Projects Overview

1. (T+P) Data Integration/Mediation
   - Implement the Global-as-View (GAV)/view unfolding approach (intro to it: TODAY)
   - Examples, documentation, presentation (needed for all projects)

2. (T: deep) Data Integration/Mediation
   - Read/learn about other DI algorithms: Local-as-View (LAV), mixed approaches
   - Summarize, report, present

3. (T: broad) Intro/Overview on Schema Matching approaches

4. (T) Knowledge Representation & Ontologies of Biological Information
   - Gene Ontology, BioCyc, Pathway databases, …

5. (P) Scientific Workflows/Kepler
   - (5A) Analysis-intensive workflows (focus on data analysis, some data transformations, some visualization)
   - (5B) Data-intensive workflows / SDSC Storage Resource Broker (collection management on the “Data Grid”)
   - (5C) Compute-intensive workflows / Condor, NIMROD, APST, … (job scheduling on cluster computers, simulations, “Compute Grid”)
   - (5D) Database-intensive workflows (TBD)
   - (5E) Real-time data access workflows / ROADNet (accessing real-time data streams, some analysis and visualization)

Data Integration (Mediator System)
Global-as-View (GAV) & View Unfolding

- In GAV data integration, rules have the form
  \[ \ldots \text{R}_G(\ldots) \leftarrow \ldots \text{R}_L(\ldots) \ldots \]
- The *user query* is against the global (=integrated) views (relations) \( \text{R}_G \):
  \[ \text{?- R}_G(\ldots) \]
- The system must come up with a *query plan* that eliminates references to \( \text{R}_G \) relations, and only keeps \( \text{R}_L \) relations (and maybe new aux-relations)
- \( \Rightarrow \) GAV query rewriting
- Essentially “view unfolding”

Global-as-View (GAV) & View Unfolding

- Simplistic example:
- User query: ?- ans(X)
- View definitions
  - \( \text{ans}(X) \leftarrow \text{p}(X,X), \text{u}(X,Y) \)
  - \( \text{p}(X,Y) \leftarrow \text{q}(X,Z), \text{r}(Z,Y) \)
- Algorithm:
  - Given a goal (rhs) \( G \), unify it with a matching head (lhs) \( H \) of a rule \( H(\text{e}ad) \leftarrow \text{B(ody)} \)
  - Replace \( G \) by \( B \), taking into account the substitution \( \sigma \) needed for unifying \( G \) and \( H \)
  - Repeat until all relations are given (=EDB) relations

Querying vs Reasoning

Q1: answer1(S,C) \leftarrow
    student(S, N), takes(S, C), course(C, X), inCS(C), course(C, “DB”)
Q2: answer2(S,C) \leftarrow
    student(S, N), takes(S, C), course(C, X)
- We can “run” both queries \( Q \) on a given database instance \( D \) \( \Rightarrow \) Query evaluation (answer = eval\((Q,D)\))
- Note: The answers to \( Q1 \) are always contained in the answers to \( Q2 \) (why?)
- Determining whether a query \( Q \) is contained (for all database instances \( D! \)) in another query \( Q’ \) is a reasoning problem.