Presenting Proofs with Adapted Granularity*

Marvin Schiller[1] and Christoph Benzmüller[2]

[1] German Research Center for Artificial Intelligence (DFKI), Bremen, Germany
[2] International University in Germany, Bruchsal, Germany
Marvin.Schiller@dfki.de, c.benzmueller@googlemail.com

Abstract. When mathematicians present proofs they usually adapt their explanations to their didactic goals and to the (assumed) knowledge of their addressees. Modern automated theorem provers, in contrast, present proofs usually at a fixed level of detail (also called granularity). Often these presentations are neither intended nor suitable for human use. A challenge therefore is to develop user- and goal-adaptive proof presentation techniques that obey common mathematical practice. We present a flexible and adaptive approach to proof presentation based on classification. Expert knowledge for the classification task can be hand-authored or extracted from annotated proof examples via machine learning techniques. The obtained models are employed for the automated generation of further proofs at an adapted level of granularity.

Key words: Adaptive proof presentation, proof tutoring, automated reasoning, machine learning, granularity

1 Introduction

A key capability trained by students in mathematics and the formal sciences is the ability to conduct rigorous arguments and proofs and to present them. The presentation of proofs in this context is usually highly adaptive as didactic goals and the (assumed) knowledge of the addressee are taken into consideration. Modern theorem proving systems, in contrast, do often not sufficiently address this common mathematical practice. In particular automated theorem provers typically generate and present proofs only using very fine-grained and machine-oriented calculi. Of course, some theorem proving systems exists — amongst them prominent interactive theorem provers such as Isabelle/HOL[3], HOL[4], Coq[5], and Theorema[6] — that provide means for human-oriented proof presentations. Nevertheless the challenge of supporting user- and goal-adapted proof presentations has been widely neglected in the past. This constitutes an unfortunate gap, in particular since mathematics and the formal sciences are

* This work was supported by a grant from *Studienstiftung des Deutschen Volkes e.V.*
[3] http://www.cl.cam.ac.uk/research/hvg/Isabelle/
[4] http://hol.sourceforge.net/
[5] http://coq.inria.fr/
[6] http://www.risc.uni-linz.ac.at/research/theorema/

1 Let x be an element of $A \cap (B \cup C)$, 2 then $x \in A$ and $x \in B \cup C$. 3 This means that $x \in A$, and either $x \in B$ or $x \in C$. 4 Hence we either have (i) $x \in A$ and $x \in B$, or we have (ii) $x \in A$ and $x \in C$. 5 Therefore, either $x \in A \cap B$ or $x \in A \cap C$, so 6 $x \in (A \cap B) \cup (A \cap C)$. 7 This shows that $A \cap (B \cup C)$ is a subset of $(A \cap B) \cup (A \cap C)$. 8 Conversely, let y be an element of $(A \cap B) \cup (A \cap C)$. 9 Then, either (iii) $y \in A \cap B$, or (iv) $y \in A \cap C$. 10 It follows that $y \in A$, and either $y \in B$ or $y \in C$. 11 Therefore, $y \in A$ and $y \in B \cup C$ 12 so that $y \in A \cap (B \cup C)$. 13 Hence $(A \cap B) \cup (A \cap C)$ is a subset of $A \cap (B \cup C)$. 14 In view of Definition 1.1.1, we conclude that the sets $A \cap (B \cup C)$ and $(A \cap B) \cup (A \cap C)$ are equal.

Fig. 1. Proof of the statement $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$, reproduced from [1]

increasingly targeted as promising application areas for intelligent tutoring systems. We briefly illustrate the challenge with an example. In the elementary proof in basic set theory in Fig. 1 in Bartle & Sherbert's introductory textbook [1], intermediate proof steps are skipped when this seems appropriate: whereas most of the proof steps consist of the application of exactly one mathematical fact (e.g., a definition or a lemma, such as the distributivity of *and* over *or*), the step from assertion 9 to assertion 10 applies several inference steps at once, namely the application of the definition of $\cap$ twice, and then using the distributivity of *and* over *or*. Similar observations were made in the empirical studies within the DIALOG project [2], where tutors (mathematicians hired to help simulate the dialog system) identified limits for how many inference steps are to be allowed at once. An example from our DIALOG corpus [3] for a correct but unacceptably large student step that was rejected by the tutor is presented to the right.

student: $(x, y) \in (R \circ S)^{-1}$
tutor: `Now try to draw inferences from that!`
student: $(x, y) \in S^{-1} \circ R^{-1}$
tutor: `One cannot directly deduce that.`

The challenge thus consists in (i) developing means to model and assess different levels of proof granularity, (ii) providing support for the interactive or even automated acquisition of such models from well chosen proof examples, and (iii) combining these aspects with natural language (NL) generation techniques to present machine generated proofs at adaptive levels of granularity to humans.

Related work has addressed this challenge only to a moderate extent. The ΩMEGA system [4], for example, provides a hierarchically organized proof data structure that allows to represent proofs at different levels of granularity which are maintained simultaneously in the system. And ΩMEGA's proof explanation system P.rex [5] was able to generate adapted proof presentations by moving up or down these layers on request. The problem remains, however, of how to identify a particular level of granularity, how to model it, and how to ensure that this level of granularity is appropriate. A similar observation applies to the Edinburgh HiProofs system [6]. One particular level of proof granularity has been proposed by Autexier and Fiedler [7], which, in brief, refers to assertion level proofs where all assertion level inference steps are spelled out explicitly and refer only to facts readily available from the assertions or the previous inference

steps (*what-you-need-is-what-you-stated* granularity). However, they conclude that even the simple proof in Fig. 1 cannot be fully captured by their rigid notion of proof granularity.

In this paper we present a flexible and adaptive framework to proof presentation (currently used for formal proofs, not diagrammatic proofs etc.). Our approach employs rule sets (granularity classifiers) to model different levels of proof granularity. These rule sets are employed in a straightforward proof assessment algorithm to convert machine generated proofs (in our ΩMEGA system) into proofs at specific levels of detail. Both the granularity rules and the algorithm are outlined in Sect. 2. In Sect. 3 we show that our approach can successfully model the granularity of our running example proof in Fig. 1. Different models for granularity can either be hand-coded or they may be learned from samples using machine learning techniques. Ideally, the latter approach, which is described in Sect. 4, helps reducing the effort of adapting the system to new application and user contexts, and, in particular, to train the system by domain experts who are not familiar with expert systems.

2 An Adaptive Model for Granularity

We treat the granularity problem as a classification task: given a proof step, representing one or several assertion applications[7], we judge it as either *appropriate*, *too big* or *too small*. As our feature space we employ several mathematical and logical aspects of proof steps, but also aspects of cognitive nature. For example, we keep track of the background knowledge of the user in a basic (overlay) student model. We illustrate our approach with a proof step from Fig. 1: 10 is derived from 9 by applying the definition of $\cap$ twice, and then using the distributivity of *and* over *or*. In this step (which corresponds to multiple assertion level inference steps) we make the following observations:

(i) involved are two concepts: def. of $\cap$ and distributivity of *and* over *or*,
(ii) the total number of assertion applications is three,
(iii) all involved concepts have been previously applied in the proof,
(iv) all manipulations apply to a common part in 9,
(v) the names of the applied concepts are not explicitly mentioned, and
(vi) two of the assertion applications belong to *naive set theory* (def. of $\cap$) and one of them relates to the domain of propositional logic (distributivity).

These observations are represented as a feature vector, where, in our example, the feature "distinct concepts" receives a value of "2", and so forth. Currently, our system computes the following set of features for each (single- or multi-inference) proof step:

total: the total number of (assertion level) inference steps combined into one proof step,

[7] We use the notion of *assertion application* for inference steps that are justified by a mathematical fact (such as a definition, theorem or a lemma).

conceptsunique: the number of different concepts applied within the proof step,
mastered-concepts-unique (m.c.u.): the number of different employed mathematical facts assumed to be known to the user according to the very basic user model (which is updated in the course of the proof[8]),
unmastered-concepts-unique (unm.c.u.): the number of different employed mathematical facts assumed to be unknown to the user,
verb: whether the step is accompanied by a verbal explanation,
unverbalized-unknown: the number of assertions not accompanied by a verbal description and not known to the user,
lemmas: the number of employed assertions that are lemmas (in contrast to basic definitions),
hypintro: indicates whether a (multi-inference) proof step introduces a new hypothesis,
subgoals: indicates whether (and how many) new subgoals have been introduced,
same-subformula: indicates whether all manipulations apply to a common formula part,
newinst: indicates whether a variable has been instantiated,
close: indicates whether a branch of the proof has been finished,
parallelism: indicates when it is possible to apply the same assertion several times, but it is applied only on fewer occasions than possible,
forward: indicates inference applications in forward direction,
backward: indicates inference applications in backward direction,
direction-change: indicates whether the direction of inferences has changed w.r.t. the previous step,
step-analog: indicates whether the assertions applied within the current step have been applied before in the proof, in the same order and as a single step,
multi-step-analog: indicates whether the assertions applied within the current step have been applied before in the proof, in the same order, but not necessarily within a single step,
settheory, relations, topology, geometry, etc. the number of inference applications from each (mathematical) domain,
$\cap$-*Defn,* $\cup$-*Defn, eq-Defn, etc.* indicator feature for each concept.

These features were motivated by the corpora obtained from the experiments in the DIALOG project (cf. [2] and [3]) and discussions with domain experts. We express our models for classifying granularity as rule sets (cf. Fig. 4), which associate specific combinations of feature values to a corresponding granularity verdict ("appropriate", "too big" or "too small"). Our straightforward algorithm for granularity-adapted proof presentation takes two arguments, a granularity rule set and an assertion level proof[9] as generated by ΩMEGA. The assertion

[8] All concepts that were employed in an "appropriate" or "too small" proof step obtain the status of being known in the subsequent proof steps/proofs.

[9] In principle, our approach is not restricted to assertion level proofs and is also applicable to other proof calculi. However, in mathematics education we consider single assertion level proof steps as the finest granularity level of interest. We gained evidence for this choice from the empirical investigations in the DIALOG project (cf. [2] and [3]).

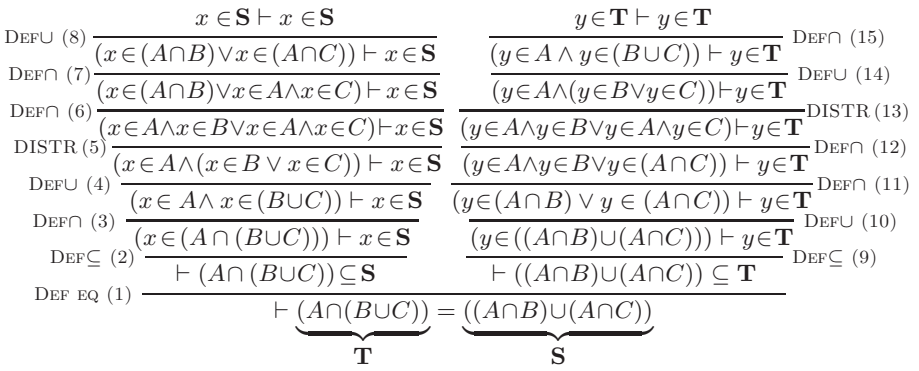

Fig. 2. Assertion level proof for the statement $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

level proof generated by ΩMEGA for our running example is given in Fig. 2; this proof is represented as a tree (or acyclic graph) in sequent-style notation and the proof steps are ordered. Currently we only consider plain assertion level proofs, and do not assume any prior hierarchical structure or choices between proof alternatives (as is possible in ΩMEGA). Our algorithm performs an incremental categorization of steps in the proof tree (where $n = 0, \ldots, k$ denotes the ordered proof steps in the tree; initially n is 1):

while there exists a proof step n **do**
evaluate the granularity of the compound proof step n (i.e., the proof step consisting of all assertion level inferences performed after the last step labeled "appropriate with explanation" or "appropriate without explanation" — or the beginning of the proof, if none exists yet) with the given rule set under the following two assumptions: (i) the involved concepts are mentioned in the presentation of the step (an *explanation*), and (ii) only the resulting formula is displayed.

1. **if** n is appropr. w. expl.
   then label n as "appropr. w. expl."; set $n := n+1$;
2. **if** n is too small w. expl., but appropr. wo. expl.
   then label n as "appropr. wo. expl."; set $n := n+1$;
3. **if** n is too small both w. and wo. explanation
   then label n as "too small"; set $n := n+1$;
4. **if** n is too big **then** label $n-1$ as "appropr. wo. expl." (i.e. consider the previous step as appropr.), unless $n-1$ is labeled "appropr. w. expl." or "appropr. wo. expl." already or n is the first step in the proof (in this special case label n as "appropr. w. expl." and set $n := n+1$).

We thereby obtain a proof tree with labeled steps (labeled nodes) which differentiates between those steps that are categorized as appropriate for presen-

tation and those which are considered too fine-grained. Proof presentations are generated by walking through the tree,[10] skipping the steps labeled *too small*.[11]

3 Modeling the Granularity of our Example Proof

We exemplarily model the granularity of the textbook proof in Fig. 1. Starting point is the initial assertion level proof from Fig. 2. This proof assumes the basic definitions and concepts in naive set theory (such as equality, subset, union, intersection and distributivity) and first-order logic. Notice that the Bartle & Sherbert proof in Fig. 1 starts in 1 with the assumption that an element x is in the set $A \cap (B \cup C)$. The intention is to show the subset relation $A \cap (B \cup C) \subseteq (A \cap B) \cup (A \cap C)$, which is not explicitly revealed until step 7, when this part of the proof is already finished. The same style of *delayed* justification for prior steps is employed towards the end of the proof, where statements 13 and 14 justify (or recapitulate) the preceding proof. For the comparison of proof step granularity in this paper, however, we consider a re-ordered variant of the steps in Fig. 1, which is displayed in Fig. 3 (a).[12] We now employ suitable granularity rule sets to automatically generate a proof presentation from our ΩMEGA assertion level proof which exactly matches the twelve steps of the Bartle & Sherbert proof, skipping intermediate proof steps according to our feature-based granularity model. Fig. 4 shows two sample rule sets which both lead to the automatically generated proof presentation in Fig. 3 (b). For instance, the three assertion level steps (11), (12) and (13) in the initial assertion level proof are combined into one single step from 9. to 10. in the proof presentation in Fig. 3 (b), like in the textbook proof. The rule set in Fig. 4 (a) was generated by hand, whereas the rule set in Fig. 4 (b) was automatically learned[13] (cf. Sect. 4). The rules are ordered by utility for conflict resolution. Note that rules 4–6 in Fig. 4 (a) express the relation between the appropriateness of steps and whether the employed concepts are mentioned verbally (feature *verb*), e.g. rule 6) enforces that the definition of equality is explicitly mentioned (as in step 1. in Fig 3 (b)). All other cases, which are not covered by the previous rules, are subject to a default rule. Natural language is produced here via simple patterns and more

[10] In case of several branches, a choice is possible which subtree to present first, a question which we do not address in this paper.

[11] Even though the intermediate steps which are *too small* are withheld, the presentation of the output step reflects the results of all intermittent assertion applications, since we include the names of all involved concepts whenever a (compound) step is appropriate with explanation.

[12] Note that step (1) in the re-ordered proof corresponds to the statements 7, 13 and 14 in the original proof which jointly apply the concept of set equality. The ordering of proof presentations can be dealt with using dialog planning techniques, as explored in [5].

[13] The sample proof was used to fit a rule set to it via C5.0 machine learning [8]. All steps in the sample proof were provided as training instances with label *appropriate*, all tacit intermediate assertion level steps were labeled as *too small*, and always the next bigger step to each step in the original proof was provided as a *too big* step.

1. In view of Definition 1.1.1, we [show] that the sets $A \cap (B \cup C)$ and $(A \cap B) \cup (A \cap C)$ are equal. 14 [First we show] that $A \cap (B \cup C)$ is a subset of $(A \cap B) \cup (A \cap C)$. 7 [Later we show] $(A \cap B) \cup (A \cap C)$ is a subset of $A \cap (B \cup C)$. 13
2. Let x be an element of $A \cap (B \cup C)$, 1
3. then $x \in A$ and $x \in B \cup C$. 2
4. This means that $x \in A$, and either $x \in B$ or $x \in C$. 3
5. Hence we either have (i) $x \in A$ and $x \in B$, or we have (ii) $x \in A$ and $x \in C$. 4
6. Therefore, either $x \in A \cap B$ or $x \in A \cap C$, 5
7. so $x \in (A \cap B) \cup (A \cap C)$. 6
8. Conversely, let y be an element of $(A \cap B) \cup (A \cap C)$. 8
9. Then, either (iii) $y \in A \cap B$, or (iv) $y \in A \cap C$. 9
10. It follows that $y \in A$, and either $y \in B$ or $y \in C$. 10
11. Therefore, $y \in A$ and $y \in B \cup C$, 11
12. so that $y \in A \cap (B \cup C)$. 12

(a)

1. We show that $((A \cap B) \cup (A \cap C) \subseteq A \cap (B \cup C))$ and $(A \cap (B \cup C) \subseteq (A \cap B) \cup (A \cap C))$ ...because of definition of equality
2. We assume $x \in A \cap (B \cup C)$ and show $x \in (A \cap B) \cup (A \cap C)$
3. Therefore, $x \in A \wedge x \in B \cup C$
4. Therefore, $x \in A \wedge (x \in B \vee x \in C)$
5. Therefore, $x \in A \wedge x \in B \vee x \in A \wedge x \in C$
6. Therefore, $x \in A \cap B \vee x \in A \cap C$
7. We are done with the current part of the proof (i.e., to show that $x \in (A \cap B) \cup (A \cap C)$). [It remains to be shown that $(A \cap B) \cup (A \cap C) \subseteq A \cap B \cup C$]
8. We assume $y \in (A \cap B) \cup (A \cap C)$ and show $y \in A \cap (B \cup C)$
9. Therefore, $y \in A \cap B \vee y \in A \cap C$
10. Therefore, $y \in A \wedge (y \in B \vee y \in C)$
11. Therefore, $y \in A \wedge y \in B \cup C$
12. This finishes the proof. Q.e.d.

(b)

Fig. 3. Comparison between (a) the (re-ordered) proof by Bartle and Sherbert [1] and (b) the proof presentation generated with our rule set from the ΩMEGA proof in Fig. 2

exciting natural language generation is easily possible with Fiedler's mechanisms [5]. The rule sets in Fig. 4 can be successfully reused for other examples in the domains as well (as demonstrated with a different proof exercise in [9]).

4 Learning from Empirical Data

We employ off-the-shelf machine learning tools to learn granularity rule sets (classifiers) from annotated examples (*supervised* learning), i.e. proof steps with the labels *appropriate*, *too small* or *too big*. Currently, our algorithm calls the C5.0 data mining tools [8]. To assess the performance of learning classifiers from human judgments, we have conducted a study where a mathematician (with tutoring experience) judged the granularity of 135 proof steps. These steps were presented to the mathematician via an ΩMEGA-assisted environment which computed the feature values for granularity classification in the background. The steps were (with some exceptions) generated at random step size, such that each presented step corresponded to one, two, or three assertion level inference steps (we also included a few single natural deduction (ND) steps for comparison[14]).

[14] We found that, unlike the assertion-level steps, single natural deduction steps were mostly rated as "too small" by the expert.

1) hypintro=1 ∧ total> 1 ⇒ too-big
2) ∪-Defn∈{1, 2}∧∩-Defn∈{1, 2} ⇒ too-big
3) ∩-Defn< 3 ∧ ∪-Defn=0 ∧ m.c.u.=1 ∧ unm.c.u.=0 ⇒ too-small
4) total<2 ∧ verb=true ⇒ too-small
5) m.c.u.<3 ∧ unm.c.u.=0 ∧ verb=true ⇒ too-small
6) eq-Defn>0 ∧ verb=false ⇒ too-big
7) _ ⇒ app.

(a)

1) conceptsunique ∈{0, 1} ∧ eq-Defn=0 ∧ verb=true ⇒ too-small
2) hypintro=0 ∧ eq-Defn=0 ∧ ∪-Defn=0 ∧ verb=true ⇒ too-small
3) conceptsunique ∈{2, 3, 4} ∧ ∪-Defn ∈{1, 2, 3} ⇒ too-big
4) hypintro ∈{1, 2, 3, 4} ∧ conceptsunique ∈{2, 3, 4} ⇒ too-big
5) unm.c.u.=0 ∧ total ∈{0, 1, 2} ∩-Defn ∈{1, 2} ∧ close=false ⇒ too-small
6) eq-Defn∈{1, 2} ∧ verb=false ⇒ too-big
7) eq-Defn∈{1, 2} ∧ verb=true ⇒ app.
8) eq-Defn=0 ∧ verb=false ⇒ app.
9) _ ⇒ app.

(b)

Fig. 4. Rule sets for our running example: (a) rule set generated by hand, (b) rule set generated the using C5.0 data mining tool (ordered by the rules' confidence values)

The presented proofs belonged to one exercise in naive set theory and three different exercises about binary relations. We used the Weka suite[15] to compare the performance of the PART classifier [10] which is inspired by Quinlan's C4.5 to the support vector machines implementation SMO [11], resulting in 86.9% and 85.4% of correct classification and Cohen's (unweighted) $\kappa = 0.65$ and $\kappa = 0.61$, respectively, in 10-fold cross validation, using only the 130 steps that were generated from assertion-level inferences (excluding the single ND steps). The results were achieved after we excluded some of the attributes (in particular those that refer to the use of specific concepts, i.e., Def. of ∩, Def. of ∘, etc.), which were relevant only in some of the exercises (possibly hampering generalizability of the learned classifiers), otherwise we obtained slightly worse 85.4% of correct classification and $\kappa = 0.61$ with PART.

5 Conclusion

Granularity has been a challenge in AI for decades [12, 13]. Here we have focused on adaptive proof granularity, which we treat as a classification problem. We model different levels of granularity using rule sets, which can be hand-authored or learned from sample proofs. Our granularity classifiers are applied dynamically to proof steps, taking into account changeable information such as the user's familiarity with the involved concepts. Using assertion level proofs as the basis for our approach is advantageous for the generation of natural language output, and the relevant information for the classification task (e.g., the concept names) is easily read off the proofs. Future work consists in further empirical evaluations of the learning approach — to address the questions: (i) what are the most useful features for judging granularity, and are they different among distinct experts

[15] `http://www.cs.waikato.ac.nz/~ml/weka/`

and domains, and (ii) what is the inter-rater reliability among different experts and the corresponding classifiers generated by learning in our framework? The resulting corpora of annotated proof steps and generated classifiers can then be used to evaluate the appropriateness of the proof presentations generated by our system.

Acknowledgments We thank four anonymous reviewers for their useful comments, and Marc Wagner and Claus-Peter Wirth for internal review.

References

1. Bartle, R.G., Sherbert, D.: Introduction to Real Analysis. 2 edn. Wiley (1982)
2. Benzmüller, C., Horacek, H., Kruijff-Korbayová, I., Pinkal, M., Siekmann, J.H., Wolska, M.: Natural language dialog with a tutor system for mathematical proofs. In Lu, R., Siekmann, J.H., Ullrich, C., eds.: Cognitive Systems. Volume 4429 of LNCS., Springer (2005) 1–14
3. Benzmüller, C., Horacek, H., Lesourd, H., Kruijff-Korbajova, I., Schiller, M., Wolska, M.: A corpus of tutorial dialogs on theorem proving; the influence of the presentation of the study-material. In: Proc. Intl. Conference on Language Resources and Evaluation (LREC 2006), Genoa, Italy, ELDA (2006)
4. Autexier, S., Benzmüller, C., Dietrich, D., Meier, A., Wirth, C.P.: A generic modular data structure for proof attempts alternating on ideas and granularity. [14] 126–142
5. Fiedler, A.: *P.rex*: An interactive proof explainer. In Goré, R., Leitsch, A., Nipkow, T., eds.: Automated Reasoning — IJCAR 2001. Number 2083 in LNAI, Siena, Italy, Springer Verlag (2001) 416–420
6. Denney, E., Power, J., Tourlas, K.: Hiproofs: A hierarchical notion of proof tree. In: MFPS XXI. Volume 155 of LNCS., Elsevier (2006) 341 – 359
7. Autexier, S., Fiedler, A.: Textbook proofs meet formal logic - the problem of underspecification and granularity. [14] 96–110
8. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)
9. Schiller, M., Benzmüller, C.: Granularity-adaptive proof presentation. Technical report, SEKI Working-Paper (2009) `http://arxiv.org/pdf/0903.0314v4`.
10. Frank, E., Witten, I.H.: Generating accurate rule sets without global optimization. In: Proc. 15th Intl. Conf. on Machine Learning, Morgan Kaufmann (1998) 144–151
11. Platt, J.C.: Fast training of support vector machines using sequential minimal optimization. In Schoelkopf, B., Burges, C., Smola, A., eds.: Advances in Kernel Methods - Support Vector Learning. MIT Press (1998) 185–208
12. Hobbs, J.R.: Granularity. In: Proc. of the 9th Int. Joint Conf. on Artificial Intelligence (IJCAI). (1985) 432–435
13. McCalla, G., Greer, J., Barrie, B., Pospisil, P.: Granularity hierarchies. Computers & Mathematics with Applications **23**(2-5) (1992) 363–375
14. Kohlhase, M., ed.: MKM'05. Volume 3863 of LNCS., Springer (2006)