Revision programming

by

Inna Pivkina

Outline

1. Basic concepts of revision programming (RP).
2. Connection between logic programming and RP.
3. Extensions: disjunctive RP.
4. Dealing with uncertainty: Annotated RP.

RP:

- Formalism for describing and enforcing constraints on databases introduced by Marek and Truszczyński.
- Database - a collection of atomic facts from some universe.
- Revision rules
  - specify constraints on a database,
  - specify a preferred way to satisfy constraints.
- Arbitrary initial database.
- Justified revisions
  - satisfy all constraints,
  - all changes are justified by revision rules.

Example

Candidates: *Ann, Bob, David, Tom.*

Constraints: *(1) either Ann or Bob; (2) Tom can be only with David; (3) if David then no Ann; (4) if Bob then no David.*

Initial proposal: *David, Tom.*

Goal: *form a committee that*

- *satisfies all constraints*
- *differs minimally from the initial proposal*
- *all changes to the initial proposal are justified*

Example, cont'd

$$P: \quad \begin{aligned} \mathbf{in}(Ann) &\leftarrow \mathbf{out}(Bob) \\ \mathbf{in}(Bob) &\leftarrow \mathbf{out}(Ann) \\ \mathbf{in}(David) &\leftarrow \mathbf{in}(Tom) \\ \mathbf{out}(Tom) &\leftarrow \mathbf{out}(David) \\ \mathbf{out}(Ann) &\leftarrow \mathbf{in}(David) \\ \mathbf{out}(David) &\leftarrow \mathbf{in}(Bob) \end{aligned}$$

$$P_{I,R}: \quad \begin{aligned} \mathbf{in}(Ann) &\leftarrow \mathbf{out}(Bob) \\ \mathbf{in}(Bob) &\leftarrow \\ \mathbf{in}(David) &\leftarrow \mathbf{in}(Tom) \\ \mathbf{out}(Tom) &\leftarrow \mathbf{out}(David) \\ \mathbf{out}(Ann) &\leftarrow \mathbf{in}(David) \\ \mathbf{out}(David) &\leftarrow \mathbf{in}(Bob) \end{aligned}$$

Initial database: $I = \{David, Tom\}$.

Revision: $R = \{Bob\}$.

Inertia (no justification is needed): **out***(Ann)*.

Necessary change: $\mathbf{in}(Bob)$, $\mathbf{out}(David)$, $\mathbf{out}(Tom)$.

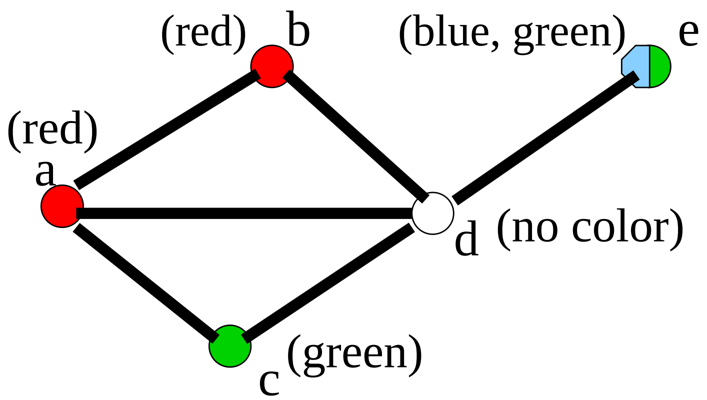
Updating I: $(I \cup \{Bob\}) \setminus \{David, Tom\}$.

Graph 3-colorability

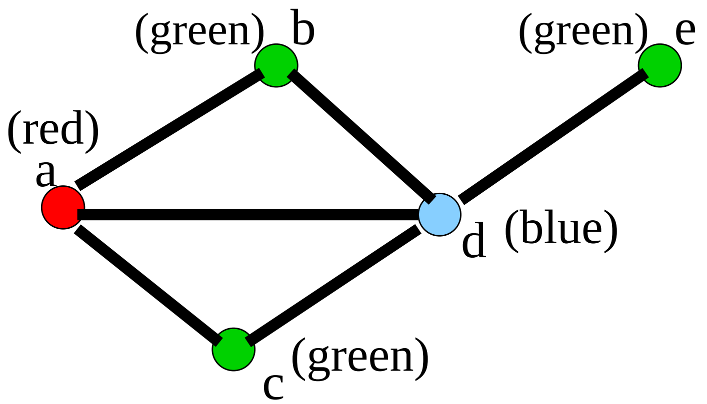
Problem: Given a coloring (which may be partial or inconsistent) find a coloring that differs minimally from the initial coloring and satisfies the following conditions:

- every vertex has exactly one color;
- any two vertices that are connected by an edge have different colors.

Initial database:



Revision:



Basic concepts

- Universe U. Elements of U - *atoms*. Subsets of U - *databases*.
- Revision literals: $\mathbf{in}(a)$, $\mathbf{out}(a)$ $(a \in U)$.
- *Revision rules*:

$$\mathbf{in}(a) \leftarrow \mathbf{in}(a_1), \ldots, \mathbf{in}(a_m), \mathbf{out}(b_1), \ldots, \mathbf{out}(b_n), \qquad \textit{(in-rule)}$$

$$\mathbf{out}(a) \leftarrow \mathbf{in}(a_1), \ldots, \mathbf{in}(a_m), \mathbf{out}(b_1), \ldots, \mathbf{out}(b_n), \qquad \textit{(out-rule)}$$

where a, a_i, $b_i \in U$ $\;(1 \leq i \leq n)$.

- *Revision program* - collection of revision rules.

Justified revisions: necessary change

- **Example.**

$$
\begin{array}{rcl}
P: \ \mathbf{in}(Ann) & \leftarrow & \\
\mathbf{out}(Bob) & \leftarrow & \mathbf{in}(Ann) \\
\mathbf{out}(Tom) & \leftarrow & \mathbf{out}(Ann)
\end{array}
\qquad NC(P) = \{\mathbf{in}(Ann), \mathbf{out}(Bob)\}
$$

- The *necessary change* of P, $NC(P)$, is the least model of P treated as a Horn program.

- α^D - *dual* of a literal α. $\quad \mathbf{in}(a)^D = \mathbf{out}(a)$, $\mathbf{out}(a)^D = \mathbf{in}(a)$.
- A set of literals L is *coherent* if it does not contain a pair of dual literals.
- Coherent L specifies a revision:

$$I \oplus L = (I \cup \{a\colon \mathbf{in}(a) \in L\}) \setminus \{a\colon \mathbf{out}(a) \in L\}.$$

Justified revisions

- To compute:
  - propose a candidate
  - check whether it is a justified revision or not
  - if not, consider another candidate
- *Inertia set* for databases I, R:
  $I(I, R) = \{\mathbf{in}(a) : a \in I \cap R\} \cup \{\mathbf{out}(a) : a \notin I \cup R\}$.
- Literals from inertia need no justification.
- *Reduct of P with respect to* (I, R) (denoted $P_{I,R}$) – the revision program obtained from P by eliminating from the body of each rule in P all literals in $I(I, R)$.
- P - a revision program, I and R - databases. R is called a *P-justified revision* of I if $NC(P_{I,R})$ is coherent and $R = I \oplus NC(P_{I,R})$.

Models

- $B \subseteq U$ is a *model* of (or *satisfies*) a literal $\mathbf{in}(a)$ if $a \in B$.
  $B \subseteq U$ is a *model* of (or *satisfies*) a literal $\mathbf{out}(a)$ if $a \notin B$.

- $B \subseteq U$ is a *model* of (or *satisfies*) a rule of the form $\alpha \leftarrow \alpha_1, \ldots, \alpha_n$ if either B is not a model of at least one literal α_i, or B is a model of α.

- $B \subseteq U$ is a *model* of (or *satisfies*) a revision program P if B is a model of every rule in P.

Basic properties

- P-justified revisions are models of P;
- justified revisions of a database differ from the database by as little as possible;
- if the current database satisfies the revision program, then no nontrivial change is justified;
- additional evidence does not destroy justified revisions;
- the problem of existence of P-justified revisions is NP-complete.

Stable models of logic programs

A logic program is a set of clauses of the form

$$p \leftarrow q_1, \ldots, q_m, not\ s_1, \ldots, not\ s_n$$

P - a logic program. Reduct of P relative to M, P^M, is obtained from P by

- removing all clauses which contain "*not q*" such that q is true in M,
- deleting all negative premises "*not q*" from the remaining clauses.

P^M has a unique least model $Least(P^M)$.

M is a *stable model* of P if $M = Least(P^M)$.

Relation to logic programming (LP)

Logic program clauses correspond to revision in-rules.

For a logic program clause $c = p \leftarrow q_1, \ldots, q_m, not\ s_1, \ldots, not\ s_n$,

$$\text{in-rule} \quad r = \mathbf{in}(p) \leftarrow \mathbf{in}(q_1), \ldots, \mathbf{in}(q_m), \mathbf{out}(s_1), \ldots, \mathbf{out}(s_n)$$

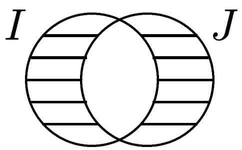
We define $rp(c) = r, \quad lp(r) = c.$

For a logic program P define $rp(P) = \{rp(c) : c \in P\}$.

For a revision program P define $lp(P) = \{lp(r) : r \in P\}$.

Theorem. *A set of atoms X is a stable model of a logic program P if and only if X is an $rp(P)$-justified revision of $\emptyset$.*

Shifting



$$W = I \div J = (I \setminus J) \cup (J \setminus I)$$

W - a set of atoms that change status.
Define a *W-transformation (shift)* as follows.

For a literal α $\quad(\alpha = \mathbf{in}(a)$ or $\alpha = \mathbf{out}(a))$, $\quad T_W(\alpha) = \begin{cases} \alpha^D, & \text{when } a \in W \\ \alpha, & \text{when } a \notin W. \end{cases}$

For a set of literals L, $\quad T_W(L) = \{T_W(\alpha) \colon \alpha \in L\}$.

For a set of atoms X,

$$T_W(X) = \{\ a :\ \mathbf{in}(a) \in T_W(\{\mathbf{in}(b) \colon b \in X\} \cup \{\mathbf{out}(b)\ \mathit{colonb} \notin X\})\}.$$

For a revision program P, $\quad T_W(P)$ is obtained from P by applying T_W to each literal in P.

Shifting theorem

Let I_1, I_2 be databases. Let $W = I_1 \div I_2$ be their symmetric difference.
Then $T_W(I_1) = I_2$.

Theorem. *Let P be a revision program. Let I_1 and I_2 be databases. Let $W = I_1 \div I_2$. Then, a database R is a P-justified revision of I_1 if and only if $T_W(R)$ is a $T_W(P)$-justified revision of I_2.*

Corollary. *For each I and R, R is P-justified revision of I if and only if $T_I(R)$ is $T_I(P)$-justified revision of $\emptyset$.*

Example (shifting)

$$
\begin{array}{rrcl}
P: & \mathbf{out}(Ann) & \leftarrow & \mathbf{in}(Bob) \\
 & \mathbf{in}(Tom) & \leftarrow & \mathbf{out}(Bob) \\
 & \mathbf{in}(David) & \leftarrow & \mathbf{in}(Tom)
\end{array}
$$

Let $I = \{Ann, Bob, David\}$.
The only P-justified revision of I is $R = \{Bob, David\}$.

$$
\begin{array}{rrcl}
T_I(P): & \mathbf{in}(Ann) & \leftarrow & \mathbf{out}(Bob) \\
 & \mathbf{in}(Tom) & \leftarrow & \mathbf{in}(Bob) \\
 & \mathbf{out}(David) & \leftarrow & \mathbf{in}(Tom)
\end{array}
$$

$T_I(I) = \emptyset$. The only justified revision of $\emptyset$ is $\{Ann\}$.

Observe that $\{Ann\} = T_I(\{Bob, David\})$.

Revision programming = logic programming + constraints

$InRules(P)$ - a set of all in-rules of P

Theorem. *Let P be a revision program. Then, R is a P-justified revision of $\emptyset$ if and only if R is a $InRules(P)$-justified revision of $\emptyset$ and R is a model of $P \setminus InRules(P)$.*

Corollary. *Let P be a revision program. Then, R is a P-justified revision of $\emptyset$ if and only if R is a stable model of $lp(InRules(P))$ and R is a model of $P \setminus InRules(P)$.*

Computing justified revisions

by means of LP

$$\boxed{\begin{array}{c} P \\ I \end{array}} \longrightarrow \boxed{\begin{array}{c} P_1 = T_I(P) \\ \emptyset \end{array}} \longrightarrow \boxed{\begin{array}{c} P' = InRules(P_1) \\ P'' = P_1 \setminus P' \end{array}} \longrightarrow$$

$$\longrightarrow \boxed{\begin{array}{c} lp(P') \\ constraints\ (P'') \end{array}} \longrightarrow \boxed{\begin{array}{c} X\ -\ stable\ model\ of\ lp(P') \\ X\ \ satisfies\ \ P'' \end{array}} \longrightarrow$$

$$\longrightarrow \boxed{\begin{array}{c} X\ -\ P_1\text{-}just.\ \ rev.\ \ of \\ \emptyset \end{array}} \longrightarrow \boxed{\begin{array}{c} T_I(X)\ -\ P\text{-}just.\ \ rev.\ \ of \\ I \end{array}}$$

Graph 3-colorability

Problem: Find a coloring that differs minimally from the initial coloring and satisfies the following condition. Any two vertices that are connected by an edge should have different colors.

Graph description: $vtx(a), \ldots, edg(a, b), \ldots, color(red), \ldots$.

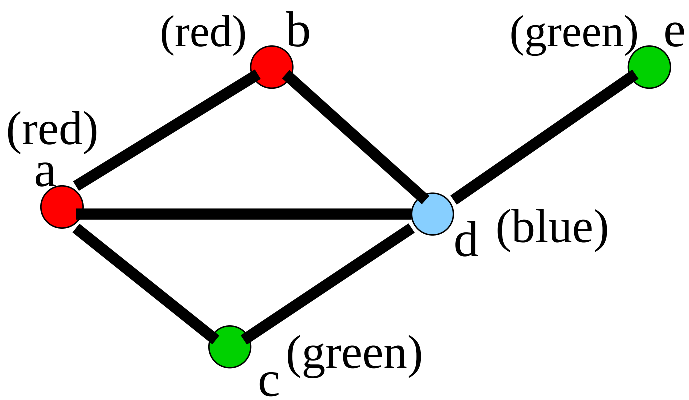
Revision program: (constraints)

$$
\begin{array}{rcl}
out(clr(Y, C)) & \leftarrow & color(C),\ edg(X, Y),\ in(clr(X, C)). \\
out(clr(X, C)) & \leftarrow & color(C),\ edg(X, Y),\ in(clr(Y, C)). \\
in(clr(X, red)) & \leftarrow & vtx(X),\ out(clr(X, green)),\ out(clr(X, blue)). \\
in(clr(X, green)) & \leftarrow & vtx(X),\ out(clr(X, red)),\ out(clr(X, blue)). \\
in(clr(X, blue)) & \leftarrow & vtx(X),\ out(clr(X, red)),\ out(clr(X, green)).
\end{array}
$$

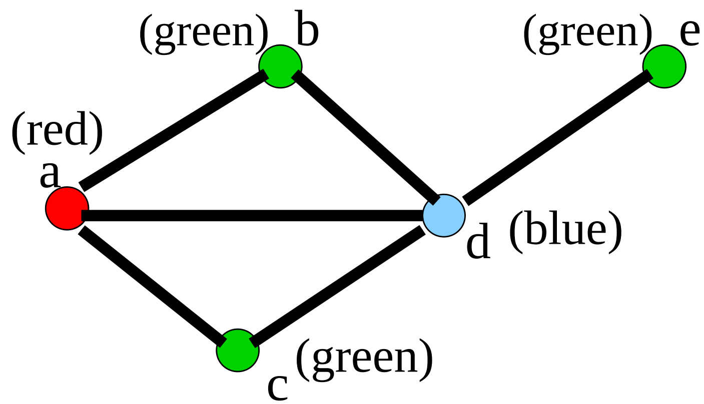
Initial database: $in(clr(a, red)), in(clr(b, red)), in(clr(c, green)), \ldots$.

Example

Initial database:



Revision:




```
% conv coloring | lparse -d none | smodels | convback
smodels version 2.8. Reading...done
Answer: 1
Revision: clr(b,green) clr(a,red) clr(c,green) clr(d,blue) clr(e,green)
True
%
```

Extensions

Disjunctive revision programs (DRP)

Rules are not definite:

$$\begin{aligned} \mathbf{in}(Ann) \mid \mathbf{out}(Tom) \mid \mathbf{in}(Bob) &\leftarrow \ \mathbf{in}(\text{David}) \\ \mathbf{in}(Tom) \mid \mathbf{out}(David) &\leftarrow \ \mathbf{in}(Ann),\ \ \mathbf{out}(Bob) \\ \mathbf{out}(Ann) &\leftarrow \mathbf{in}(David) \\ \mathbf{out}(David) &\leftarrow \mathbf{in}(Bob) \end{aligned}$$

Initial database: $I = \{David, Tom\}$. Revision: R – ?

Same key concepts: reduct, necessary change.

Justified revisions for DRP

P - a disjunctive revision program, I and R - databases.

A *reduct* of P with respect to (I, R) (denoted by $P^{I,R}$) is defined in four steps:

1. Eliminate from P every rule whose body is not satisfied by R.
2. From the body of each remaining rule eliminate each literal that is satisfied by I.
3. Remove all rules r, such that $head(r) \cap I(I, R) \neq \emptyset$.
4. Remove from the heads of the rules all literals that contradict R.

A database R is a *P-justified revision* of a database I if for some necessary change L of $P^{I,R}$, L is coherent and $R = I \oplus L$.

Example (DRP)

P:

$$\begin{aligned} in(Ann) \mid in(Bob) &\leftarrow \\ out(Tom) \mid in(David) &\leftarrow \\ out(Ann) &\leftarrow \quad in(David) \\ out(David) &\leftarrow \quad in(Bob) \end{aligned}$$

$I = \{Ann, Tom\}$, $\quad R = \{Ann\}$.

Inertia $I(I, R) = \{in(Ann), out(Bob), out(David)\}$.

The reduct $P^{I,R} = \{out(Tom) \leftarrow\}$.

$NC(P^{I,R}) = \{out(Tom)\}$ - coherent. $\qquad R = I \oplus NC(P^{I,R})$.

Therefore, R is a P-justified revision of I.

DRP: properties

- The semantics of disjunctive revision programming reduces to the semantics of justified revisions on DRPs consisting of rules with a single literal in the head.
- The shifting theorem generalizes to the case of DRPs.
- The semantics of disjunctive revision programming over the empty initial database reduces to the Lifschitz and Woo semantics for General Disjunctive Logic Programs (GDLP).

Dealing with uncertainty:

Annotated revision programs

Example

A group of experts is about to discuss and then vote whether to accept or reject a proposal.

Each person has an opinion on the proposal that may be changed during the discussion as follows:

- any person can convince an optimist to vote for the proposal,

- any person can convince a pessimist to vote against the proposal.

The group consists of two optimists (Ann and Bob) and one pessimist (Pete). Given everybody's opinion on the subject before the discussion, what are possible outcomes of the vote?

Example, cont'd

$$
\begin{array}{rcl}
(\mathbf{in}(\mathit{accept}){:}\{\mathit{Bob}\}) & \leftarrow & (\mathbf{in}(\mathit{accept}){:}\{\mathit{Ann}\}) \\
(\mathbf{in}(\mathit{accept}){:}\{\mathit{Bob}\}) & \leftarrow & (\mathbf{in}(\mathit{accept}){:}\{\mathit{Pete}\}) \\
(\mathbf{in}(\mathit{accept}){:}\{\mathit{Ann}\}) & \leftarrow & (\mathbf{in}(\mathit{accept}){:}\{\mathit{Bob}\}) \\
(\mathbf{in}(\mathit{accept}){:}\{\mathit{Ann}\}) & \leftarrow & (\mathbf{in}(\mathit{accept}){:}\{\mathit{Pete}\}) \\
(\mathbf{out}(\mathit{accept}){:}\{\mathit{Pete}\}) & \leftarrow & (\mathbf{out}(\mathit{accept}){:}\{\mathit{Ann}\}) \\
(\mathbf{out}(\mathit{accept}){:}\{\mathit{Pete}\}) & \leftarrow & (\mathbf{out}(\mathit{accept}){:}\{\mathit{Bob}\})
\end{array}
$$

Initially: $B_I(\mathit{accept}) = \langle\{\mathit{Pete}\}, \{\mathit{Bob}\}\rangle.$

(Pete is for. Bob is against. Ann is indifferent.)

Revisions:

1. $B_R(\mathit{accept}) = \langle\{\mathit{Ann}, \mathit{Bob}, \mathit{Pete}\}, \{\}\rangle$ (All are for.)
2. $B'_R(\mathit{accept}) = \langle\{\}, \{\mathit{Bob}, \mathit{Pete}\}\rangle$
   (Bob and Pete are against. Ann remains indifferent.)

Example

Two sources of light: a and b.
Dust in the air, light pollution.

$$
\begin{array}{rcl}
(\mathbf{in}(a){:}1) & \leftarrow & (\mathbf{in}(a){:}0.8), (\mathbf{out}(b){:}0.6) \\
(\mathbf{out}(b){:}1) & \leftarrow & (\mathbf{in}(a){:}0.8), (\mathbf{out}(b){:}0.6) \\
(\mathbf{in}(b){:}1) & \leftarrow & (\mathbf{in}(b){:}0.8), (\mathbf{out}(a){:}0.6) \\
(\mathbf{out}(a){:}1) & \leftarrow & (\mathbf{in}(b){:}0.8), (\mathbf{out}(a){:}0.6)
\end{array}
$$

Observed brightness: $B_I(a) = <0.3, 0.7>$ and $B_I(b) = <0.9, 0.1>$.
Revision (actual brightness): $B_R(a) = <0, 1>$, and $B_R(b) = <1, 0>$.

Preliminaries

- $\mathcal{T}$ - a complete infinitely distributive lattice with a *de Morgan complement* (denoted by $^-$) It satisfies de Morgan laws

$$\overline{a \vee b} = \overline{a} \wedge \overline{b}, \qquad \overline{a \wedge b} = \overline{a} \vee \overline{b}$$

- $(\mathbf{in}(b){:}\alpha)$, $(\mathbf{out}(b){:}\alpha)$ - *annotated revision atoms* ($\alpha \in \mathcal{T}$, $b \in U$).
- *Annotated revision rules*:

$$p \leftarrow q_1, \ldots, q_n,$$

  where p, q_i $(1 \leq i \leq n)$ are annotated revision atoms.
- *Annotated revision program* is a set of annotated revision rules.

$\mathcal{T}$-valuation

$\mathcal{T}$*-valuation* is a mapping v from revision atoms to $\mathcal{T}$

v *satisfies* $(\mathbf{in}(b){:}\alpha)$ if $v(\mathbf{in}(b)) \geq \alpha$

v *satisfies* $(\mathbf{out}(b){:}\alpha)$ if $v(\mathbf{out}(b)) \geq \alpha$

$t_{\mathcal{P}}(v)$ - the set of all annotated revision atoms that occur as the head of a rule in $\mathcal{P}$ whose body is satisfied by v

Operator on $\mathcal{T}$-valuations:

$$T_{\mathcal{P}}(v)(l) = \bigvee\{\alpha \mid (l{:}\alpha) \in t_{\mathcal{P}}(v)\}$$

$\mathcal{T}^2$-valuation

- $\mathcal{T}^2$ - complete, infinitely distributive lattice:
  - Domain: $\mathcal{T} \times \mathcal{T}$
  - $\langle \alpha_1, \beta_1 \rangle \leq_k \langle \alpha_2, \beta_2 \rangle$ if $\alpha_1 \leq \alpha_2$ and $\beta_1 \leq \beta_2$
  - $\otimes$, $\oplus$ - meet and join under $\leq_k$
  - Conflation: $-\langle \alpha, \beta \rangle = \langle \bar{\beta}, \bar{\alpha} \rangle$
- An element $A \in \mathcal{T}^2$ is *consistent* if $A \leq_k -A$.

Example: $\mathcal{T} = \{\emptyset, \{p\}, \{q\}, \{p, q\}\}$

$\langle \{p, q\}, \{p\} \rangle$ - inconsistent $-\langle \{p, q\}, \{p\} \rangle = \langle \{q\}, \{\} \rangle$

$\langle \{\}, \{q\} \rangle$ - consistent $-\langle \{\}, \{q\} \rangle = \langle \{p\}, \{p, q\} \rangle$

$\mathcal{T}^2$-valuation, cont'd

- $\mathcal{T}^2$-valuations are used to represent databases
- Let v be a $\mathcal{T}$-valuation. Then, $\mathcal{T}^2$-valuation $\theta(v)$ is defined as
  $\theta(v)(b) = \langle \alpha, \beta \rangle$, where $v(in(b)) = \alpha$ and $v(out(b)) = \beta$
- Operator on $\mathcal{T}^2$-valuations:

$$T^b_{\mathcal{P}} = \theta \circ T_{\mathcal{P}} \circ \theta^{-1}$$

Models and c-models (example)

Let $\mathcal{T} = \{\emptyset, \{p\}, \{q\}, \{p, q\}\}$.

$P = \{ \;\; (\mathbf{in}(a){:}\{q\}) \leftarrow (\mathbf{out}(a){:}\{p\}) \;\; \}$.

$B_1(a) = \langle \{p, q\}, \{p\} \rangle$

$B_2(a) = \langle \{p, q\}, \{p, q\} \rangle$

Explicit evidence provided by P for $\mathbf{in}(a)$ is $\{q\}$, for $\mathbf{out}(a)$ is $\{\}$.

Implicit bound provided by P for $\mathbf{out}(a)$ is $\overline{\{q\}} = \{p\}$,

for $\mathbf{in}(a)$ is $\overline{\{\}} = \{p, q\}$.

Both B_1 and B_2 are models of P. However, only B_1 agrees with the implicit bounds imposed by P.

c-models

Let P be an annotated revision program.

Theorem. *A $\mathcal{T}^2$-valuation B is a model of P if and only if $B \geq_k T_P^b(B)$.*

Definition 1 *Let B be a $\mathcal{T}^2$-valuation. We say B is a* c-model *of P if*

$$T_P^b(B) \leq_k B \leq_k T_P^b(B) \oplus (-T_P^b(B)).$$

Theorem. *A consistent $\mathcal{T}^2$-valuation B is a c-model of P if and only if B is a model of P.*

Justified revisions - goals

- Extend the semantics of justified revisions in the original case so that key properties are preserved.
- Satisfy the *invariance under join* principle: replacing rule

$$r = \quad \ldots \leftarrow \ldots, (l{:}\beta_1), \ldots, (l{:}\beta_2), \ldots$$

by rule

$$r' = \quad \ldots \leftarrow \ldots, (l{:}\beta_1 \vee \beta_2), \ldots$$

in annotated revision program should have no effect on justified revisions.

Justified revision - definitions

B_I - initial knowledge, P - annotated revision program.

For $\alpha, \beta \in \mathcal{T}$, $\quad pcomp(\alpha, \beta)$ denotes the least γ such that $\alpha \vee \gamma \geq \beta$.

Necessary change of P (denoted as NC(P)) is the least fixpoint of the operator T_P^b.

Result of applying a change C to a database B_I is defined as

$$B_R = (B_I \otimes -C) \oplus C.$$

Definition of justified revision

Definition 2 *The* reduct $P_{B_R}|B_I$ *is obtained from* P *by*

1. *removing every rule whose body contains an annotated atom that is not satisfied in* B_R*,*
2. *replacing each annotated atom* $(l{:}\beta)$ *from the body of each remaining rule by the annotated atom* $(l{:}\gamma)$*, where* $\gamma = pcomp((\theta^{-1}(B_I))(l), \beta)$*.*

Definition 3 B_R *is a* P-justified revision *of* B_I *if* $B_R = (B_I \otimes -C) \oplus C$*, where* $C = NC(P_{B_R}|B_I)$ *is the necessary change for* $P_{B_R}|B_I$*.*

Properties of justified revisions

Theorem. *Let P be an annotated revision program and let B_I and B_R be $\mathcal{T}^2$-valuations. If B_R is a P-justified revision of B_I then B_R is a c-model of P (and, hence, also a model of P).*

Theorem. *Let a $\mathcal{T}^2$-valuation B_I be a model of an annotated revision program P. Then, B_I is a P-justified revision of itself if and only if B_I is a c-model of P.*

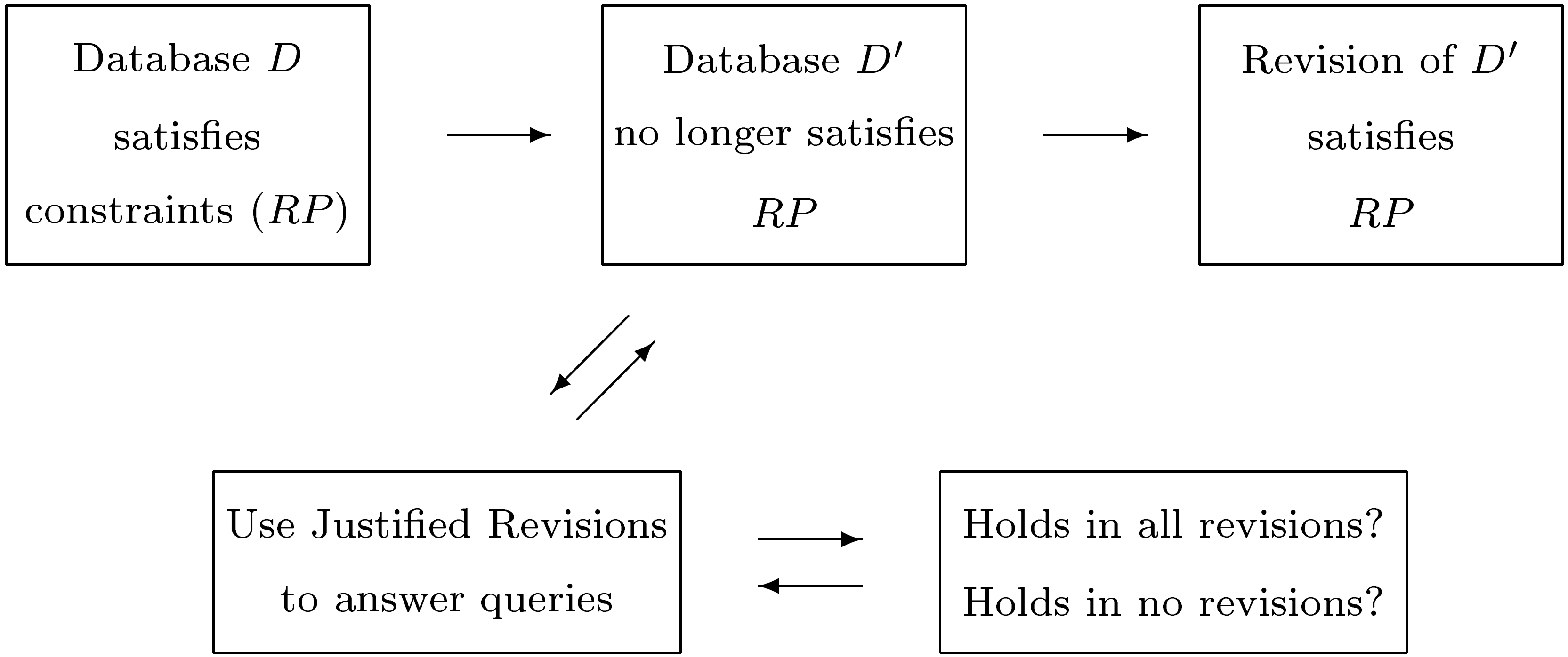
More properties of justified revisions

Theorem. *Let B_I be a model of an annotated revision program P. Let B_R be a P-justified revision of B_I. Then, $B_R \leq_k B_I$.*

Theorem. *Let B_I be a consistent model of an annotated revision program P. Then, B_I is the only P-justified revision of itself.*

A database scenario

- RP models integrity constraints
- Two tasks: how to fix, how to reason (without fixing)



Current and future work

- Well-founded semantics.
- Database connection (database repairs).
- Iterated revisions.
- 'Industrial grade' implementation?