Fuzzy logic as Hájek's comparative notion of truth and applications in preference modeling

Peter Vojtáš

ABSTRACT. In this essay we are giving an overview of our work with our students and colleagues on fuzzy rules and similarities originally motivated by Petr Hájek's teaching and work. Our main starting points are Petr's visions of fuzzy logic in narrow sense and understanding of fuzzy values as a comparative notion of truth. Fuzzy logic in narrow sense is in our work reflected by formal models of fuzzy logic programming and similarities. Comparative notion of truth led us to understand fuzzy values as degree of user preference. As far as mathematical fuzzification of a domain can lead to several possible models, requirements of computer science application can help to prefer one of them. We overview our work (originally published with several coauthors) on models of fuzzy logic programming and its connection to generalized annotated programs and similarity reasoning; on fuzzy inductive logic programming; application to user preference learning and querying; applications to web information extraction and web semantization and conclude with some observations and lessons learned.

1 Introduction, motivation, problems.

Origins of fuzzy sets are connected to seminal work of L. A. Zadeh [?]. Original motivation was image processing, where different grades of shade of image pixel were first practical examples of fuzzy sets (see e.g. picture 1a [?]). Fuzzy set theory has developed rapidly with main applications in control (see e.g. pictures 1b and 1c [?], where rules for an inverted pendulum 1b are trained using fuzzy sets, e.g. 1c). In the case of fuzzy controllers, the system can be in a continuum of states which are a combination of values from an interval. That is why fuzzy sets form a partition and rules have to describe the control action for all possible combinations. This is a main difference in comparison with computer science applications.

Another motivation for fuzzy sets was modeling vague concepts like tall, young, etc. Many applications needed formal models. Here is the first con-

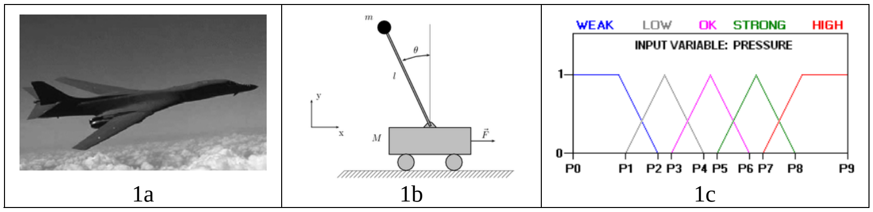


Figure 1. (a, b, c)

tribution of Petr Hájek. In [?] he further develops Zadeh's terms "fuzzy logic in broad or wide and narrow sense." In broad sense the term fuzzy logic has been used as synonymous with fuzzy set theory and its applications. In difference to Zadeh, which understood the emerging narrow sense fuzzy logic as a theory of approximate reasoning based on many valued logic, Hájek claims "... a logician will first study classical logical questions on completeness, decidability, complexity etc. of the symbolic calculi in question and then try to reduce question of Zadeh's agenda to questions of deduction as far as possible. This is the approach in monograph of Petr Hájek [?].

Petr Hájek introduced fuzzy logic in narrow sense as a mathematical theory of many valued logic. First contribution was done by J. Pavelka in [?] where he used graded statements (propositions), e.g.

$$p.0,5 \quad p \to q.0,7$$

to development of an axiomatic system of Lukasiewicz logic.

Basic development by Petr Hájek [?] was focused in axiomatization of 1-tautologies of various fuzzy logics. Just to motivate our view of this approach, let us notice that classical 0-1 tautology

$$(1) \qquad (\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\psi \to \chi))$$

is no more a tautology even in a 3 valued logic. This has led P. Hájek to the study of metamathematics of fuzzy logic, which is more detailed presented in some papers of this volume. This concerns especially introduction and study of so called Basic Logic BL, where (1) is not provable and there are some BL axioms replacing this form of composed implications, e.g. $(\varphi \to \psi) \to ((\psi \to \chi) \to (\varphi \to \chi))$.

For tuning real world applications to data, and moreover online with response in a second, it is hard to follow this way (to look for and work with an axiomatic systems).

Nevertheless our main motivation came from attending lectures of Petr Hájek on fuzzy logic [?], where he introduced fuzzy values as a comparative notion of truth (see also [?]). This was quite well in concordance to different representation of preferences in computer science applications. There are various ways of graphical representation of preference, importance, relevance etc. For user feedback we can use colors as in traffic lights (2a), different smiley's (2b), stars (2c) or sliders (2d, [?]).

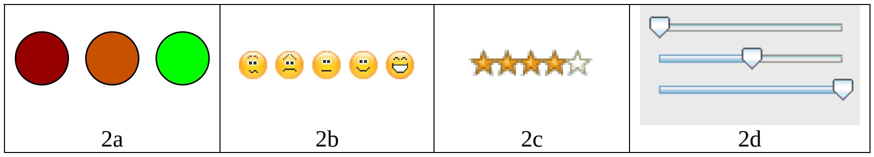
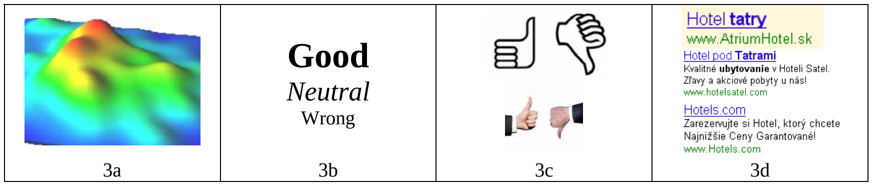


Figure 2. (a, b, c, d)

For representation of results we can use Google like graphics (3d) and mnemonics of top-down and let-right reading, enhanced by font size (3b). Another possibility thumbs up/down mnemonic (3c) and understanding colors in geographic maps, the higher terrain the better answer (3a, [?]).



Hotel **tatry**
www.AtriumHotel.sk

Hotel pod **Tatrami**
Kvalitné **ubytovanie** v Hoteli Satel.
Zľavy a akciové pobyty u nás!
www.hotelsatel.com

Hotels.com
Zarezervujte si Hotel, ktorý chcete
Najnižšie Ceny Garantované!
www.Hotels.com

Figure 3. (a, b, c, d)

To resume our starting point, these were computer science motivated problems with a small number of truth values (usually 7+-2), data intensive, with adaptation to different users (although in formal models we often use [0,1] interval).

2 Formal models

For some approaches we have developed formal models based on our real world motivation. It is our conviction, that formal models should describe a real world problem and its formal solution should contribute to a real world solution / improvement.

2.1 Fuzzy logic programming versus fuzzy resolution

In fuzzy logic programming we have first to decide whether rules are implications or clauses; whether my deduction procedure is resolution based refutation or database querying (in the crisp case these are equivalent, [**?**]). We have studied both approaches.

In [**?**] we have developed a model of fuzzy logic programming with implicative rules and database querying. Having

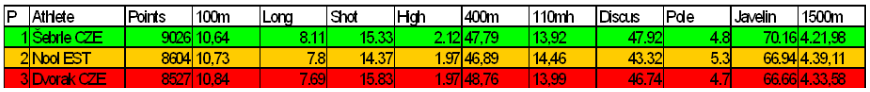
$$p.0,5 \quad p \to q.0,7$$

and truth function of → is an implicator $I : [0,1]^2 \to [0,1]$, then the result q can be deduced by many valued modus ponens with truth at least $C_I(0,5;\ 0,7)$, where C_I is a residual conjunctor to I. For such approach we have proved correctness of fuzzy logic programming (every computed answer is correct) and approximate completeness (every correct answer can be arbitrarily precisely computed). More a fixed point theory was developed too.

In a real world situation, both for user feedback and for query results presentation, we need finite valued fuzzy logic (more over a very low finite number of values). This was already used by the psychologist Rensis Likert "Often five ordered response levels are used, although many psychometricians advocate using seven or nine levels; a recent empirical study found that a 5- or 7- point scale may produce slightly higher mean scores relative to the highest possible attainable score, compared to those produced from a 10-point scale, and this difference was statistically significant (see [**?**])". This is also a common practice when refereeing papers, usually 7 values range from strong accept to strong reject.

Small finite valued fuzzy logic programming was studied in [**?**], where results of [**?**] were extended for conjunctors which are results of rounding t-norms to a finite scale. Note that rounding a conjunctor C upwards (in x axis to n values, in y axis to m values and in result z axis to k values) gives a conjunctor $C^k_{n,m}$ which need not be associative nor commutative (rounding upwards preserves these conjunctors to be left continuous). Our theory of fuzzy logic programming was extended also to this case.

In practical application it often comes to a situation that user preferences are aggregated from particular objectives. Situation is similar as in light athletic decathlon. Here individual achievements of an athlete are first converted to points (a fuzzy degree usually between 0 and 1000) and then summed up (fuzzy aggregation without normalization). See Fig. 4 with results of a race in Götzis on 27.5.2001 where R. Sebrle established first WR above 9000 points.



P	Athlete	Points	100m	Long	Shot	High	400m	110mh	Discus	Pole	Javelin	1500m
1	Šebrle CZE	9026	10,64	8.11	15.33	2.12	47,79	13,92	47.92	4.8	70.16	4.21,98
2	Nool EST	8604	10,73	7.8	14.37	1.97	46,89	14,46	43.32	5.3	66.94	4.39,11
3	Dvorak CZE	8527	10,84	7.69	15.83	1.97	48,76	13,99	46.74	4.7	66.66	4.33,58

Figure 4. Athletic decathlon – Götzis on 27.5.2001

Also in web shops and other user decision making similar phenomena appear. Namely, it can happen that even if users have similar objectives one user can have different weights for aggregation, or different users can have just opposite directions of preference.

For such situations, we have developed a model of fuzzy logic programming where fuzzy aggregations can appear in the body of rule. Such a rule can have form (in graded Prolog notation)

$$H \leftarrow @(B_1, ..., B_n).r$$

We have also showed in [**?**] that these are, in a sense, isomorphic to rules of GAP-Generalized annotated programs (with crisp ← and &, see [**?**])

$$H : @(b_1, ..., b_n) \leftarrow B_1 : b_1\&...\&, B_n : b_n$$

and procedural and declarative semantic are also in good connection (for more details see [**?**]).

Concerning fuzzy resolution, we consider a rule to be a clause so instead of $H \leftarrow B$ we write $H \vee \neg B$ and deduction is a refutation process initialized by $\neg H$. We have studied this problem in [**?**] in a more general setting, namely graded many valued resolution. This means, having

$$(A \vee B).x \quad and \quad (\neg A \vee C).y$$

We are looking for a function calculating

$$(B \vee C).f_{\vee}(x, y)$$

correctly, that means in models of fuzzy propositional logic. Some interesting combination occurred, especially $f_{\vee}(x, y) = 0 \quad for \quad x + y \leq 1$, for more see [**?**]. Nevertheless a practical question remains, who will create these clauses, an untrained user probably not.

2.2 Fuzzy similarity

Fuzzy similarity is another phenomenon which is important for practical computer science applications. Somebody looking for a resource (web page, document, product, genetic entity ...) has some requirements; nevertheless it can happen that there are no objects fulfilling these requirements. Then a (most) similar resource can make user happy. Similarity is in a sense dual to distance, triangular inequality which is necessary to make a distance metric is dual to transitivity for fuzzy similarity. For a similarity s on a domain A,

$$s : A \times A \to [0, 1]$$

the T-transitivity has form

$$T(s(x, y), s(y, z)) \leq s(x, z)$$

where T is a t-norm (or a conjunctor).

We say that the space (A, x) is a T-similarity space if s is fulfilling T-transitivity.

We say, that a triple

$$s(x, y), s(y, z), s(x, z)$$

is a *nontrivial* similarity triple, if all three numbers s(x, y), s(y, z) and s(x, z) are mutually different.

We have

Observation. Assume that similarity s is symmetric and space (A, s) is a min-similarity space. Then there are no nontrivial similarity triples in (A, s).

Observation is easy to conclude. By contradiction, assume, there is a nontrivial triple. Using symmetry (and renaming if necessary) this triple can be chosen such that $s(x, z) < s(x, y) < s(y, z)$. But then $s(x, z) < min(s(x, y), s(y, z))$, a contradiction with min transitivity.

Even notice, that min-similarity triples have to have two smaller numbers equal and only one bigger (see Fig. 5). This shows that min-similarity spaces have hierarchical structure and α-cuts form an equivalence (partition) on A. In real world situations this is very often not a case, data are more randomly distributed. A nontrivial similarity triple (Fig. 6) forces us to consider other t-norms in similarity transitivity than min. If such a triangle is colored in graph terminology we call such a triangle *colorful* if it contains all tree colors.

This motivates us to following

Problem. Assume we have K_n a coloring of the complete graph on n vertices by three colors. Is there a colorful triangle?

Of course, in general, without any assumption it is not true (see min-transitivity generated colorings). So we can ask, under what conditions on colorings there is a colorful triangle? Note that in a random coloring each triangle is colorful with probability $\frac{2}{9} > 0.2$. This observation is especially

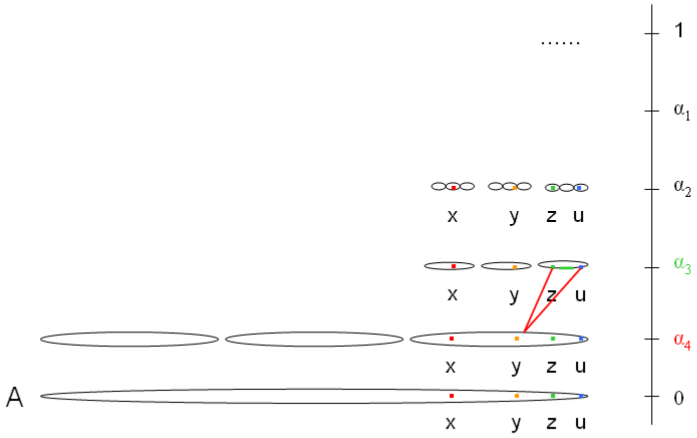


Figure 5.

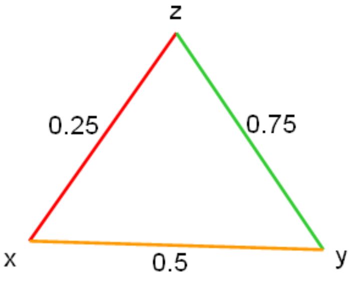


Figure 6.

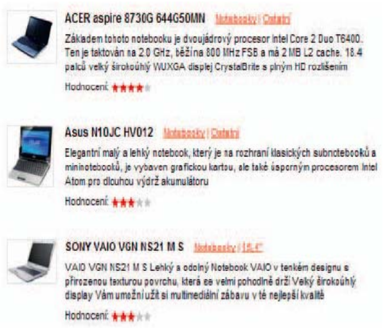
useful when considering similarity spaces which are non-metric and similarity distribution is highly random. For such spaces (e.g. multimedia, genetic databases ...) we have developed in [**?**] an indexing method based on T-similarity, where T is in a sense best similarity under which the space is still a T-similarity space. In [**?**] we have developed the formal theory of fuzzy similarity for a general class of t-norms.

3 Tools, Data, Experiments

In this essay style paper I have to confess, that main reason why I have started to study fuzzy induction and/or data mining, was a referee, which recommended to reject my paper with an argument that it is not clear where the rules (of my fuzzy logic programming contribution) are coming from.

Hence we have developed fuzzy Inductive Logic Programming (FILP) and various descendants.

3.1 Mining user preferences and top query answering



ACER aspire 8730G 644G50MN Notebooky | Ostatní

Základem tohoto notebooku je dvoujádrový procesor Intel Core 2 Duo T6400. Ten je taktován na 2.0 GHz, běží na 800 MHz FSB a má 2 MB L2 cache. 18.4 palců velký širokoúhlý WUXGA displej CrystalBrite s plným HD rozlišením

Hodnocení: ★★★★☆

Asus N10JC HV012 Notebooky | Ostatní

Elegantní malý a lehký notebook, který je na rozhraní klasických subnotebooků a mininotebooků, je vybaven grafickou kartou, ale také úsporným procesorem Intel Atom pro dlouhou výdrž akumulátoru

Hodnocení: ★★★☆☆

SONY VAIO VGN NS21 M S Notebooky | 15,4"

VAIO VGN NS21 M S Lehký a odolný Notebook VAIO v tenkém designu s přirozenou texturou povrchu, která se velmi pohodlně drží Velký širokoúhlý display Vám umožní užít si multimediální zábavu v té nejlepší kvalitě

Hodnocení: ★★★☆☆

Figure 7.

$$\begin{tikzcd}
S \subseteq \text{Objects} \arrow[r, "r_u"] \arrow[d, "d"'] & {[0,1]} \\
\prod_{i=1}^{n} D_{A_i} \arrow[r, "{f_i,\ i=1,\dots,n}"] & {[0,1]^n} \arrow[u, "@"']
\end{tikzcd}$$

Figure 8.

Main motivation for this was learning user preferences from user rating of a set sample set S of objects (see Fig. 7, stars correspond to mapping r_u), where a user interactively evaluates objects by number of stars, without making any comment on item properties. Basis for this is the object attribute representation of data (mapping d in Fig. 8), which assigns to each object its data values in the Cartesian product of domains of attributes $\prod_{i=1}^{n} D_{A_i}$. The learning task is to find user objectives on particular attributes (fuzzy sets on attribute domains f_i) and a fuzzy aggregation function combining these attribute preference degrees. In a formal model we require the whole diagram should commute, in practical setting we require it gives good advice for the user. What is a good advice can be measured in several ways, most appropriate for web search is to get best objects (with highest fuzzy degree) first. For this we use Kendal correlation coefficient

$$\tau = \frac{n_c - n_d}{\frac{1}{2}n(n-1)}$$

where n_c is the number of concordant pairs, and n_d is the number of discordant pairs in the data set (see [**?**]). For more on FILP and user preference mining see [**?**, **?**].

There is another feature in learning user objectives on particular attributes (fuzzy sets on attribute domains f_i). This gives an ordering of domains - for user U_1 North East direction is better (red rectangle can be objects good in degree at least 0.7) and for user U_2 the South West direction is better (blue rectangle can be objects good in degree at least 0.7) in

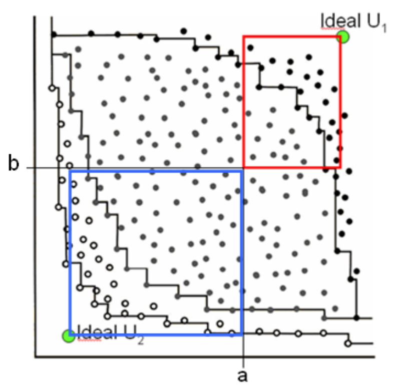


Figure 9.

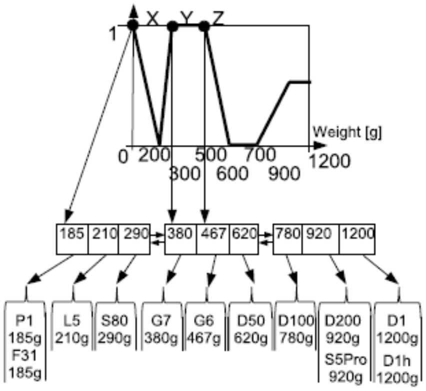


Figure 10.

ordered domain on Fig. 9. Fuzzy aggregation @ can be glued together from GAP rules like

Good_for_U1 : 0.7 IF A1 better_for_U1 than a AND
A2 better_for_U1 than b

Most of top-k algorithm use ordered approach to data by user preference. But considering different users, this ordering can change and it would be very costly reorder data each time a new user comes. In [**?**] we have developed an index structure, which given a fuzzy set can search data starting from best (see Fig. 10).

3.2 Web information extraction for web semantization

Web semantization is an idea understanding process of semantization of web resources by third party annotation (as opposed to semantic web idea where it is assumed that web resource creators will annotate their pages by ontology). Of course annotating web resources by third party is a difficult task. We have tried to make a progress to this task by dividing it to smaller subtasks. First is to consider only tabular product pages and dominantly textual pages. Second idea is to split the task to a domain independent annotation and to domain dependent annotation (here user feedback is necessary, as far we do not have ontology for every domain).

A strategy for web information extraction of textual pages in depicted in Fig. 11, [**?**]. For textual domain independent annotation we use a third party linguistic annotator. We use fuzzy ILP for user feedback learning of extracted items. For tabular product pages we have a different approach (Fig. 12). We parse the HTML structure by a DOM tree and looking for

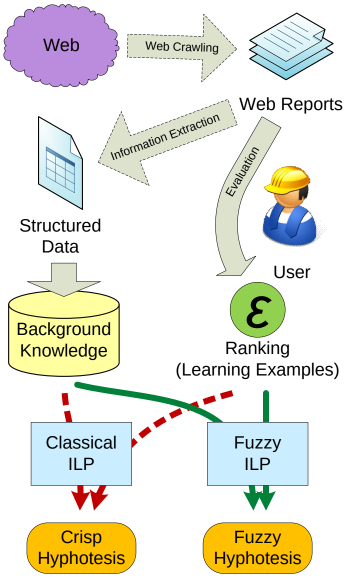


Figure 11.

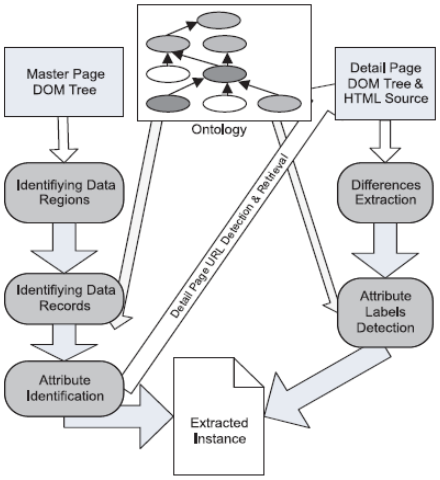

Master Page DOM Tree
Ontology
Detail Page DOM Tree & HTML Source
Identifiying Data Regions
Identifiying Data Records
Attribute Identification
Detail Page URL Detection & Retrieval
Differences Extraction
Attribute Labels Detection
Extracted Instance

Figure 12.

(fuzzy) similarities we can identify data regions and data records with a quite high accuracy. For attribute value identification we can make use of ontology and some regular expressions. If there is no ontology, we can use a heuristics which uses differences in detailed pages [?]. All this projects are part of uncertainty reasoning in the web, especially of fuzzy techniques.

4 Conclusions

In this paper a tribute to Petr Hájek is given, who brought us to fuzzy logic and modeling. We have tried to give an overview of our work in this area and to Petr's influence, especially to consider fuzzy values as a comparative notion of truth which is very well suited for user preference modeling. This work was mainly done with our (former) students J. Dedek, A. Eckhardt, P. Gursky, T. Horvath, S. Krajci, R. Lencses (left for industry), D. Maruscak (left for industry), R. Novotny, J. Pribolova (left for industry) and V. Vanekova (which can be hence considered as scientific (fuzzy) grand children of P. Hajek). We have totally ignored contributions of other authors to these fields of research, references to related work can be found in respective papers.

Concluding we can state that ideas of Petr Hájek - both on fuzzy logic in narrow sense and on fuzzy as a comparative notion of truth, have brought new insight to many problems, especially to user preference modeling and has led even to proof of concepts of these ideas based on experimental tools and experiments on real world data.

Lessons learned show:
- In the case of applications with signal (analogue, physical) data, typically a fuzzy controller, the state of the system ranges in a continuous range of all possible data. In this case fuzzy rules are built upon fuzzy partitions of domain of controller variables. In the case of applications with human created data (e.g. web data) - state of the system contains only data of existing resources (which usually do not fill a continuum of possibilities). In this case fuzzy rules are built upon (usually one) fuzzy subset of domain of variables which indicate user preference, objective.
- Fuzzy logic in narrow sense can be developed either in the direction of axiomatization of 1-tautologies, aiming for completeness and complexity issues. Fuzzy logic in narrow sense for data intensive applications usually does not need any logical axioms. There is a difference between systems based on implicative rules (modus ponens deduction and database querying) and disjunctive rules (resolution deduction and refutation).
- Learning of implicative rules is easier, based on equivalence of FLP - fuzzy logic programming and GAP - Generalized annotated programs and understanding learning FLP rules as learning of objectives and utility function in a user preference model as in operational research (see [**?**]).
- several application scenarios led to new requirements on indexing, top-k query answering, user preference learning and web information extraction and new measures of evaluating results / solutions.

We plan to work in this direction also in the future.

BIBLIOGRAPHY

Web pages of offspring

http://www.ksi.mff.cuni.cz/~dedek/
http://www.ksi.mff.cuni.cz/~eckhardt/
http://ics.upjs.sk/~gursky/
http://ics.upjs.sk/~horvath/
http://ics.upjs.sk/~krajci/
http://ics.upjs.sk/~novotnyr/
http://ics.upjs.sk/~vanekova/

Peter Vojtáš
Department of software engineering

Charles University
Malostranské nám. 25
118 00 Prague
Czech Republic
E-mail: vojtas@ksi.mff.cuni.cz