Inconsistency Management for Traffic Regulations

Harald Beck

Supervisors: Thomas Eiter & Thomas Krennwallner

November 18, 2013
Inconsistency Management for Traffic Regulations

- Traffic regulation order: 30 km/h speed limit along the blue line
Inconsistency Management for Traffic Regulations

- Traffic regulation order: 30 km/h speed limit along the blue line
- Traffic measure: legal information (intention)
Inconsistency Management for Traffic Regulations

- Traffic regulation order: 30 km/h speed limit along the blue line
- Traffic measure: legal information (intention)
- Q: Which traffic signs are required to *announce* this measure?
Inconsistency Management for Traffic Regulations

- Traffic regulation order: 30 km/h speed limit along the blue line
- Traffic measure: legal information (intention)
- Q: Which traffic signs are required to announce this measure?
Inconsistency Management for Traffic Regulations

- Traffic regulation order: 30 km/h speed limit along the blue line
- **Traffic measure**: legal information (intention)
- Q: Which traffic signs are required to *announce* this measure?
Inconsistency Management for Traffic Regulations

- Traffic regulation order: 30 km/h speed limit along the blue line
- Traffic measure: legal information (intention)
- Q: Which traffic signs are required to announce this measure?
Inconsistency Management for Traffic Regulations

- Traffic regulation order: 30 km/h speed limit along the blue line
- Traffic measure: legal information (intention)
- Q: Which traffic signs are required to announce this measure?
Inconsistency Management for Traffic Regulations

- Traffic regulation order: 30 km/h speed limit along the blue line
- No need for repeated start sign in this case
Inconsistency Management for Traffic Regulations

- Traffic regulation order: 30 km/h speed limit along the blue line
- Updates may have side effects
Data Management Goals (Use Cases)

- **Consistency**: Given a set of measures and/or signs on a street, are they consistent (w.r.t. the traffic regulation)?
Data Management Goals (Use Cases)

- **Consistency**: Given a set of measures and/or signs on a street, are they consistent (w.r.t. the traffic regulation)?

- **Correspondence**: Do measures and signs express the same “effects,” i.e., are there no unannounced measures or unjustified traffic signs?
Data Management Goals (Use Cases)

- **Consistency**: Given a set of measures and/or signs on a street, are they consistent (w.r.t. the traffic regulation)?

- **Correspondence**: Do measures and signs express the same “effects,” i.e., are there no unannounced measures or unjustified traffic signs?

- **Diagnosis**: Which minimal set of measures/signs explain inconsistency or non-correspondence?
Data Management Goals (Use Cases)

- **Consistency**: Given a set of measures and/or signs on a street, are they *consistent* (w.r.t. the traffic regulation)?

- **Correspondence**: Do measures and signs express the same “effects,” i.e., are there no unannounced measures or unjustified traffic signs?

- **Diagnosis**: Which minimal set of measures/signs *explain* inconsistency or non-correspondence?

- **Repair**: Which minimal *changes to the scenario* can resolve these problems?
Data Management Goals (Use Cases)

- **Consistency**: Given a set of measures and/or signs on a street, are they consistent (w.r.t. the traffic regulation)?
- **Correspondence**: Do measures and signs express the same “effects,” i.e., are there no unannounced measures or unjustified traffic signs?
- **Diagnosis**: Which minimal set of measures/signs explain inconsistency or non-correspondence?
- **Repair**: Which minimal changes to the scenario can resolve these problems?
- **Strict repair**: Repair measure & sign data at the same time
  - Practical use cases obtained as special cases
High-level approach (overview)

- Street maps: labelled, directed graphs
High-level approach (overview)

- Street maps: labelled, directed graphs
- Logic-based approach. Edges and labels reflected as atoms
High-level approach (overview)

- Street maps: labelled, directed graphs
- Logic-based approach. Edges and labels reflected as atoms
- Represent measures and signs by edge/node labels
High-level approach (overview)

- Street maps: labelled, directed graphs
- Logic-based approach. Edges and labels reflected as atoms
- Represent measures and signs by edge/node labels
- Traffic regulation: 2-stage evaluation approach by logical formulas
  - Translate into "effect" labels (i.e., a common language) by an effect mapping
  - Evaluate effects by a conflict specification, potentially creating "conflict" labels
High-level approach (overview)

- Street maps: labelled, directed graphs
- Logic-based approach. Edges and labels reflected as atoms
- Represent measures and signs by edge/node labels
- Traffic regulation: 2-stage evaluation approach by logical formulas
  - Translate into “effect” labels (i.e., a common language) by an effect mapping
  - Evaluate effects by a conflict specification, potentially creating “conflict” labels
- Inconsistency, if a conflict can be derived
High-level approach (overview)

- Street maps: labelled, directed graphs
- Logic-based approach. Edges and labels reflected as atoms
- Represent measures and signs by edge/node labels
- Traffic regulation: 2-stage evaluation approach by logical formulas
  - Translate into “effect” labels (i.e., a common language) by an effect mapping
  - Evaluate effects by a conflict specification, potentially creating “conflict” labels
- Inconsistency, if a conflict can be derived
- Leave open which predicate logic is used
Scenario

- **Labelled street graph** $G$. Sets of edge atoms
  \[\{\ldots, e(\text{lane}, v_2, v_3), e(\text{straight}, v_3, y_1), e(\text{right}, x_2, y_1), \ldots\}\]
Scenario

- Labelled street graph $G$. Sets of edge atoms
  \{ \ldots, e(lane, v_2, v_3), e(straight, v_3, y_1), e(right, x_2, y_1), \ldots \}

- Traffic measures $M$ (edge labels $\longleftarrow \longrightarrow$), e.g.: ($spl=$ speed limit)
  \{ m(spl(30), v_2, v_3), m(spl(30), v_3, y_1), m(spl(30), y_1, y_2) \}
Scenario

- **Labelled** street graph $G$. Sets of edge atoms
  \[
  \{ \ldots , e(\text{lane}, v_2, v_3), e(\text{straight}, v_3, y_1), e(\text{right}, x_2, y_1), \ldots \}
  \]

- Traffic **measures** $M$ (edge labels $\rightarrow$), e.g.: ($spl=$ speed limit)
  \[
  \{ m(spl(30), v_2, v_3), m(spl(30), v_3, y_1), m(spl(30), y_1, y_2) \}
  \]

- Traffic **signs** $S$ (node labels), e.g.:
  \[
  \{ s(\text{start}(spl(30)), v_2), s(\text{start}(spl(30)), y_1), s(\text{end}(spl(30)), y_2) \} \]
Effects (edge labels \( \rightarrow \rightarrow \)): common language to define meaning of both measures and signs, e.g.: \( \text{maxsp} = \text{maximum speed} \)

\[
\{f(\text{maxsp}(30), v_2, v_3), f(\text{maxsp}(30), v_3, y_1)\}
\]
effects (edge labels $\rightarrow$): common language to define meaning of both measures and signs, e.g.: ($maxsp = \text{maximum speed}$)

\[
\{f(maxsp(30), v_2, v_3), f(maxsp(30), v_3, y_1)\}
\]

- Effect mapping $P$: formulas to obtain effect labels in 1st mapping
Effects (edge labels $\rightarrow$): common language to define *meaning* of both measures and signs, e.g.: $(\text{maxsp} = \text{maximum speed})$

$$\{f(\text{maxsp}(30), v_2, v_3), f(\text{maxsp}(30), v_3, y_1)\}$$

- **Effect mapping** $P$: formulas to obtain effect labels in 1st mapping

- **$\mathcal{F}_G^P(I)$**: effects of (measure and sign) input $I$ on graph $G$ due to $P$
Conflicts

- Conflicts (node labels)
  
  \[ c(\text{bad-end}(\text{maxsp}(30)), y_1) \]

- Conflict specification \( Sp \): formulas to obtain conflict labels in 2nd mapping due to effects
Conflicts (node labels)

\[ c(\text{ambig-spl}, y_1) \]

Conflict specification \( Sp \): formulas to obtain conflict labels in 2nd mapping due to effects
Conflicts

- Conflicts (node labels)
  \[ c(\text{ambig-spl}, y_1) \]

- Conflict specification \( Sp \): formulas to obtain conflict labels in 2nd mapping due to effects

- \( C_G^{P,Sp}(I) \): conflicts of input \( I \) on graph \( G \) due to \( Sp \) and (effects obtained by) \( P \)
ASP mapping examples

- Effect mapping $P$

\[
  f(\text{maxsp}(K), V, W) \leftarrow m(\text{spl}(K), V, W)
\]
ASP mapping examples

▶ Effect mapping \( P \)

\[
f(\text{maxsp}(K), V, W) \leftarrow m(\text{spl}(K), V, W)
\]
\[
f(\text{maxsp}(K), V, W) \leftarrow s(\text{start}(\text{spl}(K)), V), e(\text{lane}, V, W)
\]
\[
f(\text{maxsp}(K), V, W) \leftarrow s(\text{start}(\text{spl}(K)), V), e(\text{straight}, V, W)
\]
ASP mapping examples

- Effect mapping $P$

\[
\begin{align*}
f(maxsp(K), V, W) & \leftarrow m(spl(K), V, W) \\
f(maxsp(K), V, W) & \leftarrow s(start(spl(K)), V), e(lane, V, W) \\
f(maxsp(K), V, W) & \leftarrow s(start(spl(K)), V), e(straight, V, W) \\
f(maxsp(K), V, W) & \leftarrow f(maxsp(K), U, V), in-dir(U, V), \\
    & \text{in-dir}(V, W), \text{not block-prop}(maxsp(K), V)
\end{align*}
\]
ASP mapping examples

- **Effect mapping** $P$

  \[
  f(\text{maxsp}(K), V, W) \leftarrow m(\text{spl}(K), V, W)
  \]

  \[
  f(\text{maxsp}(K), V, W) \leftarrow s(\text{start}(\text{spl}(K)), V), e(\text{lane}, V, W)
  \]

  \[
  f(\text{maxsp}(K), V, W) \leftarrow s(\text{start}(\text{spl}(K)), V), e(\text{straight}, V, W)
  \]

  \[
  f(\text{maxsp}(K), V, W) \leftarrow f(\text{maxsp}(K), U, V), \text{in-dir}(U, V),
  \]

  \[
  \text{in-dir}(V, W), \text{not block-prop}(\text{maxsp}(K), V)
  \]

  \[
  \text{block-prop}(\text{maxsp}(K), V) \leftarrow s(\text{end}(\text{spl}(K)), V)
  \]

  \[
  \cdots
  \]
ASP mapping examples

- **Effect mapping** \( P \)
  
  \[
  f(\text{maxsp}(K), V, W) \leftarrow m(\text{spl}(K), V, W)
  \]
  
  \[
  f(\text{maxsp}(K), V, W) \leftarrow s(\text{start}(\text{spl}(K)), V), e(\text{lane}, V, W)
  \]
  
  \[
  f(\text{maxsp}(K), V, W) \leftarrow s(\text{start}(\text{spl}(K)), V), e(\text{straight}, V, W)
  \]
  
  \[
  f(\text{maxsp}(K), V, W) \leftarrow f(\text{maxsp}(K), U, V), \text{in-dir}(U, V),
  \]
  
  \[
  \text{in-dir}(V, W), \text{not block-prop}(\text{maxsp}(K), V)
  \]
  
  \[
  \text{block-prop}(\text{maxsp}(K), V) \leftarrow s(\text{end}(\text{spl}(K)), V)
  \]
  
  \[
  \vdots
  \]

- **Conflict specification** \( Sp \)
  
  \[
  c(\text{ambig-spl}, V) \leftarrow f(\text{maxsp}(K), V, W),
  \]
  
  \[
  f(\text{maxsp}(L), V, W), K \neq L.
  \]
Given a set of measures and/or signs on a street, are they consistent (w.r.t. the traffic regulation)?
Given a set of measures and/or signs on a street, are they *consistent* (w.r.t. the traffic regulation)?

\[ \mathcal{C}_G^{P,Sp}(I) = \emptyset? \]
Given a set of measures and/or signs on a street, are they consistent (w.r.t. the traffic regulation)?

$\mathcal{CP}^{P,Sp}(I) = \emptyset$?

Example. $\mathcal{CP}^{P,Sp}(I) = \{c(ambig-spl, y_1)\}$
Do measures and signs express the same effects, i.e., are there no unannounced measures or unjustified traffic signs?
Do measures and signs express the same effects, i.e., are there no unannounced measures or unjustified traffic signs?

\[ \mathcal{F}_G^P(M) = \mathcal{F}_G^P(S) \]
Do measures and signs express the same effects, i.e., are there no unannounced measures or unjustified traffic signs?

\[
\mathcal{F}_G^P(M) = \mathcal{F}_G^P(S)^
\]

**Example.**

\[
f(\text{maxsp}(30), y_1, y_2) \in \mathcal{F}_G^P(M) \text{ unannounced: not in } \mathcal{F}_G^P(S)
\]

\[
f(\text{maxsp}(40), y_1, y_2) \in \mathcal{F}_G^P(S) \text{ unjustified: not in } \mathcal{F}_G^P(M)
\]
Which minimal set of measures/signs explain a set of conflicts?
Which minimal set of measures/signs explain a set of conflicts?

Given $C \subseteq C_{G}^{P,Sp}(I)$, find min. $J \subseteq I$ s.t. $C \subseteq C_{G}^{P,Sp}(J)$
Which minimal set of measures/signs explain a set of conflicts?

Given \( C \subseteq C_{G}^{P,Sp}(I) \), find min. \( J \subseteq I \) s.t. \( C \subseteq C_{G}^{P,Sp}(J) \)

Example. \( C = \{c(ambig\text{-}spl, y_1)\} \).

\[ J = \{m(spl(30), y_1, y_2), s(start(spl(40)), y_1)\} \]
Repair & Strict repair

Which minimal changes to the scenario can resolve the conflicts?
Repair & Strict repair

- Which minimal changes to the scenario can resolve the conflicts?
- Find “good” deletions $I^- \subseteq I$ and additions $I^+ \subseteq I_G \setminus I$
  s.t. $C_{G}^{P,Sp}(I') = \emptyset$, where $I' = (I \setminus I^-) \cup I^+$
Which minimal changes to the scenario can resolve the conflicts?

Find “good” deletions $I^- \subseteq I$ and additions $I^+ \subseteq I_G \setminus I$

s.t. $C^P,Sp_G(I') = \emptyset$, where $I' = (I \setminus I^-) \cup I^+$

Strict repair: ... and $F^P_G(I' \cap M_G) = F^P_G(I' \cap S_G)$
Strict repair example

- Conflicts $C = \{c(\text{ambig-spl}, y_1)\}$
Strict repair example

- Conflicts $C = \{c(ambig\text{-}spl, y_1)\}$

- Repair 1. Preference: Minimal number of changes. 
  $I^- = \{m(spl(30), y_1, y_2)\}, I^+ = \{m(spl(40), y_1, y_2)\}$
Strict repair example

- Conflicts $C = \{c(\text{ambig-spl}, y_1)\}$

- Repair 2. Prefer sign changes over measures changes.
  
  $I^- = \{s(\text{start}(\text{spl}(40)), y_1), s(\text{end}(\text{spl}(40)), y_2)\}$
  
  $I^+ = \{s(\text{start}(\text{spl}(30)), y_1)\}$
Summary /1

- Domain analysis
  - No scientific literature was available
  - Meaning of traffic measures & signs
  - Problems which may occur (conflicts)
  - Identification of use cases
  - Challenges & technical approach
Summary

- **Domain analysis**
  - No scientific literature was available
  - Meaning of traffic measures & signs
  - Problems which may occur (conflicts)
  - Identification of use cases
  - Challenges & technical approach

- **Formal model** for traffic regulations
  - Street graph, traffic measures & signs
  - Effects & conflicts
  - Logic-based traffic regulation / specification
Summary /2

- **Reasoning tasks**
  - Consistency evaluation
  - Correspondence
  - Diagnosis
    - Independence / context of conflicts
  - Repair
    - Relations between diagnoses and repairs
  - Strict repair
    - Adjustment, Generation
### Summary /2

- **Reasoning tasks**
  - Consistency evaluation
  - Correspondence
  - Diagnosis
    - Independence / context of conflicts
  - Repair
    - Relations between diagnoses and repairs
  - Strict repair
    - Adjustment, Generation

- **Complexity** results for different logics (FOL + 3 ASPs)
Summary /2

- **Reasoning tasks**
  - Consistency evaluation
  - Correspondence
  - Diagnosis
    - Independence / context of conflicts
  - Repair
    - Relations between diagnoses and repairs
  - Strict repair
    - Adjustment, Generation

- **Complexity** results for different logics (FOL + 3 ASPs)
- **Implementation** prototype with Answer Set Programming
  - uniform encoding and specification for all use cases
  - core program + simple extensions
  - highly modular due to formal model & *sets* of rules
C’est ça
C'est ça
Thank you!

Thomas Eiter
Thomas Krennwallner
Stefan Kollarits
Torsten Schönberg
Marlene Handschuh
Christoph Hillinger
Roman Steiner
Traffic Regulation Problem

- Scenario $Sc = (G, M, S)$
  - Street graph $G$
  - Traffic measures $M$
  - Traffic signs $S$
  - Ground atoms of form $e(t, v, w), m(t, v, w), s(t, v)$

- Traffic regulation $\Pi = (P, Sp)$ in a predicate logic $L$
  - Effect mapping $P: M, S \mapsto \text{effects } F^P_G(M \cup S)$
  - Conflict specification $Sp: F^P_G(M \cup S) \mapsto \text{conflicts } C^P_{Sp}G(M \cup S)$

- Traffic Regulation Problem $\mathcal{T} = (Sc, \Pi)$
2-stage Mapping

- Closed world operator: \( \hat{X} = X \cup \{ \neg x \mid x \in Y \setminus X \} \), \( Y \) implicit
2-stage Mapping

- Closed world operator: $\hat{X} = X \cup \{\neg x \mid x \in Y \setminus X\}$, $Y$ implicit

- Theory $T$, atom sets $X$ (input), $Y$ (base set), graph $G$

$$Cn_G(T, X, Y) = \{y \in Y \mid T \cup \hat{G} \cup \hat{X} \models y\}$$
2-stage Mapping

- Closed world operator: \( \hat{X} = X \cup \{ \neg x \mid x \in Y \setminus X \} \), \( Y \) implicit

- Theory \( T \), atom sets \( X \) (input), \( Y \) (base set), graph \( G \)

\[
Cn_G(T, X, Y) = \{ y \in Y \mid T \cup \hat{G} \cup \hat{X} \models y \}
\]

- Base sets \( M_G/S_G/F_G/C_G \): measures/signs/effects/conflicts on \( G \)
2-stage Mapping

- Closed world operator: \( \hat{X} = X \cup \{ \neg x \mid x \in Y \setminus X \} \), \( Y \) implicit
- Theory \( T \), atom sets \( X \) (input), \( Y \) (base set), graph \( G \)
  \[ Cn_G(T, X, Y) = \{ y \in Y \mid T \cup \hat{G} \cup \hat{X} \models y \} \]
- Base sets \( M_G/S_G/F_G/C_G \): measures/signs/effects/conflicts on \( G \)
- 2 stages: Effect mapping \( P \), Conflict specification \( Sp \)
2-stage Mapping

- Closed world operator: $\hat{X} = X \cup \{ \neg x \mid x \in Y \setminus X \}$, $Y$ implicit
- Theory $P$, atom sets $I$ (input), $F_G$ (base set: effects), graph $G$
  \[ Cn_G(P, I, F_G) = \{ f(t, v, w) \in F_G \mid P \cup \hat{G} \cup \hat{I} \models f(t, v, w) \} \]
- Base sets $M_G/S_G/F_G/C_G$: measures/signs/effects/conflicts on $G$
- 2 stages: Effect mapping $P$, Conflict specification $Sp$

<table>
<thead>
<tr>
<th>operator</th>
<th>$Cn_G(P, I, F_G)$</th>
<th>$Cn_G(Sp, \mathcal{F}_G^P(I), C_G)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>labels</td>
<td>$I$</td>
<td>$\mathcal{F}_G^P(I)$</td>
</tr>
<tr>
<td>base set</td>
<td>$M_G \cup S_G$</td>
<td>$F_G$</td>
</tr>
</tbody>
</table>
2-stage Mapping

- Closed world operator: \( \hat{X} = X \cup \{ \neg x \mid x \in Y \setminus X \} \), \( Y \) implicit

- Theory \( Sp \), atom sets \( F^P_G(I) \) (input), \( C_G \) (base set: conflicts), graph \( G \)

\[
Cn_G(Sp, F^P_G(I), C_G) = \{ c(t, v) \in C_G \mid Sp \cup \hat{G} \cup \widehat{F^P_G(I)} \models c(t, v) \}
\]

- Base sets \( M_G / S_G / F_G / C_G \): measures/signs/effects/conflicts on \( G \)

- 2 stages: Effect mapping \( P \), Conflict specification \( Sp \)

<table>
<thead>
<tr>
<th>operator</th>
<th>( Cn_G(P, I, F_G) )</th>
<th>( Cn_G(Sp, F^P_G(I), C_G) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>labels</td>
<td>( I )</td>
<td>( F^P_G(I) )</td>
</tr>
<tr>
<td>base set</td>
<td>( M_G \cup S_G )</td>
<td>( F_G )</td>
</tr>
</tbody>
</table>
Answer Set Programming prototype

- Experiments with clingo and dlv

- Uniform approach towards all reasoning tasks. Idea:
  - Repair will potentially require new atoms \( \rightsquigarrow \)
  - Input atoms form initial pool
  - Each measure/sign from the pool can either be used or not
  - Only the effects of used measures & signs are computed
  - View reasoning tasks as constraints on this usage

- Measure & signs: function symbols \( x \in \{ m, s \} \):
  - input\( (x(\ldots)) \). measure/sign is given as input
  - pool\( (x(\ldots)) \). measure/sign is in pool (for guessing)
  - use\( (x(\ldots)) \). measure/sign is used
Appendix

ASP Implementation Prototype

- **guess:** $\Pi \cup G \cup I \cup Pool$
  - $\Pi$: traffic regulation / specification
  - $G$: street graph
  - $I$: input (measures & signs); initial pool
  - $Pool$: $use(X) \lor \neg use(X) :- pool(X)$. 

check: additional constraints based on reasoning task

Eval:

Diagnosis:

Repair:
**ASP Implementation Prototype**

- **guess:** $\Pi \cup G \cup I \cup \text{Pool}$
  - $\Pi$: traffic regulation / specification
  - $G$: street graph
  - $I$: input (measures & signs); initial pool
  - $\text{Pool}$: \text{use}(X) ∨ \neg \text{use}(X) :- \text{pool}(X)$

- **check:** additional constraints based on reasoning task
  - **Eval:** $\text{use}(X) :- \text{input}(X)$
  - **Diagnosis:** $:- \not c(t, v)$ for resp. conflicts $c(t, v)$
  - **Repair:** $:- c(t, v)$ for resp. conflicts $c(t, v)$
Implementation of conflict evaluation (consistency)

- Compute $C_{G}^{P,Sp}(I)$, i.e., the set of conflicts derivable from the input $I \subseteq M_{G} \cup S_{G}$ whether $T$ is consistent.

- Approach: Use entire input. Add rule to effect mapping:

  $$\text{use}(x(\ldots)) :\text{- input}(x(\ldots)).$$

- $T$ is consistent iff answer set does not contain a conflict atom.
Implementation of conflict evaluation (consistency)

- Compute $C^P_{Sp} (I)$, i.e., the set of conflicts derivable from the input $I \subseteq M_G \cup S_G$ whether $T$ is consistent

- Approach: Use entire input. Add rule to effect mapping:

  \[ \text{use}(x(...)) :- \text{input}(x(...)). \]

- $T$ is consistent iff answer set does not contain a conflict atom.

- Example:

  \begin{verbatim}
  input(m(spl(30),v2,v3)).
  input(m(spl(30),v3,y1)).
  input(m(spl(30),y1,y2)).
  \end{verbatim}

  in effect mapping:

  \[ f(\text{maxsp}(K),V,W) :- \text{e}(T,V,W), \#\text{int}(K), \text{use}(m(\text{spl}(K),V,W)). \]
Flexible modifications to the input

Let $x \in \{m, s\}$.

- Input is initial pool, which may be used or not.
  
  $\text{pool}(x(...)) :- \text{input}(x(...)).$
  $\text{use}(x(...)) \lor \neg \text{use}(x(...)) :- \text{pool}(x(...)).$

- General modifications possible:
  
  $\text{keep}(x(...)) :- \text{use}(x(...)), \text{input}(x(...)).$
  $\text{del}(x(...)) :- \neg \text{use}(x(...)), \text{input}(x(...)).$
  $\text{add}(x(...)) :- \text{use}(x(...)), \neg \text{input}(x(...)).$
Diagnosis implementation

- Approach: Given a set of conflicts $C$ to be diagnosed, add to conflict specification for each $c(t,v) \in C$ a rule
  
  $$:- \text{not } c(t,v).$$

- Diagnosis: Keep as few input atoms $J \subseteq I$ as possible such that $C \subseteq C^{P,Sp}_G(J)$, and do not allow additions.
  
  $$:- \text{add}(x(...)). \quad \% \text{adding not allowed}$$
  $$:\sim \text{keep}(x(...)). \quad \% \text{keep as few as possible}$$
Evaluation: \( \{c(\text{ambig-spl},y_1)\} \)

Add to conflict specification

\[\begin{align*}
  &\text{: not } c(\text{ambig-spl},y_1) \\
  &\text{: add } (m(T,X,V)). \quad \text{: add } (s(T,V)). \\
  &\text{: keep } (m(T,X,V)). \quad \text{: keep } (s(T,V)).
\end{align*}\]
Result:
\[
\{ \begin{align*}
\text{keep}(m(\text{spl}(30),y_1,y_2)) & . \\
\text{keep}(s(\text{start}(\text{spl}(30)),y_1)) & . 
\end{align*} \}
\]

Add to conflict specification
\[
\begin{align*}
& \text{:- not c(ambig-spl,y_1)} \\
& \text{:- add}(m(T,X,V)) . \text{:- add}(s(T,V)) . \\
& \text{~ keep}(m(T,X,V)) . \text{~ keep}(s(T,V)) .
\end{align*}
\]
Repair implementation

- Must add new measures/signs to the pool based on domain knowledge, e.g.,
  - If there is a measure \( m(T, X, Y) \) in the pool, add a start sign at \( X \) and an end sign at \( Y \) to the pool.
    
    \[
    \text{pool}(s(\text{start}(T), X)) :- \text{pool}(m(T, X, Y)).
    \]
    
    \[
    \text{pool}(s(\text{end}(T), Y)) :- \text{pool}(m(T, X, Y)).
    \]
Repair implementation

- Must add new measures/signs to the pool based on domain knowledge, e.g.,
  - If there is a measure $m(T, X, Y)$ in the pool, add a start sign at $X$ and an end sign at $Y$ to the pool.
    
    $\text{pool}(s(\text{start}(T), X)) : \text{pool}(m(T, X, Y))$.
    $\text{pool}(s(\text{end}(T), Y)) : \text{pool}(m(T, X, Y))$.

- Approach: Add/delete as little as possible such that no conflict is derived
  
  $\text{:} \sim \text{c}(T, V)$. % forbid any conflict
  $\text{:} \sim \text{del}(s(T, X))$.
  $\text{:} \sim \text{add}(s(T, X))$.
  $\text{:} \sim \text{del}(m(T, X, Y))$.
  $\text{:} \sim \text{add}(m(T, X, Y))$.
Repair implementation

- Must add new measures/signs to the pool based on domain knowledge, e.g.,
  - If there is a measure $m(T, X, Y)$ in the pool, add a start sign at $X$ and an end sign at $Y$ to the pool.
    
    $$
    \text{pool}(s(\text{start}(T), X)) :\neg \text{pool}(m(T, X, Y)).
    $$
    $$
    \text{pool}(s(\text{end}(T), Y)) :\neg \text{pool}(m(T, X, Y)).
    $$
  
- Approach: Add/delete as little as possible such that no conflict is derived
  
  $$
  :\neg c(T, V). \quad \% \text{forbid any conflict}
  $$
  $$
  :\sim \text{del}(s(T, X)). \quad \% \text{prefer changes of signs} \; [:1]
  $$
  $$
  :\sim \text{add}(s(T, X)). \quad \% \text{prefer changes of signs} \; [:2]
  $$
  $$
  :\sim \text{del}(m(T, X, Y)). \quad \% \text{then deletions} \; [1:] 
  $$
  $$
  :\sim \text{add}(m(T, X, Y)). \quad \% \text{then additions} \; [2:].
  $$

- $dlv$ optimizes hierarchically: $:\sim <\text{body}>$. [Weight:Level]
Strict repair example

- Add to conflict specification (rules shown before and)
  :- c(ambig-spl, y1)

- Result (without preferences)
  \[
  \{ \text{del}(\text{m}(\text{spl}(30), y1, y2)). \text{add}(\text{m}(\text{spl}(40), y1, y2)). \}
  \]
Appendix

Strict repair example /2

- Add to conflict specification (rules shown before and)
  :- c(ambig-spl,y1)

- Result with preference to change signs
  
  \[
  \{ \text{del}(s(\text{start}(\text{spl}(40)),y1))., \\
  \text{del}(s(\text{end}(\text{spl}(40)),y2))., \\
  \text{add}(s(\text{start}(\text{spl}(30)),y1))., \\
  \text{add}(s(\text{end}(\text{spl}(30)),y2)). \}
  \]
Adjustment & Generation

- Restricted scenarios / restricting repairs lead to special cases, relevant for data imports and merging.
- **Adjustment** of signs, s.t. they correspond with measures. Amounts to finding a repair consisting exclusively of traffic signs. (Recall 30 km/h example.)
- **Generation** of signs from scratch, s.t. they correspond with measures. Corresponds to a repair \((\emptyset, I^+)\) on scenario \((G, M, \emptyset)\), where \(I^+\) consists exclusively of signs.

Example (encoding of special domain knowledge)

- Favor changes in signs over changes in measures
- Favor deletions of linear measures over zones
- Never delete a residential area
- . . .
Reasoning Tasks: Theory

- **Def.** Set of conflicts $C$ is independent of $Y \subseteq I$ if for each diagnosis $J$ for $C$ and each $Y' \subseteq Y$, $J \setminus Y'$ is also a diagnosis for $C$.

- **Def.** A context for $C$ is a set $X \subseteq I$ s.t. i) $C$ is independent of $I \setminus X$ and ii) $C$ is not independent of any non-empty $X' \subseteq X$

- **Prop.** Context of each $C$ is unique

- **Prop.** All $\subseteq$-minimal diagnoses are in the context

- **Thm.** The context is the union of minimal elements of maximal convex subsets of the set of diagnoses
  - **Def.** Collection of sets $S$ convex if it has no ‘holes’, i.e., the property that $S \subseteq S'' \subseteq S'$ and $S, S' \in S$ implies $S'' \in S$

- **Cor.** If set of diagnoses is convex, then context equals union of $\subseteq$-minimal diagnoses
Reasoning Tasks: Theory /2

- **Def.** $\mathcal{D}_T(C)$: set of diagnoses for conflicts $C$
- **Def.** $H \subseteq \bigcup_{i=1}^{n} X_i$ hitting set for $S = \{X_1, \ldots, X_n\}$ if $H \cap X \neq \emptyset$ for all $X \in S$
- **Def.** $T[I^-, I^+]$ updated $T$ due to $(I^-, I^+)$
- **Prop.** If $J \subseteq I$ is a hitting set for $\mathcal{D}_T(C)$, then $C \nsubseteq C(T[J, \emptyset])$
- Due to potential side effects $C \cap C(T[J, \emptyset]) = \emptyset$ is *not* guaranteed
- **Consequence:** In general, it does *not* suffice to delete a minimal hitting set for all ($\subseteq$-minimal) diagnoses
Reasoning Tasks: Decision problems

Let $I \subseteq M_G \cup S_G$ be a set of measures and/or signs on a graph $G$

- **CONS**: decide $C_G^{P,Sp}(I) = \emptyset$, i.e., whether $\mathcal{T}$ is consistent

- **UMINDIAG**: decide, whether a unique $\subseteq$-minimal diagnosis exists, i.e., for given $C \subseteq C_G^{P,Sp}(I)$ a set $J \subseteq I$, s.t. $C \subseteq C_G^{P,Sp}(J)$

- **CORR**: decide $M$ and $S$ correspond, i.e., $F_G^P(M) = F_G^P(S)$

- **REPAIR**: decide whether an admissible repair exists, i.e., deleting some $I^- \subseteq I$ and adding new measures and signs $I^+$ s.t. modification is consistent
### Complexity of Reasoning Tasks

<table>
<thead>
<tr>
<th>Logic $\mathcal{L}$</th>
<th>IMPL</th>
<th>CONS</th>
<th>CORR</th>
<th>UMinDiag</th>
<th>REPAIR</th>
</tr>
</thead>
<tbody>
<tr>
<td>general</td>
<td>FO+DCA</td>
<td>co-NExp</td>
<td>P$^{\text{NExp}}$</td>
<td></td>
<td>p$^{\text{NExp}}$</td>
</tr>
<tr>
<td></td>
<td>ASP$^{-\text{s}}$</td>
<td>Exp</td>
<td>Exp</td>
<td>Exp</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ASP$^{-}$</td>
<td>co-NExp</td>
<td>P$^{\text{NExp}}$</td>
<td></td>
<td>p$^{\text{NExp}}$</td>
</tr>
<tr>
<td></td>
<td>ASP$^{\lor,\neg}$</td>
<td>co-NExp$^{\text{NP}}$</td>
<td>P$^{\text{NExp}}^{\text{NP}}$</td>
<td></td>
<td>p$^{\text{NExp}}^{\text{NP}}$</td>
</tr>
<tr>
<td>BPA</td>
<td>FO+DCA</td>
<td>PSpace</td>
<td>PSpace</td>
<td>PSpace</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ASP$^{-\text{s}}$</td>
<td>P$^{\text{NP}}$</td>
<td>P$^{\text{NP}}$</td>
<td>in P$^{\Sigma_2^p}$, $\Pi_2^p$-hard</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ASP$^{-}$</td>
<td>$\Pi_2^p$</td>
<td>P$^{\Sigma_2^p}$</td>
<td>in P$^{\Sigma_3^p}$, $\Pi_3^p$-hard</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ASP$^{\lor,\neg}$</td>
<td>$\Pi_3^p$</td>
<td>P$^{\Sigma_3^p}$</td>
<td>in P$^{\Sigma_4^p}$, $\Pi_4^p$-hard</td>
<td></td>
</tr>
</tbody>
</table>

Legend: general case / bounded predicate arities (completeness results unless stated otherwise)

- **FO+DCA:** first-order logic with domain closure assumption
- **ASP$^{-\text{s}}$, ASP$^{-}$, ASP$^{\lor,\neg}$:** stratified, normal, disjunctive answer set programs
- **IMPL:** Known logical entailment complexities
- **P$^O_\parallel$:** restricted P$^O$ s.t. all queries for $O$ are evaluable in parallel
Example: Loop

Scenario: Four mandatory left turns cause a loop