RANKING THE COMPLEXITY OF NIAM CONCEPTUAL SCHEMAS BY ALPHA METRIC

Peter Kokol, Vili Podgorelec
University of Maribor, FERI, Laboratory for System Design
Smetanova 17, 2000 Maribor, Slovenia
Kokol@uni-mb.si

Henri Habrias, Nassim Hadj Rabia
Université de Nantes, Département Informatique - I.U.T. de Nantes
3 rue Mal Joffre, 44 041 Nantes cedex 01, France

Abstract: The general aim of formal methods to program development is to provide correctness of the problem specification and then to transform that specification into a working program. Often a formal specification is expressed using the NIAM conceptual schema. If we consider only valid types of constraints and if we keep all the consistent schemas, we have 28 possible schemas and 7 sub – schemas for each schema. The schemas and sub - schemas have very different complexities and there exists an intuitive order of these complexities which seems to be problematical. The aim of this paper is to verify this order using a complexity measure called α (information content/entropy metrics) and to define a new ranking functions generating more correct order.

1. Introduction: Formal Specification

The general aim of formal methods to program development is to provide correctness of the problem specification and then to transform that specification into a working program. The transformation process is continuously monitored by mathematical verification of each step. The verification is closely bound to the refinement process. In such manner we should eliminate the ambiguities of the natural languages and at the same time provide exactness to both the specification and the transformation process. Often the formal specification is expressed using state invariant. A NIAM conceptual schema is a state invariant with sets constraints between relations, domains and ranges of relations. If we consider only valid types of constraints and if we keep all the consistent schemas, we have 28 possible schemas and 7 sub – schemas for each schema. The schemas and sub - schemas have very different complexities and there exists an intuitive order of these complexities, which seems to be problematical. While the order is very important for efficient, effective and more user friendly design we decide to verify this order and if possible to generate a better and more correct one.

1.1. Scope and aim of the paper

In the paper we first present the B formal specification method and NIAM schemas, then the α – complexity metric is briefly introduced. Next the schemas written in B specification language are analysed using α – metric and the intuitive complexity order is compared with the complexities calculated. According to the comparison a new ranking function is presented and a new order generated. In the last part of the paper the results are discussed and further research is indicated.

The two main contributions of the paper are:

- The introduction of a the α – complexity measure into the formal specification field;
- A new order of NIAM conceptual schemas based on a ranking function derived from the complexities calculated with α – metric.

2. The B - Method

B [Abrial 1996] is a formal approach to software specification and development based on the set theory and the predicate logic of first order. It has been created by Jean-Raymond Abrial who is also the creator of the Z specification language [Spivey 1994]. B has been successfully applied in industry. For instance, a part of the

critical security system of the new metro line Meteor in Paris has been specified, developed with B and with Atelier B, a workbench including a prover (in B, the development is a proven development).
The basic mechanism of the B approach is that of the abstract machine. The invariant (for the variables) and the properties (for the constants) are predicates written in set language and, if necessary in predicate language. The operations are expressed in the language of the generalized substitutions.

2.1. NIAM Conceptual Schema

A NIAM conceptual schema is a state invariant with sets constraints between relations, domains and ranges of relations (Figure 2). All the combinations of constraints are not consistent. To have a subset constraint and a disjunction constraint, we need to have the bottom relation empty. And it is considered as abnormal (outside initialization) to have an empty relation. If we consider only the allowed types of constraints and if we keep all the consistent schemas, we have only 28 schemas (26 of them being analyzed are shown in Figure 1).

3. α – Metrics: An overview

In order to asses α – metrics (see [2, 4] for more detailed explanation about α – metrics) for a given program we have first to analyse the long-range correlation in that program. To do that we have first to regard a program as a string of symbols (i.e. characters), transform each character into a six bit long binary representation according to a fixed **code table,** map the string into a Brownian walk using every 0 bit as a step down and every 1 bit as a step up, and then calculate the root of mean square fluctuation F about the average of the displacement. In a two-dimensional Brownian walk model the F is defined as:

$$F(l) \equiv \sqrt{\overline{[\Delta y(l,l_0)]^2} - \left[\overline{\Delta y(l,l_0)}\right]^2}$$

where

$$\Delta y(l,l_0) \equiv y(l_0 + l) - y(l_0)$$

The $F(l)$ can distinguish two possible types of behaviour:

1) if the string sequence is uncorrelated (normal random walk) or there are local correlations extending up to a characteristic range i.e Markov chains or symbolic sequences generated by regular grammars, then

$$F(l) \approx l^{0.5}$$

2) if there is no characteristic length and the correlations are "infinite" then the scaling property of $F(l)$ is described by a power law

$$F(l) \approx l^{\alpha} \text{and } \alpha \neq 0.5.$$

The power law is most easily recognised if we plot $F(l)$ and l on a double logarithmic scale. If a power law describes the scaling property then the resulting curve is constant and the slope of the curve represents α. In the case that there are long range correlation in the strings analysed, α should not be equal to 0.5. α – metrics measures the complexity of the text. This complexity is related to the information content and the entropy of the text being analysed in the manner - greater complexity means greater information content and thereafter lesser entropy.

4. Analysis and Results

α – metric has been used to analyse the NIAM conceptual schemes. and sub – schemes. All sub - schemes have been ranked according to their assessed α – metric values. The sub – scheme's ranks for each conceptual scheme have been then averaged. According to these average ranks the conceptual schemes have been ordered. The ranks of this order are shown in parentheses in figure 1. Opposite to our expectation the intuitive order and the calculated order didn't match. Indeed the statistical analysis using Spearman correlation showed statistically insignificant (Spearman $R = -0.28$) reciprocal relation between α – metric complexity and intuitive complexity. Contrary, the internal ranking of sub – schemes within each conceptual scheme was in match with intuitive order both individually and in average. In fact the majority of internal and intuitive orders within conceptual schemes where statistically significantly correlated (on the $p = 0.05$ level). This last observation revealed the hypothesis that there is something wrong with the intuitive order of conceptual schemes. A closer examination of the figure 1 exposed the following three assumptions about the complexity of sub – schemas:

1. conceptual schemes with ***equality relation*** have the highest ranks and are thereafter the most complex;
2. conceptual schemes with exactly one ***totality relation*** have the lowest ranks and are as a consequence less complex, in fact it seems that exactly one totality relation per scheme reduces the complexity.
3. Total functions (relation in the middle) have highest middle ranks and thereafter "high – middle" complexities.

According to above we constructed a simple ***ranking function*** f_R

$$10 \text{ X equality relation} + 5 \text{ X middle relation} + 3 \text{ X other relations} +$$
$$+ 2 \text{ X two totality relations} - 5 \text{ X exactly one totality relation}$$

The order generated by above function is shown in Table 1. We can see that the ranks generated by the α – metric and the ranks generated by the ranking function f_R are in close match. Indeed, the Spearman's correlation coefficient is 0.88, meaning that the correlation between these two orders is statistically significant ($p = 0.000$) which proves the assumptions 1 to 3 introduced before.

5. Conclusion

The analysis of NIAM formal specification conceptual schemas using α – metric, with the aim to assess their complexity, has been presented in the paper. It has been shown that the order of schemas generated on the basis of intuitive assessment of schema's complexities didn't match with the order generated by using α – metric. In contrary the intuitive order of sub - schemas within each schema match significantly in majority of cases. Encouraged with this last finding we constructed a ranking function based on the three assumptions evident from the order provided by α – metric. This ranking function produced an order which match significantly ($p=0.000$) with the order generated by α – metric. In that manner we can make following conclusions:

1. The complexity of a schema depends largely on ***equality relation***;
2. exactly one ***totality relation*** per schema reduces its complexity;
3. total functions (relation in the middle of the scheme) increase the complexity of a schema approximately by the half of the complexity of a equality relation;
4. all other relations contributes by approximately one quarter of the equality relation.

These conclusions help us very much by gaining insight into the complexity of the formal specification. As a consequence we can use them as a guideline for designing less complex specifications which are then first easier to understand by end users (making them more valid) and second easier to implement.

6. References

1. Abrial, JR, Borger, E., Langmaack H. (1996) (Eds.) *Formal Methods for Industrial Applications*, Lecture Notes in Computer Science 1165.
2. Kokol, P., Brest, J., Žumer, V. (1997) Long-range correlations in computer programs. *Cybernetics and systems*, 28(1), 43-57.
3. Abrial, JR (1996) *The B – Book: Assigning Programs to meanings*, Cambridge University Press.
4. Kokol, P, Brest, J. (1998) Fractal structure of random programs, *SIGPLAN notices* 33(6), 33-38.
5. Habrias H. — Dictionnaire encyclopédique du génie logiciel, Préface de J-P. Finance, Masson, Paris, 1997 ISBN : 2-225-85328-2
6. Habrias H (1993). Introduction à la spécification, Préface de H. Gallaire, Paris, Masson.
7. Spivey J.M (1994) La notation Z, Traduction de M. Lemoine, Paris, Masson.

Table 1: NIAM conceptual schemas ranks defined intuitively, with α – metric and ranking function

SCHEMA	Rank defined by α – metric	Rank defined by ranking function
1	6	20
2	10	13
3	10	10
4	20	16
5	10	6
6	20	19
7	16	17
8	23	22
9	20	21
10	16	18
11	14	15
12	2	1
13	9	12
14	10	8
15	16	14
16	26	26
17	19	23
18	6	7
19	5	9
20	25	24
21	1	4
22	23	25
23	14	11
24	2	3
25	4	5
26	6	2

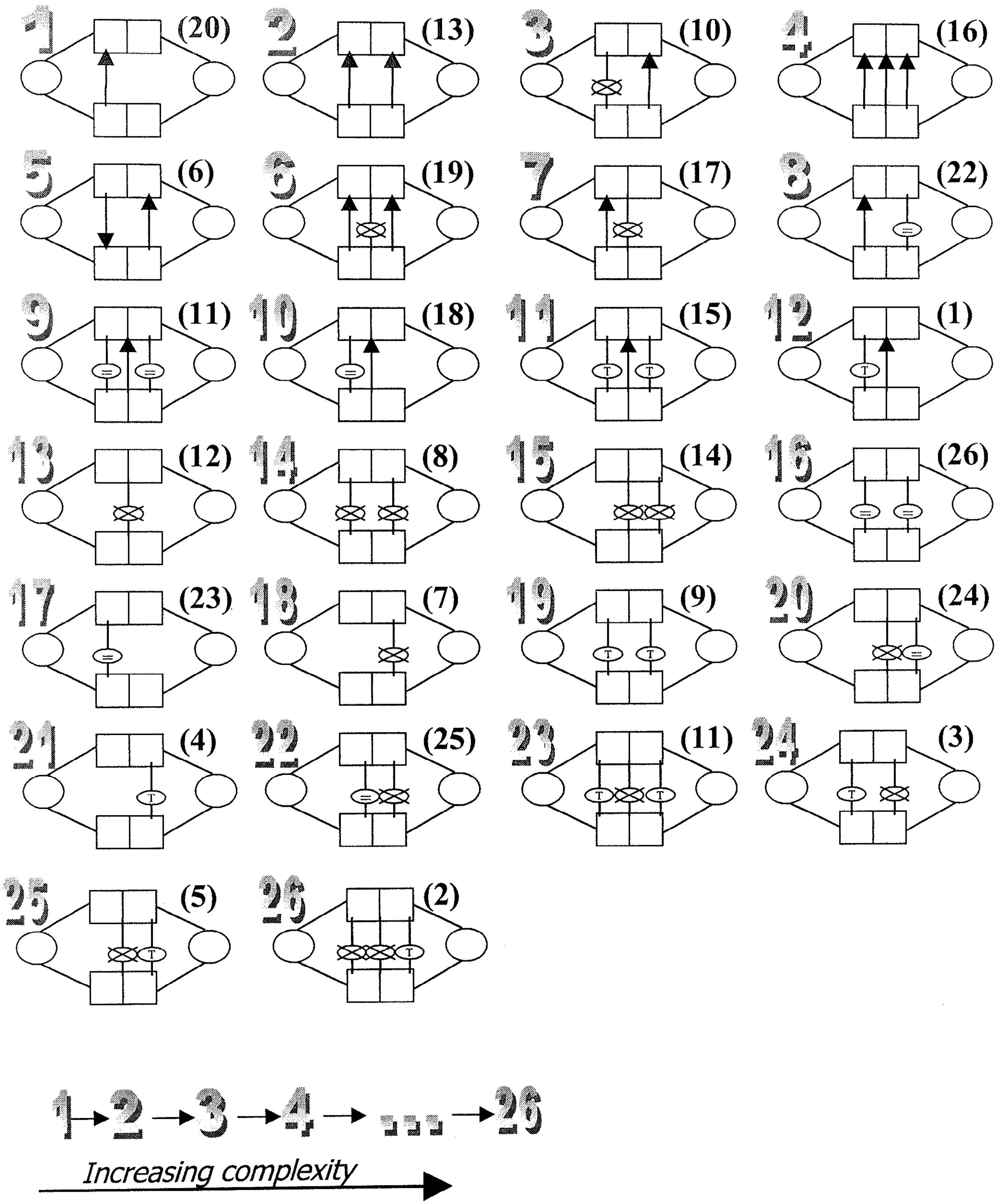

Figure 1: The NIAM conceptual schemes being analysed. The numbers in parentheses represent the actual ranking according to α – metric.

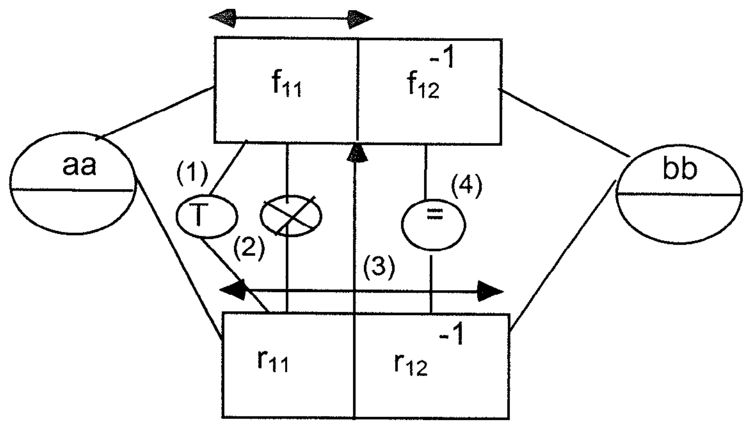

Figure 2. A NIAM conceptual scheme. (1) totality constraint, (2) disjunction constraint,(3) subset constraint, (4) equality constraint