Safety and Domain Independence:

Domain Independence

Additional slides, Part I

for

Introduction to Database Systems 236363-Spring 2012
Domain-independent queries

Let $A_1,\ldots,A_m$ be attributes and $R,S,T$ be relation schemas.

Let $D_1,\ldots,D_m$ be domains for these attributes.

Let $E$ be a query expression (in RA or DRC).

Let $r,s,t$ be relations over $D_1,\ldots,D_m$ interpreting $R,S,T$.

A set of domains $D'_1,\ldots,D'_m$ is **admissible for** $r,s,t$ if for each $i \leq m$ the projections $\pi_{A_i}r$, $\pi_{A_i}s$ and $\pi_{A_i}t$ are in $D'_i$.

A query $\phi(\bar{x})$ is **domain independent** if for all domains **admissible for** $r,s,t$ the result of evaluating the query over $r,s,t$ is the same.
RA expressions $E$ are domain independent

We prove this by structural induction over $E$.

- $E$ is $R, S$ or $T$.
  By the definition of admissible domains.

- Intersection, Union and Difference.
  By the induction hypothesis and the definition of admissible domains.

- Selection and Projection.
  Both make the table smaller.

- Cartesian product.
  By the induction hypothesis and the definition of admissible domains.

- Renaming.
  Does not change the content of the tables.
There are DRC expressions which are not domain independent

Examples of formulas which are not domain independent.

- $\neg R(\bar{x})$
- $\forall \bar{x} S(\bar{x})$

Examples of formulas which are domain independent.

- $S(\bar{x}) \land \neg R(\bar{x})$
- $\forall \bar{x} (S(\bar{x}) \rightarrow T(\bar{x}))$
Why is domain independence important?

- The *compiler (interpreter)* of queries processes RA-expressions.
- The *human user* writes queries in DRC or Datalog.
- Therefore we have to know which DRC expressions are compilable, i.e., *translatable into RA*.

**Note:** There is **no algorithm** which takes a DRC expression \( E \) as input and decides whether \( E \) is domain independent.

We prove such results in Logic 2

**However,** we can define *easily recognizable* subsets of DRC expressions which are all domain independent.
Adding predicates for the actual domain, I

Let $\delta_i^R(y)$ the formula which defines $\pi_{A_i}R$.

For $R[A_1, \ldots, A_m]$ the formula $\delta_i^R(y)$ for $\pi_{A_i}R$ is

$$\exists x_1 \exists x_2 \ldots \exists x_{i-1} \exists x_{i+1} \exists x_m R(x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_m)$$

Similarly for $S[A_1, \ldots, A_m]$ the formula $\delta_i^S(y)$ for $\pi_{A_i}S$ is

$$\exists x_1 \exists x_2 \ldots \exists x_{i-1} \exists x_{i+1} \exists x_m S(x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_m)$$

and the analogously for $\delta_i^T(y)$.

Finally, $\delta_i(y)$ is the formula $\delta_i^R(y) \land \delta_i^S(y) \land \delta_i^T(y)$.
Adding predicates for the actual domain, II

We can define inductively ADRC (DRC with actual domains):

- The atomic formulas $R(\bar{x})$, $S(\bar{x})$ and $T(\bar{x})$ are in ADRC.
- $\delta_i(x)$ is in ADRC.
- The formulas $\delta_i(x) \land \delta_i(y) \land x = y$ are in ADRC.
- Closure under $\land, \lor$
- If $\phi(x_1, \ldots, x_m)$ is in ADRC the also $\delta_1(x_1) \land \ldots \land \delta_m(x_m) \land \neg \phi$ is in ADRC.
- Closure under $\exists x (\delta_i(x) \land \phi(x))$ and $\forall x (\neg \delta_i(x) \lor \phi(x))$

**Exercise:** Show that all formulas of ADRC are domain independent.
Problems with ADRC

- It can be shown that every domain independent DRC expression is equivalent to an ADRC expression.

- Every ADRC expression can be converted into an RA expression and vice versa.

- **However**, ADRC is **not a convenient** formalism. Formulas get too long and are awkward to read.

- We now discuss an alternative: **Safe Range DRC**.

- There is another alternative: **SafeDRC** defined in the TIRGUL 3b_DBMStut-RC.ppt.
Safety and Domain Independence:

Safe Range DRC Formulas

Additional slides, Part II

for

Introduction to Database Systems 236363-Spring 2012

Based on [AHV95]

Foundations of Databases
by S. Aibiteboul, R. Hull and V. Vianu
Addison-Wesley 1995
pages 83ff.
Safety and domain independence

J.A. Makowsky

Safe range normal form SRNF

We first preprocess formulas of DRC

**Variable substitution:** Change name of all the variables of $\phi$ such that free and bound variables are different and no two bound variables bound by different quantifiers are the same.

**remove $\forall$:** Replace $\forall x \phi$ by $\neg \exists \neg \phi$.

**remove $\rightarrow$:** Replace by $(\neg \phi \lor \psi)$

**Push negations inside:** Replace $\neg \neg \phi$ by $\phi$, Apply de Morgan rules to move negations inside as much as possible. $\neg \phi$ stays only if $\phi$ is atomic or is of the form $\neg \exists x \psi$.

**Flatten the formula:** Replace

$$(\phi_1 \land (\phi_2 \land \phi_3))$$

by

$$(\phi_1 \land \phi_2 \land \phi_3)$$

and similarly for $\lor$. 

and similarly for $\lor$. 

10
Example: \((\forall x (T(x) \to S(x, y)) \land \exists x S(x, y))\)

- The free variable is \(y\), and \(x\) is bound by two quantifiers.
- \((\forall x_1 (T(x_1) \to S(x_1, y)) \land \exists x_2 S(x_2, y))\)
- \((\forall x_1 (\neg T(x_1) \lor S(x_1, y)) \land \exists x_2 S(x_2, y))\)
- \((\neg \exists x_1 (\neg \neg T(x_1) \lor S(x_1, y)) \land \exists x_2 S(x_2, y))\)
- \((\neg \exists x_1 (\neg \neg T(x_1) \land \neg S(x_1, y)) \land \exists x_2 S(x_2, y))\)
- \((\neg \exists x_1 (T(x_1) \land \neg S(x_1, y)) \land \exists x_2 S(x_2, y))\)

\((\neg \exists x_1 (T(x_1) \land \neg S(x_1, y)) \land \exists x_2 S(x_2, y))\) is in SRNF.
Example: $\forall x (\exists y S(x, y) \rightarrow T(x, y))$

- The free variable is $y$ in $T(x, y)$, in $S(x, y)$ $y$ is bound.
- $\forall x (\exists y_1 S(x, y_1) \rightarrow T(x, y))$
- $\neg \exists x \neg (\exists y_1 S(x, y_1) \rightarrow T(x, y))$
- $\neg \exists x (\neg \exists y_1 S(x, y_1) \lor T(x, y))$
- $\neg \exists x (\neg \neg \exists y_1 S(x, y_1) \land \neg T(x, y))$
- $\neg \exists x (\exists y_1 S(x, y_1) \land \neg T(x, y))$

$\neg \exists x (\exists y_1 S(x, y_1) \land \neg T(x, y))$ is in SRNF.
Normal Form Theorem for SRNF

**Theorem:** Every DRC-formula is equivalent to a formula in SRNF with the same set of free variables.

**Proof:**

- Rename all the necessary variables.
- Eliminate all occurrences $\forall$ and $\rightarrow$.
- Apply de Morgan rules as long as you can.
- All this can be done while preserving logical equivalence.
For a formula $\phi$ we define the set of **range restricted variables** $rr(\phi)$ inductively as follows:

Atomic formulas: If $\phi$ is of the form $R(v_1, \ldots, v_n)$ or $v_i = a$ or $a = v_i$ then $rr(\phi) = free(\phi)$.

Conjunction: If $\phi = (\phi_1 \land \phi_2)$ then $rr(\phi) = rr(\phi_1) \cup rr(\phi_2)$.

Equalities: If $\phi = (\psi \land v_i = v_j)$ then
\[
rr(\phi) = \begin{cases} 
rr(\psi) & \text{if } \{v_i, v_j\} \cap rr(\psi) = \emptyset \\
rr(\psi) \cup \{v_i, v_j\} & \text{else}
\end{cases}
\]

Disjunction: If $\phi = (\phi_1 \lor \phi_2)$ then $rr(\phi) = rr(\phi_1) \cap rr(\phi_2)$.

Negations: If $\phi = \neg \psi$ then $rr(\phi) = \emptyset$.

Existential quantifier: If $\phi = \exists v_1, \ldots, v_m \psi$ then
\[
rr(\phi) = \begin{cases} 
rr(\psi) - \{v_1, \ldots, v_m\} & \text{if } \{v_1, \ldots, v_m\} \subseteq rr(\psi) \\
\bot & \text{else}
\end{cases}
\]
Range restricted variables, II

We define for any set of variables $V$:
$\bot \cup V = \bot \cap V = \bot - V = V - \bot = \bot$ and use commutativity of $\cup$ and $\cap$.

The outcome of this check is

either $rr(\phi)$ is a set of free variables of $\phi$
or $rr(\phi) = \bot$

$\phi$ is **safe range** if $rr(\phi) = free(\phi)$. 
Which formulas can be made safe range?

If $V$ is a proper subset of $\text{free}(\phi)$ and $rr(\phi) = V$ then $\phi$ is not safe range.

However, we can look for some $\theta$ with $rr(\theta) = \text{free}(\theta) = \text{free}(\phi) - V$.

and the query $\phi \land \theta$ will be safe range, but not necessarily equivalent to $\phi$.

Let $V = \{v_1, \ldots, v_r\}$ and $\text{free}(\phi) - V = \{y_1, \ldots, y_k\}$ we can take $\theta(\bar{y})$ to be

$$\Delta(\bar{y}) = \bigwedge_j \left( \bigvee_i \delta_i(y_j) \right)$$

We then have

$$\models \forall y_1, \ldots, \forall y_k (\Delta(\bar{y} \rightarrow ((\phi(\bar{y}, \bar{v}) \land \Delta(\bar{y}) \leftrightarrow \phi(\bar{y}, \bar{v})))$$

which says that $\phi$ and $\phi \land \Delta$ produce the same table if restricted to the actual domain.
Example: $\phi = \forall x (\exists y S(x, y) \rightarrow T(x, y))$ and $\text{free}(\phi) = \{y\}$

We first put $\phi$ in SRNF $\psi$: $\psi = \neg \exists x (\exists y_1 S(x, y_1) \land \neg T(x, y))$.

Then we compute $rr(\psi)$ as follows:

- $rr(S(x, y_1)) = \{x, y_1\}$.
- $rr(T(x, y)) = \{x, y\}$.
- $rr(\neg T(x, y)) = \emptyset$
- $rr(\exists y_1 S(x, y_1)) = \{x\}$
- $rr((\exists y_1 S(x, y_1) \land \neg T(x, y))) = \{x\}$
- $rr(\exists x (\exists y_1 S(x, y_1) \land \neg T(x, y))) = \emptyset$
- $rr(\psi) = \emptyset$

$\psi$ is not safe range but $(\psi(y) \land \exists x T(x, y))$ is safe range.

Note, $\psi$ begins with a negation. Still we can not conclude that $rr(\psi) = \emptyset$ without the previous steps, because if $rr(\phi_1) = \bot$ then $rr(\neg \phi_1) = \bot$ as well.
THEOREM: The standard translation of RA into DRC gives safe range formulas

We have to assume that $\theta$ the select operator $\sigma_\theta E$ is quantifier free and uses equality only.

We prove this by (structural) induction:

We have the boolean operators $\lor, -$ and projection $\pi_A E$, selection $\sigma_\theta E$ and cartesian products $E_1 \times E_2$.

- If $E = R$ atomic, then $\phi_E = R(\bar{v})$ and $\text{free}(R(\bar{v})) = \{\bar{v}\}$.

- Let $E = E_1 \times E_2$, and the free variables in $\phi_{E_1}$ and $\phi_{E_2}$ form disjoint sets $V_1$ and $V_2$.
  If both $\phi_{E_1}$ and $\phi_{E_2}$ are safe range, i.e.,
  $\text{free}(\phi_{E_1}) = \text{rr}(\phi_{E_1}) = V_1$ and $\text{free}(\phi_{E_2}) = \text{rr}(\phi_{E_2}) = V_2$,
  then $\phi_E = \phi_{E_1} \land \phi_{E_2}$ and $\text{rr}(\phi_E) = \text{free}(\phi_E) = V_1 \cup V_2$. 
Proof of the THEOREM continued

• Let $E = E_1 - E_2$, and
  
  $free(\phi_{E_1}) = free(\phi_{E_2}) = rr(\phi_{E_1}) = rr(\phi_{E_2})$.

  Then $\phi_E = \phi_{E_1} \land \neg \phi_{E_2}$ and
  
  $rr(\neg \phi_{E_2}) = \emptyset$ and
  
  $rr(\phi_E) = rr(\phi_{E_1}) = free(\phi_{E_1}) = free(\phi_E)$.

• For $E = \sigma_{\theta} E_1$ we first have to compute $rr(\theta)$.

  $\theta$ is quantifier free, so $rr(\theta) \neq \bot$.

  By induction hypothesis $rr(\phi_{E_1}) = free(\phi_{E_1})$ and $rr(\theta) = free(\theta)$,
  and $\phi_{\sigma_{\theta} E_1} = \phi_{E_1} \land \theta$.

  Therefore, $\phi_{E_1} \land \theta$ is safe range.

• The cases for $E = E_1 - E_2$ and $E = \pi_A E_1$ are left as exercises.
Translating $S \div T$ gives a safe range formula

We have seen that

$$\phi = (\neg \exists x_1 (T(x_1) \land \neg S(x_1, y)) \land \exists x_2 S(x_2, y))$$

is in SRNF.

We also know from previous lectures that for $E = S \div T$
the standard translation of $S \div T$ is logically equivalent to $\phi$.

**Homework:** Verify, that $\phi$ is safe range.
Safe Range and Domain Independence.

- We know that RA expressions are domain independent.
- We have shown that standard translations of RA expressions give safe range DRC formulas.
- Therefore, every RA expression corresponds to a safe range DRC formula.

**Theorem:** [AHV95, Theorem 5.4.6]

For every safe range DRC formula $\psi$ we can find algorithmically an RA expression $E_\psi$ such that its standard translation $\phi_{E_\psi}$ is logically equivalent to $\psi$.

In other words, every safe range DRC formula corresponds to an RA expression.

In particular, every safe range DRC formula is domain independent.
Summary: What you have to know for the exam

- SRNF: Convert formulas in SRNF
- Compute $rr(\phi)$ of DRC formulas in SRNF.
- Check whether a DRC formula is safe range.
- If a DRC formula $\phi$ is not safe range, can you always find a safe range DRC formula $\theta$ such that $\phi \land \theta$ is safe range?
Comparing to SafeDRC

Read the definition of SafeDRC from the TIRGUL 3b_DBMStut-RC.ppt.

Show the following:

- Every formula in SafeDRC is domain invariant.  
  \textbf{Hint:} Use induction.

- Show that every SafeDRC formula is a safe range formula.  
  \textbf{Hint:} Put it into SRNF and apply the algorithm.

- Find a formula which is safe range but not in SafeDRC.