A Database Design Tool in Prolog

Paul Douglas & Steve Barker

Overview of Talk

- Motivations;
- Design Method;
- Implementation;
- Input/output;
- Evaluation;
- Future Work.

Motivations

Problem: Limitations of texts for assisting students to learn dependency theory (a core component of Computer Science).

Aim: to develop a learning tool for database design (using the process of “normalization”) that encourages *exploratory learning*.

Tool Design Methodology

- Dialogue-based analysis of students learning design principles;
- Gagne’s event-based theory of instruction;
- Formative evaluation for continuous iterative design.

Implementation Overview

Key features of our implementation:

- Based on well-known algorithms from the literature (cf. Ullman, 1988).
- Algorithms translated into Prolog (XSB).
- Java interface (YAJXB).

NB. We only consider *functional dependencies*.

User Interaction

Interaction is via menus.

User input:

- Schema/relation heading.
- FDs applicable to schema (in *right-reduced form*).
- Selection of 3NF/BCNF decomposition.

An explanation facility may be invoked by a user.

System Output

System output is a pair: $(\mathcal{S}, \mathcal{K})$ where:

- $\mathcal{S}$ is set of relations in a decomposition, $\mathcal{S} = \{s_1, \ldots, s_n\}$; and
- $\mathcal{K} = \{k_1, \ldots, k_n\}$ is the key for the s_i subschema (for $i = (1..n)$).

Invocation of the explanation facility provides details of the process of decomposition.

Evaluation

Formative evaluation conducted by:

- Expert reviewers;
- Small-group testing.

Note: A summative evaluation has recently been conducted.

Future Work

- Extension of existing software e.g., inclusion dependencies, MVDs, JDs.
- Further evaluation of current system.
- Integration of tool into a composite e-learning tool for database students.