

Sequential Cognition Processes: A Framework For Reasoning with Non-Monotonic Logics

Axel Ind
axeltind@gmail.com
University of Freiburg

February 11, 2020

Research Question: *“Is there a way to systematically represent some (or all) non-monotonic logics in a framework that standardises cognitive modelling under these logics without sacrificing their expressiveness?”*

1 Overview

For my Master’s thesis I intend to describe, implement, and test a novel framework for modelling human reasoners called the Sequential Cognitive Process (SCP). SCPs retain most of the expressiveness of the hitherto *ad hoc* approaches to modelling reasoners with non-monotonic logics, but also introduce a number of desirable properties:

1. They enable modelling of individual reasoners through addition or removal of well-described mental operations. This is in contrast to most non-monotonic logic approaches which can model the general reasoner across a number of cognitive tasks but do not extend well to explaining reasoning that deviates from the norm.
2. They standardise a broad class of non-monotonic logical frameworks so that cognitive comparisons, even across different frameworks are possible, in particular I have given some thought to the use of string-matching algorithms borrowed from the field of bioinformatics (such an extended version of the Needleman Wunsch Aglorithm(Needleman and Wunsch, 1970)) to comparing SCPs as operation sequences.
3. Because they are sequential (essentially linked lists of mental operations) they lend themselves well to traditional search techniques, and their utility in finding – though not yet justifying – models of the Suppression Task has already been demonstrated for both general and individual reasoners.
4. They standardise a set of well-founded cognitive processes (such as, the Suppression Effect, Affirmation of the Antecedent, Conjunction Fallacy bias, and the introduction of abnormalities) as reusable epistemic operations which can be constrained or expanded according the needs of the researcher.
5. Because they are sequences of actions with standardised input and output formats, there is certainly potential for traditional heuristic approaches to problem solving with SCPs as well as purpose-built heuristic adapted to the cognitive task being considered.

A limited Python version of this system has already been implemented¹ for the Weak Completion Semantics (Hölldobler, 2015). This implementation has proved effective at modelling both the Suppression Task (Byrne, 1989) and the Wason Selection Task (Wason, 1968). Work to expand the implementation is ongoing.

¹Available on Github: <https://github.com/AxelInd/SequentialCognitionProcoesses>. For access, please contact me with your details.

2 What is an SCP

An SCP is a model of the thought process of a reasoner, whether individuals or a group. The SCP assumes that some epistemic state (of whatever form the nonmonotonic logic requires) is modified by mental operations in the SCP to reach a final epistemic state that satisfies some goal properties. An SCP Problem $\pi = (s_i, \gamma, M)$ describes the requirements and limitations of the SCP before it is created². s_i contains the initial state of the SCP, and γ is the goal of the SCP (e.g. to mimic the classical case of the Suppression Task), M is the set of allowable cognitive operations.

The SCP framework makes several core assumptions about the nature of cognition:

1. **Evolving Epistemic States:** SCPs are built on the idea that some mental model of the world exists before the problem-solving begins (even if the model is empty), and is updated by each mental process that acts on it, yielding a new epistemic state as output from mental process m_n to be used as input for mental process m_{n-1} .
2. **Sequential Reasoning:** the assumption that a reasoner will perform only one mental operation at a time before moving on to the next one. Although it initially seems very limiting, the complexity of mental operations is unconstrained, and could itself involve parallel processes provided that a unified mental state is achieved before the next operation begins.
3. **Reusable Operations:** in order to be more than just a container for some other non-monotonic logic, cognitive tasks must share at least some complex operations. If this were not the case, standardising a set of cognitive operations to model unseen tasks would be impossible. Fortunately, there is ample evidence of consistent mental approaches to reasoning (Ragni et al., 2017).
4. **Limited mental processing power:** SCPs assume a sequential and finite number of mental operations to reach a goal. Without this property, it would be impossible to accurately model human cognition. Further, there would be no reasonable scoring system for determining the best explanation when multiple SCPs all model the same thought process (the simplest scoring system intuitively is to compare the length of the two candidate SCPs and to prefer the smaller).

3 State of the Art

3.1 More Questions Than Answers

The example of the Suppression Task (Byrne, 1989) illustrates one of the glaring problems with cognitive modelling in non-monotonic reasoning as it stands at present: it is unstructured. What rules should be followed when adding abnormalities? How do we choose the order to apply these operations? Is this the only reasonable sequence in which to apply these operations? All these questions arise from a lack of structure in the frameworks being used. For example, work by ? discusses the idea of obligate and factual conditionals, which are treated differently in terms of how abnormalities are added. From this it immediately follows that extensions to the WCS are possible, but where can they occur? Which extensions and changes are well founded in the psychological literature? What are the properties of the search spaces of these extensions? Is it even possible to compare different cognitive models to one other if they produce identical output? SCPs are an attempt to create a framework for non-monotonic logics that is suitable to answer these questions.

One of the most important questions to answer, is if it possible for these non-monotonic logics to model individual reasoners as well as general reasoners. Work by (?) showed that simple extensions to the Weak Completion Semantics allowed researchers to model the four most common decisions of participants in the Wason Selection Task. There is significant potential for research into modelling the unusual choices of participants who do not conform to the normally observed results. Part of my thesis will be dedicated to justifying a view of these unusual reasoners as deviations from the normal cognitive process who still follow a very similar reasoning approach but with small and well-founded differences when compared to general reasoners. In the following sections we briefly discuss this idea in terms of the Suppression Task.

²This formulation is inspired by the field of AI Planning

3.2 Example: The Suppression Task

3.2.1 What is the Suppression Task?

The Suppression Task refers to an experiment conducted by Byrne (1989) and is a classical example of the inadequacy of monotonic logics for modelling human reasoning. In classical logic, if our knowledge base kb is such that $kb \models \phi$, then it must be the case that $kb \cup \psi \models \phi$. However, in the suppression task participants no longer draw classically valid inferences when new information is added. The task is often formulated as follows:

- $e \rightarrow l$: If she has an essay to write (e), she will study late in the library (l).
- $\top \rightarrow e$: She has an essay to write (e).
- $o \rightarrow l$: If the library is open (o), she will study late in the library (l).

Given only the rules ($e \rightarrow l$) and ($\top \rightarrow e$), the participants consistently concluded that she would study late in the library, seemingly drawing the classical logic inference $\frac{e \rightarrow l, e}{l}$ with *modus ponens*. But when given the additional rule $o \rightarrow l$, participants no longer believe that they have enough information to judge whether she will study late in the library, and a significant portion of them no longer draw the classical conclusion. This effect, called Suppression, demonstrates the need for something more than classical logic for modelling human reasoning.

3.2.2 Modelling the Suppression Task with the WCS

Work by ? has shown that the Weak Completion Semantics is an adequate non-monotonic logic for modelling the Suppression Task. The reader is assumed to be familiar with the Weak Completion Semantics and so it is only briefly discussed in Appendix 7.1.

Under the Weak Completion Semantics and Łukasiewicz 3-valued logic the task is usually modelled as follows for the case without suppression:

1. Initial logic program: $P = \{e \rightarrow l, \top \rightarrow e\}$. This program represents the task without information about what happens if the library is open.
2. Addition of Abnormality: $P = \{e \wedge \neg ab_1 \rightarrow l, \top \rightarrow e, \perp \rightarrow ab_1\}$. The program now reflects the possibility that some abnormal event may prevent her from going to the library, but because we have no information about the nature of this event, it is set to false by default.
3. Weak Completion: $wc(P) = \{e \wedge \neg ab_1 \leftrightarrow l, \top \leftrightarrow e, \perp \leftrightarrow ab_1\}$. Weak completion is applied to the logic program and, in this case, only implications need to be changed to bijections.
4. Semantic Operator:
 - Execution 1: $\top = \{e\}, \perp = \{ab_1\}$
 - Execution 2: $\top = \{e, l\}, \perp = \{ab_1\}$

After application of the semantic operator l is true in the least model, and so participants conclude that she will study late in the library (as when P is evaluated classically). However, in the case where Suppression is observed, the same process yields a different result because of the presence of the extra conditional ($o \rightarrow l$).

1. Initial logic program: $P = \{e \rightarrow l, \top \rightarrow e, o \rightarrow l\}$. The initial program now includes information about the extra (suppressing) conditional.
2. Addition of Abnormality: $P = \{e \wedge \neg ab_1 \rightarrow l, o \wedge \neg ab_2 \rightarrow l, \top \rightarrow e, \neg o \rightarrow ab_1, \neg e \rightarrow ab_2\}$. Adding abnormalities is a poorly described process and often relies on intuitionist views of what actually constitutes an abnormal situation. In this case we adopt the practise of identifying abnormalities as described in Appendix 7.2.

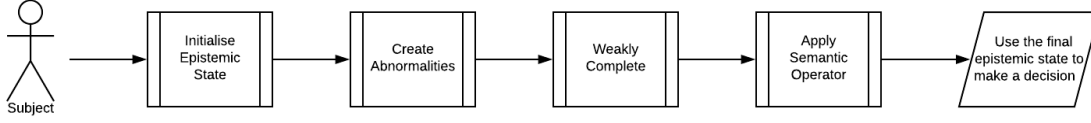


Figure 1: A generalised illustration of the WCS in an SCP.

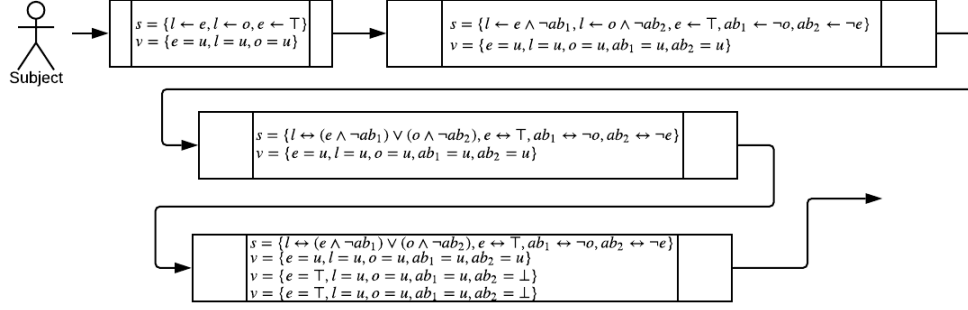


Figure 2: The standard case of the Suppression Task, demonstrating the suppression effect. Where the epistemic state in the boxes represents the output of that cognitive operation.

3. Weak Completion: $wc(P) = \{((e \wedge \neg ab_1) \vee (o \wedge \neg ab_2)) \leftrightarrow l, \top \leftrightarrow e, \neg o \leftrightarrow ab_1, \neg e \leftrightarrow ab_2\}$. Weak completion is applied to the logic program, combining rules with shared heads and replacing implications with bijections.
4. Semantic Operator:
 - Execution 1: $\top = \{e\}, \perp = \{ab_2\}$

Now suppression has been displayed in the logic program and the variable l remains unknown in the least model.

4 Modelling the Suppression Task with the SCPs

Figure 1 illustrates a generalised process to describe the Suppression Task as a series of sequential steps, each process passing information to the next process³

Under SCPs the implicit sequence of operations in the Suppression Task are systematized and refined into complex operations. Every complex operation takes as its input an epistemic state s_i and outputs an epistemic state s_{i+1} to be used as input to the next complex operation. For the Weak Completion Semantics it suffices to add an initial epistemic state $s_i = (KB, V)$ consisting of a set of rules KB , and a set of variable assignments V (in the case of the Wason Selection Task additional information about obligate and factual conditions can be added to model *modus tollens* inferences in the WCS, but that will be discussed in more detail in a later paper).

One interpretation of the requirements of the suppression task $\pi = (s_i, \gamma, M)$ using SCPs and the WCS is as follows:

- $s_i = \{kb_i, v_i\}$

³It is important to note that a diagram like this is valid for *any* cognitive modelling task because any process may be arbitrarily complex and non-sequential. and so the overall linear process of (actor, complex decision, observed results) is always valid for retroactive modelling, and at least as powerful as the non-monotonic logic framework it uses for modelling.

- $kb_i = \{e \rightarrow l, \top \rightarrow e, o \rightarrow l\}$
- $v_i = \{e : u, l : u, o : u\}$
- $\gamma = l \models \top$ or $l \models \perp$ (that is, l is no longer mapped to unknown in the final epistemic state.)
- $M = \{\text{init}, \text{addAB}, \text{WeaklyComplete}, \text{semanticOperator}\}$

where **init** is always the first cognitive operation and adds the initial variables and rules to epistemic state; **addAB** adds abnormalities to the current epistemic state using the procedure described in Algorithm 1 (but now also adds those abnormalities to the variable list of the epistemic state; **WeaklyComplete** weakly completes the knowledge base of the current epistemic state; and **semanticOperator** returns an epistemic state that leaves the knowledge base unchanged but updates the variables of that state to return the least model of the epistemic state.

Treating Figure 1 as an SCP, we observe the sequence of output states seen in Figure 2. Note that in the final state l remains unknown and the suppression effect is demonstrated.

4.0.1 Extending the Suppression Task with the SCPs

One of the most significant tasks of my thesis will be to find and implement a variety of cognitive operations for which there are evidence in the psychological literature. The selection of allowable complex operations M will have a significant impact on which tasks the SCP is able to handle, and only those for which there is supporting evidence should be added.

The previous example merely showed that SCP are suitable for modelling the suppression task. In this example we consider one of the most powerful characteristics of SCPs, the ability to model unusual results as deviations from general reasoning. A significant portion of people still believe that she will study late in the library when presented with the suppression task. Several possible explanations are intuitive, the first and simplest, is the assumption that the reasoner is using classical logic and drawing the classical conclusion. However, what if that is not the case? What if they do reason in exactly the same way as the other reasoners, except for one or two small deviations?

In this example we consider two possible deviations that could explain the classical result of the Suppression Task: variable deletion, and variable fixation. Both of these operations will be discussed in a way that may seem overly prosaic to many readers, but the reason for the examples given is to reinforce how reasonably we might expect these cognitive operations occur in reasoning tasks.

Variable Deletion

Consider the sequence of numbers (1, 44, 27, 8, 0, -4, 6, 7, 346, 7, 74, 7, 234, -55, 2.4, 18). Now without looking back at the numbers, ask yourself some questions: how many numbers were there? Were any of them prime? How many numbers were repeated? In all probability you are not entirely sure. This simple thought experiment provides support for our first extension, the idea that variables can be “forgotten”, that is, that information that existed in the knowledge base at one point in time might no longer exist at a later timepoint.

This is not the only imaginable case where a variable might be removed from the knowledge base of the person being modelled. The size of the knowledge base used for cognitive modelling is always implicitly restricted to relevant variables, only those variables whose values might reasonably be expected to affect the final conclusions drawn with regard to the research question. Finding which variables and rules are relevant is, however, non-trivial. For another real-life example, imagine a mystery novel. Three hundred pages of plot descriptions, character actions, and dialogues. In a good murder mystery novel every piece of information that reveals the killer’s identity is hidden in the story itself, yet we do not hold every fact and interaction in the book in our epistemic model of the book, so discerning the identity of the killer remains a mystery until the last page. But when the mystery is solved, many details that we internalised while reading (and recall in retrospect) suddenly make the conclusion seem obvious. We have not forgotten this information, we had merely incorrectly deemed it irrelevant at the time and ignored it in our cognitive processing.

The exact details of how to delete a variable from a knowledge base is non-trivial, and there is no best practice for doing so. But in simple cases the process can be intuitive. Given $kb = \{a \leftarrow \top, b \leftarrow \top\}$ deleting

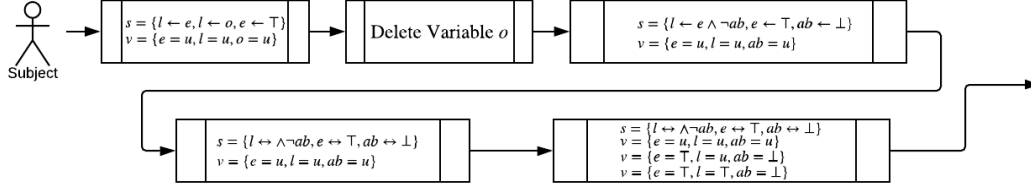


Figure 3: The Suppression Task in which the additional operation of deleting the variable o occurs.

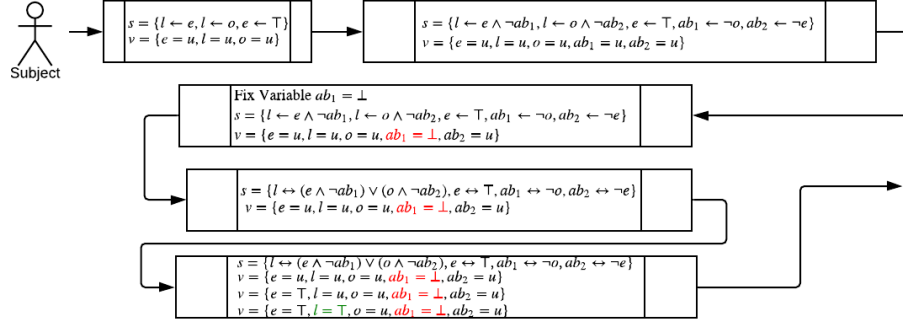


Figure 4: The Suppression Task in which the additional operation of fixing the variable ab_1 to false occurs.

variable a might be as simple as removing all rules that mention it from the knowledge base (though an argument could also be made that it should be set to unknown).

In the case of the Suppression Task we argue that one cognitively valid reason for drawing the classical conclusion to the task may be forgetting (or disregarding) the variable o . Figure 3 illustrates this case, and shows how the insertion of a complex operation can completely change the final epistemic state.

Variable Fixing

The second case of a potential complex operation to add to the search space of our SCPs is the idea of Variable Fixing. The idea that some conclusions can be fixed *a priori*. Consider a person who strongly doubts the effectiveness of vaccines, we will call her Karen. Karen started her day convinced that giving her child the MMR vaccine is more dangerous than the disease itself. Later that day Karen spoke to her doctor who strongly advised that she vaccinate her child. He offered her a variety of peer-reviewed papers and studies that showed the relative safety of the vaccination. Karen listened carefully to the trained medical professional, and then went home. After some thought Karen decided that he was wrong, and her opinion on vaccines didn't change.

In this example Karen shows a very powerful type of cognitive bias, the unwillingness to change change her opinions, despite powerful evidence to the contrary. This phenomenon has observed across a great many fields of study, from medical psychology (Brown et al., 2010) (Wroe et al., 2005) to political sciences (Tappin et al., 2017). In the context of cognitive modelling with logics, it indicates that a mental rules or variables are immutable, regardless of new evidence or valid beliefs that would logically contradict them. Non-monotonic logics, as a class, are already capable of dealing with bias effect, as non-monotonic logics are built on the basis of a preference operation. However, for the case of the Weak Completion Semantics and SCPs, it serves to introduce a complex operation *fixX* which can fix the value of a rule or variable at every point after it occurs in the SCP.

In practise this necessitates a way of denoting fixed variables so that future operations do not change them, it also strongly suggests the plausibility of some operation *unfixX* which would allow a variable to

be fixed to a new value again after it occurs. For the purposes of the Suppression Task it suffices to show that the *fixX* operation can change the conclusions of the general reasoner to those of the deviant reasoner.

Figure 4 shows the effect of adding a complex operation which fixes the value of the abnormality to false in *v* so that, no matter what rules are present in *kb* when the semantic operator is applied, *ab*₁ will remain false⁴.

5 Implementation

A simple implementation of the SCP framework has already been implemented for the WCS, it has been shown able to model both the Suppression Task and the Wason Selection Task in terms of both general and deviant reasoning. The overall structure of the SCP implementation is as a pipeline, a linked list of operations each of which relies on input from the previous operation in the list.

5.1 Class Diagram

Figure 5 shows a simplified class diagram of the implementation of the SCP program. The `scp` class makes use of a basic implementation of 3-valued logic and a sequence of `complexOperation` objects in order to evaluate the knowledge base and variables of the SCP after each operation.

Each `complexOperation` is a pipeline, it contains no data other than pointers to its successor and predecessor and passes the results of its `evaluatev()` and `evaluatekb()` functions to the next operation as input. The code itself is heavily commented and contains files which implement and execute examples of the Suppression Task and Wason Selection task under WCS.

5.1.1 Additional Features

Several other features will be added to the implementation soon:

- DeNovo Searching: finding an SCP that satisfies a goal condition from scratch, iterating over the set of cognitive operation in *M*. This search is important for coming up with an explanation for the general (most common) results of a cognitive task.
- Insertion Searching: finding which additional complex operations can be added to an existing SCP to produce results consistent with the deviant reasoner. This search is important for describing unusual results as deviations from the general reasoner. It is also important for scoring the likelihood of these the SCP.
- Scoring: a modified version of Needleman-Wusch algorithm for string matching has already been theoretically described and must now be implemented in order determine how similar different models are to each other.

⁴The question of which solution best models deviant reasoners in the suppression task is one of significant interest and I intend to go into some detail about possible scoring algorithms and heuristics for determining the most likely SCP for a cognitive task when multiple candidate SCPs come to the desired conclusions.

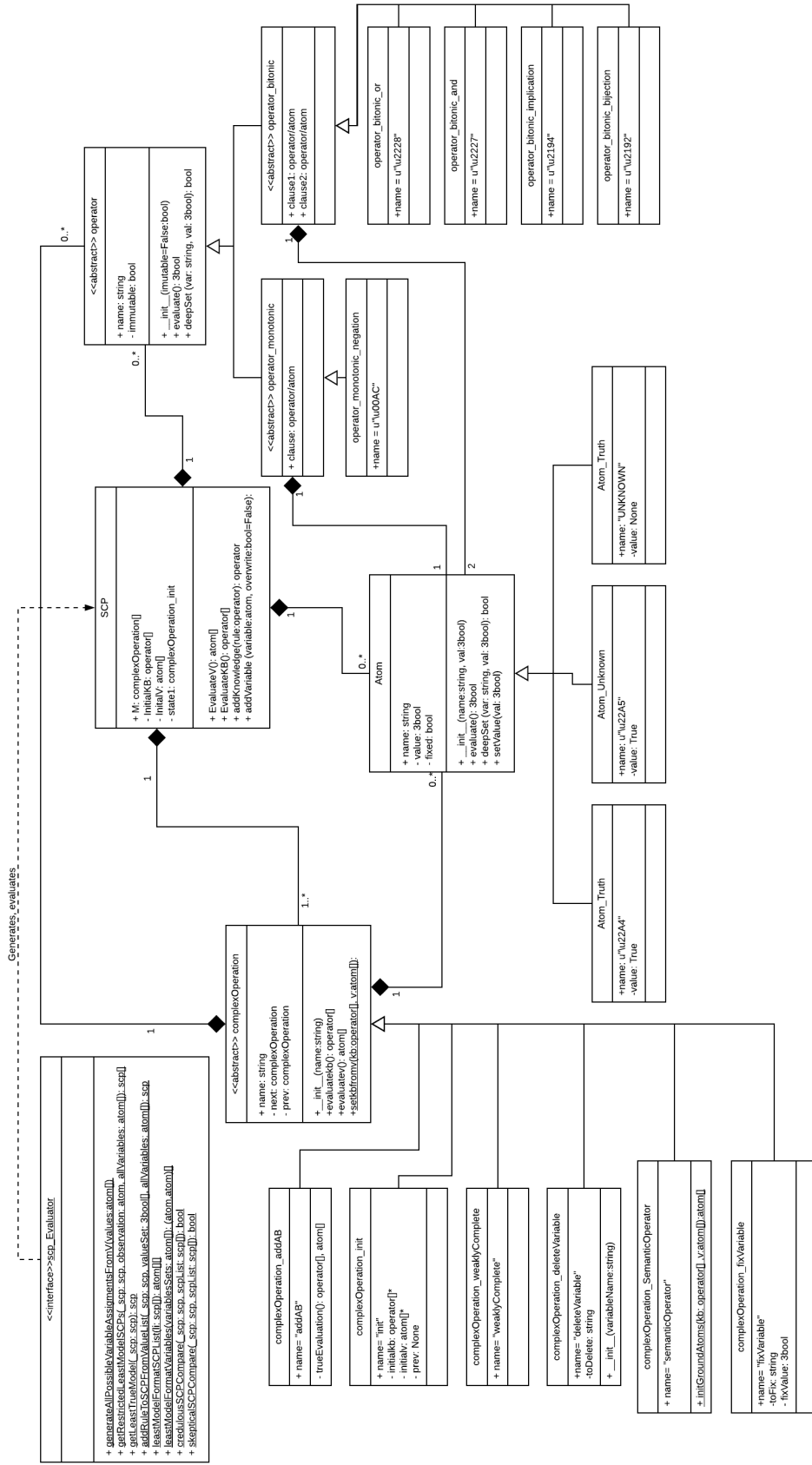


Figure 5: A simplified illustration of class interactions in the SCP implementation.

6 Milestones

A portion of the work to create and formalise SCPs has already been completed. A literature review, time spent programming, and time spent formalising the idea of SCPs has already devoted to the problem. however there is still a significant amount of work to be done:

Timeline

- Standardise implementation interface: 3 Feb - 9 Feb
- Extend Interface to Conjunction Fallacy under WCS: 10 Feb - 16 Feb
- Complete paper on SCPs under the WCS: 17 Feb - 23 Feb
- Conceptually extend SCPs to Reiter's Default Logic: 24 Feb - 1 March
- Extend the existing implementation to handle Default Logic: 2 March - 8 March
- Explore scoring algorithms for SCps: 3 March - 9 March
- Examine application of heuristics to SCPs: 10 March - 15 March
- Complete paper on SCPs under Default Logic: 16 March - 23 March
- Write Thesis: 23 March - 20 April
- Correction, updates, and new ideas: 21 April - ...

Deliverables

- Conference ready paper on SCPs and the WCS: 23 Feb
- Conference ready paper on SCPs and the default logic: 23 March
- Complete, documented, extensible implementation of SCPs containing various examples, search functionality, and best-practice design: 1 April

References

- Brown, K. F., J. S. Kroll, M. J. Hudson, M. Ramsay, J. Green, C. A. Vincent, G. Fraser, and N. Sevdalis (2010). Omission bias and vaccine rejection by parents of healthy children: implications for the influenza a/h1n1 vaccination programme. *Vaccine* 28(25), 4181–4185.
- Byrne, R. M. (1989). Suppressing valid inferences with conditionals. *Cognition* 31(1), 61–83.
- Hölldobler, S. (2015). Weak completion semantics and its applications in human reasoning. In *Bridging@CADE*, pp. 2–16.
- Needleman, S. B. and C. D. Wunsch (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology* 48(3), 443–453.
- Ragni, M., C. Eichhorn, T. Bock, G. Kern-Isberner, and A. P. P. Tse (2017). Formal nonmonotonic theories and properties of human defeasible reasoning. *Minds and Machines* 27(1), 79–117.
- Tappin, B. M., L. van der Leer, and R. T. McKay (2017). The heart trumps the head: Desirability bias in political belief revision. *Journal of Experimental Psychology: General* 146(8), 1143.
- Wason, P. C. (1968). Reasoning about a rule. *Quarterly journal of experimental psychology* 20(3), 273–281.
- Wroe, A. L., A. Bhan, P. Salkovskis, and H. Bedford (2005). Feeling bad about immunising our children. *Vaccine* 23(12), 1428–1433.

\rightarrow	\top	u	\perp
\top	\top	u	\perp
u	\top	\top	u
\perp	\top	\top	\top

Table 1: A table showing the implication operator in 3-valued Łukasiewicz logic.

7 Appendix

7.1 Weak Completion

The Weak Completion Semantics is a non-monotonic logic which procedurally encodes several well-known cognitive phenomenon. The WCS makes use of 3-valued Łukasiewicz logic (Table 1). It adds abnormalities to non-ground inferences, and replaces the classical inference (\leftarrow), with a bijective (\leftrightarrow).

The Weak Completion of a program P is defined as follows:

1. Replace all clauses of the form $A \leftarrow body_1, \dots, A \leftarrow body_n$ with $A \leftarrow body_1 \vee \dots \vee body_n$.
2. Replace all occurrences of \leftarrow with \leftrightarrow .

Applying this procedure to P results in wcP which is the weak completion of P .

The next requirement to apply the WCS framework is the introduction of a semantic operator $\phi_{SvL} (?)$. Let J be the result of applying the semantics operator to an interpretation I and logic program P . The J is defined as follows:

$$J^\top = \{A \mid \text{there exists a clause } A \leftarrow Body \in P \text{ with } I(Body) = \top\}$$

$$J^\perp = \{A \mid \text{there exists a clause } A \leftarrow Body \in P \\ \text{and for all clauses } A \leftarrow Body \in P \text{ we find } I(Body) = \perp\}$$

Using $I = \langle \emptyset, \emptyset \rangle$, the least model ($\text{lm}_L wcP$) can be calculated by iterating $\phi_{SvL, P}$.

7.2 Systematically Adding Abnormalities

Adding abnormalities is a poorly described process and often relies on intuitionist views of what actually constitutes an abnormal situation. For this reason I have provided one concrete method for adding abnormalities to a logic program (though many others exist and can be extended at will). This particular one is based on the principle that if a variable v_1 has the power to make another variable v_2 true, then it may be the case that participants believe that any other variables V' satisfying v_2 may only do so when v_1 holds. This intuition is counter to classical logic, but follows intuitively in cases like the Suppression Task where participants believe that the library not being open may be the reason that she doesn't study late in the library, even when she has an essay to write. The precise procedure is as follows:

In the case of SCPs, the procedure for creating abnormalities is handled by a single complex epistemic operation and so may be substituted for any competing method with ease without disrupting the flow of the SCP.

Algorithm 1 One way of adding abnormalities to a logic program prior to applying the Weak Completion Semantics.

```

1: function ADDABNORMALITIES( $P$ )
2:    $P' := []$ 
3:   for Unique  $head$  with  $(body \rightarrow head) \in P$  do
4:      $headDependencies :=$  [all non-ground bodies s.t.  $(body \rightarrow head) \in P$ ]
5:     for  $body_i$  with  $(body \rightarrow head) \in P$  do
6:        $ab_x :=$ new abnormality  $\triangleright$  Where  $x$  is a unique identifier
7:       create new rule  $head \leftarrow body_i \wedge \neg ab_x$ 
8:        $abDependencies = headDependencies \setminus body_i$   $\triangleright \perp$  if it would be empty
9:       rule=create new rule  $(ab_x \leftarrow dep_1 \wedge \dots \wedge dep_n)$  for  $dep \in abDependencies$ 
10:       $P' += rule$ 
11:    end for
12:  end for
13:  return  $P'$ 
14: end function

```
