Exploiting Preference Rules for Querying Databases

Sergio Greco, Cristian Molinaro, Francesco Parisi

D.E.I.S.

Università della Calabria

{greco, cmolinaro,fparisi}@deis.unical.it

15th Italian Symposium on Advanced Database Systems, SEBD 2007
Torre Canne, Fasano (BR),Italy

Motivations

Large

volume of

available

data

Information Filtering and Extraction

Expressing preferences on alternative scenarios is natural

Preferences can be exploited to reduce the volume of data presented to the user, thus improving the query answering

Main Idea

- We answer to queries by deriving only supported and preferred information

DB: { beef, red-wine, white-wine }

P: { fruit-salad $\leftarrow$ white-wine,
pie $\leftarrow$ red-wine,
biscuits $\leftarrow$ red-wine }

Φ: { red-wine > white-wine $\leftarrow$ beef,
pie > biscuits $\leftarrow$ }

UNIVERSITÀ DELLA CALABRIA
Dipartimento di ELETTRONICA, INFORMATICA E SISTEMISTICA

Main Idea

- We answer to queries by deriving only supported and preferred information

DB: { beef, red-wine, white-wine }   beef   red-wine   white-wine

P: { fruit-salad ← white-wine,
pie ← red-wine,
biscuits ← red-wine }

Φ: { red-wine > white-wine ← beef,
pie > biscuits ← }

UNIVERSITÀ DELLA CALABRIA
Dipartimento di ELETTRONICA, INFORMATICA E SISTEMISTICA

Main Idea

- We answer to queries by deriving only supported and preferred information

DB: { beef, red-wine, white-wine }

beef red-wine ~~white-wine~~

P: { fruit-salad ← white-wine,
pie ← red-wine,
biscuits ← red-wine }

Φ: { red-wine > white-wine ← beef,
pie > biscuits ← }

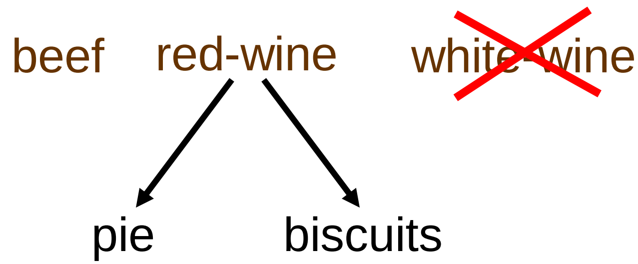
Main Idea

- We answer to queries by deriving only supported and preferred information

DB: { beef, red-wine, white-wine }

P: { fruit-salad ← white-wine,
pie ← red-wine,
biscuits ← red-wine }

Φ: { red-wine > white-wine ← beef,
pie > biscuits ← }



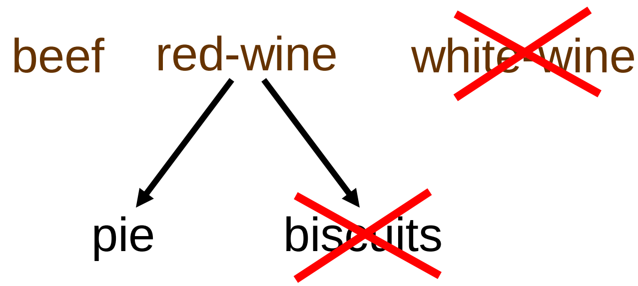
Main Idea

- We answer to queries by deriving only supported and preferred information

DB: { beef, red-wine, white-wine }

P: { fruit-salad ← white-wine,
pie ← red-wine,
biscuits ← red-wine }

Φ: { red-wine > white-wine ← beef,
pie > biscuits ← }



Answer={ beef, red-wine, pie }

Preference Rules

- A *preference rule* is of the form

$$A > C \leftarrow B_1, \ldots, B_m,\ not\ B_{m+1}, \ldots,\ not\ B_n,\ \varphi$$

  - *A* is *preferable* to *C* if the body of the rule is *true*
  - *C* is *dominated* by *A* if the body of the rule is *true*

red-wine > white-wine ← beef

beef red-wine white-wine

Preference Rules

- A *preference rule* is of the form

$$A > C \leftarrow B_1, \ldots, B_m, \textit{ not } B_{m+1}, \ldots, \textit{ not } B_n, \varphi$$

  - *A* is *preferable* to *C* if the body of the rule is *true*
  - *C* is *dominated* by *A* if the body of the rule is *true*
  - *dominated atoms* cannot be used to infer new information

red-wine > white-wine ← beef

beef red-wine ~~white-wine~~

white-wine is dominated by red-wine

UNIVERSITÀ DELLA CALABRIA
Dipartimento di ELETTRONICA, INFORMATICA E SISTEMISTICA

Preference Rules

- A *preference rule* is of the form

$$A > C \leftarrow B_1, \ldots, B_m, \text{ not } B_{m+1}, \ldots, \text{ not } B_n, \varphi$$

  - *A* is *preferable* to *C* if the body of the rule is *true*
  - *C* is *dominated* by *A* if the body of the rule is *true*
  - *dominated atoms* cannot be used to infer new information

P: { fruit-salad ← white-wine,
pie ← red-wine,
biscuits ← red-wine }

beef red-wine ~~white-wine~~

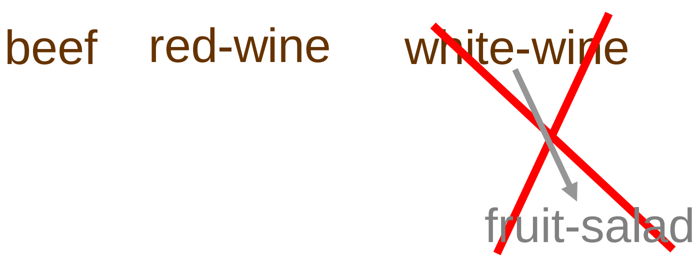
Preference Rules

- A *preference rule* is of the form

$$A > C \leftarrow B_1,\ldots,B_m,\ not\ B_{m+1},\ldots,\ not\ B_n,\ \varphi$$

  - *A* is *preferable* to *C* if the body of the rule is *true*
  - *C* is *dominated* by *A* if the body of the rule is *true*
  - *dominated atoms* cannot be used to infer new information

P: { fruit-salad ← white-wine,
pie ← red-wine,
biscuits ← red-wine }



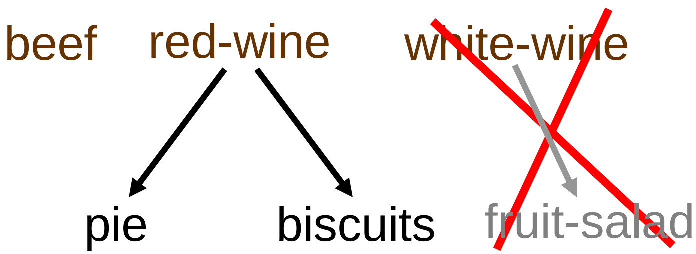
Preference Rules

- A *preference rule* is of the form

$$A > C \leftarrow B_1,\ldots,B_m,\ not\ B_{m+1},\ldots,\ not\ B_n,\ \varphi$$

  - *A* is *preferable* to *C* if the body of the rule is *true*
  - *C* is *dominated* by *A* if the body of the rule is *true*
  - *dominated atoms* cannot be used to infer new information

P: { fruit-salad ← white-wine,
pie ← red-wine,
biscuits ← red-wine }



Preferences on Base Atoms

- Preference program Φ

Φ: { ρ_1 = beef > fish $\leftarrow$,
ρ_2 = white-wine > red-wine $\leftarrow$ fish,
ρ_3 = red-wine > white-wine $\leftarrow$ beef }

  – intuitively, the evaluation of ρ_2 and ρ_3 depends on the evaluation of ρ_1

- Φ is layered as follows:

Layer 0 : { ρ_1 **}**
Layer 1 : { ρ_2 , ρ_3 **}**

DB: { beef, fish, red-wine, white-wine }

UNIVERSITÀ DELLA CALABRIA
Dipartimento di ELETTRONICA, INFORMATICA E SISTEMISTICA

Preferences on Base Atoms

- Preference program Φ

Φ: { ρ_1 = beef > fish $\leftarrow$,
ρ_2 = white-wine > red-wine $\leftarrow$ fish,
ρ_3 = red-wine > white-wine $\leftarrow$ beef }

  – intuitively, the evaluation of ρ_2 and ρ_3 depends on the evaluation of ρ_1

- Φ is layered as follows:

Layer 0 : { ρ_1 }
Layer 1 : { ρ_2 , ρ_3 }

$\rightarrow$ beef ~~fish~~ red-wine white-wine

DB: { beef, fish, red-wine, white-wine }

UNIVERSITÀ DELLA CALABRIA
Dipartimento di ELETTRONICA, INFORMATICA E SISTEMISTICA

Preferences on Base Atoms

- Preference program Φ

  Φ: { ρ_1 = beef > fish $\leftarrow$,
  ρ_2 = white-wine > red-wine $\leftarrow$ fish,
  ρ_3 = red-wine > white-wine $\leftarrow$ beef }

  – intuitively, the evaluation of ρ_2 and ρ_3 depends on the evaluation of ρ_1

- Φ is layered as follows:

Layer 0 : { ρ_1 **}**	$\rightarrow$	beef ~~fish~~ red-wine white-wine
Layer 1 : { ρ_2 , ρ_3 **}**	$\rightarrow$	beef red-wine ~~white-wine~~

DB: { beef, fish, red-wine, white-wine }   Answer = { beef, red-wine }

UNIVERSITÀ DELLA CALABRIA
Dipartimento di ELETTRONICA, INFORMATICA E SISTEMISTICA

Preferences on Base Atoms

- Preference program Φ

  Φ: { ρ_1 = beef > fish $\leftarrow$,
  ρ_2 = white-wine > red-wine $\leftarrow$ fish,
  ρ_3 = red-wine > white-wine $\leftarrow$ beef }

  – intuitively, the evaluation of ρ_2 and ρ_3 depends on the evaluation of ρ_1

- Φ is layered as follows:

Layer 0 : { ρ_1 } → beef ~~fish~~ red-wine white-wine

Layer 1 : { ρ_2 , ρ_3 } → beef red-wine ~~white-wine~~

- It is possible to define sufficient conditions which guarantee that the set of preference rules Φ can be partitioned into layers

General Preferences

- Preferences on both base and derived atoms
- Stratified semantics
  - a program P is partitioned into strata
  - preference rules are *associated* with strata of P
  - for each stratum of P, its preference rules are divided into layers
  - P is evaluated by computing one stratum at a time
    - for each stratum of P, the *associated* preference rules are applied one layer at a time

UNIVERSITÀ DELLA CALABRIA
Dipartimento di ELETTRONICA, INFORMATICA E SISTEMISTICA

General Preferences

- (Stratified) Datalog program P

P: Lunch (X) $\leftarrow$ Menu (X)
Dinner (X) $\leftarrow$ Menu (X), not Lunch (X)
Dinner (fruit-salad) $\leftarrow$ Dinner (white-wine)
Dinner (ice-cream) $\leftarrow$ Dinner (white-wine)
Dinner (pie) $\leftarrow$ Dinner (red-wine).

- Preference program Φ

Φ: $\rho_1 =$ Lunch (beef) > Lunch (fish) $\leftarrow$,
$\rho_2 =$ Lunch (red-wine) > Lunch (white-wine) $\leftarrow$ Lunch (beef)
$\rho_3 =$ Lunch (white-wine) > Lunch (red-wine) $\leftarrow$ Lunch (fish)
$\rho_4 =$ Dinner (fruit-salad) > Dinner (ice-cream) $\leftarrow$ Dinner (fish)
$\rho_5 =$ Dinner (ice-cream) > Dinner (fruit-salad) $\leftarrow$ Dinner (beef)

General Preferences

- (Stratified) Datalog program P

P: Lunch (X) ← Menu (X) **Stratum S_1**

Dinner (X) ← Menu (X), not Lunch (X)
Dinner (fruit-salad) ← Dinner (white-wine)
Dinner (ice-cream) ← Dinner (white-wine)
Dinner (pie) ← Dinner (red-wine).

Stratum S_2

- Preference program Φ

Φ: $\rho_1 =$ Lunch (beef) > Lunch (fish) ←,
$\rho_2 =$ Lunch (red-wine) > Lunch (white-wine) ← Lunch (beef)
$\rho_3 =$ Lunch (white-wine) > Lunch (red-wine) ← Lunch (fish)
$\rho_4 =$ Dinner (fruit-salad) > Dinner (ice-cream) ← Dinner (fish)
$\rho_5 =$ Dinner (ice-cream) > Dinner (fruit-salad) ← Dinner (beef)

General Preferences

- (Stratified) Datalog program P

P: **Lunch** (X) ← Menu (X) **Stratum S_1**
Dinner (X) ← Menu (X), not Lunch (X)
Dinner (fruit-salad) ← Dinner (white-wine)
Dinner (ice-cream) ← Dinner (white-wine)
Dinner (pie) ← Dinner (red-wine).

- Preference program Φ

Φ: ρ_1= Lunch (beef) > **Lunch** (fish) ←, preferences on atoms defined by S_1
ρ_2 = Lunch (red-wine) > **Lunch** (white-wine) ← Lunch (beef)
ρ_3 = Lunch (white-wine) > **Lunch** (red-wine) ← Lunch (fish)
ρ_4 = Dinner (fruit-salad) > Dinner (ice-cream) ← Dinner (fish)
ρ_5 = Dinner (ice-cream) > Dinner (fruit-salad) ← Dinner (beef)

UNIVERSITÀ DELLA CALABRIA
Dipartimento di ELETTRONICA, INFORMATICA E SISTEMISTICA

General Preferences

- (Stratified) Datalog program P

P: Lunch (X) $\leftarrow$ Menu (X)
Dinner (X) $\leftarrow$ Menu (X), not Lunch (X)
Dinner (fruit-salad) $\leftarrow$ Dinner (white-wine)
Dinner (ice-cream) $\leftarrow$ Dinner (white-wine)
Dinner (pie) $\leftarrow$ Dinner (red-wine).

Stratum S_2

- Preference program Φ

Φ: ρ_1= Lunch (beef) > Lunch (fish) $\leftarrow$,
ρ_2 = Lunch (red-wine) > Lunch (white-wine) $\leftarrow$ Lunch (beef)
ρ_3 = Lunch (white-wine) > Lunch (red-wine) $\leftarrow$ Lunch (fish)
ρ_4 = Dinner (fruit-salad) > **Dinner** (ice-cream) $\leftarrow$ Dinner (fish)
ρ_5 = Dinner (ice-cream) > **Dinner** (fruit-salad) $\leftarrow$ Dinner (beef)

preferences on atoms defined by S_2

General Preferences

- (Stratified) Datalog program P

P: Lunch (X) ← Menu (X)
Dinner (X) ← Menu (X), not Lunch (X)
Dinner (fruit-salad) ← Dinner (white-wine)
Dinner (ice-cream) ← Dinner (white-wine)
Dinner (pie) ← Dinner (red-wine).

Menu	beef	fish	red-wine	white-wine

- Preference program Φ

Φ: ρ_1= Lunch (beef) > Lunch (fish) ←,
ρ_2 = Lunch (red-wine) > Lunch (white-wine) ← Lunch (beef)
ρ_3 = Lunch (white-wine) > Lunch (red-wine) ← Lunch (fish)
ρ_4 = Dinner (fruit-salad) > Dinner (ice-cream) ← Dinner (fish)
ρ_5 = Dinner (ice-cream) > Dinner (fruit-salad) ← Dinner (beef)

DB: { Menu (beef), Menu (fish), Menu (red-wine), Menu (white-wine) }

UNIVERSITÀ DELLA CALABRIA
Dipartimento di ELETTRONICA, INFORMATICA E SISTEMISTICA

General Preferences

- (Stratified) Datalog program P

P: Lunch (X) ← Menu (X)
Dinner (X) ← Menu (X), not Lunch (X)
Dinner (fruit-salad) ← Dinner (white-wine)
Dinner (ice-cream) ← Dinner (white-wine)
Dinner (pie) ← Dinner (red-wine).

Menu	beef	fish	red-wine	white-wine
Lunch	beef	fish	red-wine	white-wine

- Preference program Φ

Φ: ρ_1= Lunch (beef) > Lunch (fish) ←,
ρ_2 = Lunch (red-wine) > Lunch (white-wine) ← Lunch (beef)
ρ_3 = Lunch (white-wine) > Lunch (red-wine) ← Lunch (fish)
ρ_4 = Dinner (fruit-salad) > Dinner (ice-cream) ← Dinner (fish)
ρ_5 = Dinner (ice-cream) > Dinner (fruit-salad) ← Dinner (beef)

DB: { Menu (beef), Menu (fish), Menu (red-wine), Menu (white-wine) }

UNIVERSITÀ DELLA CALABRIA
Dipartimento di ELETTRONICA, INFORMATICA E SISTEMISTICA

General Preferences

- (Stratified) Datalog program P

P: **Lunch** (X) ← Menu (X)
Dinner (X) ← Menu (X), not Lunch (X)
Dinner (fruit-salad) ← Dinner (white-wine)
Dinner (ice-cream) ← Dinner (white-wine)
Dinner (pie) ← Dinner (red-wine).

Menu	beef	fish	red-wine	white-wine
Lunch	beef	~~fish~~	red-wine	~~white-wine~~

- Preference program Φ

Φ: ρ_1 = Lunch (beef) > **Lunch** (fish) ←,
ρ_2 = Lunch (red-wine) > **Lunch** (white-wine) ← Lunch (beef)
ρ_3 = Lunch (white-wine) > **Lunch** (red-wine) ← Lunch (fish)
ρ_4 = Dinner (fruit-salad) > Dinner (ice-cream) ← Dinner (fish)
ρ_5 = Dinner (ice-cream) > Dinner (fruit-salad) ← Dinner (beef)

Layer 0 : { ρ_1 }
Layer 1 : { ρ_2 , ρ_3 }

DB: { Menu (beef), Menu (fish), Menu (red-wine), Menu (white-wine) }

UNIVERSITÀ DELLA CALABRIA
Dipartimento di ELETTRONICA, INFORMATICA E SISTEMISTICA

General Preferences

- (Stratified) Datalog program P

P: Lunch (X) ← Menu (X)
Dinner (X) ← Menu (X), not Lunch (X)
Dinner (fruit-salad) ← Dinner (white-wine)
Dinner (ice-cream) ← Dinner (white-wine)
Dinner (pie) ← Dinner (red-wine).

Menu	beef	fish	red-wine	white-wine
Lunch	beef	~~fish~~	red-wine	~~white-wine~~
Dinner		fish		white-wine
		fruit-salad	ice-cream	

- Preference program Φ

Φ: ρ_1 = Lunch (beef) > Lunch (fish) ←,
ρ_2 = Lunch (red-wine) > Lunch (white-wine) ← Lunch (beef)
ρ_3 = Lunch (white-wine) > Lunch (red-wine) ← Lunch (fish)
ρ_4 = Dinner (fruit-salad) > Dinner (ice-cream) ← Dinner (fish)
ρ_5 = Dinner (ice-cream) > Dinner (fruit-salad) ← Dinner (beef)

DB: { Menu (beef), Menu (fish), Menu (red-wine), Menu (white-wine) }

UNIVERSITÀ DELLA CALABRIA
Dipartimento di ELETTRONICA, INFORMATICA E SISTEMISTICA

General Preferences

- (Stratified) Datalog program P

P: Lunch (X) ← Menu (X)
Dinner (X) ← Menu (X), not Lunch (X)
Dinner (fruit-salad) ← Dinner (white-wine)
Dinner (ice-cream) ← Dinner (white-wine)
Dinner (pie) ← Dinner (red-wine).

Menu	beef fish red-wine white-wine
Lunch	beef ~~fish~~ red-wine ~~white-wine~~
Dinner	fish white-wine fruit-salad ~~ice-cream~~

- Preference program Φ

Φ: $\rho_1 =$ Lunch (beef) > Lunch (fish) ←,
$\rho_2 =$ Lunch (red-wine) > Lunch (white-wine) ← Lunch (beef)
$\rho_3 =$ Lunch (white-wine) > Lunch (red-wine) ← Lunch (fish)
$\rho_4 =$ Dinner (fruit-salad) > **Dinner** (ice-cream) ← Dinner (fish)
$\rho_5 =$ Dinner (ice-cream) > **Dinner** (fruit-salad) ← Dinner (beef)

Layer 0 : { ρ_4 , ρ_5 **}**

DB: { Menu (beef), Menu (fish), Menu (red-wine), Menu (white-wine) }

UNIVERSITÀ DELLA CALABRIA

Dipartimento di ELETTRONICA, INFORMATICA E SISTEMISTICA

General Preferences

- (Stratified) Datalog program P

P: Lunch (X) ← Menu (X)
Dinner (X) ← Menu (X), not Lunch (X)
Dinner (fruit-salad) ← Dinner (white-wine)
Dinner (ice-cream) ← Dinner (white-wine)
Dinner (pie) ← Dinner (red-wine).

Menu	beef	fish	red-wine	white-wine
Lunch	beef	~~fish~~	red-wine	~~white-wine~~
Dinner		fish		white-wine
		fruit-salad		~~ice-cream~~

- Preference program Φ

Φ: $\rho_1 =$ Lunch (beef) > Lunch (fish) ←,
$\rho_2 =$ Lunch (red-wine) > Lunch (white-wine) ← Lunch (beef)
$\rho_3 =$ Lunch (white-wine) > Lunch (red-wine) ← Lunch (fish)
$\rho_4 =$ Dinner (fruit-salad) > Dinner (ice-cream) ← Dinner (fish)
$\rho_5 =$ Dinner (ice-cream) > Dinner (fruit-salad) ← Dinner (beef)

The answer to the prioritized query < Dinner, P, Φ > is
{ Dinner (fish), Dinner (white-wine), Dinner (fruit-salad) }

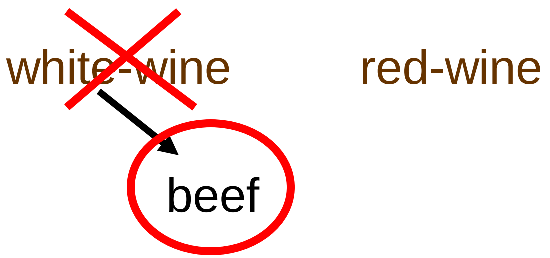
Well-Formed Queries

- A prioritized query <q, *P*, Φ> is well-formed if
  - Φ is layered, and
  - for each $A > C \leftarrow B_1, \ldots, B_m,\ not\ B_{m+1}, \ldots,\ not\ B_n$, it holds that A, B_1, B_m do not depend on C in P

DB: { white-wine, red-wine }

P: { beef ← white-wine }

Φ: { red-wine > white-wine ← beef }



Complexity Result

Let DB be a database and Q = < q, P, Φ > be a well-formed prioritized query.

The computational complexity of evaluating Q on DB is polynomial time.

Conclusions

- We have presented prioritized queries
  - preferences can be defined on both base and derived atoms
- A stratified semantics for prioritized queries has been introduced
- The computational complexity of evaluating prioritized queries is still polynomial

Thank you!

...*any questions?*

backstage

Layers

- A (ground) preference program Φ is layered if it is possible to partition Φ into n layers as follows:
  - for each atom C such that there is no rule $A > C \leftarrow B_1,\ldots,B_m,$ *not* $B_{m+1},\ldots,$ *not* $B_n,$, layer(C)=0;
  - for each atom C such that there is a rule $A > C \leftarrow B_1,\ldots,B_m,$ *not* $B_{m+1},\ldots,$ *not* $B_n,$, layer(C) > max { layer(B_1),…, layer(B_n),0 } and layer(C) ≥ layer(A);
  - the layer of a preference rule $A > C \leftarrow B_1,\ldots,B_m,$ *not* $B_{m+1},\ldots,$ *not* $B_n,$, is layer(C);
  - Φ[i] consists of all preference rules having layer i

- It is possible to define sufficient conditions which guarantee that the set of preference rules Φ can be partitioned into layers

Prioritized query

- A *prioritized query* is a triplet <q, *P*, Φ>,
  - *q* is a predicate symbol denoting the output relation,
  - *P* is a (stratified) Datalog program
  - Φ is a preference program

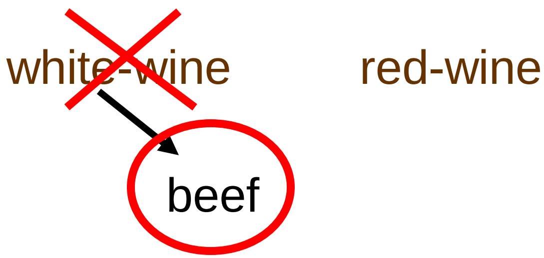
Well-Formed Queries

- A prioritized query <q, *P*, Φ> is well formed if
  - the ground transitive closure of Φ is layered, and
  - for each $A > C \leftarrow B_1,\ldots,B_m, \mathit{not}\ B_{m+1},\ldots, \mathit{not}\ B_n$, it holds that A, B_1, B_m do not depend on C in P

DB: { white-wine, red-wine }

P: { beef ← white-wine }

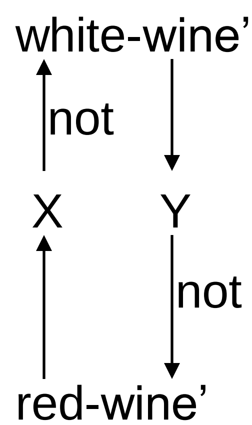
Φ: { red-wine > white-wine ← beef }



Naive Translation

Φ: { red-wine > white-wine ← beef
white-wine > red-wine ← fish }

white-wine’ ← white-wine, not X
X ← red-wine’, beef’
red-wine’ ← red-wine
beef’ ← beef
red-wine’ ← red-wine, not Y
Y ← white-wine’, fish’
white-wine’ ← white-wine
fish’ ← fish



results in a non-stratified program