

Reasoning About Priming Effects

MSc Thesis (*Afstudeerscriptie*)

written by

Albert Emil Mølgaard Thayssen

(born December 28th, 1998 in Aarhus, Denmark)

under the supervision of **Prof. Dr. Sonja Smets**, and submitted to the Examinations Board in
partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam*.

Date of the public defense: **Members of the Thesis Committee:**
July 7, 2026

Dr. Maria Aloni (chair)

Prof. Dr. Sonja Smets

Prof. Dr. Francesco Berto

Dr. Martha Lewis



INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

Abstract

This thesis develops a formal-logical framework for representing and reasoning about content priming effects and their influence on belief-formation, where the relevant contents are properties or concepts. Content priming effects are cases in which exposure to a stimulus increases the accessibility of contents stored in memory, thus influencing a response to a subsequent stimulus or task. This response can manifest as observable behavior that can provide new information about the acting agent to any observing agent. In turn, this newly acquired information can lead to the formation of new beliefs. In developing our framework, we start by explaining the conceptual spaces framework, which we will use to (i) represent properties and concepts, and (ii) specify a measure of similarity between such contents. We then embed (i) and (ii) into a model of how content-processing can increase content accessibility, which constitutes the first original contribution of this thesis. In doing so, we first develop a general means for representing processing-induced accessibility increases. Then, we present a similarity-based account, which is used throughout the rest of the thesis. The second original contribution of this thesis is a formal model of how content accessibility can influence behavior. In this model, the key notions are those of *selection score* and *executability condition*. Selection scores measure how *strongly* an action is selected for based on content accessibility, while executability conditions tell us *which* action is selected based on this measure. We then use this model of action selection, together with our similarity-based account of processing-induced accessibility increases, to analyze two famous examples of content priming effects popularized by Daniel Kahneman. Finally, we cast the newly developed machinery in a formal-logical framework. This is done in two steps. First, we introduce a logic that can capture the effects of content processing on content accessibility, and express selection scores and executability conditions. Then, we revisit one of the earlier analyzed scenarios and show that our logic delivers the correct results. Secondly, we extend this logic to capture the beliefs of agents, as well as how agents update their beliefs with new information gained from observed behavior, where the behavior in question is caused by underlying degrees of content accessibility. This extension allows us to formally capture how content priming can affect belief-formation, which is the third original contribution of this thesis. Lastly, we put the machinery to use, presenting and analyzing a multi-agent scenario that show how content priming can induce a false belief.

Acknowledgments

First and foremost, I want to thank Sonja, my supervisor, whose enthusiasm, willingness to help, and passion for logic helped me bring the ideas of this thesis to life, and whose expertise helped guide me through the intricacies of building a logic. I'm excited to use the skills I've learned from you in the future. Thank you, Sonja, for your unwavering support during my MoL journey; it's been a pleasure working with you, and I'm excited to work together in the future!

I also want to give my biggest thanks to Aybüke. I could not have asked for a more helpful, supportive, and competent mentor and teacher. Thank you for your guidance, whether in-person or online. I look forward to (hopefully!) working with you again.

Next, I want to thank my other committee members: Franz, Martha, and Maria. Thank you for taking the time to read through my thesis and participate in my defense. It was a pleasure hearing all of your insights and engaging with your questions.

A special thank you to Karolina and Martha for meeting with me to discuss some of the ideas of my thesis. Your expertise greatly helped me develop my ideas further and progress my thesis.

I also extend my thanks and gratitude to two logicians from Aarhus University. Thank you, Jens and Lars, for showing me the wonderful world of logic during my time as an undergraduate student, and for sparking my interest in dynamic epistemic logic — the reason why I applied to the MoL. You have both been a great influence on my thinking and writing.

I also want to thank Anika for being with me through thick and thin. Your support and willingness to help have been amazing. Lastly, I want to thank my friends. A special thank you goes to Laurits, William, and Anton, who have helped me stay sane through the writing of this thesis, whether it's been through gaming, working out, chess, or just hanging out.

Contents

1	Introduction	3
2	Conceptual spaces	6
2.1	Dimensions, domains, and conceptual spaces	7
2.2	Distances and similarities	11
2.3	Properties, concepts, and prototypes	17
2.4	Updating accessibility	25
3	Examples	29
3.1	Accessibility-based action selection	29
3.2	Canonical examples	31
3.2.1	SOUP or SOAP	31
3.2.2	The Florida Effect	39
4	A logic of content priming	44
4.1	Content processing and executability	44
4.1.1	Syntax	44
4.1.2	Semantics	45
4.1.3	Executability conditions	48
4.1.4	SOUP or SOAP, revisited	50
4.2	Belief dynamics	52
4.2.1	Syntax	53
4.2.2	Doxastic content priming models	53
4.2.3	Model transformations and truth clauses	55
4.2.4	Putting the machinery to use	56
5	Conclusion	60
5.1	Summary	60
5.2	Future work	61
	Bibliography	62

Chapter 1

Introduction

This thesis develops a formal-logical framework for representing and reasoning about content priming effects and their influence on belief-formation, where the relevant contents are properties or concepts. Priming effects are cases in which exposure to a stimulus influences a response to a subsequent stimulus or task, typically without conscious awareness [Kah11; Rei18].¹ The focus of this thesis is on the type of priming known as *content priming*. Content priming effects are cases in which exposure to a stimulus increases the accessibility of contents stored in memory², thus influencing a response to a subsequent stimulus or task [JW14].³ The phenomenon is studied in a wide range of disciplines, including, but not limited to, cognitive science [YHT15], cognitive psychology [HS12], social psychology [Mol14; BC00], consumer psychology [JW14], and behavioral economics [Xio25]. Consider the following example of a content priming effect, presented by Daniel Kahneman [Kah11, page 52]:

“If you have recently seen or heard the word EAT, you are temporarily more likely to complete the word fragment SO_P as SOUP than as SOAP. The opposite would happen, of course, if you had just seen WASH.”

In this example, exposure to the stimulus of hearing or seeing the word ‘EAT’ influences one’s response to the subsequent task of completing the word fragment ‘SO_P’ by briefly increasing one’s tendency to complete it as ‘SOUP’ instead of ‘SOAP’. The standard explanation is that seeing or hearing the word ‘EAT’ brings about the processing of certain content stored in one’s long-term memory (LTM), thus leading to an increase in the accessibility of the concept expressed by ‘SOUP’. This, in turn, makes one temporarily more likely to complete the fragment as ‘SOUP’ over ‘SOAP’ (and analogously for ‘WASH’ and ‘SOAP’) [JW14, pages 97–98; Kah11, pages 52–53]. Other cases of content priming found in the literature include studies showing that the processing of words typically associated with the elderly can lead to reduced walking speed (the so-called Florida effect) [Kah11, pages 53–54], and that exposure to a stimulus leading to the processing of evaluative-concepts (e.g. concepts like the ones expressed by ‘good’ and ‘bad’) can affect product evaluations of consumers, even when the initial stimulus is irrelevant to the product being evaluated [JW14, page 100]. The examples above illustrate how content priming can influence our decisions, actions, and attitudes.⁴ Furthermore, content priming effects can

¹In empirical studies on priming effects, the first stimulus is often called ‘the priming stimulus’ and the second ‘the target stimulus’ [JW14, page 97]. When the priming stimulus influences a response to a subsequent task, we will this ‘the target task’.

²We will give a precise interpretation of the term ‘accessibility’ in Section 2.4. Intuitively, the more accessible a piece of content is, the more easily it is retrieved from memory.

³Henceforth, we will use ‘priming’ and ‘content priming’ interchangeably.

⁴For more examples, we refer the reader to the review on content priming by Janiszewski and Wyer [JW14].

indirectly influence belief-formation.

We adopt the general picture of content priming effects from the review by Janiszewski and Wyer [JW14]: content priming effects can be explained by increases in the accessibility of content stored in agents’ LTM, brought about by exposure to an initial stimulus. These increases in content accessibility, in turn, influence a response to a subsequent stimulus or task. In particular, the types of content we focus on are properties and concepts.⁵ Furthermore, we will think of an agent’s exposure to an initial stimulus as (1) bringing about the processing of some content P stored in that agent’s LTM and (2) the increase in accessibility of any content C stored in that agent’s LTM as a function of some specified relation between P and C .⁶

With these preliminaries in place, we list the components needed in a framework for representing and reasoning about content priming effects and their influence on belief-formation. Firstly, we need a representation of the contents stored in agents’ memory. Secondly, we need a model of content-processing-induced accessibility increases. Thirdly, a model of how content accessibility can influence behavior. And fourthly, an embedding of the previous components into a model of belief and belief-update. We now give an overview of the rest of the thesis.

Chapter 2 deals with the first and second components, for which we rely on the conceptual spaces framework [Gär00]. In Sections 2.1 through 2.3, we explain the basics of the framework. The framework allows for a “structured” representation of properties and concepts, such that it is possible to formally represent different relations between content based on this structure [BK17b]. These relations can provide a basis for computing content accessibility increases as induced by content processing. In particular, we will adopt the picture that processing a piece of content increases the accessibility of other content in proportion to their similarity. This ensures that accessibility increases are not stipulated, but derived from the structure of the contents. Modeling processing-induced accessibility increases by utilizing the conceptual spaces framework is our own contribution and will be explicated in Section 2.4.

In Chapter 3, we start by dealing with the third component and build a model of accessibility-based action selection. This allows us to bridge processing-induced increases in content accessibility and behavioral changes. The central idea behind the model is that the accessibility levels of contents matter for behavior insofar as it is relevant to the task at hand, as stressed by Janiszewski and Wyer [JW14, pages 97–98]. To capture this, we introduce the notion of a *relevance function*. A relevance function specifies what content is relevant to the selection of an action. We then introduce the notion of a selection score that is intended to measure how much support an action has, given the accessibility levels of content that is relevant to the selection of that action. Candidate actions are then thought of as “competing for selection” by their respective selection scores. If one candidate action has a strictly higher score than the others, it is selected; otherwise, a designated fallback action is selected. The culmination of this Chapter is to use the tools presented till this point and formally analyze a scenario based on the earlier presented “SOUP or SOAP” example, and formally analyze a scenario based on the Florida effect as explained above.

In Chapter 4, we cast the machinery developed in Chapters 2 and 3 within a formal logic. First, in Section 4.1, we introduce a logic of content processing that also allows for the comparison between different selection scores of actions. We then use this logic to analyze the SOUP or SOAP scenario

⁵Henceforth, we will use the term ‘content’ to mean a property or concept.

⁶We specify the relation we will use throughout this thesis in Section 2.4.

presented in Section 3.2.1, thus showing that the logic does a satisfactory job in representing processing-induced accessibility increases. Then, in Section 4.2, we extend this logic with belief operators, as well as dynamic operators for capturing how agents update their beliefs with new information gained from observed behavior, with that behavior brought about by accessibility-based action selection. This allows us to reason not only about how content processing can affect action selection, but also about how the behavioral changes caused by such processing can affect belief. This takes care of the fourth component needed in our framework. Finally, in Section 4.2.4, we put this extended logic to use by analyzing a scenario in which content priming affects the belief of an agent.

In Chapter 5, we summarize the thesis and present some ideas for future research.

Chapter 2

Conceptual spaces

In developing our framework for representing and reasoning about priming effects, we know from the previous Chapter that we need to represent the contents agents have stored in their LTM and specify how processing one piece of content increases the accessibility of others. A desideratum, we think, is to find a principled method for computing the accessibility increases of contents induced by this processing, as opposed to arbitrarily stipulating the increases directly.¹ Developing a principled method can help identify underlying mechanisms of content priming effects, whereas arbitrary stipulation fails to do so. As mentioned in the introductory Chapter, the conceptual spaces framework [Gär00] jointly allows us to represent the contents stored in agents' LTM, and can, as we will show, be utilized for computing accessibility increases induced by content processing in a principled way with. It is standard to use the conceptual spaces framework to represent content, particularly properties and concepts. Using the framework to compute accessibility increases induced by content processing is our own contribution. We now briefly situate and explain the framework before proceeding with the formal presentation.

The conceptual spaces framework was developed by Peter Gärdenfors in several papers [Gär96; Gär88; Gär90], and culminated in his seminal book *Conceptual Spaces: The Geometry of Thought* [Gär00]. In general, the framework serves as a tool for modeling the representations cognitive systems use to make classifications [Gär00, pages 1–2]. Its uses range from theorizing about the cognition of humans or other animals, to more application-based “constructive” purposes, such as building an artificial system to solve some specified task [Gär00, pages 1–4]. The guiding idea behind the framework is that our experience is inherently structured and that we have the capacity to judge our experiences as similar or different based on these structures [Gär00, pages 4–8].² When perceiving an apple, we are not perceiving an unstructured “blob” of matter. Depending on how we are interacting with the apple and which of our senses are functioning, we are perceiving it as having a color, shape, smell, weight, taste, and so on. Our concept of an apple is structured around these different “dimensions” along which we judge our experiences. Furthermore, we can judge apples as similar or different along these dimensions. Intuitively, two apples that are close along the dimension of color are similar with respect to their color. The conceptual spaces framework can capture these intricacies in the representations

¹To clarify, let us denote the accessibility of a piece of content C by ‘ $\text{Acc}(C)$ ’, and the accessibility of a piece of content C after processing content P by ‘ $\text{Acc}_P(C)$ ’. Then, roughly speaking, we think a principled method for computing accessibility increases should provide (i) an agent-neutral general statement of the form: For any contents C and P , $\text{Acc}_P(C) = \text{Acc}(C) + \iota(P, C)$ where ι is a function specifying the accessibility increase of C upon processing P , and (ii) an explanation of why the given specification of ι is appropriate for capturing accessibility increases induced by content processing.

²Gärdenfors uses ‘experience’ as an umbrella term for perception, imagination, and memory [Gär00, page 5].

that cognitive systems use in categorization. Our exposition of the framework is based on [Gär00], combined with the more recent work of [PG17; PM16; BK17b; BK17a].

2.1 Dimensions, domains, and conceptual spaces

We start by introducing the notion of a dimension as used in the conceptual spaces framework. These serve as the “building blocks” of the framework by underlying the framework’s representation of properties and concepts [Gär00, page 5]. Dimensions are ways in which stimuli can be judged to be similar or different [Gär00, pages 6–8; BK17b, page 2; BK17a, page 59]. Examples include weight, height, bitterness, and smoothness. For example, we can judge dark-roasted coffee to be more bitter than a candy bar. Similarly, we can judge Pedro Pascal to be shorter than Hafthór Júlíus Björnsson. And, we can judge a piece of sandpaper to be less smooth than freshly washed and dried sheets. A value along a dimension is a feature, or “quality”, that a thing can satisfy with respect to that dimension. Consider the dimension of weight and suppose that we measure weight in kilograms. An object can satisfy having a weight of 200 kilograms. As such, 200 kilograms is a value along the weight dimension. Furthermore, if we have three objects x , y , and z that weigh 25, 50, and 75 kilograms, respectively, then x is closer to y than z along the weight dimension, while y is equally close to x and z . In this sense, x is more similar to y than z along the weight dimension, whereas y is equally similar to x and z . In general, the intuition is that the closer two objects are along a dimension, the more similar they are with respect to that dimension; the further apart they are, the more dissimilar [Gär00, page 5]. In stating our formal definition of a dimension, we follow Bechberger and Kühnberger [BK17b; BK17a] and model them as subsets of real numbers.³ In addition to this, we assume that dimensions are *non-empty*.⁴

Definition 1 (Dimension (adapted from [BK17b, page 2; BK17a, page 59])). *A dimension d is a non-empty subset of $\mathbb{R}$, with each $x \in d$ representing a specific value along that dimension.*

The reason for modeling dimensions as sets of real numbers is two-fold. Firstly, it ensures that we have an intuitive notion of distance along any dimension: following Bechberger and Kühnberger [BK17b, page 2; BK17a, page 59], we define distances along dimensions using absolute values. In particular, for any dimension d and any points $x_d, y_d \in d$, the distance between x_d and y_d is given by $|x_d - y_d|$, where $|\cdot|$ is the standard absolute value function. Secondly, using absolute values allows us to use the only two metrics that have currency in the psychological literature for measuring distances in the conceptual spaces framework [Dou+22a, page 3]: the Euclidean metric and the Manhattan metric. We follow Bechberger and Kühnberger [BK17a, pages 2–3] and use a weighted Euclidean metric for measuring distances within what is called a domain and a weighted Manhattan metric for measuring distances within a conceptual space. Domains and conceptual spaces are notions we turn to soon. We leave distances aside for the rest of this section, giving them a full treatment in the following one. We now return to dimensions.

³While Bechberger and Kühnberger [BK17b; BK17a] do not explicitly state that dimensions are sets of real numbers, they implicitly make this identification as they define distances along dimensions using absolute values, their example dimensions are sets of real numbers, and their formalization, in general, seems to rest on the assumption that dimensions are sets of real numbers.

⁴We make the non-emptiness of dimensions explicit. The requirement is arguably already implicit in the treatments [Gär00; BK17b; BK17a]. We state it because it helps ensure that the notions we introduce for our logic are well-defined further on.

Following Gärdenfors [Gär00], and Gärdenfors and Massimo Warglien [GW12], we distinguish between *quality dimensions* [Gär00, pages 6–8; GW12, pages 2–4] and *force dimensions* [Gär00, pages 94–99; GW12, pages 9–14]. When a dimension represents a way in which objects can be judged to be similar or different, it is called a quality dimension [Gär00, pages 6–8; GW12, pages 2–4]. As an example, consider weight measured in kilograms. This is a way in which objects can be judged to be similar or different, and we can model this with the set of real numbers $d = \mathbb{R}_{\geq 0}$ where each $x \in d$ represents a weight of x kilograms. Thus, if an object is assigned $x \in d$ for some agent, it means that they judge that object as weighing x kilograms. Given the above, d is a quality dimension. When a dimension represents a way in which actions can be judged to be similar or different, it is called a force dimension [Gär00, pages 94–99; GW12, pages 9–14]. As an example, consider the notion of duration in seconds. This is a way in which actions can be judged as similar or different, and we can model this with the set of real numbers $d = \mathbb{R}_{\geq 0}$, where each $x \in d$ represents an action as having a duration of x seconds. Thus, if an action is assigned $x \in d$ for some agent, it means they judge the action to have a duration of x seconds. Given the above, d is a force dimension.

Some dimensions inherently belong together in the sense that one cannot assign a stimulus a value on one without assigning it a value on the others [Gär00, page 24]. Such dimensions are called *integral* [Gär00, page 24]. Using an example from Gärdenfors [Gär00, page 24], consider pitch and loudness. Pitch and loudness are quality dimensions, as sounds can be judged to be similar or different based on their pitch and loudness. Furthermore, when hearing a sound, one cannot hear its pitch without its loudness, and vice versa. As such, the quality dimensions of pitch and loudness are integral. There are also dimensions where one can assign a stimulus a value along one without having to assign it a value along all the others. In this sense, such dimensions do not inherently belong together. Such dimensions are called *separable* [Gär00, page 24]. As an example, consider the quality dimensions of pitch and weight. One can judge a stimulus as having a pitch without simultaneously judging it as having a weight —if one hears the chirping of a bird and judges it as having a certain pitch, one does not simultaneously judge the chirping as having a weight (whatever that would mean). We follow the explicit set-level definition of integrality and separability by Masolo and Porello [PM16, page 2] and say that a set of dimensions is integral if and only if of assigning a stimulus a value along one of its dimensions forces one to assign it a value along its other dimensions, and separable if and only if it is not integral — that is, if one assigns a value to a stimulus along one of the dimensions of that set, one is not forced to assign it a value along the other dimensions of the set.⁵ With these notions in place, we can define a domain. In stating our formal definition of a domain, we follow Gärdenfors [Gär00] and take a domain to be a set of integral dimensions that is separable from all dimensions not in that set. Following Masolo and Porello [PM16, page 6], who take a domain to comprise finitely many quality dimensions, we assume that domains are finite. In addition to this, we assume that domains are *non-empty*:⁶

Definition 2 (Domain (adapted from [Gär00, page 26])). *Let D be a set of dimensions. Then a non-empty set of dimensions $\delta \subseteq D$ is a domain if and only if δ is integral and for every dimension $d \in \delta$, $\{d\} \cup (D \setminus \delta)$ is separable.*

⁵These notions are not explicitly defined for sets of dimensions by Gärdenfors, although he uses them in passing [Gär00, pages 25–26].

⁶As with the corresponding requirement for dimensions, we make the non-emptiness of domains explicit. However, the requirement is arguably already implicit in the treatments [Gär00; PM16; PG17; BK17b; BK17a]; we state it because, as before, it helps ensure that the notions we introduce for our logic are well-defined further on.

When applying the conceptual spaces framework to modeling tasks, it may be necessary to treat a set of dimensions as a domain even if they do not strictly satisfy the integrality condition [Gär00, pages 25–26]. What seems to matter most in such cases is that the chosen sets of dimensions are mutually separable from one another, so that the partitioning of the relevant dimensions into domains still does meaningful work in organizing information and structuring distance computations. As an example of an idealization of the aforementioned kind, one might treat the sets of dimensions $\delta_{taste} = \{d_{sweet}, d_{salt}\}$ and $\delta_{shape} = \{d_{round}, d_{square}\}$ as domains. These sets of dimensions are extensions of ones presented on [BK17b, page 10]. In particular, we have extended δ_{taste} with d_{salt} , and δ_{shape} with d_{square} . Our interpretation of d_{salt} and d_{square} are inspired by Bechberger and Kühnberger’s [BK17b, page 10] interpretations of d_{sugar} and d_{round} , respectively. The dimensions of the taste “domain” δ_{taste} are interpreted as follows:

- $d_{sweet} = [0, 1]$ represents the relative sugar content of an object, ranging from zero sugar, indicated by a value of 0.00, to high levels of sugar, indicated by a value of 1.00;
- $d_{salt} = [0, 1]$ represents the relative salt level of objects, ranging from zero salt, indicated by a value of 0.00, to high levels of salt, indicated by a value of 1.00,

and the dimensions of the shape “domain” δ_{shape} are interpreted as follows:

- $d_{round} = [0, 1]$ represents the roundness of objects, measured by the percentage of the bounding sphere of an object that is filled by that object;
- $d_{square} = [0, 1]$ represents the squareness of objects, measured by the percentage of the bounding rectangle of an object that is filled by that object.

Intuitively, one can taste sweetness without tasting salt, and one can judge the roundness of an object without judging the squareness of an object. However, as the former dimensions are both relevant for taste, they are treated as a domain. And, as the latter dimensions are both relevant for judging the shapes of objects, they are treated as a domain. And, clearly, each dimension in δ_{taste} is separable from δ_{shape} , and vice versa.

We now have all the requisite notions to define conceptual spaces. While Gärdenfors proposes that a conceptual space can be defined as a set of one or more domains [Gär00, page 26], we follow the modern treatments of [PG17; PM16; BK17b; BK17a] and define conceptual spaces as a type of product space.⁷ In particular, we follow [PM16] and [PG17] in taking an underlying finite non-empty set of domains over which a conceptual space is explicitly defined. But, whereas [PM16] and [PG17] defines a conceptual space as any subset of the cartesian product of this set of domains⁸, we take a conceptual space as being the product of every dimension in every domain of the underlying set of domains. This is in line with the definition of a conceptual space in [BK17a] and [BK17b], where a conceptual space is defined as the cartesian product of a set of dimensions. However, where we differ from [BK17a] and [BK17b] is that we assume an underlying finite non-empty set of domains from which the space is constructed. We now state our formal definition:

⁷We note that while defining a conceptual space as a *set* of one or more domains is one of the definitions that Gärdenfors proposes, he also talks of “points” in a conceptual space and take these points to represent possible objects [Gär00, page 26]. This talk of points of a conceptual space seems very suggestive of some underlying product-space structure, where the points are tuples of values defined over the dimensions contained in the underlying set of domains.

⁸The only reason why [PM16] and [PG17] take a conceptual space as a subset of a product of domains, as opposed to a subset of a product of dimensions, is that for their purposes, the authors do not find it relevant focusing on dimensions [PM16, page 6; PG17, page 4].

Definition 3 (Conceptual space (adapted from [PM16, page 6; PG17, page 4; BK17b, page 2; BK17a, page 59])). *Let $\Delta = \{\delta_1, \dots, \delta_n\}$ be a finite non-empty set of domains and $D = \bigcup_{i=1}^n \delta_i$ the collection of all dimensions in any of the domains in Δ . Then $CS = \prod_{d \in D} d$ is a conceptual space.*

We now give an example of a conceptual space that can be used in classifications based on color and sound. The domain we use for colors is taken from [BK17b, page 10]. Consider the set of domains $\Delta = \{\delta_{color}, \delta_{sound}\}$ where:

- $\delta_{color} = \{d_{hue}, d_{brightness}, d_{chromaticness}\}$ is a domain for representing colors;
- $\delta_{sound} = \{d_{pitch}, d_{loudness}, d_{duration}\}$ is a domain for representing sounds.

The dimensions of the color domain are interpreted as follows [BK17b, page 10]:

- $d_{hue} = [0, 1]$ represents the hue of objects, ranging from purple, indicated by a value of 0.00, to red, indicated by a value of 1.00;
- $d_{brightness} = [0, 1]$ represents the brightness of objects, ranging from black, indicated by a value of 0.00, to white, indicated by a value of 1.00;
- $d_{chromaticness} = [0, 1]$ represents the chromaticness of objects, ranging from zero color intensity (gray), indicated by a value of 0.00, to the highest color intensity, indicated by a value of 1.00.

The dimensions of the sound domain are interpreted as follows:

- $d_{pitch} = [20, 20000]$ represents the pitch of sounds measured in Hz, with lower frequencies being deeper sounds, and higher frequencies more shrill sounds;
- $d_{loudness} = [0, \infty)$ represents the loudness of sounds measured in decibels (dB);
- $d_{duration} = [0, \infty)$ represents the duration of sounds measured in seconds;

Given $\Delta = \{\delta_{color}, \delta_{sound}\}$, the collection of all dimensions in any of the domains in Δ is $D = \bigcup_{\delta \in \Delta} \delta = \{d_{hue}, d_{brightness}, d_{chromaticness}, d_{pitch}, d_{loudness}, d_{duration}\}$. Thus, $CP = \prod_{d \in D} d = d_{hue} \times d_{brightness} \times d_{chromaticness} \times d_{pitch} \times d_{loudness} \times d_{duration}$ is a conceptual space. Elements of a conceptual space are tuples of values along the dimensions of that space, where each such value represents a feature or quality something can satisfy with respect to the given dimension. These can be used to represent several things [PM16, pages 3–6]. One is to have them represent objects [Gär00, pages 134–135; PM16, page 3], thus identifying an object by the values the tuple representing it has along the dimensions of the space. However, Masolo and Porello [PM16, pages 3–6] argue that this is an inadequate model of agents’ representations of objects when they have incomplete information about those objects.⁹ In cases where an agent has incomplete information about some object, Masolo and Porello propose to model that agent’s representation of the object as a subset of the agent’s conceptual space, with each element in this set representing a possible set of features for the object in question (as opposed to the object itself), given that agent’s information [PM16, pages 7–13]. Then, the more points the agent considers as describing possible set of values for the object in question, the larger their uncertainty about the object. The fewer points they consider, the larger their certainty. Furthermore, we note that some objects cannot take values along certain dimensions [PM16, page 3]. As an example from [PM16,

⁹It is too big of an aside going into their argument. We refer the reader to [PM16, pages 3–6].

page 3], consider any paradigmatic abstract objects, such as numbers or sets. Such objects cannot be assigned values along the dimensions of spatial location, or time [PM16, page 3] (or other dimensions capturing features characteristic of *physical* objects, such as weight). To formally account for irrelevant or inapplicable dimensions, dimensions can be equipped with a distinguished element, denoted by $\star$, indicating that no value has been chosen from that dimension [PM16, p. 3]. Then, if we are working with abstract objects in a conceptual space with an inapplicable dimension, we assign them $\star$ along these dimensions. We leave further a discussion and full implementation of the $\star$ -value for future work. For now, we assume that objects are assigned a value along every dimension in a conceptual space.

2.2 Distances and similarities

In this Section, we specify our chosen notions of distance, as well as show how similarity between elements of a conceptual space is modeled. The background of this Section is [Gär00], whereas we use the formal definitions of [BK17b] and [BK17a].

As mentioned in the previous section, distances in a conceptual space have a natural interpretation: The closer two points are along a dimension, domain, or conceptual space, the more similar they are with respect to that dimension, domain, or conceptual space, respectively [Gär00, page 5].¹⁰ In the conceptual spaces framework, similarity is modeled as an exponentially decreasing function of distance [Gär00, pages 20–21; BK17a, page 60; BK17b, page 3]. The rest of the chapter will flesh out these ideas formally. We start by presenting measures of distance for dimensions, domains, and conceptual spaces. We follow Bechberger and Kühnberger [BK17b; BK17a] and use absolute value distances along dimensions, a weighted Euclidean metric for domains, and a weighted Manhattan metric for full spaces. These notions of distance will be used in defining similarity as a decreasing function of distance. Once again, we follow the formal definition of similarity by Bechberger and Kühnberger [BK17b; BK17a]. All the notions of distance of Bechberger and Kühnberger [BK17b; BK17a] are defined with respect to their definition of a conceptual space. And, as mentioned in the previous Section, Bechberger and Kühnberger [BK17b; BK17a] define a conceptual space as a product of dimensions, whereas we define a conceptual space as a product of dimensions from some underlying set of domains. As both our definitions of a conceptual space take them as products of dimensions, we readily use the definitions of distance as they appear in [BK17b] and [BK17a]. In multiple cases, we slightly modify the notation of the definitions of [BK17b] and [BK17a], albeit the definitions themselves are the same. Lastly, we borrow some notation from [BK17b] and [BK17a]: Let CS is a conceptual space built on the set of dimensions D . Then, if $x \in CS$ and $d \in D$, we will denote the value of x along dimension d by x_d . We now state the formal definition of distances along dimensions as follows:¹¹

Definition 4 (Distance along dimensions ([BK17a, page 59; BK17b, page 2])). *Let CS be a conceptual space, d a dimension on which CS is built, and $x, y \in CS$ points. The distance between x and y along*

¹⁰We note that similarity does not have to be perceptual, such as when two pieces of fruit are similar in having the same color and shape. It can also be functional, such as a chair and a sofa being functionally similar insofar as they afford sitting. Or, it can be occupational similarity, such as a nurse and a doctor being similar insofar as they typically work in the same environment. And, there are many other examples. The point is that similarity is understood in a very specific, albeit broad, way. It is specific as it is measured by means of distances along dimensions, domains, or full spaces. It is broad insofar as it is understood as giving a notion of similarity for anything that can be represented at the level of dimensions, domains, and full spaces.

¹¹The notation ' d_{CS}^d ' is our own, albeit it is inspired by the notation Bechberger and Kühnberger [BK17b] and [BK17a] use for their domain and conceptual space distance measures.

dimension d in space CS is denoted by $d_{CS}^d(x, y)$ and defined as:

$$d_{CS}^d(x, y) := |x_d - y_d|.$$

As an example, suppose that we have a conceptual space CS built on the set of dimensions D in which $d_{weight} = \mathbb{R}_{\geq 0} \in D$ is a dimension representing weight in kilograms. Now consider two elements $x, y \in CS$ where $x_{d_{weight}} = 10$ and $y_{d_{weight}} = 1000$. Then, the distance between x and y along dimension d_{weight} in space CS is:

$$\begin{aligned} d_{CS}^{d_{weight}}(x, y) &= |x_{d_{weight}} - y_{d_{weight}}| \\ &= |10 - 1000| \\ &= |-990| \\ &= 990 \end{aligned}$$

Moving on to distances within domains. These are defined using distances along dimensions together with the notion of a *salience weight* [Gär00, pages 18–20, 103–104][BK17a, pages 59–60; BK17b, pages 2–3]. Salience weights represent the relative importance of a dimension in a domain in some given context. The intuition behind salience weights is that the relative importance of each dimension in a domain is context-dependent [Gär00, pages 103–104; BK17a, page 60; BK17b, page 3]. As an example, consider the previously introduced domain $\delta_{sound} = \{d_{pitch}, d_{loudness}, d_{duration}\}$ for representing sounds.¹² In the context of judging whether two notes are the same, the dimension d_{pitch} has higher salience than the dimensions $d_{loudness}$ and $d_{duration}$, as notes are distinguished by their pitch. On the other hand, in the context of designing a hearing protection aid, $d_{loudness}$ has higher salience than both d_{pitch} and $d_{duration}$, as the main way that a hearing protection aid works is by reducing the loudness of sounds for the bearer of the aid. Following Bechberger and Kühnberger [BK17b; BK17a], where δ is a domain and $d \in \delta$ a dimension, a salience weight for d in δ is a non-negative real number denoted by ' w_d '. Furthermore, we follow Bechberger and Kühnberger [BK17b; BK17a] and impose the following restriction on sets of salience weights for domains: where $W_\delta = \{w_d \in \mathbb{R}_{\geq 0} : d \in \delta\}$ is a set of salience weights for the domain δ , we require that $\sum_{w_d \in W_\delta} w_d = 1$. We now state the formal definition of distances within domains:

Definition 5 (Distance within domains ([BK17b, page 2; BK17a, page 59])). *Let CS be a conceptual space, δ a domain of that space, W_δ a set of salience weights for δ , and $x, y \in CS$ any points. The distance between x and y in domain δ given W_δ is denoted by $d_{CS}^\delta(x, y, W_\delta)$ and defined as the weighted Euclidean metric:*

$$d_{CS}^\delta(x, y, W_\delta) = \sqrt{\sum_{d \in \delta} w_d \cdot |x_d - y_d|^2} = \sqrt{\sum_{d \in \delta} w_d \cdot (d_{CS}^d)^2}$$

We now present the intuition behind using this metric for intra-domain distances.¹³ A domain is a set of integral dimensions. And, integral dimensions belong together, so we cannot simply consider them

¹²For the interpretation of the dimensions of δ_{sound} , we refer back to the example of a conceptual space (Definition 3) in Section 2.1.

¹³It is beyond the scope of this thesis to review the psychological evidence for this choice. However, we note that (i) it is one of the only metrics with currency in the psychological literature [Dou+22b, page 3], (ii) Bechberger and Kühnberger justify the choice of this metric with reference to psychological evidence presented in the papers [She64; Att50; Joh01], (iii) and Gärdenfors prefers this choice as well [Gär00, pages 24–25].

in isolation from each other. The weighted Euclidean metric captures the integrality of the dimensions [Gär00, pages 24–25]. Firstly, it is captured by the weights being salience weights, as these represent how an agent’s salience, or attention, is distributed along the dimensions of the domain. Secondly, it is captured insofar as a Euclidean metric “considers” the intra-domain distances together, rather than in isolation. It considers them together in the sense that Euclidean distance is computed by summing the squared differences along the dimensions before taking the square root of this sum, rather than, say, adding the dimension-wise distances directly.¹⁴

As an example of a computation of intra-domain distance, consider a conceptual space CS built on the previously introduced color domain $\delta_{color} = \{d_{hue}, d_{brightness}, d_{chromaticness}\}$.¹⁵ Now take the set $W_{\delta_{color}} = \{w_{d_{hue}}, w_{d_{brightness}}, w_{d_{chromaticness}}\}$ where $w_{d_{hue}} = 0.8$, $w_{d_{brightness}} = 0.1$, and $w_{d_{chromaticness}} = 0.1$. Clearly, $W_{\delta_{color}} = \{w_d \in \mathbb{R}_{\geq 0} : d \in \delta_{color}\}$ and $\sum_{w_d \in W_{\delta_{color}}} w_d = 1$, and so $W_{\delta_{color}}$ is a set of salience weights for δ_{color} . Consider the elements $x, y, z \in CS$ where:

	δ_{color}		
	d_{hue}	$d_{brightness}$	$d_{chromaticness}$
x	0.9	0.7	1
y	0.7	0.2	0.1
z	0.2	0.8	0.8

Table 2.1: Coordinates of x , y , and z in δ_{color}

We now compute the distance between x and y in domain δ_{color} given $W_{\delta_{color}}$. In the computation below, we use the notation $h = d_{hue}$, $b = d_{brightness}$, and $c = d_{chromaticness}$:

$$\begin{aligned}
d_{CS}^{\delta_{color}}(x, y, W_{\delta_{color}}) &= \sqrt{\sum_{d \in \delta_{color}} w_d \cdot |x_d - y_d|^2} \\
&= \sqrt{w_h \cdot |x_h - y_h|^2 + w_b \cdot |x_b - y_b|^2 + w_c \cdot |x_c - y_c|^2} \\
&= \sqrt{0.8 \cdot |0.9 - 0.7|^2 + 0.1 \cdot |0.7 - 0.2|^2 + 0.1 \cdot |1 - 0.1|^2} \\
&= \sqrt{0.032 + 0.025 + 0.081} \\
&= \sqrt{0.138} \\
&\approx 0.37
\end{aligned}$$

All pairwise distances in δ_{color} given $W_{\delta_{color}}$ are summarized in the table below:

The remaining distance computations are left to the reader.

Distances within domains, together with the notion of a *salience profile*, will allow us to define distances within a conceptual space. It is important to note that the name ‘salience profile’ is our own term for a mathematical object we have not come up with ourselves. However, as the specific object did not have a name, we wanted a term for it and found ‘salience profile’ appropriate. We now introduce the formal object we will call a salience profile. The formal object is defined by Bechberger

¹⁴While Gärdenfors does not fully explicate the above reasons in [Gär00], they seem implicit throughout. What he does note, however, is that the weighted Euclidean metric is the appropriate choice when dealing with distances for sets of integral dimensions [Gär00, pages 24–25].

¹⁵For the interpretation of the dimensions of δ_{color} , we refer back to the example of a conceptual space (Definition 3) in Section 2.1.

	x	y	z
x	0	0.37	0.63
y	0.37	0	0.53
z	0.63	0.53	0

Table 2.2: Pairwise distances in δ_{color} given $W_{\delta_{color}}$

and Kühnberger [BK17a, pages 2–3; BK17b, pages 2–3]. Given a set of domains Δ , a salience profile is a pair $(W_\Delta, \{W_\delta\}_{\delta \in \Delta})$ in which $W_\Delta = \{w_\delta \in \mathbb{R}_{\geq 0} : \delta \in \Delta\}$ is a set containing a salience weight w_δ for each domain $\delta \in \Delta$ with the constraint that $\sum_{\delta \in \Delta} w_\delta = |\Delta|$, where $|\Delta|$ denotes the cardinality of Δ , and $\{W_\delta\}_{\delta \in \Delta}$ is a family of sets of salience weights for each domain in Δ , i.e., for each $W_\gamma \in \{W_\delta\}_{\delta \in \Delta}$, $W_\gamma = \{w_d \in \mathbb{R}_{\geq 0} : d \in \gamma\}$ and $\sum_{w_d \in W_\gamma} w_d = 1$. Similar to salience weights for dimensions, salience weights for domains represent the relative importance of each domain in a given context [BK17a, page 60; BK17b, page 3]. Since a salience profile contains a salience weight for each domain *and* a set of salience weights for the dimensions in each of these domains, it jointly represents the relative importance of each individual domain, as well as the relative importance of each dimension within each domain. As an example of salience weights for domains, consider the set of domains $\Delta = \{\delta_{color}, \delta_{sound}\}$ in which δ_{color} and δ_{sound} are the previously defined domains for representing color and sound, respectively.¹⁶ In the context of identifying birds, both domains receive some salience since the color of a bird’s plumage, as well as the sound of their chirping, are used in identifying them. The relative weight of the domains will depend on the agent’s strengths, as well as the conditions under which the identification is done. If one’s vision is heavily obstructed due to trees, bushes, fog, or the like, sound receives higher salience. If one’s hearing is severely obstructed due to heavy wind, a roaring river, or the like, color receives higher salience. Having introduced the notion of a salience profile, we formally state the definition of distances within a conceptual space:

Definition 6 (Distance within a conceptual space ([BK17b, pages 2–3; BK17a, pages 59–60])). *Let CS be a conceptual space built over the set of domains Δ , $W = (W_\Delta, \{W_\delta\}_{\delta \in \Delta})$ a salience profile for Δ , and $x, y \in CS$ any points. Then the distance between x and y in CS given W is denoted by $d_{CS}^\Delta(x, y, W)$ and defined as the weighted Manhattan metric:*

$$d_{CS}^\Delta(x, y, W) = \sum_{\delta \in \Delta} w_\delta \cdot d_{CS}^\delta(x, y, W_\delta) = \sum_{\delta \in \Delta} w_\delta \cdot \sqrt{\sum_{d \in \delta} w_d \cdot |x_d - y_d|^2}$$

As an example of a full-space distance computation, consider the conceptual space CS defined over the set of domains $\Delta = \{\delta_{color}, \delta_{sound}\}$ from our previous example. Now take the salience profile $W = (W_\Delta, \{W_\delta\}_{\delta \in \Delta})$ where $W_\Delta = \{w_{\delta_{color}}, w_{\delta_{sound}}\}$ with $w_{\delta_{color}} = 1.5$ and $w_{\delta_{sound}} = 0.5$, $W_{\delta_{color}} = \{w_{d_{hue}}, w_{d_{brightness}}, w_{d_{chromaticness}}\}$ with $w_{d_{hue}} = 0.8$, $w_{d_{brightness}} = 0.1$, and $w_{d_{chromaticness}} = 0.1$, and $W_{\delta_{sound}} = \{w_{d_{pitch}}, w_{d_{loudness}}\}$ with $w_{d_{pitch}} = w_{d_{loudness}} = 0.5$. Consider the elements $x, y, z \in CS$ that have the following coordinates:

¹⁶For the interpretation of the dimensions of these domains, we refer back to the example of a conceptual space (Definition 3) in Section 2.1.

	δ_{color}			δ_{sound}	
	d_{hue}	$d_{brightness}$	$d_{chromaticness}$	d_{pitch}	$d_{loudness}$
x	0.9	0.7	1	0.9	0.2
y	0.7	0.2	0.1	0.6	0.9
z	0.2	0.8	0.8	0.3	0.5

Table 2.3: Coordinates of x , y , and z in CS

We now compute the distance between x and y in space CS given W . In the computation below, we use the notation $p = d_{pitch}$ and $l = d_{loudness}$. As we already computed the distance between x and y in δ_{color} given $W_{\delta_{color}}$ in our earlier example of an intra-domain distance computation, to obtain $d_{CS}^{\Delta}(x, y, W)$ it suffices to compute the distance between x and y in δ_{sound} given $W_{\delta_{sound}}$, and then take their weighted sum given by the salience weights in W_{Δ} . The distance between x and y in δ_{sound} given $W_{\delta_{sound}}$ is:

$$\begin{aligned}
d_{CS}^{\delta_{sound}}(x, y, W_{\delta_{sound}}) &= \sqrt{\sum_{d \in \delta_{sound}} w_d \cdot |x_d - y_d|^2} \\
&= \sqrt{w_p \cdot |x_p - y_p|^2 + w_l \cdot |x_l - y_l|^2} \\
&= \sqrt{0.5 \cdot |0.9 - 0.6|^2 + 0.5 \cdot |0.2 - 0.9|^2} \\
&= \sqrt{0.045 + 0.245} \\
&= \sqrt{0.29} \\
&\approx 0.54
\end{aligned}$$

As such, the distance between x and y in CS given W is:

$$\begin{aligned}
d_{CS}^{\Delta}(x, y, W) &= \sum_{\delta \in \Delta} w_{\delta} \cdot d_{CS}^{\delta}(x, y, W_{\delta}) \\
&= w_{\delta_{color}} \cdot d_{CS}^{\delta_{color}}(x, y, W_{\delta_{color}}) + w_{\delta_{sound}} \cdot d_{CS}^{\delta_{sound}}(x, y, W_{\delta_{sound}}) \\
&= 1.5 \cdot \sqrt{0.138} + 0.5 \cdot \sqrt{0.29} \\
&\approx 1.5 \cdot 0.37 + 0.5 \cdot 0.54 \\
&\approx 0.83
\end{aligned}$$

This concludes the part of this section that covers distances. We now take stock. Following the conceptual spaces literature, we have introduced separate measures of distance for dimensions, domains, and conceptual spaces, and motivated their intuitive appeal. The distance between elements of a conceptual space along a dimension is measured as their absolute difference. The distance between elements of a conceptual space within a domain is measured by the weighted Euclidean distance between the dimensions of the domain relative to a set of salience weights for the given domain. The distance between elements of a conceptual space is measured by the weighted Manhattan metric relative to a salience profile. We now present the notion of similarity we will use in constructing our logic.

Start by noting that elements of a conceptual space can be similar in several respects. They can be

similar with respect to a single dimension, a domain, and the space as a whole. Recall the intuition that distances in a conceptual space encode (dis)similarities in that the closer two points are, the more similar they are. The further apart two points are, the more dissimilar they are. With our three notions of distance at hand, we can formally define similarity along dimensions, domains, and spaces as a whole. But, for the purposes of the logic we will be developing, the only notion of similarity we need for our framework is that defined with respect to full space distances. Thus, we do not give a treatment of similarities along dimensions or similarities within domains. As mentioned at the beginning of this Section, it is standard within the conceptual spaces framework to model similarity as an exponentially decreasing function of distance. We do the same, and use the definition of Bechberger and Kühnberger [BK17b, pages 2–3; BK17a, pages 59–60].¹⁷ Bechberger and Kühnberger state their definition for an arbitrary metric and do not cast it in an explicit function form; we cast it in explicit function form and instantiate this metric with the conceptual-space distance d_{CS}^Δ :

Definition 7 (Similarity in a conceptual space (adapted from [BK17b, pages 2–3; BK17a, pages 59–60])). *Let CS be a conceptual space built over the set of domains Δ , W a salience profile, and $\lambda \in \mathbb{R}_{>0}$ a sensitivity parameter. The similarity function on CS given W and λ is denoted by Sim and defined as the function:*

$$Sim : CS \times CS \rightarrow (0, 1]$$

where for all $(x, y) \in CS \times CS$, $(x, y) \mapsto e^{-\lambda \cdot d_{CS}^\Delta(x, y, W)}$. Where $x, y \in CS$ are points, the expression ‘ $Sim(x, y)$ ’ denotes the similarity between x and y , and so sentences of the form ‘ $Sim(x, y) = r$ ’ reads ‘the similarity between x and y is r ’.

As an example, suppose we have a conceptual space CS built on the set of domains Δ , a salience profile W on Δ , and set the sensitivity parameter as $\lambda = 2$. Now take the elements $x, y \in CS$ and suppose that $d_{CS}^\Delta(x, y, W) = 0.95$. Then, the similarity between x and y is:

$$\begin{aligned} Sim(x, y) &= e^{-\lambda \cdot d_{CS}^\Delta(x, y, W)} \\ &= e^{-2 \cdot 0.95} \\ &= e^{-1.9} \\ &\approx 0.15 \end{aligned}$$

Having defined our measures of distance and shown how they are used to define similarities, we move on to using our notions of distance to build toward representations of properties and concepts using the conceptual spaces framework. We also extend the notion of similarity between points to similarity between properties and concepts.

¹⁷This is one of the functions that Gärdenfors presents in [Gär00, page 20] as a way of modeling similarity between points. Gärdenfors also presents the Gaussian function $e^{-c \cdot d(x, y)^2}$ as an alternative [Gär00, page 20]. He does not go into a debate as to which function is most appropriate, and we will not either. The reason is that this is an empirical question. However, the important thing is that similarity drops quickly when the distance between two elements of a space is relatively small, and drops much more slowly when the distance is relatively large [Gär00, page 21]. And, both choices yield the correct result for our purposes.

2.3 Properties, concepts, and prototypes

A fundamental aspect of our cognition is the abilities to assign properties to things and make categorizations. We may assign the property of being green to an apple, the property of being round to a tennis ball, and the property of being sour to a lemon. And, we use the concept of a dog to categorize a class of animals, the concept of a banana to categorize a class of fruits, and the concept of being tasty to categorize a class of foods. These abilities are crucial for us to identify commonalities in distinct things in a given context, as well as invariance across contexts. In the conceptual spaces framework, properties and concepts are distinguished by treating properties as a special kind of concept [Gär00, page 60]. The main distinguishing feature is that properties are characterized with respect to a single domain, while concepts are characterized with respect to several domains [Gär00, page 101]. Gärdenfors claims that the distinction between properties and concepts is grammatically expressed in the distinction between adjectives and nouns [Gär00, page 60]. In particular, adjectives typically represent properties, whereas (common) nouns typically represent concepts [Gär00, page 60]. For example, consider the adjective ‘heavy’. The content of ‘heavy’ can be characterized along the domain consisting of the dimension of weight.¹⁸ The exact region along the weight dimension that its content is represented as depends on the agent in question, as well as the context. Borrowing an example from Gärdenfors [Gär00, pages 102–105], consider the content of the common noun ‘apple’. This does not seem to be characterizable using a single domain. The reason is that the jointly separable domains of shape, taste, nutritional value, among others, are all involved in the concept of an apple (see Gärdenfors’ representation of an apple [Gär00, pages 102–103]). Besides characterizing concepts with respect to several domains, Gärdenfors identifies two further components desirable in a framework for representing concepts [Gär00, pages 102–105]. Firstly, a set of salience weights for each domain involved in characterizing the concept that encodes the contextual part of how concept meaning can change [Gär00, pages 103–104]. Secondly, information about how the domains over which the concept is defined are correlated in characterizing that concept [Gär00, pages 104–105]. Having already defined salience weights, the first component is taken care of. While Gärdenfors formally introduces salience weights [Gär00, pages 18–20, 103–104], he does not rigorously lay out means of capturing the correlations between domains. As this is not paramount to our intended application of the conceptual spaces framework, we leave a full treatment for future work, albeit it will be briefly touched upon in the coming Section. Having explained how properties and concepts are understood within the conceptual space framework, we present different ways to model regions of a conceptual space intended to represent properties or concepts. There are multiple ways of construing the regions representing properties and concepts in the domains in which they are characterized. But, what matters for the logic we will be constructing is that properties and concepts have a prototype for the agent in question, as we will use prototypes in defining similarities between contents. In turn, similarities between contents will play a pivotal role in our framework. A prototype of a concept or property for an agent is a representative object of that concept or property, for that agent [Gär00, page 84].¹⁹ The intuition behind the notion of a prototype is that certain members of a concept are more representative of that concept than others [Gär00, page 84]. Using an example from Gärdenfors [Gär00, page 84], consider the concept of a bird. Intuitively, a robin is more representative of the concept of a bird than a penguin or an emu

¹⁸In this case, a singleton domain is even enough.

¹⁹The notion of a prototype is not restricted to categories of objects instantiating properties and concepts, but applies to categories of objects in general [Gär00, page 84]. Our focus here, however, is on prototypes of concepts and properties.

[Gär00, p. 84], presumably because, among other things, a robin can fly as opposed to penguins or emus — and birds can typically fly. In this sense, a robin can be called a prototypical bird. Now, as the particular structure of properties and concepts is not important for the logic we are developing, we model properties and concepts as subsets of a conceptual space.²⁰ While we impose no structural constraints on the sets representing properties and concepts, we note that two of the main ways of imposing constraints on these sets in the conceptual spaces literature provide a natural way of obtaining a prototype for that property or concept. These constraints are motivated independently of their ability to obtain prototypes. There are also ways of obtaining a prototype that do not require such constraints. In the rest of this Section, we present four ways to obtain a prototype for a property or concept. Two of these ways are based on the structure of the set representing the property or concept in question. The other two rely on different information about the agent in question. Lastly, we present two ways of defining regions of a conceptual space to represent properties or concepts from a prototype.

The first means of obtaining a prototype for a property or concept relies on representing that property or concept by means of a convex set of the underlying space [Gär00, pages 86–92]. For some of the arguments in favor of this representation, as well as a discussion of some of the empirical evidence in favor of it, we refer the reader to Chapters 3 and 4 of [Gär00]. For our formal definitions below, we rely on the work of Bechberger and Kühnberger [BK17b; BK17a]. We now define the notion of convexity and show how a prototype can be computed from a convex set intended to represent a property or concept. In defining convexity, we first need to define betweenness. As we already have well-defined measures of distances, defining betweenness is straightforward:

Definition 8 (Betweenness ([BK17b, page 3; BK17a, page 60])). *Let d be any metric on a set A . Then the betweenness relation given d is denoted by B_d and defined, for any $x, y, z \in A$, as:*

$$B_d(x, y, z) \text{ if and only if } d(x, y) + d(y, z) = d(x, z).$$

where for any $x, y, z \in A$, $B_d(x, y, z)$ reads ‘ y is between x and z given metric d ’.

Intuitively, a point y is between points x and z if we can draw line segment from point x to point z passing through y . Having defined the betweenness relation, we introduce the definition of convexity:

Definition 9 (Convexity ([BK17b, page 3; BK17a, page 60])). *Let A be any set and d any metric on A . Then a set $S \subseteq A$ is convex under d if and only if for any elements $x, z \in S$ and any element $y \in A$, if $B_d(x, y, z)$ then $y \in S$.*

Intuitively, a set is convex if there are no gaps between any two members of that set. Note that in a convex set, positions can be described as being more or less central [Gär00, pages 86–87]. And, as such, the central point (i.e., the centroid) of a convex region representing a concept can be taken to correspond to the most prototypical member of that concept [Gär00, page 87]. The centroid of a convex region is obtained by taking, for each quality dimension, the mean of the values that points in the region take along that dimension. In this way, the centroid captures, in a single point, how the concept is distributed in the underlying conceptual space, without privileging any particular instance of that concept. This makes it a good candidate for a prototype. The examples considered in this thesis only

²⁰This is similar to Porello and Guizzardi [PG17] and Masolo and Porello [PM16], which also do not put many constraints on the representations of properties and concepts. In particular, they allow that properties and concepts can be represented as subsets of a conceptual space, but also that they can be represented as so-called *rough sets* of an underlying conceptual space (see [PG17, pages 4–5; PM16, page 7]).

take centroids of convex regions that can be written as products of bounded intervals, with each such interval corresponding to the range of values that instances of the concept can take along the relevant dimension.²¹ In such cases, the coordinate-wise mean can be obtained by taking the midpoint of each bounded interval. More explicitly, suppose that we have a conceptual space $CS = d_1 \times \dots \times d_n$ and a convex set $C \subseteq CS$ that can be written as $C = I_1 \times \dots \times I_n$, where each I_i is a bounded interval giving the range of values that instances of the concept can take along dimension d_i , with lower endpoint a_i and upper endpoint b_i . Then, for each $i \in \{1, \dots, n\}$, the i -th coordinate p_i of the centroid p of C is given by:

$$p_i = \frac{a_i + b_i}{2}.$$

As an example, suppose that we want to represent the concept of an orange as a convex set of a conceptual space and compute its prototype in this way. We use a simplified conceptual space for representing fruits presented by Bechberger and Kühnberger [BK17b, page 10], as well as their representation of the concept of an orange in that space [BK17b, page 11]. In particular, they define the conceptual space CS built on the set of domains $\Delta = \{\delta_{color}, \delta_{shape}, \delta_{taste}\}$ where [BK17b, page 10]:

- $\delta_{color} = \{d_{hue}\}$ is a domain for representing colors;
- $\delta_{shape} = \{d_{round}\}$ is a domain for representing shapes;
- $\delta_{taste} = \{d_{sweet}\}$ is a domain for representing taste,

With the dimensions interpreted as follows [BK17b, page 11]:

- $d_{hue} = [0, 1]$ represents the hue of objects, ranging from red, indicated by a value of 0.00, to purple, indicated by a value of 1.00;
- $d_{round} = [0, 1]$ represents the roundness of objects, measured by the percentage of the bounding sphere of an object that is filled by that object;
- $d_{sweet} = [0, 1]$ represents the relative sugar content of an object, ranging from zero sugar, indicated by a value of 0.00, to high levels of sugar, indicated by a value of 1.00.

Bechberger and Kühnberger [BK17b, p. 11] then define the convex set involved in their representation of the concept of an orange as $O = \{x \in CS : p^- \leq x \leq p^+\}$ where $p^- = (0.1, 0.9, 0.6)$ and $p^+ = (0.2, 1.0, 0.7)$.²² That is, O contains all points $x \in CS$ such that $0.1 \leq x_{d_{hue}} \leq 0.2$, $0.9 \leq x_{d_{round}} \leq 1.0$, and $0.6 \leq x_{d_{sweet}} \leq 0.7$. To see that O is convex, let $x, z \in O$ and suppose that y lies between them. Then, by the definition of betweenness, the value of y along each dimension lies between the corresponding values of x and z . Since x and z satisfy the bounds along each dimension, so does y , and hence $y \in O$. We now compute the prototype p of O . By the definition of the centroid of a convex set given above, each coordinate is the mean of the values that elements in O take along the corresponding dimension. Notice that O occupies an interval along each dimension. In particular, O occupies the intervals $[0.1, 0.2] \subseteq d_{hue}$, $[0.9, 1] \subseteq d_{round}$, and $[0.6, 0.7] \subseteq d_{sweet}$. Thus, as described above, we can obtain the mean of the values that O occupies along each dimension by the midpoints $(a + b)/2$ of the intervals:

²¹Whether the intervals are open, closed, or half-open does not affect the midpoint computation. What matters for the simple formulation used here is that each interval has finite lower and upper endpoints, i.e., that it is bounded.

²²We note that in Bechberger and Kühnberger [BK17b, page 11] representation of the concept of an orange, they use further components, such as salience weights. We omit these as they do not bear on the present example.

Dimension	Interval		Midpoint
	a_i	b_i	p_i
d_{hue}	0.1	0.2	0.15
d_{round}	0.9	1.0	0.95
d_{sweet}	0.6	0.7	0.65

Table 2.4: Midpoint p_i of each dimension’s interval, giving the prototype p of O in $CS = d_{hue} \times d_{round} \times d_{sweet}$.

Reading off Table 2.4, the prototype of the concept of an orange is $p = (0.15, 0.95, 0.65)$, which indicates, based on the interpretations of d_{hue} , d_{round} , and d_{sweet} , that a prototypical orange has an orange-red hue, is nearly perfectly round, and has a moderately high sugar content. The second means of obtaining a prototype for a property or concept relies on representing that property or concept as a so-called star-shaped set of the underlying space.²³ Bechberger and Kühnberger [BK17a] take star-shaped sets as a better alternative to convex sets for representing concepts, claiming that they allow for the encoding of correlations between the domains over which the concept is defined. We refer the reader to [BK17a, pages 61–62] for their argument in favor of this claim. For our formal treatment of star-shaped sets, we rely on the work of Bechberger and Kühnberger [BK17b; BK17a]. Before we can present their construction of so-called simple star-shaped sets [BK17b, page 4; BK17a, page 63], we need to introduce their definitions of star-shapedness and an axis-parallel cuboid. For star-shapedness, we use the definitions from Bechberger and Kühnberger [BK17b; BK17a]. Starting with star-shapedness:

Definition 10 (Star-shapedness ([BK17b, page 3; BK17a, page 60])). *Let A be any set and d any metric on A . Then a set $S \subseteq A$ is star-shaped under d with respect to a set $P \subseteq S$ if and only if for any $x \in P$, any element $y \in A$, and any $z \in S$, if $B_d(x, y, z)$ then $y \in S$.*

Intuitively, a set S is star-shaped with respect to a subset $P \subseteq S$ if and only if S includes any points lying between P and any point in S that we can “see” from P . It is important to note that a star-shaped set is not necessarily convex [Gär00, pages 68–69]. Moving on to the definition of an axis-parallel cuboid:

Definition 11 (Axis-parallel cuboid ([BK17b, page 4; BK17a, page 62])). *Let CS be a conceptual space built on the set of domains Δ , $\Delta_C \subseteq \Delta$ a subset of domains, and $p^-, p^+ \in CS$ any points. Then the set*

$$C = \{x \in CS : \forall \delta \in \Delta_C, \forall d \in \delta (p_d^- \leq x_d \leq p_d^+)\}$$

is an axis-parallel cuboid.

Here is how to understand axis-parallel cuboids. Suppose we have an axis-parallel cuboid C defined with respect to $\Delta_C \subseteq \Delta$ and the points p^- and p^+ , for some underlying conceptual space CS built on the set of domains Δ . p^- and p^+ give the upper and lower bounds of each dimension $d \in \delta$ in every domain $\delta \in \Delta_C$. In particular, for each domain $\delta \in \Delta_C$ and each dimension $d \in \delta$ of that domain, every

²³For the basics of using star-shaped sets within the conceptual spaces framework, we refer the reader to Chapters 3 and 4 of [Gär00]. For a more mathematical treatment of star-shaped sets within the framework, we refer the reader to [BK17b; BK17a].

point in C must have its d -coordinate in the closed interval $[p_d^-, p_d^+]$. Notice that for any dimension $d \in \delta$ of any domain $\delta \in \Delta \setminus \Delta_C$ that is not in Δ_C , C extends across the full dimension d . The intuition is that when axis-parallel cuboids are used in representing properties or concepts, the only dimensions that matter for membership are the ones contained in the domains over which it is defined, which is given by Δ_C . An important thing noted by Bechberger and Kühnberger [BK17b, page 4; BK17a, page 63], without proof, is that axis-parallel cuboids are convex under the full space metric d_{CS}^Δ .²⁴

Proposition 1 ([BK17b, page 4]). *Let CS be a conceptual space built on the set of domains Δ , W a salience profile, and C an axis-parallel cuboid defined on CS . Then C is convex under d_{CS}^Δ with respect to W .*

We now present Bechberger and Kühnberger’s [BK17b; BK17a] definition of a simple star-shaped set:

Definition 12 (Simple star-shaped set ([BK17b, page 4; BK17a, page 63])). *Let CS be a conceptual space built on the set of domains Δ , $(\Delta_C, \{C_1, \dots, C_n\})$ a pair where $\Delta_C \subseteq \Delta$ is a set of domains and each $C_i \in \{C_1, \dots, C_n\}$ is an axis-parallel cuboid defined with respect to Δ_C such that the central region $P = \bigcap_{i=1}^n C_i$ is non-empty. Then the set*

$$S = \bigcup_{i=1}^n C_i$$

is a simple star-shaped set.

Bechberger and Kühnberger [BK17b, page 4] further note that any union of convex sets with a non-empty intersection is star-shaped, citing Smith [Smi68]. While Bechberger and Kühnberger [BK17b] do not explicitly state that this implies that a simple star-shaped set is, in fact, star-shaped with respect to its central region, we credit this Proposition to them as it is what warrants calling a simple star-shaped set ‘star-shaped’ in the first place, and since the claim follows trivially from their Definition of a simple star-shaped set [BK17b, page 4; BK17a, page 63] that we have employed above. Furthermore, they have proved the corresponding result for fuzzy simple star-shaped sets in Bechberger and Kühnberger [BK17a, page 65].

Proposition 2 (Implicit in [BK17b, page 4]). *Let CS be a conceptual space built on the set of domains Δ and $S = \bigcup_{i=1}^n C_i$ a simple star-shaped set defined with respect to the pair $(\Delta_C, \{C_1, \dots, C_n\})$. Then S is star-shaped with respect to its central region P .*

To derive a prototype from a simple star-shaped set representing a property or concept, we take the centroid of its central region. Bechberger and Kühnberger [BK17b, page 10] already proceed in essentially this way, applying their similarity and betweenness measures to the midpoint of a concept’s central region. We simply make this fact explicit, as our use of the centroid as a prototype depends on it.

Proposition 3. *Let CS be a conceptual space built on the set of domains Δ , $S \subseteq CS$ a simple star-shaped set based on the pair $(\Delta_C, \{C_1, \dots, C_n\})$, and W a salience profile. Then the central region $P = \bigcap_{i=1}^n C_i$ of S is convex under d_{CS}^Δ given W .*

²⁴Bechberger and Kühnberger [BK17a, page 63] has an intuitive explanation.

Proof. Take an arbitrary conceptual space CS built on a set of domains Δ , any salience profile W , and consider a simple star-shaped set $S \subseteq CS$ based on the pair $(\Delta_C, \{C_1, \dots, C_n\})$. To show that the central region $P = \bigcap_{i=1}^n C_i$ of S is convex under d_{CS}^Δ given W , take arbitrary elements $x, z \in P$ and $y \in CS$, and suppose that $B_{d_{CS}^\Delta}(x, y, z)$. We now need to show that $y \in P$. Since $P = \bigcap_{i=1}^n C_i$, it suffices to show that $y \in C_j$ for each $C_j \in \{C_1, \dots, C_n\}$. So take an arbitrary axis-parallel cuboid $C_j \in \{C_1, \dots, C_n\}$. By Proposition 1, C_j is convex. Furthermore, $x, z \in C_j$ since $P = \bigcap_{i=1}^n C_i \subseteq C_j$ and $x, z \in P$. It follows that $y \in C_j$ by definition of convexity given our assumption that $B_{d_{CS}^\Delta}(x, y, z)$. As $C_j \in \{C_1, \dots, C_n\}$ was arbitrary, we conclude that $y \in P$. By definition of convexity, we conclude that P is convex under d_{CS}^Δ given W . $\square$

Having shown that the central region of a simple star-shaped set is convex, we can compute its centroid as explained above. Then, given a simple star-shaped set, we can take the centroid of its central region to represent the prototype of the property or concept modeled by that set.

We now give an example of a concept modeled as a simple star-shaped set and compute the centroid of its central region. In particular, we model the concept of a banana. When a banana goes through its ripening process, its properties correlate in a predictable manner, i.e., its color, firmness, and sweetness. The way these properties of a banana correlate makes certain combinations of its properties inadmissible. For example, a firm, brown, and non-sweet banana, or a soft, green, and sweet banana, are inadmissible combinations. Now, if one represents a banana as a convex set, it is impossible to avoid including these inadmissible combinations in case we want to represent the vast variation between bananas in their ripening process. Modeling a banana as a simple star-shaped set can allow us to exclude these inadmissible combinations, as shown below. Take the conceptual space CS built on the set of domains $\Delta = \{\delta_{\text{color}}, \delta_{\text{texture}}, \delta_{\text{taste}}\}$, where:

- $\delta_{\text{color}} = \{d_{\text{banana-color}}\}$ is a domain for representing color;
- $\delta_{\text{texture}} = \{d_{\text{firm}}\}$ is a domain for representing texture;
- $\delta_{\text{taste}} = \{d_{\text{sweet}}\}$ is a domain for representing taste.

With the dimensions interpreted as follows:

- $d_{\text{banana-color}} = [0, 1]$ is a specialized dimension for representing the color of bananas, ranging from green, indicated by a value of 0.00, to yellow, indicated by a value of 0.50, to brown, indicated by a value of 1.00.
- $d_{\text{firm}} = [0, 1]$ represents the firmness of an object, ranging from very soft, indicated by a value of 0.00, to very firm, indicated by a value of 1.00.
- $d_{\text{sweet}} = [0, 1]$ represents the relative sugar content of an object, ranging from zero sugar, indicated by a value of 0.00, to high levels of sugar, indicated by a value of 1.00.

We now use $CS = d_{\text{banana-color}} \times d_{\text{firm}} \times d_{\text{sweet}}$ to define two axis-parallel cuboids, denoted by C_1 and C_2 , that each represent the properties of bananas in two different ripening periods. We will define both axis-parallel cuboids for the entire set of domains Δ . C_1 will represent bananas that are under-ripe to ripe and C_2 will represent bananas that are ripe to over-ripe. Starting with C_1 , take the points $p_1^- = (0, 0.45, 0)$ and $p_1^+ = (0.55, 1, 0.6)$ of CS . By Definition 11, $C_1 = \{x \in CS : p_1^- \leq x \leq p_1^+\}$ is an axis-parallel cuboid. Equivalently, we can represent C_1 as the product of intervals

$C_1 = [0, 0.55] \times [0.45, 1] \times [0, 0.6]$. Given the interpretations of the dimensions of CS , C_1 represent bananas whose color ranges from green to yellow, whose firmness ranges from moderately firm to very firm, and whose sweetness ranges from non-sweet to moderately sweet. I.e., under-ripe to ripe bananas. Moving on to cuboid C_2 , take the points $p_2^- = (0.5, 0, 0.45)$ and $p_2^+ = (1, 0.55, 1)$ of CS . By Definition 11, $C_2 = \{x \in CS : p_2^- \leq x \leq p_2^+\}$ is an axis-parallel cuboid. Equivalently, we can represent C_2 as the product of intervals $C_2 = [0.5, 1] \times [0, 0.55] \times [0.45, 1]$. Thus, C_2 represent bananas whose color ranges from yellow to brown, whose firmness ranges from moderately soft to very soft, and whose sweetness ranges from moderately sweet to very sweet. I.e., ripe to over-ripe bananas. We can now represent the concept of a banana as the union of cuboids C_1 and C_2 , $C_1 \cup C_2$. To verify that $C_1 \cup C_2$ is a simple star-shaped set, we check that its central region $P = C_1 \cap C_2$ is non-empty. Since the cuboids C_1 and C_2 can be written as products of intervals, a point is in their intersection if it is in both of the corresponding products of intervals. We check interval by interval, starting with the intervals of C_1 and C_2 . For the color intervals representing color, we have that $[0, 0.55] \cap [0.5, 1] = [0.5, 0.55]$. For the intervals representing firmness, we have that $[0.45, 1] \cap [0, 0.55] = [0.45, 0.55]$. Lastly, for the intervals representing sweetness, we have that $[0, 0.6] \cap [0.45, 1] = [0.45, 0.6]$. This gives us that $P = C_1 \cap C_2 = [0.5, 0.55] \times [0.45, 0.55] \times [0.45, 0.6]$, which is non-empty. Thus, $C_1 \cup C_2$ is a simple star-shaped set by Definition 12. Notice that $C_1 \cup C_2$ excludes the aforementioned inadmissible combinations of properties. For example, an object that is brown, firm and non-sweet would be represented in CS as having a high value along $d_{banana-color}$ and d_{firm} , but a low value along d_{sweet} . And, such a point neither exists in C_1 or C_2 . Lastly, we compute the centroid of P . As we can write P as the product of intervals $P = [0.5, 0.55] \times [0.45, 0.55] \times [0.45, 0.6]$, its centroid is given by:

Dimension	Interval		Midpoint
	a_i	b_i	p_i
$d_{banana-color}$	0.5	0.55	0.525
d_{firm}	0.45	0.55	0.50
d_{sweet}	0.45	0.6	0.525

Table 2.5: Midpoint p_i of each dimension’s interval, giving the prototype p of $C_1 \cup C_2$ in $CS = d_{banana-color} \times d_{firm} \times d_{sweet}$.

Reading off 2.5, the prototype of the concept of a banana is $p = (0.525, 0.50, 0.525)$, which indicates, based on the interpretations of $d_{banana-color}$, d_{firm} , and d_{sweet} , that a prototypical banana is yellow, moderately firm, and moderately sweet.

A third option is to compute the prototype from an agent’s exemplars of a concept [Gär00, pages 123–126]. An exemplar is a particular instance of the concept that the agent has encountered [Gär00, pages 123–125]. The features the agent represents the exemplars as having are modeled as points in the agent’s conceptual space [Gär00, pages 123–125]. The computation of the prototype is then done by taking the coordinate-wise mean of the points representing these exemplars [Gär00, page 124]. Formally, the i -th coordinate p_i of the prototype p is given by

$$p_i = \frac{1}{n} \sum_{k=1}^n x_k^{(i)},$$

where $x_1, \dots, x_n$ are the n exemplars, $x_k^{(i)}$ denotes the i -th coordinate of the k -th exemplar, and $i \in D$

[Gär00, page 124].²⁵ In this sense, the prototype reflects the average of the concept across the instances the agent has been exposed to. As such, this option for obtaining a prototype is structurally analogous to the two earlier options in that they all derive the prototype as a coordinate-wise mean, the difference being that the first takes the mean over the entire set representing the property or concept, the second takes the mean over the central region of a simple star-shaped set representing that property or concept, and the third takes it over a finite number of exemplars. The first two options assume that we have a predefined region representing the concept, whereas this option does not. As an example, suppose an agent has encountered the three exemplar bananas x_1 , x_2 , and x_3 where $x_1 = (0.45, 0.70, 0.35)$, $x_2 = (0.55, 0.50, 0.60)$, and $x_3 = (0.75, 0.25, 0.85)$. The exemplar-based prototype given x_1, x_2, x_3 is given by:

Dimension	Exemplars			Mean
	x_1	x_2	x_3	p_i
$d_{banana-color}$	0.45	0.55	0.75	0.58
d_{firm}	0.70	0.50	0.25	0.48
d_{sweet}	0.35	0.60	0.85	0.60

Table 2.6: Coordinate-wise means p_i of the exemplars x_1, x_2 , and x_3 , giving the exemplar-based prototype p in $CS = d_{banana-color} \times d_{firm} \times d_{sweet}$.

The fourth method is to stipulate a prototype with reference to information about what the agent considers a typical object, or typical objects, instantiating that property or concept. For example, suppose we have an agent that takes the typical banana to be yellow, moderately firm, and moderately sweet. Using the banana space $CS = d_{banana-color} \times d_{firm} \times d_{sweet}$ we can stipulate their prototype as the point $p = (0.5, 0.5, 0.6)$. The advantage of this method is that it does not require a predefined region representing the property or concept, or a set of known exemplars. We will make use of this method in the examples analyzed in Chapter 3. Furthermore, we note that it is possible to combine the above methods. This may be necessary if the mean of a set representing a property or concept is undefined along some dimension, or if the modeler has only partial information about the agent’s representation of a given property or concept. Which combination of methods is appropriate will depend on the specific modeling task.

Lastly, when a prototype is stipulated or computed from exemplars, without some given set representing the property or concept in question, one may still want to define a set of the underlying conceptual space representing that property or concept. One way to define a region that can represent a concept given a prototype is to use a construction similar to the type of *prototype threshold* construction in Lewis and Lawry [LL16] and Lawry and Tang [LT09]. The basics of the construction are as follows [LL16, page 209]: Suppose we have some label L_i representing a concept for some underlying conceptual space CS and a set of prototypes $P_i \subseteq CS$ for L_i . Then, given some threshold $r \in \mathbb{R}_{\geq 0}$ indicating when something is sufficiently close to a prototype $p_i \in P_i$ to count as an instance of L_i , we can think of the set $\{x \in CS : d(x, P_i) \leq r\}$ as giving the extension of the concept labeled by L_i , where d is some distance metric between points of CS that is extended to handle sets of points by defining

²⁵Given that Gärdenfors restricts the computation to finitely many exemplars [Gär00, page 124], he tacitly presumes that we can only encounter finitely many. This squares with the intuitive idea that, at least in sensory perception, we can only encounter a finite number of objects.

$d(x, Y) = \min\{d(x, y) : y \in Y\}$.²⁶ Where we differ from Lewis and Lawry [LL16] is that we only consider the single-prototype case (i.e., the case where their set of prototypes P_i is a singleton $P_i = \{p_i\}$), and we make a distinction between what we call *space thresholds*, which is the threshold explained above from [LL16], and *dimension thresholds*. Starting with space thresholds. These are non-negative real numbers $r \in \mathbb{R}_{\geq 0}$. Using our representation an agent’s prototype $p \in CP$ and their space threshold $r \in \mathbb{R}_{\geq 0}$ for some content, we can define the convex region representing that content as the closed ball $B(p, r) = \{x \in CS : d(x, p) \leq r\}$, i.e., the set of all points within the distance threshold r from the prototype p . As mentioned above and noted in [LL16], a set like $B(p, r)$ intuitively represents all points of CS that are sufficiently similar to the prototype p for the agent in question to count as an instance of the content for which p is a prototype. Now, if we want some content to take up a large portion of a single quality dimension using a threshold construction, or if that content is simply defined with respect to few quality dimensions, we may need to use an individual distance threshold for each dimension of the space. The reason is that in order to capture the large deviance across a single quality dimension, a larger space threshold would need to be used. However, this risks allowing too much variation across other dimensions. Thus, the ball construction is most appropriate when the content one intends to represent is somewhat homogeneous across the individual dimensions. But, in the other cases, it can be useful to specify individual dimension thresholds r_i for each quality dimension d_i , and then define the convex region representing the given content as the set of all points $(q_1, \dots, q_n) \in CP$ such that $|q_i - p_i| \leq r_i$ for each $i \in \{1, \dots, n\}$, where $p = (p_1, \dots, p_n)$ represents the prototype for the relevant concept. Using this construction, we obtain an axis-parallel cuboid in the sense of Definition 11. In Section 3.2.1, we will make use of the dimension threshold construction. In Section 3.2.2, we will make use of the space threshold construction.

We have now presented a repertoire of tools from the conceptual spaces framework for representing properties and concepts and measuring their similarity using prototypes. We have also presented four different methods for obtaining prototypes, as well as how to define a set representing a property or concept using a prototype and a distance threshold or a range of dimension thresholds. In the following Section, we extend the conceptual spaces framework and use the above-mentioned tools to describe how processing a piece of content increases the accessibility of other content stored in memory.

2.4 Updating accessibility

In the previous Sections, we introduced the conceptual spaces framework. As described, we will use it to represent contents stored in memory. Specifically, we will model contents as subsets of an underlying conceptual space associated with the agent. What is important for the machinery we will be developing is the assumption that agents have a prototype for each piece of content they have stored in their LTM, for which we laid out four different ways of modeling and/or obtaining a prototype. We now embed the conceptual spaces framework into a model of how exposure to a stimulus increases the accessibility of contents stored in one’s LTM. The section will go as follows. First, we adopt a common distinction in the memory literature between *available* and *accessible* information [Gro20, p. 278], and introduce a function for capturing the accessibility levels of contents stored in an agent’s LTM. Secondly, we introduce an *increment function* for capturing how much processing a piece of content increases the accessibility of another. We then use this increment function, together with the accessibility function,

²⁶We note that Lewis and Lawry [LL16, page 209] take these thresholds to be drawn from some underlying probability distributions. As this is irrelevant for our present purposes, we refer the interested reader to [LL16].

to define a general notion of accessibility update. Finally, we specify this increment in terms of a notion of similarity between contents [BK17b, page 10], while leaving other possible specifications for future work.

Following a distinction drawn in memory research [Gro20, page 278], we distinguish between *available* and *accessible* information. Available information is information that is stored in memory. To utilize one’s stored information, it needs to be located and brought into an active state if not already in such a state.²⁷ The process of locating information in memory and bringing it into an active state is known as memory retrieval [Rei18, page 197; EK20, page 239; Gro20, page 277]. While availability of a piece of information is a necessary condition for retrieval of that piece of information, it is an insufficient condition for that same purpose [Gro20, page 278]. Accessible information is information that can be retrieved from memory [Gro20, page 278]. Accessibility can be interpreted as a binary notion or a graded notion. Under a binary interpretation, information is accessible if and only if it is retrievable, without reference to how easily that information can be retrieved. Under a graded interpretation, information with higher levels of accessibility are easier to retrieve than information with lower levels of accessibility. This ease of retrieval can be a function of many things, including recency of use, frequency of activation, and relevant and short retrieval paths. Importantly, contents that are more accessible than others are, all else being equal, more readily brought to bear on the agent’s ongoing cognitive processing. Thus, since agents’ actions are influenced by their cognitive processing, increased content accessibility can influence the actions agents take. We will formally capture accessibility by a function from contents to non-negative real numbers, indicating the degree of accessibility of each piece of content:

Definition 13 (Content accessibility function). *Let CS be a conceptual space and $\mathcal{C} \subseteq \mathcal{P}(CS)$ a set of contents. Then $\text{Acc} : \mathcal{C} \rightarrow \mathbb{R}_{\geq 0}$ is a content accessibility function indicating the accessibility of each content in $\mathcal{C}$. For any $C \in \mathcal{C}$, $\text{Acc}(C) \in \mathbb{R}_{\geq 0}$ denotes the accessibility of C and ‘ $\text{Acc}(C) = r$ ’ reads ‘the accessibility of C is r ’.*

For example, suppose that we are working with the set of contents $\mathcal{C} = \{\text{APPLE}, \text{LEMON}\}$ for an agent in which ‘APPLE’ represents the concept of an apple and ‘LEMON’ represents the concept of a lemon. Now take an accessibility function ‘Acc’ defined on $\mathcal{C}$ where $\text{Acc}(\text{APPLE}) = 0.5$ and $\text{Acc}(\text{LEMON}) = 0.75$. Since $\text{Acc}(\text{LEMON}) > \text{Acc}(\text{APPLE})$, we have that, all else being equal, the concept of a lemon will bear more on the agent’s ongoing processing than the concept of an apple.

The final ingredient we need in order to specify our general notion of accessibility update is a means of capturing how much processing a piece of content increases the accessibility of other content. For this purpose, we introduce the notion of an *increment function*:

Definition 14 (Increment function). *Let CS be a conceptual space and $\mathcal{C} \subseteq \mathcal{P}(CP)$ a set of contents. Then $\iota : \mathcal{C} \times \mathcal{C} \rightarrow \mathbb{R}_{\geq 0}$ is an increment function. For any $P, C \in \mathcal{C}$, we will use the notation $\iota_P(C)$ for $\iota(P, C)$.*

Where CS is a conceptual space, $\mathcal{C} \subseteq \mathcal{P}(CP)$ is a set of contents, and $\iota : \mathcal{C} \times \mathcal{C} \rightarrow \mathbb{R}_{\geq 0}$ an increment function, we will say that ι is an increment function on $\mathcal{C}$. Furthermore, for any contents $P, C \in \mathcal{C}$, sentences of the form ‘ $\iota_P(C) = x$ ’ read ‘processing P increases the accessibility of C by x ’.

We now define our general notion of accessibility update:

²⁷On many standard models of memory, this active state is identified with working memory (WM) (see the chapters on memory in [Gro20; Rei18; EK20]). There is some debate, however, as to whether retrieval can bypass WM entirely. We refer the reader to [MV25] for a discussion on this.

Definition 15 (Accessibility update). *Let CS be a conceptual space, $\mathcal{C} \subseteq \mathcal{P}(CS)$ a set of contents, and $P \in \mathcal{C}$ a piece of content. The accessibility update of any content $C \in \mathcal{C}$ induced by processing P is given by the equation:*

$$\text{Acc}_P(C) = \text{Acc}(C) + \iota_P(C),$$

where $\text{Acc}_P(C)$ denotes the accessibility of C after processing P , $\text{Acc}(C)$ denotes the accessibility of C before processing P , and ι is an increment function on $\mathcal{C}$ giving the accessibility increment of C induced by processing P .

In plain English, the equation reads as follows: The accessibility of a piece of content C upon processing content P is obtained by incrementing the prior accessibility of C by $\iota_P(C)$, where ‘prior accessibility of C ’ means the accessibility of C before processing the priming stimulus. Different specifications of the increment function will yield different operations of accessibility update, each isolating a potentially different underlying mechanism. Our chosen specification of the increment function for this thesis will be based on a notion of similarity between contents. Bechberger and Kühnberger [BK17b, page 10] define a “naive” notion of similarity between concepts by applying their Definition of similarity [BK17b, page 3] to the midpoint of central regions of the sets representing those concepts, for which they use simple star-shaped sets (refer back to 12). We adapt their definition of similarity between concepts to fit our own setting:

Definition 16 (Content similarity function (adapted from [BK17b, page 10])). *Let CS be a conceptual space, $\mathcal{C} \subseteq \mathcal{P}(CS)$ a set of contents, W a salience profile, and $\lambda \in \mathbb{R}_{>0}$ a sensitivity parameter. For each $C \in \mathcal{C}$, we denote the prototype of C by $p_C \in CS$ and assume it exists. The content similarity function is defined as:*

$$\text{Sim} : \mathcal{C} \times \mathcal{C} \rightarrow (0, 1]$$

where for all $(C, C') \in \mathcal{C} \times \mathcal{C}$, $(C, C') \mapsto \text{Sim}(p_C, p_{C'})$.

This choice of increment function is motivated by two considerations. Firstly, we can derive the similarity relations from the underlying conceptual space and set of contents under the assumption that each piece of content has a prototype. As such, we avoid arbitrarily having to stipulate the increment. Secondly, the notion of similarity is widely used and well-understood in the conceptual spaces literature.²⁸ We are not claiming that this is the correct way of specifying the increment, or that it provides the most natural way of understanding content priming phenomena. Rather, it is a tool in the toolbox we are developing that, as will be shown in our worked examples, delivers the correct verdicts.

Moving on to the formal definition of similarity-based accessibility update:

Definition 17 (Similarity-based accessibility update). *Let CS be a conceptual space built on the set of domains Δ , $\mathcal{C} \subseteq \mathcal{P}(CS)$ a set of contents, $\lambda \in \mathbb{R}_{>0}$ a sensitivity parameter for the similarity relation, and for each $C \in \mathcal{C}$, $p_C \in CS$ the prototype of C . The similarity-based accessibility update between P and C is given by the accessibility update of C induced by processing P where the increment function ι*

²⁸We refer the reader to Section 2.2 for our exposition on the notion of similarity in the conceptual spaces framework.

is the content similarity function:

$$\text{Acc}_P(C) = \text{Acc}(C) + \text{Sim}(P, C)$$

As mentioned earlier, there are other options for defining the accessibility increment that also utilize the underlying conceptual space. Here, any established relation between contents in a conceptual space could in principle be used. One option that we find particularly natural is to measure the degree of “fit” or “subthood” between two pieces of content as found in [BK17b; BK17a]. Measuring the degree of fit relies on representing contents as so-called *fuzzy* subsets of a conceptual space. A fuzzy set is a set that admits of a graded notion of membership. Given a conceptual space CP , fuzzy sets can be identified with a membership function $\mu : CP \rightarrow [0, 1]$, specifying the degree $\mu(x) \in [0, 1]$ to which an object with features $x \in CP$ is a member of the given piece of content. For example, suppose we wanted to represent the concept of a soccer ball. Presumably, we would define this concept in a conceptual space with dimensions representing roundness, surface thickness, and perhaps size. However, there may not be an exact cut-off for when some degree of roundness qualifies something as a ball. Instead of treating membership in the concept as all-or-nothing, we can use a membership function $\mu : CP \rightarrow [0, 1]$ to represent degrees of membership. This would then allow us to define an increment function in terms of the degree to which one content fits, overlaps with, or is included in another. For example, processing a content P might increase the accessibility of a content C in proportion to the degree to which P is contained in, or fits with, C . Another option is to embed similarity within a richer representation of concepts.

Chapter 3

Examples

In this Chapter, we do two things. Firstly, in Section 3.1, we specify a model of accessibility-based action selection that connects accessibility to action selection, thus allowing us to represent how the processing of certain content can affect behavior by increasing the accessibility of said content. Secondly, in Section 3.2, we show how the tools of conceptual spaces, similarity-based accessibility update, and the soon-to-be introduced action-selection mechanism can be used to represent and reason about content priming effects. In the presentation and analysis of our first example, we examine in depth several modeling decisions that warrant particular attention. Having explained these in detail for the first example, we present the other example and our analysis of it succinctly.

3.1 Accessibility-based action selection

Having presented our model of how processing a piece of content increases the accessibility of other content stored in an agent’s LTM, we now turn to modeling how accessibility levels of contents can translate into performing certain actions over others.

We start by noting that not every piece of content bears on the selection of every action [JW14, pages 97–98]. Presumably, the accessibility levels of the concept of a unicorn do not influence the choice of an agent deciding to buy an apple over an orange at the farmers’ market. On the contrary, the accessibility levels of the concept of an apple pie may bear on an agent deciding to buy an apple over an orange. Or, the accessibility levels of the concept of being angry may bear on an agent deciding to fill in the word fragment ‘YE_ _’ as ‘YELL’ instead of ‘YEAR’. The guiding intuition with these examples is that not every piece of content in an agent’s memory bears equally on every action [JW14, pages 97–98]. As Janiszewski and Wyer [JW14, pages 97–98] stress in their review on content priming, an increase in content accessibility does not by itself produce a behavioral response. Content must also be relevant to the given response [JW14, pages 97–98]. To connect the contents stored in an agent’s memory to the selection of actions, we introduce the notion of a *relevance function*. We will model relevance as a graded notion, allowing contents to be more or less relevant for different actions. In particular, a relevance function will map content-action pairs to a real number in the unit interval, indicating the degree to which that content is relevant to the selection of that action. A value of 0 means that the content is irrelevant to the selection of that action, whereas a value of 1 indicates that the content is fully relevant to the selection of that action. The formal definition of a relevance function is as follows:

Definition 18 (Relevance function). *Let $\mathcal{C}$ be a non-empty set of contents and $\mathcal{A}$ a non-empty set of actions. Then $\text{Rel} : \mathcal{A} \times \mathcal{C} \rightarrow \mathbb{R}_{\geq 0}$ is a relevance function, specifying for each piece of content $C \in \mathcal{C}$ and each action $\alpha \in \mathcal{A}$ the degree $\text{Rel}(\alpha, C) \in [0, 1]$ to which C is relevant for the selection of action α*

With the notion of relevance in place, we now introduce a measure of how much support an agent’s current cognitive state lends to the selection of each candidate action. In our model, an agent’s cognitive state is given by the contents stored in their memory and their accessibility levels. We define the measure of support for an action, given an agent’s current cognitive state, as the sum of the accessibility levels of the contents stored in their memory weighted by their relevance to the selection of that action. We will call this measure a *selection score*.¹ The intuition behind selection scores is as follows. Suppose that two pieces of content are equally relevant to the selection of two different actions, but one piece of content has a higher level of accessibility than the other. All else being equal, the action for which the more accessible piece of content is relevant receives greater overall support, since more accessible content bears more on cognitive processing, and so also on the cognitive process consisting of selecting an action to perform. This intuition extends to cases where multiple pieces of content bear on the selection of a response. We stress that the selection score is a deliberate abstraction in that it isolates the contribution of concept accessibility and relevance to action selection, setting aside all other factors that may influence an agent’s behavior. Formally, we define the selection score of an action as follows:

Definition 19 (Selection score). *Let CS be a conceptual space, $\mathcal{C} \subseteq CS$ a set of contents, $\mathcal{A}$ a set of actions, $\text{Acc} : \mathcal{C} \rightarrow \mathbb{R}_{\geq 0}$ a content accessibility function, and $\text{Rel} : \mathcal{A} \times \mathcal{C} \rightarrow [0, 1]$ a relevance function. The selection score of an action $\alpha \in \mathcal{A}$ is denoted by ‘ $\text{score}(\alpha)$ ’ and defined as:*

$$\text{score}(\alpha) := \sum_{C \in \mathcal{C}} \text{Acc}(C) \cdot \text{Rel}(\alpha, C)$$

Thus, each piece of content contributes to the score of an action in proportion to both how accessible it currently is and how relevant it is to the selection of that action. A piece of content that is highly accessible but irrelevant to a given response contributes nothing to its score. Conversely, a relevant but inaccessible piece of content contributes nothing either. Only content that is both accessible and relevant drives the score upward.

We now use selection scores to model accessibility-based action selection. The basic idea is that, in different situations, agents face a set of candidate actions that “compete” to be selected. If no candidate action comes out on top, the agent instead performs a designated fallback action α_0 . The fallback action can be interpreted as a degenerate “null” action, indicating that nothing happens.² But, it may also be some more substantive action, such as the agent’s typical or habitual response in the given situation. In formally fleshing out the notion of candidate actions “competing” to be selected, we adopt the following picture. A candidate action is selected if and only if its support score is strictly greater than the support score of every other candidate action. If no candidate action satisfies this condition, the agent performs their fallback action. We will call the condition that a candidate action has a strictly greater support score than every other candidate action its *executability condition*. And, we will take the executability condition of the fallback action as the condition that no

¹Treating selection scores as a weighted sum is an idealization, since, supposedly, the contributions of multiple contents may interact in more complex, non-additive ways.

²The interpretation of the fallback action as a “null” action is inspired by how announcements of $\top$, representing some tautology, is handled in the Dynamic Epistemic Logic tradition [BR16].

candidate action has its executability condition satisfied. Thus, we can state the picture as follows: a candidate action is selected if and only if its executability condition is satisfied, and the fallback action is selected if and only if no candidate action satisfies its executability condition. Suppose we have a set of actions $\mathcal{A} \cup \{\alpha_0\}$ with $\mathcal{A}$ the candidate actions and α_0 the null action. We then define the executability condition for each candidate action $\alpha_i \in \mathcal{A}$ as follows, where ‘ $\neg$ ’ denotes truth-functional negation:

$$\text{Exe}(\alpha_i) := \forall a_j \in \mathcal{A} \setminus \{\alpha_i\}, \text{score}(a_i) > \text{score}(a_j),$$

and the executability condition of the fallback action α_0 as:

$$\text{Exe}(\alpha_0) := \forall \alpha \in \mathcal{A}, \neg \text{Exe}(\alpha).$$

The idea is then that when an agent is faced with a set of actions, modeled as consisting of a set of candidate actions and a single fallback action, they perform whichever one’s executability condition is satisfied. We note that given the above construction, it is guaranteed that a unique executability condition will be satisfied. When implementing executability conditions for our logic in Section 4.1, we prove this property of executability conditions (see Propositions 7 and 8).

3.2 Canonical examples

We now show how the tools of conceptual spaces, similarity-based accessibility update, and the just introduced action-selection mechanism can be used to model content priming effects. We will structure our presentation and analysis of the examples as follows. Firstly, we give Kahneman’s description of the relevant priming effect as they occur on [Kah11, pages 50–54]. Then, we cast these examples in a standard Dynamic Epistemic Logic fashion and construe them as involving a particular agent interacting with their environment (see any of the examples in [BR16; DHK07; Bol18; BS06]). Lastly, we apply the formal-conceptual tools presented in Chapter 2 and Section 3.1 to rigorously analyze these scenarios. We analyze separate priming effects in Sections 3.2.1 and 3.2.2.

3.2.1 SOUP or SOAP

Start by considering the earlier introduced priming effect as presented by Kahneman [Kah11, page 52]:

“If you have recently seen or heard the word EAT, you are temporarily more likely to complete the word fragment SO_P as SOUP than as SOAP. The opposite would happen, of course, if you had just seen WASH.”

And, as also mentioned previously, the standard explanation of this type of effect is given in terms of contents stored in an agent’s LTM and an increase in the accessibility of some contents by means of processing the stimuli of seeing or hearing the words ‘EAT’ and ‘WASH’, respectively [JW14, pages 97–98; Kah11, pages 52–53]. In particular, seeing or hearing the word ‘EAT’ brings about the processing of content that increases the accessibility of the content expressed by ‘SOUP’, thus making one fill in the word-fragment ‘SO_P’ as ‘SOUP’. Conversely, seeing or hearing the word ‘WASH’ brings about the processing of content that increases the accessibility of the content expressed by ‘SOAP’, thus making one fill in the word fragment ‘SO_P’ as ‘SOAP’. There are multiple ways of construing

the content whose processing is brought about by seeing or hearing EAT and WASH. We will discuss these in the course of our formal analysis below.

We will now present and analyze a scenario based on Kahneman’s example above. The scenario involves an agent, Alex, who is a typical English speaker with standard familiarity of foods and household items:

Alex is sitting at a terminal participating in a word-fragment completion study involving the fragment ‘SO_P’. Now, the terminal displays the word ‘EAT’, after which Alex has to directly complete the word fragment ‘SO_P’. She completes the word fragment as ‘SOUP’. Had the terminal instead shown the word ‘WASH’, she would have completed the word-fragment as ‘SOAP’.

We start by constructing a conceptual space for Alex. The first step in constructing a conceptual space for an agent is picking appropriate quality dimensions and partitioning these into domains. This, in turn, depends on the content we need to represent as stored in that agent’s LTM. We know that the concepts expressed by ‘SOUP’ and ‘SOAP’ must be represented, since these are the contents whose increased accessibility leads to a behavioral change. But, the content whose processing leads to these accessibility increases, and whose processing is brought about by processing the words ‘EAT’ and ‘WASH’, warrants particular attention. There are two reasons for this. Firstly, the words ‘EAT’ and ‘WASH’ are polysemous.³ For example, ‘EAT’ can function as a verb expressing the action of eating, but also as a common noun expressing a concept roughly similar to that expressed by ‘food’. Similarly, ‘WASH’ can function as a verb expressing the action of washing and also function as a common noun expressing a concept roughly similar to that expressed by ‘liquid cleaning agent’ (bleach, vinegar, detergents, etc.), i.e., so-called “washes”. Furthermore, ‘SOAP’ can function as a verb expressing the action of applying soap, and it can function as a common noun denoting soaps. Because the different interpretations of the above words are non-synonymous and of different grammatical categories, we need to choose our interpretations carefully. The reason is that when using the conceptual spaces framework, (common) nouns and adjectives are typically defined with respect to domains consisting of quality dimensions [WGW12; GW12]. In contrast, verbs are typically defined with respect to domains consisting of force dimensions [WGW12; GW12]. And, while we could construct a conceptual space with both force dimensions and quality dimensions, it would be difficult to compare action-concepts and object-concepts because force dimensions do not readily apply to objects, and quality dimensions do not readily apply to actions. For example, consider the force dimensions of degree of compressive jaw force, rate of mandibular movement, and tongue pressure, which all might be used to represent the concept expressed by ‘EAT’ when interpreted as a verb. It is unclear what the degree of compressive jaw force, rate of mandibular movement, and tongue pressure of a soup or a soap would be. Furthermore, consider the force dimensions of frictional surface force and directional sweeping force, which might both be used to represent the concept expressed by ‘WASH’ when interpreted as a verb. Once again, it is unclear what the frictional surface force or directional sweeping force of a soup or a soap would be. An intuitively plausible solution is to include the value $\star$, as discussed in Section 2.1, in every dimension. This would allow us to indicate when a dimension is inapplicable to an object or action. However, for our purposes, this is only an apparent solution. Despite introducing $\star$, actions and objects, and

³‘SOUP’ is also polysemous to the extent that it can be used as part of the verb phrase ‘soup up’. While the word ‘WASH’ seems highly polysemous, ‘EAT’ is less so. ‘SOAP’ seems to have only two standard interpretations, with its common noun interpretation seemingly the most standard.

action-concepts and object-concepts, cannot be meaningfully compared along any of the dimensions that matter to either, since one or the other will involve $\star$ to some extent. Thus, this move reintroduces the issue rather than solving it. There are at least three strategies for overcoming the above issues: (i) choose appropriate interpretations of the relevant words that place all the relevant concepts in the same type of conceptual space, (ii) idealization, or (iii) abstraction. Note that strategy (i) may itself involve some degree of idealization and abstraction. Where possible, we prefer strategy (i), resorting to (ii) or (iii) when no suitable interpretation is available.

Secondly, the content driving accessibility increases brought about by processing a word may not be its literal meaning (for examples, see [JW14]). In the case of verbs, this content could plausibly be an agent’s representation of the typical thing being “acted on” by the action expressed by that verb.⁴ For example, if we interpret ‘EAT’ as a verb expressing the action of eating, the relevant content driving accessibility increase could be the agent’s representation of a typically eaten thing.⁵ Similarly, this content could plausibly be an agent’s representation of a typical instrument used in carrying out the action expressed by that verb.⁶ For example, if we interpret ‘WASH’ as a verb expressing the action of washing, then Alex’s representation of a typical “washing-tool” (brushes, sponges, etc.) or a typical “washing cleaning-agent” could be the content driving accessibility increases. But, no matter whether we take the literal meaning of a word, or content related to it, to drive accessibility increases, we need to model that content in the conceptual space we are constructing. Our formal-conceptual machinery does not itself determine which content drives accessibility increases upon an agent’s exposure to a stimulus. This is left to the modeler, informed by background knowledge about the agent and the situation. Often, when the relevant stimulus is a word, the content brought about by processing it can simply be modeled as its literal meaning. In these cases, identifying the content driving accessibility increases is straightforward. However, content priming is a much broader phenomenon than word-based priming, and stimuli need not be linguistic at all [JW14]. In fact, a smell, an image, a physical environment, or social circumstances can all bring about the processing of content stored in an agent’s memory, thus leading to accessibility increases [JW14]. In such cases, as well as in cases of word-based priming where the relevant content is not the literal meaning of the word, the identification of the priming-relevant content brought about by processing the stimulus may require more careful consideration on the part of the modeler. With these things in mind, we turn to our particular case.

For the current scenario, there are at least two plausible ways of construing the content driving accessibility increases. The first is to opt for (i) and treat ‘EAT’ as expressing a concept roughly similar to that expressed by ‘food’, and ‘WASH’ as expressing a concept roughly similar to that expressed by ‘liquid cleaning agent’. This allows us to have a conceptual space consisting solely of contents that apply to objects. The second option is to treat ‘EAT’ and ‘WASH’ as expressing verbs, but take the content that drives accessibility increases to be Alex’s representation of a typically eaten thing and Alex’s representation of a typical cleaning agent used in washing. Since food is typically eaten, and ‘EAT’ interpreted as a common noun is similar in meaning to ‘food’, both construals would be represented by roughly the same region in an appropriate conceptual space. Furthermore, since cleaning agents are a typical instrument used in washing, and ‘WASH’ interpreted as a common noun is similar in meaning

⁴In the analysis of verbs using the conceptual spaces framework, the thing being acted on by an action is called the ‘patient’ of that action [WGW12; BK17a].

⁵While not explicitly stated by Kahneman [SBS21, pages 52–53], such intuitions seem implicit.

⁶This use of the word ‘instrument’ is typical in the analysis of verbs using the conceptual spaces framework, where it is an object used in carrying out an action [WGW12].

to ‘liquid cleaning agent’, the two construals would likewise be represented by similar regions in an appropriate conceptual space.⁷ Given all of the above, the choice between the different construals does not affect the structural outcome of the analysis. For this thesis, we proceed with the common noun construals for both ‘EAT’ and ‘WASH’. Thus, we treat ‘EAT’ as expressing a concept roughly similar to that expressed by ‘food’, and ‘WASH’ as expressing a concept roughly similar to that expressed by ‘liquid cleaning agent’. This content is what we will take to drive accessibility increases. We do not commit to either construal being the single correct one, and we leave it to the reader to adopt whichever they find most fitting. We will henceforth use the notation ABC to denote the content of the word ‘abc’.

Moving on to our choice of quality dimensions and domains. We introduce the four quality dimensions d_c , d_t , d_n , and d_l , with the following interpretations:

- $d_c = [0, 1]$ represents the cleaning efficacy of objects, interpreted as the degree to which an object is capable of removing dirt, impurities, and the like. The dimension ranges from an incapability for removing dirt, impurities, and the like, indicated by a value of 0.00, to being fully capable of removing dirt, impurities, and the like, indicated by a value of 1.00.
- $d_t = [0, 1]$ represents the toxicity of objects, interpreted as the degree to which an object is capable of causing harm to a human through direct skin contact, eye contact, or ingestion. The dimension ranges from an incapability for causing harm, indicated by a value of 0.00, to causing significant harm, indicated by a value of 1.00.
- $d_n = [0, 1]$ represents the nutritional value of objects, interpreted as how nourishing an object is if consumed. The dimension ranges from no nourishment, indicated by a value of 0.00, to highly nourishing, indicated by a value of 1.00.
- $d_l = [0, 1]$ represents the liquidity of objects, interpreted as the degree to what an object is liquid rather than solid. The dimension ranges from fully solid, indicated by a value of 0.00, to fully liquid, indicated by a value of 1.00.

We partition the above dimensions into singleton domains, as each dimension is mutually separable from the others. Thus, we work with the four domains $\delta_c = \{d_c\}$, $\delta_t = \{d_t\}$, $\delta_n = \{d_n\}$, and $\delta_l = \{d_l\}$, with each domain inheriting the interpretation of the single dimension they contain.⁸ We will work with the conceptual space CS built on the set of domains $\Delta = \{\delta_c, \delta_t, \delta_n, \delta_l\}$, giving us that $CS = d_c \times d_t \times d_n \times d_l$.

The next step in representing the contents stored in Alex’s LTM is specifying which regions in CS that EAT, WASH, SOUP, and SOAP occupy, as well as choosing appropriate prototypes for this

⁷The main difference is that the common noun construal restricts the category to liquid cleaning agents, whereas the typical instrument construal may also include less liquid or solid cleaning agents, such as scrubbing powders or cleaning pastes.

⁸It is important to note that we do not claim that the quality dimensions d_c , d_t , d_n , and d_l are psychologically accurate in that actual people represent the content we will be modeling along these dimensions. Nor do we claim that they provide the best representation of this content from a modeling perspective. A more refined model of Alex’s conceptual space could include further quality dimensions. For example, one could consider dimensions such as texture, smell, color, etc. Moreover, each of the four dimensions we have chosen could itself be decomposed into finer quality dimensions. As an example, consider the dimension d_n . d_n could be replaced by dimensions tracking macronutrient content (e.g. protein content, fat content, carbohydrate content), micronutrient content (e.g. vitamin C content, iron content, etc.), or a standardized nutritional score as used by health organizations. Similarly, d_t could be decomposed into dimensions tracking dermal toxicity and ingestion toxicity separately, and d_c could be decomposed into dimensions tracking efficacy against organic matter and efficacy against mineral deposits. We abstract from all these nuances, as we do not need this level of detail in our analysis.

content. We stipulate Alex’s prototypes by relying on the information given in the scenario that she has standard familiarity with foods and household products. We use the stipulation method because we do not know any exemplars of the different contents for Alex, and because we do not have any predefined convex sets or simple star-shaped sets representing the relevant content.⁹ Since Alex’s prototypes are stipulated, we will use a threshold construction as introduced in Section 2.3 to define the corresponding content regions. For all four contents, we opt for dimension thresholds. While some of the contents differ significantly across some dimensions and not so much along others, such as EAT and SOAP, others have more uniform but still unequal variation across their dimensions, such as SOUP and WASH. In either case, using a space threshold would produce too large or too small variation along some dimensions to capture the content appropriately. We list the Alex prototypes for each of the contents, as well as her distance thresholds in Table 3.1 below:

Content	Prototype				Thresholds			
	q_c	q_t	q_n	q_l	r_c	r_t	r_n	r_l
EAT	0.05	0.05	0.75	0.50	0.05	0.05	0.25	0.50
SOUP	0.05	0.05	0.80	0.875	0.05	0.05	0.10	0.10
WASH	0.90	0.80	0.05	0.80	0.10	0.20	0.05	0.20
SOAP	0.90	0.75	0.05	0.50	0.10	0.10	0.05	0.50

Table 3.1: Prototypes and distance thresholds for Alex’s content in $CS = d_c \times d_t \times d_n \times d_l$. All contents use dimension thresholds.

We now explain our choices of prototypes and distance thresholds. Starting with prototypes. Since Alex is described as a typical English speaker with familiarity of foods and household products, we model her prototypes as what we consider a reasonable representation such an agent would have.¹⁰ We start by explaining the choice of prototype p_{EAT} of EAT and prototype p_{SOUP} of SOUP. p_{EAT} and p_{SOUP} are both assigned near-zero values along dimensions d_c and d_t , indicating that, for Alex, the prototypical food and soup are non-toxic and have negligible cleaning efficacy. In particular, we assign p_{EAT} and p_{SOUP} a value of 0.05 along both of these dimensions. As for nutritional value, p_{EAT} is assigned $q_n = 0.75$ and p_{SOUP} is assigned $q_n = 0.80$, indicating that the prototypical soup is slightly more nutritious than the prototypical food item. Our reasoning for this is that soups tend to be vegetable-rich and prepared by simmering, which increases bio-availability. On the other hand, the prototypical food is much more wide-ranging, and, we think, is less likely to be vegetable-heavy (or a vegetable, for that matter). Moving on to liquidity. p_{EAT} is assigned $q_l = 0.50$, a neutral value reflecting the wide range of food items from fully solid to fully liquid. p_{SOUP} is assigned $q_l = 0.875$, reflecting that soups are very liquid. Thus, the prototypes of EAT and SOUP are $p_{\text{EAT}} = (0.05, 0.05, 0.75, 0.50)$ and $p_{\text{SOUP}} = (0.05, 0.05, 0.80, 0.875)$, respectively. We now explain our choices for the prototype p_{WASH} of WASH and p_{SOAP} of SOAP. p_{WASH} and p_{SOAP} are both assigned near-zero values along dimension d_n , but large values along dimensions d_t d_c . This indicates that, for Alex, the prototypical liquid cleaning agent and soap is non-nutritious and toxic, but are highly effective at removing dirt and impurities. In particular, we assign p_{WASH} and p_{SOAP} a value of 0.05 along d_n , and a value of 0.90 along q_c . As for d_t , p_{WASH} is assigned $q_t = 0.80$ while the value and p_{SOAP} is assigned $q_t = 0.75$, indicating that the prototypical liquid cleaning agent

⁹We refer the reader back to Section 2.3 for our exposition on the different methods for obtaining a prototype.

¹⁰Once again, we do not claim these are the only reasonable choices, nor that they generalize to all English speakers. Rather, our choices are modeling decisions, which we motivate in this and the following paragraph.

is slightly more toxic than the prototypical soap. Our reasoning is that WASH may include stronger liquid cleaning agents than soaps, such as detergents or disinfectants. As for liquidity, p_{WASH} is assigned $q_l = 0.80$, indicating that liquid cleaning agents may vary in viscosity. p_{SOAP} is assigned $q_l = 0.50$, reflecting that soaps range from solid bars to fully liquid with no particular degree of liquidity being more prototypical. Thus, the prototypes of WASH and SOAP are $p_{\text{WASH}} = (0.90, 0.80, 0.05, 0.80)$ and $p_{\text{SOAP}} = (0.90, 0.75, 0.05, 0.50)$, respectively.

Moving on to our choice of distance thresholds. For EAT, we set the thresholds along d_c and d_t to $r_c = r_t = 0.05$, reflecting that foods vary little in cleaning efficacy and toxicity. The threshold along d_n is set to $r_n = 0.25$, reflecting the considerable variation in nutritional value across food items. The threshold along d_l is set to $r_l = 0.50$, reflecting that food spans the full liquidity range, from dry crackers to broths. For SOUP, the thresholds along d_c and d_t are both set to $r_c = r_t = 0.05$, for the same reason as EAT. The thresholds along d_n and d_l are both set to $r_n = r_l = 0.10$, reflecting that soups cluster quite tightly around what a typical soup is like in terms of nutritional value and liquidity. For WASH, the threshold along d_c is set to $r_c = 0.10$, reflecting that liquid cleaning agents are high in cleaning efficacy across the whole category. The threshold along d_t is set to $r_t = 0.20$, reflecting some variation in toxicity across cleaning agents. The threshold along d_n is set to $r_n = 0.05$, reflecting that cleaning agents are non-nutritious across the whole category. The threshold along d_l is set to $r_l = 0.20$, reflecting that liquid cleaning agents have some variation in viscosity. For SOAP, the thresholds along d_c and d_t are both set to $r_c = r_t = 0.10$, reflecting that soaps vary little in cleaning efficacy and toxicity. The threshold along d_n is set to $r_n = 0.05$, since soaps are non-nutritious across the whole category. The threshold along d_l is set to $r_l = 0.50$, reflecting that soaps span the full liquidity range, from solid bars to fully liquid. Having constructed Alex’s conceptual space as well as specified the sets representing the relevant contents, we now turn to computing the similarity-based accessibility increments between her contents as defined in Section 2.4.

Start by recalling that the similarity-based increment of some content C upon processing content P is given by $\text{Sim}(P, C)$, where Sim is the content similarity function as introduced in Section 2.4. Further note that $\text{Sim}(P, C)$ is defined as the similarity between the prototypes of P and C , which in turn is defined as an exponentially decreasing function of the full-space distance metric d_{CS}^Δ in Section 2.2. Lastly, the full-space distance metric d_{CS}^Δ , as defined in Section 2.2, is the weighted Manhattan metric of the intra-domain distances between the relevant points. And, as discussed in that Section, such distances are always computed relative to a salience profile. Thus, we need to specify a salience profile fit for the present scenario. As we have no information regarding Alex’s salience or attention, we take all salience weights to be uniform. Starting with the sets of intra-domain salience weights W_{δ_i} for each domain $\delta_i \in \Delta$. Observe that for each domain $\delta_i \in \Delta$ is a singleton $\{d_i\}$, the set of intra-domain salience weights W_{δ_i} contains a single weight w_{d_i} . As a set of salience weights for a domain must sum to 1, we get that $w_{d_i} = 1$ for each i . As for the inter-domain weights, we need to stipulate these. We set $w_{\delta_c} = w_{\delta_t} = w_{\delta_n} = w_{\delta_l} = 1$, which satisfies the requirement that the domain weights in a salience profile must sum to $|\Delta| = 4$. This gives us the salience profile $W = (W_\Delta, \{W_{\delta_i}\}_{\delta_i \in \Delta})$, where $W_\Delta = \{w_{\delta_c}, w_{\delta_t}, w_{\delta_n}, w_{\delta_l}\}$ with $w_{\delta_c} = w_{\delta_t} = w_{\delta_n} = w_{\delta_l} = 1$, and $W_{\delta_i} = \{w_{d_i}\}$ with $w_{d_i} = 1$ for each $\delta_i \in \Delta$. We now compute the full-space Manhattan distances, relative to salience profile W , between

the prototypes of Alex’s contents, with the prototypes given in Table 3.1:

$$\begin{aligned} d_{CS}^\Delta(p_{\text{EAT}}, p_{\text{SOUP}}, W) &= |0.05-0.05| + |0.05-0.05| + |0.75-0.80| + |0.50-0.875| \\ &= 0 + 0 + 0.05 + 0.375 = 0.425, \end{aligned}$$

$$\begin{aligned} d_{CS}^\Delta(p_{\text{EAT}}, p_{\text{SOAP}}, W) &= |0.05-0.90| + |0.05-0.75| + |0.75-0.05| + |0.50-0.50| \\ &= 0.85 + 0.70 + 0.70 + 0 = 2.25, \end{aligned}$$

$$\begin{aligned} d_{CS}^\Delta(p_{\text{WASH}}, p_{\text{SOAP}}, W) &= |0.90-0.90| + |0.80-0.75| + |0.05-0.05| + |0.80-0.50| \\ &= 0 + 0.05 + 0 + 0.30 = 0.35, \end{aligned}$$

$$\begin{aligned} d_{CS}^\Delta(p_{\text{WASH}}, p_{\text{SOUP}}, W) &= |0.90-0.05| + |0.80-0.05| + |0.05-0.80| + |0.80-0.875| \\ &= 0.85 + 0.75 + 0.75 + 0.075 = 2.425. \end{aligned}$$

The last formal component we need to specify is the sensitivity parameter $\lambda \in \mathbb{R}_{>0}$ for the similarity function. Here, we set $\lambda = 2$, which gives us the following values for the similarities between Alex’s contents:

$$\begin{aligned} \text{Sim}(\text{EAT}, \text{SOUP}) &= \text{Sim}(p_{\text{EAT}}, p_{\text{SOUP}}) = e^{-2 \cdot 0.425} = e^{-0.85} \approx 0.427, \\ \text{Sim}(\text{EAT}, \text{SOAP}) &= \text{Sim}(p_{\text{EAT}}, p_{\text{SOAP}}) = e^{-2 \cdot 2.25} = e^{-4.50} \approx 0.011, \\ \text{Sim}(\text{WASH}, \text{SOAP}) &= \text{Sim}(p_{\text{WASH}}, p_{\text{SOAP}}) = e^{-2 \cdot 0.35} = e^{-0.70} \approx 0.497, \\ \text{Sim}(\text{WASH}, \text{SOUP}) &= \text{Sim}(p_{\text{WASH}}, p_{\text{SOUP}}) = e^{-2 \cdot 2.425} = e^{-4.85} \approx 0.008. \end{aligned}$$

Having computed the relevant similarity-based increments, we are now in a position to apply the action-selection mechanism introduced in Section 3.1. Recall that the action-selection mechanism captures how accessibility levels of contents bear on the selection of actions, with the basic idea being that candidate actions receive a selection score, computed by aggregating the accessibility levels of contents weighted by their relevance to each action. Then, if the executability condition of a candidate action is satisfied, that action is executed. If no candidate action has its executability condition satisfied, then the fallback action has its executability condition satisfied. The task Alex has to perform is that of completing the word fragment SO_P , for which we take there to be two possible candidate actions. Namely, the action of completing the fragment as ‘SOUP’, which we denote by α_{SOUP} , and the action of completing it as ‘SOAP’, which we denote by α_{SOAP} . For this particular scenario, we do not formally specify the fallback action as it does not play a significant role. To connect these candidate actions to the contents stored in Alex’s memory, we make use of the relevance function introduced in Section 3.1, which specifies for each content $C \in \mathcal{C}$ and each action $\alpha \in \mathcal{A}$ the degree $\text{Rel}(\alpha, C)$ to which content C is relevant to the selection of action α . In the present scenario, SOUP is relevant to α_{SOUP} because ‘SOUP’ is a completion of the fragment SO_P and expresses the content SOUP . Similarly for SOAP and

α_{SOAP} . All other content-action pairs receive a relevance of 0. Formally:¹¹

$$\begin{aligned}\text{Rel}(\alpha_{\text{SOUP}}, \text{SOUP}) &= 1, \\ \text{Rel}(\alpha_{\text{SOAP}}, \text{SOAP}) &= 1, \\ \text{Rel}(\alpha, C) &= 0 \quad \text{otherwise.}\end{aligned}$$

We assume that each content has a baseline accessibility of zero. Then, by applying the similarity-based accessibility update from Section 2.4, we obtain that:

$$\begin{aligned}\text{Acc}_{\text{EAT}}(\text{SOUP}) &= 0 + \text{Sim}(\text{EAT}, \text{SOUP}) \approx 0.427, \\ \text{Acc}_{\text{EAT}}(\text{SOAP}) &= 0 + \text{Sim}(\text{EAT}, \text{SOAP}) \approx 0.011.\end{aligned}$$

We are now in a position to compute selection scores as defined in Section 3.1. Starting with the selection score of action α_{SOUP} after Alex processes content EAT, we simply note that the only content with a non-zero relevance for α_{SOUP} is SOUP, and so we get that:

$$\begin{aligned}\text{score}(\alpha_{\text{SOUP}}) &= \sum_{C \in \mathcal{C}} \text{Acc}_{\text{EAT}}(C) \cdot \text{Rel}(\alpha_{\text{SOUP}}, C) \\ &= \text{Acc}_{\text{EAT}}(\text{SOUP}) \cdot \text{Rel}(\alpha_{\text{SOUP}}, \text{SOUP}) \\ &\approx 0.427\end{aligned}$$

For the selection score of α_{SOAP} after Alex processes content EAT, we note that the only content with a non-zero relevance for α_{SOAP} is SOAP, giving us that:

$$\begin{aligned}\text{score}(\alpha_{\text{SOAP}}) &= \sum_{C \in \mathcal{C}} \text{Acc}_{\text{EAT}}(C) \cdot \text{Rel}(\alpha_{\text{SOAP}}, C) \\ &= \text{Acc}_{\text{EAT}}(\text{SOAP}) \cdot \text{Rel}(\alpha_{\text{SOAP}}, \text{SOAP}) \\ &\approx 0.011\end{aligned}$$

Having computed accessibility scores, we can now determine which executability condition holds. The only two actions competing for selection are α_{SOUP} and α_{SOAP} . And, since we now know that $\text{score}(\alpha_{\text{SOUP}}) > \text{score}(\alpha_{\text{SOAP}})$ since $\text{score}(\alpha_{\text{SOUP}}) \approx 0.427$ and $\text{score}(\alpha_{\text{SOAP}}) \approx 0.011$, the executability condition of α_{SOUP} is satisfied, meaning that if Alex were to be asked to complete the word fragment ‘SO_P’, she would complete it as ‘SOUP’.

The situation is symmetric for the case where the terminal displays the word ‘WASH’ instead of ‘EAT’, with WASH the content brought about by exposure to that stimulus. Here, the accessibility update yields:

$$\begin{aligned}\text{Acc}_{\text{WASH}}(\text{SOAP}) &\approx 0.497, \\ \text{Acc}_{\text{WASH}}(\text{SOUP}) &\approx 0.008,\end{aligned}$$

¹¹In this particular example, the relevance function is binary. The role of the relevance function becomes important in more complex scenarios where each piece of content bears on multiple responses, or where relevance is graded.

and accordingly

$$\begin{aligned}\text{score}(\alpha_{\text{SOAP}}) &\approx 0.497, \\ \text{score}(\alpha_{\text{SOUP}}) &\approx 0.008.\end{aligned}$$

Thus, the executability condition of α_{SOAP} is satisfied, meaning that if Alex were to be asked to complete the word fragment ‘SO_P’, she would complete it as ‘SOAP’.

The two cases together illustrate the mechanism underlying content priming as modeled in our framework. Exposure to the given stimuli (the words ‘EAT’ and ‘WASH’, respectively) brings about the processing of a piece of content stored in Alex’s memory (EAT and WASH, respectively), and this processing raises the accessibility of other contents in proportion to their similarity to it. In turn, these different accessibility increases would lead to different behaviors if Alex were asked to fill in the word-fragment ‘SO_P’.

3.2.2 The Florida Effect

The second example we consider is the so-called Florida effect. Kahneman describes it as follows [Kah11, pages 52–53], with the study in question being [BCB96]:

“In an experiment that became an instant classic, the psychologist John Bargh and his collaborators asked students at New York University—most aged eighteen to twenty-two—to assemble four-word sentences from a set of five words (for example, “finds he it. yellow instantly). For one group of students, half the scrambled sentences contained words associated with the elderly, such as Florida, forgetful, bald, gray, or wrinkle. When they had completed that task, the young participants were sent out to do another experiment in an office down the hall. That short walk was what the experiment was about. The researchers unobtrusively measured the time it took people to get from one end of the corridor to the other. As Bargh had predicted, the young people who had fashioned a sentence from words with an elderly theme walked down the hallway significantly more slowly than the others.”

We will now present and analyze a scenario based on the above priming effect. The scenario involves an agent, Bob, whom we take to be a typical American student:

Bob is participating in a word arrangement study where he is given a set of scrambled words. One of these words is ‘forgetful’. After successfully rearranging all the words into a sentence, Bob is directly asked to walk down a corridor. Bob walks slower than his normal walking speed. The same effect would have been observed if ‘forgetful’ were replaced by ‘wrinkly’, ‘gray’, or ‘Florida’. Had Bob instead been given a set of words not typically associated with old age, he would have walked at his normal speed.¹²

¹²In the original study, the behavioral change is plausibly the result of an aggregate accessibility increase induced by processing multiple words across several sentences that are associated with old age, rather than a single word. In particular, it may be the case that the accessibility increase induced by processing a single word related to old age is insufficient in producing a behavioral change, whereas the aggregate of several such accessibility increases would be. This aggregate update could be modeled as $\text{Acc}_P(C) = \text{Acc}(C) + \sum_{i=1}^n \text{Sim}(P_i, C)$, where $P = \{P_1, \dots, P_n\}$ are the contents whose processing is brought about by the agent’s exposure to each of the words related to old age across the sentences. One might further refine selection scores as introduced in Section 3.1 by parameterizing them with a threshold $\theta \geq 0$, such that only contents C satisfying $\text{Acc}(C) \geq \theta$ contribute to selection scores. In this scenario, we simplify the original

The standard explanation mirrors the one from the SOUP or SOAP example (see 3.2.1). Firstly, we will take the content OLD as the content that is relevant to Bob’s walking speed. As such, we need to represent this content in the space we are constructing. Secondly, the content whose processing is brought about by the Bob’s processing of the adjectives ‘forgetful’, ‘bald’, and ‘wrinkly’ is straightforward. In particular, it is simply the property expressed by the respective adjectives. Similar to the SOUP or SOAP example, we denote these by FORGETFUL, BALD, and WRINKLY, respectively. Lastly, for the place noun ‘Florida’, our treatment requires a bit more care. As discussed in Section 3.2.1, the content driving accessibility increases need not be the literal meaning of the stimulus word, but may instead be content related to it. This is the stance we will take with respect to ‘Florida’. As it is a place noun, a natural candidate for the content whose processing is brought about by Bob’s exposure to it is his representation of the typical inhabitant of Florida. Intuitively, this content seems to be driving the accessibility increase for OLD, particularly since it is a cultural stereotype prevalent in America that a typical Floridian is a retiree. And, since Bob is American, it seems appropriate to assume that his representation of a typical Floridian should be close to that of a retiree, which in turn should be close to his representation of someone old. We therefore model the content driving the accessibility increase brought about by processing ‘Florida’ as a typical inhabitant of Florida, which is close to that of a retiree. We denote this content by FLORIDA.

We introduce three quality dimensions d_m , d_h , and d_s , each forming a singleton domain $\delta_m = \{d_m\}$, $\delta_h = \{d_h\}$, and $\delta_s = \{d_s\}$. Each domain inherits the interpretation of the single dimension it contains. d_m is the quality dimension of memory function, interpreted as the degree to which a person’s memory functions well. We set $d_m = [0, 1]$, in which a value of 1 indicates that a person has excellent memory function, whereas a value of 0 indicates that a person has very poor memory function. d_h is the quality dimension of hair coverage, interpreted as the proportion of hair a person retains. We set $d_h = [0, 1]$, in which a value of 1 indicates that a person has a full head of hair, whereas a value of 0 indicates that a person is completely bald. d_s is the quality dimension of skin smoothness, interpreted as the degree to which a person’s skin is free of wrinkles. We set $d_s = [0, 1]$, in which a value of 1 indicates that a person has very smooth skin, whereas a value of 0 indicates that a person has very wrinkled skin. We will work with the conceptual space CS built on the set of domains $\Delta = \{\delta_m, \delta_h, \delta_s\}$, giving us that $CS = d_m \times d_h \times d_s$.

We model FORGETFUL as a region in the lower end of d_m . In particular, we take it to occupy the interval $[0, 0.3]$ of d_m , reflecting that for Bob, being forgetful means having a significantly below-average memory function. We do not impose any restrictions on what counts as forgetful along the other dimensions, as we take forgetful as a property characterized along the dimension d_m . Thus, $\text{FORGETFUL} = [0, 0.3] \times [0, 1] \times [0, 1]$. We model BALD as a region in the lower end of d_h . In particular, we take it to occupy the interval $[0, 0.5]$, reflecting that for Bob, being bald means having lost at least half of one’s hair. Since baldness is a property just like being forgetful, we do not impose restrictions for dimensions other than d_h , giving us that $\text{BALD} = [0, 1] \times [0, 0.5] \times [0, 1]$. We model WRINKLY as a region in the lower end of d_s . In particular, we take it to occupy the interval $[0, 0.3]$, reflecting that for Bob, the property of being wrinkly means having significantly reduced skin smoothness. Similarly to the two previously modeled properties, we set $\text{WRINKLY} = [0, 1] \times [0, 1] \times [0, 0.3]$.

The next step in representing the contents stored in Bob’s memory is specifying which region in CS

study by having Bob assemble a single sentence containing a single word that is typically associated with old age. This is a simplification for illustrative purposes, and the aggregate generalization together with the threshold Θ seems a good candidate for an extension of the framework.

that OLD occupies. We specify this content by means of a prototype p together with a space threshold r , defining a closed ball $B(p, r)$ centered at the prototype p ¹³. We list the prototype and distance radius in Table 3.2 below: We now explain our choice of prototype and radius for OLD. We assign p_{OLD}

Content	Prototype			Radius
	q_m	q_b	q_s	r
OLD	0.25	0.25	0.20	0.15

Table 3.2: Prototype coordinates and distance radius for OLD defined for $CS = d_m \times d_b \times d_s$.

low values along all three dimensions, reflecting that old people tend to have reduced memory function ($q_m = 0.25$), lower hair coverage ($q_b = 0.25$), and less smooth skin ($q_s = 0.20$). Since we represent OLD by a single space threshold, defining the closed ball $B(p_{\text{OLD}}, r)$, we set its radius to $r = 0.15$.¹⁴ We use a single threshold because the things that count as old vary to a comparable degree along all three dimensions, and we make the radius relatively large. The prototype also sits far enough from the extremes of each dimension that the ball remains within CS with p_{OLD} as its centroid. In order to compute the similarity-based accessibility update for each of the contents we have just defined, we still need prototypes for FORGETFUL, BALD, and WRINKLY. We obtain these by computing their centroids as laid out in Section 2.3:

Property	Dimension	Interval		Mean
		a_i	b_i	p_i
FORGETFUL	d_m	0	$0.\bar{3}$	$0.1\bar{6}$
	d_h	0	1	0.5
	d_s	0	1	0.5
BALD	d_m	0	1	0.5
	d_h	0	0.5	0.25
	d_s	0	1	0.5
WRINKLY	d_m	0	1	0.5
	d_h	0	1	0.5
	d_s	0	$0.\bar{3}$	$0.1\bar{6}$

Table 3.3: Coordinate-wise means $p_i = \frac{a_i + b_i}{2}$ along each dimension for the properties FORGETFUL, BALD, and WRINKLY in $CS = d_m \times d_h \times d_s$.

To compute the relevant similarity-based accessibility updates, we still need a salience profile W and a sensitivity parameter λ . Like with the SOUP or SOAP example, we use a uniform salience profile and set the sensitivity parameter to $\lambda = 2$.

Lastly, the task Bob has to perform is that of walking down the corridor, for which we take there to be one candidate action, namely the action of walking slowly, denoted by α_{SLOW} , and the fallback action of walking at his regular walking speed, denoted by α_0 . Slow walking is a tendency characteristic of old age, and so the content OLD is relevant to the selection of α_{SLOW} . As such, we set $\text{Rel}(\alpha_{\text{SLOW}}, \text{OLD}) = 1$.

¹³See Section 2.3 for our coverage of this construction.

¹⁴We refer the reader to Section 2.3 for our explanation of this construction.

All other content-action pairs receive a relevance of 0. We now go case-by-case, verifying that the executability condition of α_{slow} is satisfied in each case after processing the relevant content. We assume that the accessibility of any content before processing is zero. For each case, we first compute the relevant prototype distance. Then, we compute the similarity of these prototypes, which gives us the similarity between their associated contents. Lastly, we compute the selection score of α_{slow} . Where ABC represents some content, we denote its prototype by p_{ABC} below:

Case 1: Processing FORGETFUL. The distance from $p_{\text{FORGETFUL}}$ to p_{OLD} in CS is

$$\begin{aligned} d_{CS}^{\Delta}(p_{\text{FORGETFUL}}, p_{\text{OLD}}, W) &= |0.15-0.25| + |0.50-0.25| + |0.50-0.20| \\ &= 0.10 + 0.25 + 0.30 = 0.65, \end{aligned}$$

and so we get the following similarity-based increment $\text{Sim}(\text{FORGETFUL}, \text{OLD}) = \text{Sim}(p_{\text{FORGETFUL}}, p_{\text{OLD}}) = e^{-2 \times 0.65} = e^{-1.30} \approx 0.272$. Thus, the similarity based accessibility update yields $\text{Acc}_{\text{FORGETFUL}}(\text{OLD}) \approx 0.272$. Applying the selection score formula from Section 3.1, we obtain $\text{score}(\alpha_{\text{slow}}) \approx 0.272$. Thus, the executability condition of α_{slow} is satisfied.

Case 2: Processing BALD. The distance from p_{BALD} to p_{OLD} in CS is

$$\begin{aligned} d_{CS}^{\Delta}(p_{\text{BALD}}, p_{\text{OLD}}, W) &= |0.50-0.25| + |0.25-0.25| + |0.50-0.20| \\ &= 0.25 + 0 + 0.30 = 0.55, \end{aligned}$$

and so we get the following similarity-based increment $\text{Sim}(\text{BALD}, \text{OLD}) = \text{Sim}(p_{\text{BALD}}, p_{\text{OLD}}) = e^{-2 \times 0.55} = e^{-1.10} \approx 0.333$. Thus, the similarity based accessibility update yields $\text{Acc}_{\text{BALD}}(\text{OLD}) \approx 0.333$. Applying the selection score formula from Section 3.1, we obtain $\text{score}(\alpha_{\text{slow}}) \approx 0.333$. Thus, the executability condition of α_{slow} is satisfied.

Case 3: Processing WRINKLY. The distance from p_{WRINKLY} to p_{OLD} in CS is

$$\begin{aligned} d_{CS}^{\Delta}(p_{\text{WRINKLY}}, p_{\text{OLD}}, W) &= |0.50-0.25| + |0.50-0.25| + |0.15-0.20| \\ &= 0.25 + 0.25 + 0.05 = 0.55, \end{aligned}$$

and so we get the following similarity-based increment $\text{Sim}(\text{WRINKLY}, \text{OLD}) = \text{Sim}(p_{\text{WRINKLY}}, p_{\text{OLD}}) = e^{-2 \times 0.55} = e^{-1.10} \approx 0.333$. Thus, the similarity based accessibility update yields $\text{Acc}_{\text{WRINKLY}}(\text{OLD}) \approx 0.333$. Applying the selection score formula from Section 3.1, we obtain $\text{score}(\alpha_{\text{slow}}) \approx 0.333$. Thus, the executability condition of α_{slow} is satisfied.

Case 4: Processing FLORIDA. The distance from p_{FLORIDA} to p_{OLD} in CS is

$$\begin{aligned} d_{CS}^{\Delta}(p_{\text{FLORIDA}}, p_{\text{OLD}}, W) &= |0.30-0.25| + |0.30-0.25| + |0.25-0.20| \\ &= 0.05 + 0.05 + 0.05 = 0.15, \end{aligned}$$

and so we get the following similarity-based increment $\text{Sim}(\text{FLORIDA}, \text{OLD}) = \text{Sim}(p_{\text{FLORIDA}}, p_{\text{OLD}}) = e^{-2 \times 0.55} = e^{-1.10} \approx 0.741$. Thus, the similarity based accessibility update yields $\text{Acc}_{\text{FLORIDA}}(\text{OLD}) \approx 0.741$. Applying the selection score formula from Section 3.1, we obtain $\text{score}(\alpha_{\text{slow}}) \approx 0.741$. Thus, the executability condition of α_{slow} is satisfied.

Thus, all four cases deliver the correct result. In each case, the content whose processing is brought

about by the agent’s processing of the word in question is sufficiently similar to OLD in Bob’s conceptual space to produce an accessibility increase such that the executability condition of α_{SLOW} is satisfied. Before any priming occurs, $\text{Acc}(\text{OLD}) = 0$ and so $\text{score}(\alpha_{\text{SLOW}}) = 0$, implying that no candidate action wins. As such, had Bob not been primed, he would have performed the fallback action α_0 of walking at his normal pace.

Chapter 4

A logic of content priming

In this chapter, we present two logics for reasoning about content priming effects. Both logics will be multi-agent, albeit we only purposefully introduce the first logic as multi-agent to make the extension to the second logic smoother. The components of the class of models over which the second logic is defined are separated into a *behavior-relevant* part and a *belief-relevant* part, allowing us to capture accessibility-based action selection and the informational effects of actions executed on the basis of this mechanism. This will be fully explicated below. The rest of the chapter will go as follows. In Section 4.1, we introduce the first logic. The main features of this logic are that it is capable of comparing the selection scores of actions as introduced in Section 3.1, and that it has content processing operators that capture similarity-based accessibility increases as introduced in Section 2.4. We start by introducing a general accessibility update operation based on Definition 15, before introducing the similarity-based accessibility update operation based on Definition 17. The logic uses strict inequalities of selection scores to formalize executability conditions of actions, as introduced in Section 3.1. Lastly, we revisit the SOUP or SOUP example of Section 3.2.1, and show how it can be formalized using content processing operators and formalized executability conditions. As such, the first logic provides an analysis of a *mechanism* driving priming-induced behavior. In Section 4.2, we introduce our second logic, building on the one presented in the preceding section. The main features new to this logic are that it has operators for representing beliefs, as well as dynamic operators that capture the informational effects of actions performed on the basis of accessibility-based action selection. Then, we put the machinery to use by presenting and analyzing a scenario that display one way in which content priming effects can affect belief formation.

4.1 Content processing and executability

Through the rest of this Chapter, we assume a non-empty set of actions, denoted by $\mathcal{A}$, to be fixed in the background. We also assume a non-empty finite set of agents, denoted by Ag , to be fixed in the background.

4.1.1 Syntax

To represent the selection scores of actions, we introduce the notion of a *score term*:

Definition 20 (Score terms). *Let $\mathcal{A}$ be a non-empty set of actions. The set of score terms is defined*

as:

$$T := \{\text{score}(\alpha) : \alpha \in \mathcal{A}\}.$$

A term ‘ $\text{score}(\alpha)$ ’ represents the selection score of action α . We will relativize the comparison of selection scores to agents using strict inequalities labeled by agents, as each agent may have different selection scores for different actions.

We now introduce our language $\mathcal{L}_P$, which contain the following operators: $\neg$ expressing truth-functional negation, $\wedge$ expressing truth-functional conjunction, and dynamic operators of the form $[P]$ capturing similarity-based accessibility increases upon processing the content denoted by P . Lastly, $\mathcal{L}_P$ contains strict inequalities between score terms for comparing selection scores of actions. To talk about the contents stored in an agent’s LTM, we assume a countable set of *content terms*, denoted by Con .

Definition 21 (Language $\mathcal{L}_P$). *The language $\mathcal{L}_P$ is defined by the following BNF:*

$$\varphi ::= t >_a t' \mid \neg\varphi \mid \varphi \wedge \varphi \mid [P]_a\varphi,$$

where $t, t' \in T$ are score terms, $a \in \mathbf{Ag}$ is an agent, and $P \in Con$ is a content term.

Formulas of the form $t >_a t'$ are strict inequalities over score terms. We define inequalities as follows:

$$t \geq_a t' := \neg(t' >_a t).$$

Furthermore, we abbreviate the connectives ‘ $\vee$ ’, ‘ $\rightarrow$ ’, and ‘ $\leftrightarrow$ ’ in the usual way. Lastly, formulas of the form ‘ $[P]_a\varphi$ ’ read: after agent a processes content P , φ .¹ Lastly, formulas for the form ‘ $t >_a t'$ ’ read: score term t is strictly greater than score term t' for agent a . Now that our syntactic notions are in place, we proceed to define our semantics.

4.1.2 Semantics

Firstly, we introduce our class of models called and explain how to interpret score terms in this class of models. Secondly, we define model-changing operations for capturing accessibility increases induced by the processing of content. Thirdly, we prove that these operations yield a new model from our defined class. As these operations are model-changing operations, they need not return a model of the same class. Establishing that they do return a model of the same class is what lets us readily use the operations in our truth clauses and not have to present new definitions for the resulting type of model. While the verification is routine, it is a new model-changing operation we introduce, and the result is required for the machinery to function. Lastly, we give truth clauses.

Starting with our class of models:

Definition 22 (Multi-agent content priming model). *A multi-agent content priming model is a structure $M = (CS_a, \mathcal{C}_a, \sigma_a, \text{Acc}_a, \text{Rel}_a)_{a \in \mathbf{Ag}}$ where for each agent $a \in \mathbf{Ag}$:*

- CS_a is a conceptual space;
- $\mathcal{C}_a \subseteq \mathcal{P}(CS_a)$ is a set of contents;

¹Strictly speaking, ‘ P ’ is a content term and not a piece of content. However, given our intended semantics, we will read content processing operators as if the content term indexing them denotes a piece of content.

- $\sigma_a : \text{Con} \rightarrow \mathcal{C}_a$ is a content interpretation function;
- $\text{Acc}_a : \mathcal{C}_a \rightarrow \mathbb{R}_{\geq 0}$ is a content accessibility function;
- $\text{Rel}_a : \mathcal{A} \times \mathcal{C}_a \rightarrow [0, 1]$ is a relevance function;

and we say that the tuple $M_a = (CS_a, \mathcal{C}_a, \sigma_a, \text{Acc}_a, \text{Rel}_a)$ is a content priming model for agent a . We assume that each piece of content $C \in \mathcal{C}_a$ has a prototype $p_{C_a} \in CS_a$. Furthermore, we assume that each content priming model comes parametrized with a salience profile W_a for the set of domains Δ_a on which the conceptual space CS_a is built, as well as a sensitivity parameter $\lambda_a \in \mathbb{R}_{>0}$.

Where $M_a = (CS_a, \mathcal{C}_a, \sigma_a, \text{Acc}_a, \text{Rel}_a)$ is a content priming model for agent a , the components of M_a are used in the following way: CS_a is a conceptual space used in structuring and defining the contents in a 's memory, $\mathcal{C}_a$ represents the contents stored in a 's memory, σ_a is a function specifying what contents is denoted by the content terms, Acc_a is a function specifying the accessibility levels of contents in $\mathcal{C}_a$ for a , and Rel_a is a function specifying what content is relevant for the selection of which action, and to what extent, for agent a . While CS_a allows us to define the set of contents $\mathcal{C}_a$ and σ_a allows us to talk about that content, it is $\mathcal{C}_a$, together with Acc_a and Rel_a , that allows us to define the interpretation of score terms in a content priming model of agent a :

Definition 23 (Interpretation of score terms). *Let $M = (CS_a, \mathcal{C}_a, \sigma_a, \text{Acc}_a, \text{Rel}_a)_{a \in \text{Ag}}$ be a multi-agent content priming model. The interpretation of a score term $\text{score}(\alpha) \in T$ for agent a in M is given by:*

$$\text{score}(\alpha)_a^M := \sum_{C \in \mathcal{C}_a} \text{Acc}_a(C) \cdot \text{Rel}_a(\alpha, C).$$

Our interpretation of score terms exactly mirrors our informal definition of score terms from Section 3.1.

We now make precise the effects of content processing on content accessibility. Remember that operators of the form $[P]_a$ with P a piece of content are intended to capture the effects of agent a processing content P . And, for our purposes, we isolate the effects that the processing of content has on content accessibility. As such, we need a formal device that tells us how to make these increments in content accessibility. We develop this device in two steps. Firstly, we define a general accessibility update operation on content priming models using the notion of an increment function as introduced in Section 2.4. This gives us a general setting for describing the results on content accessibility of processing content. We then show that this operation yields a new content priming model. Secondly, we show that the content similarity function of Definition 16 is an increment function, and define similarity-based accessibility update operation as a special case of the accessibility update operation. As such, we know that this update operation also yields a new content priming model. Now for the general definition of the accessibility update operation:

Definition 24 (Accessibility update). *Let $M = (CS_a, \mathcal{C}_a, \sigma_a, \text{Acc}_a, \text{Rel}_a)_{a \in \text{Ag}}$ be a multi-agent content priming model, $a \in \text{Ag}$ an agent, $\iota : \mathcal{C}_a \times \mathcal{C}_a \rightarrow \mathbb{R}_{\geq 0}$ an increment function on $\mathcal{C}_a$, and $P \in \mathcal{C}_a$ any piece of content. The accessibility update of M for agent a with P using ι , is written $M_{\iota, a}^P$, and defined as the multi-agent content priming model that agrees with M on every agent $b \neq a$ and replaces M_a with:*

$$(M_a)_\iota^P := (CS_a, \mathcal{C}_a, \sigma_a, (\text{Acc}_a)_\iota^P, \text{Rel}_a),$$

where $(M_a)_l^P$ differs from M_a only in its accessibility function, in which for each $C \in \mathcal{C}_a$, $(\text{Acc}_a)_l^P(C)$ is defined as:

$$(\text{Acc}_a)_l^P(C) := \text{Acc}_a(C) + \iota_P(C).$$

When the increment function ι is clear from context, we omit subscripting with ι .

The Proposition below tells us that applying the accessibility update operation to a content model yields a new content model:

Proposition 4. *Let M be a content priming model, $a \in \mathbf{Ag}$ an agent, ι an increment function on $\mathcal{C}_a$, and $P \in \mathcal{C}_a$ a piece of content. Then $M_{l,a}^P$ is a multi-agent content priming model.*

Proof. Let $M = (CS_a, \mathcal{C}_a, \sigma_a, \text{Acc}_a, \text{Rel}_a)_{a \in \mathbf{Ag}}$ be a multi-agent content priming model, $a \in \mathbf{Ag}$ an agent, $\iota : \mathcal{C}_a \times \mathcal{C}_a \rightarrow \mathbb{R}_{\geq 0}$ an increment function on $\mathcal{C}_a$, and $P \in \mathcal{C}_a$ any piece of content. Now consider the accessibility update of M for agent a with P using ι , $M_{l,a}^P$. By Definition 24, the only component of $M_{l,a}^P$ that differs from M is $(\text{Acc}_a)_l^P$ of $(M_a)_l^P$. Thus, to show that $M_{l,a}^P$ is a content model it is sufficient to show that $(\text{Acc}_a)_l^P$ is a content accessibility function on $\mathcal{C}_a$. As such, we need to show that $(\text{Acc}_a)_l^P$ is a function from $\mathcal{C}_a$ to $\mathbb{R}_{\geq 0}$. So, take an arbitrary piece of content $C \in \mathcal{C}_a$. By Definition 13 and Definition 14, we know that $\text{Acc}_a(C) \in \mathbb{R}_{\geq 0}$ and $\iota_P(C) \in \mathbb{R}_{\geq 0}$, respectively. Thus, since $(\text{Acc}_a)_l^P(C) = \text{Acc}_a(C) + \iota_P(C)$ by Definition 24, we know that there is a unique $x \in \mathbb{R}_{\geq 0}$ such that $(\text{Acc}_a)_l^P(C) = x$. x is unique since Acc_a , ι , and $+$ are functions. x is non-negative since the sum of two non-negative numbers is non-negative. As $C \in \mathcal{C}_a$ was arbitrary, we conclude that $(\text{Acc}_a)_l^P : \mathcal{C}_a \rightarrow \mathbb{R}_{\geq 0}$. By Definition 13, $(\text{Acc}_a)_l^P$ is a content accessibility function on $\mathcal{C}_a$. By Definition 22, we conclude that $M_{l,a}^P$ is a multi-agent content priming model. $\square$

Before specifying the operation for similarity-based accessibility update, we show that the similarity function between contents is an increment function:

Proposition 5. *Let CS be a conceptual space built on the set of domains Δ , let $\mathcal{C} \subseteq \mathcal{P}(CS)$ be a set of contents, and suppose that each $C \in \mathcal{C}$ has a prototype $p_C \in CS$. Fix a salience profile W for Δ and a sensitivity parameter $\lambda \in \mathbb{R}_{>0}$. Let Sim be the content similarity function on $\mathcal{C}$ induced by these choices. Then Sim is an increment function on $\mathcal{C}$.*

Proof. Let CS be a conceptual space built on the set of domains Δ , let $\mathcal{C} \subseteq \mathcal{P}(CS)$ be a set of contents, and suppose that each $C \in \mathcal{C}$ has a prototype $p_C \in CS$. Fix a salience profile W for Δ and a sensitivity parameter $\lambda \in \mathbb{R}_{\geq 0}$. Let Sim be the content similarity function on $\mathcal{C}$ induced by these choices. By the Definition 16 of the content similarity function,

$$\text{Sim} : \mathcal{C} \times \mathcal{C} \rightarrow (0, 1].$$

Since $(0, 1] \subseteq \mathbb{R}_0$, $\text{Sim} : \mathcal{C} \times \mathcal{C} \rightarrow \mathbb{R}_{\geq 0}$. Thus, by Definition 14, Sim is an increment function on $\mathcal{C}$. $\square$

We now specify similarity-based accessibility update:

Definition 25 (Similarity-based accessibility update). *Let M be a multi-agent content priming model, $a \in \mathbf{Ag}$ an agent, and $P \in \mathcal{C}_a$ a piece of content. Let Sim_a be the content similarity function on $\mathcal{C}_a$ induced by CS_a , the salience profile W_a , and the sensitivity parameter λ_a . The similarity-based*

accessibility update of M for agent a with P , written $M_{Sim,a}^P$, is defined as the accessibility update of M for agent a with P using Sim_a :

$$M_{Sim,a}^P := M_{Sim_a,a}^P.$$

Finally, the Proposition below shows that applying the similarity-based accessibility update operation to a content priming model yields a content priming model:

Proposition 6. *Let M be a multi-agent content priming model, $a \in \mathbf{Ag}$ an agent, and $P \in \mathcal{C}_a$ a piece of content. Then $M_{Sim,a}^P$ is a multi-agent content priming model.*

Proof. Let Sim_a be the content similarity function on $\mathcal{C}_a$ induced by CS_a , W_a , and λ_a . By Proposition 5, Sim_a is an increment function on $\mathcal{C}_a$. By Definition 25, $M_{Sim,a}^P = M_{Sim_a,a}^P$ is the accessibility update of M for agent a with P using Sim_a , so by Proposition 4 it is a multi-agent content priming model. $\square$

We can now complete our semantics for $\mathcal{L}_P$:

Definition 26 (Satisfaction). *Let $M = (CS_a, \mathcal{C}_a, \sigma_a, \text{Acc}_a, \text{Rel}_a)_{a \in \mathbf{Ag}}$ be a multi-agent content priming model. Then for any formulas $\varphi, \psi \in \mathcal{L}_P$, score terms $t, t' \in T$, content term $P \in \text{Con}$, and agent $a \in \mathbf{Ag}$:*

- $M \models t >_a t' \text{ iff } t_a^M > t_a'^M$;
- $M \models \neg\varphi \text{ iff } M \not\models \varphi$;
- $M \models \varphi \wedge \psi \text{ iff } M \models \varphi \text{ and } M \models \psi$;
- $M \models [P]_a\varphi \text{ iff } M_{Sim,a}^{\sigma_a(P)} \models \varphi$.

Where $\Gamma \subseteq \mathcal{L}$ is a set of formulas and M a multi-agent content priming model, we will write $M \models \Gamma$ to mean that for all $\varphi \in \Gamma$, $M \models \varphi$. When $M \models \Gamma$, we will say that M satisfies Γ .

Lastly, logical consequence is defined in the usual way as truth-preservation in all models between a set of sentences Γ and a sentence φ :

Definition 27 (Logical consequence). *Let $\Gamma \subseteq \mathcal{L}_P$ be a set of formulas and $\varphi \in \mathcal{L}_P$ a formula. We say that φ is a logical consequence of Γ iff for all multi-agent content priming models M , $M \models \Gamma$ implies $M \models \varphi$.*

4.1.3 Executability conditions

We define the formal counterpart of the executability conditions introduced in Section 3.1. Executability conditions will play a crucial role in our semantics for the logic presented in the coming section. Recall that if the executability condition of an action is satisfied, it means that the agent would have performed that action. Similarly to our definition of executability conditions from 3.1, we define separate executability conditions for the fallback action and the other actions. We will also define executability conditions relative to what we call an *action context*:

Definition 28. *Let $a \in \mathbf{Ag}$ be an agent. An action context for agent a is a finite non-empty set of actions $X_a \subseteq \mathcal{A}$ with a designated fallback action $\alpha_X \in X_a$.*

An action context X_a for an agent a represents a context in which agent a has to perform a single action from X . We now formally define executability conditions:

Definition 29 (Executability condition). *Let $a \in \mathbf{A}$ be an agent and $X_a \subseteq \mathcal{A}$ be an action context for agent a , with designated fallback action $\alpha_{X_a} \in X_a$. Let $\alpha \in X_a$ be a non-fallback action. The executability condition of action α for agent a given X_a is denoted by $\text{Exe}_{\alpha}^{X_a}$ and defined as the formula of $\mathcal{L}_P$:²*

$$\text{Exe}_{\alpha}^{X_a} := \bigwedge_{\alpha' \in X_a \setminus \{\alpha, \alpha_{X_a}\}} \text{score}(\alpha) >_a \text{score}(\alpha').$$

Let $\alpha_{X_a} \in X_a$ be the fallback action. The executability condition of α_{X_a} for agent a given X_a is denoted by $\text{Exe}_{\alpha_{X_a}}^{X_a}$ and defined as the formula of $\mathcal{L}_P$:

$$\text{Exe}_{\alpha_{X_a}}^{X_a} := \bigwedge_{\alpha \in X \setminus \{\alpha_{X_a}\}} \neg \text{Exe}_{\alpha}^{X_a}.$$

If distinguishing the executability condition of the fallback action is needed, we will write Fall^{X_a} for $\text{Exe}_{\alpha_{X_a}}^{X_a}$.

We now prove some useful properties of executability conditions. Firstly, the Proposition below says that in any action context, there is always an action whose executability condition is satisfied:

Proposition 7. *Fix an action context $X_a \subseteq \mathcal{A}$ with $X_a = \{\alpha_0, \alpha_1, \dots, \alpha_n\}$ and $\alpha_0 \in X_a$ the designated fallback action. Then $\models \bigvee_{i=0}^n \text{Exe}_{\alpha_i}^{X_a}$.*

Proof. Let $X_a = \{\alpha_0, \alpha_1, \dots, \alpha_n\}$ be an action context with $\alpha_0 \in X_a$ the designated fallback action, and let $M = (CS_b, \mathcal{C}_b, \sigma_b, \text{Acc}_b, \text{Rel}_b)_{b \in \mathbf{Ag}}$ be an arbitrary multi-agent content priming model. To show that $M \models \bigvee_{i=0}^n \text{Exe}_{\alpha_i}^{X_a}$, suppose towards contradiction that $M \not\models \bigvee_{i=0}^n \text{Exe}_{\alpha_i}^{X_a}$. Then $M \models \neg (\bigvee_{i=0}^n \text{Exe}_{\alpha_i}^{X_a})$, which implies that $M \models \bigwedge_{i=0}^n \neg \text{Exe}_{\alpha_i}^{X_a}$. But, notice that $\bigwedge_{i=1}^n \neg \text{Exe}_{\alpha_i}^{X_a} = \bigwedge_{\alpha \in X_a \setminus \{\alpha_0\}} \neg \text{Exe}_{\alpha}^{X_a}$ since $X_a = \{\alpha_0, \alpha_1, \dots, \alpha_n\}$. Thus, by Definition 29, $\bigwedge_{i=1}^n \neg \text{Exe}_{\alpha_i}^{X_a} = \text{Exe}_{\alpha_0}^{X_a}$. This gives us that $M \models \neg \text{Exe}_{\alpha_0}^{X_a} \wedge \text{Exe}_{\alpha_0}^{X_a}$, implying a contradiction. Therefore, $M \models \bigvee_{i=0}^n \text{Exe}_{\alpha_i}^{X_a}$. Since M was an arbitrary multi-agent content priming model, we conclude that $\models \bigvee_{i=0}^n \text{Exe}_{\alpha_i}^{X_a}$. $\square$

The Proposition below says that in any action context, executability conditions are mutually exclusive. In other words, only one action can have its executability condition satisfied at a time:

Proposition 8. *Fix an action context $X_a \subseteq \mathcal{A}$. Then for any action $\alpha \in X_a$, $\models \text{Exe}_{\alpha}^{X_a} \leftrightarrow \bigwedge_{\beta \in X_a \setminus \{\alpha\}} \neg \text{Exe}_{\beta}^{X_a}$.*

Proof. Let X_a be an arbitrary action context with $\alpha_0 \in X_a$ the designated fallback action, $M = (CS_b, \mathcal{C}_b, \sigma_b, \text{Acc}_b, \text{Rel}_b)_{b \in \mathbf{Ag}}$ an arbitrary multi-agent content model, and consider an arbitrary action $\alpha \in X_a$. We consider the cases where $\alpha = \alpha_0$ and $\alpha \neq \alpha_0$. ($\alpha = \alpha_0$). Suppose that $\alpha = \alpha_0$. We need to show that $M \models \text{Exe}_{\alpha_0}^{X_a} \leftrightarrow \bigwedge_{\beta \in X_a \setminus \{\alpha_0\}} \neg \text{Exe}_{\beta}^{X_a}$. By Definition 29 of the executability condition of the fallback action, we get that:

$$M \models \text{Exe}_{\alpha_0}^{X_a} \iff M \models \bigwedge_{\beta \in X_a \setminus \{\alpha_0\}} \neg \text{Exe}_{\beta}^{X_a},$$

from which it follows that $M \models \text{Exe}_{\alpha_0}^{X_a} \leftrightarrow \bigwedge_{\beta \in X_a \setminus \{\alpha_0\}} \neg \text{Exe}_{\beta}^{X_a}$.

²We note that X_a is finite, ensuring that the formula is well-defined.

($\alpha \neq \alpha_0$). Suppose that $\alpha \neq \alpha_0$. For this case, we prove the directions separately. ($\Rightarrow$) Suppose that $M \models \text{Exe}_\alpha^{X_a}$. We need to show that $M \models \bigwedge_{\beta \in X_a \setminus \{\alpha\}} \neg \text{Exe}_\beta^{X_a}$. So take an arbitrary $\beta \in X_a \setminus \{\alpha\}$. Either $\beta = \alpha_0$ or $\beta \neq \alpha_0$. In each case, we show that $M \models \neg \text{Exe}_\beta^{X_a}$. Start by supposing that $\beta = \alpha_0$. Then $M \models \neg \text{Exe}_{\alpha_0}^{X_a}$ since $\text{Exe}_{\alpha_0}^{X_a} = \bigwedge_{\gamma \in X_a \setminus \{\alpha_0\}} \neg \text{Exe}_\gamma^{X_a}$ by Definition 29 and $M \models \text{Exe}_\alpha^{X_a}$. Now suppose that $\beta \neq \alpha_0$. By Definition 29, $M \models \bigwedge_{\gamma \in X_a \setminus \{\alpha, \alpha_0\}} \text{score}(\alpha) >_a \text{score}(\gamma)$ since $M \models \text{Exe}_\alpha^{X_a}$. Thus, since $\beta \neq \alpha_0$ and $\beta \neq \alpha$, $M \models \text{score}(\alpha) >_a \text{score}(\beta)$. By Definition 26, $\text{score}(\alpha)^M >_a \text{score}(\beta)^M$. Thus $\text{score}(\beta)^M \not\geq_a \text{score}(\alpha)^M$, and so by Definition 26 we get $M \not\models \text{score}(\beta) >_a \text{score}(\alpha)$. But then $M \not\models \text{Exe}_\beta^{X_a}$ by Definition 29, and so $M \models \neg \text{Exe}_\beta^{X_a}$. Since each case led to $M \models \neg \text{Exe}_\beta^{X_a}$, we conclude that this holds simpliciter. As $\beta \in X_a$ was arbitrary, we get that $M \models \bigwedge_{\beta \in X_a \setminus \{\alpha\}} \neg \text{Exe}_\beta^{X_a}$, as required. ($\Leftarrow$) Suppose that $M \models \bigwedge_{\beta \in X_a \setminus \{\alpha\}} \neg \text{Exe}_\beta^{X_a}$. By Proposition 7, $M \models \bigvee_{\gamma \in X_a} \text{Exe}_\gamma^{X_a}$. Thus $M \models \text{Exe}_\alpha^{X_a}$, as required. Having shown that $M \models \text{Exe}_\alpha^{X_a}$ iff $M \models \bigwedge_{\beta \in X_a \setminus \{\alpha\}} \neg \text{Exe}_\beta^{X_a}$ in each case, we conclude it holds simpliciter. Thus, we get that $M \models \text{Exe}_\alpha^{X_a} \leftrightarrow \bigwedge_{\beta \in X_a \setminus \{\alpha\}} \neg \text{Exe}_\beta^{X_a}$. As M was an arbitrary multi-agent content priming model, this holds for all content models. By Definition 27, $\models \text{Exe}_\alpha^{X_a} \leftrightarrow \bigwedge_{\beta \in X_a \setminus \{\alpha\}} \neg \text{Exe}_\beta^{X_a}$. $\square$

It follows straightforwardly from Proposition 7 and Proposition 8 that in any action context, there is always a unique action whose executability condition is satisfied. Given the unique satisfiability of executability conditions in action contexts, we introduce the following notation that will be used in our semantics for the language presented in the next section:

Definition 30 (Selected action). *Let $X_a \subseteq \mathcal{A}$ be an action context. For any multi-agent content priming model M , let $\text{sel}_{X_a}(M)$ denote the unique action $\alpha \in X_a$ such that $M \models \text{Exe}_\alpha^{X_a}$.*

4.1.4 SOUP or SOAP, revisited

We now model the SOUP or SOAP scenario presented in Section 3.2.1 using the above language and semantics. We reuse all the components defined in that Section for modeling the SOUP or SOAP scenario. The main difference is that we cast it in a multi-agent content priming model (that, in this case, reduces to a single-agent content priming model) and show how this logic can provide an analysis of why the priming effect of the SOAP or SOAP scenario takes place. We start by fixing the relevant actions. Let the underlying set of actions be $\mathcal{A} = \{\alpha_{\text{SOUP}}, \alpha_{\text{SOAP}}, \alpha_0\}$ and the underlying set of agents be the singleton consisting of Alex $\text{Ag} = \{\text{Alex}\}$, whom we introduced in the SOUP or SOAP scenario. We interpret α_{SOUP} as the action of filling in the word-fragment ‘SO_P’ as ‘SOUP’, α_{SOAP} as the action of filling in the word-fragment ‘SO_P’ as ‘SOAP’, and α_0 is the fallback action, that we, for our analysis here, interpret as the action that Alex does not fill in the word-fragment before the experiment she is a part of ends. We thus model the present action context for Alex X_{Alex} as the underlying set of actions $X_{\text{Alex}} = \mathcal{A}$, thus representing that Alex either has to fill in the fragment SO_P’ as ‘SOUP’, ‘SOAP’, or not fill in the fragment on time. Given the underlying set of actions $\mathcal{A}$, the set of score terms is $T = \{\text{score}(\alpha_{\text{SOUP}}), \text{score}(\alpha_{\text{SOAP}}), \text{score}(\alpha_0)\}$. To talk about the contents stored in Alex’s memory, we define the set of content terms Con as $\text{Con} = \{\text{EAT}, \text{WASH}, \text{SOUP}, \text{SOAP}\}$. While we have not formally specified the content interpretation function for Alex yet, every member of the set $\{\text{EAT}, \text{WASH}, \text{SOUP}, \text{SOAP}\}$ inherits its interpretation from Section 3.2.1. Intuitively, ABC represents the content expressed by the word ‘abc’, with ‘EAT’ and ‘WASH’ interpreted as common nouns as in Section 3.2.1 as well. We now define Alex’s content priming model $M = (CS, \mathcal{C}, \sigma, \text{Acc}, \text{Rel})$, where each component is the one specified for Alex in Section 3.2.1, except for the function σ that we did not have

to specify in that Section: $CS = d_c \times d_t \times d_n \times d_l$ is her conceptual space; $\mathcal{C} = \{\text{EAT}, \text{WASH}, \text{SOUP}, \text{SOAP}\}$ are the contents in her memory; σ_{Alex} interprets each content term above as the corresponding content in $\mathcal{C}$; Acc is her accessibility function, which assigns 0 to every content in $\mathcal{C}$; and Rel is her relevance function, assigning 1 to the pairs $(\alpha_{\text{SOUP}}, \text{SOUP})$ and $(\alpha_{\text{SOAP}}, \text{SOAP})$, and 0 to everything else. Since every formula below concerns Alex, we omit indexing to indicate that Alex is the agent in question throughout. And, since there are no other agents in this scenario, our model M is simply the degenerate multi-agent content priming model consisting only of Alex's content priming model. To ensure that our logic delivers the right analysis of the SOUP or SOAP example, we verify that the following claims hold:

- (1) $M \models [\text{EAT}] \text{Exe}_{\alpha_{\text{SOUP}}}^X$
- (2) $M \models [\text{WASH}] \text{Exe}_{\alpha_{\text{SOAP}}}^X$
- (3) $M \models \text{Fall}^X$

(1) says that the executability condition of the action of filling in the word-fragment 'SO_P' as 'SOUP' is satisfied after Alex processes content EAT. (2) says that the executability condition of the action of filling in the word-fragment 'SO_P' as 'SOAP' is satisfied after Alex processes content WASH. (3) says that if no processing occurs, Alex performs the fallback action, i.e., she does not fill in the word-fragment on time.

Starting with (1). Observe that by Definition 29 of the executability conditions of actions, we get that $\text{Exe}_{\alpha_{\text{SOUP}}}^X = \text{score}(\alpha_{\text{SOUP}}) > \text{score}(\alpha_{\text{SOAP}})$ since $X = \{\alpha_{\text{SOUP}}, \alpha_{\text{SOAP}}, \alpha_0\}$ with α_0 the designated fallback action. Thus, by the satisfaction clause for content processing operators (Definition 26), we get:

$$\begin{aligned} M \models [\text{EAT}] \text{Exe}_{\alpha_{\text{SOUP}}}^X &\iff M \models [\text{EAT}] \text{score}(\alpha_{\text{SOUP}}) > \text{score}(\alpha_{\text{SOAP}}) \\ &\iff M_{\text{Sim}}^{\text{EAT}} \models \text{score}(\alpha_{\text{SOUP}}) > \text{score}(\alpha_{\text{SOAP}}) \\ &\iff \text{score}(\alpha_{\text{SOUP}})^{M_{\text{Sim}}^{\text{EAT}}} > \text{score}(\alpha_{\text{SOAP}})^{M_{\text{Sim}}^{\text{EAT}}}. \end{aligned}$$

We now check that $\text{score}(\alpha_{\text{SOUP}})^{M_{\text{Sim}}^{\text{EAT}}} > \text{score}(\alpha_{\text{SOAP}})^{M_{\text{Sim}}^{\text{EAT}}}$. To compute the interpretations of the score terms, we need to know Alex's relevance function Rel and her updated accessibility function $\text{Acc}_{\text{Sim}}^{\text{EAT}}$ from $M_{\text{Sim}}^{\text{EAT}}$. The relevance function stays the same as in model M , as similarity-based accessibility update only changes content accessibility. The updated accessibility function $\text{Acc}_{\text{Sim}}^{\text{EAT}}$ is given for each $C \in \mathcal{C}$ (Definition 24):

$$\text{Acc}_{\text{Sim}}^{\text{EAT}}(C) = \text{Acc}(C) + \text{Sim}(\text{EAT}, C).$$

In Section 3.2.1, we already compute both $\text{score}(\alpha_{\text{SOUP}})$ and $\text{score}(\alpha_{\text{SOAP}})$ based on the value of $\text{Acc}_{\text{Sim}}^{\text{EAT}}$, $\text{Acc}_{\text{Sim}}^{\text{WASH}}$, and Rel , which are the values that $\text{score}(\alpha_{\text{SOUP}})$ and $\text{score}(\alpha_{\text{SOAP}})$ take in model $M_{\text{Sim}}^{\text{EAT}}$. In particular, we obtain that $\text{score}(\alpha_{\text{SOAP}})^{M_{\text{Sim}}^{\text{EAT}}} \approx 0.427$ and $\text{score}(\alpha_{\text{SOAP}})^{M_{\text{Sim}}^{\text{EAT}}} \approx 0.011$. As such, $\text{score}(\alpha_{\text{SOUP}})^{M_{\text{Sim}}^{\text{EAT}}} > \text{score}(\alpha_{\text{SOAP}})^{M_{\text{Sim}}^{\text{EAT}}}$, thus verifying (1). We now move on to the verification of (2)

The verification of (2) proceeds similarly to the verification of (1), except that we need to check that $\text{score}(\alpha_{\text{SOAP}})^{M_{\text{Sim}}^{\text{EAT}}} > \text{score}(\alpha_{\text{SOUP}})^{M_{\text{Sim}}^{\text{EAT}}}$. And, similarly to the verification of (1), we simply refer back to Section 3.2.1 to see that $\text{score}(\alpha_{\text{SOAP}})^{M_{\text{Sim}}^{\text{EAT}}} \approx 0.497$ and $\text{score}(\alpha_{\text{SOUP}})^{M_{\text{Sim}}^{\text{EAT}}} \approx 0.008$. As such, $\text{score}(\alpha_{\text{SOAP}})^{M_{\text{Sim}}^{\text{EAT}}} > \text{score}(\alpha_{\text{SOUP}})^{M_{\text{Sim}}^{\text{EAT}}}$, thus verifying (2).

We now verify (3). Recall that Fall^X abbreviates the executability condition of the fallback action α_0 in action context X . By Definition 29, this condition is satisfied just in case none of the other actions in X are executable. Since

$$X = \{\alpha_{\text{SOUP}}, \alpha_{\text{SOAP}}, \alpha_0\},$$

we get that

$$\begin{aligned} M \models \text{Fall}^X &\iff M \models \text{Exe}_{\alpha_0}^X \\ &\iff M \models \neg \text{Exe}_{\alpha_{\text{SOUP}}}^X \wedge \neg \text{Exe}_{\alpha_{\text{SOAP}}}^X. \end{aligned}$$

We therefore need to show that neither the executability condition of α_{SOUP} nor α_{SOAP} is satisfied in M . We know from the verification above that the respective executability conditions are $\text{score}(\alpha_{\text{SOUP}}) > \text{score}(\alpha_{\text{SOAP}})$ for α_{SOUP} and $\text{score}(\alpha_{\text{SOAP}}) > \text{score}(\alpha_{\text{SOUP}})$ for α_{SOAP} . But, since Acc of M assigns a content accessibility of 0 to every piece of content in $\mathcal{C}$ of M , we have by Definition 23 of score term interpretation that $\text{score}(\alpha_{\text{SOUP}})^M = 0$ and $\text{score}(\alpha_{\text{SOAP}})^M = 0$, giving us that neither $(\alpha_{\text{SOUP}})^M > \text{score}(\alpha_{\text{SOAP}})^M$ nor $(\alpha_{\text{SOAP}})^M > \text{score}(\alpha_{\text{SOUP}})^M$ is true. Hence $M \models \neg \text{Exe}_{\alpha_{\text{SOUP}}}^X \wedge \neg \text{Exe}_{\alpha_{\text{SOAP}}}^X$, and therefore, by Definition 29, $M \models \text{Fall}^X$, thus verifying (3).

Having shown that our logic can handle the SOUP or SOAP example, we proceed to the second logic.

4.2 Belief dynamics

In this section, we expand the previous machinery to capture how priming effects can influence belief formation. The central idea is that priming does not change belief directly. Rather, priming can change which action is selected, and its execution can provide agents with new information they can use to update their beliefs. The first of these steps is already in place, as the logic of the previous section captures how processing content can change which action an agent performs. What remains is the second step, in which agents update their beliefs with new information in light of observed behavior. We model this second step using the radical upgrade operation from the Dynamic Epistemic Logic tradition [Ben07; BS15; BR16; BS06; SBS21].

Before specifying our logic, we start by introducing a device that records what new information agents update their beliefs with when observing behavior. We introduce this device now, as it will figure in the formulas of our new language.

Definition 31 (Informational-effect function). *Let $X_a \subseteq \mathcal{A}$ be an action context and define $\mathcal{X} := \{(X_a, \alpha) : \alpha \in X_a\}$. An informational-effect function on X_a is a function $u : \mathcal{X} \rightarrow \mathcal{L}_{BP}$ assigning to each pair $(X, \alpha) \in \mathcal{X}$, a formula $u(X, \alpha) \in \mathcal{L}_{BP}$. Given an action context X_a , the set of all informational-effect functions on X_a is denoted by $\mathbf{u}(X_a)$.*

Where u is an informational-effect function for some agent b in action context X_a , the formula $u(X_a, \alpha)$ represents the new information that b update their beliefs with after action α is executed in action context X_a . As an example, suppose we have two agents called Alex and Ben, an action context X_b for Ben, and consider the action $\alpha_{\text{slow}} \in X_b$ of Ben walking slowly, and the atomic sentence ‘ o_b ’ interpreted as ‘Ben is old’. If we want to represent Alex updating her beliefs with the information

that Ben is old after observing Ben’s slow walking, we can use the informational effect function u and set $u(X_a, \alpha_{slow}) = o_b$, where u represents how Alex updates her belief with new information when she observes certain actions. In particular, $u(X_a, \alpha_{slow}) = o_b$ means that Alex updates with the information that Ben is old when Ben walks slowly, given action context X_a . It is important to note that informational-effect functions do not tell us *how* agents update their beliefs with new information. What such functions tell us is *what* new information agents update their beliefs with. We will cover the issue of how agents update their beliefs with new information in Section 4.2.3. Where X_a is an action context, we will say that ‘ X_a is executed’ to mean that the unique action whose executability condition is satisfied in X_a is executed.

4.2.1 Syntax

The language $\mathcal{L}_{BP}$ extends $\mathcal{L}_P$ with propositional atoms, a belief operator, and dynamic operators for capturing how agents update their beliefs with new information gained from observed behavior, with that behavior caused by underlying levels of content accessibility:

$$\varphi ::= p \in \Phi \mid t > t' \mid \neg\varphi \mid \varphi \wedge \varphi \mid B_a\varphi \mid [P]_a\varphi \mid [X_a]_u^b\varphi,$$

where Φ is a countable set of atomic sentences, $t, t' \in T$ are score terms, $P \in Con$ is a content term, $a, g \in Ag$ are agents, X_a is an action context, and u is any informational-effect function on X_a . The intended readings of the new sentence types are:

- $B_a\varphi$: agent a believes that φ ;
- $[X_a]_u^b\varphi$: after X_a is executed and b updates their beliefs with the new information prescribed by u , φ .

4.2.2 Doxastic content priming models

We now define the models used for the expanded language. They will be called *multi-agent doxastic content priming models* and extend content priming models with three components. The three components make up part of what is called a multi-agent plausibility model [BS15; BS16; BDM08; Ben07; BS06; BR16].³ Plausibility models are used to represent and reason about a multitude of ways for agents to update their beliefs in light of new information.⁴ Importantly, plausibility models allow for representing how agents update their beliefs with “soft information”, where soft information can be understood as information that is reliable enough to warrant incorporation into one’s belief system, but not so authoritative as to be irrevocable [Ben07; BS15]. Formally, content priming models are extended by a structure $(S, \preceq_a, V)_{a \in Ag}$ where S is a non-empty set of states, or so-called “possible worlds”, for each agent $a \in Ag$, $\preceq_a \subseteq S \times S$ is a plausibility relation over the states in S , where for any $w, u \in S$, ‘ $u \preceq_a w$ ’ reads ‘agent a considers w at least as plausible as u ’, and $V : S \rightarrow \mathcal{P}(\Phi)$ is a valuation function assigning to each state $s \in S$ the atomic sentences $V(s)$ true at that state.⁵ As is standard, we impose the condition that plausibility relations $\preceq_a$ are locally well-preordered [BR16; Ben07; BS15].

³Besides the three components we will use, multi-agent plausibility models typically involve a so-called *epistemic indistinguishability* relation for each agent $a \in Ag$, denoted by $\sim_a$ [BS06; BS16]. We do not include these in our models as they do not bear on our exposition.

⁴For an introduction, see [BS15].

⁵Again, for a more thorough exposition on these components, we refer the reader to [BS15].

This means that they are reflexive, transitive, locally connected, and conversely well-founded. While reflexivity and transitivity are well-known properties of binary relations, we briefly explain local connectedness and converse well-foundedness. In specifying the property of local connectedness, we rely on Baltag and Renne [BR16], and in specifying the property of converse well-foundedness, we rely on Baltag et. al. [BFS26]. Fix a set of states S and a plausibility relation $\preceq_a \subseteq S \times S$ on S for agent $a \in \mathbf{Ag}$. Let $w \in S$ be any state. The connected component of w is denoted by $\text{cc}(w)$ and defined as the set of states $\text{cc}(w) := \{v \in S : w(\preceq_a \cup \succsim_a)^* v\}$ with $\succsim_a := \{(u, z) \in S \times S : z \preceq_a u\}$ and $*$ denoting the reflexive-transitive closure of a binary relation. Then, $\preceq_a$ is locally connected if and only if for all $w, u \in S$, if $u \in \text{cc}(w)$, then either $w \preceq_a u$ or $u \preceq_a w$ [BR16]. If we think of $\preceq_a$ geometrically as specifying “arrows” between states such that there is an arrow pointing from state w to state u if and only if $w \preceq_a u$, then the property of local connectedness says that whenever there is an undirected arrow-path between any two states w and u , then there is a direct arrow between w and u in one direction or the other [BR16]. Moving on to the property of converse well-foundedness. First, define the strict plausibility relation $\prec_a$ for any states $u, w \in S$ by $u \prec_a w$ and not $w \preceq_a u$. Thus, $u \prec_a w$ means that agent a considers w strictly more plausible than u . Then $\preceq_a$ is conversely well-founded if and only if there are no infinite ascending chains $s_0 \prec_a s_1 \prec_a s_2 \prec_a \dots \prec_a s_m \prec_a \dots$ of strictly more plausible states [BFS26]. Following Baltag et. al. [BFS26], we define, for any set non-empty $S' \subseteq S$, $\text{Max}_{\preceq_a} S' := \{w \in S' : w \not\prec_a u \text{ for any } u \in S'\}$, where ‘ $u \not\prec_a w$ ’ means that it is not the case that $u \prec_a w$. The set of states $\text{Max}_{\preceq_a} S'$ consists of the maximally plausible states for agent a among the states in S' [BFS26]. As noted by Baltag et. al. [BFS26], the above conditions imposed on plausibility relations ensure that for every non-empty set of states $S' \subseteq S$, $\text{Max}_{\preceq_a} S'$ is non-empty.

As will be formally specified in our semantics below, we will follow standard treatments of belief using plausibility models [Ben07, page 10; BS15, page 309] and model agents as believing what is the case across the worlds that they consider maximally plausible. We now extend content priming models with the above components of plausibility models:

Definition 32 (Multi-agent doxastic content priming model). *A multi-agent doxastic content priming model is a tuple*

$$M = (CS_a, \mathcal{C}_a, \sigma_a, \text{Acc}_a, \text{Rel}_a, S, \preceq_a, V)_{a \in \mathbf{Ag}}$$

where:

- $(CS_a, \mathcal{C}_a, \sigma_a, \text{Acc}_a, \text{Rel}_a)$ is a content priming model for agent $a \in \mathbf{Ag}$;
- S is a non-empty set of states;
- $\preceq_a \subseteq S \times S$ is a locally well-preordered plausibility relation on S for agent $a \in \mathbf{Ag}$;
- $V : S \rightarrow \mathcal{P}(\Phi)$ is a valuation function

and we say that $M_a = (CS_a, \mathcal{C}_a, \sigma_a, \text{Acc}_a, \text{Rel}_a, S, \preceq_a, V)$ is a doxastic content priming model for agent, where $M_a = (CS_a, \mathcal{C}_a, \sigma_a, \text{Acc}_a, \text{Rel}_a)$ is agent a ’s content priming model and $(S, \preceq_a, V)$ is agent a ’s plausibility model.

Where $M_a = (CS_a, \mathcal{C}_a, \sigma_a, \text{Acc}_a, \text{Rel}_a, S, \preceq_a, V)$ is a doxastic content priming model for an agent a , the components of M_a are used in the following way. The model has two layers. The first layer is the

content priming model part $M_a = (CS_a, \mathcal{C}_A, \sigma_a, \text{Acc}_a, \text{Rel}_a)$. We call this the *sub-doxastic* layer, or part, of the model. The reason for this is that this layer is not used in representing the beliefs of agents. Rather, it is the part of the model used in determining accessibility-based action selection, and the part changed by accessibility updates induced by the processing of content. The second layer is the *doxastic* layer $(S, \preceq_a, V)$. This layer is used to represent the beliefs of the agent.

4.2.3 Model transformations and truth clauses

In the previous logic, we made precise the effects of content processing on content accessibility through the model-changing operations of Definitions 24 and 25. While these definitions apply to content priming models (Definition 22), they lift to doxastic content priming models in the obvious way: where $M = (CS_a, \mathcal{C}_a, \sigma_a, \text{Acc}_a, \text{Rel}_a, S, \preceq_a, V)_{a \in \text{Ag}}$ is a multi-agent doxastic content priming model, we change the content priming part of the model, as specified by Definitions 24 and 25, while leaving the rest unchanged. Furthermore, the selected action notation introduced in Definition 30 also extends to multi-agent doxastic content priming models in the obvious way. Lastly, the interpretation of score terms (Definition 23) in doxastic content priming models is handled in the same way as in content priming models. Given the above, we will readily use the notation of Definitions 24, 25, 30, and 23 for multi-agent doxastic content priming models without further specification.

Before making precise the informational effects of observed behavior, we clarify what we mean by *observed behavior*. We are interested in what an agent takes behavior as *indicating* about themselves or another agent, as opposed to the ontic, or factual changes, caused by that behavior. For example, if an agent observes someone walking slowly, they may update their beliefs with the new information that the person is old, in case they take walking slowly as an indication of old age. This type of information has two important features. Firstly, different agents may interpret the same behavior differently. Secondly, behavior is not a perfect guide to a person's qualities or features. An agent may walk slowly because they are old, but may also walk slowly because they are tired, distracted, or injured. Thus, this type of information is soft information as explained earlier, in that it is information that is reliable enough to be incorporated into one's belief system, but not so authoritative as to be irrevocable.

For this reason, we model the updating of one's beliefs by the new information gained through observed behavior by *radical upgrade* (see van Benthem and Smets [BS15, pages 316–319]). Using radical upgrade in this context amounts to treating new information gained through observed behavior as strongly indicative of certain features of the agent being observed. Here is how we adapt the information update of radical upgrade, as presented on [BS15, pages 316–319], to the present context: When an agent updates with some new information (represented by a formula φ) that they have gained through observing some behavior, they update their plausibility relation such that all states that make φ true become more plausible than all states making φ false. In this sense, the observer takes the behavior largely at face value and allows it to guide their beliefs.⁶

In stating the formal definition for radical upgrade, we utilize the following notation (adapted from [BS15]): Where M is any doxastic content priming model with the set of states S and $\varphi \in \mathcal{L}_{BP}$ any

⁶Nothing in our general setup forces us to use radical upgrade as opposed to another type of information update. If observed behavior is treated more cautiously, our dynamic operators could instead be formulated in terms of a conservative upgrade (see van Benthem and Smets [BS15, page 318]). If the information extracted from behavior is treated as hard information, it could be formulated in terms of a public announcement (see van Benthem and Smets [BS15, pages 304–306]). For the present purposes, however, a radical upgrade seems to be the most natural choice, as it lets the observer come to believe what the behavior indicates for them, while still preserving the possibility that the behavior was misleading.

formula, let $\|\varphi\|_M := \{s \in S : (M, s) \models \varphi\}$ and $\overline{\|\varphi\|}_M := S \setminus \|\varphi\|_M$. When M is clear from context, we omit the subscript and write $\|\varphi\|$ and $\overline{\|\varphi\|}$.

We now present the formal definition of radical upgrade. Following Baltag et. al. [BFS26], we define radical upgrade as:

Definition 33 (Radical upgrade (adapted from [BFS26])). *Let $M = (CS_a, \mathcal{C}_a, \sigma_a, \text{Acc}_a, \text{Rel}_a, S, \preceq_a, V)_{a \in \text{Ag}}$ be a multi-agent doxastic content priming model, $b \in \text{Ag}$ an agent, and $\varphi \in \mathcal{L}_{BP}$ a formula. The radical upgrade of M for agent b with φ , denoted by $M^{\uparrow_b \varphi}$, is the multi-agent doxastic content priming model that differs from M only in agent b 's plausibility relation, which becomes $\preceq_b^{\uparrow \varphi}$, defined for all $u, w \in S$ by:*

$$\begin{aligned} u \preceq_b^{\uparrow \varphi} w \quad \text{iff} \quad & u \notin \|\varphi\|_M \text{ and } w \in \|\varphi\|_M; \text{ or} \\ & u, w \in \|\varphi\|_M \text{ and } u \preceq_b w; \text{ or} \\ & u, w \notin \|\varphi\|_M \text{ and } u \preceq_b w. \end{aligned}$$

We now specify the semantic clauses for the language $\mathcal{L}_{BP}$.

Definition 34 (Satisfaction). *Let $M = (CS_a, \mathcal{C}_a, \sigma_a, \text{Acc}_a, \text{Rel}_a, S, \preceq_a, V)_{a \in \text{Ag}}$ be a multi-agent doxastic content priming model and let $s \in S$. Satisfaction of $\mathcal{L}_{BP}$ -formulas at s is defined as follows:*

- $(M, s) \models p$ iff $p \in V(s)$;
- $(M, s) \models t >_a t'$ iff $t_a^M > t_a'^M$;
- $(M, s) \models \neg \varphi$ iff $(M, s) \not\models \varphi$;
- $(M, s) \models \varphi \wedge \psi$ iff $(M, s) \models \varphi$ and $(M, s) \models \psi$;
- $(M, s) \models B_a \varphi$ iff $(M, w) \models \varphi$ for all $w \in \text{Max}_{\preceq_a} S$;
- $(M, s) \models [P]_a \varphi$ iff $(M_{\text{Sim}, a}^{\sigma_a(P)}, s) \models \varphi$;
- $(M, s) \models [X_a]_u^b \varphi$ iff $(M^{\uparrow_b u(X_a, \text{sel}_{X_a}(M))}, s) \models \varphi$;

where $p \in \Phi$, $a, b \in \text{Ag}$, $P \in \text{Con}$, X_a is an action context, and u is an informational-effect function on X_a .

Definition 35 (Logical consequence). *Let $\Gamma \cup \varphi \subseteq \mathcal{L}_{BP}$. We say that φ is a logical consequence of Γ , written $\Gamma \models \varphi$, iff for every multi-agent doxastic content priming model M and every state $s \in S$ of M , if $(M, s) \models \gamma$ for every $\gamma \in \Gamma$, then $(M, s) \models \varphi$.*

4.2.4 Putting the machinery to use

We now use the newly constructed logic $\mathcal{L}_{BP}$ to analyze a scenario in which a content priming effect induces a false belief.⁷ Firstly, we present the scenario. Then, we explain how the scenario is a case of content priming. Lastly, we show that the logic $\mathcal{L}_{BP}$ delivers the correct analysis of the scenario. The scenario is as follows:

⁷It is important to note that this is only one of many ways in which content priming effects can influence belief formation. We leave a thorough analysis of the different ways in which content priming effects can influence belief formation for future work.

Bob is a student in Ann’s 9 am Metalogic class. Before class, Bob participates in a study testing for the Florida effect, although Bob is not aware that this is what the study is about. As part of the study, he completes a sentence-scrambling task containing the word ‘forgetful’. Afterward, Bob walks into the auditorium, where Ann is already sitting. Before Bob enters the room, Ann has no belief about whether Bob is tired. Ann observes Bob walking unusually slowly into the room. Since it is their 9 am class, and since Ann knows Bob from class and does not take slow walking to be typical of him, she takes his behavior as indicating that he is tired. She thereby forms the belief that Bob is tired. In fact, Bob is not tired: his slow walking was induced by the prime. Had Bob not participated in the study before class, he would have walked at his normal pace, and Ann would still have no belief about whether he is tired.

In summary, we have an acting agent, called Bob, and an observing agent, called Ann. The stimulus that changes Bob’s behavior is the sentence-scrambling task containing the word ‘forgetful’. This initial stimulus leads Bob to walk slowly into the auditorium as opposed to walking at his normal pace. Thus, it influences Bob’s response to the subsequent task of walking into the auditorium. Ann then observes Bob’s behavior and updates her beliefs about Bob on that basis. We now formally analyze the scenario using $\mathcal{L}_{BP}$. We start by specifying the underlying set of agents $\mathbf{Ag}$, set of actions $\mathcal{A}$, set of content terms $\mathbf{Con}$, atomic sentences Φ , the action context X_{Bob} of Bob having to walk into the auditorium, and the informational effect function u on X_{Bob} that specifies what new information Ann gains by observing Bob perform each individual action in X_{Bob} . Then, we construct an appropriate multi-agent doxastic content priming model M for representing the scenario, where we re-use some components from our analysis of the Florida Effect scenario from Section 3.2.2. Lastly, we show that M satisfies sentences that correctly describe the above scenario. In particular, where $@$ is the actual world of model M , we show that:

- (4) $(M, @) \models [X_{\text{Bob}}]_u^{\text{Ann}} \neg B_{\text{Ann}} t$
- (5) $(M, @) \models [X_{\text{Bob}}]_u^{\text{Ann}} \neg B_{\text{Ann}} \neg t$
- (6) $(M, @) \models [\text{FORGETFUL}]_{\text{Bob}} [X_{\text{Bob}}]_u^{\text{Ann}} B_{\text{Ann}} t$
- (7) $(M, @) \models \neg t$

Here are what sentences (4) through (7) capture. (4) and (5) jointly capture that Ann has no beliefs about Bob’s tiredness if he enters the auditorium without having done any prior processing. In particular, (4) says that Ann does not believe that Bob is tired after observing him walk into the auditorium and updating her beliefs accordingly. (5) says that Ann does not believe that Bob is not tired after observing him walk into the auditorium and updating her beliefs accordingly. (6) and (7) jointly capture that Ann comes to falsely believe that Bob is tired after observing him walk into the auditorium if he processes FORGETFUL before entering. In particular, (6) says that after Bob processes the content FORGETFUL and Ann *then* observes him walking into the auditorium and updates her beliefs accordingly, she comes to believe that Bob is tired. But, (7) says that Bob is not tired.

We take our underlying set of agents to be $\mathbf{Ag} = \{\text{Bob}, \text{Ann}\}$ and the underlying set of actions to be $\mathcal{A} = \{\alpha_{\text{slow}}, \alpha_0\}$, where α_{slow} is the action of Bob walking slowly into the auditorium, and α_0 is the action of Bob walking into the auditorium at a normal pace. Bob’s action context for walking into the auditorium is $X_{\text{Bob}} = \{\alpha_{\text{slow}}, \alpha_0\}$, with designated fallback action α_0 . We set our set of content

terms to $Con = \{\text{FORGETFUL}, \text{OLD}\}$. Furthermore, we will work with the single atomic sentence $t \in \Phi$ which we read as ‘Bob is tired’. Lastly, we define the informational-effect function u on X_{Bob} as $u(X_{\text{Bob}}, \alpha_{\text{slow}}) = t$ and $u(X_{\text{Bob}}, \alpha_0) = \top$, where $\top$ is any tautology of $\mathcal{L}_{BP}$. That is, Ann takes Bob’s unusually slow walking to indicate that he is tired ($u(X_{\text{Bob}}, \alpha_{\text{slow}}) = t$), while she takes his normal walking as indicating nothing ($u(X_{\text{Bob}}, \alpha_0) = \top$).

We now specify a multi-agent doxastic content priming model M for representing the above scenario. We intentionally leave some parts of M unspecified, as they do not bear on the scenario. In particular, we leave the doxastic part of Bob’s doxastic content priming model M_{Bob} of M unspecified, since the scenario only describes Bob’s behavior and how it is primed. For Ann’s doxastic content priming model M_{Ann} of M , we leave the sub-doxastic part unspecified since the scenario only describes her beliefs about Bob. Starting with the sub-doxastic part of Bob’s doxastic content priming model $(CS_{\text{Bob}}, \mathcal{C}_{\text{Bob}}, \sigma_{\text{Bob}}, \text{Acc}_{\text{Bob}}, \text{Rel}_{\text{Bob}})$, we take Bob’s conceptual space to be the space defined for our earlier analysis of the Florida Effect in Section 3.2.2, i.e., $CS_{\text{Bob}} = d_m \times d_h \times d_s$. Recall that d_m is a quality dimension representing memory function, d_h is a quality dimension representing hair coverage, and d_s is a quality dimension representing skin smoothness. We model the contents stored in Bob’s LTM as the set $\mathcal{C}_{\text{Bob}} = \{\text{FORGETFUL}, \text{OLD}\}$, with $\text{FORGETFUL} = [0, 0.3] \times [0, 1] \times [0, 1]$ and $\text{OLD} = [0.1, 0.4] \times [0.1, 0.4] \times [0.05, 0.35]$ defined as from Section 3.2.2.⁸ We set the content interpretation function σ_{Bob} to $\sigma_{\text{Bob}}(\text{FORGETFUL}) = \text{FORGETFUL}$ and $\sigma_{\text{Bob}}(\text{OLD}) = \text{OLD}$, in which the foregoing arguments for σ_{Bob} are the content terms in Con and the values specified for these arguments are the contents in $\mathcal{C}_{\text{Bob}}$.⁹ We set Bob’s accessibility function Acc_{Bob} to 0 for both pieces of content in $\mathcal{C}_{\text{Bob}}$. Lastly, we set Bob’s relevance function Rel_{Bob} to 1 for the pair $(\alpha_{\text{slow}}, \text{OLD})$, and to 0 to all other action-content pairs.¹⁰ What is left to specify are the components of M we will use in representing the doxastic part of Ann’s doxastic content priming model M_{Ann} , namely $(S, \preceq_{\text{Ann}}, V)$. In particular, we will use the set of states $S = \{\@, s_t\}$, where $\@$ is the actual state and s_t is the state where Bob is tired. To reflect this, we set the valuation function V to $V(\@) = \emptyset$ and $V(s_t) = \{t\}$, thus indicating that t is false at $\@$ and that t is true at s_t . Lastly, we define Ann’s plausibility relation by $\@ \preceq_{\text{Ann}} s_t$ and $s_t \preceq_{\text{Ann}} \@$.¹¹ As such, Ann considers the state where Bob is tired as equally plausible to the state where Bob is not tired. This captures the fact that Ann has no reason to rank one state above the other before Bob enters the room. Having specified all the relevant components of M , we proceed with verifying sentences (4) through (7).

Starting with (7). Since $V(\@) = \emptyset$, it follows trivially that $(M, \@) \models \neg t$. Moving on to (4). Specifically, we need to verify $(M, \@) \models [X_{\text{Bob}}]_u^{\text{Ann}} \neg B_{\text{Ann}} t$. We proceed in three steps. First, we compute the selected action in action context X_{Bob} of model M , i.e., we compute $\text{sel}_{X_{\text{Bob}}}(M)$. Then, we look at what new information Ann gains by observing the execution of action $\text{sel}_{X_{\text{Bob}}}(M)$ as prescribed by the informational effect function u , i.e., we look at what formula $u(X_{\text{Bob}}, \text{sel}_{X_{\text{Bob}}}(M))$ denotes. Afterwards, we perform a radical upgrade with $u(X_{\text{Bob}}, \text{sel}_{X_{\text{Bob}}}(M))$ for Ann, i.e., we make all worlds satisfying $u(X_{\text{Bob}}, \text{sel}_{X_{\text{Bob}}}(M))$ more plausible than worlds satisfying $\neg u(X_{\text{Bob}}, \text{sel}_{X_{\text{Bob}}}(M))$ in Ann’s plausibility order. This operation yields the new model $M^{\uparrow_{\text{Ann}} u(X_{\text{Bob}}, \text{sel}_{X_{\text{Bob}}}(M))}$, in which we need to check whether $(M^{\uparrow_{\text{Ann}} u(X_{\text{Bob}}, \text{sel}_{X_{\text{Bob}}}(M))}, \@) \models \neg B_{\text{Ann}} t$. To find $\text{sel}_{X_{\text{Bob}}}(M)$, start by noticing that $\text{Acc}_{\text{Bob}}(\text{OLD}) = 0$, which in this case implies that $\text{score}(\alpha_{\text{slow}})_{\text{Bob}}^M = 0$. As such, no

⁸For a justification of these modeling choices, we refer the reader back to Section 3.2.2.

⁹Once again, we abuse notation for the sake of simplicity.

¹⁰Once again, we refer the reader back to Section 3.2.2 for a justification of these choices.

¹¹Here, we use $u \preceq_{\text{Ann}} w$ as a shorthand for $(u, w) \in \preceq_{\text{Ann}}$, for any $u, w \in S$.

candidate action has a strictly higher selection score than all other actions. Thus, the executability condition of the fallback action is satisfied, i.e., $M, @ \models \text{Exe}_{\alpha_0}^{X_{\text{Bob}}}$, giving us that $\text{sel}_{X_{\text{Bob}}}(M) = \alpha_0$. As such, we have that $u(X_{\text{Bob}}, \text{sel}_{X_{\text{Bob}}}(M)) = u(X_{\text{Bob}}, \alpha_0) = \top$. And, since $\top$ is a tautology, we get that $\preceq_{\text{Ann}} = \preceq_{\text{Ann}}^{\uparrow \top} = \preceq_{\text{Ann}}^{u(X_{\text{Bob}}, \alpha_0)}$. Thus $@ \preceq_{\text{Ann}}^{u(X_{\text{Bob}}, \alpha_0)} s_t$ and $s_t \preceq_{\text{Ann}}^{u(X_{\text{Bob}}, \alpha_0)} @$, since $@ \preceq_{\text{Ann}} s_t$ and $s_t \preceq_{\text{Ann}} @$. Now observe that $\text{Max}_{\preceq_{\text{Ann}}^{u(X_{\text{Bob}}, \alpha_0)}} S = \{ @, s_t \}$, since $@ \preceq_{\text{Ann}}^{u(X_{\text{Bob}}, \alpha_0)} s_t$ and $s_t \preceq_{\text{Ann}}^{u(X_{\text{Bob}}, \alpha_0)} @$ with $S = \{ @, s_t \}$. Thus, by the satisfaction clause for belief of Definition 26, we obtain that $(M^{\uparrow \text{Ann } u(X_{\text{Bob}}, \alpha_0)}, @) \not\models B_{\text{Ann}} t$ since $V(@) = \emptyset$ and $V(s_t) = \{t\}$, and so $(M^{\uparrow \text{Ann } u(X_{\text{Bob}}, \alpha_0)}, @) \models \neg B_{\text{Ann}} t$. As such, we conclude that $(M, @) \models [X_{\text{Bob}}]_u^{\text{Ann}} \neg B_{\text{Ann}} t$. Verifying (5) proceeds similarly to the verification of (4).

Lastly, we verify (6), namely that $(M, @) \models [\text{FORGETFUL}]_{\text{Bob}} [X_{\text{Bob}}]_u^{\text{Ann}} B_{\text{Ann}} t$. By the satisfaction clause for content-processing operators of Definition 26, we thus need to show that $(M_{\text{Sim}, \text{Bob}}^{\sigma_{\text{Bob}}(\text{FORGETFUL})}, @) \models [X_{\text{Bob}}]_u^{\text{Ann}} B_{\text{Ann}} t$ where $M_{\text{Sim}, \text{Bob}}^{\sigma_{\text{Bob}}(\text{FORGETFUL})}$ is the model obtained from M by applying the similarity-based accessibility update induced by Bob's processing of $\text{FORGETFUL} \in \mathcal{C}_{\text{Bob}}$, as specified by Definition 25. As a notational convenience, we set $M_f := M_{\text{Sim}, \text{Bob}}^{\sigma_{\text{Bob}}(\text{FORGETFUL})}$. To show $(M_f, @) \models [X_{\text{Bob}}]_u^{\text{Ann}} B_{\text{Ann}} t$, we start by finding the formula denoted by $\text{sel}_{X_{\text{Bob}}}(M_f)$. The computation of $\text{sel}_{X_{\text{Bob}}}(M_f)$ is similar to the computation done in Case 1 for showing that the executability condition of α_{slow} is satisfied in our analysis of the Florida Effect in Section 3.2.2. The reason is that we re-used all the relevant components from that analysis in M . In particular, this gives us that the executability condition of α_{slow} is satisfied in M_f at $@$, i.e., $(M_f, @) \models \text{Exe}_{\alpha_{\text{slow}}}^{X_{\text{Bob}}}$, and so $\text{sel}_{X_{\text{Bob}}}(M_f) = \alpha_{\text{slow}}$. Thus $u(X_{\text{Bob}}, \text{sel}_{X_{\text{Bob}}}(M_f)) = u(X_{\text{Bob}}, \alpha_{\text{slow}}) = t$. We now perform a radical upgrade for Ann with $u(X_{\text{Bob}}, \text{sel}_{X_{\text{Bob}}}(M_f)) = u(X_{\text{Bob}}, \alpha_{\text{slow}}) = t$, thus yielding the updated model $M_f^{\uparrow \text{Ann } t}$ where $@ \preceq_{\text{Ann}}^{\uparrow t} s_t$ while it is not the case that $s_t \preceq_{\text{Ann}}^{\uparrow t} @$. This gives us that $\text{Max}_{\preceq_{\text{Ann}}^{\uparrow t}} S = \{s_t\}$. Thus, since S is the set of states in model $M_f^{\uparrow \text{Ann } t}$ and $(M_f, s_t) \models t$ as $V(s_t) = \{t\}$, we have by the satisfaction clause for belief of Definition 26 that $(M_f^{\uparrow \text{Ann } t}, @) \models B_{\text{Ann}} t$. Thus, since $u(X_{\text{Bob}}, \text{sel}_{X_{\text{Bob}}}(M_f)) = u(X_{\text{Bob}}, \alpha_{\text{slow}}) = t$, we have that $(M_f^{\uparrow \text{Ann } u(X_{\text{Bob}}, \text{sel}_{X_{\text{Bob}}}(M_f))}, @) \models B_{\text{Ann}} t$. By the satisfaction clause for our belief-update operators of Definition 26, it follows that $(M_f, @) \models [X_{\text{Bob}}]_u^{\text{Ann}} B_{\text{Ann}} t$. By the satisfaction clause for content-processing operators of Definition 26, it follows that $(M, @) \models [\text{FORGETFUL}]_{\text{Bob}} [X_{\text{Bob}}]_u^{\text{Ann}} B_{\text{Ann}} t$, as required.

Having verified claims (4) through (7), we have shown that the above scenario can be analyzed using the logic $\mathcal{L}_{BP}$.

Chapter 5

Conclusion

5.1 Summary

This thesis has developed a formal framework for representing and reasoning about content priming effects. By enriching the conceptual spaces framework with ways of capturing accessibility increases induced by processing content, as well as developing a model of how accessibility levels of content can affect behavior, which in turn can affect belief formation, we have made some first steps towards treating content priming effects using logical tools.

In Chapter 2, we started by introducing the basic machinery of the conceptual spaces framework, followed by our contribution of a model of accessibility updates induced by content processing. In Section 2.1, we presented the notions of a dimension, a domain, and a conceptual space. In Section 2.2, we presented some commonly used distance measures for points in a conceptual space and showed how these distance measures can be used to model similarities between points. Then, in Section 2.3, we showed how properties and concepts can be represented as regions in conceptual spaces, and how prototypes can be obtained for different representations of properties and concepts, or without some pre-defined region in a conceptual space. Finally, in Section 2.4, we presented the notion of a content accessibility function and an increment function, and used these to define a general accessibility update operation. We then gave a specific interpretation of this operation, leading to our notion of similarity-based accessibility update, where processing a piece of content increases the accessibility of other contents in proportion to their similarity. In spelling out this notion, prototypes played a pivotal role, since similarity between contents is defined in terms of similarity between their prototypes.

In Chapter 3, we connected the accessibility-update machinery developed in Chapter 2 to behavior. In Section 3.1, we introduced a model of accessibility-based action selection. To devise this model, we introduced the notion of a relevance function to capture what content is relevant to the selection of what action, as well as the notion of the selection score of an action. The score of an action is determined by the accessibility levels of the contents relevant to that action. In Sections 3.2.1 and 3.2.2, we applied the framework to two scenarios based on two canonical examples of priming effects presented by Daniel Kahneman in [Kah11]. The analyses of these scenarios showed that our framework can handle canonical cases of content priming.

In Chapter 4, we cast the tools developed in Chapter 2 and Section 3.1 within a formal logic. In Section 4.1, we devised a multi-agent logic for reasoning about how processing a piece of content can increase content accessibility, and formalized the notion of executability conditions of actions developed in Section 3.1. We then analyzed the SOUP or SOAP scenario from Section 3.2.1 using this logic. In

Section 4.2, we extended the previous logic with belief operators, as well as operators for capturing the informational effects of observed behavior, where that behavior is brought about by underlying degrees of content accessibility. Then, in Section 4.2.4, we put this machinery to use and analyzed a scenario involving two agents that displayed how content priming can affect belief formation.

5.2 Future work

There are several natural directions for future work. One concerns the logical properties of the systems introduced in Chapter 4. Our emphasis has been on developing a framework for representing and reasoning about content priming effects, and using it in analyzing examples of content priming, as opposed to exploring the properties of the two logical systems themselves. Thus, a natural next step would be to study the proof theory of the logics systematically. In particular, one could formulate reduction principles for the dynamic operators, use these to develop sound and complete axiomatizations, and compare the resulting systems with familiar dynamic logics for belief revision [BS15; BS16; BDM08].

A second direction concerns using the framework to analyze more complex multi-agent scenarios. While the scenario that was presented and analyzed in Section 4.2.4 was a multi-agent scenario that showed how content priming can have consequences for the beliefs of agents, it is still quite a simple example. Future work could use the framework to analyze cases where several agents observe the same behavior but interpret it differently, or where agents prime each other.

A third direction concerns the accessibility-update mechanism itself. In this thesis, we used a similarity-based accessibility update as our preferred type of accessibility update. However, this is just a starting point. Future work can explore different mechanisms for deriving these content accessibility increases induced by content processing that still use the underlying conceptual space over which the content is defined. And, there are already many different formal ways of specifying relations between contents in a conceptual space, such as “fuzzy” subset-hood and more [BK17b].

A fourth direction concerns extending the framework to be able to handle further cognitive notions that may interact with content accessibility. For example, future work could investigate how to combine the logics presented in this thesis with logics for cognitive notions that already have a logical treatment, such as attention [BBW25; Bel+25; Bel23] awareness [vV10; FH87; Fer21], or logical frameworks for reasoning about other cognitive biases, such as framing effects [BÖ23], or frameworks for reasoning about agents with limited cognitive resources [SBS21]. Another option is to make things like salience profiles dynamic, allowing content processing not only to increase the accessibility of contents, but also to change which dimensions or domains are weighted most heavily in later processing.

Bibliography

- [Att50] Fred Attneave. “Dimensions of Similarity”. In: *The American Journal of Psychology* 63.4 (1950), pages 516–556 (cited on page 12).
- [BBW25] Gaia Belardinelli, Thomas Bolander, and Sebastian Watzl. *A Logic of General Attention Using Edge-Conditioned Event Models (Extended Version)*. 2025. <https://arxiv.org/abs/2505.14539> (cited on page 61).
- [BC00] John A. Bargh and Tanya L. Chartrand. “The mind in the middle: A practical guide to priming and automaticity research”. In: *Handbook of research methods in social and personality psychology*. Edited by Heinrich T. Reis and Chris M. Judd. New York: Cambridge, 2000, pages 253–285 (cited on page 3).
- [BCB96] John A. Bargh, Mark Chen, and Lara Burrows. “Automaticity of social behavior: direct effects of trait construct and stereotype-activation on action.” In: *Journal of personality and social psychology* 71 2 (1996), pages 230–44. <https://api.semanticscholar.org/CorpusID:6654763> (cited on page 39).
- [BDM08] A. Baltag, H. P. van Ditmarsch, and L. S. Moss. “Epistemic Logic and Information Update”. In: *hilosophy of Information*. Edited by P. Adriaans and J. van Benthem. MIT Press, 2008 (cited on pages 53, 61).
- [Bel+25] Gaia Belardinelli, Lei Li, Sonja Smets, and Anthia Solaki. “Logics for Personalized Announcements and Attention Dynamics”. In: *Review of Analytic Philosophy* 5 (Jan. 2025), page 1. <https://doi.org/10.18494/SAM.RAP.2025.0020> (cited on page 61).
- [Bel23] Gaia Belardinelli. “Epistemic Logic Models of Awareness, Attention, and Implicit Reasoning”. PhD thesis. University of Copenhagen, June 2023, page 155 (cited on page 61).
- [Ben07] Johan van Benthem. “Dynamic Logic for Belief Revision”. In: *Journal of Applied Non-Classical Logics* 17.2 (2007), pages 129–155. <https://doi.org/10.3166/janc1.17.129-155> (cited on pages 52–54).
- [BFS26] Alexandru Baltag, Virginie Fiutek, and Sonja Smets. “Suspending Judgement: Belief Contraction in Dynamic Epistemic Logic”. In: *Proceedings of the 23rd International Conference on Principles of Knowledge Representation and Reasoning (KR 2026)*. Forthcoming. Lisbon, Portugal, 2026 (cited on pages 54, 56).
- [BK17a] Lucas Bechberger and Kai-Uwe Kühnberger. “A Thorough Formalization of Conceptual Spaces”. In: *KI 2017: Advances in Artificial Intelligence*. Edited by Gabriele Kern-Isberner, Johannes Fürnkranz, and Matthias Thimm. Cham: Springer International Publishing, 2017, pages 58–71. ISBN: 978-3-319-67190-1 (cited on pages 7–12, 14, 16, 18, 20, 21, 28, 33).

- [BK17b] Lucas Bechberger and Kai-Uwe Kühnberger. “Measuring Relations Between Concepts In Conceptual Spaces”. In: *CoRR* abs/1707.02292 (2017). <http://arxiv.org/abs/1707.02292> (cited on pages 4, 7–12, 14, 16, 18–21, 26–28, 61).
- [BÖ23] Francesco Berto and Aybüke Özgün. “The Logic of Framing Effects”. In: *Journal of Philosophical Logic* 52.3 (2023), pages 939–962. <https://doi.org/10.1007/s10992-022-09694-0> (cited on page 61).
- [Bol18] Thomas Bolander. “Seeing is Believing: Formalising False-Belief Tasks in Dynamic Epistemic Logic”. In: *Jaakko Hintikka on Knowledge and Game Theoretical Semantics*. Edited by Hans van Ditmarsch and Gabriel Sandu. Springer, 2018, pages 207–236 (cited on page 31).
- [BR16] Alexandru Baltag and Bryan Renne. “Dynamic Epistemic Logic”. In: *The Stanford Encyclopedia of Philosophy*. Edited by Edward N. Zalta. Winter 2016. Metaphysics Research Lab, Stanford University, 2016 (cited on pages 30, 31, 52–54).
- [BS06] Alexandru Baltag and Sonja Smets. “Dynamic belief revision over multi-agent plausibility models”. In: *Proceedings of LOFT* (Jan. 2006) (cited on pages 31, 52, 53).
- [BS15] Johan van Benthem and Sonja Smets. “Dynamic Logics of Belief Change”. In: *Handbook of Epistemic Logic*. Edited by Hans van Ditmarsch, Joseph Y. Halpern, Wiebe van der Hoek, and Barteld Kooi. London: College Publications, 2015, pages 313–394. ISBN: 9781848901582 (cited on pages 52–55, 61).
- [BS16] Alexandru Baltag and Sonja Smets. “A Qualitative Theory of Dynamic Interactive Belief Revision”. In: *Readings in Formal Epistemology: Sourcebook*. Edited by Horacio Arló-Costa, Vincent F. Hendricks, and Johan van Benthem. Cham: Springer International Publishing, 2016, pages 813–858. ISBN: 978-3-319-20451-2. https://doi.org/10.1007/978-3-319-20451-2_39. https://doi.org/10.1007/978-3-319-20451-2_39 (cited on pages 53, 61).
- [DHK07] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic Epistemic Logic*. Dordrecht, Netherland: Springer, 2007 (cited on page 31).
- [Dou+22a] Igor Douven, Shira Elqayam, Peter Gärdenfors, and Patricia Mirabile. “Conceptual Spaces and the Strength of Similarity-Based Arguments”. In: *Cognition* 218.C (2022), page 104951. <https://doi.org/10.1016/j.cognition.2021.104951> (cited on page 7).
- [Dou+22b] Igor Douven, Shira Elqayam, Peter Gärdenfors, and Patricia Mirabile. “Conceptual spaces and the strength of similarity-based arguments”. In: *Cognition* 218 (2022), page 104951. ISSN: 0010-0277. <https://doi.org/https://doi.org/10.1016/j.cognition.2021.104951>. <https://www.sciencedirect.com/science/article/pii/S0010027721003747> (cited on page 12).
- [EK20] M.W. Eysenck and M.T. Keane. *Cognitive Psychology: A Student’s Handbook*. Taylor & Francis, 2020. ISBN: 9781351058506. <https://books.google.dk/books?id=LErpDwAAQBAJ> (cited on page 26).
- [Fer21] Claudia Fernández-Fernández. *Awareness in Logic and Epistemology: A Conceptual Schema and Logical Study of the Underlying Main Epistemic Concepts*. Berlin: Springer Verlag, 2021 (cited on page 61).

- [FH87] Ronald Fagin and Joseph Y. Halpern. “Belief, awareness, and limited reasoning”. In: *Artificial Intelligence* 34.1 (1987), pages 39–76. ISSN: 0004-3702. [https://doi.org/https://doi.org/10.1016/0004-3702\(87\)90003-8](https://doi.org/https://doi.org/10.1016/0004-3702(87)90003-8). <https://www.sciencedirect.com/science/article/pii/0004370287900038> (cited on page 61).
- [Gär00] Peter Gärdenfors. *Conceptual Spaces: The Geometry of Thought*. 55 Hayward St., Cambridge, MA, United State: MIT Press, 2000 (cited on pages 4, 6–13, 16–18, 20, 23, 24).
- [Gär88] Peter Gärdenfors. “Semantics, Conceptual Spaces and the Dimensions of Music”. In: *Essays on the philosophy of music*. Edited by Veikko Rantala, Lewis Eugene Rowell, and Eero Tarasti. Akateeminen Kirjakauppa, 1988, pages 9–27 (cited on page 6).
- [Gär90] Peter Gärdenfors. “Induction, Conceptual Spaces and Ai”. In: *Philosophy of Science* 57.1 (1990), pages 78–95. <https://doi.org/10.1086/289532> (cited on page 6).
- [Gär96] Peter Gärdenfors. “Mental Representation, Conceptual Spaces and Metaphors”. In: *Synthese* 106.1 (1996), pages 21–47. <https://doi.org/10.1007/bf00413612> (cited on page 6).
- [Gro20] Richard D Gross. *Psychology [electronic book] : the science of mind and behaviour / Richard Gross*. eng. 8th ed. London: Hodder Education, 2020. ISBN: 9781510468450 (cited on pages 25, 26).
- [GW12] Peter Gärdenfors and Massimo Warglien. “Using Conceptual Spaces to Model Actions and Events”. In: *Journal of Semantics* 29 (Nov. 2012), pages 487–519. <https://doi.org/10.1093/jos/ffs007> (cited on pages 8, 32).
- [HS12] N. Hsu and Z. Schütt. *Psychology of Priming*. Perspectives on cognitive psychology. Nova Science Publishers, 2012. ISBN: 9781621003649. <https://books.google.dk/books?id=IFxHXwAACAAJ> (cited on page 3).
- [Joh01] Mikael Johannesson. *The Problem of Combining Integral and Separable Dimensions*. Technical Report HS-IDA-TR-01-002. University of Skövde, School of Humanities and Informatics, 2001 (cited on page 12).
- [JW14] Chris Janiszewski and Robert S. Wyer. “Content and process priming: A review”. In: *Journal of Consumer Psychology* 24.1 (2014), pages 96–118. ISSN: 1057-7408. <https://doi.org/https://doi.org/10.1016/j.jcps.2013.05.006>. <https://www.sciencedirect.com/science/article/pii/S105774081300082X> (cited on pages 3, 4, 29, 31, 33).
- [Kah11] Daniel Kahneman. *Thinking, Fast and Slow*. New York: New York: Farrar, Straus and Giroux, 2011 (cited on pages 3, 31, 39, 60).
- [LL16] Martha Lewis and Jonathan Lawry. “Hierarchical conceptual spaces for concept combination”. In: *Artificial Intelligence* 237 (2016), pages 204–227. ISSN: 0004-3702. <https://doi.org/https://doi.org/10.1016/j.artint.2016.04.008>. <https://www.sciencedirect.com/science/article/pii/S0004370216300492> (cited on pages 24, 25).

- [LT09] Jonathan Lawry and Yongchuan Tang. “Uncertainty modelling for vague concepts: A prototype theory approach”. In: *Artif. Intell.* 173 (Dec. 2009), pages 1539–1558. <https://doi.org/10.1016/j.artint.2009.07.006> (cited on page 24).
- [Mol14] Daniel Molden. “Understanding Priming Effects in Social Psychology: What is “Social Priming” and How does it Occur?”. In: *Social Cognition* 32 (June 2014), pages 1–11. <https://doi.org/10.1521/soco.2014.32.suppl.1> (cited on page 3).
- [MV25] Michael Mugno and Timothy Vickery. “Retrieval from long-term memory does not bypass working memory”. In: *Attention, Perception, & Psychophysics* 87 (Aug. 2025). <https://doi.org/10.3758/s13414-025-03145-z> (cited on page 26).
- [PG17] Daniele Porello and Giancarlo Guizzardi. “Towards a Cognitive Semantics of Type”. In: *AI*IA 2017 Advances in Artificial Intelligence - XVIth International Conference of the Italian Association for Artificial Intelligence, Bari, Italy, November 14-17, 2017, Proceedings. Lecture Notes in Computer Science 10640*. Edited by Daniele Porello and Giancarlo Guizzardi. 2017, pages 428–440 (cited on pages 7–10, 18).
- [PM16] Daniele Porello and Claudio Masolo. “Understanding Predication in Conceptual Spaces”. In: *Formal Ontology in Information Systems. Proceedings of the Ninth International Conference (FOIS 2016)*. Edited by Roberta Ferrario and Werner Kuhn. IOS Pres, 2016, pages 139–152 (cited on pages 7–11, 18).
- [Rei18] Daniel Reisberg. *Cognition: Exploring the Science of the Mind (Seventh Edition)*. W. W. Norton & Company, 2018. ISBN: 978-0393624137 (cited on pages 3, 26).
- [SBS21] Anthia Solaki, Francesco Berto, and Sonja Smets. “The Logic of Fast and Slow Thinking”. In: *Erkenntnis* (1975-) 86.3 (2021), pages 733–762. ISSN: 01650106, 15728420. <http://www.jstor.org/stable/45378172> (visited on 02/28/2026) (cited on pages 33, 52, 61).
- [She64] Roger N. Shepard. “Attention and the metric structure of the stimulus space”. In: *Journal of Mathematical Psychology* 1.1 (1964), pages 54–87. ISSN: 0022-2496. [https://doi.org/https://doi.org/10.1016/0022-2496\(64\)90017-3](https://doi.org/https://doi.org/10.1016/0022-2496(64)90017-3). <https://www.sciencedirect.com/science/article/pii/0022249664900173> (cited on page 12).
- [Smi68] C. R. Smith. “A Characterization of Star-Shaped Sets”. In: *The American Mathematical Monthly* 75.4 (1968), pages 386–386. ISSN: 00029890, 19300972. <http://www.jstor.org/stable/2313423> (visited on 05/06/2026) (cited on page 21).
- [vV10] Johan van Benthem and Fernando R. Velázquez-Quesada. “The dynamics of awareness”. In: *Synthese* 177 (2010), pages 5–27. ISSN: 00397857, 15730964. <http://www.jstor.org/stable/40985730> (visited on 03/04/2026) (cited on page 61).
- [WGW12] Massimo Warglien, Peter Gärdenfors, and Matthijs Westera. “Event structure, conceptual spaces and the semantics of verbs”. In: *Theoretical Linguistics* 38 (Jan. 2012). <https://doi.org/10.1515/tl-2012-0010> (cited on pages 32, 33).
- [Xio25] Chenxi Xiong. “The Priming Effect: Unveiling Its Multifaceted Impacts on Decision-Making and Behavior”. In: *Highlights in Business, Economics and Management* 58 (June 2025), pages 292–296. <https://doi.org/10.54097/1yk91m43> (cited on page 3).
- [YHT15] Melvin Yap, Keith Hutchison, and Luuan Tan. *Individual Differences in Semantic Priming Performance: Insights from the Semantic Priming Project*. Feb. 2015 (cited on page 3).