Paraconsistent Relational Model: A Quasi-Classic Logic Approach
Authors: Badrinath Jayakumar* and Rajshekhar Sunderraman
Institute: Georgia State University, Georgia, USA
Corresponding author: bjayakumar2@cs.gsu.edu
Overview

- Quasi-classic logic and quasi-classic models for logic programs
- Paraconsistent relational model
- The advantages of paraconsistent relational model
- Quasi-classic models using paraconsistent relational model
- The future works and short comings
Quasi-Classic Logic

- It is a paraconsistent logic.
- Unlike Belnap’s four-valued logic [5], Hunter’s quasi-classic logic [2] supports disjunctive syllogism, disjunction introduction, etc.
- It is moves one step towards classical logic.
- Its power comes from the resolution inference rule.
Z. Zhang's quasi-classic logic programs [1] inspired from Hunter's quasi-classic logic notion and Sakma's paraconsistent minimal models notion [3].

Z. Zhang's quasi-classic logic program determines minimal quasi-classic models based on the set inclusion.

Logic rules of the form:

\[ r \ (\text{rule}) = l_0 \lor \cdots \lor l_n \leftarrow l_{n+1} \cdots l_m \]

Literals are either positive or negative atoms.
Quasi-Classic Logic Programs (2)

**Fixed Point Semantics.** Let $P$ be a positive extended disjunctive logic program and $\mathcal{I}$ be a set of interpretations, then $\mathcal{T}_P(\mathcal{I}) = \bigcup_{I \in \mathcal{I}} T_P(I)$

\[
T_P(I) = \begin{cases}
\emptyset, & \text{if } l_{n+1}, \ldots, l_m \subseteq I \text{ for some ground constraint } \leftarrow l_{n+1} \ldots l_m \text{ from } P. \\
\{J \mid \text{for each ground rule } r_i : l_0 \vee \cdots \vee l_n \leftarrow l_{n+1} \ldots l_m \text{ such that } \\
\{l_{n+1} \ldots l_m\} \subseteq I, J = I \cup \bigcup_{r_i} J' \text{ where } \\
J' \in \text{Lits}(\text{focus}(l_0 \vee \cdots \vee l_n, I))\}, & \text{otherwise.}
\end{cases}
\]

- $T_P(I)$ always terminates in finite time.

Let $\alpha \vee \beta \vee \gamma$ be a clause. Then $\text{focus}(\alpha \vee \beta \vee \gamma, \alpha) = \beta \vee \gamma$
Paraconsistent Relational Model (1)

- The normal relation stores only information that is believed to be true.
- The paraconsistent relation stores information that is believed to be true and believed to be false.
- We define two types of algebraic operators:
  - Set Theoretic: union ($\cup$), complement ($\neg$ (unary)), intersection ($\cap$), and difference ($\setminus$ (binary)).
  - Relation Theoretic: Join ($\Join$), selection ($\sigma$), and projection ($\pi$).
Paraconsistent Relational Model (2)

Normal Relation (Closed World Assumption):

<table>
<thead>
<tr>
<th>set of attributes ($\Sigma$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tuple ($\tau(\Sigma)$)</td>
</tr>
<tr>
<td>tuple ($\tau(\Sigma)$)</td>
</tr>
</tbody>
</table>

Paraconsistent Relation (Open World Assumption):

<table>
<thead>
<tr>
<th>set of attributes ($\Sigma$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tuple ($\tau(\Sigma)$)</td>
</tr>
<tr>
<td>tuple ($\tau(\Sigma)$)</td>
</tr>
<tr>
<td>tuple ($\tau(\Sigma)$)</td>
</tr>
<tr>
<td>tuple ($\tau(\Sigma)$)</td>
</tr>
</tbody>
</table>
Paraconsistent Relational Model (3)

**Union.** The union of $R$ and $S$, denoted $R \cup S$, is a paraconsistent relation on scheme $\Sigma$, given that

$$(R \cup S)^+ = R^+ \cup S^+ , (R \cup S)^- = R^- \cap S^-$$

**Complement.** The complement of $R$, denoted $\dot{R}$, is a paraconsistent relation on scheme $\Sigma$, given that

$$\dot{R}^+ = R^- , \dot{R}^- = R^+$$
Paraconsistent Relational Model (4)

**Intersection.** The intersection of $R$ and $S$, denoted $R \cap S$, is a paraconsistent relation on scheme $\Sigma$, given that

$$(R \cap S)^+ = R^+ \cap S^+, \quad (R \cap S)^- = R^- \cup S^-$$

**Difference.** The difference of $R$ and $S$, denoted $R - S$, is a paraconsistent relation on scheme $\Sigma$, given that

$$(R - S)^+ = R^+ \cap S^-, \quad (R - S)^- = R^- \cup S^+$$
Paraconsistent Relational Model

Example (5)

Example 1. Let \( R = \{X\} \)
\[
\begin{array}{c}
(a) \\
(b) \\
(c)
\end{array}
\]

and \( S = \{X\} \)
\[
\begin{array}{c}
(c) \\
(b) \\
(a)
\end{array}
\].

Then
\[
R \cup S = \begin{array}{c}
\{X\} \\
(a) \\
(b) \\
(b) \\
\end{array}
\]

(union)

and
\[
R \cap S = \begin{array}{c}
\end{array}
\]

(intersection)
Paraconsistent Relational Model

Example (6)

\[ R \hat{\cap} S = \begin{array}{c}
(X) \\
(b) \\
(a) \\
(c)
\end{array} \quad R \hat{\setminus} S = \begin{array}{c}
(X) \\
(a) \\
(b) \\
(c)
\end{array} \quad \hat{\cdot} R = \begin{array}{c}
(X) \\
(c) \\
(a) \\
(b)
\end{array} \]
**Join.** The join of $R$ and $S$, denoted $R \Join S$, is a paraconsistent relation on scheme $\Sigma \cup \Delta$ given that

$$(R \Join S)^+ = R^+ \Join S^+, \ (R \Join S)^- = (R^-)^{\Sigma \cup \Delta} \cup (S^-)^{\Sigma \cup \Delta}$$

**Projection.** The projection of $R$ onto $\Delta$, denoted $\hat{\pi}_\Delta(R)$, is a paraconsistent relation on $\Delta$ given that

$$\hat{\pi}_\Delta(R)^+ = \pi_\Delta((R^+)^{\Sigma \cup \Delta})$$

$$\hat{\pi}_\Delta(R)^- = \{t \in \tau(\Delta) \mid t^{\Sigma \cup \Delta} \subseteq (R^-)^{\Sigma \cup \Delta}\}$$

where $\pi_\Delta$ is the usual projection over $\Delta$ of ordinary relations.
Paraconsistent Relational Model (8)

**Selection.** Let $F$ be any logic formula involving attribute names in $\Sigma$, constant symbols, and any of these symbols {$=, \neg, \land, \lor$}. Then the selection of $R$ by $F$, denoted $\dot{\sigma}_F(R)$, is a paraconsistent relation on scheme $\Sigma$, given that

$$\dot{\sigma}_F(R)^+ = \sigma_F(R)^+, \dot{\sigma}_F(R)^- = R^- \cup \sigma_{\neg F}(\tau(\Sigma))$$

where $\sigma_F$ is a usual selection of tuples satisfying $F$ from ordinary relations.
Example 2. Let $R = \begin{align*}
  \langle X, Y \rangle \\
  (b, b) \\
  (b, c) \\
  (a, a) \\
  (a, b) \\
  (a, c)
\end{align*}$ and $S = \begin{align*}
  \langle Y, Z \rangle \\
  (a, c) \\
  (c, a) \\
  (c, b)
\end{align*}$. Here attributes are ordered sequence and tuples are lists of values.
Paraconsistent Relational Model
Example (10)

\[
R \bowtie S = \\
\begin{array}{c}
\langle X, Y, Z \rangle \\
(b, c, a) \\
(a, a, a) \\
(a, a, b) \\
(a, a, c) \\
(a, b, a) \\
(a, b, b) \\
(a, b, c) \\
(a, c, a) \\
(a, c, b) \\
(a, c, c) \\
(b, c, b) \\
(c, c, b)
\end{array}
\]

\[
\pi_{\langle X, Z \rangle}(R \bowtie S) = \\
\begin{array}{c}
\langle X, Y \rangle \\
(b, a) \\
(a, a) \\
(a, b) \\
(a, c)
\end{array}
\]

\[
\sigma_{x = z}(\pi_{\langle X, Z \rangle}(R \bowtie S)) = \\
\begin{array}{c}
\langle X, Y \rangle \\
(b, a) \\
(a, a) \\
(a, b) \\
(a, c) \\
(b, b) \\
(c, c)
\end{array}
\]
Advantages of Using Paraconsistent Relational Model

- Three main advantages:
  - works with a set of tuples instead of a tuple at a time,
  - can apply various laws of equality,
  - suits good for query intensive applications.
Quasi-Classic Models Construction (1)

- Here, we consider positive extended disjunctive deductive databases.
- The model construction involves two steps:
  - associate every literal to a paraconsistent relation and construct an equation for every clause;
  - solve the equations.
It is hard to represent disjunctive information in paraconsistent relation.
We introduce disjunctive paraconsistent relation.

<table>
<thead>
<tr>
<th>Paraconsistent Relation</th>
<th>Disjunctive Paraconsistent Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ set of attributes (\Sigma) }</td>
<td>{ set of attributes sets (2^{2^\Sigma}) }</td>
</tr>
<tr>
<td>tuple (\tau(\Sigma))</td>
<td>disjunctive tuple (\tau(2^{2^\Sigma}))</td>
</tr>
<tr>
<td>tuple (\tau(\Sigma))</td>
<td>disjunctive tuple (\tau(2^{2^\Sigma}))</td>
</tr>
<tr>
<td>tuple (\tau(\Sigma))</td>
<td>conjunctive tuple (\tau(2^{2^\Sigma}))</td>
</tr>
<tr>
<td>tuple (\tau(\Sigma))</td>
<td>conjunctive tuple (\tau(2^{2^\Sigma}))</td>
</tr>
</tbody>
</table>

We allow sometimes the conjunctive tuple in the positive part.
Quasi-Classic Models Construction
Example (3)

Let $P$ be a positive extended disjunctive deductive database. It has the following facts and rules:

\[
\begin{align*}
r(a, c), p(a), p(c), &\, \neg f(a, b), s(c) \\
w(X) \lor g(X) \lor \neg p(X) &\iff r(X, Y), s(Y) \\
w(X) \lor g(X) \lor \neg p(X) &\iff \neg f(X, Y)
\end{align*}
\]

Converting the rules into equations:

1. \( (\bar{\pi}_{\{X\}}(w(X) \cup g(X) \cup \neg p(X))[X] = (\bar{\pi}_{\{X\}}(r(X, Y) \bowtie s(Y)))[X] \)

2. \( (\bar{\pi}_{\{X\}}(w(X) \cup g(X) \cup \neg p(X))[X] = (\bar{\pi}_{\{X\}}(\neg f(X, Y)))[X] \)

LHS in both equations are the same. So,

\[
(\bar{\pi}_{\{X\}}(w(X) \cup g(X) \cup \neg p(X))[X] = (\bar{\pi}_{\{X\}}(r(X, Y) \bowtie s(Y)))[X] \cup (\bar{\pi}_{\{X\}}(\neg f(X, Y)))[X]
\]
First, facts are added to the paraconsistent relation.

\[ \text{SModel} = \{r, p, s, f\} \text{ where} \]

\[ r = \frac{\{X, Y\}}{(a, c)} \quad p = \frac{\{X\}}{(a)} \quad s = \frac{\{Y\}}{(c)} \quad f = \frac{\{X, Y\}}{(a, b)} \]

Copies are created. Copies are the same, but have different relation name.

\[ \text{SModel} = \{r', p', s', f'\} \text{ (COPIES) where} \]

\[ r' = \frac{\{X, Y\}}{(a, c)} \quad p' = \frac{\{X\}}{(a)} \quad s' = \frac{\{Y\}}{(c)} \]

\[ f' = \frac{\{X, Y\}}{(a, b)} \]
Quasi-Classic Models Construction Example (5)

- Mapping both definite tuples and disjunctive tuples from LHS of the equation to the disjunctive paraconsistent relation.

$$DR_1 = \begin{array}{ccc}
\{w.X\} & \{g.X\} & \{p.X\} \\
(a) & (a) & (a) \\
\hline
& (a) & \\
& (c) & 
\end{array}$$

- We renamed the attribute before we map.
- Inconsistency is in the disjunctive relation.
Quasi-Classic Models Construction Example (6)

- Applying focus, which removes complementary tuples from the disjunctive relation with respect to SModel.

\[
DR_1 = FOCUS_D(DR_1, SModel)
\]

<table>
<thead>
<tr>
<th>{w.X}</th>
<th>{g.X}</th>
<th>{p.X}</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>(a)</td>
<td></td>
</tr>
</tbody>
</table>

\[
DR_1 = (a)
\]

(c)
The disjunctive paraconsistent relation contains disjunctive information which leads to more relations called proper disjunctive paraconsistent relations. Therefore, 

\[ PDR_1 = \{ PDR_1^1, PDR_1^2, PDR_1^3 \} \]

\[ PDR_1^1 = \begin{array}{c|c|c}
\{w.X\} & \{g.X\} & \{p.X\} \\
(a) & (a) & (c) \\
\hline
\{w.X\} & \{g.X\} & \{p.X\} \\
(a) & (c) & (c) \\
\end{array} \]

\[ PDR_1^2 = \begin{array}{c|c|c}
\{w.X\} & \{g.X\} & \{p.X\} \\
(a) & (a) & (c) \\
\hline
\end{array} \]

\[ PDR_1^3 = \begin{array}{c|c|c}
\{w.X\} & \{g.X\} & \{p.X\} \\
(a) & (a) & (a) \\
\hline
\end{array} \]
Quasi-Classic Models Construction
Example (8)

Map every p in $PDR_1$ back to a set of base relations (replicas). After relationalizing the set of relations,

$C_1 = \{ \{ w, p \}_1^1, \{ g, p \}_2^2, \{ w, g, p \}_3^3 \}$

then create an exact relation in $DModel$ for every relation in $SModel$.

$w = \begin{array}{c} \{X\} \\ (a) \end{array}$  $p = \begin{array}{c} \{X\} \\ (a) \\ (c) \end{array}$  $\{ w, g, p \}_3^3$

$g = \begin{array}{c} \{X\} \\ (a) \end{array}$  $p = \begin{array}{c} \{X\} \\ (a) \\ (c) \end{array}$

Relationalizing: removing paraconsistent unions among paraconsistent relations. Then, create an exact relation in $DModel$ for every relation in $SModel$. 

$DModel = DModel \cup C_1$
Quasi-Classic Models Construction
Example (9)

\[
\text{DModel} = \{\{w, p, r, s, f\}^1, \{g, p, r, s, f\}^2, \{w, g, p, r, s, f\}^3\}
\]

\[
w = \begin{array}{c}
\{X\} \\
(a)
\end{array} \quad p = \begin{array}{c}
\{X\} \\
(a) \\
(c)
\end{array} \quad r = \begin{array}{c}
\{X, Y\} \\
(a, c)
\end{array} \quad s = \begin{array}{c}
\{Y\} \\
(c)
\end{array} \quad f = \begin{array}{c}
\{X, Y\} \\
(a, b)
\end{array}
\]

\[
g = \begin{array}{c}
\{X\} \\
(a)
\end{array} \quad p = \begin{array}{c}
\{X\} \\
(a) \\
(c)
\end{array} \quad r = \begin{array}{c}
\{X, Y\} \\
(a, c)
\end{array} \quad s = \begin{array}{c}
\{Y\} \\
(c)
\end{array} \quad f = \begin{array}{c}
\{X, Y\} \\
(a, b)
\end{array}
\]

\[
w = \begin{array}{c}
\{X\} \\
(a)
\end{array} \quad g = \begin{array}{c}
\{X\} \\
(a)
\end{array} \quad p = \begin{array}{c}
\{X\} \\
(a) \\
(c)
\end{array} \quad r = \begin{array}{c}
\{X, Y\} \\
(a, c)
\end{array} \quad s = \begin{array}{c}
\{Y\} \\
(c)
\end{array} \quad f = \begin{array}{c}
\{X, Y\} \\
(a, b)
\end{array}
\]

\[
\begin{array}{c}
\{X, Y\} \\
(a, b)
\end{array}
\]

Add DModel to TempModel.
Quasi-Classic Models Construction
Example (10)

SModel = Minimize (TempModel). Repeat from step 2 of Slide 20.

The algorithm stops when there is no change in SModel. We then skip further iterations and write the final result:

Minimal QC Model = \{ \{ w, p, r, s, f \}^1, \{ g, p, r, s, f \}^2 \}

\{ w, p, r, s, f \}^1

\[
\begin{align*}
w &= \frac{\{ X \}}{(a)} \quad p &= \frac{\{ X \}}{(c)} \\
\{ g, p, r, s, f \}^2
\end{align*}
\]

\[
\begin{align*}
g &= \frac{\{ X \}}{(a)} \quad p &= \frac{\{ X \}}{(c)} \\
\end{align*}
\]

Minimize removes redundant sets.
Quasi-Classic Models Construction
Example (11)

Minimal model by size implies minimal model by set inclusion (vice versa is not true).

In other words, Minimal QC Model = \{\{w(a), p(a), p(c), r(a, c), s(c), \neg f(a, b)\}, \\
\{g(a), p(a), p(c), r(a, c), s(c), \neg f(a, b)\}\}
Future Works/Short Comings

- The algorithm does not work in the presence of disjunctive facts, and constants and duplicate variables in disjunctive literals.
- The algorithm finds only strong models and no constraints/recursions are allowed.
- The algorithm could be extended to allow default negation.
- The algorithm lacks the proof of correctness and complexity.
Thank You
Bibliography


