COBA 2.0 Manual

Daphne Liu <daphnel@sfu.ca>
Sven Thiele <sthiele@rz.uni-potsdam.de>

April 8, 2005

1 Introduction

COBA 2.0 is a Java implementation of the consistency-based approach for belief change invented by James P. Delgrande and Torsten Schaub. This approach presents a framework within which different operators for the revision, contraction and merging of knowledge bases are defined. With COBA, you can construct any belief change scenario that entails a combination of merging, revision, and contraction and compute the resultant knowledge base.

2 Getting Started

To compile and run COBA, you need to have Java VM version 1.4 or higher installed on your machine.

2.1 Compiling the Source Code

Once you have the Java VM version 1.4 or higher installed, simply download and unpack the archive `../coba2.tar.gz`. You will find inside the unpacked archive several scripts readily available for compilation and execution purposes. You may re-configure the paths in the scripts for your local machine.

To recompile COBA, you can use the included `Makefile` with two targets:

- `make app` : to compile the command-line version and build the JAR archive `applet/cobaapp.jar` to store the compiled command-line version.
- `make applet` : to compile the applet and build the JAR archive `applet/cobaUI.jar` to store the compiled applet version.

To run COBA, you can either start the command-line version with: `run_app KB_File(or none) Revisio`
or start the applet version with: `run_applet`

`run_app` is a script that starts the executable `cobaapp.jar` - the compiled command-line version of COBA. Please ensure that the path to java and the path to `cobaapp.jar` are set correctly for your machine.

`run_applet` is a script that calls the appletviewer to start the JAR archive `cobaUI.jar` - the compiled applet version of COBA - with the security policy policy/revision.policy. Please ensure that the path to appletviewer and the path to `cobaUI.jar` are set correctly for your machine.

2.2 The COBA Command-line Tool

The command-line version of COBA presents an interactive menu with which you can specify tasks to perform. The default type of belief change is skeptical change; the default type of merging, basic merge; and the default equivalence set search algorithm, AllDetEQ2.

2.2.1 Running the Command-line Tool Script

You may start the command-line tool by running the script `run_app` with three arguments: `run_app KB_File(or none) Revision_File(or none) Contraction_File(or none)`

If any one of the three arguments is not used, input "none" without the double quotes instead.

The first argument `KB_File` is the path of the file containing the sentences of the knowledge base(s). Each knowledge base should be preceded by a line "KB :" without the double quotes; each sentence should be listed on a separate line.

The second argument `Revision_File` is the path of the file containing the sentences for revision. Each sentence should be listed on a separate line. A single revisor will be constructed as a conjunction, by taking each sentence as a conjunct.

The third argument `Contraction_File` is the path of the file containing the sentences for contraction. Each line is interpreted as an independent contraction sentence. Consequently, while `(p&~q)` means that `(p&~q)` is to be obliterated from the consequences of the resultant knowledge base,

```
p
~q
```

listed on two seperate lines means that both `p` and `~q` are to be obliterated from the consequences of the resultant knowledge base.

2.2.2 Using the COBA Command-line Tool

The COBA command-line tool provides an interactive menu of tasks to perform:

1. Specify the path of the KB input file.
   You will be prompted for the path of the KB input file you want to load.

2. Specify the path of the revision input file.
   You will be prompted for the path of the revision input file you want to load.

3. Specify the path of the contraction input file.
   You will be prompted for the path of the contraction input file you want to load.

4. Specify the path of the entailment-based integrity constraints input file.
   You will be prompted for the path of the entailment-based integrity constraints input file you want to load.

5. Specify the path of the consistency-based integrity constraints input file.
   You will be prompted for the path of the consistency-based integrity constraints input file you want to load.

6. Set the type of change (default=skeptical change).
   You will be prompted for the belief change type (skeptical change or choice change).

7. Set the equivalence set search algorithm (default=AllDetEQ2).
   You will be prompted for the search algorithm (AllDetEQ1 or AllDetEQ2).

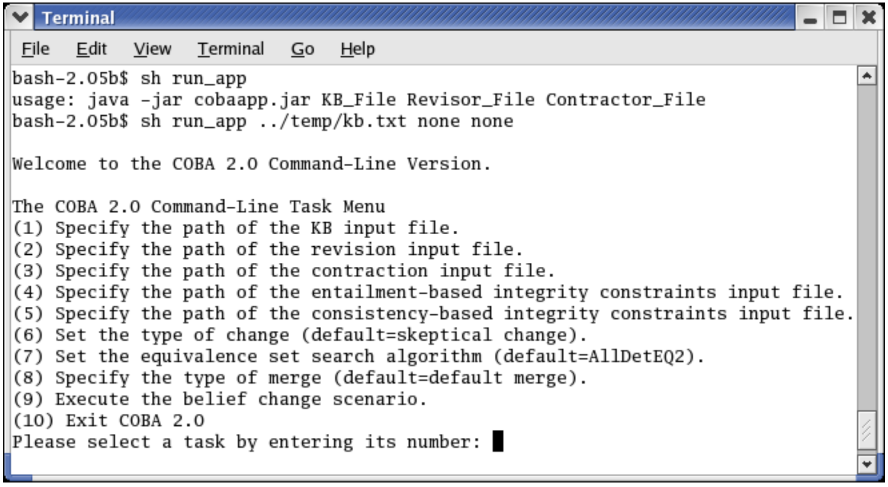
8. Specify the type of merge (default=default merge).
   You will be prompted for the merge type (default basic merge or projection merge).

9. Execute the belief change scenario.
   You will see the resultant knowledge base(s) for the current belief change scenario.

10. Exit COBA 2.0
    You will terminate and exit the COBA command-line tool.

For tasks (1) - (5) and (9), you may see error messages if any input sentence is syntactically malformed, or if there is an inconsistency error. You should fix the error(s) and try to perform the task again.

To select a specific task from the menu, simply enter its number at the prompt.



```
bash-2.05b$ sh run_app
usage: java -jar cobaapp.jar KB_File Revisor_File Contractor_File
bash-2.05b$ sh run_app ../temp/kb.txt none none

Welcome to the COBA 2.0 Command-Line Version.

The COBA 2.0 Command-Line Task Menu
(1) Specify the path of the KB input file.
(2) Specify the path of the revision input file.
(3) Specify the path of the contraction input file.
(4) Specify the path of the entailment-based integrity constraints input file.
(5) Specify the path of the consistency-based integrity constraints input file.
(6) Set the type of change (default=skeptical change).
(7) Set the equivalence set search algorithm (default=AllDetEQ2).
(8) Specify the type of merge (default=default merge).
(9) Execute the belief change scenario.
(10) Exit COBA 2.0
Please select a task by entering its number: 
```

3 Using the COBA Applet

The applet version of COBA offers several additional and distinct advantages over the command-line tool. Specifically, you may

1. add input from your local files or to type in the input yourself.

2. view the history logs for the changes made to the knowledge base (KB) list, the entailment-based integrity constraints (EB IC) list, and the consistency-based integrity constraints (CB IC) list.

3. revert to an older KB, EB IC, or CB IC snapshot.

4. save any list to an output file.

5. view sentences in the default, CNF, or DNF format.

6. turn off consistency checks on an entry and its mutual consistency with its list.

7. reject or commit a resultant knowledge base.

8. view the COBA user manual and JavaDocs in external browser windows (if the applet is running in an html document).

You may start the COBA applet with the script `run_applet.`

The COBA applet needs your permission to read and create files on your local machine. You must explicitly grant this permission to `cobaUI.jar`, if you wish to import files into the applet or save the applet output to files. You may do so by

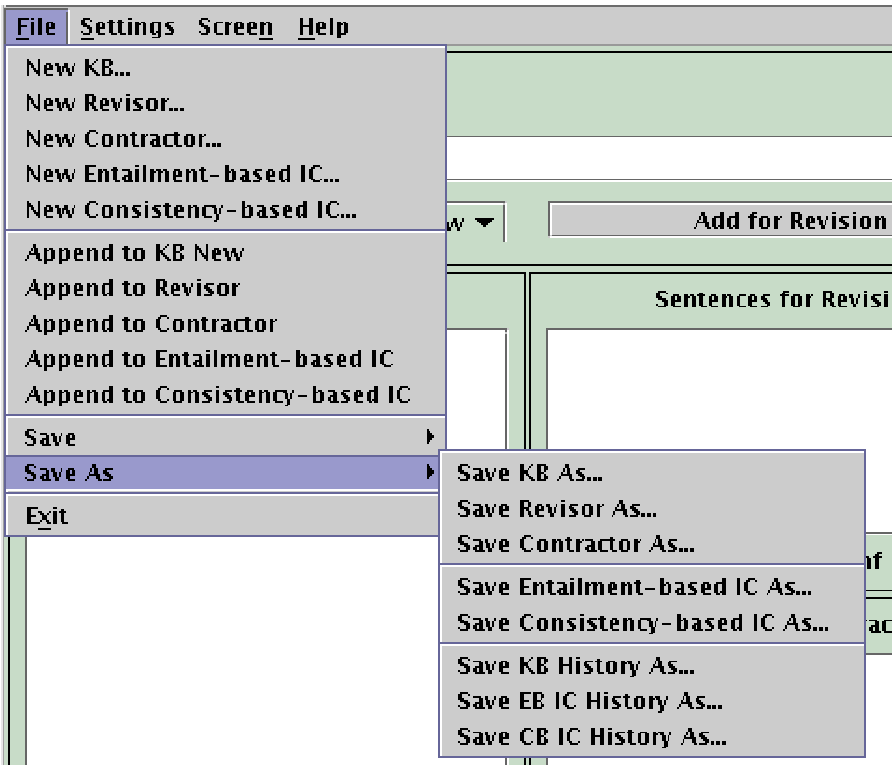
- either running the default `run_applet` script with the policy file policy/revision.policy.

- or modifying the codeBase path and copying the policy in policy/revision.policy to your local .java.policy file. See Sun's Web page at `http://java.sun.com/j2se/1.4.2/docs/guide/security/PolicyFiles.html#DefaultLocs` for the default location of your .java.policy file.

3.1 Using the Applet Menu

The COBA applet has a menu bar of four items: File, Settings, and Screen. You will learn about each of these four menus in the following subsections.

3.1.1 File Menu



- New KB...
  If the KB list is non-empty, you will be presented with a confirmation dialog about your decision to replace the current knowledge base(s).
  You will then be asked to select an input file to import as the new knowledge base(s).

- Revisor...
  If the revision list is non-empty, you will be presented with a confirmation dialog about your decision to replace the current sentences for revision.
  You will then be asked to select an input file to import as the new sentences for revision.

- New Contractor...
  If the contraction list is non-empty, you will be presented with a confirmation dialog about your decision to replace the current sentences for contraction.
  You will then be asked to select an input file to import as the new sentences for contraction.

- New Entailment-based IC...
  If the EB IC list is non-empty, you will be presented with a confirmation dialog about your decision to replace the current EB IC(s).

You will then be asked to select an input file to import as the new EB ICs.

- New Consistency-based IC...
  If the CB IC list is non-empty, you will be presented with a confirmation dialog about your decision to replace the current CB IC(s).
  You will then be asked to select an input file to import as the new CB ICs.

==

- Append to KB Selected
  You will be asked to select an input file to append to the selected knolwedge base (KB).
  In other words, the sentences in the first KB in the input file will be added to the selected KB, and the sentences in the second KB in the input file will be added to KB immediately after the selected KB, and so on.
  The indexing of the sentences in a knowledge base starts at 0.
  The indexing of knowledge bases starts at 0.
  The selected knowledge base is the one whose index appears at the top of the drop-down list next to the "Add to KB" button on the main screen.

- Append to Revisor
  You will be asked to select an input file to append to current revision list.

- Append to Contractor
  You will be asked to select an input file to append to current contraction list.

- Append to Entailment-based IC
  You will be asked to select an input file to append to current list of entailment-based integrity constraints.

- Append to Consistency-based IC
  You will be asked to select an input file to append to current list of consistency-based integrity constraints.

==

Save −>
You can save any one of the eight lists to the most recent file you last saved this list to.
If such a file does not already exist, you will be prompted for the name of a new output file.

The output file format for any one of the three history lists is as follows:

1. The snapshots before the changes in chronological order, followed by

2. The line "***THE FOLLOWING ARE THE CORRESPONDING CHANGES.***" (without the double quotes), followed by

3. The corresponding changes for the snapshots in chronological order.

- KB
  You will save the current list of knowledge base(s).

- Revisor
  You will save the current revision list.

- Contractor
  You will save the current contraction list.

- Entailment-based IC
  You will save the current list of entailment-based integrity constraints.

- Consistency-based IC
  You will save the current list of consistency-based integrity constraints.

- KB History
  You will save the list of changes made to the knowledge base(s).

- EB IC History
  You will save the list of changes made to the entailment-based integrity constraints.

- CB IC History
  You will save the list of changes made to the consistency-based integrity constraints.

- All
  You will save all of the eight lists above.

==

Save As ->
You will be prompted for the name of the output file to which you can save one of the eight lists.

The output file format for any one of the three history lists is as follows:

1. The snapshots before the changes in chronological order, followed by

2. The line "***THE FOLLOWING ARE THE CORRESPONDING CHANGES.***" (without the double quotes), followed by

3. The corresponding changes for the snapshots in chronological order.

- Save KB As
  You will save the current list of knowledge base(s).

- Save Revisor As
  You will save the current revision list.

- Save Contractor As
  You will save the current contraction list.

- Save Entailment-based IC As
  You will save the current list of entailment-based integrity constraints.

- Save Consistency-based IC As
  You will save the current list of consistency-based integrity constraints.

- Save KB History As
  You will save the list of changes made to the knowledge base(s).

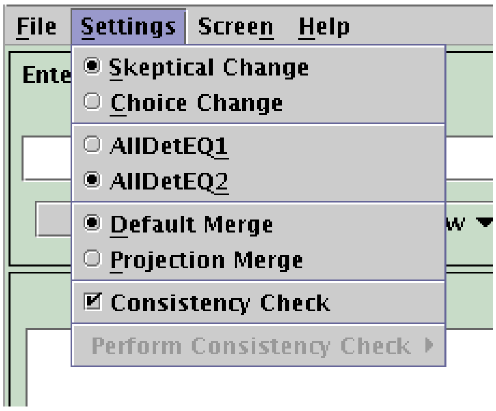
- Save EB IC History As
  You will save the list of changes made to the entailment-based integrity constraints.

- Save CB IC History As
  You will save the list of changes made to the consistency-based integrity constraints.

==

- Exit
  You can terminate the applet and close the enclosing browser which displays the applet.

3.1.2 Settings Menu



- Skeptical Change
  You can click this item to change the type of belief change to skeptical change.

- Choice Change
  You can click this item to change the type of belief change to choice change.

==

- AllDetEQ1
  You can click this item to change the search algorithm for finding equivalence sets to AllDetEQ1 as described in the COBA paper by James Delgrande and Torsten Schaub.

- AllDetEQ2
  You can click this item to change the search algorithm for finding equivalence sets to AllDetEQ2, which is more efficient as described in the COBA paper by James Delgrande and Torsten Schaub.

==

- Default Merge
  You can click this item to change the type of merge to default/basic merge.

- Projection Merge
  You can click this item to change the type of merge to projection merge.

==

- Consistency Check
  You can click this item to turn the default consistency check on or off.
  A checked box indicates that the default consistency check is on, which means that a consistency check will be performed automatically every time a sentence is added to the selected KB, the revision list, or the entailment-based integrity constraints list.
  An unchecked box means that this default consistency check will not be performed automatically.

==

Perform Consistency Check –>
This item is enabled once you have switched the default consistency check off. The reason is that with default consistency check on, a consistency check is automatically performed every time a sentence is added to the selected KB, the revision list, or the entailment-based integrity constraints list.

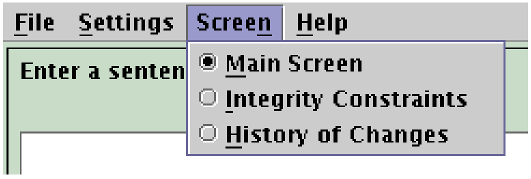
- KB Selected
  If there is any inconsistencies or mutual inconsistencies in the selected knowledge base, you will be advised of them in a dialog

window.
The indexing of the sentences for the selected knowledge base starts at 0.
The selected knowledge base is the one whose index appears at the top of the drop-down list next to the "Add to KB" button on the main screen.
The indexing of knowledge bases starts at 0.

- Revision List
  If there is any inconsistencies or mutual inconsistencies among the sentences in the revision list, you will be advised of them in a dialog window.
  The indexing of the sentences for revision starts at 0.

- EB IC List
  If there is any inconsistencies or mutual inconsistencies among the entailment-based integrity constraints, you will be advised of them in a dialog window.
  The indexing of the entailment-based integrity constraints starts at 0.

3.1.3 Screen Menu



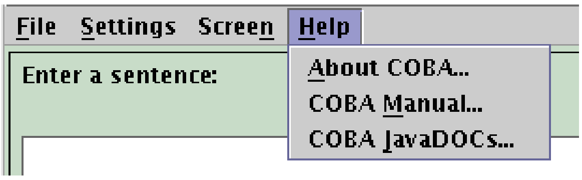
- Main Screen
  You will be presented with the main user interface through which you may:
  1. add sentences to, or remove sentences from, the knowledge base(s), the revision list, and the contraction list;
  2. change the type of belief change, the search algorithm for finding equivalence sets, and the type of merging; and
  3. carry out parallel change on a belief change scenario and subsequently reject or commit a resultant knowledge base.

- Integrity Constraints
  You will be presented with the integrity constraints (IC) interface through which you may add sentences to, or remove sentences from, the entailment-based IC list and the consisteny-based IC list.

- History of Changes
  You will be presented with the history of changes interface through which you may revert to an older knowledge base snapshot, as

well as view the log of changes made to the knowledge base list, the entailment-based IC list, and the consistency-based IC list.

3.1.4 Help Menu



- About COBA...
  You will find information on the authors and implementers of COBA 2.0 in a dialog window.

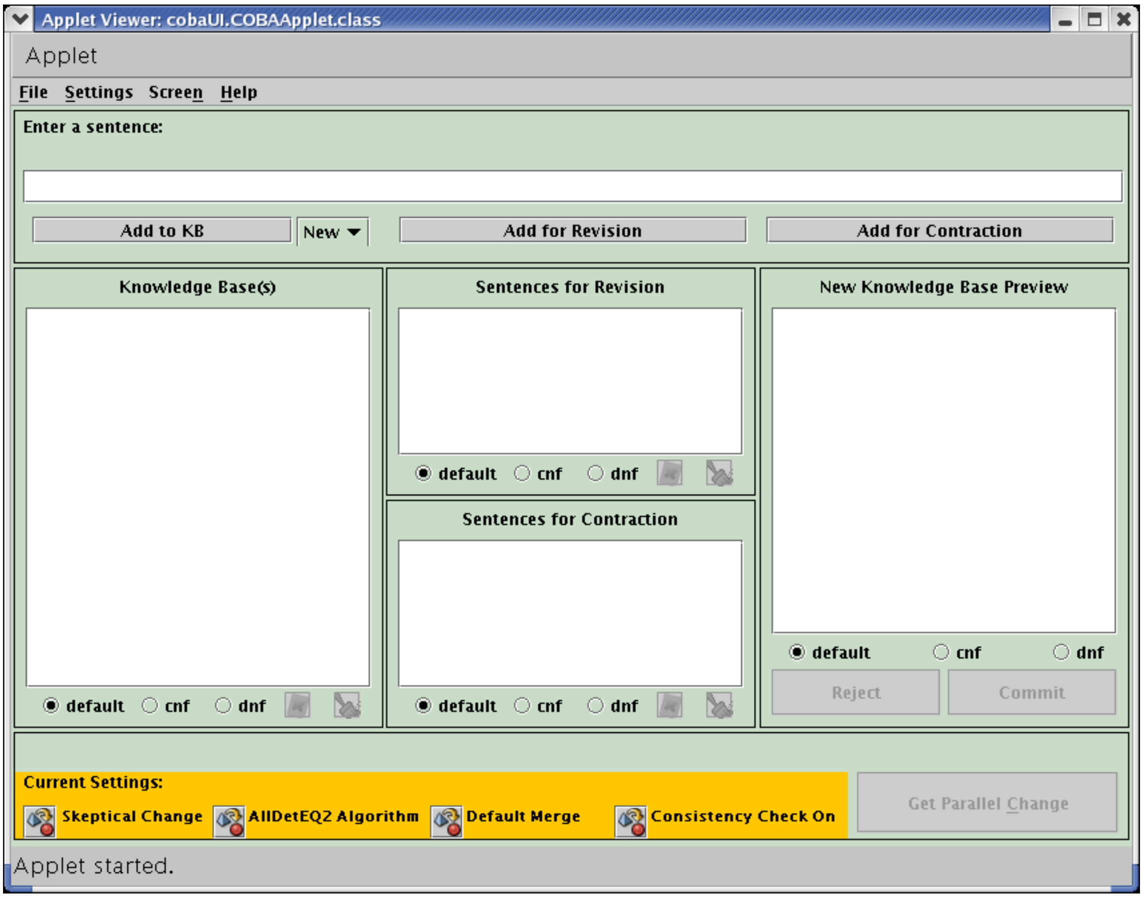
- COBA Manual...
  You will see the COBA 2.0 user manual in a new browser window. This function is disabled if you are not running the COBA applet in a Web browser.

- COBA JavaDOCs...
  You will see the COBA 2.0 JavaDOCs in a new browser window. This function is disabled if you are not running the COBA applet in a Web browser.

3.2 The Main Screen

The first screen is the main user interface through which you may:

1. add sentences to, or remove sentences from, the knowledge base(s), the revision list, and the contraction list;
2. change the settings for the type of belief change, the search algorithm for equivalence sets, and the type of merging; and
3. carry out parallel change on a belief change scenario, and subsequently reject or commit a resultant knowledge base.



Current Settings:
Skeptical Change
AllDetEQ2 Algorithm
Default Merge
Consistency Check On

Applet started.

The selected knowledge base (KB) is the one whose index appears at the top of the drop-down list next to the "Add to KB" button on the main screen.
To add a sentence to the selected KB, you may:

- either type a sentence into the entry field and press the "Add to KB" button;
- or append to the selected KB an input file of sentences, by selecting from the menu "File" -> "Append to KB Selected";
- or replace the KB list with an input file of sentences, by selecting from the menu "File" -> "New KB...".

To add a sentence for revision, you may:

- either type a sentence into the entry field and press the "Add for Revision" button;
- or append to the revision list an input file of sentences, by selecting from the menu "File" -> "Append to Revisor";
- or replace the revision list with an input file of sentences, by selecting from the menu "File" -> "New Revisor...".

To add a sentence for contraction, you may:

- either type a sentence into the entry field and press the "Add for Contraction" button;
- or append to the contraction list an input file of sentences, by selecting from the menu "File" -> "Append to Contractor";
- or replace the contraction list with an input file of sentences, by selecting from the menu "File" -> "New Contractor...".

If a sentence being added has a syntactic error, the applet will display a red error message above the entry field and will not add the sentence.

Also, if the default consistency check is on and a sentence being added is either inconsistent or mutually inconsistent with the selected KB, the revision list, or the contraction list it is being added to, the applet will display a red error message above the entry field and will not add the sentence.

You may click the Default, CNF, and DNF buttons to view the sentences in a list in the default, CNF, and DNF format, respectively.

To remove a sentence from the selected KB, the revision list, or the contraction list, click the sentence and press the x button in the lower right hand corner of the corresponding list.

To erase the entire KB list, revision list, or contraction list, press the broom button in the lower right hand corner of the corresponding list. A dialog box with options Erase and Cancel will ask you to confirm the erasure.

To specify the settings for computing the belief change scenario, you may either use the "Settings" menu, or click the toggle buttons near the bottom of the main screen.

This bottom panel will also display informative messages and instructions where appropriate.

Here are the steps you should follow to carry out a belief change:

1. Once you have specified the settings and the belief change scenario you want to compute, press the "Get Parallel Change" button.

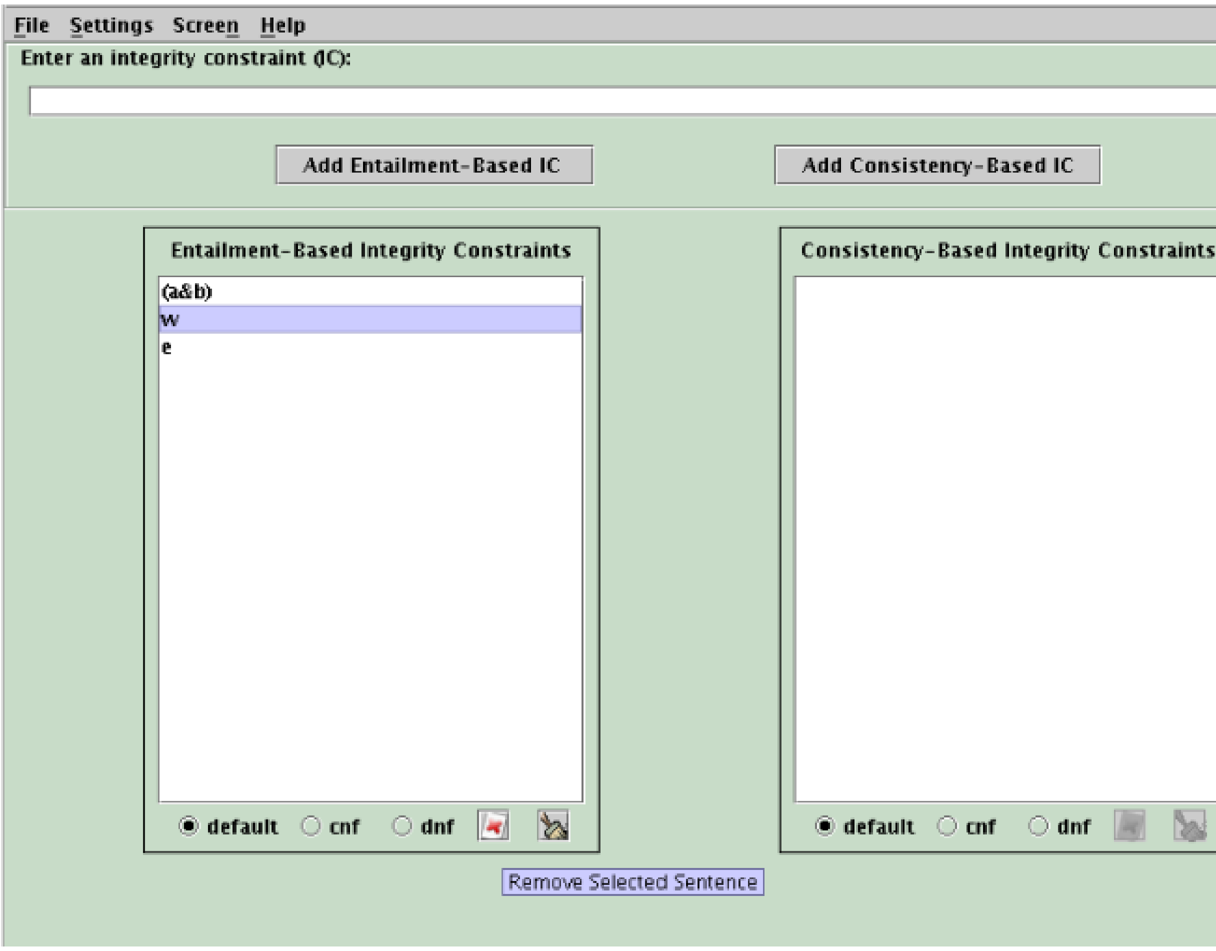
2. If there are inconsistencies in a knowledge base, the revision list, or the EB IC list, or if there are mutual inconsistencies between the revision list, the EB IC list, the contraction list, and the CB IC list, then a dialog box will notify you of all such inconsistencies.
   Until you resolve all such inconsistencies and press the "Get Parallel Change" button again, the belief change scenario will not be computed.

3. Preview the proposed resultant KB(s) in the New Knowledge Base Preview list.
   In the case of choice change, there may be more than one resultant KBs, each one listed on a separate line.

4. Either press either the "Reject" button if you wish to discard the proposed KB(s) and keep the current KB list,
   or choose a proposed KB by clicking it, and press the "Commit" button to replace the current KB list with the selected proposed KB.

3.3 The Integrity Constraints Screen

The second screen is the integrity constraints (IC) interface through which you may add sentences to, or remove sentences from, the entailment-based (EB) IC list and the consistency-based (CB) IC list.


File
Settings
Screen
Help
Enter an integrity constraint (IC):
Add Entailment-Based IC
Add Consistency-Based IC
Entailment-Based Integrity Constraints
Consistency-Based Integrity Constraints
Remove Selected Sentence

Entailment-Based Integrity Constraints:

(a&b)
w
e

● default ○ cnf ○ dnf

Consistency-Based Integrity Constraints:

● default ○ cnf ○ dnf

To add an entailment-based integrity constraint (EB IC), you may:

- either type a sentence into the entry field and press the "Add Entailment-Based IC" button;
- or append to the list an input file of sentences, by selecting from the menu "File" -> "Append to Entailment-based IC";
- or replace the list with an input file of sentences, by selecting from the menu "File" -> "New Entailment-based IC...".

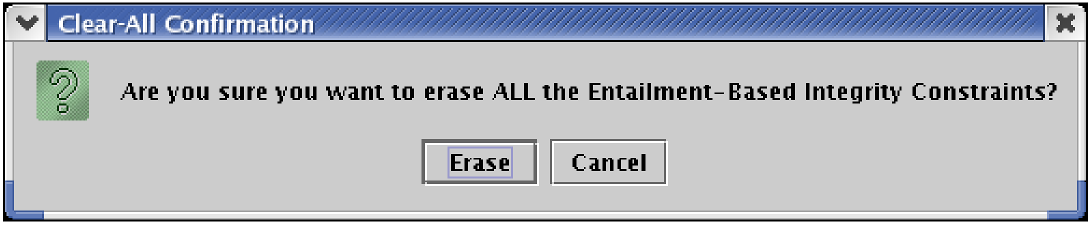
To add a consistency-based integrity constraint (CB IC), you may:

- either type a sentence into the entry field and press the "Add Consistency-Based IC" button;
- or append to the list an input file of sentences, by selecting from the menu "File" -> "Append to Consistency-based IC";
- or replace the list with an input file of sentences, by selecting from the menu "File" -> "New Consistency-based IC...".

If a sentence being added has a syntactic error, the applet will display a red error message above the entry field and will not add the sentence.
Also, if the default consistency check is on and a sentence being added is either inconsistent or mutually inconsistent with the list it is being added to, the applet will display a red error message above the entry field and will not add the sentence.

You may click the Default, CNF, and DNF buttons to view the sentences in a list in the default, CNF, and DNF format, respectively.

To remove an entailment-based integrity constraint (EB IC), click the IC you want to remove and press the x button in the lower right hand corner of the EB IC list.
To erase the entire EB IC list, press the broom button in the lower right hand corner of the EB IC list. A dialog box with options Erase and Cancel will ask you to confirm the erasure.

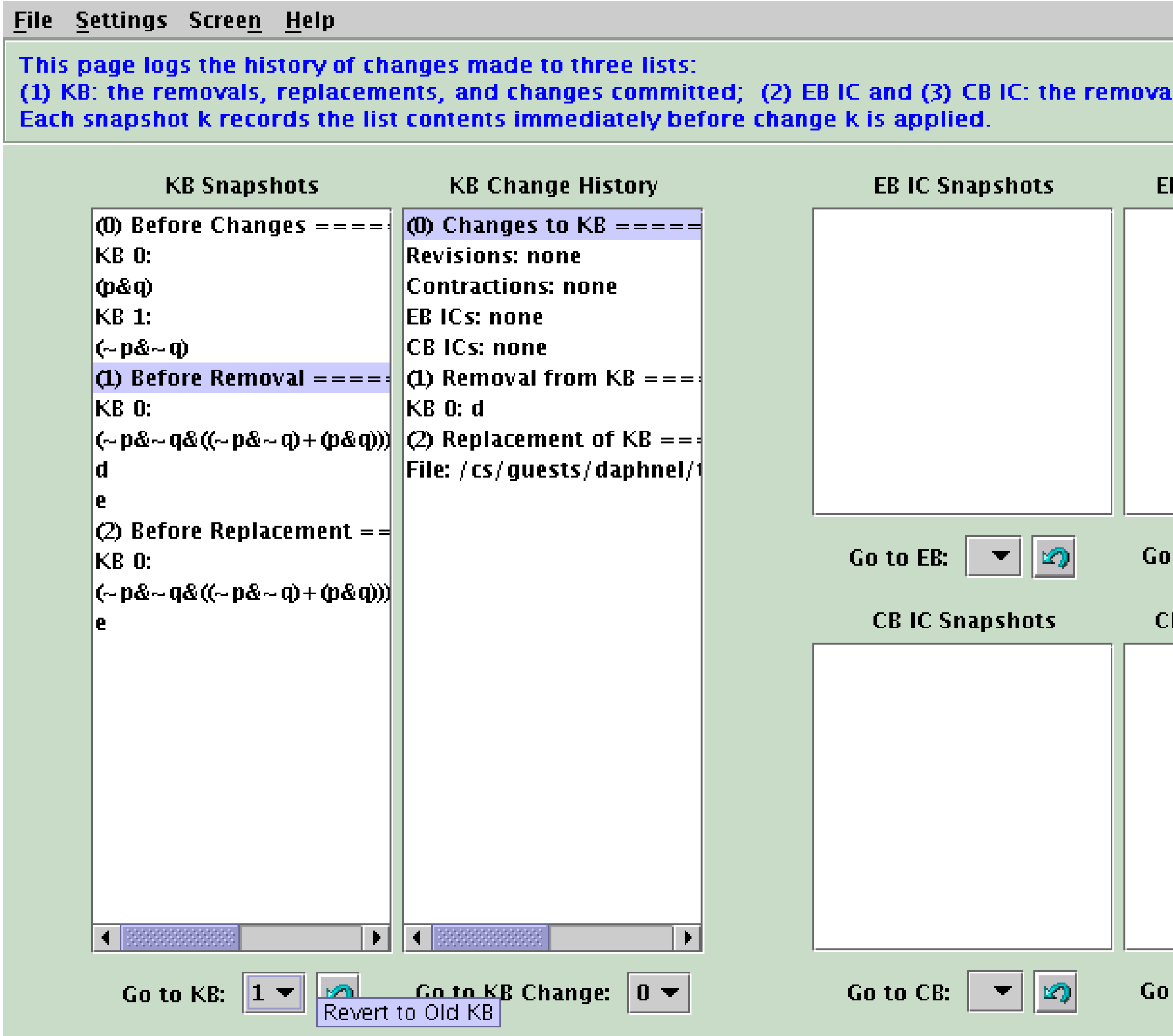


Are you sure you want to erase ALL the Entailment-Based Integrity Constraints?

To remove a consistency-based integrity constraint (CB IC), click the IC you want to remove and press the x button in the lower right hand corner of the CB IC list.
To erase the entire CB IC list, press the broom button in the lower right hand corner of the CB IC list. A dialog box with options Erase and Cancel will ask you to confirm the erasure.

3.4 The History Screen

The third and final screen presents three lists of older snapshots you may revert to. You will find the log of changes for three lists: (1) KB: the removals, replacements, and changes committed; (2) EB IC and (3) CB IC: the removals and replacements made.
Each snapshot k captures the list contents immediately before change k is applied. The numbering of the snapshots starts at 0 and continues in ascending order.



This page logs the history of changes made to three lists:
(1) KB: the removals, replacements, and changes committed; (2) EB IC and (3) CB IC: the remova
Each snapshot k records the list contents immediately before change k is applied.

```
(0) Before Changes ====
KB 0:
(p&q)
KB 1:
(~p&~q)
(1) Before Removal ====
KB 0:
(~p&~q&((~p&~q)+(p&q))
d
e
(2) Before Replacement ==
KB 0:
(~p&~q&((~p&~q)+(p&q))
e
```

```
(0) Changes to KB =====
Revisions: none
Contractions: none
EB ICs: none
CB ICs: none
(1) Removal from KB ===
KB 0: d
(2) Replacement of KB ==
File: /cs/guests/daphnel/t
```

To view a particular KB snapshot, you may:

- either use the vertical scrolling bar that appears on the right hand side of the KB Snapshots list when there are more snapshots than can be displayed on one screen;
- or select its snapshot number from the drop-down list below the KB Snapshots list.

To view a particular EB IC snapshot, you may:

- either use the vertical scrolling bar that appears on the right hand side of the EB IC Snapshots list when there are more snapshots than can be displayed on one screen;
- or select its snapshot number from the drop-down list below the EB IC Snapshots list.

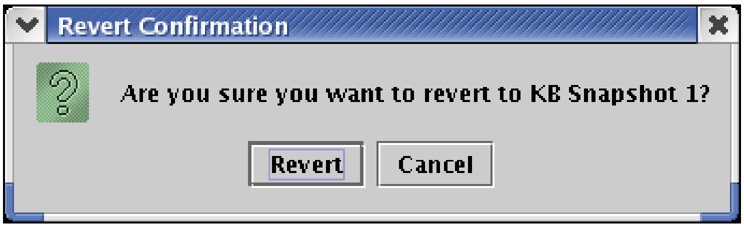
To view a particular CB IC snapshot, you may:

- either use the vertical scrolling bar that appears on the right hand side of the CB IC Snapshots list when there are more snapshots than can be displayed on one screen;
- or select its snapshot number from the drop-down list below the CB IC Snapshots list.

Selecting a snapshot number from a drop-down list will cause the first line of the selected snapshot to be highlighted and displayed in the snapshots list above.
Similarly, selecting a change number from a drop-down list will cause the first line of the selected change to be highlighted and displayed in the change history list above.

To revert the current KB list on the main screen to a particular KB snapshot, do the following steps:

1. Select the number of the snapshot from the drop-down list below the KB Snapshots list;
2. Press the revert button immediately to the right of the drop-down list; and
3. A dialog box with options Revert and Cancel will ask you to confirm the revert operation. Either click option Revert to revert, or click option Cancel to cancel the revert.



Are you sure you want to revert to KB Snapshot 1?

To revert the current EB IC list on the integrity constraints screen to a particular EB IC snapshot, do the following steps:

1. Select the number of the snapshot from the drop-down list below the EB IC Snapshots list;
2. Press the revert button immediately to the right of the drop-down list; and
3. A dialog box with options Revert and Cancel will ask you to confirm the revert operation. Either click option Revert to revert, or click option Cancel to cancel the revert.

To revert the current CB IC list on the integrity constraints screen to a particular CB IC snapshot, do the following steps:

1. Select the number of the snapshot from the drop-down list below the CB IC Snapshots list;

2. Press the revert button immediately to the right of the drop-down list; and

3. A dialog box with options Revert and Cancel will ask you to confirm the reversion. Either click option Revert to revert, or click option Cancel to cancel the reversion.

4 Encoding Input in COBA Syntax

To use COBA, you must encode all your input sentences according to the COBA syntax.

COBA accepts almost all alphanumerical strings for atom names. The exceptions are the symbols in the following comma-separated list: `'` (apostrophe), `+`, `&`, `^`, `~`, `=`, `>`, (and).
Note that `T` and `F` stand for a tautology and a contradiction, respectively.

You can use the following connectives to express more complex formulas.

- `~A` the negation of formula A
- `(A&B)` the conjunction of formula A and formula B
- `(A+B)` the disjunction of formula A and B
- `(A>B)` formula A implies formula B
- `(A=B)` formula A is equivalent to formula B

The syntax of the input sentences is strict. A top-level sentence with a connective operator & (and), + (or), > (implies), or = (is equivalent to) must be enclosed in parentheses. Parentheses inside a sentence, however, are optional and are used only to enforce parsing preference. For example, (a&b+c) is a valid input sentence and is different from (a&(b+c)), whereas a top-level sentence like a&b is syntactically malformed.

4.1 Encoding Input Files in COBA Syntax

Formats of input files vary according to the type of list the sentences are added to.

- KB input file: Each knowledge base should be preceded by a line "KB :" without the double quotes; each sentence should be listed on a separate line.
- Revision input file: Each sentence should be listed on a separate line. A single revisor will be constructed as a conjunction, by taking each sentence as a conjunct.
- Contraction input file: Each line is interpreted as an independent contraction sentence. Consequently, while `(p&~q)` means that `(p&~q)` is to be obliterated from the consequences of the resultant

knowledge base, p ~q listed on two seperate lines means that both p and ~q are to be obliterated from the consequences of the resultant knowledge base.

- Entailment-based integrity constraints (EB IC) input file: Each sentence should be listed on a separate line. A single EB IC will be constructed as a conjunction, by taking each sentence as a conjunct.

- Consistency-based integrity constraints (CB IC) input file: Each line is interpreted as an independent CB IC sentence.

As an example, the following table shows the format of some valid input files.

KB_File "samplekb"	Revision_File "samplerev"	Contraction_File "samplecont"	EBIC_Fi
KB 1:	q	p	(a&b+c)
(p&q&r)	~p	~q	(x&(y+z
KB :			
(~q+~s)			

Special Cases of Belief Change Scenarios to Avoid

The following special cases of belief change scenarios should be avoided.

1. A tautolgy as a contraction sentence.
   error message: `The contraction (p+~p) is a tautology!`

2. A contradiction as a CB IC sentence.
   No error message; sentence not added.

3. A contradiction as a revision sentence.
   error message: `The revision (a&b&~a) is a contradiction!`
   `The conjunction of revisions is inconsistent!`

4. A contradiction as an EB IC sentence.
   error message: `The EB IC (a&b&~a) is a contradiction!`
   `The conjunction of EB ICs is a contradiction!`

5. Conflicting pairs of contraction sentences and revision sentences.
   error message: `The contraction indexed 0 is inconsistent with the conjunction of revis`

6. Conflicting pairs of CB IC sentences and contraction sentences.
   error message: `The contraction indexed 2 is inconsistent with the CB IC indexed 0 (ind`

7. Conflicting pairs of CB IC sentences and revision sentences.
   error message: `The CB IC indexed 1 is inconsistent with the conjunction of revisions (`

8. Conflict between the conjunction of EB IC sentences and a CB IC sentence.
   error message: `The CB IC indexed 3 is inconsistent with the conjunction of EB ICs (ind`

9. Conflicting pairs of EB IC sentences and contraction sentences.
   error message: `The contraction indexed 6 is inconsistent with the conjunction of EB IC`

10. **Conflict between the conjunction of revision sentences and the conjunction of EB IC sentences.**
    error message: `The conjunction of revisions is inconsistent with the conjunction of EB`

5 The First Examples

Now that you have got the program running, let's start executing some belief change scenarios.

The subdirectory samples of the unzipped archive contains a collection of sample test files. In the following sections, you will see several examples of how you can compute belief changes using COBA.

5.1 Revision

The problem of revision arises when, for example, an intelligent agent has to modify his or her belief(s) because he or she may have required more recent or preferred information. The new information may conflict with the agent's older beliefs.

The ability to revise one's knowledge is essential for sound reasoning. While an agent wants to incorporate the new information into their beliefs, the agent also desires to (1) retain as much of his or her existing knowledge as consistently possible and (2) maintain a consistent set of beliefs.

Given a knowledge base $K = (p \wedge q)$and a revision sentence $\alpha = (\neg p \vee \neg q)$, what should the resulting knowledge base look like?

Simply conjoining the knowledge base with the revision sentence would violate consistency since $(p \wedge q \wedge (\neg p \vee \neg q)) - \bot$. On the other hand, disjoining the two would still yield a model with $(p \wedge q) = true$ although the revision sentence states that either p or q must be $false$.

We will now show step by step how this revision example is computed by COBA.

1. Find the common atoms between the knowledge base and the revision sentence.

   $CommonAtoms = \{p, q\}$

2. Create a new knowledge base K' by priming the common atoms in K.

   $K' = (p' \wedge q')$

3. Find all maximal equivalence sets $EQ = \{p' \equiv p | p \in CommonAtoms\}$ such that $K' \cup \{\alpha\} \cup EQ$ is satisfiable.

   $EQ_1 = \{p' \equiv p\}$

   $EQ_2 = \{q' \equiv q\}$

4. For each maximal set EQ, create a belief change extension by (a) unpriming in K' every primed atom p' if $(p' \equiv p) \in EQ$, (b) replacing every primed atom p' with $\neg p$ if $(p' \equiv p) \notin EQ$, and finally (c) conjoining this with the revisior α.

   $K \dot{+}_{c_1} \{\alpha\} = (p \wedge \neg q)$

   $K \dot{+}_{c_2} \{\alpha\} = (\neg p \wedge q)$

5. The resultant knowledge base will simply be either the disjunction of all such belief change extensions for *skeptical* belief change, or one such belief change extension for *choice* belief change.

   (Skeptical Belief Change) $K \dot{+} \{\alpha\} = (p \wedge \neg q \vee \neg p \wedge q)$

To try this example, load the files `example1.kb` as the knowledge base and `example1.rev` as the revision input file.

5.2 Contraction

Contraction is complementary to revision in that certain information is to be removed from the knowledge base. This means the sentence we want removed from the knowledge base should not be a consequence of the resulting knowledge base.

It is important to know that contracting a sentence does not neccesarily mean that the negation of this sentence is in the resulting knowledge base, as that would simply be revising the knowledge base with the negated sentence. The resulting knowledge base should have at least one model in which the contraction sentence is not true.

Given a knowledge base $K = (p \vee q)$ and a contraction sentence $\alpha = (p \vee q)$, what should the resulting knowledge base look like?

The original knowledge base has only three models: $(p, q), (\neg p, q), and (p, \neg q)$, and all are consistent with the contraction sentence. Here, a reasonable extension could be a revision with the negation of the contraction sentence; however, we may lose too much of the original knowledge base because then the resulting knowledge base would have only models where the negation of the contraction sentence is true. As you will see in the steps below, COBA avoid this problem.

We will now show step by step how this contraction example is computed by COBA.

1. Find the common atoms between the knowledge base and the contraction sentence.

   $CommonAtoms = \{p, q\}$

2. Create a new knowledge base K' by priming the common atoms in K.

   $K' = (p' \vee q')$

3. **Find all maximal equivalence sets $EQ = \{p' \equiv p | p \in CommonAtoms\}$ such that $K' \cup \{\neg\alpha\} \cup EQ$ is satisfiable.**

   $EQ_1 = \{\}$

4. **For each maximal set EQ, create a belief change extension by (a) unpriming in K' every primed atom p' if $(p' \equiv p) \in EQ$, (b) replacing every primed atom p' with $\bot$ if $(p' \equiv p) \notin EQ$, and finally (c) replacing all remaining primed atoms with $\top$.**

   $K \dot{-}_{c_1} \{\alpha\} = (\top)$

5. **The resultant knowledge base will simply be either the disjunction of all such belief change extensions for *skeptical* belief change, or one such belief change extension for *choice* belief change.**

   In this case, there is only one possible resulting knowledge base for skeptical change and for choice change:

   $K \dot{-} \{\alpha\} = (\top)$

The resulting knowledge base has now three models $(\neg p, \neg q), (\neg p, q), and (p, \neg q)$. One model conflicts with the contraction sentence while the other two are the same as in the original knowledge base.

To try this example, load the files `example2.kb` as the knowledge base and `example2.con` as the contraction input file.

5.3 Merging

The problem of merging knowledge bases with possibly conflicting pieces of information could arise if different sources, such as sensor data or reports of multiple agents, must be combined. In contrast with revision and contraction, one normally does not have the knowledge that one source is preferred over the others.

Although the simplest solution would be to take the disjunction of all sources, this may result in so much information loss that we may get a resulting knowledge base with too many models. Rather, we want to find a resulting knowledge base which only has the common models among the source knowledge bases.

Given a knowledge base 1 $K_1 = (p \wedge q \wedge r)$ and a knowledge base 2 $K_2 = (\neg p \wedge \neg q \wedge \neg r)$, what should the resulting knowledge base look like?

COBA provides two different merge operators: default/basic merge and projection merge.

We will first show step by step how this merging example is computed by COBA using default/basic merge.

1. **Find the common atoms between the two knowledge bases.**

   $CommonAtoms = \{p, q, r\}$

2. **Create a new knowledge bases K_i^i by numbering the common atoms in K_i with i.**

   $K_1^1 = (p^1 \wedge q^1 \wedge r^1)$

   $K_2^2 = (\neg p^2 \wedge \neg q^2 \wedge \neg r^2)$

3. **Find all maximal equivalence sets $EQ = \{p^i \equiv p' | p \in \mathit{CommonAtoms}\}$ such that $\bigcup K_i^i \cup \bigcap K_i^{'} \cup EQ$ is satisfiable.**

   $EQ_1 = \{p^1 \equiv p', q^1 \equiv q', r^1 \equiv r'\}$

   $EQ_2 = \{p^2 \equiv p', q^2 \equiv q', r^2 \equiv r'\}$

4. **For each maximal set EQ, create a belief change extension by (a) unnumbering in K^i every numbered atom p^i if $(p^i \equiv p') \in EQ$, (b) replacing every numbered atom p^i with $\neg p$ if $(p^i \equiv p') \notin EQ$, and finally (c) conjoining the two unnumbered knowleledge bases with the disjunction of all source knowledge bases.**

   $K_1 \triangle_{c_1} K_2 = (p \wedge q \wedge r)$

   $K_1 \triangle_{c_2} K_2 = (\neg p \wedge \neg q \wedge \neg r)$

5. **The resultant knowledge base will simply be either the disjunction of all such belief change extensions for *skeptical* belief change, or one such belief change extension for *choice* belief change.**

   (Skeptical Belief Change) $K_1 \triangle K_2 = ((p \wedge q \wedge r) \vee (\neg p \wedge \neg q \wedge \neg r))$

The default/basic merge operator at least retains as minimal information the disjunction of both sources. In contrast, the projection merge does not keep this information.

We will next show step by step how this merging example is computed by COBA using projection merge.

1. **Find the common atoms between the two knowledge bases.**

   $\mathit{CommonAtoms} = \{p, q, r\}$

2. **Create new knowledge bases K_i^i by numbering the common atoms in K_i with i.**

   $K_1^1 = (p^1 \wedge q^1 \wedge r^1)$

   $K_2^2 = (\neg p^2 \wedge \neg q^2 \wedge \neg r^2)$

3. **Find all maximal equivalence sets $EQ = \{p^i \equiv p' | p \in \mathit{CommonAtoms}\}$ such that $\bigcup K_i^i \cup EQ$ is satisfiable.**

   $EQ_1 = \{p^1 \equiv p', q^1 \equiv q', r^1 \equiv r'\}$

   $EQ_2 = \{p^1 \equiv p', q^1 \equiv q', r^2 \equiv r'\}$

   $EQ_3 = \{p^1 \equiv p', q^2 \equiv q', r^1 \equiv r'\}$

   $EQ_4 = \{p^2 \equiv p', q^1 \equiv q', r^1 \equiv r'\}$

   $EQ_5 = \{p^1 \equiv p', q^2 \equiv q', r^2 \equiv r'\}$

   $EQ_6 = \{p^2 \equiv p', q^2 \equiv q', r^1 \equiv r'\}$

   $EQ_7 = \{p^2 \equiv p', q^1 \equiv q', r^2 \equiv r'\}$

   $EQ_8 = \{p^2 \equiv p', q^2 \equiv q', r^2 \equiv r'\}$

4. **For each maximal set EQ, create a belief change extension by (a) unnumbering in K^i every numbered atom p^i if $(p^i \equiv p') \in EQ$, (b) replacing every numbered atom p^i with $\neg p$ if $(p^i \equiv p') \notin EQ$.**

   $K_1 \triangle_{c_1} K_2 = (p \land q \land r)$

   $K_1 \triangle_{c_2} K_2 = (p \land q \land \neg r)$

   $K_1 \triangle_{c_3} K_2 = (p \land \neg q \land r)$

   $K_1 \triangle_{c_4} K_2 = (\neg p \land q \land r)$

   $K_1 \triangle_{c_5} K_2 = (p \land \neg q \land \neg r)$

   $K_1 \triangle_{c_6} K_2 = (\neg p \land \neg q \land r)$

   $K_1 \triangle_{c_7} K_2 = (\neg p \land q \land \neg r)$

   $K_1 \triangle_{c_8} K_2 = (\neg p \land \neg q \land \neg r)$

5. **The resultant knowledge base will simply be either the disjunction of all such belief change extensions for *skeptical* belief change, or one such belief change extension for *choice* belief change.**

   (Skeptical Belief Change) $K_1 \triangle K_2 = \top$

To try this example, load the files `example3.kb` as the knowledge bases.