(draft) Clue complexity

Abstract. In this article we propose an advanced source code complexity measure [Blo19] from the perspective of static path testing with the conjecture of self-suffcient enclosed scopes, the measure enfolds elementary conditions, nested and cyclic formations.

1 Introduction

A Progam consist of a sequences of instructions. The instructions can be categorized into syntax, logic and arithmetic determining the program control flow.

Software complexity is related with modularity, coupling and cohesion hence quality. 40% to 80% of software costs are emerging due to maintenance and approximatly 40% on fixing defects [SG11].

Analysing source code complexity helps to identify risk, finds potential defects to test critical functionality in detail, increase quality, cohesion and decrease maintainance [RB11].

2 Existing measures

Cyclomatic complexity (CC) has been widely discussed by various authors. The most used metric has been formulated by Thomas J. McCabe in 1976 [McC76].

$$m = e - n + 2 \tag{1}$$

Where, e = the amount of edges. n = the amount of nodes.

CC is linear in it's nature and correlation with lines of source code (LOC), which is why Graylin JAY et al. [al.09] are suggesting to implement LOC as complexity measure.

Mir Muhammd Suleman Sarwar et al. [MMSS13] are pointing out that it neither takes into account the difference between combined decisions, elementary conditions nor repeating structures. Their adaptation of CC includes loop iterations

$$V(G)* = V(G) + \prod_{i=1}^{n} P_i$$
$$P_i = U_i - L_i + 1$$

Where, P_i = No. of iterations of ith loop. U_i = upper bound of ith loop. L_i = lower bound of ith loop and $V(G)*$ = adjusted cyclomatic complexity for any control flow graph "G". The measure combines control flow and statements exercised.

Tevfik et al. [LB15] are providing a computable measure: asymptotic path complexity covering nested structures and loops.

NPATH [Nej88] from Brian A. Nejmeh. facilitates acyclic path covering $O(2^n)$ neglecting unfolded conditions (q, Table 1, 2).

2.1 Test covering

Test covering metrics are providing quantitative representation of tested structures. The approach is to maximize covering while minimising testing effort. Each condition branches the flow into two control sub-paths, and determines the progression of the programm. The path of an isolated decision d_j thereby is given due to decisions and conditions (MC/DC) [RB11]. Let's define a condition c_j and it's path $\gamma : C \times \Sigma \to C^*$

$$\Sigma = \{true, false\}$$

$$D \subseteq \{C^*, S, E\}, \quad d_h \in D, \quad d_h \mapsto \Sigma$$

The transition function $alpha$

$$\alpha : D \times \Sigma^* \to \Sigma \times D$$

The condition

$$C = \{c_0, c_1, c_2, c_3, ..., c_n\}, c_i \mapsto \Sigma$$

$$\varepsilon : C \to \Sigma \times C$$

and the transition $\vdash$

$$\vdash \subseteq (\Sigma^* \times D \times \Sigma) \times (\Sigma^* \times D \times \Sigma)$$

The possible combinations of conditions on γ are arising through c_j's preceeding and succeeding condition flow. An acyclic transition path through c_j is described due to passing each condition once

$$
\begin{aligned}
B^{T}_{\lambda j} &= (v_h, d_h, b_h), \ldots \vdash (v_k, d_k, b_k) \vdash^{*} (v_m, d_m, b_m) \\
&= (c_h, b_h), \ldots \vdash (c_j, b_j) \vdash^{*} (c_n, b_n)
\end{aligned}
$$

where

$$\alpha(d_m, v_m) \mapsto E$$

with the jacket

$$B_{\lambda} = (b_h, v_h, d_h) \vdash^{*} (b_m, v_m, d_m)$$

3 Hypothesis acyclic complexity

Instead of combining arithmetic and logic we propose a metric [Blo19] reflecting the logical test effort of modified condition/decision path covering [al.01]. The concept of basic paths described by [McC76] neglects loops and nesting. 100% sub-path combination covering q is intricate due to exponential effort (Table 1). Conditions are providing the logical structure of programs (Listing 1-16). Loops and recursions are construed due to their boundary [RB11] and invariant, which is why their quantitative iteration increases probability of condition covering not condition path length. Let's define the inductive start of independent

$$
\begin{aligned}
\alpha_i =& (c_0, true) \rightarrow (c_1), (c_0, false) \rightarrow (c_1), \\
& (c_1, true) \rightarrow (E), (c_1, false) \rightarrow (E)
\end{aligned}
$$

and dependent branches

$$
\begin{aligned}
\alpha_d =& (c_0, true) \rightarrow (E), (c_0, false) \rightarrow (c_1), \\
& (c_1, true) \rightarrow (E), (c_1, false) \rightarrow (E)
\end{aligned}
$$

Isolated condition paths are $log(2^n)$ subset of all q paths. The spanning tree of the first branch expands four leaves; the last three.

Nesting is diminishing the amount of paths, the effort of combinatorial parameter testing is unaffected. The fallacy of subjective perceived complexity increases contrary due to remembering previous conditions (Table 2).

Fig. 1. Upper Bound

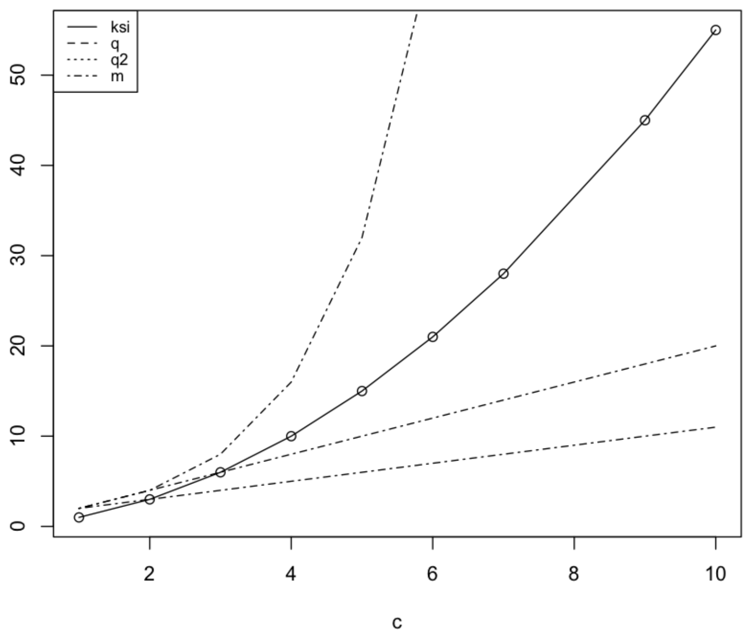

Table 1. Condition formation bound

	$O(n)$	$O(2n)$	$O(n^2)$	$O(2^n)$
$\|C\|$	m	q_2	η	q
1	2	$2-2$	1	$2-2$
2	3	$3-4$	$(1+\frac{1}{2}i)-(3+0i)$	$3-4$
3	4	$4-6$	$(1+\frac{3}{4}i)-(6+0i)$	$4-8$
4	5	$5-8$	$(1+1i)-(10+0i)$	$5-16$
5	6	$6-10$	$(1+\frac{7}{6}i)-(15+0i)$	$6-32$
6	7	$7-12$	$(1+\frac{4}{3}i)-(21+0i)$	$7-64$
7	8	$8-14$	$(1+\frac{35}{24}i)-(28+0i)$	$8-128$
8	9	$9-16$	$(1+\frac{19}{12}i)-(36+0i)$	$9-256$
9	10	$10-18$	$(1+\frac{101}{60}i)-(45+0i)$	$10-512$
...				
19	20	$20-38$	$(1+\frac{5869}{2520}i)-(190+0i)$	$20-524288$
...				
49	50	$50-98$	$(1+\frac{29412190771}{8923714800}i)-(1225+0i)$	$50 \sim 5.6 * 10^{14}$

Listing 1. plain singular

```
if (−1<1) { }
```

$\eta = 1 + 0i$

Listing 2. orthogonal dual

```
if (3>1 && 0<k) { }
```

$\eta = 1 + \frac{1}{3}i$

Listing 3. orthogonal dual

```
if (2>1 || 0<k) { }
```

$\eta = 1 + \frac{1}{2}i$

Listing 4. tertiary orthogonal

```
if (2>1 && 7>k && 10>=n) { }
```

$\eta = 1 + (\frac{1}{3} + \frac{1}{5})i$

Listing 5. tertiary

```
if (2>1 && 0<k || 4>n) { }
```

$\eta = 1 + (\frac{2}{3} + \frac{2}{4})i$

Listing 6. tertiary parallel identical scope

```
if (2>1 || 0<k || 4>n) { }
```

$\eta = 1 + (\frac{1}{2} + \frac{1}{4})i$

Listing 7. tertiary parallel

```
if (0<1 || 9>k && 1<n) { }
```

$\eta = 1 + (\frac{2}{2} + \frac{2}{5})i$

Listing 8. orthogonal dual nested

```
if (0<1) {
    if (2>k) { }
}
```

$\eta = 1 + 1 + 0i$

Listing 9. tertiary orthogonal nested

```
if (0<1 && 4>k) {
  if (11>n) { }
}
```

$\eta = 1 + \frac{1}{3}i + 1$

Listing 10. tertiary orthogonal nested

```
if (2>1) {
  if (8>k) {
    if (0<n) { }
  }
}
```

$\eta = 1 + 1 + 1 + 0i$

Listing 11. parallel

```
if (2>1) { }
/* 2 */
if (8>k) { }
```

$\eta = 1 + 2 + 0i$

Listing 12. dual parallel

```
if (0<1) { }
/* 2 */
if (9>k && 1<n) { }
```

$\eta = 1 + 2 + \frac{1}{3}i$

Listing 13. singular nested parallel

```
if (0<1) {
  if (4>k) { }
  /* 2 */
  if (8>n) { }
}
```

$\eta = 1 + 1 + 2 + 0i$

Listing 14. orthogonal singular parallel

```
if (0<1) {
  if (8>k) { }
}
/* 2 */
if (0<n) { }
```

$\eta = 1 + 1 + 2 + 0i$

Listing 15. dual parallel

```
if (0<1) { }
/* 2 */
if (9>k) {
    if (1<n) { }
}
```

$\eta = 1 + 2 + 1 + 0i$

Listing 16. tertiary parallel different scope

```
if (−4<1) { }
/* 2 */
if (4<k) { }
/* 3 */
if (8>n) { }
```

$\eta = 1 + 2 + 3 + 0i$

Fig. 2. $\frac{m}{q}$

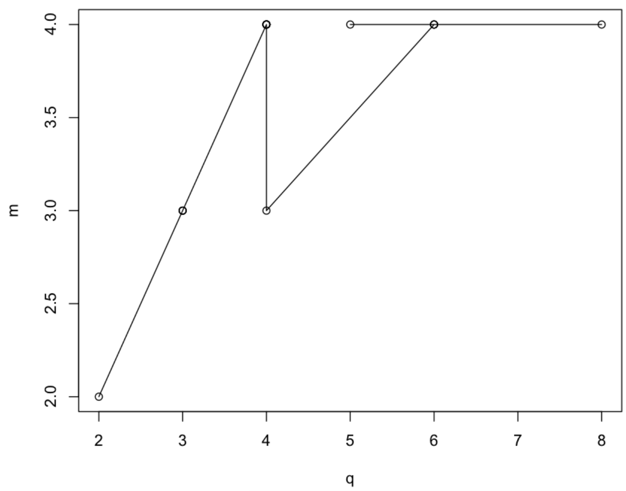

Fig. 5. η-complex

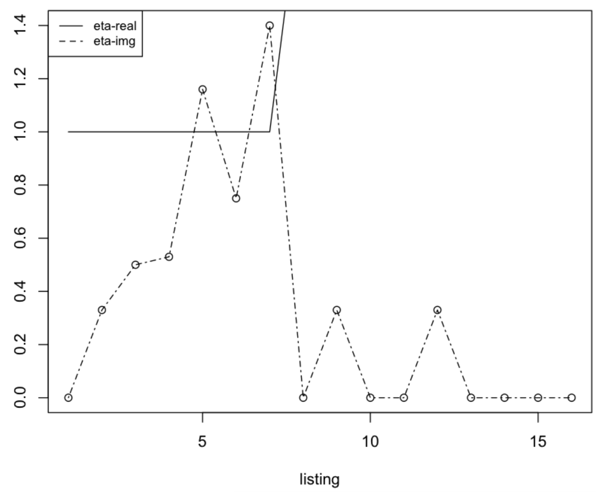

Fig. 3. $\frac{loc}{q}$

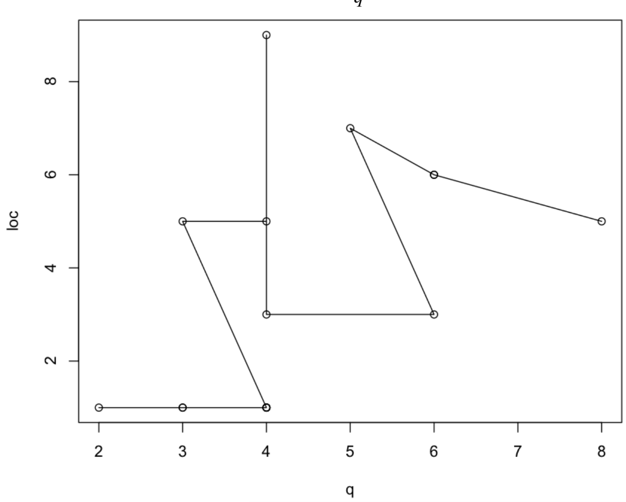

Fig. 4. $\frac{q_2}{q}$

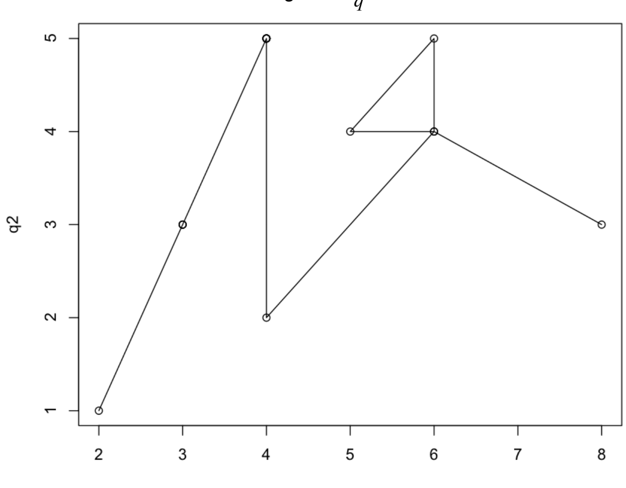

Table 2. Formation complexity

Listing	$\|C\|$	cols	loc	m	q_2	η	$\|\eta\| \approx$	q
1	1	0	1	2	1	$1+0i$	1	2
2	2	0	1	3	3	$1+\frac{1}{3}i$	1.05	3
3	2	0	1	3	3	$1+\frac{1}{2}i$	1.11	3
4	3	0	1	4	5	$1+\frac{8}{15}i$	1.13	4
5	3	0	1	4	5	$1+\frac{7}{6}i$	1.53	4
6	3	0	1	4	5	$1+\frac{3}{4}i$	1.56	4
7	3	0	1	4	5	$1+\frac{7}{5}i$	1.72	4
8	2	0	5	3	3	$2+0i$	2	3
9	3	0	5	4	5	$2+\frac{1}{3}i$	2.03	4
10	3	0	9	4	5	$3+0i$	3	4
11	2	1	3	3	2	$3+0i$	3	4
12	3	1	3	4	4	$3+\frac{1}{3}i$	3.02	6
13	3	1	7	4	4	$4+0i$	4	5
14	3	1	6	4	5	$4+0i$	4	6
15	3	1	6	4	4	$4+0i$	4	6
16	3	2	5	4	3	$6+0i$	6	8

Fig. 6. $\frac{listing}{q}$

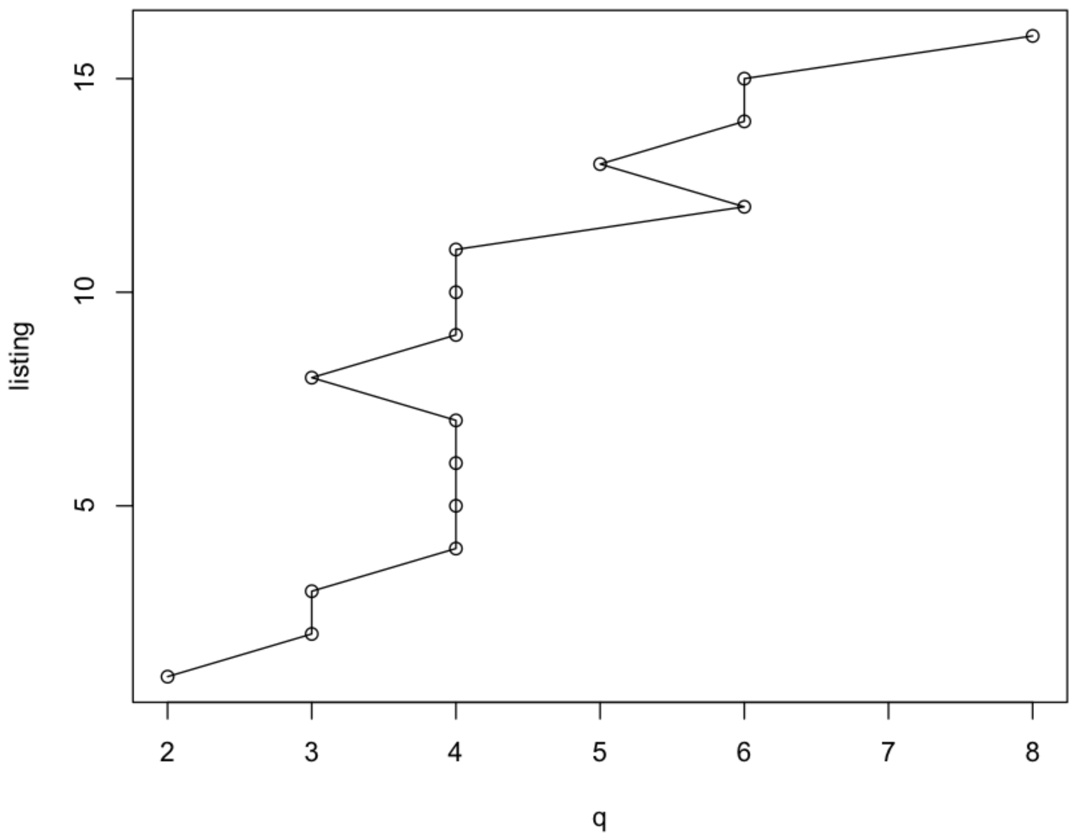

Fig. 7. $\frac{q}{\eta}$

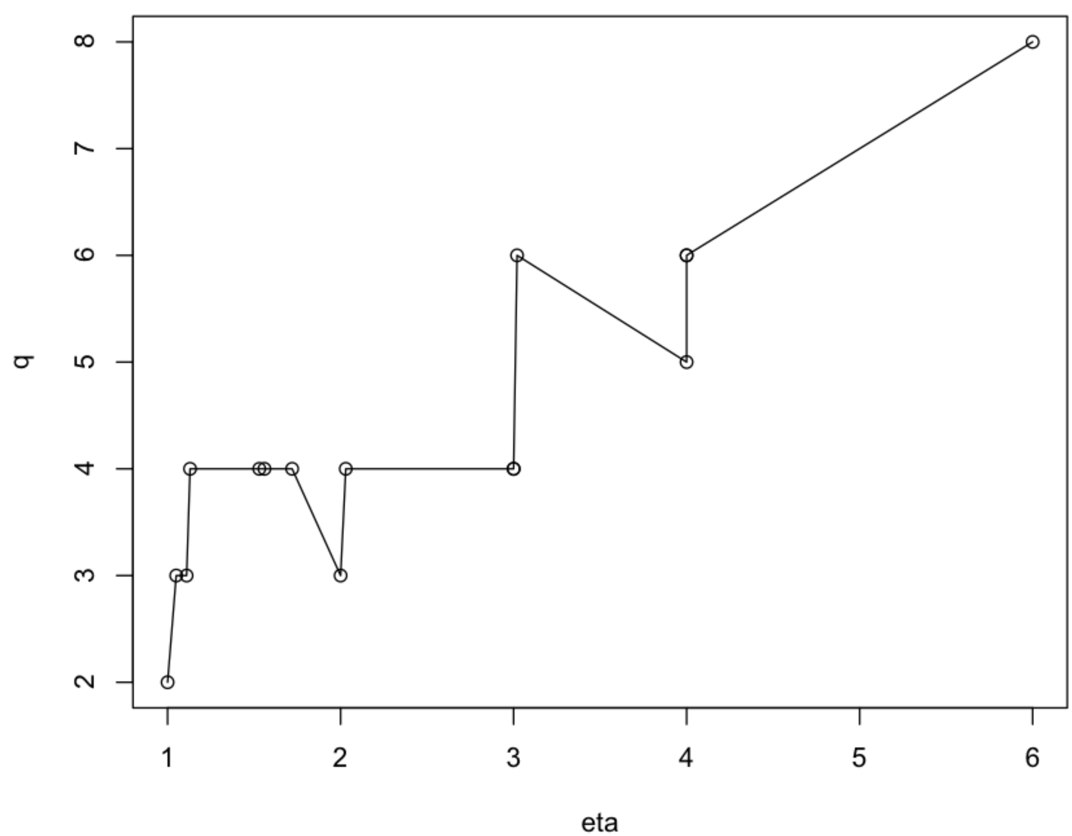

Table 3. α_i flow matrix, $q_2(\alpha_i) = 2$

	s	c_0	c_1	e
s		t		
c_0			(f,t)	
c_1				(f,t)
e				

Table 4. α_d flow matrix, $q_2(\alpha_d) = 3$

	s	c_0	c_1	e
s		t		
c_0			f	t
c_1				(f,t)
e				

The exponential amount of modified condition path combinations q is parsed into control flow q_2.

$$\begin{aligned} n &= |C|, \quad n \in \mathbb{N} \\ 2^n &<= q <= n+1 \end{aligned} \tag{2}$$

When every condition is isolated the amount of condensed edges (enclosed scope paths) q_2 in the condition flow matrix represents an intuitive measure (Table 2, 1, 3, 4).

The lower and upper bounds of possible isolated condensed condition paths are

$$\begin{aligned} &2*n >= q_2 >= n+1 \\ &C_e \subset C; \quad m_u = |C_e| + n \end{aligned} \tag{3}$$

with $C_e \subset C$ nested conditions. The metric m_u is undifferentiated regarding condition formation. Subpath combinations caused from parallel decision scope (flat-style) are neglected, as well as the subjective memory of decisions, therefor we define the nesting function φ

The dimorphism φ of decision d_i depends on the collocation m_i and conditions c_k

$$\xi \in \mathbb{C}\,; \quad c_\wedge \subset d_i\,; \quad c_\vee \subset d_i; \quad n_i = \left\lceil \frac{n}{2} \right\rceil$$

$$m < |\xi| < \frac{n(n+1)}{2}$$

$$\xi = \sum_{i=0}^{n-1} \xi_{i+1} - \xi_i = \sum_{i=0}^{n} \varphi(d_i)$$

$$\begin{aligned} \varphi(d_i) &= m_i + \varepsilon_i^* \qquad\qquad \varepsilon_i^* \in O(n) \\ &\equiv m_i + \left(\sum_{k=1}^{n} \begin{cases} \frac{1+|c_\wedge|}{2*|n_i-k|+3}, & \text{if } c_k\text{=and} \\ \frac{1+|c_\vee|}{2*|n_i-k|+2}, & \text{if } c_k\text{=or} \end{cases} \right) i \end{aligned} \tag{4}$$

$$\xi \equiv \sum_{i=0}^{l} \sum_{k=1}^{m_i} k + \sum_{i=0}^{n} \varepsilon_i^*$$

$$\eta = \sum_{i=0}^{n} o_i + e_i \tag{5}$$

Each triangular number is a mile on the path, the convex harmonic equivalence distinguishes according to positive perception of resurfacing invariant boundary test effort of self-sufficient spheres (Table 2, Listing17, Listing18).

Listing 17. Condition example

```
if (a && b && c && d) { }
```
$\eta = 1 + (\frac{1}{5} + \frac{1}{3} + \frac{1}{5})i = 1 + \frac{11}{15}i$

```
if (a || b || c || d) { }
```
$\eta = 1 + (\frac{1}{4} + \frac{1}{2} + \frac{1}{4})i = 1 + \frac{5}{3}i$

```
if (a && b && c || d) { }
```
$\eta = 1 + (\frac{2}{5} + \frac{2}{3} + \frac{3}{4})i = 1 + \frac{109}{60}i$

```
if (a || b && c && d) { }
```
$\eta = 1 + (\frac{3}{4} + \frac{2}{3} + \frac{2}{5})i = 1 + \frac{109}{60}i$

```
if (a || b && c || d) { }
```
$\eta = 1 + (\frac{2}{4} + \frac{3}{3} + \frac{2}{4})i = 1 + \frac{2}{1}i$

```
if (a || b || c && d) { }
```
$\eta = 1 + (\frac{2}{4} + \frac{2}{2} + \frac{3}{5})i = 1 + \frac{21}{10}i$

```
if (a && b || c || d) { }
```
$\eta = 1 + (\frac{3}{5} + \frac{2}{2} + \frac{2}{4})i = 1 + \frac{21}{10}i$

```
if (a && b || c && d) { }
```
$\eta = 1 + (\frac{2}{5} + \frac{3}{2} + \frac{2}{5})i = 1 + \frac{23}{10}i$

Listing 18. Collocation example

```
if (a) {
    if (b || c) {
    }
    if (d && e) {
    }
}
```
$\eta = (1 + 0i) + (1 + \frac{1}{2}i) + (2 + \frac{1}{3}i) = 4 + \frac{5}{6}i$

```
if (a) {
    if (b && c) {
    }
    if (d) {
    }
}
if (e) {
}
```
$\eta = (1 + 0i) + (1 + \frac{1}{3}i) + (2 + 0i) + (2 + 0i) = 6 + \frac{1}{3}i$

```
if (a) {
    if (b) {
        if (c) {
        }
        if (d && e) {
        }
    }
}
```
$\eta = (1 + 0i) + (1 + 0i) + (1 + 0i) + (2 + (\frac{1}{3})i) = 5 + \frac{1}{3}i$

```
if (a) {
    if (b) {
    }
    if (c) {
    }
    if (d) {
    }
    if (e) {
    }
}
```
$\eta = (1 + 0i) + (1 + 0i) + (2 + 0i) + (3 + 0i) + (4 + 0i) = 11 + 0i$

4 Theorem

The poise of decisions n is restricting the exponential path complexity of collocations $m < n$.

5 Conclusion

The suggested advanced extensional quadratic complexity measure η (5) expresses the logical test effort of condensed paths, diminishing unapparent complex patterns quantifying control flow into complex numbers.

References

[al.01] AL., Kelly J. H.: A Practical Tutorial on Modified Condition / Decision Coverage. In: *NASA* (2001)

[al.09] AL., Graylin J.: Cyclomatic Complexity and Lines of Code: Empirical Evidence of a Stable Linear Relationship. In: . *Software Engineering & Applications* 2 (2009), June, S. 137–143

[Blo19] BLOEMENDAL, Jannis: Clue complexity. (2019), April, NL Rotterdam. – http://github.com/jbloemendal/clue

[LB15] LUCAS BANG, Tevfik B. Abdulbaki Aydin A. Abdulbaki Aydin: Automatically computing path complexity of programs. In: *ESEC/SIGSOFT FSE 2015* (2015), S. 61–72

[McC76] MCCABE, Thomas J.: A complexity measure. In: *IEEE* (1976), Dec, Nr. 4, S. 308–320

[MMSS13] MIR MUHAMMD SULEMAN SARWAR, Ibrar A. Sara Shahzad S. Sara Shahzad: Cyclomatic Complexity. In: *IEEE* 1 (2013), Jan, Nr. 5, S. 274–279

[Nej88] NEJMEH, Brian A.: NPATH: A Measure of Execution Path Complexity and Its Application. In: *ACM* 31(2) (1988), S. 188–200

[RB11] REX BLACK, Jamie M.: *Advanced Software Testing Vol. 3*. Santa Barbara, CA : rookynook, 2011

[SG11] SOUMIN GHOSH, Prof. (Dr.) Ajay R. Sanjay Kumar Dubey D. Sanjay Kumar Dubey: Comparative Study of the Factors that Affect Maintainability. In: *International Journal on Computer Science and Engineering (IJCSE)* 3 (2011), Dec, Nr. 12, S. 3763–3769