学术论文

基于约束的事件序列消减

Jie Wang, “Constraint-Based Event Trace Reduction,” in Proceedings of ACM SIGSOFT International Symposium on the Foundations of Software Engineering(FSE-SRC), 2016, pp. 1106–1108.

Tel: 18801495728 (Jie Wang)、Email: wangjie12@otcaix.iscas.ac.cn

Motivation

JavaScript applications are event-driven, and record-replay techniques can facilitate JavaScript failure diagnosis. Event trace reduction techniques (e.g., dynamic slicing [1] and delta debugging [2]) are developed to remove failure-irrelevant events. **But, they suffer from efficiency or effectiveness issues.**

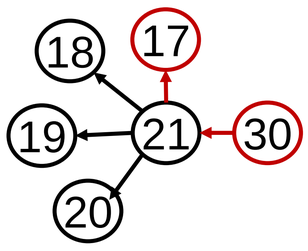
❖ ***Example - An event trace triggering a failure***

e1~e3: Create *booklist*
e14~e17: Add *book1* to *booklist*
e18~e21: Add *book2* to *booklist*
e27~e30: Add *book1* to *booklist*

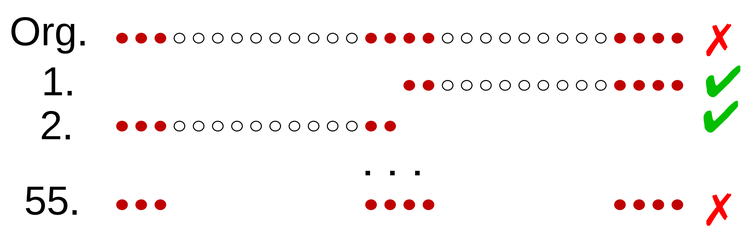
Only the 11 events are necessary to reproduce the error: {e1~e3, e14~e17, e27~30}

Error: duplicated *book1* in *booklist*

❖ ***Limitation of existing work***



Dynamic Slicing (DS) [1]
Keep all dependences
⇒ Cannot remove e18~e21 !



Delta-debugging (DD) [2]
Blindly select subtraces
⇒ too many candidates!

Approach (JsMin)

Basic idea: utilize runtime information to guide the trace generation

❖ ***Observation 1: The selected event trace should be feasible***

e3. var *booklist* = new ShoppingList()
e17. *booklist.*add(*book1*)

booklist must be defined before used:
select(e17) ⇒ *select*(e3)

❖ ***Observation 2: The exact value of a variable may not affect the occurrence of a failure***

e17. *booklist.*add(*book1*)
e21. *booklist.*add(*book2*)
e30. *booklist.*add(*book1*)

booklist can read its older value in e17 or e21:
select(e30) ⇒ *select*(e17) ∨ *select*(e21)

❖ ***Observation 3: The length of failure-related trace is usually short***

- No more than 6 events [2][3]

Generate event traces from short to long: *length* = 1, 2, 3, …

Evaluation

❖ ***Evaluation on the example***

length	Event trace
1	no valid trace
2	no valid trace
…	…
7	{e1~e3, e27~e30} ✔
…	…
11	{e1~e3, e14~e17, e27~e30} ✗

Generate traces	DD	JsMin
Average length	14	9
Count	55	2

❖ ***Result on 10 real-world failures from GitHub***

ID	Orig		DS		DD			JsMin		
	#event	#min	#event	time(s)	#event	time(s)	#trace	#event	time(s)	#trace
1	1051	5	39	14	5	1205	156	6	35	5
2	1189	5	92	18	5	1637	147	7	264	102
3	694	5	58	38	5	2771	227	12	115	18
4	342	2	2	6	2	150	38	2	8	1
5	1410	3	24	237	3	514	74	7	384	14
6	398	3	30	15	3	752	62	8	24	1
7	1326	2	11	143	2	702	48	8	165	1
8	1290	6	8	64	6	1034	116	8	134	1
9	367	2	32	30	2	148	23	2	62	6
10	617	5	12	45	5	1639	92	11	72	1
Avg.			30.8	61	2.7	1055	98	7.1	128	16

➢ *Remove 70% more failure-irrelevant events than DS*
➢ *Generate 80% less traces and save 86% time than DD*

References

[1] J. Wang et al. *Fast Reproducing Web Application Errors*, ISSRE’15.
[2] M. Hammoudi et al. *On the Use of Delta Debugging to Reduce Recordings and Facilitate Debugging of Web Applications*, FSE’16.
[3] G. Li et al. *SymJS: Automatic Symbolic Testing of JavaScript Web Applications*, FSE’14.