1 APPENDIX

1.1 36 projects used in mining and DSI+

Table 1: Projects that we inspected. **KLoC**: thousands of lines of code, **TC**: no. of test classes, **TM**: no. of test methods, **SC**: statement coverage, **BC**: branch coverage, **USE**: no. of clients, **S#**: no. of mined specifications, **DATE**: date last updated.

Project	SHA	KLoC	TC	TM	SC	BC	USE	S#	DATE
antlrworks [1]	b602e	42.5	4	17	.04	.06	-1	672	2020/12/21
asterisk [2]	8d8a0	61.7	41	232	.18	.21	23	1032	2022/02/28
edn-java [5]	c1d89	3.8	7	90	.80	.72	31	143	2022/01/17
cli [6]	98d06	7.1	28	355	.97	.94	3945	1384	2022/03/11
codec [7]	b959a	23.9	58	936	.97	.93	10618	445	2022/03/04
collections [8]	ee282	66.8	214	16720	.89	.83	3216	2870	2022/03/16
compress [9]	4246b	72.6	163	1258	.85	.74	2404	10480	2022/03/11
configuration [10]	3f28d	51.6	170	2839	.87	.83	802	27117	2022/03/05
exec [16]	2ca7c	3.6	14	91	.71	.61	753	391	2022/03/04
fileupload [18]	55dc6	4.8	9	34	.81	.77	2312	337	2022/03/09
graph [21]	f3a21	10.5	27	158	.85	.79	-1	364	2020/08/26
jexl [24]	a08f9	31.7	53	792	.87	.78	186	39932	2022/03/14
net [32]	eb018	28.3	44	290	.38	.30	1277	1738	2022/03/14
ognl [33]	6804f	20.4	60	745	.71	.62	-1	2797	2022/02/25
validator [35]	72734	16.7	70	554	.86	.74	1387	664	2022/03/04
connector [12]	e9042	2.0	10	36	.54	.42	3	1198	2018/06/13
exp4j [17]	bae81	5.2	9	311	.92	.90	-1	119	2019/09/27
dataformat [15]	d2752	3.8	9	35	.51	.46	2809	180	2017/03/19
geoserver [19]	d518e	11.9	44	136	.24	.18	8	392	2019/05/03
weiboclient4j [37]	80556	8.8	27	34	.12	.14	-1	184	2018/01/31
confucius [11]	12604	1.3	6	90	.96	.96	-1	228	2021/07/12
jmxtrans [25]	3b576	6.6	20	63	.37	.32	3	979	2021/01/07
jnr [26]	b2cc7	6.9	2	11	.16	.13	83	17	2022/01/06
beans [4]	597a6	51.3	43	881	.45	.45	16	3875	2021/04/20
convert [13]	8f8bf	5.4	9	160	.76	.72	852	210	2021/12/15
money [30]	7965e	9.1	14	1483	.97	.95	129	484	2022/03/16
jscep [27]	ddbcb	6.3	45	210	.81	.74	14	398	2022/03/17
jtar [28]	4b669	1.0	2	8	.75	.67	24	11	2020/08/19
libpam4j [29]	377d7	0.6	1	3	.26	.27	27	18	2018/08/03
mp3agic [31]	99fa1	6.9	19	205	.89	.68	11	2741	2022/01/19
storm-jms [34]	d152d	1.1	1	2	.23	.14	1	29	2016/08/25
romix [22]	03335	2.4	15	19	.77	.67	19	48	2019/08/21
axiom [3]	90701	6.6	9	29	.32	.08	-1	1213	2021/04/23
visualee [36]	410a8	3.6	17	75	.68	.72	-1	86	2017/12/12
crawler [14]	21b95	1.8	10	42	.74	.74	-1	50	2019/01/21
jandex [23]	8b384	8.3	5	9	.66	.55	259	438	2022/02/07

1.2 Formulas

Accuracy, Precision, Recall, and F1-Score. We compute DSI+ accuracy, precision, and recall using: (1) **TP** (true positives): DSI+ classifies as likely valid and manual found a true specification; (2) **FP** (false positives): DSI+ classifies as likely valid and manual found a spurious specification; (3) **TN** (true negatives): DSI+ classifies as likely spurious and manual found a spurious specification; (4) **FN** (false negatives): DSI+ classifies as likely spurious and manual found a true specification; (5) **UP**: DSI+ classifies as likely valid, but manual found NBP; (6) **UN**: DSI+ classifies as spurious but manual found NBP; (7) $\mathbf{U}^T$: DSI+ could not validate, and manual found NBP; (8) $\mathbf{U}^{fn}$: DSI+ could not validate, but manual found a true specification; (9) $\mathbf{U}^{tn}$: DSI+ could not validate, but manual found a true specification; (10) $\mathbf{M}_T$: DSI+ gave mixed verdicts, but manual found a true specification; and (11) $\mathbf{M}_N$: DSI+ gave mixed verdicts, but manual found a spurious spec.

We use formulas for multi-class classification to compute the accuracy (Acc [%]), macro-Precision (Precision[%]), macro-recall (Recall[%]), and macro-F1 score (F1-Score) [20] of DSI+: Acc = $\mathbf{TP} + \mathbf{TN} + \mathbf{U}^T$, Acc [%] = $\frac{Acc}{\Sigma}$, Precision[%] $\mathbf{TP} = \frac{\mathbf{TP}}{\mathbf{TP+FP+UP}} \times 100$, Recall[%] $\mathbf{TP} = \frac{\mathbf{TP}}{\mathbf{TP+FN+U}^{fn}} \times 100$, Precision[%] $\mathbf{TN} = \frac{\mathbf{TN}}{\mathbf{FN+TN+UN}} \times 100$, Recall[%] $\mathbf{TN} = \frac{\mathbf{TN}}{\mathbf{FP+TN+U}^{tn}} \times 100$, Avg. Precision = $\frac{Precision\mathbf{TP}+Precision\mathbf{TN}+Precision\mathbf{U}^T}{3}$, Avg. Recall = $\frac{Recall\mathbf{TP}+Recall\mathbf{TN}+Recall\mathbf{U}^T}{3}$, F1-Score = $2 \times \frac{Avg.Precision \times Avg.Recall}{(Avg.Precision^{-1}+Avg.Recall^{-1})}$.

Note that these formulas are applicable here because DSI+ classifies specifications into more than two categories (LV, LS, U, Mixed). We omit Mixed cases from our Precision, Recall, and F1-Score calculation because there are no Mixed cases in our "ground truth".

Assertions vs. Exceptions. A brief example of how % Assert^P is computed is as follows. Suppose DSI+ verdict is LV for **TP** specification X using tests T1, T2, T3, but an exception is thrown with T1, and an assertion fails with T2 and T3. Suppose also that **TP** DSI+ verdict is LV for specification Y tests T4 and T5, but an exception is thrown with T4, and an exception fails with T5. Then, "% Assert" is $\frac{2+1}{5}$ ($\frac{T2+T3+T5}{T1+T2+T3+T4+T5}$) and "% Assert^P" is $\frac{\frac{2}{3}+\frac{1}{2}}{2}$ (average of $\frac{T2+T3}{T1+T2+T3}$ for X, and $\frac{T5}{T4+T5}$ for Y).

1.3 Accuracy, Precision and Recall of DSI+

We list additional tables computing accuracy, precision, and recall of DSI+ based on alternative interpretations. Note that in Table 3, MC and MW do not include the specifications from US, and the difference between ALL-Σ in Table 5 and Table 6 is 280 because the 7 specifications that were in both Mixed and US are excluded from Table 6.

Table 2: Majority votes of DSI+ outcomes for Mixed cases. Rows are the categories of disagreements. Columns are the majority vote between the tests of each specification within the disagreement category.

DSI+ \ Majority	LV	LS	U	Σ
LV-LS	41	130	0	171
LV-U	6	0	7	13
LS-U	0	38	43	81
LV-LS-U	0	11	11	22
Σ	47	179	61	287

Table 3: Manual Inspection Outcomes for all Mixed cases per project.

ID \ Man	TS		NS	NBP	US	MC	MW	Σ	Acc [%]
	Must	May							
P1	0	2	1	0	0	3	0	3	100.0
P2	7	2	40	0	0	49	0	49	100.0
P3	4	1	34	3	4	38	4	46	82.6
P4	17	6	46	0	0	68	1	69	98.6
P5	2	45	5	0	0	52	0	52	100.0
P6	4	4	56	1	3	65	0	68	95.6
Σ	34	60	182	4	7	275	5	287	-

Table 4: Comparison of DSI+ and Manual Inspection Outcomes, if Mixed specifications are considered to be incorrect DSI+ categorizations.

ID	**TP**	**FP**	**UP**	**TN**	**FN**	**UN**	**U**^{T}	**U**^{fn}	**U**^{tn}	Acc	Total	Acc [%]	Precision[%]		Recall[%]	
													TP	**TN**	**TP**	**TN**
P1	1	0	0	2	1	0	1	0	3	4	11	36.4	100.0	66.7	50.0	40.0
P2	13	24	6	71	4	1	169	0	105	253	442	57.2	30.2	93.4	76.5	35.5
P3	16	13	0	263	6	1	101	7	210	380	659	57.7	55.2	97.4	55.2	54.1
P4	5	12	0	89	11	0	78	17	50	172	331	52.0	29.4	89.0	15.2	58.9
P5	12	25	0	29	17	0	43	3	29	84	210	40.0	32.4	63.0	37.5	34.9
P6	5	17	0	75	2	0	98	8	64	178	334	53.3	22.7	97.4	33.3	48.1
ALL	52	91	6	529	41	2	490	35	461	1071	1987	53.9	34.8	92.4	40.6	48.9
AVG	8.7	15.2	1	88.2	6.8	0.3	81.7	5.8	76.8	178.5	331.2	49.4	45.0	84.5	44.6	45.3

Table 5: Comparison of DSI+ and Manual Inspection Outcomes, if Mixed specifications are omitted.

ID	**TP**	**FP**	**UP**	**TN**	**FN**	**UN**	**U**^{T}	**U**^{fn}	**U**^{tn}	Acc	Σ	Acc [%]	Precision[%]		Recall[%]	
													TP	**TN**	**TP**	**TN**
P1	1	0	0	2	1	0	1	0	3	4	8	50.0	100.0	66.7	50.0	40.0
P2	13	24	6	71	4	1	169	0	105	253	393	64.4	30.2	93.4	76.5	35.5
P3	16	13	0	263	6	1	101	7	210	380	617	61.6	55.2	97.4	55.2	54.1
P4	5	12	0	89	11	0	78	17	50	172	262	65.6	29.4	89.0	15.2	58.9
P5	12	25	0	29	17	0	43	3	29	84	158	53.2	32.4	63.0	37.5	34.9
P6	5	17	0	75	2	0	98	8	64	178	269	66.2	22.7	97.4	33.3	48.1
ALL	52	91	6	529	41	2	490	35	461	1071	1707	62.7	34.8	92.4	40.6	48.9
AVG	8.7	15.2	1	88.2	6.8	0.3	81.7	5.8	76.8	178.5	284.5	60.2	45.0	84.5	44.6	45.3

Table 6: Comparison of DSI+ and Manual Inspection Outcomes, if the majority vote of Mixed specifications are incorporated into the DSI+ outcomes.

ID	**TP**	**FP**	**UP**	**TN**	**FN**	**UN**	$\mathbf{U}^{T}$	$\mathbf{U}^{fn}$	$\mathbf{U}^{tn}$	Acc	Σ	Acc [%]	Precision[%]		Recall[%]	
													TP	**TN**	**TP**	**TN**
P1	3	1	0	2	1	0	1	0	3	6	11	54.5	75.0	66.7	75.0	33.3
P2	16	27	6	92	9	1	169	1	121	277	442	62.7	32.7	90.2	61.5	38.3
P3	17	14	0	285	8	1	104	9	221	406	659	61.6	54.8	96.9	50.0	54.8
P4	14	33	0	109	23	0	78	19	55	201	331	60.7	29.8	82.6	25.0	55.3
P5	12	25	0	34	64	0	43	3	29	89	210	42.4	32.4	34.7	15.2	38.6
P6	7	21	0	112	6	1	98	10	79	217	334	65.0	25.0	94.1	30.4	52.8
ALL	69	121	6	634	111	3	493	42	508	1196	1987	60.1	35.2	84.7	31.0	50.1
AVG	11.5	20.2	1	105.7	18.5	0.5	82.2	7	84.7	199.3	331.2	57.8	41.6	77.5	42.9	45.5

1.4 Return Values

We list tables that show the impact of `void` return values on DSI+.

Table 7: Return values of `a()` and `b()` across all inspected projects.

b() \ a()	V	NV	Σ
V	276	128	404
NV	539	1058	1597
Σ	815	1186	2001

Table 8: Return values of `a()` and `b()` across true specifications in all inspected projects.

b() \ a()	V	NV	Σ
V	42	20	62
NV	127	33	160
Σ	169	53	222

Table 9: Return values of `a()` and `b()` across spurious specifications in all inspected projects.

b() \ a()	V	NV	Σ
V	195	83	278
NV	359	626	985
Σ	554	709	1263

Table 10: Return values of `a()` and `b()` across **FP** specifications in all inspected projects.

b() \ a()	V	NV	Σ
V	30	5	35
NV	34	22	56
Σ	64	27	91

Table 11: Return values of `a()` and `b()` across **FN** specifications in all inspected projects.

b() \ a()	V	NV	Σ
V	6	2	8
NV	32	1	33
Σ	38	3	41

1.5 Categorizations of Observations from Manual Inspection

We list additional tables that describe how often a certain categorization observed from manual inspection occured in each project, and each outcome category (following from Appendix Section 1.3). The columns follow from the descriptions in Appendix Section 1.3, and **%NS** is as described in the paper: $\textbf{\%NS} = \frac{\textbf{NS}}{\Sigma - (\textbf{NBP})} \times 100$ is the percentage of manually confirmed spurious specifications in each category and Σ is the number of specifications that we observed.

Table 12: Per project categorization of observations from manual inspection. `AReturn`: the replacement value used when `a()` is delayed.

Category	T_{ID}	Description	P1	P2	P3	P4	P5	P6	Count
Relation	T1	`a()` and `b()` are stateful but unrelated	0	37	188	18	44	20	307
	T2	`a()` XOR `b()` does not alter *P* state	2	63	126	47	7	46	291
	T3	`a()` XOR `b()` is a pure setter	0	1	88	78	0	54	221
	T4	`a()` and `b()` are connected	0	23	93	38	3	18	175
	T5	Neither `a()` nor `b()` alters *P* state	1	16	31	4	8	29	89
	T6	`a()` and `b()` are both pure setters	0	0	31	8	0	4	43
	T7	`a()` and `b()`'s caller form true specification	0	0	0	1	0	11	12
	T8	`a()` and `b()` modify disjoint part of *P* state	0	1	1	0	0	0	2
Return	T9	`a()` returns void	8	76	296	213	124	98	815
	T10	`AReturn` ! = test's expected value	0	168	75	7	23	41	314
	T11	`AReturn` == null causes NPE	0	52	62	13	16	113	256
	T12	`AReturn` unchecked	0	43	51	19	3	9	125
	T13	`AReturn` == test's expected value	1	27	25	4	11	27	95
	T14	`AReturn` == `a()`'s return	0	36	43	0	0	10	89
	T15	Discards `a()`'s return value	0	20	5	39	8	0	72
	T16	`AReturn` guards path; `b()` not called	0	3	29	1	1	6	40
Exception	T18	Test expects an exception	0	50	28	39	73	84	274
	T19	Expected exception not thrown; test fails	0	20	0	3	16	20	59
	T20	Exception swallowed in catch block	0	0	31	0	0	0	31
	T21	Expects exception	0	10	0	10	0	0	20
	T22	Delaying `a()` causes unexpected exception	0	0	9	0	0	1	10
Oracle	T23	Test has weak oracle	4	57	156	110	111	99	537
	T24	Order of assertions affects DSI+ outcome	0	75	50	18	10	18	171
Delay	T25	DSI+ pollutes state for tests	0	0	0	0	0	90	90
	T26	Delayed call of `a()` restores *P*'s state	1	0	46	19	0	21	87
	T27	Delay causes assertion to fail	0	0	0	39	0	0	39
	T28	Delay causes a timeout	0	0	0	9	0	0	9
	T29	Delay corrupts state for `b()`	0	6	1	1	0	0	8
Misc	T30	Multiple usage scenarios	0	44	153	40	20	51	308
	T31	`a()` and `b()` called in different threads	0	0	0	73	0	0	73
	T32	Dynamic dispatch slowed down inspection	0	22	5	5	5	33	70
	T33	Bug causing sanity-check-failure	4	18	3	0	0	30	55
	T34	A NBP that goes beyond method-call chains	0	8	2	23	6	3	42
	T35	Javadoc confirms manual inspection	0	0	8	5	0	1	14
	T37	Configurations influenced DSI+ outcome	0	0	12	0	0	0	12
	T38	`b()` calls `a()`	0	3	4	1	1	1	10

Table 13: Categorization of observations from manual inspection based on DSI+ vs manual inspection outcomes. Mixed specifications are ommited.

Category	T_{ID}	**TP**	**FP**	**TN**	**FN**	$\mathbf{U}^{T}$	$\mathbf{U}^{fn}$	$\mathbf{U}^{tn}$	**%NS**	Σ
Relationship	T1	0	33	138	0	0	0	109	100.0	280
	T2	5	19	116	5	10	2	96	95.1	253
	T3	9	16	110	11	0	2	39	88.2	187
	T4	0	28	46	0	0	0	80	100.0	154
	T5	0	3	38	0	1	0	31	100.0	73
	T6	0	0	42	0	0	0	0	100.0	42
	T7	0	0	2	0	0	0	4	100.0	6
	T8	0	0	2	0	0	0	0	100.0	2
Return	T9	36	64	323	38	78	20	101	83.8	666
	T10	2	1	0	0	151	1	109	97.3	264
	T11	5	9	1	1	87	10	117	88.8	230
	T12	0	3	83	0	9	0	4	100.0	100
	T14	0	2	65	0	8	0	0	100.0	77
	T15	4	2	2	0	51	1	4	61.5	64
	T13	0	2	35	0	14	0	1	100.0	53
	T16	0	0	0	0	2	1	37	97.4	40
Exception	T18	7	36	30	1	62	0	24	91.8	160
	T19	3	10	0	0	28	0	2	80.0	49
	T20	4	0	2	2	0	2	15	68.0	25
	T21	0	0	0	0	10	0	0	0.0	10
	T22	2	0	0	0	0	0	8	80.0	10
Oracle	T23	3	29	275	32	1	0	1	89.7	341
	T24	0	0	0	0	1	1	126	99.2	128
Delay	T26	0	0	80	2	0	0	1	97.6	83
	T25	0	0	0	0	19	5	19	79.2	43
	T27	0	1	2	0	11	0	4	100.0	18
	T29	1	2	0	0	0	1	1	60.0	5
	T28	0	0	0	0	0	0	0	0.0	0
Misc	T31	0	0	0	0	22	13	36	73.5	71
	T32	1	4	8	1	17	1	20	91.4	56
	T33	0	0	0	0	6	2	46	95.8	54
	T34	0	0	2	0	22	0	11	100.0	41
	T38	0	0	3	0	0	1	5	88.9	10
	T35	1	1	4	0	0	2	2	70.0	10
	T37	0	0	0	0	1	0	0	0.0	1

Table 14: Categorization of observations from manual inspection based on DSI+ vs manual inspection outcomes. The majority vote of Mixed specifications are incorporated into the DSI+ outcomes.

Category	T_{ID}	**TP**	**FP**	**TN**	**FN**	$\mathbf{U}^{T}$	$\mathbf{U}^{fn}$	$\mathbf{U}^{tn}$	**%NS**	Σ
Relationship	T1	0	38	155	0	0	0	114	100.0	307
	T2	5	24	138	5	10	2	106	95.7	290
	T3	13	26	127	14	0	2	39	86.9	221
	T4	0	35	50	0	0	0	89	100.0	174
	T5	0	5	48	0	1	0	35	100.0	89
	T6	0	0	42	0	0	0	1	100.0	43
	T7	0	0	5	0	0	0	7	100.0	12
	T8	0	0	2	0	0	0	0	100.0	2
Return	T9	45	80	363	102	78	22	111	76.6	807
	T10	3	1	18	0	154	2	135	96.9	313
	T11	7	11	7	4	87	13	124	85.5	253
	T12	0	3	98	2	12	0	9	98.2	125
	T13	2	2	50	1	14	1	23	94.9	94
	T14	0	2	75	0	10	0	0	100.0	89
	T15	7	3	3	2	51	1	5	52.4	72
	T16	0	0	0	0	2	1	37	97.4	40
Exception	T18	17	52	60	52	62	0	26	66.7	269
	T19	5	12	3	2	28	0	2	70.8	58
	T20	4	0	2	2	0	4	18	66.7	30
	T21	5	4	1	0	10	0	0	50.0	20
	T22	2	0	0	0	0	0	8	80.0	10
Oracle	T23	15	51	343	96	1	2	22	78.6	531
	T24	0	1	10	0	1	1	154	99.4	167
Delay	T25	0	1	21	4	19	7	34	83.6	87
	T26	0	0	83	2	0	0	1	97.7	86
	T27	0	7	12	5	11	0	4	82.1	39
	T28	4	3	1	1	0	0	0	44.4	9
	T29	1	3	2	0	0	1	1	75.0	8
Misc	T31	0	0	0	0	22	13	38	74.5	73
	T32	2	5	17	1	17	1	23	91.8	70
	T33	0	0	0	0	6	2	46	95.8	54
	T34	0	0	3	0	22	0	11	100.0	42
	T37	1	0	4	0	1	0	6	90.9	12
	T35	1	1	6	0	0	2	2	75.0	12
	T38	0	0	3	0	0	1	5	88.9	10

REFERENCES

[1] antlr.antlrworks. https://github.com/antlr/antlrworks.git.
[2] asterisk-java. https://github.com/srt/asterisk-java.
[3] axiom. https://github.com/ryanbrainard/axiom.git.
[4] joda-beans. https://github.com/JodaOrg/joda-beans.git.
[5] bpsm.edn-java. https://github.com/bpsm/edn-java.git.
[6] commons-cli. https://github.com/apache/commons-cli.git.
[7] commons-codec. https://github.com/apache/commons-codec.git.
[8] commons-collections. https://github.com/apache/commons-collections.
[9] commons-compress. https://github.com/apache/commons-compress.git.
[10] commons-configuration. https://github.com/apache/commons-configuration.git.
[11] Ivantrendafilov.confucius. https://github.com/IvanTrendafilov/Confucius.git.
[12] connector4java. https://github.com/osiam/connector4java.git.
[13] convert. https://github.com/JodaOrg/joda-convert.git.
[14] vidageek.crawler. https://github.com/vidageek/crawler.git.
[15] Fasterxml.jackson-dataformat-yaml. https://github.com/FasterXML/jackson-dataformat-yaml.git.
[16] exec. https://github.com/apache/commons-exec.git.
[17] fasseg.exp4j. https://github.com/fasseg/exp4j.git.
[18] commons-fileupload. https://github.com/apache/commons-fileupload.
[19] geosolutions-it.geoserver-manager. https://github.com/geosolutions-it/geoserver-manager.git.
[20] M. Grandini, E. Bagli, and G. Visani. Metrics for multi-class classification: an overview. *ArXiv*, abs/2008.05756, 2020.
[21] commons-graph. https://github.com/apache/commons-graph.git.
[22] romix.java-concurrent-hash-trie-map. https://github.com/romix/java-concurrent-hash-trie-map.git.
[23] wildfly.jandex. https://github.com/wildfly/jandex.git.
[24] commons-jexl. https://github.com/apache/commons-jexl.git.

[25] jmxtrans.embedded-jmxtrans. https://github.com/jmxtrans/embedded-jmxtrans.git.
[26] jnr-posix. https://github.com/jnr/jnr-posix.git.
[27] jscep. https://github.com/jscep/jscep.git.
[28] kamranzafar.jtar. https://github.com/kamranzafar/jtar.git.
[29] kohsuke.libpam4j. https://github.com/kohsuke/libpam4j.git.
[30] joda-money. https://github.com/JodaOrg/joda-money.git.
[31] mp3agic. https://github.com/mpatric/mp3agic.git.
[32] commons-net. https://github.com/apache/commons-net.git.
[33] commons-ognl. https://github.com/apache/commons-ognl.git.
[34] ptgoetz.storm-jms. https://github.com/ptgoetz/storm-jms.git.
[35] commons-validator. https://github.com/apache/commons-validator.git.
[36] Thomas-s-b.visualee. https://github.com/Thomas-S-B/visualee.git.
[37] hoverruan.weiboclient4j. https://github.com/hoverruan/weiboclient4j.git.