

Step-by-Step Delta Debugging

22. Oktober 2018

1 Minimizing Delta Debugging - Version 3

1.1 Rules

$$ddmin(c_f) = ddmin2(c_f, 2)$$

$$ddmin2(c_f, n) = \begin{cases} ddmin2(\Delta_i, 2) & \text{if } test(\Delta_i) = FAIL(\text{reduce to subset}) \\ ddmin2(\nabla_i, \max(n-1, 2)) & \text{else if } test(\nabla_i) = FAIL(\text{reduce to complement}) \\ ddmin2(c_f, \min(|c_f|, 2n)) & \text{else if } n < |c_f|(\text{increase granularity}) \\ c_f & \text{otherwise(done)} \end{cases}$$

1.2 Steps

1. Begin with the initialization of the Delta Debugging Algorithm:

$$ddmin(c_f) = ddmin2(c_f, 2)$$

2. Split the input c_f into n subsets Δ_i

- If one of the subsets Δ_i result in **Fail** then calculate $ddmin2(\Delta_i, 2)$ with the corresponding Δ_i and go to step 2. (*Case 1*)
- Else go to step 3.

3. Calculate the complement sets $\nabla_i = c_f \setminus \Delta_i$

- If one of the complement sets ∇_i result in **Fail** then calculate $ddmin2(\nabla_i, \max(n-1, 2))$ with the failing complement set and go to step 2. (*Case 2*)
- Else go to step 4.

4. Check $|c_f|$ (= length of c_f)

- If $|c_f| > n$ then calculate $ddmin2(c_f, \min(|c_f|, 2 * n))$ with the same c_f as before, but now you have to increase n and go to step 2. (*Case 3*)
- Else go to step 5.

5. Return c_f (*Case 4*)

1.3 Example

01234567 is the failure-inducing input string. Input 0 and 4 cannot be applied without each other. Input 2 reveals the bug.

1. Begin with Step 1 and initialize the algorithm:

$$ddmin(01234567) = ddmin2(01234567)$$

Step	n	Subset/Complement	Result	Rule	Action
0	-	$\Delta = 01234567$	FAIL		$ddmin2(01234567, 2)$

2. Split the input into $n = 2$ subsets Δ_i

Step	n	Subset/Complement	Result	Rule	Action
...
1	2	$\Delta_1 = 0123$	UNRES		
2	2	$\Delta_2 = 4567$	UNRES		

- There exist no $test(\Delta_i) = FAIL \rightarrow$ go to Step 3.

3. Split the input into $n = 2$ complement sets ∇_i

Step	n	Subset/Complement	Result	Rule	Action
...
3	2	$\nabla_1 = 4567$	UNRES		
4	2	$\nabla_2 = 0123$	UNRES		

- There exist no $test(\nabla_i) = FAIL \rightarrow$ go to Step 4.

4. Check the length of c_f :

$$\begin{aligned} |c_f| &> n \\ |01234567| &> 2 \\ 8 &> 2 \quad \checkmark \end{aligned}$$

- Increase the granularity: $ddmin2(c_f, \min(|c_f|, 2 * n)) \rightarrow ddmin2(01234567, 4)$
Case 3 and go to Step 2

Step	n	Subset/Complement	Result	Rule	Action
...
3	2	$\nabla_1 = 4567$	UNRES		
4	2	$\nabla_2 = 0123$	UNRES	(3)	$ddmin2(01234567, 4)$

5. Split the input into $n = 4$ subsets Δ_i

Step	n	Subset/Complement	Result	Rule	Action
...
5	4	$\Delta_1 = 01$	UNRES	(1)	ddmin2(23, 2)
6	4	$\Delta_2 = 23$	FAIL		
7	4	$\Delta_3 = 45$	UNRES		
8	4	$\Delta_4 = 67$	PASS		

- There exist a $test(\Delta_2) = FAIL \rightarrow \text{Case 1} \rightarrow$ call $ddmin2(\Delta_2, 2) = ddmin2(23, 2)$ and go to Step 2.

6. Split the input into $n = 2$ subsets Δ_i

Step	n	Subset/Complement	Result	Rule	Action
...
9	4	$\Delta_1 = 2$	FAIL	(1)	ddmin2(2, 2)
10	4	$\Delta_2 = 3$	PASS		

- There exist a $test(\Delta_1) = FAIL \rightarrow \text{Case 1} \rightarrow$ call $ddmin2(\Delta_2, 2) = ddmin2(2, 2)$ and go to Step 2.

7. $ddmin2(2, 2)$: $|c_f| = |2| = 1 \rightarrow$ Step 4 & **Case 4**

Step	n	Subset/Complement	Result	Rule	Action
...
9	4	$\Delta_1 = 2$	FAIL	(1), (4)	$ddmin2(2, 2) \rightarrow \text{return } 2$
10	4	$\Delta_2 = 3$	PASS		

2 Isolation Difference Delta Debugging

2.1 Rules

$$dd(c_s, c_f) = dd2(c_s, c_f, 2)$$

$$dd2(c'_s, c'_f, n) = \begin{cases} dd2(c'_s, c'_s \cup \Delta_i, 2) & \text{if } \exists\{i | test(c'_s \cup \Delta_i) = FAIL\} \\ dd2(c'_f \setminus \Delta_i, c'_f, 2) & \text{else if } \exists\{i | test(c'_f \setminus \Delta_i) = PASS\} \\ dd2(c'_s \cup \Delta_i, c'_f, \max(n-1, 2)) & \text{else if } \exists\{i | test(c'_s \cup \Delta_i) = PASS\} \\ dd2(c'_s, c'_f \setminus \Delta_i, \max(n-1, 2)) & \text{else if } \exists\{i | test(c'_f \setminus \Delta_i) = FAIL\} \\ dd2(c'_s, c'_f, \min(2n, |\Delta|)) & \text{else if } n < |\Delta| \\ (c'_s, c'_f) & \text{otherwise} \end{cases}$$

2.2 Steps

1. Begin with the initialization of the Delta Debugging Algorithm:

$$dd(c_s, c_f) = dd2(c_s, c_f, 2)$$

- The first successful input c_s is empty
 - The first failure input c_f is the whole input
2. Split the difference between c_s and c_f into n subsets Δ_i and calculate the unions of c_s and each Δ_i
 - If one input c_s together with Δ_i results in **FAIL** then calculate $dd2(c_s, c_s \cup \Delta_i, 2)$ with the corresponding Δ_i and go to Step 2. (*Case 1*)
 - Else go to Step 3.
 3. Calculate c_f without Δ_i for each subset Δ_i
 - If one input c_f without Δ_i results in **PASS** then calculate $dd2(c_f \setminus \Delta_i, c_f, 2)$ with the corresponding Δ_i and go to Step 2. (*Case 2*)
 - Else go to Step 4.
 4. Calculate the unions of c_s and each Δ_i
 - If one input c_s together with Δ_i results in **PASS** then calculate $dd2(c_s \cup \Delta_i, c_f, \max(n-1, 2))$ with the corresponding Δ_i and go to Step 2. (*Case 3*)
 - Else go to Step 5.
 5. Calculate c_f without Δ_i for each subset Δ_i
 - If one input c_f without Δ_i results in **FAIL** then calculate $dd2(c_s, c_f \setminus \Delta_i, \max(n-1, 2))$ with the corresponding Δ_i and go to Step 2. (*Case 4*)
 - Else go to Step 6.
 6. Check $|\Delta|$ (= length of the difference between c_s and c_f)
 - If $|\Delta| > n$ then calculate $dd2(c_s, c_f, \min(|\Delta|, 2 * n))$ with the same c_s and c_f as before, but now you have to increase n and go to step 2. (*Case 5*)
 - Else go to step 7.
 7. Return (c_s, c_f) (*Case 6*)
 - The Difference between the successful input c_s and the failing input c_f is the bug.

2.3 Example

01234567 is the failure-inducing input string. Input 0 and 4 must be part of the input in order to get a result different to UNRES. Input 2 reveals the bug. The test passes if the program gets an empty string as input.

1. Begin with Step 1 and initialize the algorithm

Step	n	c_s	c_f	Δ	TC	Test Input	Result	Rule	Action
1	-	-	01234567	01234567	c_s	-	PASS		
2	-				c_f	01234567	FAIL		dd2(-, 01234567, 2)

2. Split Δ (= difference between c_s and c_f) into $n = 2$ subsets Δ_i and calculate c_s combined with Δ_i

Step	n	c_s	c_f	Δ	TC	Test Input	Result	Rule	Action
...	
3	2	-	01234567	01234567	$c'_s \cup \Delta_1$	0123	UNRES		
4	2				$c'_s \cup \Delta_2$	4567	UNRES		

- There is no testcase $c_s \cup \Delta_i$ that result in FAIL \rightarrow go to Step 3.

3. Calculate c_f without Δ_i for each subset

Step	n	c_s	c_f	Δ	TC	Test Input	Result	Rule	Action
...	
3	2	-	01234567	01234567	$c'_s \cup \Delta_1$	0123	UNRES		
4	2				$c'_s \cup \Delta_2$	4567	UNRES		
5	2				$c'_f \setminus \Delta_1$	4567	UNRES		
6	2				$c'_f \setminus \Delta_2$	0123	UNRES		

- There is no testcase $c_f \setminus \Delta_i$ that result in PASS \rightarrow go to Step 4.

4. Check if there exists at least one $c_s \cup \Delta_i$ that results in PASS

Step	n	c_s	c_f	Δ	TC	Test Input	Result	Rule	Action
...	
3	2	-	01234567	01234567	$c'_s \cup \Delta_1$	0123	UNRES		
4	2				$c'_s \cup \Delta_2$	4567	UNRES		
5	2				$c'_f \setminus \Delta_1$	4567	UNRES		
6	2				$c'_f \setminus \Delta_2$	0123	UNRES		

- There is no testcase $c_s \cup \Delta_i$ that result in PASS \rightarrow go to Step 5.

5. Check if there exists at least one $c_f \setminus \Delta_i$ that results in **FAIL**

Step	n	c_s	c_f	Δ	TC	Test Input	Result	Rule	Action
...	
3	2	–	01234567	01234567	$c'_s \cup \Delta_1$	0123	UNRES		
4	2				$c'_s \cup \Delta_2$	4567	UNRES		
5	2				$c'_f \setminus \Delta_1$	4567	UNRES		
6	2				$c'_f \setminus \Delta_2$	0123	UNRES		

- There is no testcase $c_f \setminus \Delta_i$ that result in **FAIL** → go to Step 6.

6. Check the length of Δ :

$$\begin{aligned}
 |\Delta| &> n \\
 |01234567| &> 2 \\
 8 &> 2 \quad \checkmark
 \end{aligned}$$

- Increase the granularity: $dd2(c_s, c_f, \min(|\Delta|, 2 * n)) \rightarrow dd2(-, 01234567, 4)$
Case 5 and go to Step 2

Step	n	c_s	c_f	Δ	TC	Test Input	Result	Rule	Action
...	
3	2	–	01234567	01234567	$c'_s \cup \Delta_1$	0123	UNRES		
4	2				$c'_s \cup \Delta_2$	4567	UNRES		
5	2				$c'_f \setminus \Delta_1$	4567	UNRES		
6	2				$c'_f \setminus \Delta_2$	0123	UNRES	(5)	dd2(-, 01234567, 4)

7. Split Δ into $n = 4$ subsets Δ_i and calculate c_s combined with Δ_i

Step	n	c_s	c_f	Δ	TC	Test Input	Result	Rule	Action
...	
7	4	–	01234567	01234567	$c'_s \cup \Delta_1$	01	UNRES		
8	4				$c'_s \cup \Delta_2$	23	UNRES		
9	4				$c'_s \cup \Delta_3$	45	UNRES		
10	4				$c'_s \cup \Delta_4$	67	UNRES		

- There is no testcase $c_s \cup \Delta_i$ that result in **FAIL** → go to Step 3.

8. Calculate c_f without Δ_i for each subset

Step	n	c_s	c_f	Δ	TC	Test Input	Result	Rule	Action
...	
7	4	–	01234567	01234567	$c'_s \cup \Delta_1$	01	UNRES		
8	4				$c'_s \cup \Delta_2$	23	UNRES		
9	4				$c'_s \cup \Delta_3$	45	UNRES		
10	4				$c'_s \cup \Delta_4$	67	UNRES		
11	4				$c'_f \setminus \Delta_1$	234567	UNRES		
12	4				$c'_f \setminus \Delta_2$	01 4567	PASS	(2)	dd2(014567, 01234567, 2)
13	4				$c'_f \setminus \Delta_3$	0123 67	UNRES		
14	4				$c'_f \setminus \Delta_4$	012345	FAIL	(4)	

- There is a testcase $c_f \setminus \Delta_2$ that result in PASS \rightarrow **Case 2** \rightarrow call $dd2(c_f \setminus \Delta_2, c_f, 2) = (014567, 01234567, 2)$.
- Check if there exit any testcases that fullfil Case 3 or 4 and identify them in the table too!
- Go to Step 2.

9. Split Δ into $n = 2$ subsets Δ_i and calculate c_s combined with Δ_i

Step	n	c_s	c_f	Δ	TC	Test Input	Result	Rule	Action
...	
15	2	01 4567	01234567	23	$c'_s \cup \Delta_1$	012 4567	FAIL	(1)	dd2(014567, 0124567, 2)
16	2				$c'_s \cup \Delta_2$	01 34567	PASS		

- There is a testcase $c_s \cup \Delta_1$ that result in FAIL \rightarrow **Case 1** \rightarrow call $dd2(c_s, c_s \cup \Delta_1, 2) = dd2(014567, 0124567, 2)$.
- Calculate $c_f \setminus \Delta_i$ and check if there exist any testcases that fullfil Case 2, 3 or 4 and identify them in the table too!
- Go to Step 2.

Step	n	c_s	c_f	Δ	TC	Test Input	Result	Rule	Action
...	
15	2	01 4567	01234567	23	$c'_s \cup \Delta_1$	012 4567	FAIL	(1)	dd2(014567, 0124567, 2)
16	2				$c'_s \cup \Delta_2$	01 34567	PASS	(3)	
17	2				$c'_f \setminus \Delta_1$	01 34567	PASS	(2)	
18	2				$c'_f \setminus \Delta_2$	012 4567	FAIL	(4)	

10. $dd2(014567, 0124567, 2)$: $|\Delta| = |2| = 1 \rightarrow$ Step 7 & **Case 6**

Step	n	c_s	c_f	Δ	TC	Test Input	Result	Rule	Action
...	
19	2	01 4567	012 4567	2	-	-	-	(6)	(014567, 0124567)