

# Plano de Estudos

## MAC5701 - Tópicos em Ciência da Computação

Aluno - Thiago Carvalho de Sousa

Orientadora - Renata Wassermann

5 de abril de 2004

### 1 Introdução

O relatório aqui apresentado pretende descrever as atividades de pesquisa que realizaremos no primeiro semestre de 2004 como parte da disciplina de MAC5701 - Tópicos de Ciência da Computação. O trabalho será na área de Inteligência Artificial e Engenharia de Software e posteriormente pretendemos integrá-lo a nossa dissertação de mestrado.

### 2 Motivação

Durante o desenvolvimento de um sistema, há a fase de análise de requisitos. Nessa fase é frequente termos especificações que são inconsistentes umas com as outras. A capacidade de corrigir e analisar o impacto das mudanças no design de um sistema durante a fase de desenvolvimento é uma das tarefas mais árduas e custosas da engenharia de software. Pesquisas indicam que 80% dos custos de desenvolvimento de um sistema são consumidos pelas mudanças dos requisitos dos usuários. Essas mudanças ocorrem devido a alguns fatores, tais como : muitas vezes os usuários possuem visões e perspectivas diferentes do sistema; na maioria das vezes os usuários “esquecem” alguns requisitos; quase sempre os usuários desejam revisar a funcionalidade do sistema; nem sempre os engenheiros captam de maneira exata os requisitos dos usuários, etc. É preciso portanto ter uma ferramenta que mantenha de alguma forma a consistência desses requisitos e ao mesmo tempo permita revisá-los durante o processo de análise.

A nossa pesquisa tem como objetivo mostrar como a área de Revisão de Crenças se encaixa perfeitamente para solucionar esse problema. Mais especificamente, tentaremos integrar a solução proposta por esse campo da inteligência artificial a alguma linguagem de especificação formal de software.

### 3 Revisão de Crenças

A idéia principal de revisão de crenças, que é regida pelos postulados AGM, é a seguinte :

- Um agente possui um conjunto consistente de crenças (K).
- Esse agente recebe uma informação inconsistente com as suas crenças.
- Para aceitar essa nova informação, o agente deve abandonar algumas de suas crenças originais a fim de manter a consistência.

Um dos princípios que regem a teoria de revisão de crenças é que a mudança no meu conjunto original de crenças seja a mínima possível. Esse é o Princípio da Mudança Mínima. Em outras palavras, o que se deseja é reter o máximo de informação possível.

### 4 Notação Z

A especificação formal de um software nada mais é do que uma técnica de engenharia de software que usa notação matemática para descrever as funcionalidades de um sistema, e que detalha formas de validar esta especificação e a subsequente implementação.

Entre os mais conhecidos métodos formais de especificação de software encontra-se a notação Z, desenvolvida pela Universidade de Oxford no final dos anos 70. Z é uma notação pois não tem a intenção de ser algo executável, ou seja, não é uma linguagem de programação e sim apenas uma notação para detalhar funcionalidades de um sistema de software. A notação Z possui entre suas principais características o fato de ser tipada, ser baseada em lógica de primeira ordem e teoria dos conjuntos (ZF), ser popular na indústria, meios acadêmicos e governamentais e ser o único método formal de especificação com definição de padrão internacional (ISO).

## 5 Plano de Estudo

O objetivo será estudar em detalhes as ferramentas de suporte a notação Z. Dentre estas destacam-se HOL-Z, Cadiz, ProofPower e Alloy, que já foram pré-selecionadas por possuírem código aberto. Analisaremos essas ferramentas em vários aspectos, tais como tratamento de inconsistências lógicas, facilidade de uso, aceitação acadêmica e industrial, implementação e documentação. O passo seguinte será selecionar uma delas de acordo com os parâmetros mencionados e tentar estendê-la usando um algoritmo de revisão de crenças para solucionar o problema que motiva o nosso trabalho. Essa extensão vai se focar basicamente em re-escrever o código da ferramenta na parte relacionada ao provador de teoremas para detecção de inconsistências. Se possível ainda pretendemos aprofundar essa implementação usando outras idéias presentes na área de revisão de crenças.

## Referências

- [AGM85] C.E. Alchourron, P. Gardenfors, and D. Makinson. On the logic of theory change: Partial meet functions for contraction and revision. *Journal of Symbolic Logic*, (50):510–530, 1985.
- [AM82] C.E. Alchourron and D. Makinson. On the logic of theory change: Contraction functions and their associated revision functions. *Theoria*, (48):14–37, 1982.
- [DMMM95] D. Duffy, C. MacNish, J. McDermid, and P. Morris. A framework for requirements analysis using automated reasoning. In J. Iivari, K. Lyytinen, and M. Rossi, editors, *Advanced Information Systems Engineering: Proc. Seventh International Conference*, volume LNCS-932, pages 61–81. Springer-Verlag, 1995.
- [DMO97] Duffy, MacNish, and Osborne. An integrate framework for analysing changing requirements. 1997.
- [Gar92] P. Gardenfors, editor. *Belief Revision*. Cambridge Press, 1992.
- [Hal90] Anthony Hall. Seven myths of formal methods. *IEEE Software*, pages 11–20, 1990.

- [Jac02] Daniel Jackson. Alloy: a lightweight object modelling notation. *Software Engineering and Methodology*, 11(2):256–290, 2002.
- [Jon92] R.B. Jones. Icl proofpower. *BCS-FACS FACTS*, 1992.
- [KSW96] Kolyang, T. Santen, and B. Wolff. A structure preserving encoding of Z in Isabelle/HOL. In J. von Wright, J. Grundy, and J. Harrison, editors, *Proceedings of the 9th International Conference on Theorem Proving in Higher Order Logics*, pages 283–298, Turku, Finland, 1996. Springer-Verlag LNCS 1125.
- [Mar97] Andrew Martin. Why effective proof tool support for z is hard. *Software Verification Research Centre*, pages 97–34, 1997.
- [MW98] C.K. MacNish and M-A. Williams. From belief revision to design revision: Applying theory change to changing requirements. In *Learning and Reasoning with Complex Representations*, pages 207–222. Springer Verlag, 1998.
- [Spi01] Mike Spivey. The z notation: a reference manual. *Prentice Hall*, 2001.
- [TM95] Ian Toyn and John A. McDermid. Cadiz: An architecture for z tools and its implementation. *Software - Practice and Experience*, 25(3):305–330, 1995.
- [Wil97] M-A. Williams. Implementing belief revision. In G. Antoniou, editor, *Non-monotonic Reasoning*. MIT Press, 1997.