Reasoning Maintenance

IWIS Seminar
Aalborg 2008

Fred Durao
Intelligent Web and Information Systems (IWIS)
Computer Science Department
Aalborg University, Denmark

fred@cs.aau.dk

http://www.cs.aau.dk/~fred

http://iwis.cs.aau.dk

AALBORG UNIVERSITY

Summary

- Reasoning
- Reasoning Maintenance
- Truth Maintenance Systems
- Exercise

Concepts

Reasoning is the cognitive process of looking for reasons for beliefs, *conclusions*, actions or feelings [Christopher,1995].

- **Deductive reasoning**
  Premise 1: All humans are mortal.
  Premise 2: Socrates is a human.
  Conclusion: Socrates is mortal.
- **Inductive reasoning**
  Premise: The sun has risen in the east every morning up until now.
  Conclusion: The sun will also rise in the east tomorrow.
- **Argument from analogy**
  Premise1: A has characteristics x, y and z.
  Premise2: B has characteristics x and y.
  So, B has (or probably has) characteristic z

Kirwin, Christopher. 1995. 'Reasoning'. In Ted Honderich (ed.), *The Oxford Companion to Philosophy*. Oxford: Oxford University Press: p. 748

Concepts

Reason Maintenance provides a basis for both explanations and revision/versioning for reasoning.

Benefits:

Inconsistency detection;
Calls for explanations;
Revisions/versioning;
Detection of contradictory assumptions;
Fault diagnosis.

Kirwin, Christopher. 1995. 'Reasoning'. In Ted Honderich (ed.), *The Oxford Companion to Philosophy*. Oxford: Oxford University Press: p. 748

Truth Maintenance System

TMS is a knowledge representation method for representing both beliefs and their dependencies.

The name truth maintenance is due to the ability of these systems to **restore** consistency.

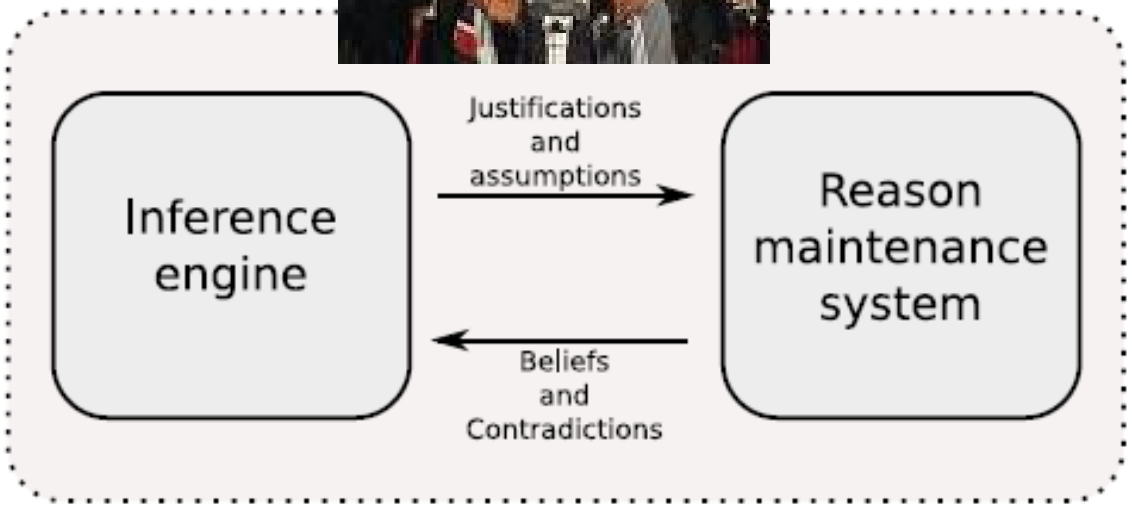
Reason Maintenance

TMS basically are problem solver that employ rules guiding choices of what to believe, what to want, and what to do

The Problem Solver Scenario

Reasoning

Reasoning Maintenance



The **inference** engine explores alternatives, makes choices, and examines the consequences of the choices. If a contradiction is detected during this process, the **TMS** eliminates it by revising the knowledge base.

TMS Variations

- **Justification-Based Truth Maintenance System (JTMS)**

It is a simple TMS where one can examine the **consequences** of the current set of assumptions. It justifies the truth as a means of tracking dependencies between inferences made by the reasoner.

Node 1: If **X** is a bug and **X** is new then it has to be fixed.

Node 2: **Y** is a bug and **Y** is new.

Node 3: **Z** is a bug and **Z** is closed.

Node 4: **Y** has to be fixed. (Outcome)

Justification of **Node 4**: Node 1 + Node 2

TMS Variations

- **Assumption-Based Truth Maintenance System (ATMS)**

It allows to maintain and reason with a number of simultaneous, possibly incompatible, current sets of assumption. It maintains alternative solution paths in parallel.

Node 1: If **X** is a bug and **X** is new then it has to be fixed.

Node 2: **Y** is a bug and **Y** is new.

Node 3: **Z** is a bug and **Z** is closed.

Justification and Assumption

Node 4: **Y** is not fixed. (Outcome)

Node 5: **Z** has to be fixed. (Belief)

Consistent Environment: **Node 4**: Node 1 + Node 2
Inconsistent Environment: **Node 5**: Node1 + **!** Node3

TMS Variations

Logical-Based Truth Maintenance System (LTMS)

More powerful than JTMS in that it recognizes the propositional **semantics** of sentences. Ideal for real time propositional reasoning

(TRUE) Node 1: If **X** is a bug and **X** is new.

(FALSE) Node 2: **Y** is a bug and **Y** is closed.

(TRUE) Node 3: **Z** is a bug.

Through **BCP** Boolean Constraint Prorogation, LTMS tries to label as TRUE of FALSE the nodes labeled as UNKNOWN.

(UNKNOWN) Node 4: If *Fred* fix **X** then *Fred* fix **Z**.

(UNKNOWN) Node 5: **Z** is closed.

Encyclopedia of Microcomputers
By Allen Kent, James G. Williams, Rosalind Kent, Carolyn M. Hall

Exercise: Reasoning Maintenance Scenario

$$featureSpecification(x) \rightarrow specification(x) \quad R_1$$
$$componentSpecification(x) \rightarrow specification(x) \quad R_2$$
$$specification(x) \wedge \neg revised(x) \rightarrow needsRevision(x) \quad R_3$$
$$needsRevision(x) \wedge reviewer(x,r) \rightarrow todo(x,r) \quad R_4$$

Rules 1: Tagging rules 1

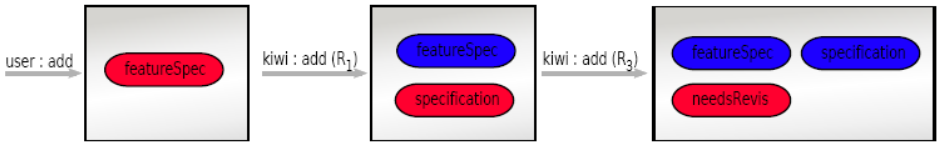


Figure 1: System infers two new tags after the user adds the "featureSpec" tag.

Then a user tags the page with the tag $reviewer(John)$, see Figure 2. Presence of this new tag enables KiWi to infer a new tag using rule R_4.

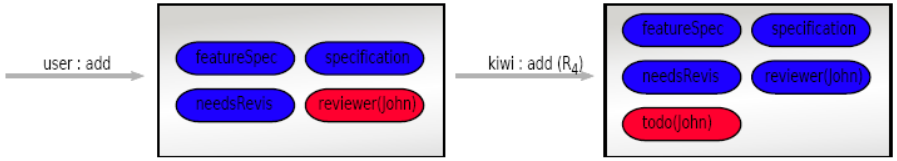


Figure 2: Deduction of tags with variables.

So far the scenario was only a straight-forward usage of tagging by user and by the system based on predefined rules. Now we will explore what interesting changes can be made in this state. There are several:

Change 1 User removes the $reviewer(John)$ tag.

Change 2 Or the user changes the $featureSpecification$ tag to $componentSpecification$ tag.

Change 3 Or the user removes the $needsRevision$ tag.

Change 4 Or the user adds a $revised$ tag.

Thank you!

Questions?

Fred Durao

Intelligent Web and Information Systems (IWIS)

Computer Science Department

Aalborg University, Denmark

`fred@cs.aau.dk`

`http://www.cs.aau.dk/~fred`

`http://iwis.cs.aau.dk`

AALBORG UNIVERSITY

Reference

Belief Revision and Truth Maintenance Systems: An Overview and a Proposal
http://www.cse.buffalo.edu/~shapiro/Papers/br-overview.pdf

Jon Doyle. A truth maintenance system. *Artificial Intelligence, 12:231–272, 1979.*

Encyclopedia of Microcomputers
By Allen Kent, James G. Williams, Rosalind Kent, Carolyn M. Hall