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

June 22, 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) and Reiter's Default Logic, 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 the SCP framework through examination of the Wason Selection Task, Suppression Task, @TODOexperiments.

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

List of Figures

List of Tables

List of Abbreviations

SCP	**S**equential **C**ognitive **P**rocesses
nSCP	**n**on-**S**equential **C**ognitive **P**rocesses
WCS	**W**eak **C**ompletion **S**emantics
WST	**W**ason **S**election **T**ask
CTM	**C**ognitive **T**ransition **M**odel
pCTM	**p**artial **C**ognitive **T**ransition **M**odel

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 inferences should be interpreted 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 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 thesis 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.

1.2 A Review of Terminology

1.3 Current State of the Art

The current state of the art in cognitive modelling using non-monotonic logics has begun to show a shift towards a computational approach to problem solving (Dietz, Holldobler, and Ragni, 2012). Stenning and Van Lambalgen, 2012 argued, in a general sense, that modelling human reasoning should be done primarily towards an appropriate representation of the underlying cognitive processes, and secondarily, it should be evaluated empirically with respect to this representation. This philosophy is in stark contrast to that of many other areas of artificial intelligence research, in which the underlying structure of the agent does not need to mimic existing cognitive processes, and instead the primary evaluation criteria is generally the effectiveness of the agent at making predictions on unseen data sets.

This thesis makes extensive reference to the work of Dietz, Holldobler, and Ragni, 2012, Dietz, Hölldobler, and Wernhard, 2014, Ragni, Kola, and Johnson-Laird, 2017 and their contributions to the emerging field of cognitive modelling.

The author of this thesis has previously shown the suitability of the Weak Completion Semantics for modelling individual reasoners in the Wason Selection Task (Breu et al., 2019) and the SCP framework that this thesis introduces is an extension of the intuition of discrete, sequential processes underlying cognition. The question of how to model individual reasoners is a pernicious problem in cognitive modelling and work by Breu et al., 2019, as well modelling tools like `CCOBRA` (Nicolas, 2018) have been able to use traditional modelling techniques, with minor variations, to replicate more than simply the most common reasoner responses.

1.4 Contributions of the SCP Framework

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. 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.

Logics which are able to interpret conditionals of the form "If ϕ, then ψ" are varied in both their implementation and underlying principles. Work by Heit, Hahn, and Feeney, 2005 has shown evidence that not all information is derived, stored, or retrieved symmetrically and thus, it is unlikely that a single logical system will be able fully model tasks which utilise information whose interpretation is dependant on multiple structurally and interpretationally diverse sources of information. The SCP framework enables researchers to use robust search techniques which make no assumptions about conditional interpretation at formulation time, and rather, branches to incorporate multiple possible epistemic knowledge structures at run-time. This technique, enables multiple non-monotonic logics to be used across a single cognition task.

This thesis also introduces a new method for modelling reasoners in the WST which makes use of the interpretation of a conditional and its contraposition to provide a more intuitive method of applying the WCS to the task. Section 6.2 shows that Dietz, Hölldobler, and Wernhard, 2014's method of adding a subset of allowable abducables to a knowledge base to model the general case of the WST can also be used to model an individual reasoner case of the Suppresion Task in which suppression is not observed.

1.5 Thesis Layout

This thesis begins with a discussion of the mathematical preliminaries related to logic programming and non-monotonic logics - particularly the WCS and Reiter's default logic (Chapter 2). Chapter 3 introduces several classical experiments in the field of cognitive modelling and discusses an existing non-monotonic interpretation of each. Chapter 4 introduces the SCP framework and the concept of SCP Tasks, SCPs, and Realised SCPs. Chapter 5 introduced the concept of the cognitive toolbox which contains well-founded tools to allow the SCP framework to model a wide variety of cognitive tasks. Chapter 6 shows how SCPs can be used to model the classical experiments introduced in the Chapter 3 in terms of both individual and generalised reasoners. Chapter 7 discusses how SCPs might be compared to one another to determine which is most plausible or to identify common motifs that are present across experiments. Chapter 8 briefly discusses the implementation of the SCP framework in Python 3 and the core philosophies of its design.solutionspace Finally, Chapter 9 summarises the contributions of this thesis and discusses potential extensions to the SCP framework.

Chapter 2

Mathematical Preliminaries

2.1 Propositional Logic

Propositional logic is one of the simplest classical logics and concerns itself with binary truth values. $\top$ denotes a tautology, and $\bot$ denotes a contradiction. An alphabet $\Sigma = \{v_1, ..., v_n\}$ defines the set of atoms available to the language. Every atom $v_i \in \Sigma$ has the domain $\{v_i, \bar{v}_i\}$. An (atom, domain) pair which associates an atom with a value in its domain is called a literal.

A formula $\theta \in \zeta$ from set of all formuli ζ is defined recursively over all conjunction $\wedge$, disjunction $\vee$, and negation $\neg$ operations as follows:

$$\theta = \{v_i \in \Sigma | \neg\theta | \theta \vee \theta | \theta \wedge \theta\}$$

The material implication $(\psi \leftarrow \phi)$ precisely substitutes for the formula $(\neg\psi \vee \phi)$ and is often used, classically to denote an "if ϕ then ψ" relationship between variables. However, in Section 2.3 other interpretations of the "if ϕ then ψ" relationship are examined in a non-monotonic setting.

An interpretation of Σ, $I_w(\Sigma)$ assigns each $v_i \in \Sigma$ to one value in its domain as defined by possible world w. The interpretation of a formula θ, $I_w(\theta)$ is recursively calculated using the associated truth tables of the operators involved, as defined in Table 2.1.

2.2 Possible Worlds

A possible world $w \in W_\sigma$ from the set of all possible worlds W over a language Σ is $(v_0, w_0) \wedge ... \wedge I(v_n, w_n)$ and is the conjunction of each atom and an associated value from its domain. Possible worlds represent one setting of the variables in the alphabet that could hypothetically hold. A model of a propositional formula $\theta \in \zeta$ is a possible world w in which θ is evaluated to $\top$ (also called satisfying) and is written $w \models \theta$.

$Mod(\theta)$ denotes the set of possible worlds in which θ holds – that is, $Mod(\theta) = \{w | w \models \theta\}$. The operator $\models$ is overloaded so that $\phi \models \psi$ if and only if $Mod(\phi) \subseteq Mod(\psi)$ for $\phi, \psi \in \zeta$.

$\wedge$	$\top$	$\bot$
$\top$	$\top$	$\bot$
$\bot$	$\bot$	$\bot$

$\vee$	$\top$	$\bot$
$\top$	$\top$	$\top$
$\bot$	$\top$	$\bot$

$\neg$	
$\top$	$\bot$
$\bot$	$\top$

$\rightarrow$	$\top$	$\bot$
$\top$	$\top$	$\bot$
$\bot$	$\top$	$\top$

$\leftarrow$	$\top$	$\bot$
$\top$	$\top$	$\bot$
$\bot$	$\top$	$\top$

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

$(\psi\|\phi)$	$\phi \models \top$	$\phi \models u$	$\phi \models \bot$
$\psi \models \top$	verification	non-applicability	non-applicability
$\psi \models u$	verification	non-applicability	non-applicability
$\psi \models \bot$	falsification	non-applicability	non-applicability

TABLE 2.2: Evaluation of conditional rules.

A set of propositional clauses W is called *deductively closed*, written $\text{th}(W)$ if it contains every formula ϕ that can be logically deduced from W.

2.3 Conditionals

Defeasible knowledge is information which is assumed, but not guaranteed, to hold. Many non-monotonic logics treat this notion as a form of typicality. A condition $(\psi|\phi)$ is used to represent the defeasible rule "if ϕ then ψ". Following Wason, 1968 we do not use the implication $\leftarrow$ to represent a conditional, and instead opt for a three-valued interpretation of what Baratgin, Over, and Politzer, 2014 called a *deFinetti truth table* (Table 2.2). This truth table for conditionals differs from the standard implication in that it treats rules of the form $(\psi|\phi)$ as inapplicable when $\phi \models \bot$ or $\phi \models u$ and more closely follows human intuition about how rules are evidenced in the absence of their antecedent.

2.3.1 Interpretation of conditionals

The precise interpretation of conditionals is a topic for philosophical and logical debate. Approaches both probabilistic and logical can be used to define precisely what a conditional means in the context of a logical system. In this thesis we adopt adopt Stenning, Van Lambalgen, et al., 2008's interpretation of conditionals as licenses for implication.

Thus, the conditional $(\psi|\phi)$ precisely means $\psi \leftarrow \phi \wedge \neg\text{ab}$ where ab is called an abnormality predicate. The abnormality predicate captures an element of uncertainty in the interpretation of the conditional and means that the conditional now reflects the statement "If ϕ and nothing abnormal is known, then ψ".

The choice of appropriate value assignments to the abnormality predicate is non-trivial and often requires expert knowledge to design. However, over the course of this thesis we will adopt an algorithmic approach to abnormality creation (Algorithm 1) that will serve to mimic the hand-designed abnormalities used by Dietz, Holldobler, and Ragni, 2012, and

Breu et al., 2019.

Algorithm 1: `Conditional to License for Implication`

Function *Create Abnormalities(KB)* **is**
  for $(\psi|\phi) \in KB[\Delta]$ **do**
    $k :=$ the lowest natural number for which $\text{ab}_k \notin$ KB;
    all dependencies $:= [A|(\psi|A) \in \text{KB}'[\Delta]]$;
    for $A \in$ *all dependencies* **do**
      head $:= \psi$;
      current dependencies := all dependencies − A;
      if *current dependencies* $= \{\}$ **then**
        current dependencies $:= \bot$;
      end
      body := (current dependencies$_1 \vee ... \vee$ current dependencies$_n$);
      `/* Add the conditional as a license for implication to the set of rules. */`
      $\text{KB}'[S] := \text{KB} \cup (\text{head} \leftarrow A \wedge \neg \text{ab}_k)$;
      $\text{KB}'[S] := \text{KB}'[S] \cup \text{ab}_k \leftarrow \neg body$;
    end
  end
  `/* Remove all conditionals now that they have been interpreted as licences for implication. */`
  $\Delta := \{\}$;
  return KB′
end

2.4 Logic Programming

2.4.1 Knowledge Base

In propositional logic, a knowledge base $S = \{\phi_1, ..., \phi_n\}$ for the language Σ is a set of propositional rules from the set of formula ζ_Σ. A propositional knowledge base is used to encoded certain knowledge and relationships about the world. S is called *consistent* only if there exists some possible world $w \in W$ such that every rule in $I_w(KB_{prop})$ evaluates to $\top$.

More generally, a non-monotonic knowledge base $KB = (S, \Delta)$ contains both a set of propositional rules (S) and a set of defeasible/conditional rules ($\Delta = \{\Delta_1, ..., \Delta_m\}$). A non-monotonic knowledge base tolerates a conditional $(\psi|\phi) \in \Delta$ if and only if there exists some possible world $w \in W$ that verifies $(\psi|\phi)$ and does not falsify any conditional in Δ. KB is said to be consistent if and only if there exists of partition of Δ, $(\Delta_0, ..., \Delta_n)$ such that Δ_i is tolerated by Δ_{i+k}, where k is any natural number not greater than m.

When $KB = (S = \{\text{rule}_1, ..., \text{rule}_n\}, \Delta = \{\})$, we will sometimes ignore Δ and refer only to the value of S whilst still using the term 'KB'.

2.5 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. Ragni et al., 2016 showed that two-valued logics were not sufficient to model human cognition.

$\rightarrow$	$\top$	u	$\bot$
$\top$	$\top$	u	$\bot$
u	$\top$	$\top$	u
$\bot$	$\top$	$\top$	$\top$

TABLE 2.3: A table showing the implication operator in 3-valued Łukasiewicz logic.

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 an extension to a classical logic which introduces a preference relation $\leftarrow_p$. 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.

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

2.6 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 2.3), rather than Kripke-Kleene logic which has been shown inadequate to model several aspects of cognition (including suppression (Dietz, Holldobler, and Ragni, 2012)). 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 $A \leftarrow body_1, ..., A \leftarrow body_n$ with $A \leftarrow body_1 \vee ... \vee body_n$.
2. Replace all occurrences of $\leftarrow$ with $\leftrightarrow$.

Applying this procedure to P results in wc P 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:

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

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

Using $I =< \emptyset, \emptyset >$, the least model of P ($\text{lm}_Ł \text{wc}P$) can be calculated by iterating $\phi_{SvL,P}$.

2.7 Reiter's Default Logic

Reteir'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 are usually true. The second type of inference is the eponymous default rule.

A Reiter Default Theory (W, D) consists of a background theory W, and a set of default rules D. For our purposes we will restrict both to a propositional, rather than a first-order logic domain.

The set of default rules D consists of a set of tritonic clauses of the form $\delta = \frac{\text{pre}(\delta):\text{just}(\delta)}{\text{cons}(\delta)}$. Where $\text{pre}(\delta)$ is a propositional clause called the precondition, $\text{just}(\delta)$ is a set of propositional clauses and is called the justification, and $\text{cons}(\delta)$ is a propositional clause called the consequence. To maintain consistency, we restrict our background theory to propositional clauses of the form $(\text{head} \leftarrow \text{body})$ and restrict D so that, for any $\delta \in D$, $\text{cons}(\delta)$ is of the form $(\text{head} \leftarrow \text{body})$.

A default δ is *applicable* to a deductively closed set $\text{th}(W)$ if and only if $\text{pre}(\delta) \in \text{th}(W)$ and there exists no $\text{just}_i(\delta) \in \text{just}(\delta)$ such that $\neg\text{just}(\delta) \in th(W)$.

A default process $\Pi = (\delta_1, ..., \delta_n)$ consists of a sequence of default rules, $\delta_i \in D\}$. Following Ragni et al., 2017, we define the sets $\text{In}(\Pi) = \{\text{th}(W \cup \text{cons}(\delta)|\delta \in \Pi)$ and $\text{Out}(\Pi) = \{\neg A|A \in \text{just}(\delta), \delta \in \Pi\}$. A default process is called *successful* if and only if $\text{In}(\Pi) \cap \text{Out}(\Pi) = \emptyset$. A default process if called *closed* if and only if there exists no $\delta \in D$, where δ is applicable to $\text{In}(\Pi)$ and $\delta \notin \Pi$. The set E is called an extension of default process D if and only if $E = \text{In}(\Pi)$, for some successful closed process Π.

We write $W \models_{\text{D}}^{\text{Reiter}} \psi$ if and only if $\psi \in \bigcap \epsilon$, where ϵ is the set of all valid extensions of the default theory (W, D).

It should be immediately apparent form this formulation that, for many possible sets of rules in default theories, the number of extensions is non-monotonic as it is dependent on the number of orders in which default rules are applied successfully. A very significant restriction on reiter's default logic is the complexity of computing deductively closed sets.

The closed world assumption $\frac{:\neg\bot}{\bot}$ states that all information which is not known to be true is assumed to be false. The closed world assumption can be used to ensure that, for every variable $v_i \in \Sigma$, $I_w(v_i) \models \top$ or $I_w(v_i) \models \bot$.

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 S is such that $I(S) \models \phi$, then it must be the case that $I(S \cup \psi) \models \phi$. However, in the suppression task participants no longer draw classically valid inferences when new information is added. The task is often formulated as follows:

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

Given only the rule $e \leftarrow \top$ and conditional $(l|e)$, the participants consistently concluded that she would study late in the library, seemingly interpreting $(l|e)$ as $l \leftarrow e$ and drawing the classical logic inference $\frac{l \leftarrow e, e}{l}$ with *modus ponens*. But when given the additional conditional $(l|o)$, 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 Modelling the Suppression Task: WCS

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 using an epistemic knowledge base $\mathrm{KB}_{\mathrm{noSup}}$, with:

$$\mathrm{KB}_{\mathrm{noSup}} = (S = \{e \leftarrow \top\}, \Delta = \{(l|e)\})$$

Which represents the task without the suppressing conditional $(l|o)$. Interpreting the conditional as a license for implication, as in Section 2.3.1, the knowledge base can be rewritten as a propositional knowledge base:

$$\mathrm{KB}_{\mathrm{noSup}} = \{e \leftarrow \top, l \leftarrow e \wedge \neg \mathrm{ab}_1, \mathrm{ab}_1 \leftarrow \bot\})$$

Weakly completing $\mathrm{KB}_{\mathrm{noSup}}$ results in:

$$\mathrm{wc}\ \mathrm{KB}_{\mathrm{noSup}} = \{e \leftrightarrow \top, l \leftrightarrow e \wedge \neg \mathrm{ab}_1, \mathrm{ab}_1 \leftrightarrow \bot\}$$

Finally application of the semantic operator results in $\top = \{e\}, \bot = \{ab_1\}$ after the first execution and converges to the least model $\top = \{e, l\}, \bot = \{ab_1\}$.

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 KB_{noSup} 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 $(l|o)$.

Now the initial knowledge base KB_{sup} includes the conditional $(l|o)$:

$$\text{KB}_{\text{sup}} = (S = \{e \leftarrow \top\}, \Delta = \{(l|e), (l|o)\})$$

And, when rewritten as a propositional knowledge base:

$$\text{KB}_{\text{sup}} = \{e \leftarrow \top, l \leftarrow e \wedge \neg \text{ab}_1, l \leftarrow o \wedge \neg \text{ab}_2, \text{ab}_1 \leftarrow \neg o, \text{ab}_2 \leftarrow \neg e\}$$

Where the abnormalities added to the interpretation of conditionals $(\psi|\text{body})$ which share ψ as a head are considered dependent on one another. Weakly completing this knowledge base results in:

$$\text{WC KB}_{\text{sup}} = \{e \leftrightarrow \top, l \leftrightarrow (e \wedge \neg \text{ab}_1) \wedge (o \wedge \neg \text{ab}_2), \text{ab}_1 \leftrightarrow \neg o, \text{ab}_2 \leftrightarrow \neg e\}$$

For which applying the semantic operator converges to the least model $\top = \{e\}, \bot = \{ab_2\}$ after the first application.

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

3.1.3 Modelling the Suppression Task: Reiter's Default Logic

Ragni et al., 2017 showed that it was not possible to model the suppression task using Reiter's Default Logic. They created a default theory (W, D'_P) as an illustration of Default Theory to illustrate this point.

A naive Default Theory for the suppression task (W, D_{naive}) can be used to model the classical logic response to the suppression task (i.e. that she will study late in the library), by treating the conditional as an implication without an associated abnormality predicate.

$$W = \{e \leftarrow \top\}$$

$$D_{\text{naive}} = \{\delta_1 : \frac{e \leftarrow \top : l \leftarrow \bot}{l \leftarrow \top}, \}$$

$$D'_{\text{naive}} = \{\delta_1 : \frac{e \leftarrow \top : l \leftarrow \bot}{l \leftarrow \top}, \{\delta_2 : \frac{o \leftarrow \top : l \leftarrow \bot}{l \leftarrow \top}\}$$

The only closed and successful default process that can generated for this task is $\Pi_{\text{naive}} = (\delta_1)$, which results in the inset $\text{In}(\Pi_{\text{naive}}) = \{e \leftarrow \top, l \leftarrow \top\}$. When the set of default rules is extended to create D'_{naive} which describes information about if the library is open, then the inset $\text{In}(\Pi_{\text{naive}}) = \{e \leftarrow \top, l \leftarrow \top\}$, and $\{\delta_1\}$ remains the only extension. Both theories assert that she will study late in the library and suppression is not observed.

Even generalising our WCS approach to the Suppression Task and defining a set of defaults D_P which capture intuition about abnormalities, fails to demonstrate suppression.

$$D_P = \{\delta_1 : \frac{e \leftarrow \top : \text{ab}_1 \leftarrow \bot}{l \leftarrow \top}, \delta_2 : \frac{: \text{ab}_1 \leftarrow \top}{\text{ab}_1 \leftarrow \top}\}$$

$$D'_P = \{\delta_1 : \frac{e \leftarrow \top : \text{ab}_1 \leftarrow \bot}{l \leftarrow \top}, \delta_2 : \frac{o \leftarrow \top : \text{ab}_2 \leftarrow \bot}{l \leftarrow \top}, \delta_3 : \frac{o \leftarrow \bot : \text{ab}_1 \leftarrow \top}{\text{ab}_1 \leftarrow \top}, \delta_4 : \frac{e \leftarrow \bot : \text{ab}_2 \leftarrow \top}{\text{ab}_2 \leftarrow \top}\}$$

	D	$Dn3$	$Dn3n7$	$Dn7$
Abstract	36	39	5	19

TABLE 3.1: The canonical results of the Wason Selection Task for the Abstract case.

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

TABLE 3.2: The canonical results of the Wason Selection Task over the Abstract, Everyday, and Deontic Cases

For the theory (W, D_P), the only closed and successful default processes which produce extensions are $\Pi_P^1 = (\delta_1, \delta_2)$ and $\Pi_P^2 = (\delta_2, \delta_1)$, with $(l \leftarrow \top) \in \text{In}(\Pi_P^1)$ and $(l \leftarrow \top) \in \text{In}(\Pi_P^2)$. For the theory (W, D_P') the only closed and successful default process is $\Pi_P' = (\delta_1)$ and $(l \leftarrow \top) \in \text{In}(\Pi_P')$, thus neither theory demonstrates suppression.

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]). The task has many formulations (including the Abstract, Everyday, and Deontic cases), but this thesis will exam one of the most mostly widely researched cases: the Abstract case.

The Abstract case of the WST presents the subject with four cards on a table as shown in Figure 3.1. Their face-up sides read $D, K, 3$, and 7. Participants are told that each card has a number on one side and a letter on the other. They are then 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."

This task requires the participants to find possible defeaters for the conditional rule. Assuming the *de Finetti* truth table from Section 2.3 for the interpretation of conditionals of the form "If ψ then ϕ", Table 3.3 shows that the cards pairs $(D, 3)$ and $(D, 7)$ are both possible defeaters to the rule $(3 \leftarrow D)$ because the rule $(3|D)$ is only verified or falsified when the

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

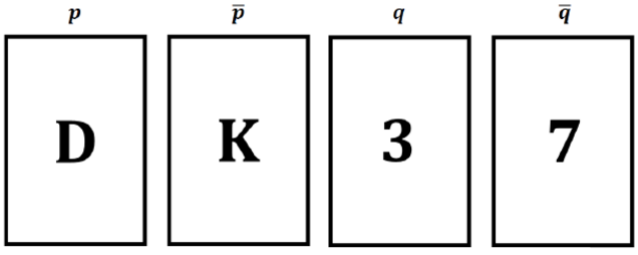


FIGURE 3.1: The Abstract case of the Wason Selection Task

D	3	$3 \leftarrow D$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\top$

TABLE 3.3: Propositional evaluation of the implication $3 \leftarrow D$ for each possible truth value of 3 and D.

D	3	$(3\|D)$
$\top$	$\top$	verification
$\top$	u	non-applicability
$\top$	$\bot$	falsification
$\bot$	$\cdot$	non-applicability
u	$\cdot$	non-applicability

TABLE 3.4: De-finetti evaluation of the conditional $(3|D)$ for each possible truth value of 3 and D.

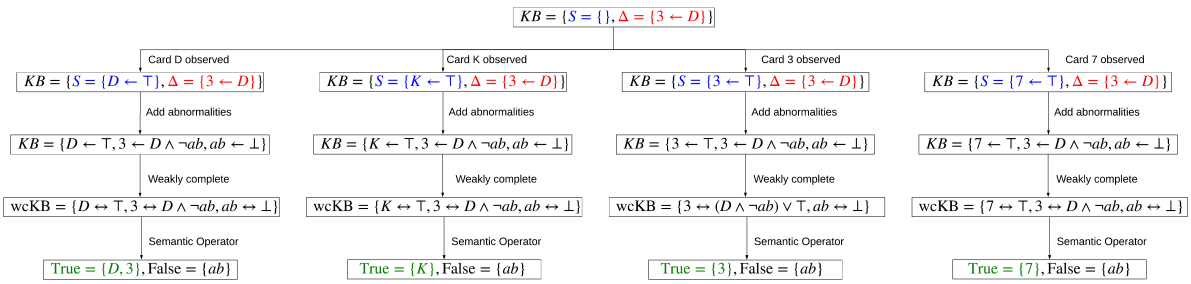


FIGURE 3.2: Application of the WCS to the abstract case of the WST for the basic weak completion approach.

Observation O	**Explanation** ϵ	**lm wc (KB** $\cup \epsilon$**)**	**Turn Function**
D	$\{D \leftarrow \top\}$	$\{D, 3\}\{\text{ab}_1\}$	turn
K	$\{K \leftarrow \top\}$	$\{K\}\{\text{ab}_1\}$	do not turn
3	$\{D \leftarrow \top\}$	$\{D, 3\}\{\text{ab}_1\}$	turn
7	$\{7 \leftarrow \top\}$	$\{7\}\{\text{ab}_1\}$	do not turn

TABLE 3.5: Dietz' computational logic approach to the abstract case of the WST.

interpretation maps $(D, \top)$ and $(3, \neg u)$. Any observed card which allows a possible world with both of those assignments is a possible defeater. If a participant were to turn the D card and find a number that was not 3, or were to a turn the card 7 and find the letter D on the other side, the classical interpretation of conditional would not hold. 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 the cards 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 (Ragni et al., 2017). This paper will focus on the WCS interpretation of the task provided by Ragni, Kola, and Johnson-Laird, 2017. This approach assumed an initial knowledge base KB containing only the conditional $(3|D)$ and an interpretation of the conditional as a license for implication.

Dietz, Holldobler, and Ragni, 2012 showed that the abstract case of the WST could be modelled using the WCS and 3-valued Łukasiewicz logic. Using the conditional as the initial logic program, we write $KB = (S = \{\}, \Delta = (3|D))$. Treating conditionals as licenses for implication KB can be rewritten as $\text{KB} = (3 \leftarrow D \wedge \neg \text{ab}_1)$ where ab_1 is an abnormality predicate. Now $\text{lm wc}(KB) = \{\}\{ab\}$, and no information is obtained that could either verify or falsify the conditional. In order to adapt the WCS to the WST they introduced the concept of observation support.

The set of abducibles A corresponds to positive and negative facts which are not assigned in KB. For the WST as formulated here $A = \{D \leftarrow \top, D \leftarrow \bot, 3 \leftarrow \top, 3 \leftarrow \bot, K \leftarrow \top, K \leftarrow \bot, 7 \leftarrow \top, 7 \leftarrow \bot\}$. An observation O is explained by a minimal explanation $\epsilon \in A$ if and only if O holds in $\text{lm wc}(KB \cup \epsilon)$. Table 3.5 shows the least models that can be drawn for the observations D, 3, K, and 7. If a least model obtained in this way contains atoms and atomic assignments that could verify or falsify the conditional (i.e. make $I_{\text{lm wc}(KB \cup \epsilon)}(3|D) \neq$ non-applicability), then the observed card should be turned. Figure 3.2 shows the derivation of $\text{lm wc}(KB \cup \epsilon)$ for $\epsilon = D, \epsilon = 3, \epsilon = K$, and $\epsilon = 7$.[2]

This simple interpretation of the WST which applies abduction to the initial knowledge base to derive observation is sufficient to model the general reasoner. 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 proportion? 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)[3] by including three simple processes: *basic weak completion*, *abduction* (as we have already discussed in this section), and *contraposition*.

[2]This approach differs in description from the author's original one, in which a card was turned if the least model assigned non-u values to every variable in the conditional (i.e. 3 and D), but is identical in practice.

[3]My contribution to this paper was the initial proposal of the extensions, their integration, the silencing mechanism used for modelling individual reasoners, and the broad idea of the stochastic evaluation technique itself.

Iteration	$\top$	$\bot$
0		
1	7	ab_1, ab_2
2	7	3, ab_1, ab_2
3	7, D'	3, ab_1, ab_2
4	7, D'	3, ab_1, ab_2, D

TABLE 3.6: Applying the Weak Completion Semantics and Contraposition to the 7 card of the WST.

By including these three 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 frequency results can also be drawn for all three cases of the WST (Table 3.2).

Basic Weak Completion

Basic weak completion ignores the existence of other least models which may validate or invalidate the inference $(3|D)$ and, instead handles only those cases where $\epsilon \leftrightarrow (O \leftarrow \top)$. That is, only information about the currently observed card is considered in the abductive framework. Figure 3.2 illustrates this process for each of the four observations: D, 3, K, and 7. Now only wc $\text{lm}(KB \cup (D \leftarrow \top))$ provides assignments which for which the de Finetti interpretation of $(3|D) \neq$ non-applicability. Observing the card 3 no longer leads to the decision to turn the card because only the case $\epsilon = (3 \leftarrow \top)$ is considered and the case $\epsilon = (D \leftarrow \top)$ is not considered.

Contraposition

Contraposition implicitly makes use of *modus tolens*, usually assumed to be silenced in human cognition, to derive certain canonical cases in the WST. Using contraposition it becomes possible to model individual participants who choose to turn the cards D and 7 (the classical response).

Contraposition is applied as follows for the initial knowledge base KB:

1. For each conditional $(\psi|\phi) \in KB'$:
   - Add the conditional $(\phi'|X)$ to Δ, where X is the other possible value for that card face[4].
   - Add the fact $\phi \leftarrow \neg\phi'$ to S.
2. Interpret conditionals as licenses for implication.
3. Apply the Abductive case of the WCS as normal.
4. Turn the card if and only if some or $I_{\text{lm WC (KB}\cup\epsilon)}(\psi|\phi) \neq$ non-applicability or some $I_{\text{lm WC (KB}\cup\epsilon)}(\phi'|X) \neq$ non-applicability, and $textlmWC(KB \cup \epsilon)$ explains O.

Using this technique, and applying basic weak completion (i.e. only considering the case $\epsilon \leftrightarrow (O \leftarrow \top)$ it is possible to model the classical case of the WST. The resolution for the 7 card is as follows:

[4] $(D, K), (3, 7)$

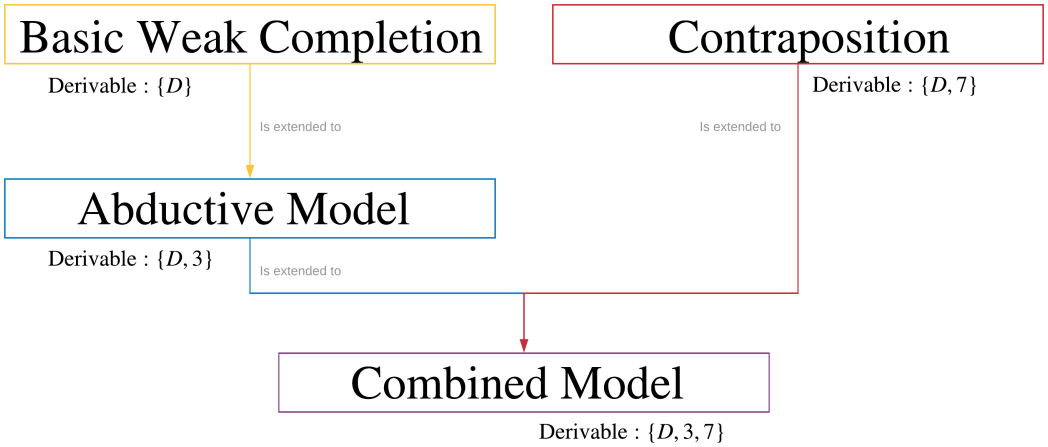


FIGURE 3.3: Using the abduction and contraposition extensions to derive all four canonical cases. Arrows indicate that the origin model produces a subset of the turn instructions of the endpoint.

$$KB = (S = \{\}, \Delta = \{(3|D)\})$$
$$KB' = (S = \{D \leftarrow \neg D'\}, \Delta\{= (3|D), (D'|7)\})$$
$$KB' = \{3 \leftarrow D \wedge \neg \text{ab}_1, D' \leftarrow 7 \wedge \neg \text{ab}_2, \text{ab}_1 \leftarrow \bot, \text{ab}_2 \leftarrow \bot\}\}$$
$$KB' \cup (7 \leftarrow \top) = \{3 \leftarrow D \wedge \neg \text{ab}_1, D' \leftarrow 7 \wedge \neg \text{ab}_2, \text{ab}_1 \leftarrow \bot, \text{ab}_2 \leftarrow \bot, 7 \leftarrow \top\}\}$$
$$\text{wc}KB' \cup (7 \leftarrow \top) = \{3 \leftrightarrow D \wedge \neg \text{ab}_1, D' \leftrightarrow 7 \wedge \neg \text{ab}_2, \text{ab}_1 \leftrightarrow \bot, \text{ab}_2 \leftrightarrow \bot, 7 \leftrightarrow \top\}\}$$
$$\text{wc}(KB' \cup (7 \leftarrow \top)) = \{3 \leftrightarrow D \wedge \neg ab_1, D' \leftrightarrow (\neg 3 \wedge ab_2) \vee (\neg D), ab_2 \leftrightarrow \bot, ab_1 \leftrightarrow \bot, 7 \leftrightarrow \top\} \tag{3.1}$$

Table 3.6 shows the computation of lm wc$(KB' \cup (7 \leftarrow \top))$. From this, it is possible to deduce that the card 7 must be turned. Figure 3.3 illustrates the use of these extensions to derive all four of the canonical cases of the WST.

Combined Model

By combining our contraposition and abductive models for inferences, it is possible to model the case where the cards D, 3, and 7 are turned. Combining the models $(A_i, ..., A_n)$ is done by turning a card c if and only if c would be turned in some model A_i.

This work has shown that all canonical cases of the weak completion semantics can be modelled by activating or silencing these three processes on the initial knowledge base. The intuition that a discrete and systematic succession of well-founded processes could be used to model multiple different results for the same experiment was the inspiration for the SCP framework which this thesis introduces.

Chapter 4

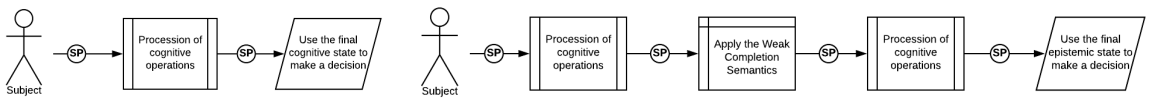
Sequential Cognition Processes

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 outlines an SCP-like structure 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 point 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.



(A) Unrestricted SCP. (B) WCS guaranteed to occur.

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.

An *epistemic state*, also called a *base point*, describes all information which the agent *believes* to be true at a given moment in its cognition. An epistemic state is an n-tuple of relevent structural variables, each of which describes some aspect of the belief of the agent at that moment (for example, a non-monotic KB as described in Section 2.4.1, is an epistemic state $s_i = (S, \Delta)$ containing structural variables S for a set of propositional rules, and Δ for a set of conditionals beliefs.)

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 operation[1]. 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 $\bar{p}$).

$J[\bar{p}, m] = p'$ describes the *application* of a base point $\bar{p}$ to a cognitive operation $m \in M$ to produce an output state point p', where p' is vector of base points. p' is *defined* if and only if $\bar{p}$ is a valid input for m (Section 4.5.1). In this way, a single single base point may produce multiple base points as output, reflecting potentially non-monotonic or non-deterministic properties of cognitive operations.

We overload the functionality of J so that $J[p, m] = p'$, where p is a state point (rather than just a base point), and define $J[p, m]$ as follows: $J[p, m] = [\texttt{replace}(\bar{p} \in_s p, J[\bar{p}, m])]$. $J[p, m] = p'$ is *defined* if and only if some $J[\bar{p} \in p, m]$ is defined.

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 contain the same structural variables as those of the input.

A cognitive operation is called monotonic if it always yields a base point as an output given a base point input ($J[\bar{p}, m] = \bar{p'}$).

This paper will focus on cases where the type of cognitive state remains constant, but it is possible for $J[\bar{p}, m] = p'$ where $\bar{p}$ is of structure (α) and $\bar{p'} \in p'$ to be of structure (β). Future models of human cognition may well rely on background knowledge which draws inferences from multiple types of non-monotonic logics.

A *cognitive transition model* (CTM) describes an ordered procession of cognitive operations following an initial epistemic state as follows: CTM $\pi = (s_i \longmapsto m_1 \longmapsto ... \longmapsto m_n)$ where s_i is a state point, and m_k is a cognitive operation. A *partial cognitive transition model* (pCTM) $\pi = (m_1 \longmapsto ... \longmapsto m_n)$ describes a procession of cognitive operations without an an initial state point.

The CTM $\pi = (s_i \longmapsto m_1 \longmapsto ... \longmapsto m_n)$, for SCP task $\Pi = (s_i, M, f(), \gamma)$ is *defined* if and only if $J[p_k, m_k]$ is defined for $p_k = J[p_{k-1}, m_k], k > 2$, and $J[s_i, m_1]$ is defined. The set of possible CTMs for SCP task Π is given by $L[\Pi] = \{\pi\}$, where π is a defined CTM for Π.

An SCP $\mu = (\pi, f())$ pairs a CTM π with an external activation function $f()$ (4.6). An SCP $\mu = (\pi, f())$ is defined for SCP task $\Pi = (s_i, M, f(), \gamma)$ if and only if $\pi \in L[\Pi]$.

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

Finally, we define a *realised SCP* $r = (k, f())$ where $k \in K[\pi]$, and $K[\pi] = \{s_i \longmapsto (m_1, \bar{p}_1) \longmapsto ... \longmapsto (m_n, \bar{p}_n)\}$ where, for every pair $(m_i, \bar{p}_i)$, $\bar{p}_i \in_s J[\bar{p}_{i-1}, m_i]$. Realised SCPs describe a single agent's interpretation of an SCP and associate only a single epistemic state, rather than a state point to the output each cognitive operation in an SCP.

$r = (k, f())$ is a realised SCP for SCP $\mu = (\pi, f())$, with $\pi = (s_i \longmapsto m_1 \longmapsto ... \longmapsto m_n)$ and is given by $k = (s_i \longmapsto (m_i, \bar{p_1} \in J[\bar{p}_0, m]) \longmapsto ... \longmapsto (m_n, \bar{p_n} \in J[\bar{p}_{n-1}, m]))$.

A cognitive operation is called *monotonic* if it is guaranteed to produce only one base point as output for each base point passed as input. When every cognitive operation in M is monotonic, it is trivially easy to transform an SCP into a realised SCP and vice versa @TODOproof. In cases where SCPs are monotonic, we will use the SCP and its realised SCP interchangeably. The number of realised SCPs for an SCP is the same as the number of base points in the final state point of that SCP.

4.3 SCP Tasks vs. SCPs vs. Realised SCPs

SCP tasks describe a problem that needs to be solved and the limitations which the solution must to adhere to; SCPs describe a single 'recipe' that can be used to describe a sequence of well-founded, non-monotonic operations which an agent might use to model a problem; and realised SCPs describe how adherence to specific SCP can result in some agent coming to a conclusion that matches empirical data.

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 7), or no possible SCPs at all.

Figure 4.2 explores a situation in which students are asked if they will pass their next exam. Though all subsequent examples in this paper will consider SCPs in a more formal context which is grounded in specific nonmonotonic logical interpretations, this conceptual captures the intuition of the different levels of granularity which SCP Tasks, SCPs, and realised SCPs capture.

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 boolean Propositional Logic can be certain that any SCP they create should be expressive enough to pass a boolean knowledge base and possible world to a cognitive operation . Thus, it might suffice to simply define $s_i = (S)$ where S 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. To model the Suppression Task or WCS it might seem intuitive simply to append a set of conditional rules Δ to the state used for the propositional case and to use lm $\mathrm{wc}(s_i)$ as an external function to evaluate the SCP. 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, we will see that using $s_k = (S, \Delta, V, R)$, where V is a set of (variable name, value pairs) and R is a categorization variable, seems 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,

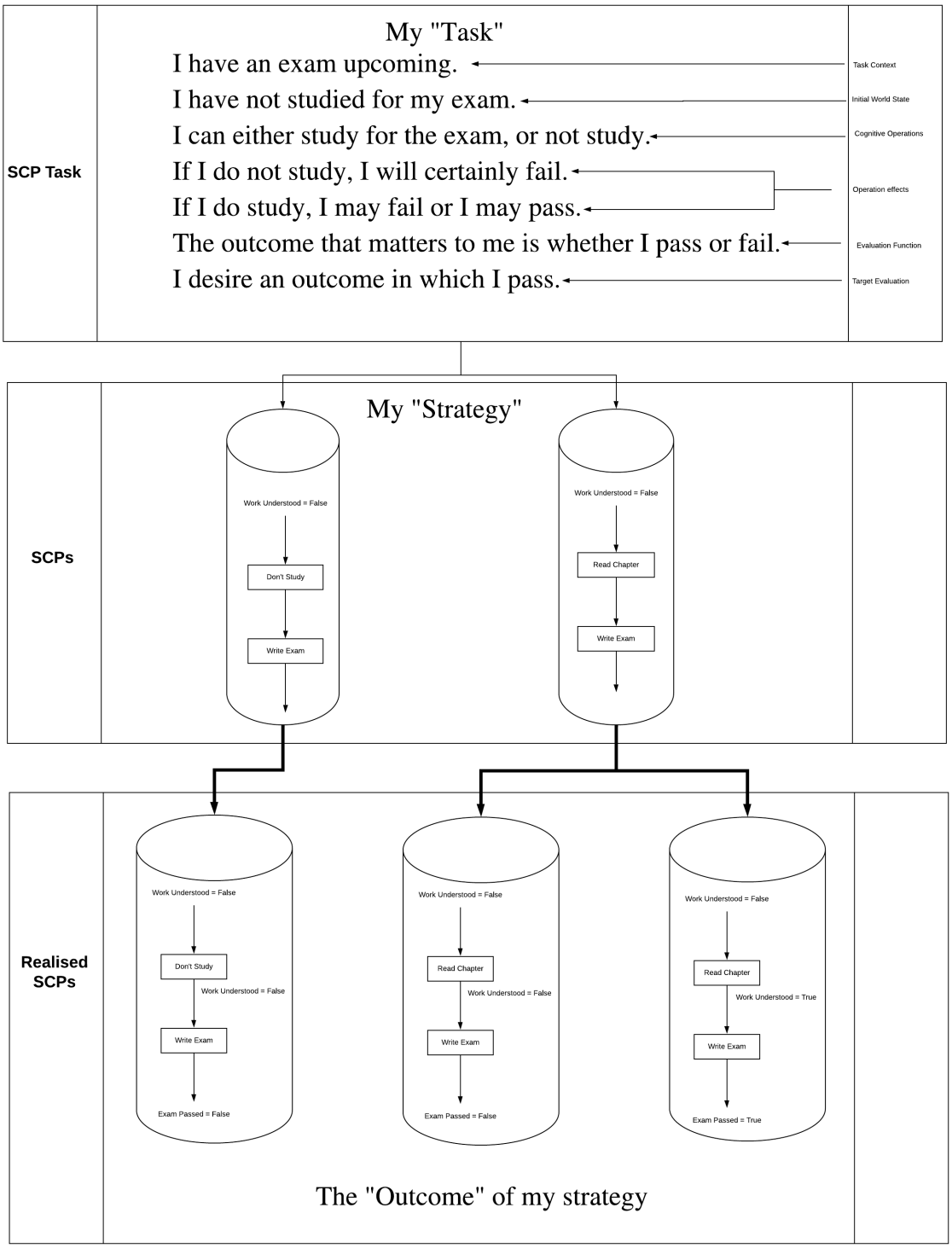


FIGURE 4.2: An SCP-like examination of a mental task, showing the relative levels of granularity of SCP Tasks, SCPs, and Realised SCPs.

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.

4.4.1 Common State Point Properties

Though state points are largely unbounded in the information that they can contain and communicate, this thesis will restrict itself to epistemic state structures which carry intuition both from the underlying logics which they utilise, and from properties of search in the field of AI Planning (Korf, 1987).

The possible world: V

The possible world variable V encodes information about the current world state the agent believes itself to be in. As in Section 2, the possible world variable consists of a conjunction of variable pairs of the form $(\text{atom}, \text{value} \in \text{domain}_{\text{atom}})$, where atom occurs only once. Unlike in a true possible world, there exist atoms $\in \Sigma$ which do not occur in V.

$V[\phi]$ returns the pair $(\phi, \text{value}) \in V$. An interpretation of formula ϕ using V, $I_v[\phi]$ occurs as with any possible world except that, when an atomic name a which occurs in ϕ when $V[a]$ is undefined is given the default value of that language during computation (u in Łukasiewicz logic, and $\bot$ in propositional logic.). This is a necessity born of the fact that infinite space is required to explicitly store every possible variable name. The reason that V is used throughout this thesis is threefold: it ensures that the possible world of the agent at any state of its cognition is always known; it allows inferences drawn using one logic to be utilised as background knowledge in another logic; and in practical implementation it can prevent recalculation of complex rules that have been previously interpreted.

Propositional Rule Set: S

The propositional rule set S is defined precisely as the knowledge base was defined in Section 2.4.1.

The set of conditionals: Δ

The set of conditionals Δ is defined precisely as in the conditional knowledge base in Section 2.4.1.

4.4.2 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 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: $R = \{obligate : \{(u|r)\}, factual : \{(s|r)\}\}$. 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 subsequent cognitive operations, and it is completely possible that some operation m_k might change R is such a way that future operation m_{k+l} produces different output. This is a technique we will exploit in several examples in this paper.

4.5 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 and the initial state s_i is a base point, then p_n will be a base point. 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 tolens* asymmetry, suppression, denial of the antecedent, etc.).

4.5.1 Pre-conditions and Effects

The precondition χ of a cognitive operation m refers to those conditions which 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 for 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 of she's been drinking. Imagine further that this operation takes base points which are either of propositional or default structure (Section 2.7) 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.

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.

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 $x \longmapsto$ `goToClub` $\longmapsto$ `goHomeByTrain` $\longmapsto$ `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 shows strengths and weaknesses. A final approach to consider is the *hybrid validity* approach. With this approach, all cognitive operations are assumed to be valid, provided that the output base points of the previous operation are of a suitable input structure 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 approached, its universal appropriateness 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.

4.5.2 Optimality, Satisfaction, Validity

Validity

As discussed in Section 4.5.1 validity can be defined for a cognitive operation and its input. Realised SCPs describe both the operation and the input for an SCP and allow us to define SCP validity as follows:

A realised SCP $r = (k \in K[\pi], f()), k = (s_i \longmapsto (m_1, s_1) \longmapsto ... \longmapsto (m_n, s_n))$ with evaluation function $f()$ is valid if and only if s_i is a state point, every $m \in \pi$ is valid (according the validity requirements defined by the researcher), and $f(k)$ is defined.

An SCP $(\pi, f())$ is valid if and only if there exists some realised SCP $r = (k \in K[\pi], f())$ which is valid.

An SCP Task $\Pi = (s_i, M, \gamma, f())$ is valid if and only if there exists some SCP $((\pi, f())$, with $(\pi = (s_i \longmapsto m_1 \in M \longmapsto ... \longmapsto m_n \in M)$ which is valid.

Validity does not require that $f(\pi) = \gamma$, only that external function $f()$ is able to make some prediction or set of predictions based on π.

Satisfaction

A realised SCP $r = (k \in K[\pi], f())$, $k = (s_i \longmapsto (m_1, s_1) \longmapsto ... \longmapsto (m_n, s_n))$ with evaluation function $f()$ satisfies goal condition γ, written $r \models \gamma$ if and only if r is valid, and $f(k) \models \gamma$.

An SCP $(\pi, f())$ satisfies γ, written $(\pi, f()) \models \gamma$, if and only if there exists some realised SCP $r = (k \in K[\pi], f())$ for which $f(k) \models \gamma$.

An SCP Task $\Pi = (s + i, M, \gamma, f())$ satisfies γ, written $\Pi \models \gamma$, if and only if there exists some SCP $((\pi, f())$, with $(\pi = (s_i \longmapsto m_1 \in M \longmapsto ... \longmapsto m_n \in M)$ which satisfies γ.

Heuristic Searches and machine learning techniques are can be to find satisfying solutions for situations in which it is possible for an answer to be good enough for practical purposes, but search time or space is infeasible.

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). As with all exhaustive search techniques, optimality can be guaranteed for searches of restricted depth even when SCP space contains infinite loops.

A realised SCP $r = (k \in K[\pi], f())$, $k = (s_i \longmapsto (m_1, s_1) \longmapsto ... \longmapsto (m_n, s_n))$ generated from SCP $(\pi = (s_i \longmapsto m_1 \longmapsto ... \longmapsto m_n)$ with evaluation function $f()$ is optimal for goal condition γ and heuristic function $g()$, if and only if $r \models \gamma$, and $\forall r' = (k' \in [K[\pi]], f()), g(r) \geq g(r')$ for all case in which $f(r') \models \gamma$.

An SCP $(\pi, f())$ generated from $\Pi = (s + i, M, \gamma, f())$ is optimal for goal condition γ and heuristic function $g()$, if and only if $(\pi, f()) \models \gamma$ and there exists no other γ - satisfying SCP $(\pi, f())$ which can be generated from Π for which $g(\pi') > g(\pi)$.

Optimality is not defined for SCP Tasks.

Formally, an SCP $\pi = (\pi_0 \longmapsto ... \longmapsto \pi_n)$ generated from SCP Task $\Pi = (s_i, M, \gamma, f())$ with evaluation function $h()$ is optimal if and only if $f(\pi)$ 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.

4.5.3 Credulous and Skeptical Validity

A planning task may permit an SCP in which all or only some of the associated realised SCPs are valid, satisfying, or optimal. An SCP is called *credulously valid* if $f(\pi)$ is valid for *at least one* epistemic state in the final state p_n. An SCP is called *sceptically valid* if $f(\pi)$ is valid.for *every* epistemic state in the final state. In cases where all operations are monotonic, sceptical validity is the same as credulous validity. Similar logic applies the concepts of sceptical satisfaction, credulous satisfaction, sceptical optimality, and credulous optimality.

4.6 External Evaluation Functions

The external evaluation function $f()$ is responsible for evaluating an SCP or reaslised SCP in order to map it to the observed or predicted empirical data observed in respondents. The $f()$ function in general is an unbounded tool for making a decision with a CTM.

As a function, defining $f()$ forces researchers to specify what information in the epistemic state corresponds to empirical data. The exact procedure by which $f()$ makes this decision should correspond to researcher's chosen criteria for decision making.

All external evaluation functions that we examine in this paper will use the final state point p_n which results from evaluating the state point returned by applying the penultimate state point to the last cognitive operation $J[p_{n-1}, m_{n-1}]$. But other external functions which take into account aspects of the structure of π or intermediate epistemic states can be formulated.

The question of how to select the right external evaluation function is not a focus of this thesis, but, where necessary, all used evaluation functions will be justified with respect to either common-sense reasoning, or evidence will be given that such a function is already used in a more informal way in the existing literature for that experiment. External evaluation functions will be provided using pseudocode.

A goal $\gamma = \{P_1 : \{\alpha_1, ...\alpha_m\}, ..., P_n : \{\beta_1, ...\beta_p\}$ is a labelled set of properties, each of which correspond to some empirically observed response to the task at hand. For example, in the Suppression Task (Section 3.1) we might define:

$$\gamma_{\text{sup}} = \{\text{'conclusion'} : \text{'we are unsure if she will study late in the library'}\}$$

. The precise mechanism by which the conclusions in γ_{sup} is reached is assumed to be unknown, and it is the purpose of the SCP framework to find a well-founded cognitive process, and corresponding external evaluation function, which matches this empirical answer set.

We say that a realised SCP, $r = (k, f())$, *satisfies* a goal $\gamma = \{P_1 : \{\alpha_1, ...\alpha_m\}, ..., P_n : \{\beta_1, ...\beta_p\}$, written $f(k) \models \gamma$, if and only if for every property label $P_i \in f(k)$, we also find that $P_i \in \gamma$, and $f(k)[P_i] \in \gamma[P_i]$.

We say that an SCP, $\mu = (\pi, f())$, *strictly satisfies* a goal γ, written $f(\pi) \models_{\text{strict}} \gamma$, if and only if, *for every* realised SCP $r = (k = K[\pi], f())$ we find that $f(k) \models \gamma$.

We say that an SCP, $\mu = (\pi, f())$, *weakly satisfies* a goal γ, written $f(\pi) \models_{\text{weak}} \gamma$, if and only if, *for some* realised SCP $r = (k = K[\pi], f())$ we find that $f(k) \models \gamma$.

These two definitions of satisfaction for SCPs will be used later to motivate an alternate method for modelling individual reasoners in the Wason Selection Task using SCPs (Section 6.3). When an SCP is monotonic, strict and weak satisfaction are identical and we simply write $f(\pi) \models_{\text{weak}} \gamma$.

Finally, we define the *preferred response* $p()$ for the output of an external evaluation function $f()$ and CTM π as follows:

Algorithm 2: $\mathrm{p}(f(\pi, \mathrm{pref}))$: a method which removes unpreferred responses from a list and replaces them with the preferred response (if it was in the original list).

```
Function p(f(π), pref) is
    responses:=f(π);
    finalResponses:={} for category ∈ responses do
        if pref[category] ∈ responses[category] then
            finalResponses[category]=pref[category]
        else
            finalResponses[category]=responses[category]
        end
        return finalResponses
    end
end
```

A preferred response allows us to specify a hierachy of achievable outputs from an SCP or realised SCP and a practical application will be shown for modelling individual reasoners in the WST (Section 6.3).

4.7 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 4.5.1), *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 (Korf, 1996) but also have some potential using backwards or biderectional search (De Champeaux, 1983).

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 4.7.1 and Section 4.7.2 discuss the philosophy, applications and mechanical considerations of these two search approaches.

4.7.1 De-Novo SCP Searching

De Novo (from new) search is a search technique in which a final desirable world state is achieved by generating a sequence of cognitive operations and using a known initial state as the input to the first cognitive function. An SCP Task contains all the information required to conduct a de Novo search through the space of allowable SCPs for a given cognitive task. The exact search techniques used can be easily varied, but we will consider de Novo search in terms of a breadth-first traversal Zhou and Hansen, 2006.

Figure 4.3 illustrates the process by which breadth first search over an SCP task $\Pi = (x, M, \gamma, f())$ can be conducted. Search terminates if a structural inconsistency between two states in $\pi = (x \longmapsto A_0 \longmapsto ... \longmapsto A_n)$ occurs (e.g. output state point structure of A_k does not match expected input structure of A_{k+1}). Operator sequences are added to the list of solutions if and only if they meet the validity requirements of the search being used, as discussed in Section 4.5.2.

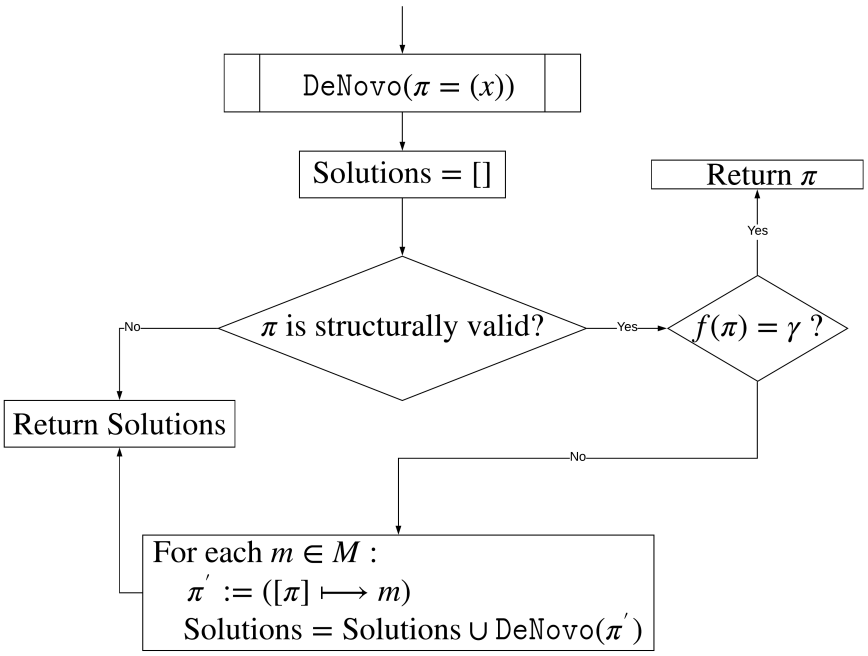


FIGURE 4.3: A breadth-first search algorithm for De Novo SCP search with SCP Task $\Pi = (x, M, \gamma, f())$.

In practice it becomes necessary to limit the search depth of the algorithm to search tractable in most cases, and in some cases, search space and solution space may both be infinitly large, as suggested by Proof **??**.

4.7.2 Insertion Search

Changing the structure of an existing SCP allows researchers to model deviations from standard reasoning for a task. For example, in Section 6.2 we show that slight modifications to an SCP that models the standard case of the suppression task can lead to an SCP which is able to model reasoners who achieve the classical valid conclusion that she will study late in the library.

The insertion search modifies an existing SCP:

$$\mu = (\pi = (s_i \longmapsto A_0 \longmapsto ... \longmapsto A_n), f())$$

to produce a new SCP:

$$\mu' = (\pi' = \{(s_i \longmapsto B_0^* \longmapsto A_0 \longmapsto\longmapsto B_1^* ... \longmapsto B_n^* \longmapsto A_n \longmapsto B_{n+1}^*), f()\})$$

Where $B_i^* = (B_0 \longmapsto ... \longmapsto B_n)$, $B \in M$ or is the trivial operation T which returns the input state point as output. When describing the SCP, T operations are generally omitted.

In general, Insertion Search is a more difficult search type than De-Novo Search (which is equivalent to an Insertion Search in which the initial SCP is empty). The reason for this is that it is possible for a single insertion to result in an invalid SCP, but adding additional cognitive states may revalidate the SCP (Proof **??**).

This consideration means that it is potentially necessary to insert an unbounded number of new cognitive operations at each point in π to find those insertions which are valid. When

the total number of new cognitive operations to insert is bounded by N, however, it becomes possible to generate every possible subsequence of operation insertions of length $\geq N - k$, where k is the number of insertions already used in the search, for each position B_i^* and determine for which of the resulting SCP cognitive operations sequences $f(\pi')$ is valid.

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. That is, $\mu = (\pi = (S_i \longmapsto m), f())$ can model any cognitive task with a complex enough definition of m.

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 $f(x \longmapsto A)$, the only possible changes are: $f(x \longmapsto \bar{B} \longmapsto A \longmapsto \bar{B})$, $f(x \longmapsto A \longmapsto \bar{B})$, $f(x \longmapsto \bar{B} \longmapsto A)$, $f(x \longmapsto \bar{B})$, 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. In Section 6.2 and Section **??** we will explore examples 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 taking a state point as input and resulting in a state point as output. 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 4.1a and Figure 4.1b in the previous chapter, captured 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 Section 2.6, its individual steps are obvious candidate cognitive operations.

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, as well as defining several well-founded cognitive operations algorithmically with respect to their preconditions, effects, and output properties in Section 5.4.

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.

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 pCTMs $(A_0 \in \Omega \longmapsto ... \longmapsto A_n \in \Omega)$ with Ω^*. There cannot and never will be an exhaustive search procedure to evaluate every possible way of manipulating an epistemic state (Proofs **??** and **??** 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 **??** defines those cases in which there is redundancy in the set of all possible cognitive operations, Lemma **??** focuses this concept on specific cognitive tasks which are to be modelled using SCPs.

The total number of possible non-trivial SCPs of length L for an SCP Task is given by $(|M|^L - R)$ where $|M|$ is the number of elements in the set of cognitive operations M, 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 **??** 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-founded[1].

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

The second considerations 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 (for example, Heit, Hahn, and Feeney, 2005's Diversity Principle). 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 to 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

This section discusses the intuition and precise definition of several cognitive operations which will be used throughout the remainder of this paper. These operations allow the SCP Framework to mimic non-monotonic logics for the experiments discussed in Chapter TODOchapter.

A cognitive operation m is always evaluated in terms of each base point of input state point, and each base point can be considered independent from the others. For these definitions, a base point $\bar{p}$ is assumed as input and a state point is returned as output. Function $\chi(\bar{p})$ determines the validity validity of the input base points. Σ_χ is called the input alphabet and defines the minimal set of an variables names which must be present in $\bar{p}$ for the operation to be allowable.

Accessing a specific element α_k in the structural variables of base point $\bar{p} = (\alpha_1, .., \alpha_n)$ is short-handed to $\bar{p}[\alpha_k]$. $\bar{p}[\alpha_k]$ returns α_k if and only if $\alpha_k \in \bar{p}$ and returns the empty set otherwise. This indexing notation is also used to append a structural variable to $\bar{p}$ and $\bar{p}[\alpha_j] := t$ removes $\bar{p}[\alpha_j]$ from $\bar{p}$ if it already exists and creates a new structural variable $\alpha_j = t$ in the resulting state point p'.

5.4.2 Preconditions

The input alphabet $\Sigma_\chi = \{\alpha_1, ..., \alpha_k\}$ of a cognitive function m describes the set of structural variables which must be present in the epistemic states $\bar{p}$ of the input state point p. The alphabet of structural variables in p is given by Σ_p. Hybrid validity, as defined in Section 4.5.2,

can be algorithmically illustrated as follows:

Algorithm 3: $\chi(\bar{p})$: Hybrid validity requirements for $J[p, m]$ to be defined.

Function $\Sigma_\chi\ (\bar{p})$ **is**
  $\Sigma_\chi = \{\alpha_1, ..., \alpha_n\}$;
  if $\forall v \in \Sigma_\chi, v \in \Sigma_p$ **then**
    return "Defined"
  else
    return "Undefined"
  end
end

All cognitive operations which we define in this section will be assumed to have a known $Sigma_\chi$ and will only proceed to their effect e if $\Sigma_\chi(p) =$ "Defined". The output state point of a defined $J[\bar{p}, m]$ is the returned value of $m[e](\bar{p})$.

5.4.3 Propositional Logic

An SCP for propositional logic must be able to take a set of rules and facts and draw the set of all valid inferences from that set using the standard propositional rules discussed in Section 2.1. To that purpose, we introduce a cognitive operation called `th` which draws all classically valid inferences from a set of logical rules[2].

`th` is by its nature a contrived cognitive operation and there is no expectation that it truly reflects human cognition. Instead, it serves as a benchmark operation with which to compare the results derived from knowledge base-driven logical approaches. Even a simpler congitive operation called `SAT` which determines if the given knowledge base it satisfiable still represents an *NP-complete* problem and is infeasible for any large knowledge base (Schaefer, 1978).

However, variations on the intuition of these two problems have real applications in cognitive modelling with SCPs and will be discussed throughout this section.

Algorithm 4: $\texttt{th}(\bar{p})$: generates the potentially infinite set of possible classical inferences from $\bar{p}[S]$.

Function $e(\bar{p})$ **is**
  $\bar{p}[S] := \bar{p}[S] \cup [(\text{atom name} \leftarrow \text{value}) |\ (\text{atom name}, \text{value}) \in \bar{p}[V]$ and value $\neq u$;
  $\bar{p}[S] := \bar{p}[S] \cup [A \mid$ clause A would not make $\bar{p}[S]$ inconsistent$]$;
  $\bar{p}[V] := [B \mid B \leftarrow body \in \bar{p}[S]$ and $I(body) = \top$ or $I(body) = \bot]$;
  return $\bar{p}$
end

A realised SCP $r = (k = K[\pi], f())$ is said to *model the classical inference*, if $f(k) = f(s_i \in \pi \longmapsto \texttt{th})$. An SCP π is said to model the classical inference if some realised SCP generated from π models the classical inference.

5.4.4 Variable Insertion

Variable insertion is a single cognitive operation with a highly intuitive interpretation and justification. By its nature, the set of variables $V \in \bar{p}$ in an epistemic state $\bar{p}$ is not a true possible world (as discussed in Section 4.4.1), and may omit variables present in the real

[2]The reason that we do not introduce the functionality of the `th` cognition operation in the external evaluation function $f()$ is that doing so would make it impossible to directly use the derived theorems as part of another SCP.

possible world. But one these variables may become apparent later. For example, when interpreting a conditional as a license for implication, or when storing the result of a calculated mathematical problem.

It seems reasonable then that a cognitive operation `addV` which appends a variable to base point V would be required to expand to the knowledge base to incorporate new facts.

Algorithm 5: `addV`$(\bar{p})$: adds a variable v, defined *a priori*

```
/* The variable to be added, defined before SCP execution.          */
```
$v := \alpha = (\text{atom name}, \text{value})$;

Function $e(\bar{p})$ **is**

```
    /* Remove all possible assignments of the variable from the list of
       variables                                                      */
```
$\bar{p}[V] := \bar{p}[V] - [\text{atom name}, x \in \text{Domain}_{\text{atom name}}]$ `/* Add the new variable and its assignment to the list of variables */`

$\bar{p}[V] := \bar{p}[V] \cup \alpha$;

return $\bar{p}$

end

In order to ensure that the SCP still acts as a pipeline of information, it is necessary that `Add`α for variable $\alpha = (\text{atom name}, \text{value})$ is defined as part of the allowable operations of the SCP Task Π which generates the SCPs and realised SCPs that will make use of it. The core principle of this pipeline framework is that each usable component (cognitive operation) of that framework which is not the input value should not be modified.

5.4.5 Variable Deletion

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

This is not the only imaginable case where a variable might be removed from the knowledge base of the person being modelled. The size of the knowledge base used for cognitive modelling is always implicitly restricted to relevant variables. Only those variables whose values might reasonably be expected to affect the final conclusions drawn with regard to the research question 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.

Forgetting a variable in an SCP is not-trivial task, just as removing a variable in an AI planning task is non-trivial. Multiple intuitions for how it may best be done occur, but we will keep this definition as simple as possible for the examples that will be discussed in Chapter 6. We introduce three cognitive operations `remove`$_{\top}V$, `remove`$_{\bot}V$, and `remove`$_{u}V$. `remove`$_{\top}$ assumes that conditionals and rules which depend on the removed variable are

validated by its removal. $\texttt{remove}_{\perp}V$ assumes that rules and conditionals are invalidated by the removal. $\texttt{remove}_{u}V$ assumes that no conclusion can be drawn without this information. These three approaches can be differentiated by these examples:

- "If aliens invade, then I will hide in my bunker."
- "If I do not fail my exam, then I will graduate."
- "Unless I am a boy, I am a girl."

In the first example, the presumed default position is that I will not hide in my bunker. In the second, the default position is that I will graduate. And in the third example there is no apparent default position. In the absence of the conditions in each each of these conditionals we would probably conclude that, I am probably not hiding in my bunker, I probably graduated, and that my gender is uncertain. This observation implies that a mechanism by which to determine the plausibility of default positions is needed to translate such default theories into a non-monotonic framework. In the absence of such a mechanism we will generally assume $\texttt{remove}_{\perp}V$, as with the closed-world assumption.

Algorithm 6: $\texttt{remove}_{\perp}\mathrm{V}(\bar{p})$: removes a variable name v, defined *a priori*

```
/* The variable to be deleted, defined before SCP execution.          */
v := atom name;
Function e(p̄) is
    /* Remove α from the possible world variables.                    */
    ∀_{α=(v,value∈domain(v))} p̄[V] := p̄[V] − α;
    for rule = (head ← body) ∈ p̄[S] do
        if head = v then
            p̄[S] := p̄[S] − rule;
        end
        if v ∈ body then
            replace all subclauses (v ∨ φ), (v ∧ φ) ∈ p̄[S] with (φ);
        end
    end
    for rule = (head|body) ∈ p̄[Δ] do
        if head = v then
            p̄[Δ] := p̄[Δ] − rule;
        end
        if v ∈ body then
            replace all subclauses (v ∨ φ), (v ∧ φ) ∈ p̄[Δ] with (φ);
        end
    end
    /* This is where remove_⊥V differs from remove_⊤V, and remove_uV  */
    replace all rules head ← v ∈ p̄[S] with ⊥;
    replace all rules head ← ¬v ∈ p̄[S] with ¬⊥;
    replace all rules (head|v) with ⊥;
    replace all rules (head|¬v) with ¬⊥;
    return p̄
end
```

5.4.6 Variable Fixation

The next 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*. This operation

may prove contentious to those who are experienced with mathematical logic, but holds some merit when justified from a psychological perspective.

Consider a person who strongly doubts the effectiveness of vaccines, we will call her Karen. Karen started her day convinced that giving her child the MMR vaccine is more dangerous than the disease itself. Later that day Karen spoke to her doctor who strongly advised that she vaccinate her child. He offered her a variety of peer-reviewed papers and studies that showed the relative safety of the vaccination. Karen listened carefully to the trained medical professional, and then went home. After some thought Karen decided that he was wrong, and her opinion on vaccines didn't change.

In this example Karen shows a very powerful type of cognitive bias, the unwillingness to change 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.

In order to implement this operation, we will make use of the categorization variable R first discussed in Section 4.4.1.

Algorithm 7: `FixV`$(\bar{p})$: fixes a variable name v, defined *a priori*

```
/* The variable to be fixed, defined before SCP execution.          */
```
$v := (\alpha = (\text{atom name}, \text{value}))$;
Function *e($\bar{p}$)* **is**
  $\bar{p}[V] := \bar{p}[V] - (\text{atom name}, \text{value} \in \text{domain}) \in \bar{p}[V]) \cup \alpha$;
  $\bar{p}[R][\text{fixed}] := \bar{p}[R][\text{fixed}] \cup v$;
  return $\bar{p}$
end

This algorithm removes 'atom name' from the possible world V and readds it with the value to which it is to be fixed. The atom is then added to the list of fixed variables kept by $\bar{p}[R]$. This list of fixed variables is used by other operations which evaluate state points later in the SCP or realised SCP, and, as per their encoding, they will not modify the assignment of this variable.

5.4.7 Adding Abnormalities

As discussed in Section 2.3.1, one way to interpret conditionals in reasoning tasks is as licenses for implication. One possible definition for a cognitive operation that performs this same task by transforming defeasible conditional facts to propositional logic rules is as given

by Algorithm 8.

Algorithm 8: `addAB`$(\bar{p})$

Function *e*$(\bar{p})$ **is**
  `/* Variables for which an abnormality has already been created */`
  for $(\psi|\phi) \in \bar{p}[\Delta]$ **do**
    $k :=$ the lowest natural number for which $\text{ab}_k \notin \bar{p}[S]$;
    all dependencies $:= [A|(\psi|A) \in \bar{p}[\Delta]]$;
    for $A \in$ *all dependencies* **do**
      head $:= \psi$;
      `/* All other variables which can make` ψ `true are treated as though they could also falsify` ψ`. */`
      current dependencies := all dependencies − A;
      if *current dependencies* $= \{\}$ **then**
        current dependencies $:= \bot$;
      end
      body $:=$ (current dependencies$_1 \vee ... \vee$ current dependencies$_n$);
      `/* Add the conditional as a license for implication to the set of rules. */`
      $\bar{p}[S] := \bar{p}[S] \cup (\text{head} \leftarrow A \wedge \neg \text{ab}_k)$;
      $\bar{p}[S] := \text{ab}_k \leftarrow \neg body$;
    end
    `/* Add the new abnormality to the possible world` V`. */`
    $\bar{p}[V] := \bar{p}[V] \cup (\text{ab}_k, u)$;
  end
  `/* Remove all conditionals now that they have been interpreted as licences for implication. */`
  $\Delta := \{\}$;
  return $\bar{p}$
end

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. Algorithm **??** shows one way in which this can be achieved.

Algorithm 9: `wcs`$(\bar{p})$

Function *e*$(\bar{p})$ **is**
  Replace all rules $\in \bar{p}[S]$ of the form $A \leftarrow \text{body}_1, ..., A \leftarrow \text{body}_n$ with $A \leftarrow \text{body}_1 \vee ... \vee \text{body}_n$;
  Replace all occurrences of $\leftarrow \in \bar{p}[S]$ with $\leftrightarrow$;
  return $\bar{p}$
end

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 (explictly), or to Unknown (implicitly). Algorithm 10 illustrates one

way in which this cognitive operation can be implemented for any epistemic state of the form $s = (KB, V, ...)$.

Algorithm 10: `semantic`$(\bar{p})$

Function *e($\bar{p}$)* **is**
  `/* All least models that can achieved by applying the semantic operator to the current state point. */`
  $p' = \{phi_{SvL}(\bar{p})\}$;
  return p'
end
Function $\phi_{SvL}(\bar{p}$ **is**
  if *there exists a clause* $A \leftarrow body \in \bar{p}[S]$ *with* $I_V(body) = \top$ **then**
    if $A \notin \bar{p}[fixed]$ **then**
      $\bar{p}[V][A] := \top$
    end
  end
  if *there exists a clause* $A \leftarrow body \in \bar{p}[S]$ *and for all clauses* $A \leftarrow body \in \bar{p}[S]$ *we find* $I_V(body) = \bot$ **then**
    if $A \notin \bar{p}[fixed]$ **then**
      $\bar{p}[V][A] := \bot$
    end
  end
  return $\bar{p}$
end

5.4.10 Adding Abducibles

Next, we define a cognitive operation which uses the set of possible abducibles, given by $\bar{p}['R'][\text{'}abducibles\text{'}]$. We have already used the intuition of this operation to handle the WST and derive the general answer set. Later we will see that it can also be used to model individual reasoners in the Suppression Task. `AddExp` adds possible explanations η to the set of rules S in the epistemic state and is defined as follows:

Algorithm 11: `addExp`

Function $e(\bar{p})$ **is**
  abducibles:=$\bar{p}$['abducibles'];
  $p' := []$;
  for *every unique subset* $A \in abducibles$ **do**
    newP := $\bar{p}$;
    newP['S'] = newP['S'] $\cup \{a \in A \leftarrow \top\}$;
    newP['η'] = A;
    $p' := p' \cup$ newP;
  end
  return p'
end

5.4.11 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 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 $\frac{condition:exception}{conclusion}$ @TODOrewriteEquationWithStandardSymbols.

5.5 Cognitive Operations as Aggregates

An obvious and interesting idea follows from Proof **??** 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 (Drummond, 2002) 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 **??** 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.

For example, the cognitive processes `wc` and semantic could be combined together to form a single wcs operator as follows:

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.

A researcher could decide that it is unreasonable to expect any cognitive operations to seperate `wc` and `semantic`, as the both form part of the same nonmonotonic logic. They might then create the new aggregate complex operation `wcs` which applies the WCS in its entirety as follows:

Algorithm 12: `wcs`$(\bar{p})$

Function $e(\bar{p})$ **is**
  $\bar{p}' :=$ `wc`$(\bar{p})$;
  $p' :=$ `semantic`$(\bar{p}')$;
  return p'
end

They could even go a step further and define the cognitive operation `abwcs` which first interprets the conditionals in $p[\Delta]$ and then applies the WCS to the result. This algorithm is

as follows:

Algorithm 13: `abwcs`$(\bar{p})$

Function *e*$(\bar{p})$ **is**
  $\bar{p}' :=$ `addAB`$(\bar{p})$;
  $\bar{p}' :=$ `wc`$(\bar{p})$;
  $p' :=$ `semantic`$(\bar{p}')$;
  return p'
end

The reader will note that `abwcs` now provides precisely the same functionality as the examples of the WCS we examined in Chapter 3, and Chapter 6 will show that these three operations in sequence can indeed be modelled in the SCP framework and come to the same conclusions as when the WCS is applied in the original experiments.

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.

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 struct(s_k) and output base point structure struct(s_{k+1}) is called a *structural transformation operation* if and only if struct$(s_k) \neq$ 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 exmaple, could be accurately represented with an initial state $s_x = (S)$ where S is simply the knowledge base of facts and rules given; but a researcher might equally use an epistemic state $s_y = (S, V)$ where a variable V is intended to store the values of each variable in S; or even $s_z = (S, V, D)$ 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 hold 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 struct(s_z) to one

of 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.

5.7 Two Approaches to Creating Cognitive Operations

5.8 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 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 (Gonçalves, Perra, and Vespignani, 2011)); 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 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 relevent 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.

5.9 Empirical Appoximation

Contrasting theoretical approximation is the field of constrained empirical approximation. Techniques ranging from neural networks and Markov models genetical 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.

@TODOfinish

Chapter 6

Modelling Experiments with SCPs

6.1 Overview

As with any cognitive framework, the most important metric for judging the success of SCPs comes from testing how well SCPs can approximate empirical data across tasks. This chapter will show the suitability of SCPs with a common set of allowable cognitive operations for modelling several well-studied experiments in cognitive modelling.

In particular, we will show that the Wason Selection Task, Suppression Task and @TODO can be modelled at both the general and individual reasoner level with SCPs in a formulation that intuitive and consistent with the WCS approach already described in Section @TODOref.

6.2 Suppression Task

To model this experiment in the SCP framework, we must first define the SCP Task $\Pi_{sup} = (s_i, M, f(), \gamma)$ which outlines the restrictions and goals of the model. The structure of s_i must be robust enough to capture information regarding a set of rules, variable assignments in the possible world, and conditionals. To this end we define:

$$s_i = \{[s_i^{\text{el}}, s_i^{\text{elo}}\}$$
$$s_i^{\text{el}} = \{S, \Delta_{\text{noSup}}, V\}$$
$$s_i^{\text{elo}} = \{S, \Delta_{\text{Sup}}, V\}$$

For the the only unconditional rule, that she has an essay to write, we define $S = \{(e \leftarrow \top)\}$. The set of conditional rules $\Delta_{\text{noSup}} = \{l|e\}$ captures the probable assumption that she will study late in the library if she has an essay to write, $\Delta_{\text{sup}} = \{(l|e), (l|o)\}$ captures the addition of the conditional that if the library is open she will study late in the library. And the possible world $V = \{(e, u), (l, u), (o, u)\}$ sets all variable names occurring in S and δ to unknown.

The second consideration M, denotes the set of possible cognitive operations which may occur in the resulting SCPs and realised SCPs of Π_{sup}. Following the intuition we used in Section @TODOref we choose cognitive operations that enable application of the WCS and interpretation of conditionals. To that end we define:

$$M = \{\texttt{addAB}, \texttt{semantic}, \texttt{wc}\}$$

The external evaluation function should translate the SCP into a decision that can be compared with empirical data. In the Suppression Task we will treat $f_{\text{sup}}()$ as a function that makes a single choice from the set {'she will strudy late in the library', she will not study late in the library,

Under the assumption only the final state point p_n determines whether she will study late, we might define $f_{\text{sup}}()$ as follows:

Algorithm 14: $\texttt{f}_{\text{sup}}$

Function $f_{sup}(\pi)$ **is**
  `/* This is a final state evaluation SCP. */`
  Let $R = [(K[\pi], f())]$ be the set of realised SCPs $r_i = (k, f())$ which can be generated from π;
  Let $P = [\bar{p}_1, ..., \bar{p}_n]$, where $\bar{p}_i$ is the final state of r_i;
  $P_{\text{el}} := [\bar{p} \in P | \bar{p}[\text{'name'}] = \text{'el'}$;
  $P_{\text{elo}} := [\bar{p} \in P | \bar{p}[\text{'name'}] = \text{'elo'}$;
  if *there is a response* $\in$ $\texttt{responses}(P_{el})$ *which is not* $\in$ $\texttt{responses}(P_{elo})$ **then**
    return'suppression observed'
  else
    return'no suppression'
  end
end
Function $\texttt{responses}(p)$ **is**
  return $[\texttt{response}(\bar{p}) | \bar{p} \in p]$
end
Function $\texttt{response}(\bar{p})$ **is**
  if $I_{\bar{p}(V)}(l) \models \top$ **then**
    return 'she will study late in the library'
  else if $I_{\bar{p}(V)}(l) \models \bot$ **then**
  else
    return 'we are uncertain if she will study late'
  end
end

It is worth noting that one could also define f_{sup} in such a way that it iterates over every variable assignment for 'el' and 'elo' in the final state point and sees if any inference is drawn in 'el' that is not drawn in 'elo'. However, the purpose of the external evaluation function is to transform an SCP into an output from the set of empirically observed outputs, and in this case the empirical knowledge is a response to the question "will she study late in the library?".

Next, we define the empirical result we wish to emulate γ. In this case we will define $\gamma_{textnoSup}$ to model cases where no suppression is observed, and γ_{sup} to model cases where suppression is observed.

$$\gamma_{\text{noSup}} = \text{no suppression}$$

$$\gamma_{\text{sup}} = \text{suppression observed}$$

Armed with these definitions, it is possible to model the task Π as follows:

$$\Pi = (s_i, M, f(), \gamma_{\text{sup}})$$

De Novo search on Π returns the satisfying SCP μ_{sup}:

$$\mu_{\text{sup}} = (\pi = (s_i \longmapsto \texttt{addAB} \longmapsto \texttt{wc} \longmapsto \texttt{semantic}), f())$$

Figure 6.1 illustrates the realised SCP for μ_{sup} and illustrates that $\mu_{\text{noSup}} \models \gamma$. This example has served to show that suppression can modelled in the SCP framework.

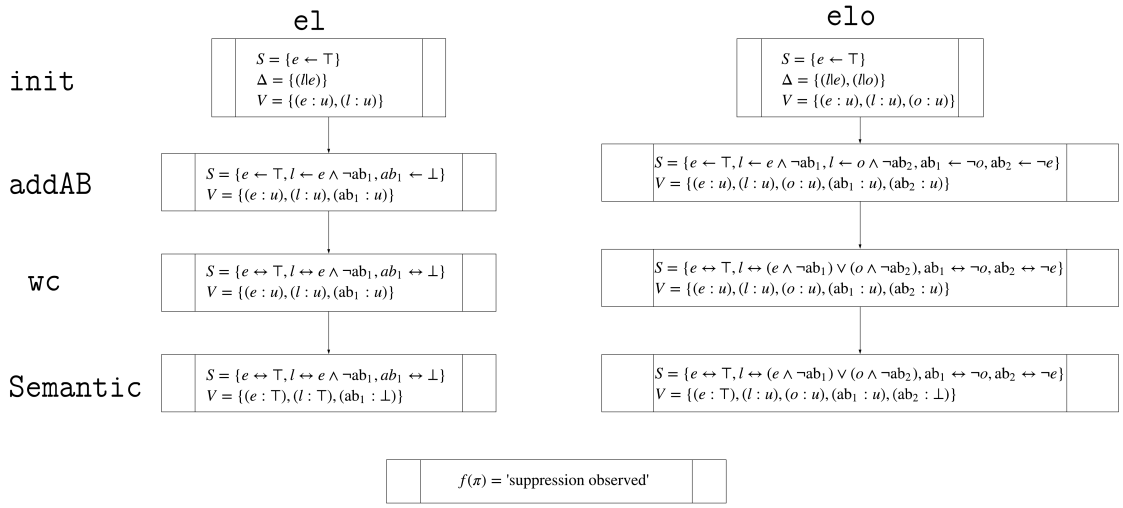


FIGURE 6.1: Suppression for the SCP $\mu_{\text{sup}} = (\pi = (s_i \longmapsto \texttt{addAB} \longmapsto \texttt{wc} \longmapsto \texttt{semantic}), f())$

.

To illustrate the ability of the framework to model individual reasoners, we examine those reasoners who despite knowing both that if she has an essay to write she will study late in the library and that if the library is open she will study late in the library still do not demonstrate suppression, and instead draw the classical conclusion that she will study late in the library. To this end we define a second SCP Task Π_{deviant}.

$$\Pi_{\text{deviant}} = (s_i^{\text{sup}}, M', f(), \gamma_{\text{noSup}})$$
$$M' = \{\texttt{addAB}, \texttt{semantic}, \texttt{wc}, \texttt{remove}_{\perp}, \texttt{addExp}\}$$

M' is contains $\texttt{remove}_{\perp}$ which is able to delete any subset of the variables mentioned in $R[\text{'delete'}]$ of the epistemic state. `addExp` allows us to add abducible facts to an epistemic state, following the intuition we used to model the WST (@TODORef). To see the explanations and justifications for these two cognitive operations, refer to Section 5.4.5, and Section 5.4.10.

Equipped with these two cognitive operations, there are several SCPs which can model Π_{deviant}.

Figure 6.2 shows the realised SCPs for one satisfying SCP for Π_{deviant}. This example shows that deleting the variable o from the epistemic state, prevents suppression from occurring because suppression the abnormality $(\text{ab}_1 \leftarrow \neg o)$ is the cause of the suppression in Π, and in Π' the abnormality is now given by $(\text{ab}_1 \leftarrow \neg\top)$.

Appendix @TODOref shows that the non-monotonic cognitive operation `addExp` can prevent suppression by adding abducibles related to the free variable o.[1] This is, as far as I can tell, the first use of @TODOs abducible approach to the WST to model the Suppression Task. By giving o an assignment, the abnormality $\text{ab}_1 \leftrightarrow \neg o$ no longer evaluates to u, and across all responses epistemic states, the 'elo' states now model every response `response` that 'el' does.

Both of these examples illustrate the derivation of the classical logical response to the suppression task as a modification of an SCP which did illustrate suppression. This evidences one of the core claims of the SCP Framework which is that it is a reasonable tool for

[1]The output is too large to graph attractively, and should be viewed in its entirety to confirm that this approach is effective.

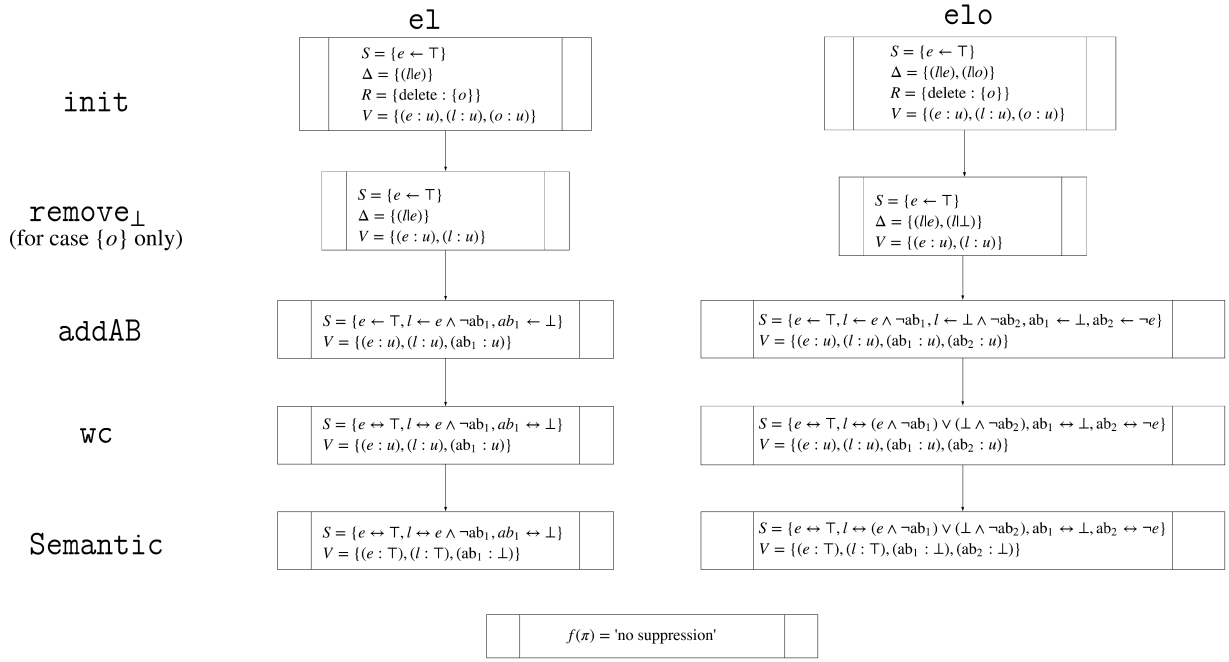


FIGURE 6.2: Inhibition of Suppression for the SCP $\mu_del = (\pi, f_{\text{sup}}()), \pi = (s_i \longmapsto \texttt{remove}_\perp \longmapsto \texttt{addAB} \longmapsto \texttt{wc} \longmapsto \texttt{semantic})$.

.

modelling individual reasoners in a cognitive task.

The value of these examples lies in that they show that classical logic response to a cognitive task is not necessarily derived through the use of classical logic.

6.3 Wason Selection Task

We define the SCP Task $\Pi = (s_i, M, f(), \gamma)$ to model the general case of the WST. The choice of external evaluation function for this task (the turn function), if we are to follow the intuition from Section 3.1.2, has two requirements: it must be able to capture whether an observation O can be explained by the least model of the weakly completed program, and it must ensure that variable assignments which could falsify or verify the conditional rule $(3|D)$, are present in the least model.

To this end, we add a set of abducibles to the categorization variable R, and combine this with the variables for the set of propositional rules S, conditional rules Δ, and possible world V to create the initial state point s_i.

Next, we require an intuition for the contents of s_i. In this case, abduction can be simulated by the SCP in one of three ways. The first is to create a unique SCP for each possible explanation for each observation as we did for the classical WCS interpretation of the task in Section 3.1.2. But the fact that SCPs may contain an initial state point, rather than just a single epistemic state, enables a more elegant solution that captures the full scope of possible observation, explanation pairings. The other two options are to start with a state point containing each abductive case, or to create cognitive operations which add the abducible cases at computation time. To this end we define:

$$s_i = (S, \Delta, R)$$

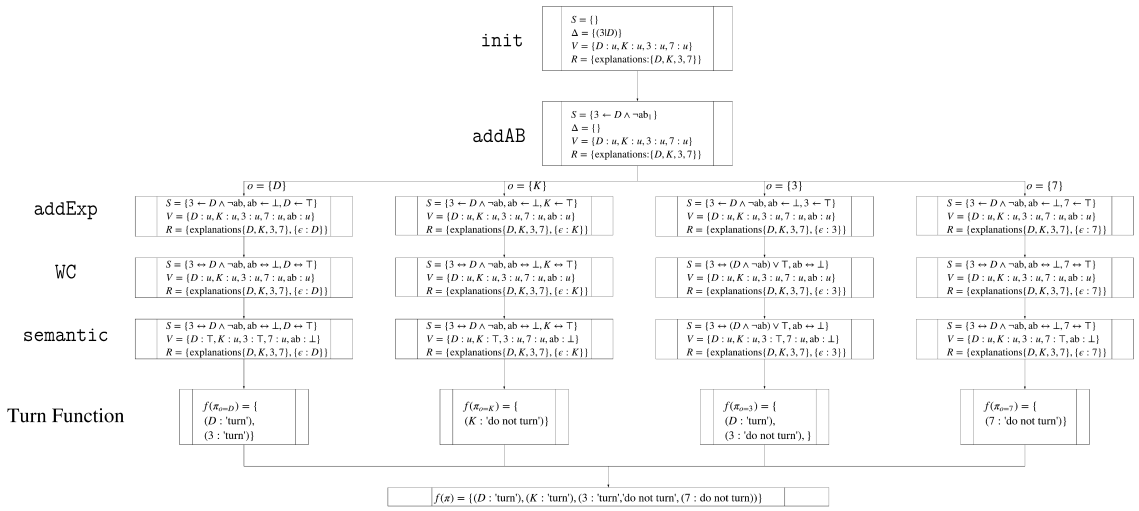


FIGURE 6.3: Realised SCPs for the SCP interpretation of the WST $\pi_1 = (s_i \longmapsto$ `addAB` $\longmapsto$ `addExp` $\longmapsto$ `wc` $\longmapsto$ `semantic`$))$. Results mimic those observed in @TODOref's WST interpretation.

$$S = \{\}$$

$$\Delta = \{(3|D)\}$$

$$R = \{\text{explanations} : \{D \leftarrow \top, D \leftarrow \bot, K \leftarrow \top, K \leftarrow \bot, 3 \leftarrow \top, 3 \leftarrow \bot, 7 \leftarrow \top, 7 \leftarrow \bot\}\}$$

Because SCPs aim to capture as much cognitive information as possible within the CTM as possible, we will prefer this third case and make use of the categorization variable R to specify both the set of observations, and the possible explanations. The cognitive operation `AddExp` is created.

For the purposes of readability, we will restrict examination of the abducibles to cases that are of length 1, and contain only positive facts[2].

[2] A reader wishing to see this process with respect to the full set of abducibles should examine the `WST.py` file in the SCP implementation.

Next, we define the external evaluation function $f_{\text{wcs}}(\pi)$ as follows:

Algorithm 15: $\texttt{f}_{\text{WST}}$

Function $f_{WST}(\pi)$ **is**
  `/* This is a final state evaluation SCP. */`
  Let $R = [(K[\pi], f())]$ be the set of realised SCPs $r_i = (k, f())$ which can be generated from π;
  Let $P = [p_1, ..., p_n]$, where p_i is the final state of r_i;
  Let $\eta_i = [\eta_1, ..., \eta_n]$, where $\eta_i = p_i[R][\eta]$;
  Let $O = \{(D, \top), (K, \top), (3, \top), (7, \top)\}$ be the set of observations;
  Let $C = \{(3|D), (D'|7)\}$ be the set of conditionals which must be falsified or verified.;
  Output={};
  for $p_i \in P$ **do**
    run `isExplanation`(p_i);
  end
  for *each minimal explanation e for observation o* **do**
    if $I_{e[\prime V\prime]}(C)$ *either falsifies or verifies some conditional in C* **then**
      Output[o]:=Output[o]∪ 'Turn Card';
    else
      Output[o]:=Output[o]∪ 'Do not turn Card';
    end
  end
  return Output
end
Function *isExplanation*(p_i) **is**
  if $I_{p_i[\prime V\prime]}(p_i[\prime S\prime]) \models (o \in O)$ **then**
    `/* Then` p_i `explains` o. `*/`
    Add p_i to the explanations for o.
  end
end

Next we create our goals. We will make one goal for each of the canonical cases of the WST. The structure of the goal associates a turn function with every possible observation that needs to be explained.

$$\gamma_{D,3} = \{(\text{D : turn card}), (3 : \text{turn card}), (\text{K : turn card}), (7 : \text{do not turn card})\}$$

$$\gamma_{D} = \{(\text{D : turn card}), (3 : \text{do not turn card}), (\text{K : turn card}), (7 : \text{do not turn card})\}$$

$$\gamma_{D,7} = \{(\text{D : turn card}), (3 : \text{do not turn card}), (\text{K : turn card}), (7 : \text{turn card})\}$$

$$\gamma_{D,3,7} = \{(\text{D : turn card}), (3 : \text{turn card}), (\text{K : turn card}), (7 : \text{turn card})\}$$

Next, we define the set of allowable cognitive operations M. In this case, because our previous modelling of the general case of the WCS required the addition of explanations from the set of abducibles, we will include the operation `addExp` (as seen in the SCP Modelling of individual cases of the Suppression Task).

$$M = \{\texttt{addAB}, \texttt{semantic}, \texttt{wc}, \texttt{AddExp}\}$$

The overall planning task, Π, can now be formulated for the general case of the WST.

$$\Pi = (s_i, M, f_{\text{WST}}(), \gamma_{D,3})$$

De Novo search for the SCP Plan yields several SCP (differentiated by where the `addExp` cognitive operation occurs), including this one:

$$\mu_{D,3} = (\pi = (s_i \longmapsto \texttt{addAB} \longmapsto \texttt{addExp} \longmapsto \texttt{wc} \longmapsto \texttt{semantic}), f())\}$$

@TODO put in pArray

$f_{\text{sup}}(\pi) = \{(\text{D} : \{\text{turn card, do not turn}\}, (3 : \{\text{turn card, do not turn}\}, (\text{K} : \{\text{do not turn}\}, (7 : \{\text{do not t}$

Figure **??** demonstrates the calculation of $f(\pi)$ for each positive explanation η of length 1. When this task is run for minimal explanations of unrestricted length, the set of conclusions for each card returned by $f_{\text{WST}}()$ do not change. We find that $\mu_{D,3} \models_{\text{weak}} \gamma_{D,3}$ and that we should turn the cards D, and 3, precisely as in Dietz, Hölldobler, and Wernhard, 2014's model.

Thus, it has been shown that the SCP Framework is capable of modelling the general response to the abstract case of the WST.

In order to model the goals γ_D, $\gamma_{D,7}$, and $\gamma_{D,3,7}$, we must again change our SCP Task. These extensions are easily introduced and, the author hopes, will serve as motivation of the practical and intuitive usefulness of the SCP Framework.

The simplest of these case to model is γ_D. This could done by creating the new SCP Task Π_D.

$$\Pi_D = (s_i, M, \gamma_D, f_{\text{WST, simple}})$$

And defining $f_{\text{WST, simple}}$ precisely as f_{WST} was defined, except that the `isExplanation` function in Algorithm 16 is changed to that of Algorithm 17.

Algorithm 16: `isExplanation` function as it appears in $\texttt{f}_{\text{WST}}$.

Function *isExplanation*(p_i) **is**
  if $I_{p_i[{}'V']}(p_i[{}'S']) \models (o \in O)$ **then**
    /* Then p_i explains o. */
    Add p_i to the explanations for o.
  end
end

Algorithm 17: `isExplanation` function as it appears in $\texttt{f}_{\text{WST, simple}}$.

Function *isExplanation*(p_i) **is**
  /* We no require that the observation being explained is an element of the proposed explanations. */
  if $I_{p_i[{}'V']}(p_i[{}'S']) \models (o \in O)$ ***AND*** $o \in p_i[R][\epsilon]$ **then**
    /* Then p_i explains o. */
    Add p_i to the explanations for o.
  end
end

This approach mimics the logic we used to create the Simple WCS Extension in Section @TODOref. However, there is an alternative approach which I prefer.

One should note that, while $\mu_{D,3} \models_{\text{weak}} \gamma_{D,3}$, it is also the case that that 'do not turn' $\in f_{\text{sup}}(\pi)[K]$, $f_{\text{sup}}(\pi)[D]$, and $f_{\text{sup}}(\pi)[7]$, but is not in $f_{\text{sup}}(\pi)[D]$. If we treat $f_{\text{sup}}(\pi)$ sceptically

and assume that we only turn a card if there is *no reason not to turn it*, we arrive at the conclusion that only D should be turned.

We define a set of preferred responses which prefers the 'do not turn' response as follows (using prefered responses as defined in Section 4.6):

$$\text{pred} = \{(\text{D} : \text{do not turn}), (3 : \text{do not turn}), (\text{K} : \text{do not turn}), (7 : \text{do not turn})\}$$

$$p(f_{\text{WST}}(\pi), \text{pred}) = \{(\text{D} : \text{turn card}), (3 : \text{do not turn}), (\text{K} : \text{do not turn}), (7 : \text{do not turn})\}$$

Now it is clear that $p(f_{\text{WST}}(\pi), \text{pred}) \models \gamma_D$. To return this to an acceptable SCP format, we may simply define $f^{\text{pref}}_{\text{WST}} = p(f_{\text{WST}}(\pi)$ and observe that $f^{\text{pref}}_{\text{WST}} \models \gamma_D$ for SCP $\mu_D = (\pi, f^{\text{pref}}_{\text{WST}}())$. Implementing contraposition in the SCP framework we have defined is suprisingly simple. We simply need to define a new plan $\Pi_{\text{D},3,7}$ as follows:

$$\Pi_{\text{D},3,7} = (s_i', M, \gamma_{D,3,7}, f_{\text{WST}}())$$

$$s_i' = (S', \Delta', R)$$

$$S' = \{(D \leftarrow \neg D')\}$$

$$\Delta' = \{(3|D), (D'|7)\}$$

Simply by adding the contrapositive conditional and its variable initialization, search returns an SCP $\mu_{D,3,7} = (\pi', f_{\text{WST}}())$, with $f_{\text{WST}}(\pi') \models_{\text{weak}} \gamma_{D,3,7}$. Figure @TODO illustrates this fact for positive explanations of length 1.

Finally, the case where the D and 7 cards are turned (the classical inference), can be modelled by combining aspects of the SCPs above. We define the SCP Plan $\Pi_{\text{D},7}$:

$$\Pi_{\text{D},7} = (s_i', M, f^{\text{pref}}_{\text{WST}}(), \gamma_{D,7})$$

An running De Novo search over this Task reveals that the SCP $\mu_{D,7} = (\pi, f^{\text{pref}}_{\text{WST}}())$ is a satisfying SCP for the goal $\gamma_{D,7}$. Figure @TODO illustrates this point.

Thus, it has been shown that two CTMs π, and π', in conjunction with the external activation functions $f_{\text{WST}}()$ and $f^{\text{pref}}_{\text{WST}}()$ are able to model individual reasoners in the Wason Selection Task.

6.4 Default Logic

@TODO

Chapter 7

Comparing SCPs

7.1 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 approaches produce the desired output. Consider a toy example of an SCP task which describes the mental process needed to bake a cake:

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

$$s_0 = (V = (cakeBaked : \bot))$$

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

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

$$\gamma = (f(\pi) = True)$$

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 $f(s_0 \longmapsto \texttt{mixIngredients})$ 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:

$$\mu_1 = (\pi_1 = s_0 \longmapsto \texttt{mixIngredients} \longmapsto \texttt{bakeIngredients}, f()) \tag{7.1}$$

$$\mu_2 = (\pi_2 = s_0 \longmapsto \texttt{mixIngredients} \longmapsto \texttt{doTheLaundry} \longmapsto \texttt{bakeIngredients}, f()) \tag{7.2}$$

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 it restricted, but this seems implausible. Most readers would agree that Equation 7.1 is more plausible that Equation 7.2.

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 7.2 discusses the question of how to compare different SCPs found using search for a single task. Section 7.3 discuses the more general problem of comparing SCPs even when the underlying SCPs tasks differ. Section 7.3.1 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 **??** discusses and justifies and extension to this algorithm for use in SCP comparisons.

7.2 Comparing Generated SCPs

7.2.1 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 (Sweetser, 1990) (Needleman and Wunsch, 1970) , 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 7.1 we will attempt to create a commonsense scoring mechanism to determine whether μ_1 or μ_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. The same logic can be applied to SCPs for those SCPs generated using either *De Novo* or *Insertion Search* (Section 4.7).

If we define the length $|\mu|$ of an SCP $\mu = \{\pi, f()\}$ to be number of cognitive operations present in π. In the case of *De Novo* search, we assume assume that the optimal length of a solution to Π is an SCP $\mu^* = (\pi^*, f())$ with length $|\mu^*| = 0$. 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: $|\mu_1| < |\mu_2|$, therefore μ_1 is preferred because it requires only 2 operations to transform the ideal SCP $f(s_0)$ into $f(s_0 \longmapsto \texttt{mixIngredients} \longmapsto \texttt{bakeIngredients})$, rather than the 3 required for mu_2.

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

Limitations

Unfortunately, the SCP Length approach fails to capture anything about the *relationship* between the SCPs being scored. Let us examine the following three SCPs, all generated whilst modelling the Wason Selection Task:

$$\pi_{\text{WST}} = (s_{WST} \longmapsto \texttt{addAB} \longmapsto \texttt{addExp} \longmapsto \texttt{wc} \longmapsto \texttt{semantic})$$

$$\pi_{\text{prop}} = (s_{WST} \longmapsto \texttt{th})$$

$$\mu_{D,3} = (\pi_{WST}, f_{\text{WST}}())$$

$$\mu_{D,7} = (\pi_{\text{WST}}, f^{\text{pref}}_{\text{WST}}())$$

$$\mu'_{D,7} = (\pi_{\text{prop}}, f^{\text{pref}}_{\text{WST}}())$$

Using SCP Length, $|\mu_{D,7}| = 4$ and $|\mu'_{D,7}| = 1$ and we would would conclude that $|\mu'_{D,7}| = 1$ is the more probable SCP.

In isolation, given only $|\mu_{D,7}|$ and $\mu'_{D,7}$, this conclusion makes some sense. $\mu'_{D,7}$ is indeed significantly shorter, and can be achieved with a single cognitive operation. But our information about these SCPs comes with more context than that. Assume that we are certain that $\mu_{D,3}$ is an accurate model for some other related cognitive task. Now it seems plausible that the solution to a related cognitive task may be an offshoot of an underlying core process. From this perspective, if we belief that $\mu_{D,3}$ has an underlying model in common (save for some 'mutations' which add or remove cognitive operations) with the solution to the task to turn the D and 3 cards, $\mu_{D,7}$ seems significantly more likely than $\mu'_{D,7}$.

7.3 Comparing External SCPs

One of the core principles of biological scoring algorithms is the intuition that the genetic structures being analysed share a common ancestor at some point in the post. With this assumption it becomes possible to model precisely what genetic mutations one sequence might have undergone to become the other.

This intuition carries surprisingly well into the field of cognitive modelling where researchers very often search for common core mechanisms shared between different reasoners, or across different cognitive tasks. Concepts like the Diversity Principle(Heit, Hahn, and Feeney, 2005), and denial of the antecedent are often shown be powerful predictors of participant responses.

To this end, I propose that the concept of a shared underlying cognitive process in cognitive modelling can be considered analogous to the concept of shared ancestry in genetic sequences. If we assume that the SCP Framework is a valid mechanism for human cognition, it becomes possible to use these biological principle to compare and score SCPs.

7.3.1 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 the genetic sequence of one of those animals into the other. Needleman and Wunsch, 1970 proposed a simple scoring algorithm to determine how similar two genetic sequences were to one another by computing how many insertions, deletions, and mutations the first sequence would have to undergo to become the second sequence, and assigned a quantitative penalty or reward to each of these events.

Event	Process	Score
$x = y$	Match	1
$x \neq y$, $x \neq$'-', $y \neq$'-'	Mismatch	-1
$x =$'-',	Insert	-1
$y =$'-'	Delete	-1

TABLE 7.1: A very simple table for scoring SCPs with the Needleman-Wunsch algorithm for inputs x and y.

		A	T	T	A	C	A
	0	-1	-2	-3	-4	-5	-6
A	-1	1	0	-1	-2	-3	-4
T	-2	0	2	1	0	-1	-2
G	-3	-1	1	1	0	-1	-2
C	-4	-2	0	0	0	1	0
T	-5	-3	-1	1	0	0	0

TABLE 7.2: Scoring matrix for the sequences $a = ATTACA$ and $b = ATGCT$ using the scoring properties defined in Table 7.1.

Needleman's recursion makes use of a dynamic programming approach and *score matrix* D to score two sequences efficiently ($O(nm)$ time and $O(nm)$ space complexity for sequences of length n and m, respectively). The recursion for his algorithm for sequences a and b is as follows:

$$D_{0,j} = \text{insertion cost} \times j$$

$$D_{i,0} = \text{deletion cost} \times i$$

$$D_{i,j} = \min \begin{pmatrix} D_{i-1,j-1} & + & s(a_i, b_j) \\ D_{i-1,j} & + & s(a_i, -) \\ D_{i,j-1} & + & s(-, b_j) \end{pmatrix}$$

Where an example of the costs $s(x, y)$ is given by Table 7.1. Matches are usually positively scored (desirable), and mismatches, insertions, and deletions, are usually negatively scored (undesirable).

Needleman also defined a traceback algorithm to show what this optimal global alignment was. The traceback begins at the bottom right corner of the D matrix and for, position $D_{i,j}$, calculates which neighbouring value $D_{i-1,j-1}$, $D_{i-1,j}$, $D_{i,j-1}$ would result in $D_{i,j}$ when added to the cost of match/mismatch, insertion and deletion respectively. We will not detail this aspect of the algorithm beyond this point because our interest lies in the scoring itself, rather than the traceback.

Table 7.2 shows the scoring matrix for the sequences $a = ATTACA$ and $b = ATGCT$, and the optimal alignment is given by:

$$\begin{pmatrix} A & T & T & A & C & A \\ A & - & T & G & C & T \end{pmatrix}$$

Where $-$ denotes an insertion or deletion.

		s_{WST}	addAB	addExp	wc	semantic
	0	-1	-2	-3	-4	-5
s_{WST}	-1	1	0	-1	-2	-3
addAB	-2	0	2	1	0	-1
addExp	-3	-1	1	3	2	1
wc	-4	-2	0	2	4	3
semantic	-5	-3	-1	1	3	5

TABLE 7.3: Needleman-Wunsch comparison of $\mu_{D,3}$ to $\mu_{D,7}$

		s_{WST}	addAB	addExp	wc	semantic
	0	-1	-2	-3	-4	-5
s_{WST}	-1	1	0	-1	-2	-3
th	-2	0	-1	-2	-3	-4

TABLE 7.4: Needleman-Wunsch comparison of $\mu_{D,3}$ to $\mu'_{D,7}$

Needleman-Wunsch for SCPs

The initial extension of the Needleman-Wunsch algorithm to the SCP framework is surprisingly simple. Because known SCPs are guaranteed to consist of a linear, finite sequence of cognitive operations, and comparison between cognitive operations is possible (two cognitive operations are equal if they have the same name), the only thing the SCP framework lacks is a way to represent insertions and deletions. To this end we define the `insert` cognitive operation.

Algorithm 18: `insert`$(\bar{p})$: an empty cognitive operation used as a place-holder for scoring.

Function *e*$(\bar{p})$ **is**
  return $\bar{p}$
end

Like the `th` operation, `insert` has no function in an SCP Task, and is not assumed to be a function with an analogue in human reasoning. Instead it is a placeholder function that represents some unknown function in the shared cognitive process from which both SCPs are assumed to originate.

Finally, we must decide how the initial epistemic state is to be scored. For now we will simply treat the s_i as if it were a cognitive operation for scoring purposes.

Now we can compare out candidate solution SCPs $\mu_{D,7}$ and $\mu'_{D,7}$ using the score system from Table 7.1 and the Needleman-Wunsch Algorithm. Table 7.4 shows global alignment score of -4 for ($\mu_{D,3}$,$\mu'_{D,7}$), whereas Table 7.3 has a global alignment score of 5 for ($\mu_{D,3}$,$\mu_{D,7}$) and so $\mu_{D,7}$ is believed to be more similar to $\mu_{D,3}$.

$$\text{align}(\mu_{D,3}, \mu'_{D,7}) = \begin{pmatrix} s_{WST} & \texttt{addAB} & \texttt{addExp} & \texttt{wc} & \texttt{semantic} \\ s_{WST} & \texttt{addAB} & \texttt{addExp} & \texttt{wc} & \texttt{semantic} \end{pmatrix}$$

$$\text{align}(\mu_{D,3}, \mu_{D,7}) = \begin{pmatrix} s_{WST} & \texttt{addAB} & \texttt{addExp} & \texttt{wc} & \texttt{semantic} \\ s_{WST} & \texttt{insert} & \texttt{insert} & \texttt{insert} & \texttt{th} \end{pmatrix}$$

It is my belief that this technique offers a great deal of insight into which cognitive operations are to be prefered, and, over a sufficiently large number of well-founded SCPs for different

cognitive tasks, may help to shape our future theories about what general processing motifs really inform certain aspects of human cognition.

Extending Needleman-Wunsch for SCPs

We have now seen that SCP similarity can be computed using the Needleman-Wunsch algorithm. `SCP_ scoring_ improved` in the implementation of the SCP framework takes this approach one step further and assigns unique scores to matches, mismatches, and insertion operations for every cognitive operation. This allows the researcher to specify intuition about the relative complexity of each cognitive operation.

I will not detail this approach further in this thesis because I do not currently have an intuition as to how these scores should be calculated for each operation, but it is worth noting that the implementation exists.

7.3.2 Limitations of the Needleman-Wunsch Algorithm

The Needleman-Wunsch algorithm is generally considered a good introductory algorithm to those learning bioinformatics, but it lacks many of the useful features of more advanced algorithms in the field. In a similar way, it was used here to show the suitability of similarity scoring algorithms to SCPs, logical extensions to more advanced string-matching algorithms are easily imagined.

A significant limitation of the Needleman-Wunsch algorithm, and by extensions our algorithm, is the inability to deal with crossovers. That is, the algorithm cannot handle cases where operators or groups of operators are shifted into a new position (imagine cutting the middle out of a piece of string and retying it to the end of the string). This limitation is imposed by many such algorithms to limit the computational complexity of the computation, without this limitation string matching becomes NP-hard.

The Smith-Waterman algorithm (Smith, Waterman, et al., 1981) is a simple variation on the Needleman-Wunsch algorithm and is used to find optimal local alignments. It could be used to find subsequences of complex actions that are preserved between different SCPs. These subsequences (for example the pCTM ($\texttt{addAB} \longmapsto \texttt{wc}$) may be shown to be present in comparisons of models of different cognitive tasks, and may be considered to show support for common motifs in general human cognition.

Chapter 8

Program Design

8.1 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 github.com/AxelInd/SCP_ Implementation.

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.

8.2 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 **??** provides a simplified class diagram to explain how the major components of the system relate to one another.

8.2.1 Most Important Modules

- `SCPFramework/basicLogic`: an implementation of basic two and three-valued logic.
- `SCPFramework/SCP_Task`: implements both the SCP Task (Π) object and a search function for generating SCPs which meet the given goals.
- `SCPFramework/CTM`: implements the CTM π and defines the $J[p, m]$ application of a cognitive operation to an input state point
- `SCPFramework/CognitiveOperations`: defines the set of cognitive operations as well as the their input and output parameters.
- `SCPFramework/StatePointOperations`: defines $f(\pi) \models_{\text{strict}} \gamma$ and $f(\pi) \models_{\text{weak}} \gamma$ determining if a given SCP meets a given set of output criteria.
- `SCPFramework/truthTables`: a set of truth tables for each operator defined in `basicLogic`.
- `ScoringAlgorithms/SCP_scoring`: an implementation of the Needleman-Wunsch Algorithm for SCPs described in Chapter 7
- `ScoringAlgorithms/SCP_scoring_improved`: an implementation of the extended Needleman Wunsch Algorithm for SCPs described in Chapter 7

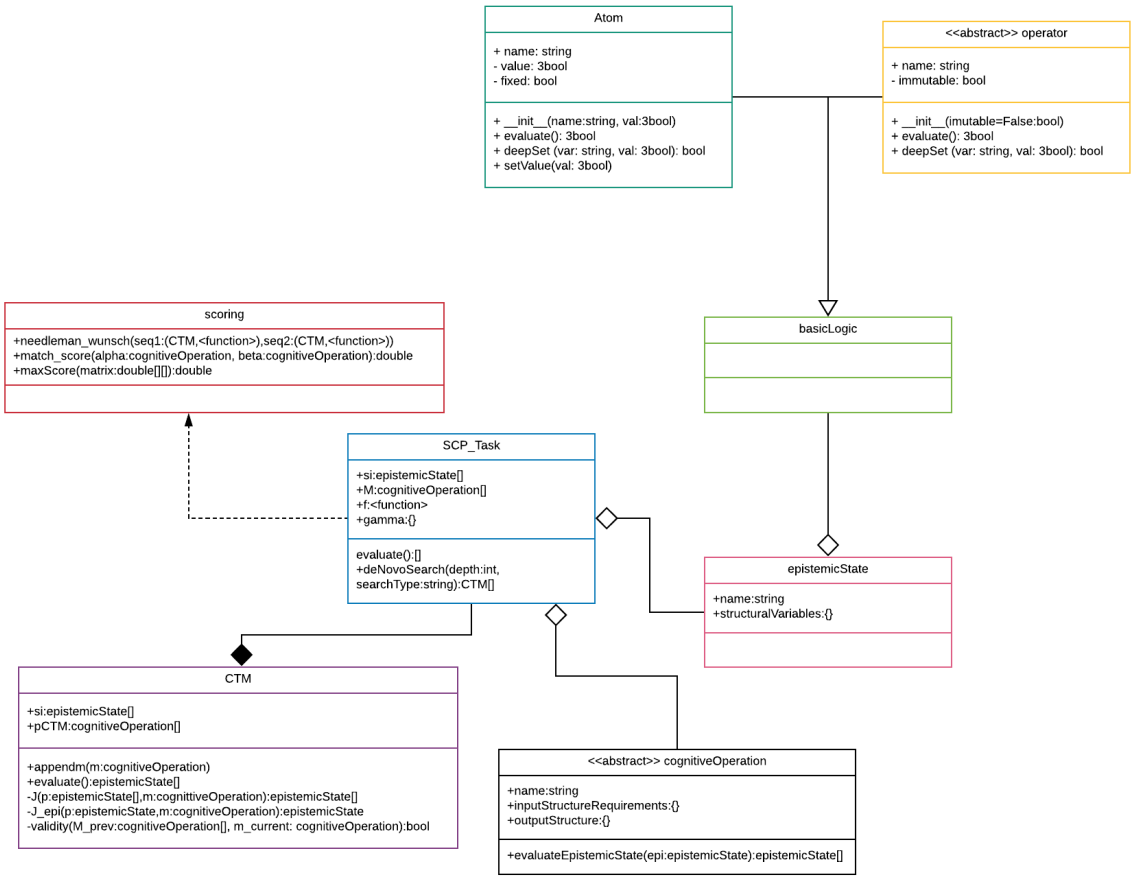


FIGURE 8.1: Class diagram for the implementation of the SCP Framework.

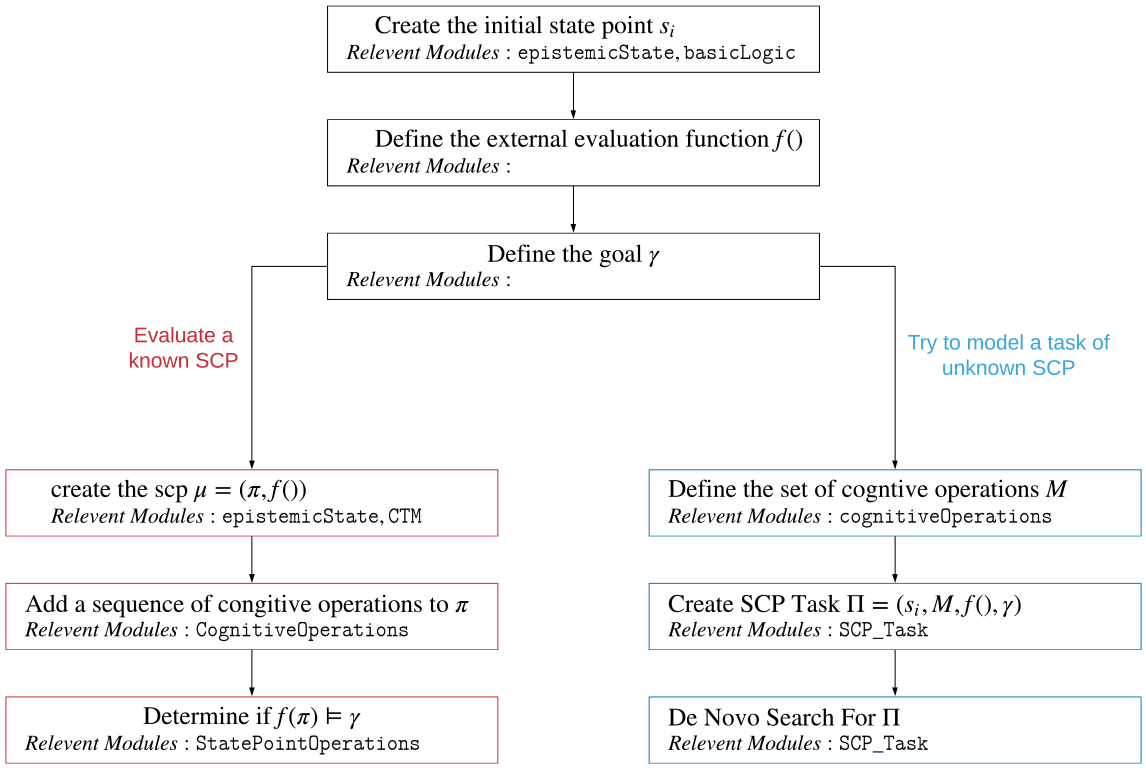


FIGURE 8.2: Two use cases for the implementation of the SCP framework

8.2.2 Program Flow

Diagram 8.2 illustrates the basic programmatic flow used to model either a known SCP or known empirical result set using the SCP Framework. In the appendix, Figure 3 provides a code snippet to show exactly how the general case of the Suppression Task is modelled in the SCP framework.

8.2.3 Modelled Experiments

- `example_wcs_suppressionTask.py`: models the Suppression Task as discussed in Chapter 6.
- `example_wcs_wst.py`: models the Wason Selection Task as discussed in Chapter 6.
- `example_default_various`: models of the Suppression Task, Penguin Case, and Nixon Diamond implemented with a default-logic capable SCP.

8.3 External Framework Integration

At present work is ongoing to integrate the SCP framework with `CCOBRA` (Nicolas, 2018).

Chapter 9

Conclusions and Future Work

9.1 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.

9.2 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.

Bibliography

Baratgin, Jean, David Over, and Guy Politzer (2014). "New psychological paradigm for conditionals and general de Finetti tables". In: *Mind & Language* 29.1, pp. 73–84.

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.

De Champeaux, Dennis (1983). "Bidirectional heuristic search again". In: *Journal of the ACM (JACM)* 30.1, pp. 22–32.

Dietz, Emmanuelle-Anna, Steffan Holldobler, 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.

Drummond, Chris (2002). "Accelerating reinforcement learning by composing solutions of automatically identified subtasks". In: *Journal of Artificial Intelligence Research* 16, pp. 59–104.

Gonçalves, Bruno, Nicola Perra, and Alessandro Vespignani (2011). "Modeling users' activity on twitter networks: Validation of dunbar's number". In: *PloS one* 6.8.

Heit, Evan, Ulrike Hahn, and Aidan Feeney (2005). "Defending diversity." In:

Hölldobler, Steffen (2015). "Weak Completion Semantics and its Applications in Human Reasoning." In: *Bridging@ CADE*, pp. 2–16.

Korf, Richard E (1987). "Planning as search: A quantitative approach". In: *Artificial intelligence* 33.1, pp. 65–88.

— (1996). *Artificial intelligence search algorithms*. Citeseer.

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.

Needleman, Saul B and Christian D Wunsch (1970). "A general method applicable to the search for similarities in the amino acid sequence of two proteins". In: *Journal of molecular biology* 48.3, pp. 443–453.

Nicolas, Riesterer (Aug. 2018). *Framework for the specification and analysis of models for human reasoning.* Version 0.0.14. URL: https://github.com/CognitiveComputationLab/ccobra.

Ragni, Marco, Ilir Kola, and Phil Johnson-Laird (2017). "The Wason Selection task: A Meta-Analysis." In: *CogSci*.

Ragni, Marco et al. (2016). "Two-Valued Logic is Not Sufficient to Model Human Reasoning, but Three-Valued Logic is: A Formal Analysis." In: *Bridging@ IJCAI*, pp. 61–73.

Ragni, Marco et al. (2017). "Formal nonmonotonic theories and properties of human defeasible reasoning". In: *Minds and Machines* 27.1, pp. 79–117.

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.

Schaefer, Thomas J (1978). "The complexity of satisfiability problems". In: *Proceedings of the tenth annual ACM symposium on Theory of computing*, pp. 216–226.

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.

Smith, Temple F, Michael S Waterman, et al. (1981). "Identification of common molecular subsequences". In: *Journal of molecular biology* 147.1, pp. 195–197.

Stenning, Keith and Michiel Van Lambalgen (2012). *Human reasoning and cognitive science*. MIT Press.

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.

Sweetser, Eve (1990). *From etymology to pragmatics: Metaphorical and cultural aspects of semantic structure*. Vol. 54. Cambridge University Press.

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.

Zhou, Rong and Eric A Hansen (2006). "Breadth-first heuristic search". In: *Artificial Intelligence* 170.4-5, pp. 385–408.

Appendix

.1 Proofs

.2 Output

.3 Code Snippets

.3.1 Program Flow

```
==================ABDUCIBLE SUPPRESSION=====================
Epistemic state:

===>el<===
S:: [(e ↔ ⊤), (l ↔ (e ∧ (¬ ab_1))), (ab_1 ↔ ⊥)]
Delta:: []
V:: [(e:True), (l:True), (ab_1:False)]
R:: {'abducibles': []}

response(p_bar): She will study late in the library
------------------------------
Epistemic state:

===>el<===
S:: [(e ↔ ⊤), (l ↔ (e ∧ (¬ ab_1))), (ab_1 ↔ ⊥), (o ↔ ⊤)]
Delta:: []
V:: [(e:True), (l:True), (ab_1:False)]
R:: {'abducibles': [(o ← ⊤)]}

response(p_bar): She will study late in the library
------------------------------
Epistemic state:

===>el<===
S:: [(e ↔ ⊤), (l ↔ (e ∧ (¬ ab_1))), (ab_1 ↔ ⊥), (o ↔ ⊥)]
Delta:: []
V:: [(e:True), (l:True), (ab_1:False)]
R:: {'abducibles': [(o ← ⊥)]}

response(p_bar): She will study late in the library
------------------------------
Epistemic state:

===>el<===
S:: [(e ↔ ⊤), (l ↔ (e ∧ (¬ ab_1))), (ab_1 ↔ ⊥), (o ↔ (⊤ ∨ ⊥))]
Delta:: []
V:: [(e:True), (l:True), (ab_1:False)]
R:: {'abducibles': [(o ← ⊤), (o ← ⊥)]}

response(p_bar): She will study late in the library
------------------------------
Epistemic state:

===>elo<===
S:: [(e ↔ ⊤), (l ↔ ((e ∧ (¬ ab_1)) ∨ (o ∧ (¬ ab_2)))), (ab_1 ↔ (¬ o)), (ab_2 ↔ (¬ e))]
Delta:: []
V:: [(e:True), (l:None), (o:None), (ab_1:None), (ab_2:False)]
R:: {'abducibles': []}

response(p_bar): We are uncertain if she will study late in the library
------------------------------
Epistemic state:
```

FIGURE 1: Epistemic states demonstrating that an `adEXP` can be used to model the individual case of the Suppression Task where Suppression is prevented. Part1.

```
===>elo<===
S:: [(e ↔ ⊤), (l ↔ ((e ∧ (¬ ab_1)) ∨ (o ∧ (¬ ab_2)))), (ab_1 ↔ (¬ o)), (ab_2 ↔ (¬ e)), (o ↔ ⊤)]
Delta:: []
V:: [(e:True), (l:True), (o:True), (ab_1:False), (ab_2:False)]
R:: {'abducibles': [(o ← ⊤)]}

response(p_bar): She will study late in the library
------------------------------
Epistemic state:

===>elo<===
S:: [(e ↔ ⊤), (l ↔ ((e ∧ (¬ ab_1)) ∨ (o ∧ (¬ ab_2)))), (ab_1 ↔ (¬ o)), (ab_2 ↔ (¬ e)), (o ↔ ⊥)]
Delta:: []
V:: [(e:True), (l:False), (o:False), (ab_1:True), (ab_2:False)]
R:: {'abducibles': [(o ← ⊥)]}

response(p_bar): She will not study late in the library
------------------------------
Epistemic state:

===>elo<===
S:: [(e ↔ ⊤), (l ↔ ((e ∧ (¬ ab_1)) ∨ (o ∧ (¬ ab_2)))), (ab_1 ↔ (¬ o)), (ab_2 ↔ (¬ e)), (o ↔ (⊤ ∨ ⊥))]
Delta:: []
V:: [(e:True), (l:True), (o:True), (ab_1:False), (ab_2:False)]
R:: {'abducibles': [(o ← ⊤), (o ← ⊥)]}

response(p_bar): She will study late in the library
------------------------------
predictions:  {'el': ['She will study late in the library', 'She will study late in the library', 'She will study late in the
library', 'She will study late in the library'], 'elo': ['We are uncertain if she will study late in the library', 'She will
study late in the library', 'She will not study late in the library', 'She will study late in the library']}

f(pi) models gamma_noSup? :  True
```

FIGURE 2: Epistemic states demonstrating that an `adEXP` can be used to model the individual case of the Suppression Task where Suppression is prevented. Part2.

```
def standardSuppression():
    print ("============================================================")
    print ("================STANDARD SUPPRESSION========================")
    print ("============================================================")
    #The epistemic state for the initial state of a reasoner who only knows
    #conditional ( l | e )
    basePoint1=epistemicState.epistemicState('el')
    delta1=["( l | e )"]
    S1 = ["( e <- T )"]
    delta1AsLogic = scpNotationParser.stringListToBasicLogic(delta1)
    S1AsLogic = scpNotationParser.stringListToBasicLogic(S1)
    basePoint1['S']=S1AsLogic
    basePoint1['Delta']=delta1AsLogic
    basePoint1['V']=[e,l]

    #The elo case expressed as the addition of information to the el case
    basePoint2=copy.deepcopy(basePoint1)
    basePoint2.setName('elo')
    #The possible starting states for the SCP
    extraConditional=["( l | o )"]
    extraConditionalAsLogic = scpNotationParser.stringListToBasicLogic(extraConditional)
    basePoint2['Delta']=basePoint2['Delta']+extraConditionalAsLogic
    basePoint2['V']=basePoint2['V']+[o]

    #Create the first state point
    statePoints=[basePoint1,basePoint2]
    s_i=statePoints

    f=f_suppression_studyLate
    #The desired output of the external evaluation function
    gamma={'el':'She will study late in the library',
           'elo':'We are uncertain if she will study late in the library'}

    #test ctm
    c = CTM.CTM()
    c.setSi(s_i)
    c.appendm(ADDAB)
    c.appendm(WC)
    c.appendm(SEMANTIC)

    predictions = f(c)
    print ('predictions: ', predictions)
    print ("Lenient Interp")
    print (StatePointOperations.predictionsModelsGamma_lenient(predictions,gamma))
    print("Strict Interp")
    print (StatePointOperations.predictionsModelsGamma_strict(predictions,gamma))
    print ("f(pi) models gamma_Sup? : ",
           (StatePointOperations.predictionsModelsGamma_lenient(predictions,gamma)))
```

FIGURE 3: Code snippet showing how the Suppression Task is modelled in the SCP Framework.