

# Serdica Journal of Computing

Volume 1, Number 1, 2007

\*\*\*\*\*

## Contents

MANEVA, N. Comparative analysis: a feasible software engineering method (pp. 1-12)

LANDJEV, I., KR. HARALAMBIEV. On multiple deletion codes (pp. 13-26)

KOYCHEV, I. Experiments with two approaches for tracking drifting concepts (pp. 27-44)

MARKOV, M. On the vertex separation of cactus graphs (pp. 45-72)

SIMOV, K., P. OSENOVA. Applying a normalized compression metric to the measurement of dialect distance (pp. 73-86)

BUKOVICS, I. Apollo 13 risk assessment revisited (pp. 87-100)

RADEV, S. Mixed theories (pp. 101-114)

## Abstracts

\*\*\*\*\*

### COMPARATIVE ANALYSIS: A FEASIBLE SOFTWARE ENGINEERING METHOD

Nelly Maneva

e-mail: neman@gbg.bg

*ACM Computing Classification System (1998): D.2.9.*

*Key words:* multiple criteria, decision making, quality models, software metrics.

Abstract. The reasonable choice is a critical success factor for decision-making in the field of software engineering (SE). A case-driven comparative analysis has been introduced and a procedure for its systematic application has been suggested. The paper describes how the proposed method can be built in a general framework for SE activities. Some examples of experimental versions of the framework are briefly presented.

\*\*\*\*\*

### ON MULTIPLE DELETION CODES

Ivan Landjev, Kristijan Haralambiev

e-mail: ilandjev@nbu.bg

*ACM Computing Classification System (1998): E.4.*

*Key words:* insertion/deletion codes, Varshamov-Tennengolts codes, multiple insertion/deletion codes

Abstract. In 1965 Levenshtein introduced the deletion correcting codes and found an asymptotically optimal family of 1-deletion correcting codes. During the years there has been a little or no research on t-deletion correcting codes for larger values of t. In this paper, we consider the problem of finding the maximal cardinality  $L_2(n; t)$  of a binary t-deletion correcting code of length n. We construct an infinite family of binary t-deletion correcting codes. By computer search, we construct t-deletion codes for  $t = 2; 3; 4; 5$  with lengths  $n \leq 30$ . Some of these codes improve on earlier results by Hirschberg-Ferreira and Swart-Ferreira. Finally, we prove a recursive

upper bound on  $L_2(n; t)$  which is asymptotically worse than the best known bounds, but gives better estimates for small values of  $n$ .

\*\*\*\*\*

## **EXPERIMENTS WITH TWO APPROACHES FOR TRACKING DRIFTING CONCEPTS**

**Ivan Koychev**

e-mail: [ikoychev@math.bas.bg](mailto:ikoychev@math.bas.bg)

*ACM Computing Classification System (1998): I.2.6.*

*Key words:* Machine Learning; Concept Drift; Forgetting Models

Abstract. This paper addresses the task of learning classifiers from streams of labelled data. In this case we can face the problem that the underlying concepts can change over time. The paper studies two mechanisms developed for dealing with changing concepts. Both are based on the time window idea. The first one forgets gradually, by assigning to the examples weight that gradually decreases over time. The second one uses a statistical test to detect changes in concept and then optimizes the size of the time window, aiming to maximise the classification accuracy on the new examples. Both methods are general in nature and can be used with any learning algorithm. The objectives of the conducted experiments were to compare the mechanisms and explore whether they can be combined to achieve a synergetic effect. Results from experiments with three basic learning algorithms (kNN, ID3 and NBC) using four datasets are reported and discussed.

\*\*\*\*\*

## **ON THE VERTEX SEPARATION OF CACTUS GRAPHS**

**Minko Markov**

e-mail: [minko@cc.bas.bg](mailto:minko@cc.bas.bg)

*ACM Computing Classification System (1998): G.2.2.*

*Key words:* algorithmic graph theory, computational complexity, vertex separation, linear layout, layout extensibility, layout stretchability, cactus graph

Abstract. This paper is part of a work in progress whose goal is to construct a fast, practical algorithm for the vertex separation (VS) of cactus graphs. We prove a "main theorem for cacti", a necessary and sufficient condition for the VS of a cactus graph being  $k$ . Further, we investigate the ensuing ramifications that prevent the construction of an algorithm based on that theorem only.

\*\*\*\*\*

## **APPLYING A NORMALIZED COMPRESSION METRIC TO THE MEASUREMENT OF DIALECT DISTANCE**

**Kiril Simov, Petya Osenova**

e-mail: [kivs@bultreebank.org](mailto:kivs@bultreebank.org), [petya@bultreebank.org](mailto:petya@bultreebank.org)

*ACM Computing Classification System (1998): J.5.*

*Key words:* Kolmogorov Complexity, compression metric, dialect distance, language contacts

Abstract. The paper discusses the application of a similarity metric based on compression to the measurement of the distance among Bulgarian dialects. The similarity metric is defined on the basis of the notion of Kolmogorov complexity of a file (or binary string). The application of Kolmogorov complexity in practice is not possible because its calculation over a file is an undecidable problem. Thus, the actual similarity metric is based on a real life compressor which only approximates the Kolmogorov complexity. To use the metric for distance measurement of Bulgarian dialects we first represent the dialectological data in such a way that the metric is applicable. We propose two such representations which are compared to a baseline distance between dialects. Then we conclude the paper with an outline of our future work.

\*\*\*\*\*

## **APOLLO 13 RISK ASSESSMENT REVISITED**

**István Bukovics**

e-mail: istvan.bukovics@katved.hu

*ACM Computing Classification System (1998): F.4.1.*

*Key words:* Fault tree, risk assessment, prime event, conjunction and disjunction.

Abstract. Fault tree methodology is the most widespread risk assessment tool by which one is able to predict - in principle - the outcome of an event whenever it is reduced to simpler ones by the logic operations conjunction and disjunction according to the basics of Boolean algebra. The object of this work is to present an algorithm by which, using the corresponding computer code, one is able to predict - in practice - the outcome of an event whenever its fault tree is given in the usual form.

\*\*\*\*\*

## **MIXED THEORIES**

**Slavian Radev**

e-mail: sradev@go.com, sradev@wp.pl

*ACM Computing Classification System (1998): F.4.1.*

*Key words:* formal theories, expert systems, logical inference, argumentation systems

Abstract. In the present paper we investigate the life cycles of formalized theories that appear in decision making instruments and science. In few words mixed theories are build in the following steps: Initially a small collection of facts is the kernel of the theory. To express these facts we make a special formalized language. When the collection grows we add some inference rules and thus some axioms to compress the knowledge. The next step is to generalize these rules to all expressions in the formalized language. For these rules we introduce some conclusion procedure. In such a way we make small theories for restricted fields of the knowledge. The most important procedure is the mixing of these partial knowledge systems. In that step we glue the theories together and eliminate the contradictions. The last operation is the most complicated one and some simplifying procedures are proposed.