4QL: Living with Inconsistency and Taming Nonmonotonicity

Jan Małuszyński and Andrzej Szałas

Oxford, 2010
The structure of talk

- Introduction and motivations.
  - Living with inconsistency.
    - Four-valued reasoning with $t, f, u$ and $i$.
    - Monotonic, intuitive and tractable rule language with unrestricted negation.
  - Taming Nonmonotonicity.
    - Layered architecture.
    - Local Closed-World Assumption.
    - Lightweight nonmonotonic reasoning.
- Conclusions.
The structure of talk

- Introduction and motivations.
- Living with inconsistency.
  - Four-valued reasoning with \( t, f, u \) and \( i \).
  - Monotonic, intuitive and tractable rule language with unrestricted negation.
- Taming Nonmonotonicity.
  - Layered architecture.
  - Local Closed-World Assumption.
  - Lightweight nonmonotonic reasoning.
- Conclusions.
The structure of talk

• Introduction and motivations.
• Living with inconsistency.
  • Four-valued reasoning with t, f, u and i.
  • Monotonic, intuitive and tractable rule language with unrestricted negation.
• Taming Nonmonotonicity.
  • Layered architecture.
  • Local Closed-World Assumption.
  • Lightweight nonmonotonic reasoning.
• Conclusions.
Closed-World Assumption?

Why CWA?

- Efficient representation of negative information.
- Natural and intuitive in many application areas.

Why not CWA?

- Non-monotonicity not controlled by users.
- Not suitable for important areas including robotics, Semantic Web, multiagent systems.
Closed-World Assumption?

Why CWA?
- Efficient representation of negative information.
- Natural and intuitive in many application areas.

Why not CWA?
- Non-monotonicity not controlled by users.
- Not suitable for important areas including robotics, Semantic Web, multiagent systems.
Example

An autonomous vehicle approaches an intersection where there is no stop sign, yield sign or traffic signal. It should yield to vehicles coming from the right:

\[
\text{halt}(X) \leftarrow \text{right}(X, Y). \quad \text{(Halt at intersection } X \text{ when there is car } Y \text{ to the right.)}
\]

If \( \text{right}(X, Y) = u \) then, under \( \text{CWA} \), \( \text{halt}(X) = f \).
Example

An autonomous vehicle approaches an intersection where there is no stop sign, yield sign or traffic signal. It should yield to vehicles coming from the right:

\[ \text{halt}(X) \leftarrow \text{right}(X, Y). \] (Halt at intersection \( X \) when there is car \( Y \) to the right.)

If \( \text{right}(X, Y) = u \) then, under CWA, \( \text{halt}(X) = f \).
Example

A web agent asks a Semantic Web service whether $X$ is a reliable seller. What should be the answer when:

- the service has no information concerning the reliability of $X$?
- the service has inconsistent information about $X$?

Remark

Such situations are typical with many information sources. The semantics can be encoded using two truth values. However, $u$ and $i$ remain more or less implicit there.
Example

A web agent asks a Semantic Web service whether \( X \) is a reliable seller. What should be the answer when:

- the service has no information concerning the reliability of \( X \)?
- the service has inconsistent information about \( X \)?

Remark

Such situations are typical with many information sources. The semantics can be encoded using two truth values. However, \( u \) and \( i \) remain more or less implicit there.
The classical two-valued setting

Aristotle (384 BC – 322 BC)

The twin foundations of Aristotle’s logic are the law of non-contradiction (LNC) and the law of excluded middle (LEM):

- LNC: \( \neg (A \land \neg A) \)
- LEM: \( A \lor \neg A \).

Substantial problem

In the classical logic inconsistency \( (A \land \neg A) \) is false, so inconsistency trivializes the theory and reasoning.
The classical two-valued setting

Aristotle (384 BC – 322 BC)

The twin foundations of Aristotle’s logic are the law of non-contradiction (LNC) and the law of excluded middle (LEM):

- LNC: \( \neg(A \land \neg A) \)
- LEM: \( A \lor \neg A \).

Substantial problem

In the classical logic inconsistency \( A \land \neg A \) is false, so inconsistency trivializes the theory and reasoning.
Aristotle (384 BC – 322 BC)

The twin foundations of Aristotle’s logic are the law of non-contradiction (LNC) and the law of excluded middle (LEM):

- **LNC**: \( \neg (A \land \neg A) \)
- **LEM**: \( A \lor \neg A \).

Substantial problem

In the classical logic inconsistency \( A \land \neg A \) is false, so inconsistency trivializes the theory and reasoning.
Aristotle (384 BC – 322 BC)

The twin foundations of Aristotle’s logic are the law of non-contradiction (LNC) and the law of excluded middle (LEM):

- LNC: \( \neg(A \land \neg A) \)
- LEM: \( A \lor \neg A \).

Substantial problem

In the classical logic inconsistency \( (A \land \neg A) \) is false, so inconsistency trivializes the theory and reasoning.
Resoning by cases (Greeks, even before Aristotle)

Lemma, Dilemma, Trilemma, Teralemma, ...

Reflect reasoning by cases, where one assumes that $C_1 \lor \ldots \lor C_n$ holds. Then, whenever:

$$C_1 \rightarrow A$$
$$\ldots$$
$$C_n \rightarrow A$$

one concludes $A$. 

Resoning by cases (Greeks, even before Aristotle)

Lemma, Dilemma, Trilemma, Teralemma, ...
Reflect reasoning by cases, where one assumes that $C_1 \lor \ldots \lor C_n$ holds. Then, whenever:

\[ C_1 \rightarrow A \]
\[ \ldots \]
\[ C_n \rightarrow A \]

one concludes $A$. 
Resoning by cases (Greeks, even before Aristotle)

Lemma, Dilemma, Trilemma, Teralemma, ...

Reflect reasoning by cases, where one assumes that $C_1 \lor \ldots \lor C_n$ holds. Then, whenever:

\[
C_1 \rightarrow A \\
\ldots \\
C_n \rightarrow A
\]

one concludes $A$. 

Jan Małuszyński and Andrzej Szałas
Example

Buddha (Gotama, c. 563 BCE to 483 BCE) has been asked the following questions:

- Does Gotama believe that the saint exists after death?
- Does Gotama believe that the saint does not exist after death?
- Does Gotama believe that the saint both exists and does not exist after death?
- Does Gotama believe that the saint neither exists nor does not exist after death?
Example

Buddha (Gotama, c. 563 BCE to 483 BCE) has been asked the following questions:

- Does Gotama believe that the saint exists after death?
- Does Gotama believe that the saint does not exist after death?
- Does Gotama believe that the saint both exists and does not exist after death?
- Does Gotama believe that the saint neither exists nor does not exist after death?
Example

Buddha (Gotama, c. 563 BCE to 483 BCE) has been asked the following questions:

• Does Gotama believe that the saint exists after death?
• Does Gotama believe that the saint does not exist after death?
• Does Gotama believe that the saint both exists and does not exist after death?
• Does Gotama believe that the saint neither exists nor does not exist after death?
Example

Buddha (Gotama, c. 563 BCE to 483 BCE) has been asked the following questions:

- Does Gotama believe that the saint exists after death?
- Does Gotama believe that the saint does not exist after death?
- Does Gotama believe that the saint both exists and does not exist after death?
- Does Gotama believe that the saint neither exists nor does not exist after death?
Example

Buddha (Gotama, c. 563 BCE to 483 BCE) has been asked the following questions:

- Does Gotama believe that the saint exists after death?
- Does Gotama believe that the saint does not exist after death?
- Does Gotama believe that the saint both exists and does not exist after death?
- Does Gotama believe that the saint neither exists nor does not exist after death?
Nāgārjuna’s catuṣkoṭi (‘four corners’, ca. 150–250 CE)

Buddhist logicians in their reasoning considered the following four cases:

- $A$
- $\neg A$
- both $A$ and $\neg A$
- neither $A$ nor $\neg A$. 
Nāgārjuna’s catuṣkoṭi ("four corners", ca. 150–250 CE)

Buddhist logicians in their reasoning considered the following four cases:

- $A$
- $\neg A$
- both $A$ and $\neg A$
- neither $A$ nor $\neg A$. 
Nāgārjuna’s catuṣkoṭi ("four corners", ca. 150–250 CE)

Buddhist logicians in their reasoning considered the following four cases:

- $A$
- $\neg A$
- both $A$ and $\neg A$
- neither $A$ nor $\neg A$. 
Nāgārjuna’s catuṣkoṭi ("four corners", ca. 150–250 CE)

Buddhist logicians in their reasoning considered the following four cases:

- $A$
- $\neg A$
- both $A$ and $\neg A$
- neither $A$ nor $\neg A$. 
Nāgārjuna’s catuṣkoṭi ("four corners", ca. 150–250 CE)

Buddhist logicians in their reasoning considered the following four cases:

- $A$
- $\neg A$
- both $A$ and $\neg A$
- neither $A$ nor $\neg A$. 
Examples

• Physics: wave–particle duality postulates that all matter exhibits both wave and particle properties, so it both consists of particles and, at the same time, does not.

• Theory of knowledge: in our current knowledge we have neither $P = NP$ nor $P \neq NP$.

• Three-valued logic (in modern logic, initiated by Łukasiewicz in 1920) rejects this law (there might be the third value).
Examples

• Physics: wave–particle duality postulates that all matter exhibits both wave and particle properties, so it both consists of particles and, at the same time, does not.

• Theory of knowledge: in our current knowledge we have neither $P = NP$ nor $P \neq NP$.

• Three-valued logic (in modern logic, initiated by Łukasiewicz in 1920) rejects this law (there might be the third value).
Examples

- Physics: wave–particle duality postulates that all matter exhibits both wave and particle properties, so it both consists of particles and, at the same time, does not.
- Theory of knowledge: in our current knowledge we have neither $P = NP$ nor $P \neq NP$.
- Three-valued logic (in modern logic, initiated by Łukasiewicz in 1920) rejects this law (there might be the third value).
Many information sources

Knowledge fusion

We have the following four cases:

- true, e.g., when some sources claim that $A$ holds and no source claims the contrary
- false, e.g., when some sources claim that $\neg A$ does not hold and no source claims the contrary
- inconsistent, e.g., when some sources claim that $A$ holds and some sources claim that $\neg A$ does not hold (and perhaps some sources claim that $A$ is unknown)
- unknown, when no source claims that $A$ and no source claims that $\neg A$. 
Knowledge fusion

We have the following four cases:

- true, e.g., when some sources claim that $A$ holds and no source claims the contrary
- false, e.g., when some sources claim that $\neg A$ does not hold and no source claims the contrary
- inconsistent, e.g., when some sources claim that $A$ holds and some sources claim that $\neg A$ does not hold (and perhaps some sources claim that $A$ is unknown)
- unknown, when no source claims that $A$ and no source claims that $\neg A$. 
Knowledge fusion

We have the following four cases:

- true, e.g., when some sources claim that $A$ holds and no source claims the contrary
- false, e.g., when some sources claim that $\neg A$ does not hold and no source claims the contrary
- inconsistent, e.g., when some sources claim that $A$ holds and some sources claim that $\neg A$ does not hold (and perhaps some sources claim that $A$ is unknown)
- unknown, when no source claims that $A$ and no source claims that $\neg A$. 
Many information sources

Knowledge fusion

We have the following four cases:

• true, e.g., when some sources claim that $A$ holds and no source claims the contrary

• false, e.g., when some sources claim that $\neg A$ does not hold and no source claims the contrary

• inconsistent, e.g., when some sources claim that $A$ holds and some sources claim that $\neg A$ does not hold (and perhaps some sources claim that $A$ is unknown)

• unknown, when no source claims that $A$ and no source claims that $\neg A$. 
Many information sources

Knowledge fusion

We have the following four cases:

- true, e.g., when some sources claim that $A$ holds and no source claims the contrary
- false, e.g., when some sources claim that $\neg A$ does not hold and no source claims the contrary
- inconsistent, e.g., when some sources claim that $A$ holds and some sources claim that $\neg A$ does not hold (and perhaps some sources claim that $A$ is unknown)
- unknown, when no source claims that $A$ and no source claims that $\neg A$. 
Orderings of truth values

Belnap’s truth ordering

truth ordering ($\leq_t$)

truth ordering ($\leq'_t$)

Conjunction and disjunction

The semantics of:

\[
\begin{align*}
A \land B &= \min\{A, B\} \\
A \lor B &= \max\{A, B\}
\end{align*}
\]
(w.r.t. truth ordering).

E.g., $t \land i = i$, $u \lor f = u$, etc.
Orderings of truth values

Belnap's truth ordering

Conjunction and disjunction

The semantics of:

\[ A \land B = \min\{A, B\} \]
\[ A \lor B = \max\{A, B\} \]

(w.r.t. truth ordering).

E.g., \( t \land i = i \), \( u \lor f = u \), etc.
Four-valued logic

Orderings of truth values

Belnap's truth ordering

truth ordering ($\leq_t$)

truth ordering ($\leq'_t$)

Conjunction and disjunction

The semantics of:

$$A \land B = \min\{A, B\}$$
$$A \lor B = \max\{A, B\}$$

(w.r.t. truth ordering).

E.g., $t \land i = i$, $u \lor f = u$, etc.
Four-valued logic

Orderings of truth values

Belnap’s truth ordering \( (\leq_t) \)

Truth ordering \( (\leq) \)

Truth ordering \( (\leq'_t) \)

Conjunction and disjunction

The semantics of:

\[
A \land B = \min\{A, B\} \\
A \lor B = \max\{A, B\}
\]

(w.r.t. truth ordering).

E.g., \( t \land i = i \), \( u \lor f = u \), etc.
Four-valued logic

Orderings of truth values

Belnap’s truth ordering

\[ t \leq i \leq u \leq f \]

truth ordering (\( \leq_t \))

truth ordering (\( \leq_{t'} \))

Conjunction and disjunction

The semantics of:

\[
\begin{align*}
A \land B &= \min\{A, B\} \\
A \lor B &= \max\{A, B\}
\end{align*}
\]

(w.r.t. truth ordering).

E.g., \( t \land i = i \), \( u \lor f = u \), etc.
Four-valued logic

Orderings of truth values

knowledge ordering

ordering ($\leq_k$)

ordering ($\leq'_k$)
Orderings of truth values

knowledge ordering

ordering ($\leq_k$)
Orderings of truth values

knowledge ordering  ordering ($\leq_k$)  ordering ($\leq'_k$)
Orderings of truth values

knowledge ordering
ordering \((\leq_k)\)
ordering \((\leq'_k)\)
Orderings of truth values

knowledge ordering

ordering \((\leq_k)\)

ordering \((\leq'_k)\)
Semantics of negation and implication

<table>
<thead>
<tr>
<th></th>
<th>f</th>
<th>u</th>
<th>i</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>t</td>
<td>t</td>
<td>t</td>
<td>t</td>
</tr>
<tr>
<td>u</td>
<td>t</td>
<td>t</td>
<td>t</td>
<td>t</td>
</tr>
<tr>
<td>i</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>t</td>
<td>t</td>
<td>t</td>
<td>t</td>
<td>t</td>
</tr>
</tbody>
</table>

|   |   |   |   |   |
|---|---|---|---|
| ¬ | f | t | u | u |
|   | i | i |   | t |
|   | f | f |   | f |

Remark

Implication $B \rightarrow C$ is $f$ only when the conclusion $C$ has to be corrected to satisfy the corresponding rule $C :\neg B$. 
### Semantics of negation and implication

<table>
<thead>
<tr>
<th>$\rightarrow$</th>
<th>$\neg$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f$</td>
<td>$t$</td>
</tr>
<tr>
<td>$u$</td>
<td>$u$</td>
</tr>
<tr>
<td>$i$</td>
<td>$i$</td>
</tr>
<tr>
<td>$t$</td>
<td>$f$</td>
</tr>
</tbody>
</table>

Remark

Implication $B \rightarrow C$ is $f$ only when the conclusion $C$ has to be corrected to satisfy the corresponding rule $C :\leftarrow B$. 

Jan Małuszyński and Andrzej Szalas
Datalog 2.0, Oxford, UK
Let:

- $\lor_k$ be the disjunction w.r.t. $\leq_k$,
- $\lor, \land$ be the disjunction and conjunction w.r.t. $\leq_t$,
- $\lor', \land'$ be the disjunction and conjunction w.r.t. $\leq'_t$.

Then:

\[
\begin{align*}
(B \rightarrow A) \land (C \rightarrow A) &= (B \lor_k C) \rightarrow A \\
\neg (A \lor B) &= \neg A \land' \neg B \\
\neg (A \land B) &= \neg A \lor' \neg B.
\end{align*}
\]
Let:

- $\lor_k$ be the disjunction w.r.t. $\leq_k$,
- $\lor, \land$ be the disjunction and conjunction w.r.t. $\leq_t$,
- $\lor', \land'$ be the disjunction and conjunction w.r.t. $\leq'_t$.

Then:

\[
\begin{align*}
&(B \rightarrow A) \land (C \rightarrow A) = [(B \lor_k C) \rightarrow A] \\
&\neg(A \lor B) = [\neg A \land' \neg B] \\
&\neg(A \land B) = [\neg A \lor' \neg B].
\end{align*}
\]
We proposed → in our previous work with A. Vitória. It reflects the following principles:

- new facts are not deduced from premises evaluating to \( f \) or \( u \)
- a fact can be assigned \( t \) only on the basis of premises evaluating to \( t \)
- true premises are allowed to imply inconsistency of a fact, since another rule can support the negation of this fact.
Discussion

We proposed → in our previous work with A. Vitória. It reflects the following principles:

- new facts are not deduced from premises evaluating to f or u
- a fact can be assigned t only on the basis of premises evaluating to t
- true premises are allowed to imply inconsistency of a fact, since another rule can support the negation of this fact.
We proposed $\rightarrow$ in our previous work with A. Vitória. It reflects the following principles:

- new facts are not deduced from premises evaluating to $\text{f}$ or $\text{u}$
- a fact can be assigned $\text{t}$ only on the basis of premises evaluating to $\text{t}$
- true premises are allowed to imply inconsistency of a fact, since another rule can support the negation of this fact.
Discussion

We proposed → in our previous work with A. Vitória. It reflects the following principles:

- new facts are not deduced from premises evaluating to f or u
- a fact can be assigned t only on the basis of premises evaluating to t
- true premises are allowed to imply inconsistency of a fact, since another rule can support the negation of this fact.
Discussion continued

- Deduction from unknown leads to nonmonotonicity. It will later be allowed in a well controlled manner.
- Deduction from false is questionable. For example:

  \[ \text{late} :\leftarrow \text{overslept}. \]

  If deductions from false premises are allowed, then the falsity of \text{overslept} makes \text{late} false which is an incorrect conclusion both intuitively and in logic.

  (\textsc{Datalog} provides the same result due to CWA.)

  In our semantics \text{late} remains unknown still satisfying the rule.
Discussion continued

- Deduction from unknown leads to nonmonotonicity. It will later be allowed in a well controlled manner.
- Deduction from false is questionable. For example:

\[ \text{late} \leftarrow \text{overslept}. \]

If deductions from false premises are allowed, then the falsity of \textit{overslept} makes \textit{late} false which is an incorrect conclusion both intuitively and in logic.

(Datalog provides the same result due to CWA.)

In our semantics \textit{late} remains unknown still satisfying the rule.
Interpretations

Definition

By an *interpretation* we mean any set of literals. *Truth value* of a literal \( \ell \) in interpretation \( \mathcal{I} \):

\[
\mathcal{I}(\ell) \overset{\text{def}}{=} \begin{cases} 
\top & \text{if } \ell \in \mathcal{I} \text{ and } \neg \ell \notin \mathcal{I} \\
\bot & \text{if } \ell \in \mathcal{I} \text{ and } \neg \ell \in \mathcal{I} \\
\perp & \text{if } \ell \notin \mathcal{I} \text{ and } \neg \ell \notin \mathcal{I} \\
\perp & \text{if } \ell \notin \mathcal{I} \text{ and } \neg \ell \in \mathcal{I}.
\end{cases}
\]

Extending the definition for all formulas

The *truth value* of a formula in interpretation \( \mathcal{I} \) is defined as usual, using truth tables provided for \( \neg, \land, \lor, \rightarrow \).
Interpretations

Definition

By an interpretation we mean any set of literals. Truth value of a literal \( \ell \) in interpretation \( \mathcal{I} \):

\[
\mathcal{I}(\ell) \overset{\text{def}}{=} \begin{cases} 
  t & \text{if } \ell \in \mathcal{I} \text{ and } (\neg \ell) \notin \mathcal{I} \\
  i & \text{if } \ell \in \mathcal{I} \text{ and } (\neg \ell) \in \mathcal{I} \\
  u & \text{if } \ell \notin \mathcal{I} \text{ and } (\neg \ell) \notin \mathcal{I} \\
  f & \text{if } \ell \notin \mathcal{I} \text{ and } (\neg \ell) \in \mathcal{I} 
\end{cases}
\]

Extending the definition for all formulas

The truth value of a formula in interpretation \( \mathcal{I} \) is defined as usual, using truth tables provided for \( \neg, \wedge, \vee, \rightarrow \).
Syntax of rules

In the sequel we consider ground rules only and assume that for each head $\ell$ there is only one rule of the form:

$$
\ell \colon= \left( b_{11}, \ldots, b_{1i_1} \right) \lor \\
\left( b_{21}, \ldots, b_{2i_2} \right) \lor \\
\ldots \lor \\
\left( b_{m1}, \ldots, b_{mi_m} \right).
$$

Disjunction in (1) gathers all ground bodies with $\ell$ as the head.

Four-valued semantics of rules

A set of literals $\mathcal{I}$ is a model of a set of rules $S$ iff for each rule $\varrho \in S$ we have that $\mathcal{I}(\text{body}(\varrho) \rightarrow \text{head}(\varrho)) = t$, assuming that the empty body takes the value $t$ in any interpretation.
Syntax of rules

In the sequel we consider ground rules only and assume that for each head $\ell$ there is only one rule of the form:

$$\ell : \neg (b_{11}, \ldots, b_{1i_1}) \lor (b_{21}, \ldots, b_{2i_2}) \lor \ldots \lor (b_{m1}, \ldots, b_{mi_m}).$$

Disjunction in (1) gathers all ground bodies with $\ell$ as the head.

Four-valued semantics of rules

A set of literals $\mathcal{I}$ is a model of a set of rules $S$ iff for each rule $\varrho \in S$ we have that $\mathcal{I}(\text{body}(\varrho) \rightarrow \text{head}(\varrho)) = t$, assuming that the empty body takes the value $t$ in any interpretation.
The monotonic layer: declarative semantics

Minimal/least models?

Let $S$ be the following set of rules:

\[
\begin{align*}
\text{wait} & : \leftarrow \text{overloaded} \lor \text{rest\_time}. \\
\text{rest\_time} & : \leftarrow \text{wait}. \\
\neg \text{overloaded} & : \leftarrow \text{rest\_time}. \\
\text{overloaded} & .
\end{align*}
\]

A minimal model of $S$ is

\[
\{ \text{overloaded}, \neg \text{overloaded}, \text{wait}, \text{rest\_time} \}.
\]

There are no facts supporting the truth of $\text{wait}$ and $\text{rest\_time}$ in this model. The intuitively correct model for $S$ is

\[
\{ \text{overloaded}, \neg \text{overloaded}, \text{wait}, \neg \text{wait}, \text{rest\_time}, \neg \text{rest\_time} \}.
\]
The monotonic layer: declarative semantics

Minimal/least models?

Let $S$ be the following set of rules:

- \( wait \leftarrow overloaded \lor rest\_time . \)
- \( rest\_time \leftarrow wait . \)
- \( \neg overloaded \leftarrow rest\_time . \)
- \( overloaded . \)

A minimal model of $S$ is

\[ \{overloaded, \neg overloaded, wait, rest\_time\} . \]

There are no facts supporting the truth of $wait$ and $rest\_time$ in this model. The intuitively correct model for $S$ is

\[ \{overloaded, \neg overloaded, wait, \neg wait, rest\_time, \neg rest\_time\} . \]
**Well-supported model (formal definition in the paper)**

Intuitively, a *well-supported model* is a model where each literal has value \( t \) or \( i \) iff this is forced by a finite derivation starting from facts.

**Theorem**

For any set of rules \( S \) there is the unique well-supported model.

**Theorem**

Computing the well-supported model is in \( \text{PTime} \) w.r.t. the size of the database domain.
The monotonic layer: declarative semantics

Well-supported model (formal definition in the paper)
Intuitively, a well-supported model is a model where each literal has value $t$ or $i$ iff this is forced by a finite derivation starting from facts.

Theorem
For any set of rules $S$ there is the unique well-supported model.

Theorem
Computing the well-supported model is in $\text{PTime}$ w.r.t. the size of the database domain.
The monotonic layer: declarative semantics

Well-supported model (formal definition in the paper)

Intuitively, a well-supported model is a model where each literal has value $t$ or $i$ iff this is forced by a finite derivation starting from facts.

Theorem

For any set of rules $S$ there is the unique well-supported model.

Theorem

Computing the well-supported model is in $\mathbf{PTime}$ w.r.t. the size of the database domain.
The monotonic layer: computing the well-supported model

**Algorithm**

**Input:** a set of rules \( S \)

**Output:** the unique well-supported model \( \mathcal{I}^S \) for \( S \)

1. *(finding basic inconsistencies):*
   - compute the least Herbrand model \( \mathcal{I}^S_0 \) of \( \text{Pos}(S) \), where by \( \text{Pos}(S) \) we understand the DATALOG program obtained from \( S \) by replacing each negative literal \( \neg \ell \) of \( S \) by its (unique and fresh) duplicate \( \ell' \)
   - let \( \mathcal{I}^S_1 \overset{\text{def}}{=} \{ \ell, \neg \ell \mid \ell, \ell' \in \mathcal{I}^S_0 \} \)

2. *(finding potentially true literals):*
   - let \( S' = \{ \rho \mid \rho \in S \text{ and } \mathcal{I}^S_1(\text{head}(\rho)) \neq i \} \)
   - set \( \mathcal{I}^S_2 \) to be the the least Herbrand model for \( \text{Pos}(S') \)
   - set \( \mathcal{I}^S \overset{\text{def}}{=} \{ \ell \mid \ell \in \mathcal{I}^S_2 \text{ and } \ell \text{ is not primed} \} \cup \{ \neg \ell \mid \ell' \in \mathcal{I}^S_2 \} \).
The monotonic layer: computing the well-supported model

Algorithm

**Input:** a set of rules $S$

**Output:** the unique well-supported model $I^S$ for $S$

1. (finding basic inconsistencies):
   - compute the least Herbrand model $I^S_0$ of $Pos(S)$, where by $Pos(S)$ we understand the DATALOG program obtained from $S$ by replacing each negative literal $\neg \ell$ of $S$ by its (unique and fresh) duplicate $\ell'$
   - let $I^S_1 \overset{\text{def}}{=} \{ \ell, \neg \ell \mid \ell, \ell' \in I^S_0 \}$

2. (finding potentially true literals):
   - let $S' = \{ \varrho \mid \varrho \in S \text{ and } I^S_1(\text{head}(\varrho)) \neq i \}$
   - set $I^S_2$ to be the the least Herbrand model for $Pos(S')$
   - set $I^S_2 \overset{\text{def}}{=} \{ \ell \mid \ell \in I^S_2 \text{ and } \ell \text{ is not primed} \} \cup \{ \neg \ell \mid \ell' \in I^S_2 \}$. 

Jan Małuszyński and Andrzej Szalas

Datalog 2.0, Oxford, UK 22/41
Algorithm

**Input:** a set of rules $S$

**Output:** the unique well-supported model $I^S$ for $S$

1. *(finding basic inconsistencies):*
   - compute the least Herbrand model $I^S_0$ of $\text{Pos}(S)$, where by $\text{Pos}(S)$ we understand the Datalog program obtained from $S$ by replacing each negative literal $\neg \ell$ of $S$ by its (unique and fresh) duplicate $\ell'$
   - let $I^S_1 \overset{\text{def}}{=} \{ \ell, \neg \ell \mid \ell, \ell' \in I^S_0 \}$

2. *(finding potentially true literals):*
   - let $S' = \{ \varrho \mid \varrho \in S \text{ and } I^S_1(\text{head}(\varrho)) \neq i \}$
   - set $I^{S'}_2$ to be the the least Herbrand model for $\text{Pos}(S')$
   - set $I^S_2 \overset{\text{def}}{=} \{ \ell \mid \ell \in I^{S'}_2 \text{ and } \ell \text{ is not primed} \} \cup \{ \neg \ell \mid \ell' \in I^{S'}_2 \}$. 
Algorithm – continued

3 (reasoning with inconsistency):

- define the following transformation $\Phi^S$ on interpretations:
  $$\Phi^S(\mathcal{I}) \overset{\text{def}}{=} \mathcal{I} \cup \{ \ell, \neg \ell \mid \text{there is a rule } [\ell : b_1 \lor \ldots \lor b_m] \in S \text{ such that } \exists k \in \{1, \ldots, m\}[\mathcal{I}(b_k) = i] \text{ and } \neg \exists n \in \{1, \ldots, m\}[ (\mathcal{I}^S_2 - \mathcal{I})(b_n) = t] \}.$$  
  - The transformation $\Phi^S$ is monotonic (!)

Denote by $\mathcal{I}^S_3$ the fixpoint of $\Phi^S$ obtained by iterating $\Phi^S$ on $\mathcal{I}^S_1$, i.e.,

$$\mathcal{I}^S_3 = \bigcup_{i \in \omega}(\Phi^S)^i(\mathcal{I}^S_1)$$

- set $\mathcal{I}^S = \mathcal{I}^S_2 \cup \mathcal{I}^S_3$. 

Jan Małuszyński and Andrzej Szałas  
Datalog 2.0, Oxford, UK  
23/41
Algorithm – continued

3 (reasoning with inconsistency):

- Define the following transformation $\Phi^S$ on interpretations:

  $$\Phi^S(\mathcal{I}) \overset{\text{def}}{=} \mathcal{I} \cup \{ \ell, \neg \ell \mid \text{there is a rule } [\ell : b_1 \lor \ldots \lor b_m] \in S \text{ such that } \exists k \in \{1, \ldots, m\}[\mathcal{I}(b_k) = i] \text{ and } \neg \exists n \in \{1, \ldots, m\}[ (\mathcal{I}^S_2 - \mathcal{I})(b_n) = t] \}.$$ 

- The transformation $\Phi^S$ is monotonic (!)

Denote by $\mathcal{I}^S$ the fixpoint of $\Phi^S$ obtained by iterating $\Phi^S$ on $\mathcal{I}^S_1$, i.e.,

$$\mathcal{I}^S_3 = \bigcup_{i \in \omega} (\Phi^S)^i(\mathcal{I}^S_1).$$

- Set $\mathcal{I}^S = \mathcal{I}^S_2 \cup \mathcal{I}^S_3$. 

Algorithm – continued

3. (reasoning with inconsistency):
   - define the following transformation $\Phi^S$ on interpretations:
     \[
     \Phi^S(\mathcal{I}) \overset{\text{def}}{=} \mathcal{I} \cup \{ \ell, \neg \ell \mid \text{there is a rule } [\ell : b_1 \lor \ldots \lor b_m] \in S \text{ such that } \exists k \in \{1, \ldots, m\}[\mathcal{I}(b_k) = i] \text{ and } \neg \exists n \in \{1, \ldots, m\}[ (\mathcal{I}^S_2 - \mathcal{I})(b_n) = t] \}.
     \]
   - The transformation $\Phi^S$ is monotonic (!)
   - Denote by $\mathcal{I}_3^S$ the fixpoint of $\Phi^S$ obtained by iterating $\Phi^S$ on $\mathcal{I}_1^S$, i.e.,
     \[
     \mathcal{I}_3^S = \bigcup_{i \in \omega} (\Phi^S)^i(\mathcal{I}_1^S).
     \]
   - set $\mathcal{I}^S = \mathcal{I}_2^S \cup \mathcal{I}_3^S$. 
Example
Consider set of rules discussed in Slide 60:

\[
\begin{align*}
\text{wait} & :\leftarrow \text{overloaded} \lor \text{rest}_t\text{ime} . \\
\text{rest}_t\text{ime} & :\leftarrow \text{wait} . \\
\neg \text{overloaded} & :\leftarrow \text{rest}_t\text{ime} . \\
\text{overloaded} & . \\
\end{align*}
\]

together with rules:

\[
\begin{align*}
\text{good}_m\text{oood} & :\leftarrow \text{rested} \lor \text{success} . \\
\neg \text{rested} & :\leftarrow \neg \text{rest}_t\text{ime} . \\
\text{rested} & . \\
\text{success} & . \\
\end{align*}
\]
Example

Consider set of rules discussed in Slide 60:

\[
\begin{align*}
\text{wait} :&= \text{overloaded} \lor \text{rest\_time}. \\
\text{rest\_time} :&= \text{wait}. \\
\lnot \text{overloaded} :&= \text{rest\_time}. \\
\text{overloaded}. \\
\end{align*}
\]

together with rules:

\[
\begin{align*}
\text{good\_mood} :&= \text{rested} \lor \text{success}. \\
\lnot \text{rested} :&= \lnot \text{rest\_time}. \\
\text{rested}. \\
\text{success}. \\
\end{align*}
\]
Example continued

- Phase 1 gives $\mathcal{I}_1^S = \{ \text{overloaded}, \neg\text{overloaded} \}$.
- Phase 2 gives the following set $S'$:

\[
\begin{align*}
\text{wait} & :\!-\! \text{overloaded} \lor \text{rest\_time} . \\
\text{rest\_time} & :\!-\! \text{wait} . \\
\text{good\_mood} & :\!-\! \text{rested} \lor \text{success} . \\
\neg\text{rested} & :\!-\! \neg\text{rest\_time} . \\
\text{rested} & . \\
\text{success} & .
\end{align*}
\]

The resulting set $\mathcal{I}_2^S$ is $\{ \text{success}, \text{rested}, \text{good\_mood} \}$.
Example continued

- Phase 1 gives $I_1^S = \{\text{overloaded}, \neg \text{overloaded}\}$.
- Phase 2 gives the following set $S'$:

  \[
  \begin{align*}
  \text{wait} & : \neg \text{overloaded} \lor \text{rest\_time} . \\
  \text{rest\_time} & : \neg \text{wait} . \\
  \text{good\_mood} & : \neg \text{rested} \lor \text{success} . \\
  \neg \text{rested} & : \neg \text{rest\_time} . \\
  \text{rested} & . \\
  \text{success} & .
  \end{align*}
  \]

The resulting set $I_2^S$ is $\{\text{success}, \text{rested}, \text{good\_mood}\}$.
Example continued

- Phase 3 gives the following iterations of $\Phi^S$:

  \[
  \begin{align*}
  \{ & \text{overloaded}, \neg \text{overloaded} \} \\
  \{ & \text{overloaded}, \neg \text{overloaded}, \text{wait}, \neg \text{wait} \} \\
  \{ & \text{overloaded}, \neg \text{overloaded}, \text{wait}, \neg \text{wait}, \text{rest\_time}, \neg \text{rest\_time} \} \\
  \{ & \text{overloaded}, \neg \text{overloaded}, \text{wait}, \neg \text{wait}, \text{rest\_time}, \neg \text{rest\_time}, \\
  & \text{rested}, \neg \text{rested} \} - \text{fixpoint.}
  \end{align*}
  \]

  Hence $\mathcal{I}_3^S = \{ \text{overloaded}, \neg \text{overloaded}, \text{wait}, \neg \text{wait}, \\
  \text{rest\_time}, \neg \text{rest\_time}, \text{rested}, \neg \text{rested} \}$.

- Finally $\mathcal{I}_S^S = \{ \text{success}, \text{good\_mood}, \text{overloaded}, \neg \text{overloaded}, \\
  \text{wait}, \neg \text{wait}, \text{rest\_time}, \neg \text{rest\_time}, \text{rested}, \neg \text{rested} \}.$
Example continued

• Phase 3 gives the following iterations of $\Phi^S$:

\[
\begin{align*}
\{ & \text{overloaded, } \neg \text{overloaded} \} \\
\{ & \text{overloaded, } \neg \text{overloaded, wait, } \neg \text{wait} \} \\
\{ & \text{overloaded, } \neg \text{overloaded, wait, } \neg \text{wait, rest_time, } \neg \text{rest_time} \} \\
\{ & \text{overloaded, } \neg \text{overloaded, wait, } \neg \text{wait, rest_time, } \neg \text{rest_time, rested, } \neg \text{rested} \} \quad \text{− fixpoint.}
\end{align*}
\]

Hence $I_3^S = \{ \text{overloaded, } \neg \text{overloaded, wait, } \neg \text{wait, rest_time, } \neg \text{rest_time, rested, } \neg \text{rested} \}$.

• Finally $I^S = \{ \text{success, good_mood, overloaded, } \neg \text{overloaded, wait, } \neg \text{wait, rest_time, } \neg \text{rest_time, rested, } \neg \text{rested} \}$. 
The architecture

module $A$

$P :\neg \ldots A_i Q_i, \ldots$
$P :\neg \ldots \neg A_j Q_j \ldots$

$Q_i :\neg \ldots \neg A_i Q_i = u,$
$A_j Q_j = i \ldots$

Layer $i + 1$

Layer $i$

module $A_1$

module $A_2$

module $A_k$
External literals: the tool for expressing nonmonotonic rules

- An *external literal* is of one of the forms:
  \[ A.R, \neg A.R, A.R \text{ in } T, \neg A.R \text{ in } T, \]
  where:
  - \( A \) is a module (the *reference module* of the external literal) and \( R \) is a relation in \( A \)
  - \( \neg A.R \text{ in } T \) is to be read as “\((\neg A.R) \text{ in } T\)”
  - \( T \subseteq \{t, f, i, u\} \) (if \( T = \emptyset \) then \( \ell \text{ in } T \) is \( f \)).

- An external literal may only appear in rule bodies of a module \( B \), provided that
  - its relation appears in the head of a rule in its reference module
  - its reference module is in a strictly lower layer than \( B \).

- We write \( \ell = \nu \) rather than \( \ell \text{ in } \{\nu\} \).
External literals: the tool for expressing nonmonotonic rules

- An external literal is of one of the forms:
  \[ A.R, \neg A.R, A.R \text{ IN } T, \neg A.R \text{ IN } T, \]
  where:
  - \( A \) is a module (the reference module of the external literal) and \( R \) is a relation in \( A \)
    \((\neg A.R \text{ IN } T \) is to be read as \( (\neg A.R) \text{ IN } T \))
  - \( T \subseteq \{t, f, i, u\} \) (if \( T = \emptyset \) then \( \ell \text{ IN } T \) is \( f \)).
- An external literal may only appear in rule bodies of a module \( B \), provided that
  - its relation appears in the head of a rule in its reference module
  - its reference module is in a strictly lower layer than \( B \).
- We write \( \ell = v \) rather than \( \ell \text{ IN } \{v\} \).
Semantics of modules and external literals

- Formally, relation symbol $R$ occurring in module $A$ is an abbreviation for $A.R$.
- Each module operates on its “local” relations, accessing “external” relations only via dotted notation.
- External literals, when used in a given module, are fully defined in modules in lower layers.
- Relations assigned to external literals, when used, cannot change.
Semantics of modules and external literals

- Formally, relation symbol $R$ occurring in module $A$ is an abbreviation for $A.R$.
- Each module operates on its “local” relations, accessing “external” relations only via dotted notation.
- External literals, when used in a given module, are fully defined in modules in lower layers.
- Relations assigned to external literals, when used, cannot change.
Semantics of modules and external literals

- Formally, relation symbol $R$ occurring in module $A$ is an abbreviation for $A.R$.
- Each module operates on its “local” relations, accessing “external” relations only via dotted notation.
- External literals, when used in a given module, are fully defined in modules in lower layers.
- Relations assigned to external literals, when used, cannot change.
Semantics of modules and external literals

- Formally, relation symbol $R$ occurring in module $A$ is an abbreviation for $A.R$.
- Each module operates on its “local” relations, accessing “external” relations only via dotted notation.
- External literals, when used in a given module, are fully defined in modules in lower layers.
- Relations assigned to external literals, when used, cannot change.
Typical sources of nonmonotonicity

Generally, attempts to fill gaps in missing knowledge, e.g.,

- efficient representation of (negative) information (like CWA, LCWA)
- drawing rational conclusions from non-conclusive information (e.g., circumscription, default logics)
- drawing rational conclusions from the lack of knowledge (e.g., autoepistemic reasoning)
- resolving inconsistencies (e.g., defeasible reasoning).
Typical sources of nonmonotonicity

Generally, attempts to fill gaps in missing knowledge, e.g.,

- efficient representation of (negative) information (like $C_{WA}$, $LC_{WA}$)
- drawing rational conclusions from non-conclusive information (e.g., circumscription, default logics)
- drawing rational conclusions from the lack of knowledge (e.g., autoepistemic reasoning)
- resolving inconsistencies (e.g., defeasible reasoning).
Typical sources of nonmonotonicity

Generally, attempts to fill gaps in missing knowledge, e.g.,

- efficient representation of (negative) information (like $C_{WA}$, $LC_{WA}$)
- drawing rational conclusions from non-conclusive information (e.g., circumscription, default logics)
- drawing rational conclusions from the lack of knowledge (e.g., autoepistemic reasoning)
- resolving inconsistencies (e.g., defeasible reasoning).
Typical sources of nonmonotonicity

Generally, attempts to fill gaps in missing knowledge, e.g.,

- efficient representation of (negative) information (like $C_{WA}$, $LC_{WA}$)
- drawing rational conclusions from non-conclusive information (e.g., circumscription, default logics)
- drawing rational conclusions from the lack of knowledge (e.g., autoepistemic reasoning)
- resolving inconsistencies (e.g., defeasible reasoning).
Local Closed World Assumption

Intuitively, one often wants to contextually close part of the world, not necessarily all relations in the database.

Example

The following rules in a module other than $B$ locally close $location$:

\[
\text{location}(X, Y, T) :\neg\text{nextTime}(T, S), \text{house}(X), B.chngPos(X, S) \in \{u, f\}, C.location(X, Y, S).
\]

\[
\neg\text{location}(X, Y, T) :\neg\text{nextTime}(T, S), B.chngPos(X, S) \in \{u, t\}, C.location(X, Y, S).
\]
Local Closed World Assumption

Intuitively, one often wants to contextually close part of the world, not necessarily all relations in the database.

Example

The following rules in a module other than $B$ locally close $\text{location}$:

$\text{location}(X, Y, T) :\neg \quad \text{nextTime}(T, S), \text{house}(X),$
$\quad B.chngPos(X, S) \text{ IN } \{u, f\},$
$\quad C.location(X, Y, S).$

$\neg\text{location}(X, Y, T) :\neg \quad \text{nextTime}(T, S), \text{movingCar}(X),$
$\quad B.chngPos(X, S) \text{ IN } \{u, t\},$
$\quad C.location(X, Y, S).$
Some results

Theorem

$4QL$ with modules has $P\text{TIME}$ data complexity.

Theorem

Stratified $\text{DATALOG}$ programs are expressible in $4QL$.

Remark

Stratified $\text{DATALOG}$ captures $P\text{TIME}$ on ordered structures.
Some results

Theorem

$4QL$ with modules has $\text{PTime}$ data complexity.

Theorem

Stratified $\text{Datalog}$ programs are expressible in $4QL$.

Remark

Stratified $\text{Datalog}$ captures $\text{PTime}$ on ordered structures.
Some results

Theorem
4QL with modules has $\text{PTime}$ data complexity.

Theorem
Stratified Datalog programs are expressible in 4QL.

Remark
Stratified Datalog captures $\text{PTime}$ on ordered structures.
Default rules

Default rules have the form:

\[ \text{prerequisite} : \text{justification} \vdash \text{consequent}, \]

with the intuitive meaning

“deduce \textit{consequent} whenever \textit{prerequisite} is true

and \textit{justification} is consistent with current knowledge”.

Example: expressing default-like rules

Default rule:

\[ \text{car}(X) \land \text{speed}(X, \text{high}) : \text{onRoad}(X) \vdash \text{onRoad}(X) \]

captures similar intuitions as

\[ \text{onRoad}(X) \leftarrow \text{car}(X), \text{speed}(X, \text{high}), \]

\[ B.\text{onRoad}(X) \ \text{IN} \ \{t, u\}. \]
Default rules

Default rules have the form:

\[ \text{prerequisite} : \text{justification} \vdash \text{consequent}, \]

with the intuitive meaning

“deduce \textit{consequent} whenever \textit{prerequisite} is true and \textit{justification} is consistent with current knowledge”.

Example: expressing default-like rules

Default rule:

\[ \text{car}(X) \land \text{speed}(X, \text{high}) : \text{onRoad}(X) \vdash \text{onRoad}(X) \]

captures similar intuitions as

\[ \text{onRoad}(X) :- \text{car}(X), \text{speed}(X, \text{high}), \]
\[ B.\text{onRoad}(X) \text{ IN } \{t, u\}. \]
Defaults for resolving inconsistencies

Module $B$:

\[
\begin{align*}
\text{stop} & : \leftarrow \text{red\_light}.
\text{\neg stop} & : \leftarrow \text{policeman\_directs\_to\_go\_through}.
\end{align*}
\]

Module $A$:

\[
\begin{align*}
\text{\neg stop} & : \leftarrow B.\text{stop} = i.
\end{align*}
\]
Lightweight autoepistemic reasoning

The idea

1. A typical pattern of autoepistemic reasoning:
   “If you do not know $A$, conclude $\neg A$.”

2. The rule stating: “If you do not know that you have a sister, conclude that you do not have a sister” can be expressed in module $A \neq B$ by a rule assuming that knowledge of the reasoner is specified in module $B$:
   
   $\neg \text{have_sister} :\!-\! B.\text{have_sister} = \text{u}$. 

Jan Małuszyński and Andrzej Szałas

Datalog 2.0, Oxford, UK
The idea

1 A typical pattern of autoepistemic reasoning:
   “If you do not know $A$, conclude $\neg A$.”

2 The rule stating: “If you do not know that you have a sister, conclude that you do not have a sister” can be expressed in module $A \neq B$ by a rule assuming that knowledge of the reasoner is specified in module $B$:
   
   \[ \neg \text{have_sister} :- B \cdot \text{have_sister} = \text{u}. \]
Abnormality theories

In general, replacing circumscription by rules is not doable. However, abnormality theories are typically expressed by formulas of the following pattern:

\[(condition \land \neg\text{abnormal}) \rightarrow \text{conclusion}.\]

In such cases one can:

- locally close abnormality
- make varied predicates heads of rules
  (this Sometimes requires finding their definitions. Even if often can be done automatically, this is not a lightweight task).
Abnormality theories

In general, replacing circumscription by rules is not doable. However, abnormality theories are typically expressed by formulas of the following pattern:

\[(\text{condition} \land \neg \text{abnormal}) \rightarrow \text{conclusion}.\]

In such cases one can:

- locally close abnormality
- make varied predicates heads of rules
  (this sometimes requires finding their definitions. Even if often can be done automatically, this is not a lightweight task).
Example

Consider the theory:

\[ \forall X[(\text{ill}(X) \land \neg \text{ab}(X)) \rightarrow \text{consults\_doctor}(X)] \]

and assume one minimizes \(\text{ab}\) varying \(\text{consults\_doctor}\).

Let \(B\) be a module with (among others) the following rule:

\[ \text{ab}(X) :\leftarrow \text{ill}(X), \neg \text{consults\_doctor}(X). \]

We define a module \(A\), consisting of rules:

\[ \neg \text{ab}(X) :\leftarrow B.\text{ab}(X) \text{ in } \{f, u\}. \]

\[ \text{consults\_doctor}(X) :\leftarrow \neg \text{ab}(X), \text{ill}(X). \]
Example

Consider the theory:

$$\forall X[(ill(X) \land \neg ab(X)) \rightarrow consults\_doctor(X)]$$

and assume one minimizes $ab$ varying $consults\_doctor$. Let $B$ be a module with (among others) the following rule:

$$ab(X) :\neg ill(X), \neg consults\_doctor(X).$$

We define a module $A$, consisting of rules:

$$\neg ab(X) : \neg B.ab(X) \text{ IN } \{f, u\}.$$  
$$consults\_doctor(X) : \neg ab(X), ill(X).$$
Example

Consider the following defeasible rules reflecting buyer’s requirements as to apartments:

\[
\begin{align*}
    r_1 & : \text{size}(X, \text{large}) \Rightarrow \text{acceptable}(X) \\
    r_2 & : \neg \text{pets\_allowed}(X) \Rightarrow \neg \text{acceptable}(X)
\end{align*}
\]

with priorities \( r_2 > r_1 \).
Example continued

Assume module $B$ contains rules:

\[
\text{acceptable}(X) :\leftarrow \text{size}(X, \text{large}). \\
\neg \text{acceptable}(X) :\leftarrow \neg \text{pets\_allowed}(X).
\]

The following rules in some other module resolves possible inconsistencies according to required priority (but note that we have also cases with $u$, not covered by defeasible rules).

\[
\text{acceptable}(X) :\leftarrow B.\text{acceptable}(X) = t. \\
\neg \text{acceptable}(X) :\leftarrow B.\text{acceptable}(X) = i.
\]
Conclusions

• The proposed $4QL$ is powerful but still lightweight and intuitive. It provides means for monotonic reasoning supported by facts together with a mechanism for expressing nonmonotonic rules.

• The intended methodology:
  • the lowest layer provides solid knowledge, supported by facts, e.g., reflecting perception, expert knowledge, etc.
  • higher layers allow one to derive conclusions still supported by facts or using various forms of nonmonotonic reasoning, usually reflecting expert knowledge.

• Open questions:
  • provide an efficient top-down query evaluation (e.g., resolution or tableaux-based). We have one, but it is complex ($\text{ExpTime}$ in the worst case)
  • is the provided algorithm for computing well-supported model time-optimal?
Conclusions

• The proposed 4QL is powerful but still lightweight and intuitive. It provides means for monotonic reasoning supported by facts together with a mechanism for expressing nonmonotonic rules.

• The intended methodology:
  • the lowest layer provides solid knowledge, supported by facts, e.g., reflecting perception, expert knowledge, etc.
  • higher layers allow one to derive conclusions still supported by facts or using various forms of nonmonotonic reasoning, usually reflecting expert knowledge.

• Open questions:
  • provide an efficient top-down query evaluation (e.g., resolution or tableaux-based).
    We have one, but it is complex (ExpTime in the worst case)
  • is the provided algorithm for computing well-supported model time-optimal?
The proposed 4QL is powerful but still lightweight and intuitive. It provides means for monotonic reasoning supported by facts together with a mechanism for expressing nonmonotonic rules.

The intended methodology:
- the lowest layer provides solid knowledge, supported by facts, e.g., reflecting perception, expert knowledge, etc.
- higher layers allow one to derive conclusions still supported by facts or using various forms of nonmonotonic reasoning, usually reflecting expert knowledge.

Open questions:
- provide an efficient top-down query evaluation (e.g., resolution or tableaux-based). We have one, but it is complex (ExpTime in the worst case)
- is the provided algorithm for computing well-supported model time-optimal?
Related work

The most relevant papers

- S. de Amo, M.S. Pais (Int. Journal of Approximate Reasoning 2007): use the same truth ordering, but assume $C_{WA}$ and only allow negation in the rule bodies.
- J. Alcântara, C.V. Damásio and L.M. Pereira (J. Applied Logic 2005): the focus on semantical integration of explicit and default negation.
- M.C. Fitting (Theoretical Computer Science 2002): syntactically the same programs, but uses Belnap’s logic.
- F. Fages (Methods of Logic in Computer Science 1994): an idea of (two-valued) well-supported models.