Testing Database Applications

Faculty: Phyllis Frankl
Students:
   David Chays
   Yuetang Deng
   (Zhongqiang Chen)
   Weijin Zhao
   Leon Krupkin

Motivation

- Database systems play an important role in virtually every modern organization
- Faults can be very costly
- Little attention has been paid to DB application program correctness
Aspects of Correctness

- Does the DBMS perform all operations correctly?
- Is concurrent access handled correctly?
- Is the system fault-tolerant?
- ...

Does the application program behave as intended?

Traditional vs. DB programs

- imperative nature
- declarative nature
AGENDA tool set

- **Agenda Parser**
  - extracts information from the user’s schema
  - extracts information from application queries
  - extracts information from sample-values files
- **State generation** consists of populating the user’s DB State.
- **Input generation** consists of instantiating the input parameters with actual values.
- **State validation** consists of examining the change in DB state.
- **Output validation** consists of examining the result of executing the test case.

On-going and Future Work

- Java/JDBC front end (and others?)
- Improve handling of semantic constraints
- Additional empirical evaluation
- Extensions to web applications
- Testing other aspects of DB applications correctness, including security
Further information

- “Testing Database Transaction Concurrency”, *ASE04*.
- “Test Database Transaction Consistency”, Submitted to *ICSE04*.
- “A framework for testing database applications”, *ISSTA 2000*.