Cooperative Query Answering for Conjunctive Queries by Iterative Generalization

Dr. Lena Wiese, lena.wiese@uni-hildesheim.de
(Institut für Informatik, Fachbereich 4, Universität Hildesheim)

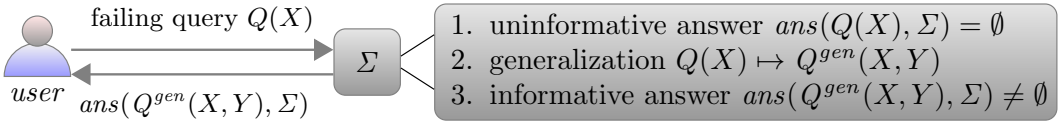
Outline

Failing queries & Informative Answers

- A database system may not be able to answer queries in a satisfactory manner
- If a database answer is empty, the corresponding query is said to be a *failing query*
- *Cooperative database systems* search for *informative answers*
  - not exactly matching the user's original query
  - provide data that are "closely related" to the user's intention
- Aim: User-friendly information systems that automatically support users in finding relevant information
- Similar: search engines, web shops or expert systems provide users with data that might be close to their interest

Logical Foundations

- Query answering in knowledge bases
- Generalization operators that can be applied to conjunctive queries
- Related work:
  - Sakama and Inoue (2007): generate "neighborhood proposals" in a negotiation process using DC, AI and GR
  - Halder and Cortesi (2011): apply generalization operators to single conjuncts in a conjunctive query incrementally
  - Motro et al (1990): Flex; Godfrey et al (1994): Carmin; Chu et al (1996): CoBase; Godfrey (1997): Ishmael; ...



Knowledge bases

- Generic data model for the representation of knowledge in multiagent systems, belief revision or ontology-based reasoning.
- Σ: set of formulas of some logic; universally closed
- Knowledge base represents a set of "possible worlds" (set of models of the knowledge base)
- We assume that a knowledge base is consistent: it has at least one possible world

Example (knowledge base)

Mary has cough or flu:

$$\Sigma = \{\mathit{Ill}(\mathsf{Mary}, \mathsf{Cough}) \vee \mathit{Ill}(\mathsf{Mary}, \mathsf{Flu})\}$$

A possible world would be $\{\mathit{Ill}(\mathsf{Mary}, \mathsf{Cough})\}$ making only this single ground atom $true$ and all other ground atoms $false$.

Queries & Answers

Definition (Conjunctive Query)

A *query* is a conjunctive formula $L_1 \wedge \ldots \wedge L_n$ where each L_i is a literal. We often abbreviate a query as $Q(X)$, where Q stands for the conjunction of literals and X is an n-tuple of variables appearing in Q.

Definition (Answer set)

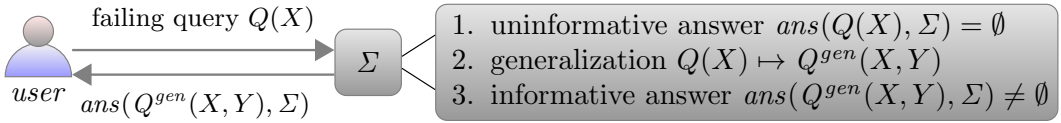
For a query $Q(X)$ and a knowledge base Σ, the set of correct answers (or *answer set*, for short) $ans(Q(X), \Sigma)$ is a set of closed formulas such that for each $\phi \in ans(Q(X), \Sigma)$ it holds that $\Sigma \models \phi$ and ϕ is derived from $Q(X)$ by some query answering semantics.

Definition (Failing query)

Let Σ be a knowledge base, $Q(X)$ be a query. If $ans(Q(X), \Sigma) = \emptyset$, the query $Q(X)$ *fails* (in Σ).

Query Generalization

$Q(X)$ is transformed into a more general query $Q^{gen}(X, Y)$ that has a non-empty answer in the knowledge base:



Definition (Deductive generalization – cf. Gaasterland, 1992)

Let Σ be a knowledge base, $\phi(X)$ be a formula (with free variables X), $\psi(X, Y)$ (with additional free variables Y). Then $\psi(X, Y)$ is a *deductive generalization* of $\phi(X)$, if it holds in Σ that ϕ ("less general") implies ψ ("more general"); for X: universal closure $\forall$; for Y existential closure $\exists$:

$$\Sigma \models \forall X \exists Y \, (\phi(X) \rightarrow \psi(X, Y))$$

Query Generalization

Example

III(y, Cough) is more general than *III*$(\text{Mary}, \text{Cough})$:
For any Σ: $\Sigma \models \exists y(\textit{III}(\text{Mary}, \text{Cough}) \rightarrow \textit{III}(y, \text{Cough}))$

III(Mary, y') and *III*(y, y') are more general than *III*$(\text{Mary}, \text{Cough})$, too

III(y, y') is more general than *III*(y, Cough):
For any Σ: $\Sigma \models \forall y \exists y' \, (\textit{III}(y, \text{Cough}) \rightarrow \textit{III}(y, y'))$

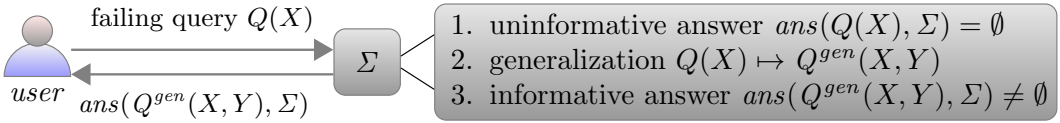
For *III*(y, Cough) and *III*(Mary, y') neither is a generalization of the other

Query Generalization

Definition (Generalized query with informative answer)

A query formula $Q^{gen}(X, Y)$ is a *generalized query with informative answer* for a query $Q(X)$ (with respect to a knowledge base Σ) if the following properties hold:

1. **Failing query:** $ans(Q(X), \Sigma) = \emptyset$
2. **Generalized query:** $Q^{gen}(X, Y)$ is deductive generalization of $Q(X)$
3. **Informative answer:** $ans(Q^{gen}(X, Y), \Sigma) \neq \emptyset$



Query Generalization

Definition (Generalization operator)

o is a *generalization operator* if $o(S) = S'$ $(S, S'$: sets of formulas), and for each $\psi \in S'$ there is $\phi \in S$: ψ is deductive generalization of ϕ.

Definition (Generalization sets / Operator-based distance)

Σ knowledge base, $\phi(X)$ formula, and $GenOp$ generalization operators;
$\mathcal{G}_i$ set of formulas obtained by applying i operators from $GenOp$ to $\phi(X)$ in any possible way (without allowing formulas that are equivalent)

$\mathcal{G}_0 := \{\phi(X)\}$

$\mathcal{G}_i := \{\psi(X,Y) \mid \psi(X,Y) \in o(\mathcal{G}_{i-1})$ where $o \in GenOp$ and for every $j \leq i$ there is no $\psi'(X,Y') \in \mathcal{G}_j$ such that $\psi'(X,Y') \equiv \psi(X,Y)\}$

Then, $\psi(X,Y)$ has *distance* l to $\phi(X)$ if $\psi(X,Y) \in \mathcal{G}_l$
($\psi(X,Y)$ is obtained by applying at least l gen. operators to $\phi(X)$)

Outline

Logical Generalization Operators

- Dropping Condition (DC) removes one conjunct from a query
  - looks for answers that satisfy less conditions than the original query
- Anti-Instantiation (AI) replaces a constant (or a variable occurring at least twice) in the original query with a new variable
  - looks for answers that have less restrictions in terms of concrete values for or equality constraints on variables at a certain position
- Goal Replacement (GR) applies a deduction rule to the query
  - rule can be a database constraint or more generally be an element of a knowledge base Σ on which the query is evaluated
  - A rule consists of a body part left of an implication arrow ($\rightarrow$) and a head part right of the implication arrow
  - GR operator then finds a substitution θ that maps the rule's body to some conjuncts in the query and replaces these conjuncts by the head (with θ applied).
- Generalization operators in terms of relational calculus and relational algebra

Selection-Projection-Join queries

- Represent positive conjunctive queries in relational algebra
- Projection π_A with A a set of attributes
- Selection σ_C with C a set of equality conditions
  - Attribute equal to a constant $A_j = a_j$
  - Attribute equal to attribute $A_j = A_k$
- Renaming ρ_R with R a set of renaming conditions
  - $A_{ij} \leftarrow A'_{ij}$
  - to avoid identical attribute names in joined relations
- Join $\bowtie_E$ with E a set of equality conditions (equi-join)
  - Attribute equal to attribute $A_j = A_k$ (join attributes)
- Relations $R_1 \dots R_n$
- SPJ-query in general: $\pi_A \, [\sigma_C \, (\rho_R(R_1) \bowtie_E \dots \bowtie_E \rho_R(R_n)) \,]$

Example: Health record

- Failing query: "Who is ill with Flu and Cough?"
- Relational calculus: $Q(x) = \mathit{Ill}(x, \text{Flu}) \wedge \mathit{Ill}(x, \text{Cough})$
- Relational algebra:

$$\pi_{N_1}\ [\sigma_{D_1=\text{Flu}, D_2=\text{Cough}}\ (\rho_{N_1 \leftarrow Name, D_1 \leftarrow Disease}(\mathit{Ill}) \bowtie_{N_1=N_2}\ \rho_{N_2 \leftarrow Name, D_2 \leftarrow Disease}(\mathit{Ill}))\ ]$$

Ill:

Name	*Disease*
Mary	Cough
Mary	BrokenLeg
Mary	Sinusitis
Pete	Flu

Treat:

Name	*Prescription*
Mary	Inhaler

Algebraic Versions of DC, AI and GR

- DC: drop one relation R_i (including the renaming ρ_R) from the join expression
  - and add or remove conditions from A, C and E
- AI: add an attribute to the set A of projection attributes
  - and remove any condition for this attribute from C or E
- GR: replace several of the relations R_i from the join expression with a single new relation (including the appropriate renaming ρ_R)
  - and add or remove conditions from A, C and E

Dropping Condition (DC) in Relational Calculus

For conjunctive queries, ignoring one of the conjuncts makes the query more general:

- $Q(X) = L_1 \wedge \ldots \wedge L_n$ of length n
- Choose literal L_j from $L_1, \ldots, L_n$
- $Q^{gen}(X) = L_1 \wedge \ldots \wedge L_{j-1} \wedge L_{j+1} \wedge \ldots \wedge L_n$

Proposition

DC is a deductive generalization operator.

Example

$Q^{gen}(X) = \mathit{Ill}(x, \text{Cough})$ (Who has cough?)
or $Q^{gen}(X) = \mathit{Ill}(x, \text{Flu})$ (Who has flu?)

Dropping Condition (DC) in Relational Algebra

- Failing query: $\pi_{N_1}\ [\sigma_{D_1=\text{Flu},D_2=\text{Cough}}\ ($
$\rho_{N_1 \leftarrow Name, D_1 \leftarrow Disease}(Ill)$
$\bowtie_{N_1=N_2}\ \rho_{N_2 \leftarrow Name, D_2 \leftarrow Disease}(Ill))\ ]$

Ill:

$Name$	$Disease$
Mary	Cough
Mary	BrokenLeg
Mary	Sinusitis
Pete	Flu

$Treat$:

$Name$	$Prescription$
Mary	Inhaler

- DC:
$\pi_{N_1}\ [\sigma_{D_1=\text{Flu}}\ (\rho_{N_1 \leftarrow Name, D_1 \leftarrow Disease}(Ill))\ ]$ with answer Pete
$\pi_{N_2}\ [\sigma_{D_2=\text{Cough}}\ (\rho_{N_2 \leftarrow Name, D_2 \leftarrow Disease}(Ill))\ ]$ with answer Mary

Anti-instantiation (AI) in Relational Calculus

With anti-instantiation a new variable is introduced in the query:

- Query $Q(X) = L_1 \wedge \ldots \wedge L_n$ of length n
- From $Q(X)$ choose a term t such that t is
  - either a variable occurring in $Q(X)$ at least twice
  - or a constant
- Choose one literal L_j where t occurs
- Let L'_j be the literal with one occurrence of t replaced with a new variable
- $Q^{gen}(X) = L_1 \wedge \ldots \wedge L_{j-1} \wedge L'_j \wedge L_{j+1} \wedge \ldots \wedge L_n$

Proposition

AI is a deductive generalization operator.

Anti-instantiation (AI) in Relational Calculus

Example

$Q(X) = \mathit{Ill}(x, \text{Flu}) \wedge \mathit{Ill}(x, \text{Cough})$ (Who has cough and flu?)

$Q^{gen}(X) = \mathit{Ill}(y, \text{Flu}) \wedge \mathit{Ill}(x, \text{Cough})$ (Who has cough and who has flu?)

or

$Q^{gen}(X) = \mathit{Ill}(x, y) \wedge \mathit{Ill}(x, \text{Cough})$

(Who has cough and possibly another disease?)

or

$Q^{gen}(X) = \mathit{Ill}(x, \text{Flu}) \wedge \mathit{Ill}(x, y)$

(Who has flu and possibly another disease?)

Anti-instantiation (AI) in Relational Algebra

- Failing query: $\pi_{N_1}\ [\sigma_{D_1=\text{Flu},D_2=\text{Cough}}\ ($
$\rho_{N_1 \leftarrow Name, D_1 \leftarrow Disease}(Ill)$
$\bowtie_{N_1=N_2}\ \rho_{N_2 \leftarrow Name, D_2 \leftarrow Disease}(Ill))\ ]$
- AI:
$\pi_{N_1,N_2}\ [\sigma_{D_1=\text{Flu},D_2=\text{Cough}}\ ($
$\rho_{N_1 \leftarrow Name, D_1 \leftarrow Disease}(Ill)$
$\bowtie\ \rho_{N_2 \leftarrow Name, D_2 \leftarrow Disease}(Ill))\ ]$

Answer:

N_1	N_2
Pete	Mary

Anti-instantiation (AI) in Relational Algebra

- Failing query: $\pi_{N_1}\ [\sigma_{D_1=\text{Flu},D_2=\text{Cough}}\ ($
$\rho_{N_1 \leftarrow Name, D_1 \leftarrow Disease}(Ill)$
$\bowtie_{N_1=N_2}\ \rho_{N_2 \leftarrow Name, D_2 \leftarrow Disease}(Ill))\ ]$
- AI:
$\pi_{N_1, D_1}\ [\sigma_{D_2=\text{Cough}}\ ($
$\rho_{N_1 \leftarrow Name, D_1 \leftarrow Disease}(Ill)$
$\bowtie_{N_1=N_2}\ \rho_{N_2 \leftarrow Name, D_2 \leftarrow Disease}(Ill))\ ]$

Answer:

N_1	D_1
Mary	Cough
Mary	BrokenLeg
Mary	Sinusitis

Anti-instantiation (AI) in Relational Algebra

- Failing query: π_{N_1} [$\sigma_{D_1=\text{Flu},D_2=\text{Cough}}$ (
$\rho_{N_1 \leftarrow Name, D_1 \leftarrow Disease}(Ill)$
$\bowtie_{N_1=N_2}$ $\rho_{N_2 \leftarrow Name, D_2 \leftarrow Disease}(Ill)$)]
- AI:
π_{N_1,D_2} [$\sigma_{D_1=\text{Flu}}$ (
$\rho_{N_1 \leftarrow Name, D_1 \leftarrow Disease}(Ill)$
$\bowtie_{N_1=N_2}$ $\rho_{N_2 \leftarrow Name, D_2 \leftarrow Disease}(Ill)$)]

Answer:

N_1	D_2
Pete	Flu

Goal Replacement (GR) in Relational Calculus

Goal replacement checks if the body of a rule in the knowledge base can be mapped (via a substitution) to a subquery
If so, the head of the rule replaces the subquery (with the substitution applied):

- Query $Q(X) = L_1 \wedge \ldots \wedge L_n$ of length n
- From Σ choose a rule $L_{i_1} \wedge \ldots \wedge L_{i_m} \rightarrow L'$ such that there is a substitution θ for which all literals $L_{i_1}\theta, \ldots, L_{i_m}\theta$ occur in $Q(X)$
- Let $L_{i_{m+1}}, \ldots, L_{i_n}$ denote remaining literals of $Q(X)$
- $Q^{gen}(X) = L_{i_{m+1}} \wedge \ldots \wedge L_{i_n} \wedge L'\theta$

Note: We allow only rules with single literal in head (right of $\rightarrow$) and only range-restricted ones (variables in head also appear in body)

Proposition

GR is a deductive generalization operator.

Goal Replacement (GR) in Relational Calculus

Example

$Q(X) = \mathit{Ill}(x, \text{Flu}) \wedge \mathit{Ill}(x, \text{Cough})$ (Who has cough and flu?)

Rule $\mathit{Ill}(x, \text{Flu}) \rightarrow \mathit{Treat}(x, \text{Medi})$ in Σ

$Q^{gen}(X) = \mathit{Treat}(x, \text{Medi}) \wedge \mathit{Ill}(x, \text{Cough})$

(Who is treated with Medi and has cough?)

Difficulty of GR: Σ has to be checked for rules with a matching body.

Goal Replacement (GR) in Relational Algebra

- Failing query: $\pi_{N_1}\ [\sigma_{D_1=\text{Flu},D_2=\text{Cough}}\ ($
$\rho_{N_1 \leftarrow Name, D_1 \leftarrow Disease}(Ill)$
$\bowtie_{N_1=N_2}\ \rho_{N_2 \leftarrow Name, D_2 \leftarrow Disease}(Ill))\ ]$

- GR: Suppose every patient suffering from Flu is treated with Inhaler, that is,
$Ill(x, \text{Flu}) \rightarrow Treat(x, \text{Inhaler})$
then, ask for patients suffering from Flu and being treated with Inhaler
$\pi_{N_1}\ [\sigma_{P_1=\text{Inhaler},D_2=\text{Cough}}\ ($
$\rho_{N_1 \leftarrow Name, P_1 \leftarrow Prescription}(Treat)$
$\bowtie_{N_1=N_2}\ \rho_{N_2 \leftarrow Name, D_2 \leftarrow Disease}(Ill))\ ]$

Answer:

N_1
Mary

Outline

Assigning weights to answer tuples

- Informative answers (obtained by generalization operators) do not exactly match the user's query.
- If there are several such informative answers: which of them are the "best"?
  - Which of them are most similar to the original query?

Answer:

N_1	D_1
Mary	Cough
Mary	BrokenLeg
Mary	Sinusitis

- Assign a similarity value to each tuple (row) in the answer table
  - Based on a similarity between constants (values in the query and the informative answers)
- Similarity values are numbers from the unit interval $[0 \ldots 1]$

Similarity between constants

- For constants a and b determine $sim(a, b)$
- Different options:
  - for proper nouns (like Flu and Cough): taxonomies/ontologies of proper nouns (like WordNet), similarity is for example defined by distance between nouns in the ontology
  - for non-proper nouns (names like Mary and Pete): acquaintance (distance in a social network or family tree), or geographic proximity
  - for numerical values: scaling function based on difference between the numbers and the range from which they are taken

Similarity-Based AI

- AI on Flu (selection condition):

  $\pi_{N_1,D_1}\ [\sigma_{D_2=\text{Cough}}\ ($
  $\rho_{N_1 \leftarrow Name, D_1 \leftarrow Disease}(Ill)$
  $\bowtie_{N_1=N_2}\ \rho_{N_2 \leftarrow Name, D_2 \leftarrow Disease}(Ill))\ ]$

  Assume $sim(Flu, Cough) = 0.8$, $sim(Flu, BrokenLeg) = 0.4$, and $sim(Flu, Sinusitis) = 0.9$

  Answer:

N_1	D_1	sim
Mary	Cough	0.8
Mary	BrokenLeg	0.4
Mary	Sinusitis	0.9

- Use a threshold: For example, remove all tuples with similarity lower than 0.5

Similarity-Based AI

- AI on Cough (selection condition):

$$\pi_{N_1, D_2}\ [\sigma_{D_1=\text{Flu}}\ (\rho_{N_1 \leftarrow Name, D_1 \leftarrow Disease}(Ill) \bowtie_{N_1=N_2}\ \rho_{N_2 \leftarrow Name, D_2 \leftarrow Disease}(Ill))\ ]$$

Answer:

N_1	D_2	sim
Pete	Flu	0.8

- Use some kind of aggregation function to decide which AI step is better

Similarity-Based DC

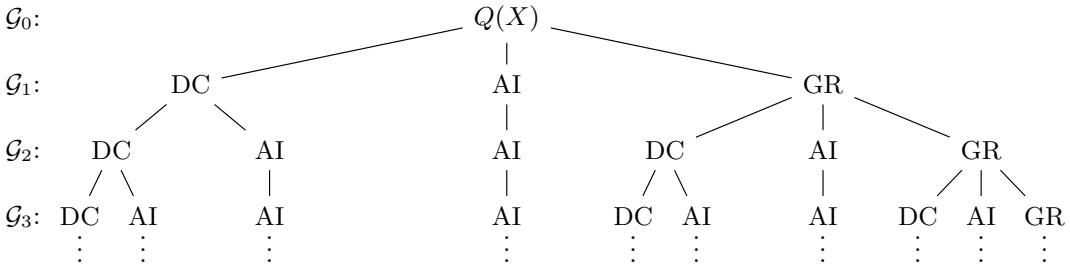
- Heuristic: predicates with less attributes (predicates with a lower arity) convey less information and hence it is preferable to drop them
  - If R_i is the removed relation, subtract from 1 the ratio of the arity $ar(R_i)$ with respect to the sum of all arities of predicates taking part in the join as $1 - (ar(R_i)/\Sigma_{j=1}^{n}(ar(R_j)))$
  - same similarity degree for all answer tuples: only Ill relation in the join, hence all answer tuples have degree 0.5
- Heuristic: dropping a condition where less attributes are removed from the projection is preferred
- ...
- Semantic (constant-based) similarity: If a constant is dropped (like Flu), determine aggregation of similarity values between the dropped constants and the constants in an answer tuple
  - Hence, ensure that those tuples in an answer table get assigned a higher similarity degree the constants of which are more related to the constants in the dropped condition

Similarity-Based GR

- In general, similar to DC
- Consider as more positive the case that attributes are replaced with GR (instead of totally dropped with DC).
  - For example, as the selection condition for Flu is replaced by one for Inhaler with the example rule, this would not be penalized as in the DC case
- Semantic similarity for GR: obtain a similarity degree by aggregating similarity values for the constants in the replacement rule. For the example rule, assuming that $sim(\text{Flu}, \text{Inhaler}) = 0.5$, the answer tuples obtained after GR would all have this similarity degree

Outline

Generalization Tree



Theorem (Operator ordering)

When combining the generalization operators DC, AI and GR, the following computations can be avoided: GR following DC, DC following AI, and GR following AI.

Operator Ordering

Lemma (DC following AI)

$S_1 = AI(DC(Q(X)))$
(set of queries obtained by dropping conditions and then anti-instantiating)
$S'_1 = DC(Q(X))$ *(set of queries obtained by dropping conditions),* $S_2 = DC(AI(Q(X)))$
(set of queries obtained by anti-instantiating and then dropping conditions)
Then $S_2 \subseteq S_1 \cup S'_1$ *(up to variable renaming).*

When anti-instantiating a term and then dropping the conjunct that contains the term, the two operations coincide with dropping the conjunct alone (and hence the query belongs to set S'_1). When however after anti-instantiating a term, a conjunct different from the one containing the term is dropped, the operators can be applied in reverse order (and hence the query belongs to set S_1). It follows that it suffices to apply AI after DC. In the reverse direction, it can also be shown that $S_1 \subseteq S_2$.

Operator Ordering

Lemma (GR following DC)

$S_1 = DC(GR(Q(X)))$
(set of queries obtained by replacing goals and then dropping conditions)
$S_2 = GR(DC(Q(X)))$
(set of queries obtained by dropping conditions and then replacing goals)
Then $S_2 \subseteq S_1$.

When a conjunct is dropped from a query and then a goal replacement is executed on the remaining query, the same can be achieved by first applying the rule and then dropping the conjunct. Hence we can avoid the application of the GR operator after the DC operator. The reverse however does not hold: in some cases DC results in a query to which GR cannot be applied anymore

Operator Ordering

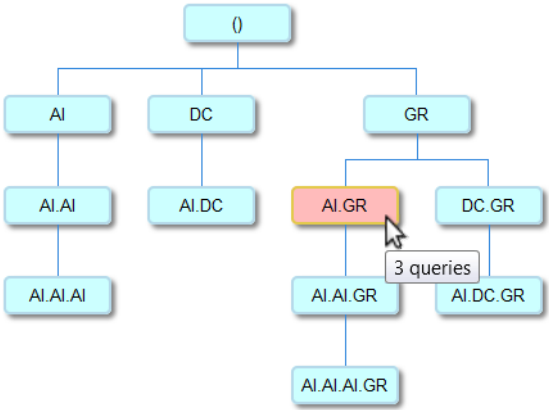
Lemma (GR following AI)

$S_1 = AI(GR(Q(X)))$
(set of queries obtained by replacing goals and then anti-instantiating)
$S_1' = GR(Q(X))$ *(set of queries obtained by goal replacement alone)*
$S_2 = GR(AI(Q(X)))$
(set of queries obtained by anti-instantiating and then replacing goals)
Then $S_2 \subseteq S_1 \cup S_1'$ *(up to variable renaming).*

Assume that some term t in the original query is anti-instantiated to the new variable y; then the substitution θ used in goal replacement maps some variable of the rule to y. However the same can be achieved by first applying goal replacement by letting θ map the variable to t and then (if it is still contained in the replacement literal) anti-instantiating it to y.
The reverse direction does not hold because after AI some rule used for GR may not be applicable anymore.

Protopype Implementation

- Generalization operators written in Java
- Query answering based on a current theorem proving system (SOLAR)



Queries and answers

View all

```
Query: ill(X,flu) & treat(X,V_6)
Answer(s):
[no answer]

Query: ill(V_6,flu) & treat(X,medi)
Answer(s):
[X->pete, V_6->mary]

Query: ill(X,V_6) & treat(X,medi)
Answer(s):
[X->pete, X->pete, V_6->cough]
```

Outline

Conclusion

- Although a given query fails, a knowledge base might still be able to return informative answers to a slightly modified query.
- We provided logical and algebraic versions of the three operators DC, AI and GR that relax conjunctive queries (Selection-Projection-Join queries) and informally discussed similarity of answer tuples
- Open issues
  - formalize the notion of similarity in a logical framework
  - neighborhood queries (by reinstantiating after AI) or conditional answers (provide a condition under which an answer holds)
  - algorithm that directly and incrementally returns informative answers without computing all the sets of generalized queries