

ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG

MASTER'S THESIS

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

Author:
Axel IND

Supervisor:
Dr. Marco RAGNI

*A thesis submitted in fulfillment of the requirements
for the degree of Master of Computer Science*

in the

Cognitive Computation
Department of Computer Science

May 8, 2020

Declaration of Authorship

I, Axel IND, declare that this thesis titled, “Sequential Cognition Processes: A Framework For Reasoning with Non-Monotonic Logics” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

“Everything should be made as simple as possible, but no simpler.”

Albert Einstein

ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG

Abstract

Faculty of Engineering
Department of Computer Science

Master of Computer Science

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

by Axel IND

Approaches to cognitive modelling with non-monotonic logics have thus far been largely *ad hoc* and poorly standardised, making inter-model comparisons difficult. As an attempt to systematically represent non-monotonic logics in a framework that standardises cognitive modelling under these logics without sacrificing their expressiveness, we introduce the Sequential Cognition Process (SCP). Under the assumption that human reasoning can be represented as a sequence of distinct cognitive operations on an initial knowledge base SCPs provide a consistent framework for generating and evaluating models of human cognition. Using an adapted interpretation of the Weak Completion Semantics (WCS), SCPs are able to accurately model several classical experiments in cognitive modelling. We use the SCP framework to model both general case reasoners – which arrive at the most frequently observed conclusions – and poorly-studied individual case reasoners – which do not. We illustrate the use of SCPs using the Suppression Task.

Acknowledgements

Special thanks to Dr. Marco Ragni, for his enthusiastic supervision of my ongoing academic studies; Raymond Ind and Julie Ind for their emotional and financial support; and to Anastasia Sushich for her tireless support.

Contents

Declaration of Authorship	iii
Abstract	vii
Acknowledgements	ix
1 Introduction	3
1.1 Overview	3
1.2 A Review of Terminology	4
1.3 Current State of the Art	4
1.4 Contributions of the SCP Framework	4
1.5 Thesis Layout	4
2 Mathematical Preliminaries	5
2.1 Propositional Logic	5
2.2 Logic Programming	5
2.3 Non-Monotonic Logics	5
2.4 The Weak Completion Semantics	6
2.5 Reiter's Default Logic	6
3 Experiments	9
3.1 Suppression Task	9
3.1.1 Intuitive Overview	9
3.1.2 Mathematical Modelling the Suppression Task	9
3.2 Wason Selection Task	10
3.3 More	14
4 Sequential Cognition Processes	15
4.1 SCPs: An Intuitionist Description	15
4.2 SCPs: Mathematical Formulation	15
4.3 SCPs vs SCP Tasks	16
4.4 Epistemic States	17
4.5 State Points	17
4.6 Cognitive Operations	17
4.7 Pre-conditions and Effects	18
4.8 The Categorization Variable	19
4.9 Search in SCP-Space	19
4.9.1 Optimality, Satisfaction, Validity	20
Validity	20
Satisfaction	20
Optimality	20
4.9.2 De-Novo SCP Searching	20
4.9.3 Insertion Search	20

5	Cognitive Operations: The Cognitive Toolbox	21
5.1	Overview	21
5.2	Cognitive Operation Space	22
5.3	A Limited Set of Allowable Operations	23
5.3.1	A Limited Set of Cognitive Operations	23
5.3.2	A Limited SCP Length	24
5.4	Purpose-Selected Cognitive Operations	24
5.4.1	Overview	24
5.4.2	Initialisation	24
5.4.3	Variable Insertion	25
5.4.4	Variable Deletion	25
5.4.5	Variable Silencing	25
5.4.6	Variable Fixation	25
5.4.7	Adding Abnormalities	25
5.4.8	Weakly Completing	25
5.4.9	Semantic Operator	26
5.4.10	Default Inferencing	26
5.5	Cognitive Operations as Aggregates	26
5.6	Epistemic State Structure Changes with Cognitive Operations	28
5.7	Purpose-Built Cognitive Operations	29
5.7.1	Propositional Logic	29
5.7.2	Weak Completion Semantics	29
5.7.3	Default Logic	29
5.8	Two Approaches to Creating Cognitive Operations	29
5.9	Theoretical Approximation	29
5.10	Empirical Approximation	29
6	Mimicking Non-Monotonic Logics With SCPs	31
6.1	The Suppression Task	31
6.2	The Weak Completion Semantics	31
7	Modelling Experiments with SCPs	33
7.1	Overview	33
7.2	Suppression Task	33
7.2.1	Extending the Suppression Task with SCPs	34
7.3	Wason Selection Task	36
8	Comparing SCPs	37
8.1	Why do we need to compare SCPs?	37
8.2	Comparing Generated SCPs	38
8.2.1	Scoring	38
	SCP Length	38
8.3	Comparing External SCPs	39
8.3.1	The Needleman-Wunsch Algorithm	39
8.3.2	A Modified Scoring Algorithm	39
9	Program Design	41
9.1	Implementing the SCP Framework	41
9.2	A Modular Programming Approach	41
9.2.1	Most Important Modules	41
9.2.2	Program Flow	41

9.3 External Framework Integration	41
10 Non-Sequential Cognitive Processes	43
11 Conclusions and Future Work	45
11.1 Conclusion	45
11.2 Future Work	45
A Frequently Asked Questions	49
A.1 How do I change the colors of links?	49
Bibliography	51

List of Figures

3.1	The Abstract case of the Wason Selection Task	11
3.2	Using the abduction and contraposition extensions to derive all four canonical cases.	13
4.1	The most general description on an SCP with and without guaranteeing the WCS is applied at least once. An agent transitions from one epistemic state to another and then uses it to make a decision. <i>SP</i> nodes indicate state points. .	15
5.1	One possible formulation of a cognitive function to add abnormalities to variables. This formulation follows the intuition of Appendix @TODOref.	25
5.2	One possible formulation of a cognitive function to weakly complete a set of rules. This formulation follows the intuition of weak completion outlined in Section @TODOsec.	25
5.3	One possible formulation of a cognitive function to apply the semantic operator a cognitive state by temporarily setting all variables in V to any value that can computed using the semantic rule in Equation @TODOref. This process is repeated until V is no longer updated. @TODOredoThisDigramToShowActualProcess	26
7.1	A generalised illustration of the WCS in an SCP.	33
7.2	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. V_f represents the assignment of V in the epistemic state in the resulting least by model. $V_{i \in \mathbb{N}}$ represents the assignments in V after i iterations of the semantic operator.	34
7.3	The Suppression Task in which the additional operation of deleting the variable o occurs.	35
7.4	The Suppression Task in which the additional operation of fixing the variable ab_1 to false occurs.	36

List of Tables

2.1	Truth tables for standard operators in propositional logic.	5
2.2	A table showing the implication operator in 3-valued Łukasiewicz logic. . . .	6
3.1	The canonical results of the Wason Selection Task	10
3.2	Applying the Weak Completion Semantics to the <i>D</i> card of the WST.	11
3.3	Applying the Weak Completion Semantics to the <i>K</i> card of the WST.	12
3.4	Applying the Weak Completion Semantics to the <i>K</i> card of the WST.	12
3.5	Applying the Weak Completion Semantics and Contraposition to the 7 card of the WST.	13
4.1	SCP property requirements for precondition types in cognitive operations. . .	19
5.1	Solution space considerations of SCPs tasks as determined by length constraints and choice of cognitive operations.	23

List of Abbreviations

SCP	Sequential Cognitive Processes
WCS	Weak Completion Semantics
WST	Wason Selection Task

@TODO include chapters as shown below

Chapter 1

Introduction

1.1 Overview

The human mind is complex. So complex that thousands of approaches from dozens of fields have failed to capture its complexity. The sheer size of the brain – containing over 5000 times as many neurons as the largest practical neural networks (Mocanu et al., 2018) – and our limited understanding of the fundamental learning processes it employs mean that researchers can neither completely describe nor predict human actions nor model human thought processes. Instead, much of the current state of the art in cognitive modelling relies on one of two general approaches: creating systems that structurally approximate the human brain, and creating systems which approximate a more abstract intuition of cognition. The first type of system encompasses many algorithms related to machine learning and deep learning; the second type, with which this paper is concerned, has existed in some form for far longer than humans have been thinking creatures (Smirnova et al., 2015). Every shark stalking its prey, every tiny proto-mammal hiding from a hungry dinosaur, and every man driving to work in the morning, has applied this type of reasoning when trying to make predictions about the actions of other agents in their world. By applying case-specific reasoning to a known world state we are able to make imperfect, but quick, predictions about the mental state of other agents in our world.

Due in part to the difficulty of cloning dinosaurs to hunt participants in cognitive research experiments, and partly due to concerns about the ease with which they could answer questionnaires on the experience later, most cognitive tasks used by researchers tend to be more dull than the examples above.

Non-monotonic logics have proven able to adequately model a large number of standard cognitive reasoning tasks such as The Wason Selection Task (Wason, 1968) and Suppression Task (Byrne, 1989). These approaches, though effective and well-founded in isolation, are often unable to integrate or be compared to data models of other tasks, even when they rely on the same underlying logic. Although the non-monotonic logics themselves are generally carefully described, procedures ranging from best practise in deciding appropriate knowledge bases to the mechanisms by which abnormalities should be created tend to be re-imagined on a case-by-case basis.

Further, cognitive frameworks using non-monotonic logics are almost always designed to describe the most common (general) conclusion drawn by participants in the experiment. Modelling other individual reasoners or classes of reasoners who differ from the norm is often a non-trivial process. It has been shown that the Weak Completion Semantics (Hölldobler, 2015) is able to model the four most general cases of the Wason Selection Task under the assumption that reasoners who differ from the general case (*deviant reasoners*) follow a sequence of mental processes that is still highly similar to that of the general reasoner (Breu et al., 2019).

This paper introduces the Sequential Cognition Process (SCP) which generalises the assumption of sequential cognitive operations, each of which uses a collection of epistemic

states as input and produces a collection of epistemic states as output. Using SCPs and a set of well-founded cognitive operations it is possible to apply traditional search techniques to problems in cognitive modelling with non-monotonic logics that have previously required expert-made models. The SCPs introduce a number of desirable properties: they introduce a partially standardised (though extensible) set of allowable cognitive operations, they standardise the structure of what constitutes an epistemic state, they are easily modified to accommodate deviant reasoners when a well-founded general model already exists, and their sequential structure makes them well-suited to scoring algorithms that allow intra- and inter-experimental modelling and comparisons.

1.2 A Review of Terminology

1.3 Current State of the Art

1.4 Contributions of the SCP Framework

1.5 Thesis Layout

Chapter 2

Mathematical Preliminaries

2.1 Propositional Logic

Propositional (or classical) logic is the form of logic most traditionally associated with problem solving in mathematics and computer science. In propositional logic features of a hypothetical world are encoded by a set of *variables*. every variable must be mapped to either \top (usually denoting truth, in some subject-specific sense) or \perp (denoting falseness). A *clause* ϕ consists of either a single atomic variable or an operator and some set of sub-clauses. Every clause has a truth value determined by the values of the sub-clauses it contains and the relevant operator which handles their interaction. Table 2.1 illustrates the classical interpretations of the allowable operators. A knowledge base KB consists of a set of clauses. In this paper we adopt the assumption all clauses (also called rules) $\in KB$ are of the form $head \leftarrow body$ and all heads are atomic variables.

2.2 Logic Programming

2.3 Non-Monotonic Logics

In the field of non-monotonic logics, reasoning is represented as a collection of defeasible inferences. Unlike in classical logic, conclusions need not hold in perpetuity, or even in the same model and revision is always possible. Monotonic logics are not capable of describing human reasoning in experiments like the Suppression Task (Dietz, Holldobler, and Ragni, 2012) because they lack this revisionist characteristic.

A large number of non-monotonic frameworks exist in the literature (McDermott and Doyle, 1980), each applicable to a different subset of cognitive problem space, and each modelling their problem space with various degrees of success. In the simplest formulation, a non-monotonic logic is simply an extension to a classical logic which introduces a preference relation \rightarrow_p . This preference relation states that, given some number of derivable facts in a knowledge base, the fact derived using the most preferential rule is to be derived first and cannot be overwritten by a less preferable assignment.

\wedge	\top	\perp	\vee	\top	\perp	\neg	\top	\perp
\top	\top	\perp	\top	\top	\top	\top	\perp	\perp
\perp	\perp	\perp	\perp	\top	\perp	\perp	\perp	\top
\rightarrow	\top	\perp	\rightarrow	\top	\perp			
\top	\top	\perp	\top	\top	\perp			
\perp	\top	\top	\perp	\perp	\perp			

TABLE 2.1: Truth tables for standard operators in propositional logic.

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

TABLE 2.2: A table showing the implication operator in 3-valued Łukasiewicz logic.

Although the non-monotonic logics discussed in this section has simple extensions to first-order logic, we instead restrict ourselves to a propositional format throughout this thesis.

2.4 The Weak Completion Semantics

The Weak Completion Semantics is a non-monotonic logic which procedurally encodes several well-known cognitive phenomena. The WCS makes use of 3-valued Łukasiewicz logic (Table 2.2). 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 ϕ_{svL} (Stenning, Van Lambalgen, et al., 2008). Let J be the result of applying the semantics operator to an interpretation I and logic program P . Then 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 of P ($\text{Im}_L wcP$) can be calculated by iterating $\phi_{svL, P}$.

2.5 Reiter's Default Logic

Reiter's Default Logic (Reiter, 1980) is a non-monotonic framework which allows us to divide the inferential capabilities of a system into those facts and inference rules which are always true (as in classical logic) and those which are usually true. The second type of inference is the eponymous default rule.

The set of immutable facts is called kb and inferences in kb are made as with propositional logic, with the addition of the conclusion of any previously evaluated default rules. The set of default rules d consists rules of the form:

$$\frac{\text{prerequisite} : \text{justification}_1, \dots, \text{justification}_n}{\text{conclusion}}$$

@TODOextend.

Where *prerequisite*, *justification_i*, and *conclusion* are propositional logic clauses. A *conclusion* is valid and added to the set of known variables in *kb* iff *prerequisite* holds and no clause *justification_i* holds.

Formally, inferences in default logic are calculated as follows:

@TODOref

It should be immediately apparent from this formulation that, for many possible sets of rules in *kb*, the output is non-monotonic as is dependent on the order in which default rules are evaluated.

A very significant restriction on reiter's default logic is the complexity of computing inferences. Even in the simple case of $kb = rule_1, \dots, rule_n$ and $d =$, the set of possible inferences is *NP-complete* as for a propositional formula @TODOnpcompleteref. When default rules are included in this computation the complexity of calculating all possible inferences for all possible default rule application sequences quickly becomes intractable for all but the most basic logic programs. With this limitation in mind, we restrict propositional rules to those in the form $a \leftarrow \phi$ where *a* is an atomic head. Further, we restrict the set of default rules to those of the form $\frac{B:C}{a}$ where *a* is an atom.

Chapter 3

Experiments

3.1 Suppression Task

3.1.1 Intuitive Overview

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.

3.1.2 Mathematical Modelling the Suppression Task

Work by Dietz, Hölldobler, and Wernhard, 2014 has shown that the WCS is an adequate non-monotonic logic for modelling the Suppression Task. Under the WCS 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:

	p	pq	$pq\bar{q}$	$p\bar{q}$
Abstract	36	39	5	19
Everyday	23	37	11	29
Deontic	13	19	4	64

TABLE 3.1: The canonical results of the Wason Selection Task

- 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: $P = \{e \wedge \neg ab_1 \rightarrow l, o \wedge \neg ab_2 \rightarrow l, \top \rightarrow e, \neg o \rightarrow ab_1, \neg e \rightarrow ab_2\}$. An abnormality ab_i is an addition to an inference which captures the idea that external factors may invalidate the conclusion of the original inference. 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 ??.
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.

3.2 Wason Selection Task

Another widely studied task in the psychological literature is the Wason Selection Task (WST), which asks participants to draw conclusions about which variables are able to falsify a given rule (*Modus Tolens*¹). We will look at this task in terms of the Abstract case of the task (Wason, 1968). Each case is identical to each other in terms of classical logic representation, but each is interpreted differently by subjects. The WST is often used to illustrate that humans do not follow classical logic.

Abstract Case

The Abstract case of the WST presents the subject with four cards on a table. Their face-up sides read $D, K, 3$, and 7 . They are asked which cards must be turned over to test the rule:

“If there is a D on one side of the card, then the other side shows 3 .”

¹If $a \rightarrow b$, then $\neg b \rightarrow \neg a$

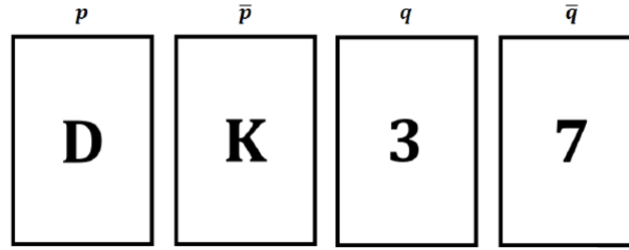


FIGURE 3.1: The Abstract case of the Wason Selection Task

Iteration	\top	\perp
0		
1	D	ab
2	$D, 3$	ab

TABLE 3.2: Applying the Weak Completion Semantics to the D card of the WST.

Using classical logic, it is easy to see that turning D to find anything but 3 would invalidate the rule, as would turning 7 and finding a D . Thus these two cards must be turned to check the rule. However, human subjects very seldom choose this classical answer set and instead overwhelmingly prefer to turn D and 3. Table 3.1 shows the most common card selection sets for participants.

A great many cognitive theories have been applied to these results, with varying degrees of success **ragni2017formal**. This paper will focus on the WCS interpretation of the task provided by **(ragni2017wason)**. This approach assumed an initial knowledge base containing only the rule $3 \leftarrow D$.

To illustrate this interpretation of the task we shall use the WCS to examine each of the four cards individually and see what conclusions can be drawn.

When D is observed:

$$KB = \{3 \leftarrow D, D \leftarrow \top\}$$

However, at this point, the authors introduce the concept of an abnormality as discussed in **(ragni2017formal)**. Now, the rule holds only provided that some abnormal even does not occur.

$$KB_D = \{3 \leftarrow D \wedge \neg ab_1, D \leftarrow \top, ab \leftarrow \perp\}$$

$$wcKB_D = \{3 \leftrightarrow D \wedge \neg ab_1, D \leftrightarrow \top, ab \leftrightarrow \perp\}$$

The authors also require that, to validate the rule, it must be evaluated after the semantic operator is iterated. Table 3.2 shows the result of repeatedly applying the semantic operator to KB_D .

When the K card is observed:

$$KB_K = \{3 \leftarrow D \wedge \neg ab_1, K \leftarrow \top, ab \leftarrow \perp\}$$

$$wcKB_K = \{3 \leftrightarrow D \wedge \neg ab_1, K \leftrightarrow \top, ab \leftrightarrow \perp\}$$

Iteration	\top	\perp
0		
1	K	ab
2	K	ab

TABLE 3.3: Applying the Weak Completion Semantics to the K card of the WST.

Iteration	\top	\perp
0		
1	3	ab
2	3	ab

TABLE 3.4: Applying the Weak Completion Semantics to the K card of the WST.

Table 3.3 shows that not enough information can be obtained using only the semantic operator to evaluate the rule $D \leftrightarrow 3$. This is identical to the case for the 7 card.

When the 3 card is observed:

$$KB_3 = \{3 \leftarrow D \wedge \neg ab_1, 3 \leftarrow \top, ab \leftarrow \perp\}$$

$$wcKB_3 = \{3 \leftrightarrow D \wedge \neg ab_1, 3 \leftrightarrow \top, ab \leftrightarrow \perp\}$$

In this case, something odd happens. Even though the WCS framework outlined is not able to deduce anything about the truth value of the variable D , participants still very often chose to turn this card. In Breu et al., 2019 we argue that this may be a result of implicit *abduction*. That is, that the participant subconsciously attempts to identify any possible way in which $3 \leftrightarrow D \wedge \neg ab$ could hold when $3 = \top$ and $ab = \perp$. The only way for this rule to hold would be for D to be true too:

Now, using Table 3.4 one can see that 3 is set to true, the abnormality is set to false, and D has been assumed via abduction. Following Łukasiewicz logic these assignments are sufficient to evaluate $3 \leftarrow D \wedge \neg ab$ to \top . This the subject will conclude to turn the card.

This simple interpretation of the WST using the WCS and using or not using abduction is sufficient to achieve the most common (general) reasoning of participants. It does not however explain any of the other deviant cases which are observed in Table 3.1. Why, for example, should it be considered accurate if it gives no information about the classically accurate choice of the D and 7 cards, chosen by 19 participants, a significant portion? Instead, consider how these aberrant reasoners could be modelled. Perhaps they use extra computational steps or omit steps? This author has previously shown that the WCS is able to model these individual reasoners (Breu et al., 2019) by including two simple processes: Abduction and Contraposition. By including these two processes, and stochastically controlling when they are activated or silenced, it has been shown that not only is this extended framework able to model the major cases of the WCS, but very close approximations to empirical results can also be drawn.

As discussed above, Abduction can be used when the rule under consideration evaluates to unknown. It uses fixing of remaining free variables to determine if any combination of these can validate or falsify the rule. In the case we have discussed, only fixing $D = \top$ will validate the rule.

Iteration	\top	\perp
0		
1	7	ab_1, ab_2
2	$K, 7$	$D, 3, ab_1, ab_2$

TABLE 3.5: Applying the Weak Completion Semantics and Contraposition to the 7 card of the WST.

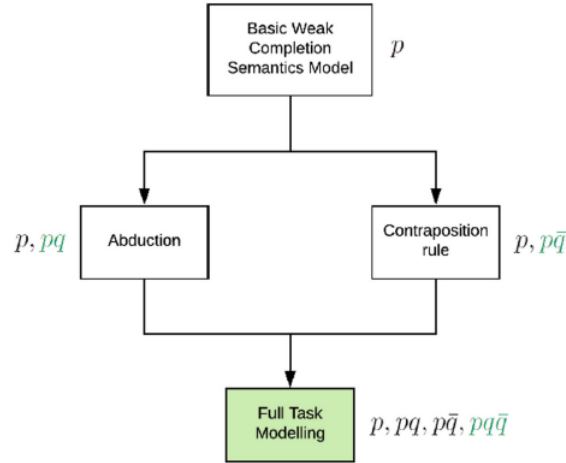


FIGURE 3.2: Using the abduction and contraposition extensions to derive all four canonical cases.

Contraposition explicitly makes use of *modus tolens*, usually assumed to be silenced in human cognition to derive certain canonical cases in the WST. In particular, for the card 7, the logic program:

$$KB_7 = \{3 \leftarrow D, 7 \leftarrow \top\}$$

is extended with the rule $\neg D \leftarrow \neg 3$, which itself simply means $K \leftarrow 7$ for our restricted domain.

$$KB_7 = \{3 \leftarrow D, \neg D \leftarrow \neg 3, 7 \leftarrow \top\}$$

Adding abnormalities yields:

$$KB_7 = \{3 \leftarrow D \wedge \neg ab_1, \neg D \leftarrow \neg 3 \wedge \neg ab_2, 7 \leftarrow \top, ab_1 \leftarrow \perp, ab_2 \leftarrow \perp\}$$

After weakly completing:

$$KB_7 = \{3 \leftrightarrow D \wedge \neg ab_1, \neg D \leftrightarrow \neg 3 \wedge \neg ab_2, 7 \leftrightarrow \top, ab_1 \leftrightarrow \perp, ab_2 \leftrightarrow \perp\}$$

Applying the semantic operator then yields:

Table 3.5 shows the result of applying the semantic operator to the resulting knowledge base. From this, it is possible to deduce that the card 7 must be turned. Figure 3.2 illustrates the use of these extensions to derive all four of the canonical cases of the WST.

@TODOrewritewholesectionintermsofpq

3.3 More

Chapter 4

Sequential Cognition Processes

4.1 SCPs: An Intuitionist Description

Although there is evidence that the brain can perform several simultaneous operations when considering a task (such as when considering an image (Sigman and Dehaene, 2008)), the SCP framework assumes that at some points in reasoning about a given task, the mental processes of the agent converge to a set of epistemic states, called a *state point*. Whatever happens between these points of convergence can contain any number of parallel processes. The collection of processes that occur between any two state points in a reasoning task is called a *cognitive operation*. It follows that any cognitive operation is valid as long it takes a set of epistemic states as input and produces a set of epistemic states as output. Figure 4.1a describes an SCP that is powerful enough to model any cognitive task that involves an epistemic state transition. However, it does not provide any useful information; the nature of the processes followed is completely undescribed. Suppose, instead, that some cognitive task is being modelled, and that researchers have reason to believe that The Weak Completion Semantics should play a part in their model. Under this new restriction, and assuming a sufficiently expressive epistemic state, Figure 4.1a is still an accurate model of the process, but now so is Figure 4.1b. By sacrificing some of the ambiguity – and, thus, expressiveness – of the model, the information content of the model description has increased. This trade-off is a feature of the SCP framework and finding the right depth of complexity to model the task accurately and still provide meaningful information is more art than science at present.

4.2 SCPs: Mathematical Formulation

An SCP Task $\Pi = (s_i, M, f(), \gamma)$ consists of an initial epistemic state s_i , a known set of cognitive operations M , a desired final output γ , and an external function $f()$ which generates those final outputs by translating the final epistemic state into an empirically grounded set of possible responses. An epistemic state s_k describes all the information available to the agent at state point k . The precise contents of an epistemic state should be chosen so that at least some $m \in M$ are able to accept that state as an input. In the case of a system

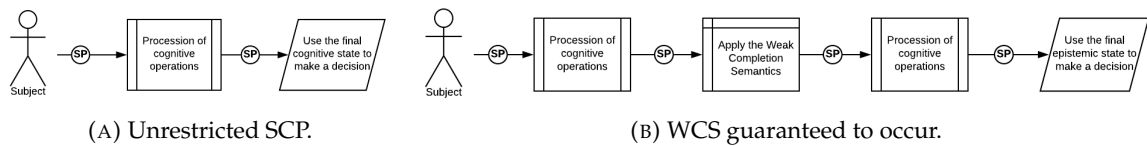


FIGURE 4.1: The most general description on an SCP with and without guaranteeing the WCS is applied at least once. An agent transitions from one epistemic state to another and then uses it to make a decision. *SP* nodes indicate state points.

containing only the one complex operation which applies the WCS, one possible epistemic state is $s = (KB, V)$, where KB describes a knowledge base consisting only of rules of the ($head \leftarrow body$) and V is a mapping of atom names appearing in KB to a truth value in Łukasiewicz logic (\top, u, \perp). In principle this definition will serve for the rest of the paper, but it is extended slightly so that $s_k = (KB, V, R)$ where R is a set of labelled categorization criteria sets (*LCS*). An *LCS* $c \in R$ consists of a category name and list of rules and atoms which fit into that category.

A base point is a single epistemic state. A state point p is defined recursively by $p = \{\bar{p} \oplus Q\}$ where \bar{p} is a base point, and Q is a set of state points, and \oplus represents the exclusive-or operation¹. State point containment \in_s for state points p and q is defined recursively as follows:

$$p \in_s q = \begin{pmatrix} p \in q & \text{True} \\ \exists_{r \in p} p \in_s r = \text{True} & \text{True} \\ \text{Otherwise} & \text{False} \end{pmatrix}$$

It is never the case that $p \in_s p$.

A cognitive operation $m = (\chi, e)$, $m \in M$ consists of a precondition χ and a process e , such that for an input state point p , every base point $\bar{p} \in_s p$ is either accepted as input ($\bar{p} \models m[\chi]$ under whatever definition of \models is used for the complex operation m), or else rejected. Every base point is evaluated by the complex operation in isolation (no other base point \bar{q} can affect the output of m on base point \bar{p}). To capture the fact that cognitive operation may utilize non-monotonic logic, applying m to an input base point $\bar{p} \in p$ always yields a state point p' (written $J[\bar{p}, m] = p'$) which describes all possible resulting epistemic states that can be generated.

Applying m to an input state point p is done by replacing every base point $\bar{p} \in p$ with $J[\bar{p}, m]$. A cognitive operation is called monotonic if it always yields a base point as an output given a base point input ($J[\bar{p}, m] = \bar{p}'$). It follows that the depth of a state point is directly related to the number of complex operations which have been performed in the SCP prior to its occurrence. If a base point does not meet the precondition, it is either ignored completely and not processed (*cruel application*), or passed exactly as is to the next complex operation (*lenient application*). It is worth noting that the type of cognitive states produced as output by m may not be the same as those of the input.

This paper will focus on cases where the type of cognitive state remains constant, but there is no reason in principle that a base point input $s_k = (KB, V, R)$ for the WCS could not be returned as a base point $s_k = (KB, V, R, D)$ where D is a set of default rules usable by cognitive operations that implement Reiter's Default Logic (Reiter, 1980). Future models of human cognition may well rely on background knowledge which draws inferences from multiple types of non-monotonic logics.

An SCP $\pi = (s_i, m_1, \dots, m_n)$ describes an initial epistemic state (or state point if the input is uncertain) and state point p_k is defined recursively by $p_k = J[p_{k-1}, m_k]$. An SCP is called *credulously valid* if $f(p_n) \models \gamma$ for at least one epistemic state in the final state p_n . An SCP is called *sceptically valid* if $f(p_n) \models \gamma$ for every epistemic state in the final state. In cases where all operations are monotonic, sceptical validity is the same as credulous validity.

4.3 SCPs vs SCP Tasks

The distinction between SCPs and SCP tasks is fairly straightforward: An SCP task describes a cognitive phenomena that has been observed and details the tools and constraints which

¹ $(X \oplus Y) = ((X \cup Y) - (X \cap Y))$ for sets X and Y .

may be used to model it; and an SCP is a single solution to the problem described in the SCP task which does not violate conditions outlined in that task.

For a given SCP task there can be many possible SCPs (the question of how to choose the most probable of these candidate SCPs is discussed in Section @TODOref), or no possible SCPs at all.

4.4 Epistemic States

The choice of epistemic state is dependent on the properties that are known or suspected to be true for the cognitive task as a whole. For example, a researcher working on drawing inferences using Propositional Logic can be certain that any SCP they create should be expressive enough to pass a knowledge base to a cognitive operation. Thus, it might suffice to simply define $s_i = (KB)$ where KB is a set of propositional rules. By contrast, a researcher using the WCS requires a system capable of both communicating a set of rules to the next complex operation, and of describing the results of repeated applications of the semantic operator. It might seem intuitive to simply keep the epistemic state used for the propositional case and to use a non-complex operation function on the weakly completed program to find the least model. And, though this approach works for many examples, it makes it impossible to perform further processing in the SCP *after* applying the semantic operator. What if the conclusion drawn was meant to form part of the background knowledge of another process? In practice using $s_k = (KB, V, R)$, where V is a set of (variable name, value pairs) and R is a categorization variable, is able to model all aspects of the WCS, including information related to the least model.

As yet, there are no definite rules for creating an epistemic state, but Albert Einstein's famous advice from 1950 still rings true: "Everything should be made as simple as possible, but no simpler." The ideal epistemic state is one that enables every reasonable cognitive operation in M that might help model the problem, without adding superfluous functionality that might render searching the SCP space infeasible.

4.5 State Points

4.6 Cognitive Operations

The set of possible complex operations M determines many attributes of the achievable final state point p_n . If every $m \in M$ is monotonic, then p_n will be monotonic. If some cognitive operation is computationally complex or produces a very large number of output state points, then search using that base point becomes less efficient. If some complex operation m' (such as weakly completing) is known or believed to occur in the SCP, then a restriction on the cognitive states exists such that either the initial state is of a format suitable as input for m' , or there exists another cognitive operation which is able to output a state point which contains base points of a suitable format.

More abstractly, the set of cognitive operations should be well-founded in the literature. The set of possible complex operations is infinite and an SCP only meaningfully describes human cognition when it contains cognitive operations that have been justified empirically (*modus ponens-modus tollens* asymmetry, suppression, denial of the antecedent, etc.).

4.7 Pre-conditions and Effects

The precondition χ of a cognitive operation A refer to those conditions the input state point must satisfy in order for that operation to be considered valid. An SCP is valid if and only if every cognitive operation it contains is valid. For example, one might have a cognitive process describing Julie's plans for a night out on the town. Let us imagine that the SCP task describing her night out includes the operation `goHomeByCar`. Semantically, this operation should take her *isHome* variable and set it to True.

The situations in which `goHomeByCar` can reasonably occur in an SCP are at the researchers' discretion. Researcher 1 might feel that it can be allowed at any point in the SCP and will simply have no effect for those input epistemic states in which she is already at home. Researcher 2 might argue that `goHomeByCar` is applicable only when *isHome*=False. Researcher 3 knows that only the cognitive operation `goToClub` changes the *isHome* variable, and so argues that *isHome* is only applicable after `goToClub` occurs. There are merits to the arguments of each researcher.

Researcher 1 argues for a property called *trivial validity*, that is that SCPs should always be considered valid, without any need evaluation. This approach, however has one significant drawback: it cannot handle changes in the structure of epistemic state inputs. Imagine an epistemic operation called `dontDriveDrunk` which corresponds to our party animal realising that she shouldn't drive home if she's been drinking. Imagine further that this operation takes base points which are either of propositional or default structure (@TODOrefsection) and outputs an epistemic state of default structure which contains the new rule. If `goHomeByCar` took only a propositional state as input, then any sequence where `dontDriveDrunk` occurs as the previous operation will cause the SCP to fail because of the input state is not of an allowed type. Trivial validity then, is not sufficient for modelling SCP which state structure changes.

Researcher 2 has opted for a *variable validity* approach. She has reasoned, correctly, that the `goHomeByCar` only results in an epistemic state change when when *isHome* is not True. She therefore, feels that it only makes sense to reduce redundancy and search complexity in creating the SCP by only allowing the action to occur when it can be said to have an effect. This argument has some merit from an intuitive perspective, but presents an unpleasant question: what if there are state points in which only some ground points actually meet the precondition? To compound the troubles with this variable state approach to preconditions is the fact that it is not possible to determine SCP validity without evaluating the SCP at runtime, which could be slow for large SCPs.

Researcher 3 has taken an *operator validity* approach, instead has focused on the structure of the SCP. This approach allows SCP validity to be determined without explicitly evaluating an SCP, one need simply search through the cognitive operations in the SCP to make sure that no operation has a precondition operation which has not yet occurred. This approach also presents drawbacks, it requires explicit knowledge of operation interactions, and adding a new cognitive operation to M in the SCP Task might force several other operations to update or change their interactions. Further, there might be other cognitive operations which mean that the precondition operation is no longer in effect. Imagine a third operation `goHomeByTrain` which also sets *isHome* to True. Now the sequence of operations $x \mapsto \text{goToClub} \mapsto \text{goHomeByTrain} \mapsto \text{goHomeByCar}$ would still be valid with Researcher 2's original requirement that `goToClub` occurred previously in SCP. However, it is obvious that `goHomeByTrain` has negated the effect of `goToClub`. Evidently this approach is not appropriate in any case in which other operations can silence the effect of those operations mentioned in preconditions.

Every approach has shows strengths and weaknesses, a final approach to consider is the *hybrid validity* approach. With this approach, all cognitive operations are assumed to

Validity	Full SCP evaluation	Uniform Epistemic Structure	Operator Silencing Knowledge
Trivial	X	✓	X
Variable	✓	X	X
Operator	X	X	✓
Hybrid	X	X	X

TABLE 4.1: SCP property requirements for precondition types in cognitive operations.

be valid, provided that the output base points of the previous operation are of a suitable input type for the current operation. Hybrid validity is an appropriate approach for all SCPs in which every cognitive operation has a known output structure. Though hybrid validity does not have the best-case search properties of the other approaches, its universal validity means that it should generally be the starting point for generating SCPs. The hybrid approach is followed implicitly throughout the rest of this thesis and preconditions are omitted.

4.8 The Categorization Variable

The final property of a cognitive operation that needs to be discussed is how it is able to interact with the categorization variable R . Imagine a case drawn from Saldanha, Hölldobler, and Rocha, 2017 where the difference between creating abnormalities for obligate and factual conditionals is discussed. The intuition behind the authors' work can be summarised by saying that there are two different types of conditional statement, those that *have to be* true, and those that are *usually* true. If it rains r , I will normally take my umbrella u (in the absence of something abnormal happening, like a plague of umbrella-stealing gnomes). This is a statement that is usually true, but some things are definitely true. For example, when it rains, water has to fall out of the sky s . This is an essential property of rain, not subject to abnormalities. Thus one useful set of categorizations for researchers seeking cosmic knowledge of weather patterns may be: $R = \{\text{obligate} : \{u \leftarrow r\}, \text{factual} : \{s \leftarrow r\}\}$. With these labels we might then expect that the process followed by the operation $m \in M$ which creates abnormalities would treat the two conditionals in KB differently, because of their assignments in R . R then is a way of expressing meta information to the cognitive operations, and it is completely possible that some operation m_k might change R in such a way that future operation m_{k+1} produces different output. This is a technique we will exploit in several examples in this paper.

4.9 Search in SCP-Space

As with any data structure in which one input can produce one or multiple outputs, it is possible to search through SCP space in order to find those SCPs which meet certain criteria. Those criteria might be *validity* (Section 4.7), *satisfaction* (conditions in γ are satisfied), or *optimality* (there exists no better solution to this problem).

SCPs lend themselves particularly well to forward search techniques (@TODOref) but also have some potential using backwards or bidirectional search (@TODOref, @TODOref).

Searching through solutions to an SCP task takes one of two forms *De Novo search*, and *Insertion search*. De Novo search generates an SCP that meets the optimality, satisfaction, or validity requirements of the researcher from scratch, using only the information contained

in the planning task. Insertion Search changes an existing SCP which models a particular response in order to model a reasoner with differing responses. Section 4.9.2 and Section 4.9.3 discuss the philosophy, applications and mechanical considerations of these two search approaches.

4.9.1 Optimality, Satisfaction, Validity

Validity

As discussed in Section 4.7 validity for an SCP is defined as follows:

An SCP $\pi = (\pi_0 \mapsto \dots \mapsto \pi_n)$ generated from SCP Task $\Pi = (s_i, M, \gamma, f())$ with evaluation function $h()$ is valid iff s_i is an epistemic state, every $m \in \pi$ is valid (according to the validity requirements defined by the researcher), and $f(\pi)$ is defined.

Validity does not require that γ holds in the final state π_n , only that external function $f()$ is able to make some prediction or set of predictions based on that final epistemic state point.

Satisfaction

An SCP $\pi = (\pi_0 \mapsto \dots \mapsto \pi_n)$ generated from SCP Task $\Pi = (s_i, M, \gamma, f())$ with evaluation function $h()$ is satisfying iff s_i is an epistemic state, every $m \in \pi$ is valid (according to the validity requirements defined by the researcher), and $f(\pi) \models \gamma$. @TODOchangedefinitionofGammaEVERYWHERE!

Heuristic Searches and machine learning techniques are generally used to find satisfying solutions for situations in which it is possible for an answer to be good enough for practical purposes.

Optimality

Optimality refers to finding the best possible SCP to describe a problem according to whatever criteria are used to evaluate the SCP (Section 5.3.1). Loop-free SCPs are candidates for exhaustive search techniques to find optimality, but in practice, loop free SCPs can seldom be guaranteed when there are many cognitive operations in M @TODOproof?. As with all exhaustive search techniques, optimality can be guaranteed for searches of restricted depth even when SCP space contains infinite loops.

Formally, an SCP $\pi = (\pi_0 \mapsto \dots \mapsto \pi_n)$ generated from SCP Task $\Pi = (s_i, M, \gamma, f())$ with evaluation function $h()$ is optimality iff $f(\pi)$ is satisfying and there exists no SCP π' such that $g(\pi) < g(\pi')$.

In an SCP context a solution may be optimal for a given task or set of empirical data, but only globally valid or satisfying. A significant part of the appeal of the SCP framework is the potential to use high-scoring local solutions to several tasks or from several reasoners and to predict which are most likely by searching for evidence of repeated structures in the disparate solutions.

4.9.2 De-Novo SCP Searching

4.9.3 Insertion Search

Chapter 5

Cognitive Operations: The Cognitive Toolbox

5.1 Overview

As discussed in Section 4.1, and illustrated in Figure @TODOref the simplest imaginable description of sequential human cognition consists of a single step transforming the initial task information (along with background information) into an epistemic state that can be externally interpreted to choose a response. Assuming that this single operation A transforms the initial knowledge base x into final state x' which is interpreted by external function $f()$, we can use the shorthand $f(x \mapsto A)$, where \mapsto is a function that takes state point x as input and produces a new state point y as output, to summarise this process.

To understand why this is poor choice of SCP, consider a general limitation of the SCP framework: we can only change the output of an SCP by changing the initial epistemic state, or by inserting/deleting a cognitive operation at an existing state point. Consider a case where the initial epistemic state is immutable. Thus, for $f(x \mapsto A)$, the only possible changes are: $f(x \mapsto \bar{B} \mapsto A \mapsto \bar{B})$, $f(x \mapsto A \mapsto \bar{B})$, $f(x \mapsto \bar{B} \mapsto A)$, $f(x \mapsto \bar{B})$, and $f(x)$. Where \bar{B} denotes the action of drawing some sequence of random, but valid cognitive state transitions. There is no way to change the operations that occur in A . Thus, even though Figure @TODOref may be an appropriate model for the general reasoner of a given task, it imposes the explicit assumption that all other test subjects who achieved differing results, achieved only by completely ignoring the black-box process the general reasoners used, or else by inserting new operations only at the very start or end of the processing task. We will see an example in Section @TODOref of cases where assumptions like that are simply not reflected by empirical data.

Instead of using a single black-box operation to describe all cognition, it follows logically to use multiple smaller black box operations, each connected by a state point. Using this interpretation, cognitive operations for the general reasoner should be as extensive as possible, provided that they do not force implausible changes to describe deviant users or when used in other cognitive tasks. Figure @TODOref captures this intuition as it applies to the Weak Completion Semantics. Indeed, as a non-monotonic logic that is already explicitly divided into discrete steps in the literature @TODOref, obvious candidate cognitive operations present themselves.

It is worth noting that this increased ability to model deviant reasoners comes at the cost of forcing more state points into the SCP being used.

The remainder of this section is devoted to expressing the exact properties, limitations, and concrete descriptions of cognitive operations.

@TODOlayout

5.2 Cognitive Operation Space

In principle, the number of cognitive operations available for use in an SCP is unlimited. We denote the space of all cognitive operations Ω , and the space of all sequences $(A_0 \mapsto \dots \mapsto A_n)$ with Ω^* . There cannot and never will be an exhaustive search procedure to evaluate every possible way of manipulating an epistemic state (Proofs 1 and 2 illustrate the existence of valid SCPs for infinitely many different cognitive operations and valid SCPs of any length, respectively). However, there are ways of limiting the number of allowable cognitive operations. Lemma 1 defines those cases in which there is redundancy in the set of all possible cognitive operations, Lemma 2 focuses this concept on specific cognitive tasks which are to be modelled using SCPs.

Proof 1 *There are an infinite number of possible, valid SCPs, that can meet any goal condition γ from input x provided that at least one SCP exists that can reach goal condition γ from input x .*

1. *There exist infinitely many cognitive operations $T \in \Omega$ which add a new variable from the set of all possible variable names $p \in P$.*
2. *There exist infinitely many cognitive operations $V \in \Omega$ which remove a variable from the set of all possible variable names $p \in P$.*
3. *It follows that there exist infinitely many pairs $(T_i \in T, P_i \in P) \in W$ where T_i adds a variable $p \in P$ to the resulting state point, and V_i removes that variable from the resulting statepoint.*
4. *Then $f(x \mapsto A) = f(x \mapsto T_i \mapsto V_i)$*
5. *Thus it follows that if $f(x \mapsto A) \models \gamma$ then $f(x \mapsto T_i \mapsto V_i) \models \gamma$ for all $(T_i \in T, P_i \in P) \in W$.*

Proof 2 *If there exists an SCP which meets goal condition γ of length n , then there exists an SCP that meets goal condition γ of length $n + k$ where k is any natural number.*

- *Proof 1 states that if $f(x \mapsto A) \models \gamma$ then $f(x \mapsto T_i \mapsto V_i) \models \gamma$ for all $(T_i \in T, P_i \in P) \in W$.*
- *Let X be a partial SCP of the form $X = T_0 \mapsto V_0 \mapsto \dots \mapsto T_N \mapsto V_N$ where $(V_i, T_i) \in W$*
- *For odd lengths:*
 - *It follows that if $f(x \mapsto A) \models \gamma$ then $f(x \mapsto A \mapsto \text{exp}(X)) \models \gamma$, where exp denotes the substitution of X for its constituent parts.*
 - *Because X always contains an even number of cognitive operations, then the length of the resulting SCP must be $n' = 1 + v$ where v is any even number.*
- *For even lengths:*
 - *Notice that $f(x \mapsto A \mapsto \text{exp}(X) \mapsto X) \models \gamma$.*
 - *A appending $\mapsto X$ to adds one to the length of any SCP.*
 - *Because we have shown that a suitable SCP of any odd length exists $n' \geq n$, it follows that a suitable SCP of any even length $n' > n$ exists.*

Lemma 1 *Given a cognitive operation sequence $A \in \Omega^*$, and external function $f()$, A is redundant iff one of the following holds:*

M	Finite Length Constraint	Finite Solution Space	Guaranteed Solution
Ω^*	None	X	✓
Ω^*	Yes	X	✓
Ω	None	X	✓
Ω	Yes	X	✓
finite	None	not guaranteed	X
finite	Yes	✓	X

TABLE 5.1: Solution space considerations of SCPs tasks as determined by length constraints and choice of cognitive operations.

- $(x \mapsto B \mapsto C) \models x'$ and $(x \mapsto A) \models x'$ for every viable epistemic state x , $B \in \Omega^*$, $C \in \Omega^*$.
- $x \mapsto A \models x$ for every viable epistemic state x .
- $f(x \mapsto B) = c$ for all $B \in \Omega^*$, where c is a constant external decision.

Lemma 2 Given a limited set of cognitive operations $M = \{A_0, \dots, A_n\}$, $A_x \in \Omega^*$, and external function $f()$, $A \in M^*$ is task redundant iff one of the following holds:

- $(x \mapsto B \mapsto C) \models x'$ and $(x \mapsto A) \models x'$ for every viable epistemic state x , $B \in M^*$, $C \in M^*$.
- $x \mapsto A \models x$ for every viable epistemic state x .
- $f(x \mapsto B) = c$ for all $B \in M^*$, where c is a constant external decision.
- There exists no epistemic state x and sequences $B, C \in M^*$ such that $f(x \mapsto B \mapsto A \mapsto C)$ is a valid SCP.

The total number of possible non-trivial SCPs of length n for an SCP Task is given by $(|M|^L - R)$ where $|M|$ is the number of elements in the set of cognitive operations M , L is the length of the SCP, and R is the number of possible non-overlapping task redundant SCPs for the task. @TODOdefineOverlappingRedundantSCPs

5.3 A Limited Set of Allowable Operations

In practice it is impossible compute the set of all candidate SCPs when the set of allowable cognitive operations M in a planning task is Ω^* or even Ω . Without a length constraint Proof 2 shows that it is not even possible to compute all possible SCPs for many SCP tasks with a finite set of allowable operations. Table 5.1 summarises the properties of solution space under different SCP task conditions.

5.3.1 A Limited Set of Cognitive Operations

Readers should recognize that using the space of all possible cognitive operations for M results in a computationally impossible task. No matter the other conditions imposed there will be infinitely many solutions to the given SCP task which satisfy $f()$.

But, beyond being just computationally infeasible, there are more empirical and philosophical objections to this approach. The first is that the human brain itself does not have infinite processing or space capabilities and so any model which allows for unrestricted data

sizes in the set of allowable cognitive operations M or in the number of resultant state points M is not biologically well-founded¹.

The second consideration is that there is considerable evidence in the literature that many problems can be modelled using logical frameworks which describe cognitive behaviour across a variety of tasks @TODOref. These frameworks reflect a more generally philosophy in cognitive psychology which is the idea of consistent cognitive motifs in reasoning. Using an infinite M shows no discrimination at the time of task-formulation between those operations for which there is significant evidence and those which are fundamentally improbable.

5.3.2 A Limited SCP Length

As discussed in Section 5.3.1, the human brain is constrained by a set of physical computational and storage limitation. In all case except those which are trivial or repetitive it would be computationally infeasible for the human mind to mimic an infinitely long SCPs. For this reason, SCPs of infinite length, though possible in a mathematical formulation, tend violate the spirit of the SCP Framework which is to *accurately* model human cognition processes, not simply their conclusions.

It is worth noting that SCPs containing infinite loops can exist and can be evaluated at any time point n , they simply cannot be evaluated in their entirety.

5.4 Purpose-Selected Cognitive Operations

5.4.1 Overview

5.4.2 Initialisation

The initialisation cognitive operation (`init`) is always the first operation in an SCP. Thus, unlike every other cognitive operation, `init` has no prior cognitive operation from which to draw an epistemic state. Instead, it creates an epistemic state that reflects the initial inferential rules and variable assignments known by the agent. These rules and assignments are drawn from two distinct sources:

- **The initial interpretation of the task given:** in the case of the Suppression Task, for example, this rule corresponds to the given rules about Alice and the Library.
- **Background information:** though this knowledge base is as large as every fact and inferential rule by known the participant, no computational is lost by only showing those facts and inferential rules that directly relate to the task in question².

In many ways it could be argued that the `init` operation is simply an aggregate cognitive operation that each encodes either a variable or rule insertion into the epistemic state (@TODOfig). The one caveat to this interpretation is that there must still be some first cognitive operation which takes no input and produces an epistemic state of suitable structure as output.

Because it is actually used as a shorthand for whatever sequence of sub-operations is needed to represent the initial epistemic state `init` is the only cognitive operation which differs on a task-by-task and a reasoner-by-reasoner basis. Every other cognitive operation is a uniquely-named pipeline whose internal structure is fixed for any task in which it appears.

¹Because Ω is the set of all possible cognitive operations there must be at least one $m \in M$ such that, for any given input x , $x \mapsto m$ results in a state point of size greater than the capacity of any storage system to hold.

²The question of precisely *how* to sift through that enormous amount of information and select appropriate knowledge for the task at hand is an open problem.

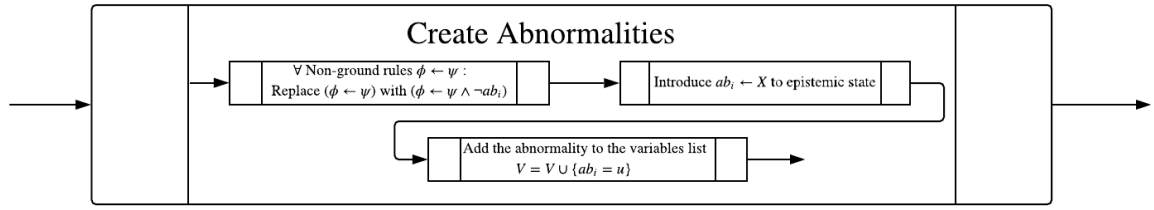


FIGURE 5.1: One possible formulation of a cognitive function to add abnormalities to variables. This formulation follows the intuition of Appendix @TODOref.

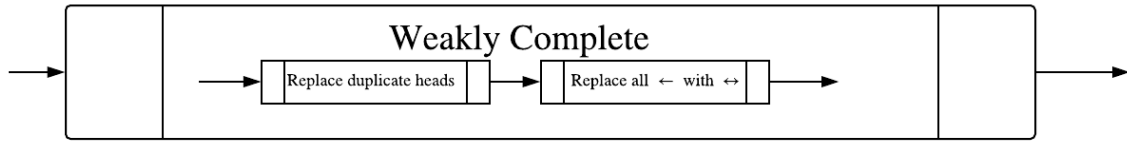


FIGURE 5.2: One possible formulation of a cognitive function to weakly complete a set of rules. This formulation follows the intuition of weak completion outlined in Section @TODOsec.

5.4.3 Variable Insertion

5.4.4 Variable Deletion

5.4.5 Variable Silencing

5.4.6 Variable Fixation

5.4.7 Adding Abnormalities

One of the most obvious candidates for a cognitive function that has evidence across a wide swathe of the cognitive literature (@TODOrefAFewPapers) is the idea of adding an abnormality variable to a problem that does not appear in the initial formulation of that problem. In Section @TODOref and Section @TODOref we discussed the idea of abnormalities as licenses for implication. The fact that we can point to several tasks in which an abnormality variable is added after formulation provides compelling evidence that creating these abnormality variables is a well-founded cognitive process that should be included in any comprehensive formulation of M . Figure 5.1 shows one possible formulation of the addAB function for adding abnormalities to a knowledge base of the form $s = (KB, \dots)$ where KB is a set of rules, and V is a set of (variable name, variable value) pairs.

5.4.8 Weakly Completing

Weak completion is an essential part of the WCS. Under the assumption that the WCS is a valid representation of human cognition in at least one scenario, any comprehensive set of cognitive operations M must be able to mimic the Weak Completion of a knowledge base. Figure 5.2 how this can be achieved for any epistemic state of the form $s = (KB, \dots)$.

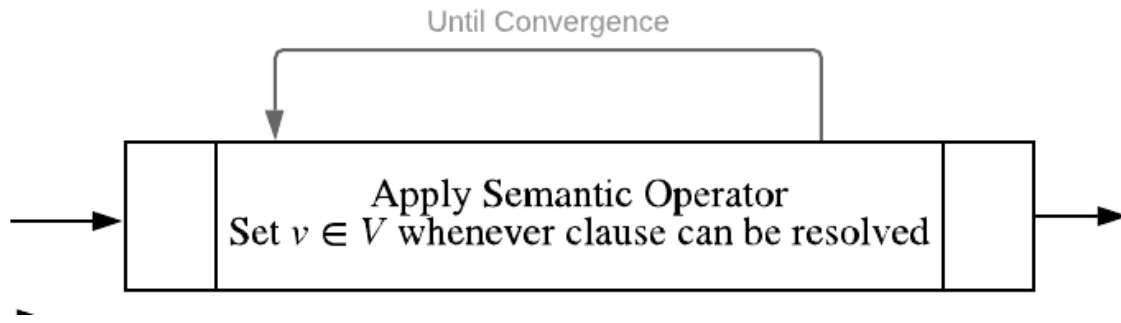


FIGURE 5.3: One possible formulation of a cognitive function to apply the semantic operator a cognitive state by temporarily setting all variables in V to any value that can be computed using the semantic rule in Equation @TODOref. This process is repeated until V is no longer updated. @TODOredoThisDiagramToShowActualProcess

5.4.9 Semantic Operator

Another essential step in the WCS, the semantic operator is used to assign all variables to either True or False (explicitly), or to Unknown (implicitly). Figure 5.3 illustrates one way in which this cognitive operation can be implemented for any epistemic state of the form $s = (KB, V, \dots)$.

5.4.10 Default Inferencing

If we assume that Reiter's default logic is a valid model of cognition for at least one task, it follows that an comprehensive formulation of M must encode a cognitive operation for drawing inferences from a set of default rules. Figure @TODOref illustrates the procedure for drawing such inferences from an epistemic state of the form $s = (KB, V, D)$ where KB is a set of inference rules, V is mapping of variable names onto variable values, and D is a set of default rules of the form $\frac{\text{condition:exception}}{\text{conclusion}}$ @TODOrewriteEquationWithStandardSymbols.

5.5 Cognitive Operations as Aggregates

An obvious and interesting idea follows from Proof 4 which is the idea of finding well-founded, uninterrupted epistemic operation sequences which are effective in modelling a variety of tasks and representing them as a single epistemic operation. We call this approach *aggregating*, and it draws inspiration from recent advances in the field of reinforcement learning @TODORef in which simple tasks previously achieved are used as allowable actions when attempting to solve complex tasks.

This approach introduces the desirable property of shortening the total length of the SCP for a given set of cognitive tasks. However, it is evident from Proof 3 that any SCP task in which an aggregated subsequence of operations replaces those individual operations in M may be less expressive than the same formulation in which the aggregated operations are still present.

Proof 3 Given a cognitive task $\Pi = (x, f(), M, \gamma)$ in which $M = \{A_0, \dots, A_n\}$ and a second cognitive planing task $\Pi' = (x, f(), M')$ in which $M' = M \setminus \{A_k, \dots, A_m\} \cup A'$, where $A' = (A_k \mapsto \dots \mapsto A_m)$ is a cognitive sequence containing some ordering of the other operations in $(\Pi \setminus \Pi')$, Π' is as expressive or less expressive than Π .

No more expressive:

- If $\pi' = (x \mapsto A_p \mapsto \dots \mapsto A' \mapsto \dots \mapsto A_q)$ and $f(\pi)$ is a solution to Π' .
- Let $\pi'' = (x \mapsto A_p \mapsto \dots \mapsto [A'] \mapsto \dots \mapsto A_q)$.
- Then $f(\pi'')$ is a solution to Π' (Lemma 3).
- $f(\pi'')$ is a valid solution to Π because π'' uses only cognitive operations which occur in Π , x is the initial epistemic state, and $f(\pi'') \models \gamma$.

Less Expressive: Proof by counterexample

- If we assume that Π' is strictly as expressive or more expressive than Π , then there exist no cases in which Π' is less expressive.
- Let $\gamma = \emptyset$, and $f(x) = \text{True}$ for all inputs (i.e. is trivially satisfied).
- Let $x = \{\}$, $M = \{A_0, A_1\}$, $M' = \{A'\}$, $A' = \{A_0 \mapsto A_1\}$.
- Then solutions of Π are $f(x)$, $f(x \mapsto A_0)$, $f(x \mapsto A_1)$, $f(x \mapsto A_0 \mapsto A_1)$ and $f(x \mapsto A_1 \mapsto A_0)$.
- Solutions of Π' are $f(x)$, $f(x \mapsto A')$.
- Thus, Π' has fewer solutions than Π . A contradiction.

@TODOexample?

The use of aggregates then, is a trade-off between sacrificing cross-reasoner accuracy and optimising a set of desirable heuristic properties such as length minimisation.

In many ways this compromise follows the philosophy of cognitive modelling in general. A neuron-by-neuron approach to predicting human behaviour (if one were ever possible) would give us extreme accuracy in modelling any human reasoner, but is too complex to be practical, and even if it were, would provide no abstracted information with which to find common motifs and inferences among reasoners. On the other extreme, modelling reasoners as *(input, output)* pairs perfectly captures the average predictions of a population, but proves very inaccurate for unseen individuals. State-of-the-art approaches to cognitive modelling like neural networks and non-monotonic logical frameworks seek generalizations which accurately capture the responses of most reasoners across as many tasks as possible by approximating human motifs in human reasoning and applying across multiple tasks and inputs.

The question of how to generate appropriate complex operation aggregates is a complex problem and is not discussed further in this thesis. It is, however a topic in which the author has significant interest.

Proof 4 Given an SCP $\pi = f(x \mapsto A_0 \mapsto \dots \mapsto A_n)$, where $f(x)$ is an external evaluation function, x is a state point, and $A_i \in \Omega$, any subsequence $A' = (A_k \mapsto \dots \mapsto A_{k+l})$, $(k+l) < n$ of the cognitive operations in π is a valid cognitive operation.

- Given that π is a valid SCP, $x \mapsto A_0$ must, by the definitions discussed in Section @TODOref result in a valid state point.
- It follows that A' takes a valid state point as input, because A_k took a valid state point as input.
- It follows that A' produces a valid state point as output, because A_{k+l} produced a valid state point as output.

- Therefore, A' is a valid cognitive operation, by the definition in Section @TODOref.

Lemma 3 Let $\pi = (x \mapsto A_0, \dots, A', \dots, \text{longmapsto} A_n)$ be a sequence of cognitive operations drawn from some cognitive task $\Pi = (x, \gamma, f(x), M)$. And let $\pi' = (x \mapsto [A_0], \dots, A', \dots, \text{longmapsto} A_n)$ where $[A]$ is the substitution of the aggregate operation A' for $(A'[0] \mapsto \dots \mapsto A'[t])$. Then $f(\pi) = f(\pi')$.

5.6 Epistemic State Structure Changes with Cognitive Operations

The conception and mathematics of cognitive operations which produce output epistemic states points which differ in structure from the epistemic input structure are fairly straightforward.

Some cognitive operation A with input base point structure $\text{struct}(s_k)$ and output base point structure $\text{struct}(s_{k+1})$ is called a *structural transformation operation* iff $\text{struct}(s_k) \neq \text{struct}(s_{k+1})$.

The intuition behind structural state changes goes back to Albert Einstein's quote that opened this thesis "Everything should be made as simple as possible, but no simpler.". In theory there is no restriction on SCP structure that would prevent every epistemic state passing an arbitrarily large number of variables and arguments. An SCP meant to represent a cognitive task where participant responses are consistent with propositional logic, for example, could be accurately represented with an initial state $s_x = (KB)$ where KB is simply the knowledge base of facts and rules given; but a researcher might equally use an epistemic state $s_y = (KB, V)$ where a variable V is intended to store the values of each variable in KB ; or even $s_z = (KB, V, D)$ where D is a set of default rules. All three of these approaches would give an SCP the information required to evaluate the propositional task, but, if we know the task structure does not deal with uncertainty or show variation in reasoner responses, it becomes apparent that the set of default rules D is unnecessary.

All of the possibilities given hold for the propositional task because s_y and s_z are both supersets of s_x and s_x is sufficient to model the task. Thus, provided the researchers do not believe that the task will make use of cognitive operations which require more complex state structures, s_x is the simplest solution which meets the requirements of the researchers.

An obvious extension to the discussion above is the fact that a transformation from one state point structure to another when the input structure is a subset of the output structure is as easy as appending more structural variables to the input state. This transition holds in the example above in which a propositional logic compliant state point s_x could be transformed into a WCS compliant state point s_y or a default rule compliant state point s_z .

Unfortunately, structural change by appending variables are not universal obvious cases of structural transformation where they do not apply spring to mind. Such as the case where a more expressive state point must be changed to a less expressive state point structure. Imagine a case where a cognitive operation B transforms a case point of $\text{struct}(s_z)$ to one of $\text{struct}(s_y)$. An intuitive example of this might be a hypothetical mental operation which handles biases in thinking. A student might begin at an epistemic state which believes "usually, if I study for tests, I fail" and, through bias confirmation, come to believe "If I study for tests, I fail", transitioning from a default rule, to immutable rule.

In cases like this, it is up to the researcher to make several decisions: do I believe that the now empty set of variables D no longer provides meaningful information? And how do I represent the change in the variables still present in the output state? These questions are very often task or function specific and cannot be answered in a general sense.

@TODOcompletetatestransformation

5.7 Purpose-Built Cognitive Operations

5.7.1 Propositional Logic

5.7.2 Weak Completion Semantics

5.7.3 Default Logic

5.8 Two Approaches to Creating Cognitive Operations

5.9 Theoretical Approximation

It seems reasonable that, given our desire to accurately model reasoning in participants, researchers creating cognitive functions for SCPs would choose to limit the set of allowable cognitive operations to those for which there is a sound theoretical basis. The reasons for this choice were discussed in Section @TODOref.

The first approach to creating cognitive operations we will consider is the use of cognitive operations for which there is already evidence in our existing models of human cognition. This evidence can take a wide variety of forms: it may follow from our knowledge of physics (for example, the impossibility of storing a set of unrelated variables above a certain physical threshold); evolutionary biology (there cannot exist a class of organisms without some kind of reproductive drive); anatomy (there must exist cognitive processes that facilitate voluntary muscle activation); sociology (some neuronal connections seem to restrict the number of close friends a human can maintain @TODOref); or any other field of science. Using this information can allow researchers to predict what cognitive functions may play a role in observed empirical data.

For the most part, theoretical approximation is a very powerful and well-justified mechanism for determining which cognitive operation could plausibly be considered in the cognitive toolbox for a given task. Restricting the space of other cognitive operations which may play a part is also enabled by all the scientific fields mentioned above, complexity theory tells us that humans are unlikely to apply exhaustive classical logic reasoning to a task when the set of variables and rules to be considered is large because of physical limitations related to solving *NP-complete* or harder tasks efficiently.

At present, the foundation of non-monotonic reasoning in cognitive modelling is this theoretical mindset, a well-founded explanation is found by borrowing from a relevant field (often psychology) and then tested against empirical evidence and across multiple cognitive tasks to show evidence that is a reasonable approximation of a mechanism in human cognition.

5.10 Empirical Approximation

Contrasting theoretical approximation is the field of constrained empirical approximation. Techniques ranging from neural networks and Markov models to genetically inspired techniques consistently outperform humans in a huge variety of fields. Drawing on enormous databases of empirical data, modern artificial intelligence researchers have come to view machine learning as the most plausible way to create the first real thinking machine @TODOref.

@TODOfinish

Chapter 6

Mimicking Non-Monotonic Logics With SCPs

6.1 The Suppression Task

6.2 The Weak Completion Semantics

Chapter 7

Modelling Experiments with SCPs

7.1 Overview

7.2 Suppression Task

Figure 7.1 illustrates a generalised SCP to describe the Suppression Task as a series of sequential steps directly mirroring the discrete steps outlined in Section 3.1, each cognitive operation passing information to the next process¹. To model the SCP the implicit sequence of operations in the Suppression Task is systematized and refined into a set of complex operations. Further we introduce an initial epistemic state $s_i = (KB, V, R)$. One interpretation of the requirements of the suppression task $\pi = (s_i, \gamma, M)$ using SCPs and the WCS is as follows:

$$\begin{aligned}
 s_i &= \{KB_i, V_i, R_i\} \\
 KB_i &= \{e \rightarrow l, \top \rightarrow e, o \rightarrow l\} \\
 V_i &= \{e : u, l : u, o : u\} \\
 R_i &= \{\} \\
 M &= \{\text{init}, \text{addAB}, \text{WeaklyComplete}, \text{semanticOperator}\} \\
 \gamma &= (l \models \top) \text{ or } (l \models \perp)
 \end{aligned}$$

where all cognitive operations require as input and produce as output a state point p where every ground point $\bar{p} \in_s p$ is of format $\bar{p} = (KB, V, R)$; `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 ?? (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`

¹It 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.

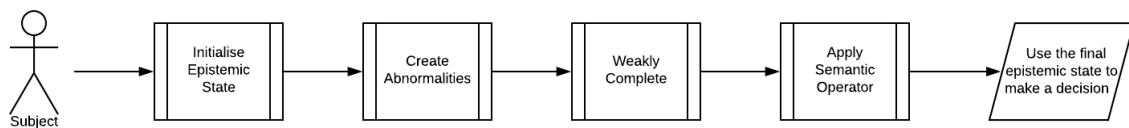


FIGURE 7.1: A generalised illustration of the WCS in an SCP.

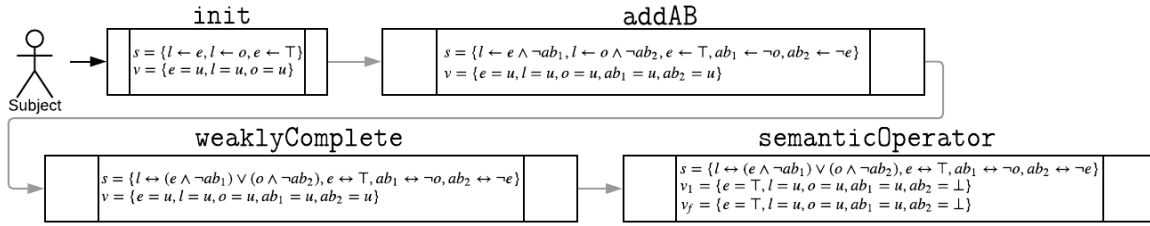


FIGURE 7.2: The standard case of the Suppression Task, demonstrating the suppression effect. Where the epistemic state in the boxes represents the output of that cognitive operation. V_f represents the assignment of V in the epistemic state in the resulting least model. $V_{i \in \mathbb{N}}$ represents the assignments in V after i iterations of the semantic operator.

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. `semanticOperator` follows the same logic seen in Section ??, but directly updating V after each iteration of the semantic operator, instead of the externalised variable set J . Thus, we ensure that the output state point is able to communicate the result of applying the semantic operator without any structural changes to the epistemic states it contains. As an additional feature of the `semanticOperator` cognitive operation, if there exists a labelled set in R called *fixed*, then the semantic operator will not set the value of any $v \in \text{fixed}$ in the variables list V . The goal γ states that l should no longer be mapped to unknown in the final epistemic state.

Treating Figure 7.1 as an SCP, we observe the sequence of output states seen in Figure 7.2. Note that in the final state l remains mapped to u , meaning that the suppression effect is demonstrated.

7.2.1 Extending the Suppression Task with SCPs

The previous example merely showed that SCPs 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. In the original Suppression Task Experiment examined in Section 3.1, a significant portion of people still believed that she would study late in the library, even though the majority suppressed the inference. 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 order to model these non-general reasoners, 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, but it is done to reinforce that we might, reasonably, expect these cognitive operations to occur in day-to-day human cognition.

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

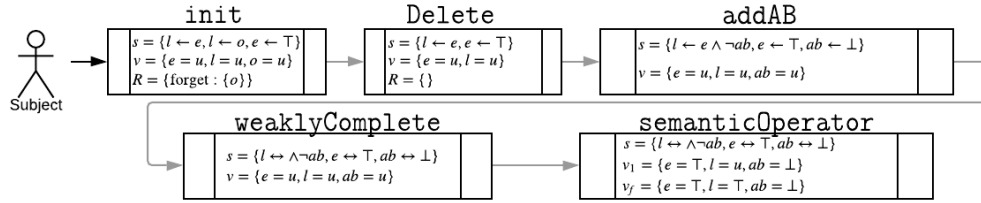


FIGURE 7.3: The Suppression Task in which the additional operation of deleting the variable o occurs.

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 should be considered. 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 are non-trivial, and there is no best practice for doing so. But in simple cases the process can be intuitive. Let `delete` be a complex operation. `delete` takes as input any state point and is applicable for any ground point \bar{p} with variable list $V \in \bar{p}$ and categorization variable R with $(delete : V_{del}) \in R$, where V_{del} is the set of variable names to delete. For every $v \in V_{del}$ remove all rules from KB that have v as body or head of the clause, and remove v from V . Then remove `delete` from R .

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 7.3 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.

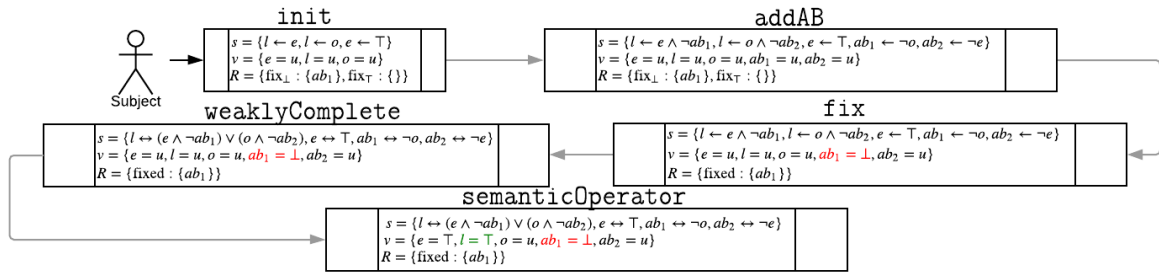


FIGURE 7.4: The Suppression Task in which the additional operation of fixing the variable ab_1 to false occurs.

In this example Karen shows a very powerful type of cognitive bias, the unwillingness to change her opinions, despite powerful evidence to the contrary. This phenomenon has been observed across a great many fields of study, from medical psychology (Brown et al., 2010) (Wroe et al., 2005) to political sciences (Tappin, Leer, and McKay, 2017). In the context of cognitive modelling with logics, it indicates that some 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 effects, as non-monotonic logics are built on the basis of a preference operation.

As one possible implementation of this idea, let us introduce a cognitive operation called *fix*. *Fix* takes as input any state point and is applicable for any ground point \bar{p} with variable list $V \in \bar{p}$ and categorization variable R with $(fix_{\top} \vee fix_{\perp}) \in R$. For all variables v such that $(v \in V) \cap (v \in fix_{\top})$ set the value of v to \top , For all variables v ($v \in V$) $\cap (v \in fix_{\perp})$ set the value of v to \perp . Then remove fix_{\top} and fix_{\perp} from R . Append $fixed : V_{fix}$ to R , where V_{fix} is the set of all variables fixed in this way.

Now, because *semanticOperator* does not change the values of variables mentioned in $V_{fix} \in fixed : V_{fix}$, $fixed \in R$, Figure 7.4 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 in KB when the semantic operator is applied, ab_1 will remain false.

7.3 Wason Selection Task

Chapter 8

Comparing SCPs

8.1 Why do we need to compare SCPs?

The ability to compare the feasibility of different solutions is an essential step in any computational process in which multiple solutions produce the desired output. Consider a toy example of an SCP task which describes the mental process needed to bake a cake:

$$\Pi = (s_0, M, \gamma, f())$$

$$s_0 = (V = (\text{cakeBaked} : \perp))$$

$$M = \{\text{mixIngredients}, \text{bakeIngredients}, \text{doTheLaundry}\}$$

$$f(x) = \begin{cases} \text{cakeBaked} \models \top & \text{True} \\ \text{cakeBaked} \models \perp & \text{False} \end{cases}$$

$$\gamma = (f(\pi) = \text{True})$$

Without specifying the precise details of the complex operations in M , and simply using our intuition of the effects of these actions, we can draw some candidate SCPs. Candidate SCPs such as $f(s_0 \mapsto \text{mixIngredients})$ do not result in the cake being baked and so are discounted immediately. However, consider a case where the modelling algorithm being used has come up with two possible SCPs to explain the processes chosen by the participant:

$$SCP_1 = f(s_0 \mapsto \text{mixIngredients} \mapsto \text{bakeIngredients}) \quad (8.1)$$

$$SCP_2 = f(s_0 \mapsto \text{mixIngredients} \mapsto \text{doTheLaundry} \mapsto \text{bakeIngredients}) \quad (8.2)$$

Intuitively, both of these operational sequences would result in a cake being baked and $f(X)$ returning *true*, and thus, both are candidate solutions to the SCP task given. However, both of these solutions may not be equally *plausible*. Why would someone need to do their laundry to make a cake? We can concoct wild scenarios in which the participant's house is so full of dirty laundry that access to the oven is restricted, but this seems implausible. Most readers would agree that Equation 8.1 is more plausible than Equation 8.2.

This toy example is evidence that, in at least some cases, one can confidently prefer one SCP to another. The question now arises: how do we precisely, and consistently prefer one SCP over another? This question is not easy to answer, and this chapter is devoted to proposing candidate solutions which may be able to quantitatively score and select preferable SCPs.

Section 8.2 discusses the question of how to compare different SCPs found using search for a single task. Section 8.3 discusses the more general problem of comparing SCPs even when the underlying SCPs tasks differ. Section 8.3.1 introduces the Needleman-Wunsch Algorithm for string matching whose underlying principles have allowed us to quantify questions of homology and evolutionary relationships in biology, and Section 8.3.2 discusses and justifies an extension to this algorithm for use in SCP comparisons.

8.2 Comparing Generated SCPs

8.2.1 Scoring

Cognitive modelling as a science that exists, in part, to replicate the empirical results of human reasoning suffers from a painful truth: just because a solution is simple, elegant and seemingly well-justified, it does not follow that that solution is correct. Indeed, that solution might completely fail to explain experimental data from a cognitive task, and must then be discounted. However in the fields of string-matching, etymology, and homological evolution @TODOref3times, mathematically consistent approaches to scoring are still generally a good starting point. And so we carry that assumption into the field of cognitive modellings and assume that, in the absence of directly contradictory empirical data, certain properties related to finding the optimal sequence of cognitive operations are desirable, whilst others are not.

In general the easiest way to compare two distinct objects is to quantify some subset of their properties and use these properties to rank the objects. Continuing with the toy example in Section 8.1 we will attempt to create a commonsense scoring mechanism to determine whether SCP_1 or SCP_2 is a more cognitively plausible solution to baking a cake. A great many possible criteria exist for scoring these two SCPs, but we will focus on just two of them: the length of the SCP, and the plausibility of each cognitive operation that occurs in either SCP.

SCP Length

Perhaps the simplest and most intuitive way to decide which of two SCPs is best suited to solving a specific problem is to prefer the shortest one. In the field of Bioinformatics, one of the earliest approaches to determine which two organisms from a set were more closely related was to directly estimate how many genetic mutations (insertions, deletions, value changes) would be necessary to turn each of these genetic sequences into each other sequence @TODOref. The same logic can be applied to SCPs for those SCPs generated using either *De Novo* or *Insertion Search* (Section @TODOref).

In the case of *De Novo* search, we assume that the optimal length of a solution to Π is an SCP length $l = |X| = 0$. This optimal solution obviously does not exist in this case $f(s_0) \not\models (cakeBaked = \top)$, but it serves as a way of implicitly preferring shorter SCPs, as those will require fewer insertion operations to satisfy $f(x)$.

Using this simple test criteria: $|SCP_1| < |SCP_2|$, therefore SCP_1 is preferred because it requires only 2 operations to transform the ideal SCP $f(s_0)$ into $f(s_0 \mapsto mixIngredients \mapsto bakeIngredients)$, rather than the 3 required for SCP_2 .

The case for *Insertion search* follows identical logic in order to model deviations from the general reasoner, but uses an ideal SCP which is the known SCP for the general case.

Though simple, this scoring procedure provides the foundations upon which more complex scoring algorithms will be built for the remainder of this section.

8.3 Comparing External SCPs

8.3.1 The Needleman-Wunsch Algorithm

In the field of Bioinformatics, one of the earliest approaches to determine which two organisms from a set were more closely related was to directly estimate how many genetic mutations (insertions, deletions) would be necessary to turn each of these sequences into each other sequence.

8.3.2 A Modified Scoring Algorithm

Chapter 9

Program Design

9.1 Implementing the SCP Framework

The SCP framework has been implemented in Python 3 as series of modules which follow strong object-oriented programming principles which support simple extensions and revisions. This implementation is used to demonstrate the practical approaches to generating and evaluating SCPs; and to model a series of cognitive tasks for which empirical results already exist. The library containing this implementation and its associated documentation is available at @TODOref.

At present, concrete implementations for epistemic state structures and cognitive operation formulations related to propositional logic, the WCS, and Reiter's default logic are provided.

9.2 A Modular Programming Approach

The principle of modular programming is followed in this implementation to the greatest extent possible, thus maximising the ease of modification and upkeep for the program. Figure @TODOref provides a simplified class diagram to explain how the major components of the system relate to one another.

9.2.1 Most Important Modules

9.2.2 Program Flow

Diagram @TODOref demonstrates the basic programatic flow used to model the Suppression Task using a WCS SCP. Diagram @TODOref demonstrates the programmatic flow used by the De Novo search in order to generate an SCP to map to known data. Finally, Diagram @TODOref shows the process by which information stored in a .csv file can be read into CCOBRA before being interpreted by the SCP framework and used for modelling.

9.3 External Framework Integration

Chapter 10

Non-Sequential Cognitive Processes

Chapter 11

Conclusions and Future Work

11.1 Conclusion

SCPs represent a novel and powerful framework for modelling non-monotonic logics. They have been shown capable of modelling the Suppression Task under the WCS for both general and individual cases. SCPs provide a dynamic framework incorporating cognitive operations which are applicable across different logics, provided that those logics share structural features in the initial epistemic state that are compatible with some subset of the known set of cognitive operations. Because the input and output epistemic states need not be structurally similar, SCPs may represent the first approach to modelling human cognition that is able to integrate multiple non-monotonic logic frameworks at run-time and at search-time.

SCPs represent a new frontier for Cognitive Modelling in non-monotonic logics and research into their capabilities and limitations may help create more robust and mathematically consistent explanations and predictions for human behaviour across an extensive array of cognitive tasks.

11.2 Future Work

The addition of two distinct search paradigms to SCPs may one day enable researchers to generate or test hundreds of thousands of models where only the initial epistemic state is known. The first search type, *De Novo Search*, involves finding an SCP that satisfies a goal condition from scratch, iterating over the set of cognitive operations in M . This search is important for producing a computationally valid explanation for the general (most common) results of a cognitive task. *Insertion Search*, by contrast, will require 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 with respect to the original SCP. Finally, some form of scoring criteria is needed to compare SCPs to one another and evaluate their relative plausibility with respect to the known complexity of the cognitive operations involved. A modified version of Needleman-Wunsch algorithm for string matching has already been theoretically described and must now be implemented in order to determine how similar different models are to each other.

Appendix

@TODO talk about adding abnormalities

Appendix A

Frequently Asked Questions

A.1 How do I change the colors of links?

The color of links can be changed to your liking using:

```
\hypersetup{urlcolor=red}, or  
\hypersetup{citecolor=green}, or  
\hypersetup{allcolor=blue}.
```

If you want to completely hide the links, you can use:

```
\hypersetup{allcolors=.}, or even better:  
\hypersetup{hidelinks}.
```

If you want to have obvious links in the PDF but not the printed text, use:

```
\hypersetup{colorlinks=false}.
```


Bibliography

- Breu, Christian et al. (2019). "The Weak Completion Semantics Can Model Inferences of Individual Human Reasoners". In: *European Conference on Logics in Artificial Intelligence*. Springer, pp. 498–508.
- Brown, Katrina F et al. (2010). "Omission bias and vaccine rejection by parents of healthy children: implications for the influenza A/H1N1 vaccination programme". In: *Vaccine* 28.25, pp. 4181–4185.
- Byrne, Ruth MJ (1989). "Suppressing valid inferences with conditionals". In: *Cognition* 31.1, pp. 61–83.
- Dietz, Emmanuelle-Anna, Steffen Hölldobler, and Marco Ragni (2012). "A computational logic approach to the suppression task". In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. Vol. 34. 34.
- Dietz, Emmanuelle-Anna, Steffen Hölldobler, and Christoph Wernhard (2014). "Modeling the suppression task under weak completion and well-founded semantics". In: *Journal of Applied Non-Classical Logics* 24.1-2, pp. 61–85.
- Hölldobler, Steffen (2015). "Weak Completion Semantics and its Applications in Human Reasoning." In: *Bridging@ CADE*, pp. 2–16.
- McDermott, Drew and Jon Doyle (1980). "Non-monotonic logic I". In: *Artificial intelligence* 13.1-2, pp. 41–72.
- Mocanu, Decebal Constantin et al. (2018). "Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science". In: *Nature communications* 9.1, pp. 1–12.
- Reiter, Raymond (1980). "A logic for default reasoning". In: *Artificial intelligence* 13.1-2, pp. 81–132.
- Saldanha, Emmanuelle-Anna Dietz, Steffen Hölldobler, and Isabelly Lourêdo Rocha (2017). "The Weak Completion Semantics." In: *Bridging@ CogSci*, pp. 18–30.
- Sigman, Mariano and Stanislas Dehaene (2008). "Brain mechanisms of serial and parallel processing during dual-task performance". In: *Journal of Neuroscience* 28.30, pp. 7585–7598.
- Smirnova, Anna et al. (2015). "Crows spontaneously exhibit analogical reasoning". In: *Current Biology* 25.2, pp. 256–260.
- Stenning, Keith, Michiel Van Lambalgen, et al. (2008). "Interpretation, representation, and deductive reasoning". In: *Reasoning: Studies of human inference and its foundations*, pp. 223–248.
- Tappin, Ben M, Leslie van der Leer, and Ryan T McKay (2017). "The heart trumps the head: Desirability bias in political belief revision." In: *Journal of Experimental Psychology: General* 146.8, p. 1143.
- Wason, Peter C (1968). "Reasoning about a rule". In: *Quarterly journal of experimental psychology* 20.3, pp. 273–281.
- Wroe, Abigail L et al. (2005). "Feeling bad about immunising our children". In: *Vaccine* 23.12, pp. 1428–1433.