Safe DRC Formulas[‡]

Real query languages based on relational calculus use only a subset of the DRC formulas, ones that are guaranteed to be domain independent. We shall define "safe" formulas to be a subset of the domain independent formulas. The important properties of safety are:

a) Every "safe" formula must be domain independent.
b) We can tell easily, just by inspecting a formula, whether or not it is "safe."
c) The formulas that are expressible in real query languages based on relational calculus are "safe."

With these criteria, the following are definitions of *safe DRC formulas*.

1. There are no uses of the $\forall$ quantifier. This constraint does not affect the expressiveness of the language, because $(\forall X)F$ is logically equivalent to $\neg(\exists X)\neg F$. That is, F is true for all X if and only if there does not exist an X for which F is false. By applying this transformation wherever we find a $\forall$, we can eliminate all universal quantifiers.
2. Whenever an OR operator is used, the two formulas connected, say $F_1 \vee F_2$, have the same set of variables; i.e., they are of the form
$$F_1(X_1,...,X_n) \vee F_2(X_1,...,X_n)$$
3. Consider any maximal subformula consisting of the conjunction of one or more formulas $F_1 \wedge \cdots \wedge F_m$. Then all variables appearing free in any of these F_i's must be *limited* in the following sense.
   a) A variable is limited if it is free in some F_i, where F_i is not an arithmetic comparison and is not negated.
   b) If F_i is $X = a$ or $a = X$, where a is a constant, then X is limited.
   c) If F_i is $X = Y$ or $Y = X$, and Y is a limited variable, then X is limited.

   It is important to note that this rule applies to atomic formulas if they do not appear in a larger group of formulas connected by logical AND. For example, if our entire formula is $X = Y$, it is not safe because it is the "conjunct" of one formula, and the variables X and Y are not limited. Likewise, the formula $X = Y \vee p(X,Y)$ is not safe because the left operand of the OR violates rule (3). However, $X = Y \wedge p(X,Y)$ is safe, because $p(X,Y)$ limits both X and Y by (a).
4. A $\neg$ operator may only apply to a term in a conjunction of the type discussed in rule (3). In particular, a subformula $\neg G$ violates the "safety" definition unless it is part of a larger subformulas
$$H_1 \wedge \cdots \wedge H_i \wedge \neg G \ \wedge I_1 \wedge \cdots \wedge I_j.$$
   where at least one of the H's and I's is positive (not negated).

[‡] This material is taken from Ullman, J. D., *Principles of Database and Knowledge-Base Systems, Vol. 1*, Computer Science Press, Rockville, Md, 1988.

Safe Tuple Relational Calculus

As in DRC, we can write TRC formulas, such as $\neg r(u)$, that denote infinite relations. We want to avoid them in real query languages, and as with DRC, the most useful approach is to define a restricted form of TRC called "safe TRC."

Since the arity of a tuple variable is not always clear from the context in a TRC formula, we shall assume that the arity of each variable is given and that the arity of one variable does not change from occurrence to occurrence, even if the two occurrences are bound by different quantifiers. The safety of a TRC formula is defined as follows, in close analogy with the definition of safe DRC.

1. There are no uses of the $\forall$ quantifier.
2. Whenever an $\vee$ operator is used, the two formulas connected, say $F_1 \vee F_2$, have only one free tuple variable, and it is the same variable.
3. Consider any subformula consisting of a maximal conjunction of one or more formulas $F_1 \wedge \cdots \wedge F_m$. Then all components of tuple variables that are free in any of these F_i's are *limited* in the following sense.
   a) If F_i is a nonnegated, ordinary atomic formula $p(\mu)$, then all components of tuple variable μ are said to be limited.
   b) If F_i is $\mu[j] = a$ or $a = \mu[j]$, where a is a constant, then $\mu[j]$ is limited.
   c) If F_i is $\mu[j] = \nu[k]$ or $\nu[k] = \mu[j]$, and $\nu[k]$ is a limited variable, then $\mu[j]$ is limited.
4. A $\neg$ operator may only apply to a term in a conjunction of the type discussed in rule (3).