

Generalised approach to software structure metrics

by P.M. van den Broek and K.G. van den Berg

The aim of this paper is to generalise the Fenton–Whitty–Kaposi (FWK) approach to structure software metrics by considering arbitrary sets of decomposition operations for flowgraphs. In the FWK approach, decomposition of flowgraphs is unique, but the number of associated metric functions is not finite and these functions are all independent. In general, the decomposition of flowgraphs is not unique, which leads to constraints on the associated metric functions. Here we derive these constraints explicitly for two special cases, where we consider only the two operations sequencing and nesting as decomposition operations. It is shown that the two resulting classes of structure metrics are contained in the class of recursive structure metrics of the FWK approach.

1 Introduction

In the Fenton–Whitty–Kaposi (FWK) approach to software structure metrics [1–3], software is modelled in terms of flowgraphs. There are two operations on flowgraphs, sequencing and nesting, with which flowgraphs can be built from smaller flowgraphs. Flowgraphs that cannot be constructed from smaller flowgraphs with these operations are called primes. A software metric is said to be a hierarchical metric with respect to a prescribed set of decomposition operations if it can be defined inductively on the decomposition trees of the flowgraphs. An important subclass of the hierarchical metrics, recursive metrics, has been identified by Prather [4]. For a structure metric, there is a function associated with each decomposition operation.

In the FWK approach, these decomposition operations are the decomposition of a sequential flowgraph as a sequence of nonsequential ones and the decomposition of a nonsequential flowgraph as a simultaneous nesting of its maximal proper one-entry subflowgraphs on a prime. So for each integer with $n \geq 2$, there are two decomposition operations, and thus two associated functions, decomposition as a sequence of n nonsequential flowgraphs, and nesting of $(n - 1)$ flowgraphs on a prime. The advantage of this choice of the decomposition operations is that each flowgraph has a unique hierarchical decomposition into primes. As a consequence, the associated functions are all independent of each other.

Although mathematically elegant, from a practical point of view it is unrealistic that the functions associated with sequencing different numbers of nonsequential flowgraphs are all independent (note that Fenton and Whitty [3] provide some *ad hoc* constraints, which restrict this independency). Indeed, if the metric value of a sequence of two nonsequential flowgraphs is obtained by the addition of the metric values of the components, the same is expected to hold for the metric value of a sequence of an arbitrary number of nonsequential flowgraphs. The same remark holds for the functions associated with nesting different numbers of flowgraphs on a prime. What is missing is some uniformity property, relating to the sequencing (and nesting, respectively) of different numbers of flowgraphs.

The aim of this paper is to generalise the FWK approach, by considering arbitrary sets of decomposition operations. In general, the decomposition of flowgraphs is not unique. The requirement that different decompositions of the same flowgraph give the same metric values leads to constraints on the associated metric functions. We derive these constraints explicitly for two special cases.

In the first special case, we choose as decomposition operations the operations of sequencing and nesting themselves, without any restriction. As there are only two decomposition operations, there are only two associated functions. The class of structure metrics defined in this way, which we call strong recursive metrics (or strong metrics), is a subclass of the class of recursive metrics of the FWK approach. We also identify a subclass of strong metrics, called symmetric strong (recursive) metrics. We argue that 'almost all' strong metrics are symmetric. The reason for introducing symmetric strong metrics is that the constraints for the associated functions to define a symmetric strong metric are particularly simple and elegant. Strong metrics, in contrast to recursive metrics, satisfy a uniformity property, relating the sequencing (resp. nesting) of different numbers of flowgraphs. However, the class of strong metrics can be criticised for being too small, as it does not contain the maximal depth of nesting metric.

Therefore, in our second special case, we consider as decomposition operations the operations of sequencing and nesting, with the restrictions that a flowgraph can only be decomposed as a nesting if it is a nonsequential and the nested flowgraph is a proper maximal one-entry subflowgraph. Note that these restrictions belong to the restrictions of recursive metrics. The class of structure metrics defined this way we call semi-strong (recursive) metrics. We derive the constraints for the two associated functions to define a semi-strong metric. Semi-strong

metrics are shown to satisfy the same uniformity property as strong metrics, and the maximal depth of nesting metric is a semi-strong metric.

The four classes of metrics introduced above satisfy the following inclusion relations:

- symmetric strong metrics \subset strong metrics
- \subset semi-strong metrics \subset recursive metrics

2 Flowgraphs

In this Section, we briefly recapitulate the theory of flowgraphs (Fenton [1]; Fenton and Kaposi [2]). We start with the definition of a flowgraph:

Definition

A flowgraph is a 3-tuple (G, a, z) , where G is a directed graph, and a and z are two different nodes of G , called *startnode* and *stopnode*, respectively, such that

- for each node x of G there is a path in G from a to z via x .
- The outdegree of z is 0

The condition that the start node and the stop node of a flowgraph are different is an addition to the previous definition of a flowgraph [1–3], whose only implication is that the graph with only one node and no edges is not a flowgraph.

The following two definitions specify operations on flowgraphs.

Definition

If $F_1 = (G_1, a_1, z_1)$ and $F_2 = (G_2, a_2, z_2)$ are flowgraphs, then the sequence $F_1; F_2$ of F_1 and F_2 is the flowgraph $(G_1; G_2, a_1, z_2)$, where $G_1; G_2$ is the directed graph obtained from the union of G_1 and G_2 by identifying the nodes z_1 and a_2 .

Definition

If $F_1 = (G_1, a_1, z_1)$ and $F_2 = (G_2, a_2, z_2)$ are flowgraphs and x is a node of G_1 with outdegree 1 (called a *procedure node*), then the nesting $F_1(F_2$ on $x)$ is the flowgraph $(G_1(G_2$ on $x), a_1, z_1)$, where $G_1(G_2$ on $x)$ is the directed graph obtained from the union of G_1 and G_2 by deleting the edge whose source is x , identifying x and a_2 , and identifying z_2 and the successor of x .

In the remainder of this paper, we write $F_1(F_2)$ instead of $F_1(F_2$ on $x)$. This is no restriction, as for the metrics we consider it is unimportant onto which procedure node a nesting takes place.

Definition

The flowgraph $F_2 = (G_2, a_2, z_2)$ is a *subflowgraph* of the flowgraph $F_1 = (G_1, a_1, z_1)$ if G_2 is a subgraph of G_1 and z_2 is the source of all edges from G_2 to $G_1 \setminus G_2$.

Definition

The subflowgraph $F_2 = (G_2, a_2, z_2)$ of the flowgraph $F_1 = (G_1, a_1, z_1)$ is a *one-entry subflowgraph* if

- the target of each edge from $(G_1 \setminus G_2) \cup \{z_2\}$ to G_2 is either a_2 or z_2 .

- if a_1 belongs to G_2 , then $a_1 = a_2$ or $a_1 = z_2$.

Definition

The subflowgraph $F_2 = (G_2, a_2, z_2)$ of the flowgraph $F_1 = (G_1, a_1, z_1)$ is a *proper subflowgraph* if $G_1 \neq G_2$ and F_2 is not equal to P_1 , the flowgraph with two nodes and one edge.

Definition

The proper one-entry subflowgraph F_2 of the flowgraph F_1 is a *maximal one-entry subflowgraph* of F_1 if there exists no proper one-entry subflowgraph F_3 of F_1 such that F_2 is a proper one-entry subflowgraph of F_3 .

The following two theorems show the way flowgraphs are decomposed into a hierarchy of primes in the FWK approach.

Theorem

Each flowgraph F can be written uniquely as a sequence of nonsequential flowgraphs $F_1; F_2; \dots; F_n$.

Theorem

Each nonsequential flowgraph F can be written as a simultaneous nesting $F(F_1, F_2, \dots, F_n)$, where $\{F_1, F_2, \dots, F_n\}$ is the unique set of maximal proper one-entry subflowgraphs of F . Moreover, F' is a prime.

3 Generalised approach to software structure metrics

In our generalised approach to software structure metrics, we first identify a set X of decomposition operations for flowgraphs.

Definition

A *decomposition operation* is a set of tuples, each of which consist of a flowgraph and a list of flowgraphs.

If A is a decomposition operation and $(F, [F_1, F_2, \dots, F_n])$ belongs to A , then F is said to have an A decomposition into $[F_1, F_2, \dots, F_n]$.

Definition

A metric μ is a *structure metric with respect to a set X of decomposition operations* if, and only if, there exists a function f_A for each $A \in X$ such that

$$\mu(F) = f_A(\mu(F_1), \mu(F_2), \dots, \mu(F_n))$$

for all $(F, [F_1, F_2, \dots, F_n])$ in A . (1)

We want a structure metric to be completely defined by the associated functions f_A and the value of the metric for the primes. This is ensured when we demand that each flowgraph can be hierarchically decomposed into primes using the decomposition operations.

However, we do not demand that there are no decompositions for primes, nor that each flowgraph which is not a prime has a unique decomposition. As a consequence, given functions f_A for each $A \in X$ and the metric values of the primes, eqn. 1 does not necessarily have a solution for μ . Thus, the second step in our generalised approach consists of deriving the constraints on the associated functions

f_A that are necessary and sufficient for the existence of a corresponding structure metric with respect to X , i.e. the existence of a solution μ of eqn. 1.

We now consider how the FWK approach fits into this general framework. In this approach, the set of decomposition operations X is given by

$$X = \{Seq_n | n \geq 2\} \cup \{Nest_n | n \geq 2\} \quad (2)$$

$$Seq_n = \{(F_1; F_2; \dots; F_n, [F_1, F_2, \dots, F_n]) | F_1, F_2, \dots, F_n \text{ nonsequential}\} \quad (3)$$

$$Nest_n = \{(F_1(F_2, F_n), [F_1, F_2, \dots, F_n]) | F_1 \text{ prime; none of } F_1, F_2, \dots, F_n \text{ equal to } P_1\} \quad (4)$$

We call the structure metrics with respect to X recursive metrics. In order that a metric μ is a recursive metric, there should exist functions f_n and g_n ($n \geq 2$), associated with Seq_n and $Nest_n$, respectively, such that

$$\mu(F_1; F_2; \dots; F_n) = f_n(\mu(F_1), \mu(F_2), \dots, \mu(F_n)) \quad \text{if } F_1, F_2, \dots, F_n \text{ are all nonsequential} \quad (5)$$

and

$$\mu(F_1(F_2, \dots, F_n)) = g_n(\mu(F_1), \mu(F_2), \dots, \mu(F_n)) \quad \text{if } F_1 \text{ is a prime and } F_1, F_2, \dots, F_n \text{ are all unequal to } P_1 \quad (6)$$

By applying the decomposition operations Seq_n and $Nest_n$, each flowgraph can be hierarchically decomposed into primes, and primes cannot be decomposed. The decomposition of a flowgraph is not unique, as in $F_1(F_2, \dots, F_n)$ the order of F_2, \dots, F_n is irrelevant. This leads to the constraints that the functions g_n should be symmetric in all but their first arguments. If the metric values of the primes are given, and if functions f_n and g_n are given that satisfy the above constraint, then eqns. 5 and 6 define the recursive metric μ unambiguously. Thus, there are no constraints that impose relations between the associated functions.

4 Strong software metrics

We define strong metrics to be the structure metrics for which the decomposition operations are the operations of sequencing and nesting, without any restriction. Thus, our set of decomposition operations is given by

$$X_{strong} = \{Seq, Nest\} \quad (7)$$

$$Seq = \{(F_1; F_2, [F_1, F_2])\} \quad (8)$$

$$Nest = \{(F_1(F_2), [F_1, F_2])\} \quad (9)$$

A mapping μ from flowgraphs to numbers is a *strong metric* if, and only if, there exist two functions f and g such that

$$\mu(F_1; F_2) = f(\mu(F_1), \mu(F_2)) \quad (10)$$

$$\mu(F_1(F_2)) = g(\mu(F_1), \mu(F_2)) \quad (11)$$

It is clear that each flowgraph has a hierarchical decomposition into primes. Our task is to determine the constraints on the functions f and g which ensure that, given the values $\mu(P)$ for all primes P , μ is defined unambiguously by eqns. 10 and 11. The result is given in the following theorem.

Theorem

Given the values of μ for the prime flowgraphs, eqns. 10 and 11 define μ unambiguously for all flowgraphs if, and only if, the functions f and g satisfy eqns. 12–16 for all flowgraphs F_1, F_2 and F_3 :

$$g(\mu(F_1), \mu(P_1)) = \mu(F_1) \quad (12)$$

$$g(\mu(P_1), \mu(F_1)) = \mu(F_1) \quad (13)$$

$$g(\mu(F_1), g(\mu(F_2), \mu(F_3))) = g(g(\mu(F_1), \mu(F_2)), \mu(F_3)) \quad (14)$$

$$g(g(\mu(F_1), \mu(F_2)), \mu(F_3)) = g(g(\mu(F_1), \mu(F_3)), \mu(F_2)) \quad (15)$$

(if F_1 has at least two procedure nodes)

$$f(\mu(F_1), \mu(F_2)) = g(g(f(\mu(P_1), \mu(P_1)), \mu(F_1)), \mu(F_2)) \quad (16)$$

Proof

We first show that the conditions are necessary. Eqn. 12 is necessary as $F_1(P_1) = F_1$. Eqn. 13 is necessary as $P_1(F_1) = F_1$. Eqn. 14 is necessary as nestings is associative. Eqn. 15 is necessary as the order of two nestings on the same flowgraph is irrelevant. Eqn. 16 is necessary as $F_1; F_2$ can be decomposed as $(P_1; P_1)(F_1)(F_2)$.

Now we show that the conditions are also sufficient conditions for the non-occurrence of ambiguities. There are two reasons why ambiguities may occur. The first is that there are decompositions of primes. The decompositions for a prime P are $(P, [P, P_1])$ and $(P, [P_1, P])$, which both belong to $Nest$. It follows from eqns. 12 and 13, respectively, that these decompositions do not lead to ambiguities. The second reason for possible ambiguities is that flowgraphs may have more than one decomposition. Suppose the flowgraph F has two distinct decompositions, $(F, [F_1, F_2])$ and $(F, [F_3, F_4])$. Below we distinguish ten different cases. We easily verify that each situation is covered by one of the cases. Each case leads to a sufficient condition for the non-occurrence of ambiguities. All ten conditions are shown to hold if eqns. 12–16 hold. This last part of the proof is given in the Appendix.

Case 1

$$F = F_1; F_2 \text{ and } F = F_3; F_4.$$

Without a loss of generality, we may assume that the start node of F_4 belongs to F_2 . There then exists a flowgraph F_5 such that $F_3 = F_1; F_5$ and $F_2 = F_5; F_4$. No ambiguities can arise in this case if

$$f(\mu(F_1), f(\mu(F_3), \mu(F_4))) = f(f(\mu(F_1), \mu(F_5)), \mu(F_4)) \quad (17)$$

Case 2

$$F = F_1; F_2, F = F_3(F_4) \text{ and } F_4 \subseteq F_1.$$

There exists a flowgraph F_5 such that $F_1 = F_5(F_4)$ and $F_3 = F_5; F_2$. No ambiguities can arise in this case if

$$f(g(\mu(F_3), \mu(F_4)), \mu(F_2)) = g(f(\mu(F_5), \mu(F_2)), \mu(F_4)) \quad (18)$$

Case 3

$$F = F_1; F_2, F = F_3(F_4) \text{ and } F_4 \subseteq F_2.$$

There exists a flowgraph F_5 such that $F_2 = F_5(F_4)$ and $F_3 = F_1; F_5$. No ambiguities can arise in this case if

$$f(\mu(F_1), g(\mu(F_3), \mu(F_4))) = g(f(\mu(F_1), \mu(F_5)), \mu(F_4)) \quad (19)$$

Case 4

$F = F_1; F_2, F = F_3(F_4)$ and neither $F_4 \subseteq F_1$ nor $F_4 \subseteq F_2$, the start node of F_4 is the start node of F_1 , and the stop node of F_4 is the stop node of F_2 .

In this case, we have $\widehat{F}_4 = \widehat{F}_1; F_2$ and $F_3 = P_1$. No ambiguities can arise in this case if

$$f(\mu(F_1), \mu(F_2)) = g(\mu(P_1), f(\mu(F_1), \mu(F_2))) \quad (20)$$

Case 5

$F = F_1; F_2, F = F_3(F_4)$ and neither $F_4 \subseteq F_1$ nor $F_4 \subseteq F_2$, the start node of F_4 is the start node of F_1 , and the stop node of F_4 is not the stop node of F_2 .

There exists flowgraphs F_5 and F_6 such that $F_2 = F_5; F_6, F_4 = F_1; F_5$ and $F_3 = P_1; F_6$. No ambiguities can arise in this case if

$$f(\mu(F_1), f(\mu(F_5), \mu(F_6))) = g(f(\mu(P_1)\mu(F_6)), f(\mu(F_1), \mu(F_5))) \quad (21)$$

Case 6

$F = F_1; F_2, F = F_3(F_4)$ and neither $F_4 \subseteq F_1$ nor $F_4 \subseteq F_2$, the start node of F_4 is not the start node of F_1 , and the stop node of F_4 is the stop node of F_2 .

There exist flowgraphs F_5 and F_6 such that $F_1 = F_6; F_5, F_4 = F_5; F_2$ and $F_3 = F_6; P_1$. No ambiguities can arise in this case if

$$f(f(\mu(F_6), \mu(F_5)), \mu(F_2)) = g(f(\mu(F_6), \mu(P_1)), f(\mu(F_5), \mu(F_2))) \quad (22)$$

Case 7

$F = F_1; F_2, F = F_3(F_4)$ and neither $F_4 \subseteq F_1$ nor $F_4 \subseteq F_2$, the start node of F_4 is not the start node of F_1 , and the stop node of F_4 is not the stop node of F_2 .

There exist flowgraphs F_5, F_6, F_7 and F_8 such that $F_1 = F_7; F_5, F_4 = F_5; F_6, F_2 = F_6; F_8$ and $F_3 = F_7; P_1; F_8$. No ambiguities can arise in this case if

$$f(f(\mu(F_7), \mu(F_5)), f(\mu(F_6), \mu(F_8))) = g(f(\mu(F_7), f(\mu(P_1), \mu(F_8))), f(\mu(F_5), \mu(F_6))) \quad (23)$$

Case 8

$$F = F_1(F_2), F = F_3(F_4) \text{ and } F_2 \subseteq F_4.$$

There exists a flowgraph F_5 such that $F_1 = F_3(F_5)$ and $F_4 = F_5(F_2)$. No ambiguities can arise in this case if eqn. 14 holds.

Case 9

$$F = F_1(F_2), F = F_3(F_4) \text{ and } F_2 \text{ and } F_4 \text{ edge-disjoint.}$$

There exists a flowgraph F_5 , which has at least two procedure nodes, such that $F_1 = F_5(F_4)$ and $F_3 = F_5(F_2)$. No ambiguities can arise in this case if eqn. 15 holds.

Case 10

$$F = F_1(F_2), F = F_3(F_4), \text{ neither } F_2 \subseteq F_4 \text{ nor}$$

$$F_4 \subseteq F_2, F_2 \text{ and } F_4 \text{ not edge-disjoint.}$$

Without a loss of generality, we may assume that the stop node of F_2 belongs to F_4 . There exist flowgraphs F_5, F_6, F_7, F_8 and F_9 such that $F_2 = F_5; F_6, F_4 = F_6; F_7, F_1 = F_8(F_7)$ and $F_3 = F_8(F_5)$ and $F_9 = F_9(P_2)$. No ambiguities

can arise in this case if

$$g(g(\mu(F_9), \mu(P_2)), \mu(F_7)), f(\mu(F_5), \mu(F_6))) = g(g(\mu(F_9), \mu(P_2)), \mu(F_5)), f(\mu(F_6), \mu(F_7))) \quad (24)$$

This proves the theorem. \square

The next theorem shows that the strong metric values of a sequence and a nesting of an arbitrary number of flowgraphs are determined by the metric values of the components in a uniform way, which is in contrast with the situation for recursive metrics. We use the function *foldl1* (which is well known from functional programming [5]), informally defined by

$$\text{foldl} (\oplus, [x_1, x_2, \dots, x_n]) = (\dots ((x_1 \oplus x_2) \oplus x_3) \dots) \oplus x_n \quad (25)$$

and formally by

$$\text{foldl1} (f, [x]) = x \quad (26)$$

$$\text{foldl1} (f, x : y : xs) = \text{foldl1} (f, f(x, y) : xs) \quad (27)$$

Theorem

Let μ be a strong metric, and f and g its functions associated with *Seq* and *Nest*, respectively, then, for each $n \geq 1$

$$\mu(F_1; F_2; \dots; F_n) = \text{foldl1} (f, [\mu(F_1), \mu(F_2), \dots, \mu(F_n)]) \quad (28)$$

$$\mu(F_1(F_2, \dots, F_n)) = \text{foldl1} (g, [\mu(F_1), \mu(F_2), \mu(F_3), \dots, \mu(F_n)]) \quad (29)$$

Proof

Straightforward, using induction with respect to n . \square

For example, if $\mu(F_1; F_2) = \mu(F_1) + \mu(F_2)$, then $\mu(F_1; F_2; \dots; F_n) = \text{sum} [\mu(F_1), \mu(F_2), \dots, \mu(F_n)]$ for all $n \geq 1$.

Theorem

Each strong metric is a recursive metric.

Proof

Let μ be a strong metric, and f and g its functions associated with *Seq* and *Nest*, respectively, and let f_n and g_n for all $n \geq 2$ be defined by

$$f_n(x_1, \dots, x_n) = \text{foldl1} (f, [x_1, \dots, x_n]) \quad (30)$$

$$g_n(x_1, \dots, x_n) = \text{foldl1} (g, [x_1, \dots, x_n]) \quad (31)$$

It is straightforward to show, using induction with respect to n , that eqns. 5 and 6 are satisfied. Thus, μ is a recursive metric. \square

With the choice of *Seq* and *Nest* as decomposition operations, the class of structure metrics is a proper subclass of the class of structure metrics in the FWK approach, and these metrics behave uniformly with respect to the simultaneous sequencing or nesting of a number of flowgraphs. In the following Section, we show a number of examples of strong metrics.

5 Examples

5.1 Additive metrics

A whole class of strong metrics is given by

- the values for the primes are arbitrarily chosen.

- $f(x, y) = x + y + a$, where a is chosen arbitrarily.
- $g(x, y) = x + y - \mu(P_1)$

Eqns. 12–16 are easily shown to hold. Thus, for each choice of the metric values of the primes and each choice of a , we have a strong metric. We call these metrics additive metrics.

5.2 McCabe's metric

M McCabe's metric [6] is an additive metric. The value for a is -1 and, for each prime P , the value $\mu(P)$ is one greater than the number of nodes of P with outdegree ≥ 2 .

5.3 Number of occurrences of a particular prime

Let Pr be a prime flowgraph, and let μ be the additive metric for which $a = 0$ and the metric values for the primes are given by $\mu(Pr) = 1$ and $\mu(P) = 0$ for all $P \neq Pr$. $\mu(F)$ then is the number of occurrences of the prime Pr in F , with respect to any decomposition of F using decompositions from *Seq* and *Nest*.

5.4 Highest valued prime

Consider the metric functions defined by $f(x, y) = g(x, y) = \max(x, y)$. We can easily check that these functions satisfy eqns. 12–16 if, and only if, $\mu(Pr) \geq \mu(P_1)$ for all primes Pr . The corresponding strong metric assigns to each flowgraph the maximum of the metric values of its prime components.

5.5 Maximal depth of nesting (not a strong metric)

Let μ be the maximal depth of nesting metric. $\mu(F_1(F_2))$ not only then depends on $\mu(F_1)$ and $\mu(F_2)$, but also on the depth where F_2 is nested on F_1 . Thus, there exists no function g that satisfies eqn. 11, and thus μ is not a strong metric.

6 Symmetric strong software metrics

In the previous Section (and the Appendix), it has been shown that for each strong metric μ the functions f and g , associated with the *Seq* and *Nest*, have the properties

- f is symmetric and associative.
- g is associative and has unit element $\mu(P_1)$.

The function g is not necessarily symmetric. However, in all examples of strong metrics we have seen up to now, this function is symmetric. The subclass of strong metrics with symmetric g seems to deserve special attention.

Definition

A mapping μ from flowgraphs to numbers is a *symmetric strong metric* if it is a strong metric whose function g , associated with *Nest*, satisfies

$$g(\mu(F_1), \mu(F_2)) = g(\mu(F_2), \mu(F_1)) \quad (32)$$

The following theorem gives a number of conditions, each of which is sufficient to ensure that a strong metric is a symmetric strong metric. This theorem seems to show that most of the strong metrics are symmetric strong.

Theorem

A mapping μ from flowgraphs to numbers is a symmetric strong metric if μ is a strong metric and one of the following conditions holds:

- the function that maps $\mu(F)$ to $\mu(F; P_1)$ has an inverse, i.e. there exists a function h such that

$$\mu(F) = h(f(\mu(F), \mu(P_1))) \quad (33)$$

- there exists a flowgraph F with more than one procedure node and

$$\mu(F) = \mu(P_1) \quad (34)$$

- eqn. 15 holds for all flowgraphs F_1 .

Proof

We have to show that, for each strong metric, each of the three conditions above implies the symmetry of g . The symbol $=_n$ is used to denote an equality that is a consequence of eqn. (n).

$$\begin{aligned} \square \text{ As } g(\mu(F_1), \mu(F_2)) & \\ &=_{33} h(f(g(\mu(F_1), \mu(F_2)), \mu(P_1))) \\ &=_{18} h(g(f(\mu(F_1), \mu(P_1)), \mu(F_2))) \\ &=_{19} h(f(\mu(F_1), g(\mu(P_1), \mu(F_2)))) \\ &=_{13} h(f(\mu(F_1), \mu(F_2))), \end{aligned}$$

the symmetry of g is implied by the symmetry of f .

$$\begin{aligned} \square g(\mu(F_1), \mu(F_2)) & \\ &=_{13} g(g(\mu(P_1), \mu(F_1)), \mu(F_2)) \\ &=_{34} g(g(\mu(F), \mu(F_1)), \mu(F_2)) \\ &=_{15} g(g(\mu(F), \mu(F_2)), \mu(F_1)) \\ &=_{34} g(g(\mu(P_1), \mu(F_2)), \mu(F_1)) \\ &=_{13} g(\mu(F_2), \mu(F_1)). \end{aligned}$$

- If eqn. 15 holds for all flowgraphs F_1 , then

$$\begin{aligned} g(\mu(F_1), \mu(F_2)) & \\ &=_{13} g(g(\mu(P_1), \mu(F_1)), \mu(F_2)) \\ &=_{15} g(g(\mu(P_1), \mu(F_2)), \mu(F_1)) \\ &=_{13} g(\mu(F_2), \mu(F_1)) \quad \square \end{aligned}$$

However, not all strong metrics are symmetric strong metrics. An example of a strong metric that is not a symmetric strong metric is given below.

Example

The metric values are $\{1, 2, 3, 4\}$. Define μ on primes such that $\mu(P) = 1$ if, and only if, $P = P_1$. Define f by $f(x, y) = 2$. Define g by $g(1, x) = g(x, 1) = x$ ($x \in \{1, 2, 3, 4\}$); $g(2, x) = g(x, 2) = 2$ ($x \in \{2, 3, 4\}$); $g(x, y) = x$ ($x, y \in \{3, 4\}$). It is easy to check that eqns. 12–16 hold. Note that, for a flowgraph F with two or more procedure nodes,

we have $\mu(F) \neq 1$, which means that eqn. 15 need not hold for $\mu(F_1)$ equal to 1. Thus, μ is a strong metric. It is not a symmetric strong metric, however, as g is not symmetric (provided primes P and P' exist with $\mu(P) = 3$ and $\mu(P') = 4$).

7 Semi-strong software metrics

In Section 4, we showed that strong metrics, in contrast to recursive metrics, have a uniformity property with respect to the sequencing and nesting of an arbitrary number of flowgraphs. This is because the decomposition operations were chosen to be the operations sequencing and nesting, instead of simultaneous sequencing of nonsequential flowgraphs and simultaneous nesting of maximal proper subflowgraphs on primes. However, the resulting class of strong metrics does not contain the depth of nesting metric, which indicates that this class might be considered to be too restricted.

In this Section, we define an intermediate class of structure metrics, called semi-strong metrics, with a uniformity property and which contains the depth of nesting metric. This is done by considering not all nestings, but only those of proper maximal subflowgraphs on nonsequential flowgraphs. Thus, the set of decomposition operations is given by

$$X_{\text{semi-strong}} = \{Seq, Nest'\} \quad (35)$$

$$Nest' = \{(F_1(F_2), [F_1, F_2]) \mid F_1 \text{ is nonsequential};$$

$$F_2 \text{ is a proper maximal subflowgraph of } F_1\} \quad (36)$$

A mapping μ from flowgraphs to numbers is a *semi-strong metric* if, and only if, there exist two functions f and g such that

$$\mu(F_1; F_2) = f(\mu(F_1), \mu(F_2)) \quad (37)$$

$$\mu(F_1(F_2)) = g(\mu(F_1), \mu(F_2))$$

if F_1 is nonsequential and F_2 is a

$$\text{proper maximal subflowgraph of } F_1 \quad (38)$$

As for strong metrics, each flowgraph has a hierarchical decomposition into primes, and our task is to determine the constraints on the functions f and g which ensure that, given the values $\mu(P)$ for all primes P , μ is defined unambiguously by eqns. 37 and 38. The result is given by the following theorem.

Theorem

Given the values of μ for the prime flowgraphs, eqns. 37 and 38 define μ unambiguously for all flowgraphs if, and only if, the functions f and g satisfy eqn. 17 and the following relaxation of eqn. 15:

$$g(g(\mu(F_1), \mu(F_2)), \mu(F_3)) = g(g(\mu(F_1), \mu(F_3)), \mu(F_2))$$

(if F_1 is nonsequential and P_1 occurs at least twice as

$$\text{maximal one-entry subflowgraph of } F_1) \quad (39)$$

Proof

The conditions are necessary: eqn. 17 is necessary as sequencing is associative, and eqn. 39 is necessary as the order of two nestings on the same flowgraph is irrelevant.

To show that the conditions are also sufficient, it is sufficient to note that, unlike the case of strong metrics, there are no decompositions of primes in this case, and from the ten cases considered in the proof for the case of strong metrics, only cases 1 and 9 occur for semi-strong metrics. \square

Just as strong metrics, semi-strong metrics have a uniformity property for simultaneous sequencing and nesting.

Theorem

Let μ be a semi-strong metric, and f and g its functions associated with *Seq* and *Nest'*, respectively. Then, for each $n \geq 1$

$$\mu(F_1; F_2; \dots; F_n) = foldl1(f, [\mu(F_1), \mu(F_2), \dots, \mu(F_n)]) \quad (40)$$

$$\mu(F_1(F_2, \dots, F_n)) = foldl1(g, [\mu(F_1), \mu(F_2), \dots, \mu(F_n)])$$

if F_1 is nonsequential and F_2, \dots, F_n are

$$\text{proper maximal subflowgraph of } F_1. \quad (41)$$

Proof

Straightforward, using induction with respect to n . \square

As its name already suggests, the class of semi-strong metrics contains the class of strong metrics and is contained in the class of recursive metrics.

Theorem

Each semi-strong metric is a recursive metric.

Proof

Similar to the corresponding theorem for strong metrics. \square

Theorem

Each strong metric is a semi-strong metric.

Proof

It is sufficient to show that eqns. 12–16 imply eqns. 17 and 39. Eqn. 39 is a relaxation of eqn. 15. A proof of eqn. 17 from eqns. 12–16 has already been given as part of the proof of the main theorem of Section 4. \square

Finally we show that the depth of nesting metric is a semi-strong metric. The depth of nesting metric is defined by

- $\mu(P) = 1$ for all primes P
- $f(x, y) = \max(x, y)$
- $g(x, y) = \max(x, y + 1)$
- eqns. 37 and 38

which is a correct definition of a semi-strong metric as eqns. 17 and 39 are satisfied.

8 References

- [1] FENTON, N.E.: 'Software metrics: a rigorous approach' (Chapman & Hall, 1991)
- [2] FENTON, N.E., and KAPOSI, A.A.: 'Metrics and software structure', *Inform. Softw. Technol.*, 1987, 6, pp. 301–320
- [3] FENTON, N.E., and WHITTY, R.: 'Axiomatic approach to software metrication through program decomposition', *Comput. J.*, 1986, 29, pp. 330–339

- [4] PRATHER, R.E.: 'On hierarchical software metrics', *Softw. Eng. J.*, 1987, 2, pp. 42-45
- [5] BIRD, R., and WADLER, P.: 'Introduction to functional programming' (Prentice Hall, 1988)
- [6] McCABE, T.J.: 'A complexity measure', *IEEE Trans.*, 1976, SE-2, pp. 308-320

9 Appendix

In this Appendix, the proof of the main theorem of Section 4 is completed. It has to be shown that eqns. 17-24 follow from eqns. 12-16.

Proof of eqn. 17:

$$\begin{aligned}
& f(\mu(F_1), f(\mu(F_2), \mu(F_3))) \\
&= {}_{16} g(g(f(\mu(P_1), \mu(P_1)), \mu(F_1)), f(\mu(F_2), \mu(F_3))) \\
&= {}_{16} g(g(f(\mu(P_1), \mu(P_1)), \mu(F_1)), g(g(f(\mu(P_1), \\
&\quad \mu(P_1)), \mu(F_2)), \mu(F_3))) \\
&= {}_{14} g(g(g(f(\mu(P_1), \mu(P_1)), \mu(F_1)), g(f(\mu(P_1), \\
&\quad \mu(P_1)), \mu(F_2))), \mu(F_3)) \\
&= {}_{15} g(g(g(f(\mu(P_1), \mu(P_1)), g(f(\mu(P_1), \mu(P_1)), \\
&\quad \mu(F_2))), \mu(F_1)), \mu(F_3)) \\
&= {}_{14} g(g(f(\mu(P_1), \mu(P_1)), g(g(f(\mu(P_1), \mu(P_1), \\
&\quad \mu(F_2)), \mu(F_1))), \mu(F_3)) \\
&= {}_{15} g(g(f(\mu(P_1), \mu(P_1)), g(g(f(\mu(P_1), \mu(P_1), \\
&\quad \mu(F_1)), \mu(F_2))), \mu(F_3)) \\
&= {}_{16} g(g(f(\mu(P_1), \mu(P_1)), f(\mu(F_1), \mu(F_2))), \mu(F_3)) \\
&= {}_{16} f(f(\mu(F_1), \mu(F_2)), \mu(F_3)).
\end{aligned}$$

Proof of eqn. 18:

$$\begin{aligned}
& f(g(\mu(F_1), \mu(F_2)), \mu(F_3)) \\
&= {}_{16} g(g(f(\mu(P_1), \mu(P_1)), g(\mu(F_1), \mu(F_2))), \mu(F_3)) \\
&= {}_{14} g(g(g(f(\mu(P_1), \mu(P_1)), \mu(F_1)), \mu(F_2)), \mu(F_3)) \\
&= {}_{15} g(g(g(f(\mu(P_1), \mu(P_1)), \mu(F_1)), \mu(F_3)), \mu(F_2)) \\
&= {}_{16} g(f(\mu(F_1), \mu(F_3)), \mu(F_2))
\end{aligned}$$

Proof of eqn. 19:

Observe that from eqns. 15 and 16 it follows that

$$f(\mu(F_1), \mu(F_2)) = f(\mu(F_2), \mu(F_1)) \quad (42)$$

Eqn. 19 is then proved by

$$\begin{aligned}
& f(\mu(F_1), g(\mu(F_2), \mu(F_3))) \\
&= {}_{42} f(g(\mu(F_2), \mu(F_3)), \mu(F_1)) \\
&= {}_{18} g(f(\mu(F_2), \mu(F_1)), \mu(F_3)) \\
&= {}_{42} g(f(\mu(F_1), \mu(F_2)), \mu(F_3))
\end{aligned}$$

Eqn. 20 follows immediately from eqn. 13.

Proof of eqn. 21:

$$\begin{aligned}
& f(\mu(F_1), f(\mu(F_2), \mu(F_3))) \\
&= {}_{17} f(f(\mu(F_1), \mu(F_2)), \mu(F_3)) \\
&= {}_{13} f(g(\mu(P_1), f(\mu(F_1), \mu(F_2))), \mu(F_3)) \\
&= {}_{18} g(f(\mu(P_1), \mu(F_3)), f(\mu(F_1), \mu(F_2)))
\end{aligned}$$

Proof of eqn. 22:

$$\begin{aligned}
& f(f(\mu(F_1), \mu(F_2)), \mu(F_3)) \\
&= {}_{17} f(\mu(F_1), f(\mu(F_2), \mu(F_3))) \\
&= {}_{13} f(\mu(F_1), g(\mu(P_1), f(\mu(F_2), \mu(F_3)))) \\
&= {}_{19} g(f(\mu(F_1), \mu(P_1)), f(\mu(F_2), \mu(F_3)))
\end{aligned}$$

Proof of eqn. 23:

$$\begin{aligned}
& f(f(\mu(F_1), \mu(F_2)), f(\mu(F_3), \mu(F_4))) \\
&= {}_{17} f(\mu(F_1), f(\mu(F_2), f(\mu(F_3), \mu(F_4)))) \\
&= {}_{17} f(\mu(F_1), f(f(\mu(F_2), \mu(F_3)), \mu(F_4))) \\
&= {}_{13} f(\mu(F_1), f(g(\mu(P_1), f(\mu(F_2), \mu(F_3))), \mu(F_4))) \\
&= {}_{18} f(\mu(F_1), g(f(\mu(P_1), \mu(F_4)), f(\mu(F_2), \mu(F_3)))) \\
&= {}_{19} g(f(\mu(F_1), f(\mu(P_1), \mu(F_4))), f(\mu(F_2), \mu(F_3)))
\end{aligned}$$

Proof of eqn. 24:

$$\begin{aligned}
& g(g(g(\mu(F_1), \mu(P_2)), \mu(F_2)), f(\mu(F_3), \mu(F_4))) \\
&= {}_{14} g(g(\mu(F_1), g(\mu(P_2), \mu(F_2))), f(\mu(F_3), \mu(F_4))) \\
&= {}_{14} g(\mu(F_1), g(g(\mu(P_1), \mu(F_2)), f(\mu(F_3), \mu(F_4)))) \\
&= {}_{16} g(\mu(F_1), f(\mu(F_2), f(\mu(F_3), \mu(F_4)))) \\
&= {}_{42} g(\mu(F_1), f(f(\mu(F_3), \mu(F_4)), \mu(F_2))) \\
&= {}_{17} g(\mu(F_1), f(\mu(F_3), f(\mu(F_4), \mu(F_2)))) \\
&= {}_{16} g(\mu(F_1), g(g(\mu(P_2), \mu(F_3)), f(\mu(F_4), \mu(F_2)))) \\
&= {}_{14} g(g(\mu(F_1), g(\mu(P_2), \mu(F_3))), f(\mu(F_4), \mu(F_2))) \\
&= {}_{14} g(g(g(\mu(F_1), \mu(P_2)), \mu(F_3)), f(\mu(F_4), \mu(F_2))).
\end{aligned}$$

© IEE: 1995

The paper was received 4 July 1994.

The authors are with Department of Computer Science, Twente University, PO Box 217, 7500 AE Enschede, The Netherlands.