 1 | 1 | 3 | 1 | 4 | 1 | 1 **4** | 1 | 1 | 1 | 5 | 1 | 6 | 7 | 1 | 8 | 9 **5** | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 1 | 1 6 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 7 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 **10** | 12 | 1 | 1 | 1 | 13 | 1 | 1 | 1 **11** | 1 | 1 | 14 | 1 | 1 | 15 | 1 | 16 | 1 | 1 **12** | 17 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 **13** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 14 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 **15** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 **16** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 **18** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 **19** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 20 1 1 1 1 1 1 1 1 1 1 1 **21** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 22 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 23 1 1 1 1 1 1 1 1 1 1 1 1 1 **24** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 25 | 1 | 1 | 1 | 1 | 1

#### Base Table-1

 0
 1
 2
 3
 4
 5
 6
 7
 8
 9

 0
 0
 14
 0
 15
 18
 34
 46
 0
 56
 63

 1
 65
 81
 92
 0
 0
 0
 0
 0
 0
 0

 2
 0
 105
 68
 111
 90
 109
 64
 117
 54
 121

 3
 97
 39
 127
 138
 144
 142
 132

#### Check Table-1

#### Def Table-1

 0
 1
 2
 3
 4
 5
 6
 7
 8
 9

 0
 0
 36
 0
 36
 0
 0
 0
 0
 5
 4

 1
 0
 0
 10
 0
 0
 0
 0
 0
 0
 0

 2
 0
 0
 5
 0
 5
 23
 6
 0
 6
 27

 3
 8
 8
 0
 32
 0
 34
 0
 0
 0

#### Next Table-1

#### MECS(Meta-Equivalence Classes)

 0
 1
 2
 3
 4
 5
 6
 7
 8
 9

 0
 0
 1
 1
 1
 2
 1
 1
 1
 1
 1

 1
 1
 1
 1
 1
 1
 1
 1
 1
 1

# ECS(Equivalence Classes)

	0	1	2	3	4	5	6	7	8	9
0	0	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	2	1	1	3	1	4	1	1
4	1	1	1	5	1	6	7	1	8	9
5	9	9	9	9	9	9	9	9	1	1
6	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	10	1
9	1	1	1	1	1	1	1	1	1	11
10	12	1	1	1	1	13	1	1	1	1
11	1	1	14	1	1	15	1	16	1	1
12	17	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1	1	1
16	1	1	1	1	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	1
18	1	1	1	1	1	1	1	1	1	1
19	1	1	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1	1	1
21	1	1	1	1	1	1	1	1	1	1
22	1	1	1	1	1	1	1	1	1	1
23	1	1	1	1	1	1	1	1	1	1
24	1	1	1	1	1	1	1	1	1	1
25	1	1	1	1	1	1				

# MECS(Meta-Equivalence Classes)

		1								
0	0	1	1	1	2	1	1	1	1	1
1	1	1	1	1	1	1	1	1		

## Base Table

	0	1	2	3	4	5	6	7	8	9
0	0	14	0	15	18	34	46	0	56	63
1	65	81	92	0	0	0	0	0	0	0
2	0	105	68	111	90	109	64	117	54	121
3	97	39	127	138	144	142	132			

## Def Table

	0	1	2	3	4	5	6	7	8	9
0	0	36	0	36	0	0	0	0	5	4
1	0	0	10	0	0	0	0	0	0	0
2	0	0	5	0	5	23	6	0	6	27
3	8	8	0	32	0	34	0			

## Advance Table

	0	1	2	3	4	5	6	7	8	9
0	1	1		1						
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1				

## Check Table

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	1	3
2	4	4	4	4	4	4	4	4	4	4
3	4	4	4	4	4	4	5	5	5	5
4	5	5	5	5	31	31	5	5	6	6
5	6	6	6	6	6	6	6	28	8	_
6	28	8	8	6	8	9	9	26	9	10
7	22	9	26	10	10	10	22	10	10	_
8	10	10	10	11	11	11	11	11	11	11
9	11	11	24	11	11	_	24	11	11	12
10	12	12	30	12	_	30	12	21	21	21
11	21	21	21	21	21	23	25	25	25	23
12	23	27	_	23	23	27	27	27	29	29
13	29	32	36	36	27	32	32	32	_	32
14	32	_	_	32	32	33	33	33	34	35
15	35	35	34	34	_	_	_	_	_	_
16	_	_								

## Next Table

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	4	2
2	5	6	7	8	9	10	11	12	13	14
3	15	16	17	18	19	20	5	21	7	21
4	22	23	24	25	31	21	15	16	21	6
5	7	21	26	27	28	29	13	28	21	_
6	21	8	30	20	31	22	26	26	30	7
7	22	21	21	10	10	13	21	15	16	_
8	18	19	20	24	28	7	31	21	32	11
9	33	13	24	15	16	_	21	19	20	10
10	12	12	30	14	_	21	17	21	21	7
11	21	21	34	21	35	7	23	25	25	23
12	23	7	_	15	16	27	27	13	27	29
13	29	7	2	3	20	32	32	13	_	15
14	16	_	_	19	20	32	33	33	7	34
15	35	35	34	34	_	_	_	_	_	_
16	_	_								

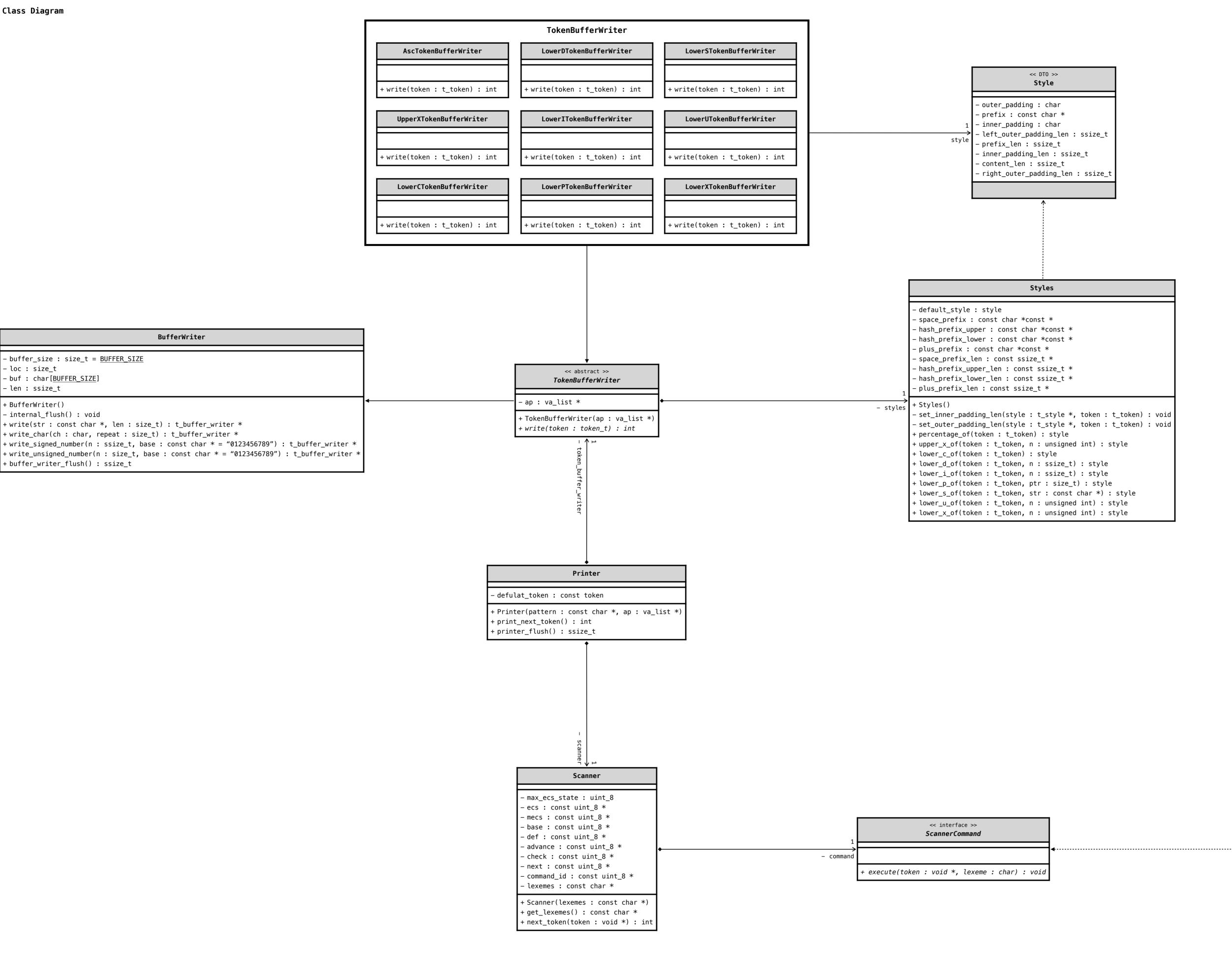
## Command ID Table

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	1
2	3	4	11	5	6	7	8	9	12	13
3	14	15	16	17	18	19	3	4	11	5
4	6	7	8	9	5	6	14	15	3	4
5	11	5	6	7	8	9	12	4	3	_
6	6	5	6	19	8	3	4	4	5	11
7	3	8	8	10	10	12	8	14	15	_
8	17	18	19	3	4	11	5	6	7	8
9	9	12	3	14	15	_	6	18	19	7
10	9	9	5	13	_	8	16	3	4	11
11	5	6	7	8	9	11	7	9	9	10
12	10	11	_	14	15	10	10	12	7	9
13	9	11	1	2	19	10	10	12	_	14
14	15	_	_	18	19	7	9	9	11	7
15	9	9	10	10	_	_	_	_	_	_
16	_	_								

## **Command Table**

	Name					
•						
0	empty					
1	set_asc					
2	increase_width					
3	set_space					
4	set_hash					
5	set_plus					
6	set_minus					
7	set_precision					
8	set_zero					
9	add_width					
10	add_precision					
11	set_percentage					
12	set_X					
13	set_c					
14	set_d					
15	set_i					
16	set_p					
17	set_s					
18	set_u					
19	set_x					

# Class Diagram



# ScannerCommand AddPrecisionScannerCommand EmptyScannerCommand -empty(token : t\_token \*, lexeme : char) : void - add\_precision(token : t\_token \*, lexeme : char) : void -execute(token : void \*, lexeme : char) : void + execute(token : void \*, lexeme : char) : void SetAscScannerCommand SetPercentageScannerCommand - set\_asc(token : t\_token \*, lexeme : char) : void - set\_percentage(token : t\_token \*, lexeme : char) : void + execute(token : void \*, lexeme : char) : void + execute(token : void \*, lexeme : char) : void IncreaseWidthScannerCommand SetUpperXScannerCommand - increase\_width(token : t\_token \*, lexeme : char) : void - set\_upper\_x(token : t\_token \*, lexeme : char) : void + execute(token : void \*, lexeme : char) : void + execute(token : void \*, lexeme : char) : void SetLowerCScannerCommand SetSpaceScannerCommand - set\_space(token : t\_token \*, lexeme : char) : void - set\_lower\_c(token : t\_token \*, lexeme : char) : void + execute(token : void \*, lexeme : char) : void + execute(token : void \*, lexeme : char) : void SetHashScannerCommand SetLowerDScannerCommand - set\_hash(token : t\_token \*, lexeme : char) : void - set\_lower\_d(token : t\_token \*, lexeme : char) : void + execute(token : void \*, lexeme : char) : void + execute(token : void \*, lexeme : char) : void SetLowerIScannerCommand SetPlusScannerCommand - set\_lower\_i(token : t\_token \*, lexeme : char) : void - set\_plus(token : t\_token \*, lexeme : char) : void -execute(token : void \*, lexeme : char) : void + execute(token : void \*, lexeme : char) : void SetMinusScannerCommand SetLowerPScannerCommand - set\_minus(token : t\_token \*, lexeme : char) : void - set\_lower\_p(token : t\_token \*, lexeme : char) : void + execute(token : void \*, lexeme : char) : void + execute(token : void \*, lexeme : char) : void SetPrecisionScannerCommand SetLowerSScannerCommand - set\_precision(token : t\_token \*, lexeme : char) : void

- execute(token : void \*, lexeme : char) : void

SetZeroScannerCommand

AddWidthScannerCommand

- add\_width(token : t\_token \*, lexeme : char) : void

+ execute(token : void \*, lexeme : char) : void

- set\_zero(token : t\_token \*, lexeme : char) : void

+ execute(token : void \*, lexeme : char) : void

- set\_lower\_s(token : t\_token \*, lexeme : char) : void

SetLowerUScannerCommand

- set\_lower\_u(token : t\_token \*, lexeme : char) : void

SetLowerXScannerCommand

- set\_lower\_x(token : t\_token \*, lexeme : char) : void

+ execute(token : void \*, lexeme : char) : void

+ execute(token : void \*, lexeme : char) : void

+ execute(token : void \*, lexeme : char) : void