Formalizing verbal theories: A tutorial by dialogue

Iris van Rooij
Radboud University, The Netherlands
Donders Institute for Brain, Cognition and Behaviour
Department of Artificial Intelligence

Mark Blokpoel
Radboud University, The Netherlands
Donders Institute for Brain, Cognition and Behaviour
The Language in Interaction Consortium

We present a tutorial for formalizing verbal theories of psychological phenomena—social or otherwise. The approach builds on concepts and tools from the mathematics of computation. We use intuitive examples and illustrate the intrinsic dialectical nature of the formalization process by presenting dialogues between two fictive characters, called *Verbal* and *Formal*. These characters' conversations and thought experiments serve to highlight important lessons in theoretical modeling.

This paper has been published: van Rooij, I., & Blokpoel, M. (2020). Formalizing verbal theories: A tutorial by dialogue. *Social Psychology 51*(5), 285-298, 10.1027/1864-9335/a000428.

Keywords: theoretical modeling, formalization, verbal theory, computational explanation

Introduction

Theoretical modeling is like sculpting (Blokpoel, 2018). There is no fixed procedure. It requires creativity and the occasional courage to try something new. While a sculptor may have a general idea of what type of sculpture they want to make, it is by looking at intermediate states that they decide how to proceed; e.g., chisel away a piece of rock, add some more clay, or start anew. They only know that the sculpture (analogously, theoretical model) is done when they see it meets their standards.

In this tutorial, we illustrate theoretical modeling as a dialogue between two fictive characters, *Verbal* and *Formal*, who are engaged in this type of sculpting process. These characters represent complementary cognitive states of mind required for successful theoretical modeling. Both characters bring raw materials to work with. *Verbal*, on the one hand, brings intuitive ideas about the 'what', 'how' and 'why' of psychological phenomena based on empiricial observations and domain knowledge. *Formal*, on the other hand, brings formal concepts and tools from mathematics and knowledge about computational principles. Together, *Verbal* and *Formal* sculpt (and polish) theoretical models of psychological phenomena. In reality, an individual modeler can take on both roles intermittedly or, alternatively, the two styles of thinking can be distributed over a pair or a team of scientists.

Fitting to the aims and scope of *Social Psychology*, we focus on social psychological phenomena for our illustrations. That said, the approach that we adopt has wide applicability in psychology and can be used to sculpt theoretical models of cognitive, developmental, linguistic and evolutionary phenomena as well (Blokpoel et al., 2019; Otworowska, Blokpoel, Sweers, Wareham, & van Rooij, 2018; Rich et al., 2019; Rich, Blokpoel, de Haan, & van Rooij, 2020; van Rooij & Baggio, 2020). Fitting to the didactical aims of this article, we furthermore focus on "toy" social psychological phenomena that are complex enough to be interesting to model, yet simple enough to be accessible for the beginning modeler. By using toy examples we also hope that it is easy enough for social psychologists to suspend judgement on *Verbal*'s intuitions. No claim of realism is made; examples serve illustrative and didactical purposes only.

The remainder of this tutorial is organized as follows. We start with a brief primer on the conceptual foundations of the theoretical modeling approach. Next, we introduce the necessary mathematical concepts and notational conventions. The body of the tutorial consists of dialogues in which *Verbal* and *Formal* sculpt various theoretical models of the example phenomena. We close by highlighting the lessons learnt and the relationship to other modeling approaches.

Preliminaries

Conceptual foundations

The theoretical modeling approach that we adopt here builds on the philosophical view that psychology's primary targets of explanation are behavioural tendencies or dispositions, also knows as *capacities* (Cummins, 1985, 2000). As van Rooij and Baggio (2020) put it:

> A capacity is a dispositional property of a system at one of its levels of organization: e.g., single neurons have capacities (firing, exciting, inhibiting) and so do minds and brains (vision, learning, reasoning) and groups of people (coordination, competition, polarization). A capacity is a more or less reliable ability

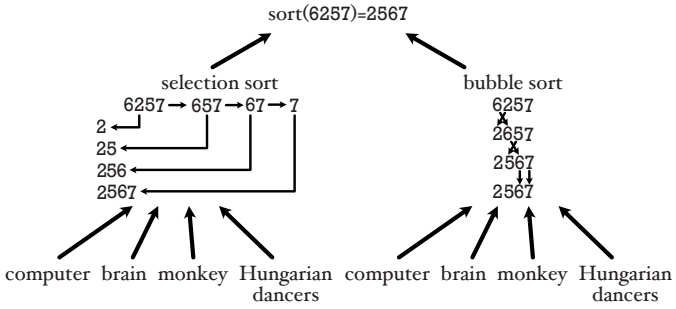


Figure 1. Any given function can be computed by different algorithms, and any given algorithm can be physically realized in different ways. This principle is illustrated for the sorting example.

> (or disposition or tendency) to transform some initial state (input) into a resulting state (output).

Take, for instance, the social-cognitive capacity for *goal inference* (Baker, Tenenbaum, & Saxe, 2007; Blokpoel, Kwisthout, van der Weide, Wareham, & van Rooij, 2013): Given observations of a person's behaviours one can make inferences about the (presumed) goals of a person. In this case the *input* is the observed behaviour (e.g., 'Jake buys flowers') and background knowledge (e.g., 'Yesterday, Jake broke Cass's car'), and the *output* is an inferred goal (e.g., 'Jake wants to apologize to Cass'). Alternatively, consider the capacity for *causal attribution* (De Houwer & Moors, 2015; Ross, 1977): given observations of a person's behaviours one can make attributions of whether the underlying causes lie in person traits and/or situational factors. In this case the *input* is again the observed behaviour (e.g., 'Jake buys flowers') and some background knowledge but the *output* is a causal attribution in terms of person traits (e.g., 'Jake bought flowers because he is an attentive person') or in terms of situational factors (e.g., as in the *goal inference* example above, or otherwise).

According to the influential tri-level framework proposed by David Marr (1982) capacities can be analyzed at three different levels: the computational level, the algorithmic level, and the implementational level. At the computational level, we ask the question, 'what is the function (or problem) being computed by the capacity?' At the algorithmic level, we ask: 'how is the function computed (by what algorithm)?' And finally, at the implementational level, we ask: 'how is the algorithm physically realized?' An important feature of this framework is that lower levels of explanation are underdetermined, though constrained, by the higher levels of explanation: A function can, in principle, be computed by different algorithms; and any given algorithm can, in principle, be physicially realized in different ways (Fig. 1).

Let's illustrate these ideas using a capacity called *sorting* (e.g., one can order people from youngest to oldest, order choice options from least to most preferred, etc.). We will adopt the convention that a computational-level model can be represented as follows:

NAME OF MODELLED CAPACITY
Input: Specification of the input.
Output: Specification of the output as a function of the input.

For the capacity *sorting*, this looks as follows:

SORTING
Input: A list of unordered numbers L
Output: An ordered list L' that consists of the elements in L.

In other words, here we stipulate that $L' = \textsc{Sorting}(L)$. For instance, if L is `6 2 5 7 3 9` then L' is `2 3 5 6 7 9`.

The SORTING function can be computed by different algorithms. For instance, one strategy can be to consider each item, from left to right, to find the smallest element in the list L, and put it in position 1 of list L'. Then repeat this for the remainder of the numbers in L, and put the next smallest number in position 2 in list L'; and so on, until one has filled up list L' using the elements in L. A different strategy, however, could be to order the numbers in L by 'swapping' adjancent numbers: i.e., consider the numbers in position 1 and 2 in L, and if the second number smaller than the first then swap the two numbers. Repeat for the numbers in position 2 and 3, positions 3 and 4, and so on. Then repeat the whole procedure starting again at position 1, and repeat until no more swaps can be made.

Both algorithms, called `selection sort` and `bubble sort` respectively (Knuth, 1968), compute the SORTING function. Besides these two algorithms there exists a host of different sorting algorithms, all of which compute exactly the same function, though their timing profiles may differ.[1] Like the SORTING function can be computed by different algorithms, each algorithm can be realized in different physical ways. For instance, a sorting algorithm can be physically realized by a computer or a brain, or even as a distributed group activity (e.g., by people walking though a maze (van Rooij &

[1] For a visual and auditory illustration of 15 distinct sorting algorithms see https://youtu.be/kPRA0W1kECg.

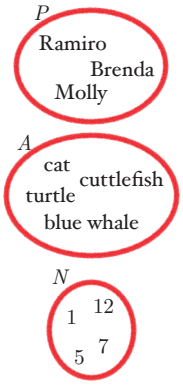
Baggio, 2020), or by a group of Hungarian dancers[2]).

Following Marr (1982), we adopt a top-down strategy. We focus on the computational level, but we will have some things to say on algorithmic and implementational level considerations insofar as they constrain computational-level theorizing (van Rooij, 2008). The computational level of analysis is specifically useful for social psychology as it seems most directly relevant to explaining social behaviour and, as some have argued, it is the most *behaviourally penetrable* (see De Houwer & Moors, 2015). For social psychologists interested in figuring out algorithmic and/or implementational details, having computational level models on offer can help guide the search into lower levels of explanation (Mitchell, 2006; Read & Miller, 1998; Thagard & Kunda, 1998).

Mathematical concepts and notation

Like sculpting, theoretical modeling requires its own set of dedicated tools. The theoretical modeler's tools are mathematical concepts, formal expressions, and notational conventions. As you will see, one can already get quite far with the basics in *set theory*, *functions* and *logic*. Below we present a brief primer. Readers who have taken introductory classes on these topics can skip this section without loss of contuinity. If, however, these materials are new to you, then we recommend carefully studying this section before proceeding. A good grasp of the concepts and notation defined here will be necessary for following the dialogues in the next section. In general, developing some fluency in mathematical language is key if one wants to formalize one's own verbal theories.

Sets. A set is a collection of distinct objects. For example, a set of people, animals or numbers. Sets are usually denoted by a capital letter and their elements listed between curly brackets. They can also be visualized as circles.



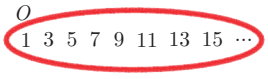
P = {Ramiro, Brenda, Molly}

A = {cat, turtle, blue whale, cuttlefish}

N = {1, 5, 7, 12}

When we want to refer to the number of objects in a set, we write $|P|$ which is called the *cardinality* of the set (in this case $|P| = 3$).

Sets can contain an infinite number of objects, e.g. all positive odd numbers:



O = {1, 3, 5, 7, . . . }

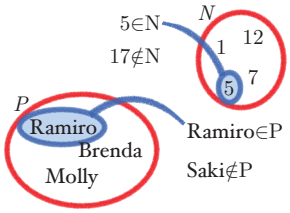
When we want to write that an object x is (or is not) part of a set X, we use *set membership* notation:

$5 \in N$

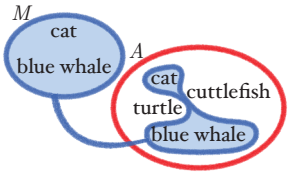
$17 \notin N$

Ramiro $\in P$

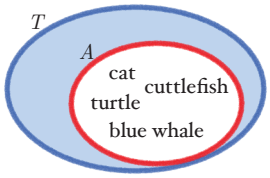
Saki $\notin P$



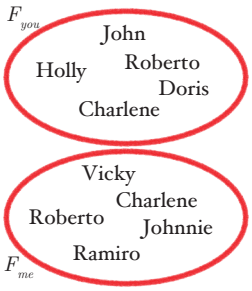
Often, we want to express things like 'the set of mammals M is part of the set of all animals A'. We then use *subset* notation: $M \subseteq A$ or $M \subset A$. The latter means that M is smaller than A.



Vice versa, we can express that 'the set of all things on earth T contains all animals A' using *superset* notation: $T \supseteq A$ or $T \supset A$. The latter means that T is bigger than A.



Let's look at what more we can do with two sets. For example, take the set of your friends and my friends.

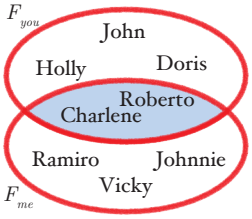


F_{you} = {John, Roberto, Holly, Doris, Charlene}

F_{me} = {Vicky, Charlene, Ramiro, Johnnie, Roberto}

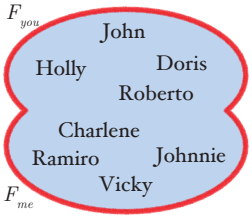
Who are our common friends? We use *set intersection*:

$F_{you} \cap F_{me}$ = {Roberto, Charlene}



Who do we know together? We use *set union*:

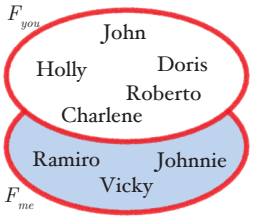
$F_{you} \cup F_{me}$ = {John, Roberto, Holly, Doris, Charlene, Vicky, Ramiro, Johnnie}



[2] See https://www.youtube.com/watch?v=lyZQPjUT5B4.

Who do I know that you do not know? We use *set difference*:

$$F_{me} \setminus F_{you} = \{\text{Vicky, Ramiro, Johnnie}\}$$

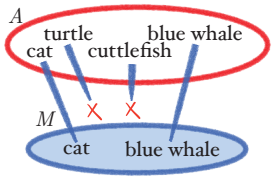


A more advanced way to denote sets is to define a set using *set builder* notation. This allows us to define (build) a new set given other set(s). A set builder consists of two parts, a variable and a logical predicate:

{variable | predicate}

Let's look at an example and build a set of all mammals from A. We explain logical predicates below, for now let's use verbal language:

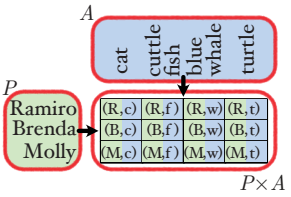
$$M = \{a \mid a \in A \text{ and } a \text{ is a mammal}\}$$



You can read this as 'M contains all a's *with the property that* $a \in A$ and a is a mammal'.

This notation is useful to filter objects from a single set, but becomes very potent when building from multiple sets. For example:

$$F = \{(p, a) \mid p \in P \text{ AND } a \in A\}$$



Read this as 'F contains all pairs of p and *a with the property that* p is a person and a is an animal'. Pairs are denoted in brackets. You can think of F containing all possible combinations of person-animal pairs. For example, these are all the options you have when trying to guess what the favorite animals are of your friends.

Many other set builders are possible too, but this specific 'pair builder' is called the *cardinal product* of two sets. It is used often enough that it has its own special symbol: $F = P \times A$.

Finally, there are some special sets which we often use that have their own symbols:

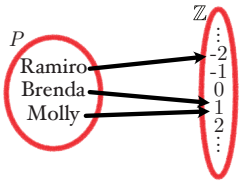
- Empty set $\varnothing = \{\}$
- Natural (whole) numbers (with zero) $\mathbb{N}_0 = \{0, 1, 2, 3, 4, \dots\}$
- Natural (whole) numbers (without zero) $\mathbb{N}^* = \{1, 2, 3, 4, \dots\}$
- Integer numbers $\mathbb{Z} = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$
- Real numbers $\mathbb{R} = \{r \mid -\infty < r < \infty\}$

Functions. Building on set theory, we can define functions. Functions are relations that map all objects from one set (the *domain*) to exactly one object from another set (the *codomain*). We define functions with the following notation, here f is the name of the function:

$$f : D \to C \text{ with } f(d) = c$$

Let's make this more concrete:

$$like : P \to \mathbb{Z}$$

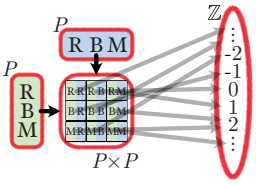


You can read this as '*like* is a function that maps persons $p \in P$ to an integer'.

We sometimes omit the exact specification of the function when it is clear what it would be. For example, here it could be a list of numbers representing how much you (dis)like the person, based on your social interactions with that person. We can also give functions more complex domains by using set theory.

What would a function that captures how much two persons like each other look like?

$$like_2 : P \times P \to \mathbb{Z}$$



The cardinal product $P \times P$ denotes all pairs of persons and $like_2$ maps pairs to an integer.

Lastly, we define summation and product. These functions iterate over members in a set and return a summary value.

- Summation $\sum$ takes all x's from X, applies $f(x)$ to each and adds all values $\sum_{x \in X} f(x) = f(x_1) + f(x_2) + f(x_3) + \dots$
- Product $\prod$ takes all x's from X, applies $f(x)$ to each and multiplies all values $\prod_{x \in X} f(x) = f(x_1) f(x_2) f(x_3) \dots$

Logic. Predicates can be thought of as a special type of function that returns a Boolean value true or false. Predicates can be thought of as asking or claiming whether or not a statement is true or false.

For example, is x bigger than 2? Is a a mammal and small? Is Emily your friend? Or, x is bigger than 2, a is a mammal and small, and Emily is my friend.

Let's introduce some formal notation to express these statements:

- *number comparisons* are familiar to most $<$, $\leq$, $>$, $\geq$, $=$, and $\neq$
- *conjunctions* (logical AND) $p \wedge q$ is true if and only if both p = true and q = true
- *disjunctions* (logical OR) $p \vee q$ is true if p = true or if q = true
- *set membership* can also be used as a predicate $a \in A$ is true if a is a member of set A

Sometimes we want to say something about all objects in a set. We can use quantifier predicates to do this. For example: all animals in the set are mammals. We use the *universal quantifier*:

$$\forall_{a \in A} \text{mammal}(a)$$

You can read this as 'it hold for all objects a in A that a is a mammal' We implicitly introduced a function mammal : $A \rightarrow \{\text{true}, \text{false}\}$ with mammal(a) = true if a is a mammal or false otherwise.

Another type of statement is: there exists someone I know that I like. We use the *existential quantifier* to express this:

$$\exists_{p \in F_{me}} [\text{like}(p) > 0]$$

We can read this as 'there exists a person p in the set of my friends F_{me} for which I like them like(p) > 0'

Finally, there is a tight relation between logic and set theory. Suppose we think of set A as the set of objects with property A and set B as the objects with property B. Then the intersection $A \cap B$ is the set of objects that have property A *and* B: a conjunction! And the union $A \cup B$ is the set of objects that have property A *or* B: a disjuntion! In fact, set intersection can be called set conjunction, and set union can be called set disjunction.

Theoretical Modeling Illustrations

In this section we illustrate theoretical modeling by dialogues between the two fictive characters, *Verbal* and *Formal*. We use two toy social psychological phenomena: (1) a host inviting people to a party, and (2) party guests self-organizing into smaller subgroups. These example scenarios can be seen as special cases of the more general capacities for *social decision-making* and *social group formation*. We present multiple dialogues. *Verbal*'s intuitions and *Formal*'s questions vary from dialogue to dialogue, and hence the developed theoretical models differ as well.

Dialogue 1: Formalizing inviting guests

V: I'd like to explain how a host decides whom to invite to a party.

F: Why would the host not invite everybody?

V: They may like some people but dislike others.

F: Then the host invites everybody they like?

V: Not all people get along. If people get into an argument that can spoil a party.

F: I see. So a host may choose to invite people they like *and* that all get along.

V: Yes, that sounds right. I think that's what a host will tend to do. Can we formalize this idea?

F: Let's see. We can start by defining the initial state (input) of the decision process as consisting of the set of people that the host chooses from. Let us denote this set as $P = \{p_1, p_2, ..., p_n\}$. Further, let's define two subsets, $L \subseteq P$ and $D \subseteq P$, denoting the subsets of people our host likes and dislikes respectively.

F: I assume you had in mind that a person cannot be both liked *and* disliked by the host, and that any given person is either liked or disliked by the host.

V: Let's indeed assume that for now.

F: Then L and D form a *partition* of P. To be precise, $L \cap D = \emptyset$ (a person cannot be both liked and disliked) and $L \cup D = P$ (any person is either liked or disliked).

F: To formalize that different pairs of people in P can also like or dislike each other, let us define a 'liking' function $like : P \times P \rightarrow \{true, false\}$ that specifies for each pair of persons $(p_i, p_j) \in P$ whether or not p_i and p_j like each other.

V: That function *like* is merely a notational device to assign 'true' or 'false' for pairs of persons in P to indicate whether or not they like each other?

F: Correct.

F: To formalize the hypothesized output we can define the set of (to be) invited guests G to be a subset of the liked people in P (i.e., $G \subseteq L$) with the additional constraint that all pairs of people in G like each other (i.e, $like(p_i, p_j) = true$ for each $p_i, p_j \in G$, or equivalently, $\forall_{p_i, p_j \in G} like(p_i, p_j) = true$).

F: Combined, these formalization choices yield a computational-level model:

> SELECTING INVITEES (VERSION 1)
>
> *Input:* A set of people P, some of whom the host likes ($L \subseteq P$) and some of whom the host dislikes ($D \subseteq P$), with $L \cap D = \emptyset$ and $L \cup D = P$, and a function $like : P \times P \rightarrow \{true, false\}$ specifying for each pair of persons $(p_i, p_j) \in P$ whether or not they like each other.
>
> *Output:* A set of liked guests $G \subseteq L$ that all like each other (i.e., $like(p_i, p_j) = true$ for each $p_i, p_j \in G$).

F: Now we know what the formal symbols mean, we can compress this description:

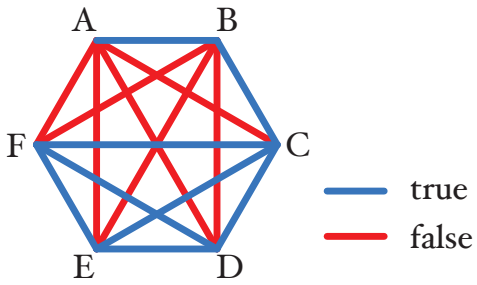
> SELECTING INVITEES (VERSION 1, COMPRESSED)
>
> *Input:* A set P, subsets $L \subseteq P$ and $D \subseteq P$ with $L \cap D = \emptyset$ and $L \cup D = P$, and a function $like : P \times P \rightarrow \{true, false\}$.
>
> *Output:* A subset $G \subseteq L$ such that $\forall_{p_i,p_j \in G} like(p_i, p_j) = true$.

V: That looks great!

F: But does it correctly express what you had in mind?

V: I think so yes.

F: We should rigorously check if it really captures what you have in mind. Let's explore this with an example. Say a host knows six people they all like, i.e., $P = L = \{A, B, C, D, E, F\}$. We can depict their like and dislike relations in a figure. The color of the lines indicate for each pair of persons $p_i, p_j \in P$ the value of $like(p_i, p_j)$:



Who would you predict the host would invite?

V: Of course in that situation the host would invite $\{C, D, E, F\}$.

F: Or they would invite $\{A, B\}$.

V: I would not think so.

F: But according to version 1 of the model, subset $\{A, B\}$ is as likely to be the selected invitees as $\{C, D, E, F\}$, or at least there is no reason why the host would select the one and not the other.

V: But a party with only two guests is not much of a party!

F: So there are more constraints on the subset of guests that you have in mind but did not tell me yet. The host wants to have at least 3 guests?

V: As many as possible, the more the merrier.

F: Ok. Here an adjusted version of the model:

> SELECTING INVITEES (VERSION 2)
>
> *Input:* A set P, subsets $L \subseteq P$ and $D \subseteq P$ with $L \cap D = \emptyset$ and $L \cup D = P$, and a function $like : P \times P \rightarrow \{true, false\}$.
>
> *Output:* A subset $G \subseteq L$ such that $\forall_{p_i,p_j \in G} like(p_i, p_j) = true$ and the size of G is maximized (i.e., there exists no G' such that $\forall_{p_i,p_j \in G'} like(p_i, p_j) = true$ and $|G'| > |G|$).

V: Yes, that is what I mean!

F: This model predicts that a host never invites people who they dislike, nor any pair of people who dislike each other. Is this really realistic?

V: I hadn't thought about that. It's something I could empirically test. I will be right back.

"A few months pass, and..."

V: You are right. In some situations, hosts invite people they do not like. I am not sure why.

F: Whatever the reason, the theory must then be adjusted. Let us remove the assumption that $G \subseteq L$. Then the adjusted theory is as follows:

SELECTING INVITEES (VERSION 3)

Input: A set P, subsets $L \subseteq P$ and $D \subseteq P$ with $L \cap D = \emptyset$ and $L \cup D = P$, and a function $like : P \times P \rightarrow \{true, false\}$.

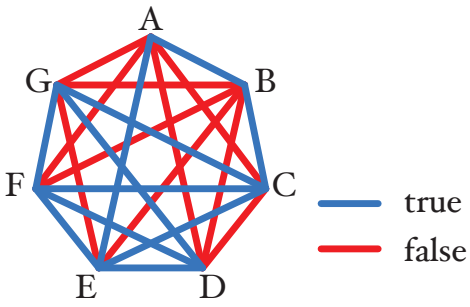
Output: A subset $G \subseteq P$ such that $\forall_{p_i,p_j \in G} like(p_i, p_j) = true$ and $|G|$ is maximized.

F: But this does not seem right, because in this case the theory predicts that the host may end up with a party where everyone likes each other, but the host likes no-one.

V: It seems that hosts invite at most a few people they do not like, and also sometimes some people who do not like each other, as long as sufficiently many people like each other.

F: I am not sure how to formalize that. Can you be more precise?

V: For instance, if the situation is like this, with $L = \{B, C, D, E, G\}$ and $D = \{A, F\}$:



then hosts tend to invite $\{C, D, E, F, G\}$.

F: What do you think is the reason?

V: Possibly hosts tend to *minimize* the number of disliked guests while at the same time *maximizing* the *total* number of guests and *maximizing* the number of guests that like each other?

F: That makes sense intuitively. But formally it is underdefined. We now have three quantities that all need to be optimized (minimized or maximized), but there can be trade-offs such that it's impossible to always optimize them all at the same time.

V: What do you propose?

F: Perhaps we can come up with different model versions that all more or less capture this intuition, but in different well-defined ways, and then compare them empirically?

V: Great idea! Let's do that.

F: OK. Here are three qualitatively different ways to formalize the idea:

SELECTING INVITEES (VERSION 4)

Input: A set P, subsets $L \subseteq P$ and $D \subseteq P$ with $L \cap D = \emptyset$ and $L \cup D = P$, a function $like : P \times P \rightarrow \{true, false\}$, and a threshold value k.

Output: $G \subseteq P$ such that $|G \cap D| \leq k$ and $|X| + |G|$ is maximized (where $X = \{p_i, p_j \in G \mid like(p_i, p_j) = true \wedge i \neq j\}$).

SELECTING INVITEES (VERSION 5)

Input: A set P, subsets $L \subseteq P$ and $D \subseteq P$ with $L \cap D = \emptyset$ and $L \cup D = P$, and a function $like : P \times P \rightarrow \{true, false\}$.

Output: $G \subseteq P$ such that $|G \cap L| + |X| + |G|$ is maximized (where $X = \{p_i, p_j \in G\} \mid like(p_i, p_j) = true \wedge i \neq j\}$).

SELECTING INVITEES (VERSION 6)

Input: A set P, subsets $L \subseteq P$ and $D \subseteq P$ with $L \cap D = \emptyset$ and $L \cup D = P$, a function $like : P \times P \rightarrow \{true, false\}$, and a threshold value k.

Output: $G \subseteq P$ such that $|Y| \leq k$ and $|G \cap L| + |G|$ is maximized (where $Y = \{p_i, p_j \in G\} \mid like(p_i, p_j) = false \wedge i \neq j\}$).

F: There may be other possibilities.[3] I have created simulation code for these versions, available online here https://computationalcognitivescience.github.io/socialpsychtutorial/, so you can explore their empirical implications via computer simulation.

We end Dialogue 1 here. At this juncture you can either go online and check out the code that *Formal* made for *Verbal* and run some simulations of your own; or, alternatively, you can continue with reading Dialogue 2.

Dialogue 2: Formalizing inviting guests, again

In this Dialogue 2, *Formal* again formalizes *Verbal*'s intuitions about how a hosts selects invitees for a party, but after

[3] We leave this as an exercise for the reader to explore further.

initially starting off the same way as in Dialogue 1, *Formal*'s questions quickly lead the dialogue in a different direction. You can think of this dialogue as taking place in a different possible parallel universe from the one in which Dialogue 1 took place.

V: I'd like to explain how a host decides whom to invite to a party.

F: Why would the host not invite everybody?

V: They may like some people but dislike others.

F: Is liking and disliking all or none, or does it come in degrees?

V: I guess it comes in degrees.

F: Then the host invites everyone they like enough?

V: Well, it is also important that the guests like each other enough.

F: OK. So both the host and everyone at the party must like everyone at the party enough?

V: Yes, that sounds right. Can we formalize that idea?

F: Let's see. We can start by defining the initial state (input) of the decision process as consisting of the set of people that the host chooses from, P, and define a function $like_1 : P \to \mathbb{Z}$ that assigns a number for each person $p \in P$ indicating how much the host (dis)likes that person. Negative numbers indicate degrees of disliking and positive numbers indicate degrees of liking.

F: Can a host also sometimes not like nor dislike a person? I mean, can they be neutral about a person?

V: It may be rare, but it seems possible.

F: Ok, then we keep the value '0' as an option for the $like_1$ function.

F: To formalize that different pairs of people in P can also like or dislike each other to various degrees, let us define a function $like_2 : P \times P \to \mathbb{Z}$ that specifies for each pair of persons $(p_i, p_j) \in P$ how much they like each other, again including '0' as an option.

F: To formalize the hypothesized output we define a threshold number k for 'liking enough'. Shall I set $k \geq 0$ or $k > 0$?

V: I suppose, one minimally likes a person if they invite them to ones party? But guests may not be too bothered if there are people around who they feel neutral about.

F: If that is the case, then let's define a separate threshold for the host $k_h > 0$ and a threshold for the guests $k_g \geq 0$.

V: Sounds good.

F: Combined, these formalization choices yield a computational-level model:

> SELECTING INVITEES (VERSION 7)
>
> *Input:* A set P, two functions $like_1 : P \to \mathbb{Z}$ and $like_2 : P \times P \to \mathbb{Z}$, and two threshold values $k_h > 0$ and $k_g \geq 0$.
>
> *Output:* A subset $G \subseteq P$ such that $\forall_{p \in G}\ like_1(p) \geq k_h$ and $\forall_{p_i, p_j \in G}\ like_2(p_i, p_j) \geq k_g$.

We end Dialogue 2 here. At this juncture you could choose to compare how this dialogue ended with where Dialogue 1 ended (e.g., ask yourself 'Which of the models best matches my own intuitions? How would I change or adapt the models?'); or, alternatively, you can continue reading the next dialogue.

Dialogue 3: Formalizing inviting guests, one last time

In this Dialogue 3, *Formal* again formalizes *Verbal*'s intuitions about how a hosts selects invitees for a party, but after initially starting off the same way as in Dialogues 1 and 2, *Verbal*'s different intuitions quickly lead the dialogue in a different direction. Again, you can think of this dialogue as taking place in a different possible parallel universe from the one in which Dialogues 1 and 2 took place.

V: I'd like to explain how a host decides whom to invite to a party.

F: Why would the host not invite everybody?

V: They may have limited space and budget available.

F: Then they invite the number of people that fit the space and the budget?

V: Well, yes. But I think there may be more to it. A host wants a party to be fun. So they probably invite a mix of people that make for a maximally fun party.

F: What type of mix makes for maximal fun?

V: Probably lots of fun people, with great synergies. That boosts the atmosphere. And as few people as possible who do not interact well with others. You know? The right *mix*.

F: Not sure if I get it yet. But perhaps we can start by trying to model those interactions you hint at.

V: Great!

F: Let's see. Let's define a function $fun : P \rightarrow \mathbb{Z}$ that assigns a number for each person $p \in P$ indicating how much *fun* they are individually, and a second function $synergy : P \times P \rightarrow \mathbb{Z}$ that assigns a number for each pair of persons $p_i, p_j \in P$ indicating how much their interaction adds to, or subtracts from the overall *synergy*.

V: So $synergy(p_i, p_j) = 0$ means no synergy between p_i and p_j?

F: Indeed. And if $synergy(p_i, p_j) > 0$ or $synergy(p_i, p_j) < 0$ then their combined presence adds additional fun or decreases overall fun, respectively. Does this make sense?

V: Not sure where you are going with this.

F: Well, given these assumptions, we could formalize the overall party fun as follows:

$$fun(G) = \sum_{p \in G} fun(p) + \sum_{p_i, p_j \in G} synergy(p_i, p_j)$$

V: Makes sense. But this only allows for synergies between pairs of people, while sometimes synergies are really a function of small subgroups of people who interact well with each other.

F: No problem, we can generalize the formalization to include that idea:

$$fun(G) = \sum_{p \in G} fun(p) + \sum_{G' \subseteq G} synergy(G')$$

F: Better in line with your thoughts?

V: Yes!

F: Good. Let us use $2^P = P \times P \times ... \times P$ to denote the *powerset* of P, i.e., all possible subsets of P. Then we can finalize the model as follows:

SELECTING INVITEES (VERSION 8)

Input: A set P, two functions $fun : P \rightarrow \mathbb{Z}$ and $synergy : 2^P \rightarrow \mathbb{Z}$.

Output: A subset $G \subseteq P$ such that $fun(G) = \sum_{p \in G} fun(p) + \sum_{G' \subseteq G} synergy(G')$ is maximized.

V: Wait! You forgot about the most important constraints. The host has limited space and budget.

F: Right! I forgot about those. Can I assume that there is a fixed cost per person, or are some people more expensive to host than others?

V: Let's assume for simplicity sake that cost is the same for everyone.

F: Then here is my proposal:

SELECTING INVITEES (VERSION 9)

Input: A set P and two functions $fun : P \rightarrow \mathbb{Z}$, $synergy : 2^P \rightarrow \mathbb{Z}$. Further, three constants: the cost per person $c \in \mathbb{R}$, a space constraint $S \in \mathbb{N}$ and a budget constraint $B \in \mathbb{R}$.

Output: A subset $G \subseteq P$ such that $|G| \leq S$ and $c|G| \leq B$ and $fun(G) = \sum_{p \in G} fun(p) + \sum_{G' \subseteq G} synergy(G')$ is maximized.

F: If you would like to extend this model to include variable costs, then that is possible. For instance, like this:

SELECTING INVITEES (VERSION 10)

Input: A set P, three functions $fun : P \rightarrow \mathbb{Z}$, $synergy : 2^P \rightarrow \mathbb{Z}$, and $c : P \rightarrow \mathbb{R}$. Further, a space constraint $S \in \mathbb{N}$ and a budget constraint $B \in \mathbb{R}$.

Output: A subset $G \subseteq P$ such that $|G| \leq S$ and $\sum_{p \in G} c(p) \leq B$ and $fun(G) = \sum_{p \in G} fun(p) + \sum_{G' \subseteq G} synergy(G')$ is maximized.

We end Dialogue 3 here. Again, we invite you to compare how this dialogue ended with how Dialogues 1 and 2 ended (e.g., ask yourself, 'Could the constraints of cost also be adopted in the earlier versions 1-7?' 'What other constraints may I want to build in?' 'How do the contraints affect which selections a host makes under different circumstances?').

Dialogue 4: Formalizing group formation

Dialogues 1-3 illustrated three distinct ways in which *Verbal* and *Formal* could settle on formalizations of the social decision-making capacity of a host deciding whom to invite to their party. This social decision-making capacity is a cognitive capacity of an individual, i.e., the host (their brain/mind) is doing the deciding. Not all social psychological capacities need to be like this. Some capacities emerge from the *interaction between* individuals, as we will see in the next and last dialogue.

V: Can we explain how people naturally divide into smaller subgroups at a party?

F: Sure. You think there is systematicity to it?

V: Clearly. People tend to want to be with people who are like them.

F: You mean, to be with people they like?

V: Well, the question is what makes people like each other. Previous research suggests people like being with people who are similar to them; have a lot in common, such as hobbies, preferences, political beliefs, etc.

F: So similarity between people may drive how they form groups?

V: Indeed. Can we model this?

F: It is easy to define a similary measure $sim : P \times P \rightarrow \mathbb{R}$ for every pair of persons in P. But how does it factor into the grouping process? Any intuitions?

V: I think guests at a party tend to self-organize into subgroups with high ingroup similarity.

F: By 'high' do you mean 'maximum average ingroup similarity', or 'satisfactory high levels of ingroup similarity'?

V: Good question. Either could be the case. I honestly don't know.

F: How about we formalize both ideas, and then test empirically which one accounts best for natural grouping behaviour?

V: Sounds good!

F: Here you go.

PARTY SUBGROUPING (VERSION 1)

Input: A set of guests G and a function $sim : G \times G \rightarrow \mathbb{R}$.

Output: A partition of G into non-overlapping subsets $G_1, G_2, ..., G_k$ that maximizes average ingroup similarity:

$$\frac{1}{k} \sum_{i=1,2,...k} sim(G_i)$$

Where ingroup similarity for subset G_i is defined as mean pair-wise similarity:

$$sim(G_i) = \frac{1}{|G_i|} \sum_{g_i, g_j \in G_i} sim(g_i, g_j)$$

PARTY SUBGROUPING (VERSION 2)

Input: A set of guests G, a function $sim : G \times G \rightarrow \mathbb{R}$, and threshold of satisfactory similarity $s \in \mathbb{R}$.

Output: A partition of G into non-overlapping subsets $G_1, G_2, ..., G_k$ where each partition has satisfactory ingroup similarity:

$$\forall_{i=1,2,...k} [sim(G_i) \geq s]$$

Where ingroup similarity for subset G_i is defined as mean pair-wise similarity:

$$sim(G_i) = \frac{1}{|G_i|} \sum_{g_i, g_j \in G_i} sim(g_i, g_j)$$

V: OK, I see how the two models formalize the intuitive idea of 'maximum ingroup similarity' and 'satisfactory ingroup similarity', thanks.

F: Be aware, there may be other possibilities! These two ways of formalizing were just two ways I came up with.

V: Understood. But for now I think they are good working hypotheses.

V: One last question. The models look very similar. How can I test which of them best explains patterns of group formation that I observe in my studies?

F: You can run simulations for these models for different parameter settings and compare the output to

the subgroup formations you've observed in your empirical studies. I have created code for the simulations here https://computationalcognitivescience.github.io/socialpsychtutorial/. Check it out!

Discussion

We close this tutorial by reflecting on some of the lessons learned from the examples and dialogues. We furthermore give pointers on how formalized verbal theories of psychological capacities can be assessed and refined using both empirical and theoretical tests, and how these types of formal models can interface and/or integrate with other modeling frameworks (algorithmic-level or agent-based modeling).

Lessons and reflections

We illustrated with various fictive dialogous how theoretical models can end up different based on the intuitions expressed in verbal theories and formalization choices made. The reader may have wanted to take different turns in the conversations than we explored here. We encourage such exploration. There is no reason why the models created by *Verbal* and *Formal* are to be the preferred ones. The different conversations served to illustrate several didactical points we want to highlight:

- formalization is a dialogical process: It is often through the *interaction* between verbally expressable intuitions and the questions raised in the process of formalization that one comes to more and more well-defined formal characterizations of a (hypothesized) capacity.

- formalization is a revealing process: It makes invisible holes and inconsistencies and hidden assumptions in a verbal theory visible. These hidden problems can be discovered and resolved by explicitly making different model variants.

- formalizations are transparent specifications: formalizations define mathematically precise and well-defined functions that can serve as specifications for computer simulations. Yet, they can be communicated and understood without reliance on any code, as they specify the theory independently of implementational details.[4]

- formalizations make transparent predictions: Unlike predictions 'derived' from verbal theories, formalizations allow predictions to be derived in a transparent and reproducible way. Predictions can be derived analytically (e.g., proof by example) or using simulations one can discover more intricate, and potentially counterintuitive, predictions.[5]

Other than these important lessons, the dialogues also illustrated how formalizing verbal theories is a *non-deterministic* and *open-ended* process: Each dialogue ended with one or more different candidate theoretical models, which are at best working hypotheses. In order to assess and refine or revise these working hypotheses, theoretical modelers need to engage in further steps to which we turn now.

Testing of theoretical models

The dialogues between *Verbal* and *Formal* illustrated a core part of theoretical modeling (Fig. 2, central panel), but the scientific process does not stop here. We highlight two qualitatively different ways in which theoretical models can be assessed and revised.

Most familiar to psychologists is the *empirical cycle* (Fig. 2, right). In this cycle, theoretical models are assessed by deriving predictions that can be tested empirically. Based on any potential mismatches between predicted and observed behaviour, the theoretical modeler may choose to update, refine, revise or throw away a model and start anew. As mentioned above, predictions can be derived from models either analytically or via simulations. In the online Supplementary materials we provided code for the latter option for some of the theoretical models resulting from the dialogues. Testing whether or not such predictions are borne out, of course, requires careful design of empirical studies and making sound statistical inferences. As these research activities build on skills and tools that are already part of the standard psychologist's scientific toolkit, little needs to be said about them here, except the following: when one sets out to empirically test one's theoretical models, keep in mind that scientific inference does not reduce to statistical inference (Szollosi et al., 2020). The interpretation of any findings and their implications for how a model should be revised, if at all, requires good scientific judgment. We refer the reader to Navarro (2019) for an insightful discussion of this important point.

Less familiar to psychologists may be the *theoretical cycle* (Fig. 2, left). In this cycle, theoretical models are assessed on their *a priori* plausibility (van Rooij & Baggio, 2020). For instance, consider the various SELECTING INVITEES models. The number of possible sets of invitees grows exponentially with the number of people a host knows: if the host knows n people, then there are 2^n possible sets of invitees. When $n = 5$, then searching all these possibilities may be a feasible, albeit boring, procedure. However, when $n = 20$, the host would be generating and checking 1,048,576 (more than a million!) possible sets of invitees. Clearly this is unrealistic. Arguably, the host may try to cut some corners to find 'good

[4] See also (Guest & Martin, 2020) for the related notion of 'open theory' and (Cooper & Guest, 2014) for the important distiction between specifications and implementations.

[5] For illustrations see the online Supplementary materials: https://computationalcognitivescience.github.io/socialpsychtutorial/.

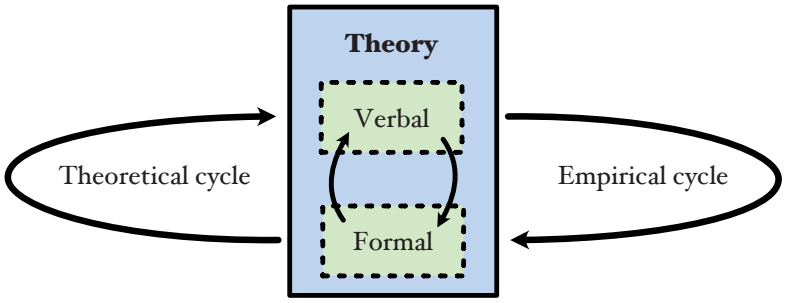


Figure 2. Formal theory is built by translating verbal intuitions into formalizations. This process is cyclical, because formalizations can reveal implications that change the verbal intuitions, which may in turn lead to new and different formalization choices, etc. Once a formal theory has been constructed it can be assessed for its *a priori* plausibility (e.g., tractability or emergeability) and revised as needed (*theoretical cycle*) and it can be assessed for its empirical adequacy and revised when its predictions are disconfirmed (*empirical cycle*). [Figure adapted from van Rooij and Baggio (2020)]

enough' sets of invitees by ignoring large swaths of the space of possibilities via heuristic decision-making. However, it is known that for some (called 'intractable') input-output functions this is proveably impossible. Finding out for which of the SELECTING INVITEES models heuristic decision-making is (im)possible requires its own theoretical tools beyond the scope of this tutorial. We refer the reader to (van Rooij, Blokpoel, Kwisthout, & Wareham, 2019) for a textbook on such tractability analyses.

To illustrate the generality of the *theoretical cycle*, we give another example of an *a priori* test. Consider the PARTY SUBGROUPING models. These models presuppose that groups of individuals can self-organize into subgroups with high ingroup similarity. A self-organised process operates without a central controller; the global pattern emerges solely from the local actions and interactions of inviduals. Can the input-output functions as specified by PARTY SUBGROUPING (version 1 and 2, or otherwise) be realized by a self-organized, distributed social process? In other words, are subgroupings with high-ingroup similary, in principle, emergeable? Again, finding out for which PARTY SUBGROUPING functions this is (im)possible requires theoretical tools. One possible method can be to use agent-based modeling: e.g., each individual's local actions may be modeled by a simulated agent who decides to move closer over time to other simulated agents that are similar to it, and move further away from agents that are dissimilar; simulated runs could then show whether or not such a distributed process stabilizes in subgroups with high-ingroup similarity. While agent-based modeling is not the focus of this tutorial (but see e.g. Macal and North, 2010), we do consider it in the next section as a point of interface with computational-level theoretical models as pursued here.

Relationship to other types of models

As explained in the Introduction, this tutorial gives a primer on formalizing verbal theories at Marr's computational level. This type of modeling does not span all of theoretical modeling relevant for psychology. We briefly discuss how computational-level models can interface with models at different levels of *explanation* and apply at different levels of *organisation* relevant for social psychology (e.g., the level of individuals and the level of social interaction; see Fig. 3).[6]

First of all, computational-level models interface with, and constrain (Blokpoel, 2018), explanations at the other levels of Marr, i.e., the algorithmic and implementational levels. While we did not develop any algorithmic-level (let alone implementational level) models in this tutorial,[7] computational-level models such as illustrated here can guide research into the development of such lower-level models. For instance, for each computational-level theory *Verbal* may have intuitions about the kind of decision *procedure* hosts use to select invitees. This could be a conscious strategy (e.g., a greedy heuristic), but it could also be an unconscious process (e.g., speading activation in a neural network). These ideas can be formalized as algorithms. By simulating both computational- and algorithmic-level models, one can assess when the two kinds of models are compatible and can be integrated. For examples of this type of integration in the domains of emotion, beliefs, impression formation, and social categorisation, we refer the reader to (Thagard, 2000, 2006) and (Klapper, Dotsch, van Rooij, & Wigboldus, 2018).

Second, computational-level modeling can be used to inform and enrich agent-based models (Smith & Conrey, 2007). This can be achieved in at least two ways. The

[6]Levels of organisation—e.g, chemicals, neuronal processes, brain areas, cognitive subprocesses, persons, groups of people, societies—are to be distinguished from levels of explanation. Marr's levels of explanations apply to each level of organisation.

[7]The computer simulations in the online Supplementary materials are not to be confused with algorithmic or implementational level theories of social psychological processes. Those computer implementations are merely *a way* for the scientist to compute the specified computational-level functions.

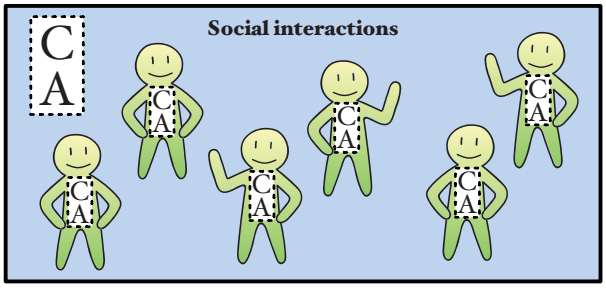


Figure 3. Computational-level models (C) interface with algorithmic level models (A), on two levels of organisation relevant for social psychology: the level of the individual and the level of social interaction. In the first case, C and A describe the cognitive behaviour of a person and their underlying cognitive processes, respectively. In the second case, C and A desribe the global group behaviour and the underlying social distributed processes, respectively. The A at the interaction level is the process usually simulated in agent-based models, and the C at the individual level then describes the cognition of the simulated agents.

formalization of cognitive capacities can be used to model the agents' capacities in agent-based simulations and the formalizations of group-level capacities can be used to model group-level invariants, i.e., stable patterns or properties that group processes dynamically converge on over time (see Section Testing of theoretical models). Traditionally, agent-based simulations have focussed on explaining emergent phenomena that result from the complex and dynamic interaction of a large group of cognitively relatively simple agents. By building more sophisticated computational-level models of cognitive capacities, simulated agents can be given more human-level cognitive capacities and richer behavioural tendencies. The resulting *cognitive*-agent based modeling approach is useful for systematically studying complex social-cognitive phenomena. Examples include pragmatic communication in dyadic interaction (Blokpoel et al., 2019), the cultural co-evolution of social cognition and language (Woensdregt, Cummins, & Smith, 2020), and the formation of 'echo chambers' (Perfors & Navarro, 2019).

Concluding Remarks

This theoretical modeling tutorial is intended for an audience of social psychologists without any prior training in this area.

Did you briefly glance over this tutorial only to be taken aback by the abundance of formal notation and think you are unable to understand this material? Then we encourage you to start at the beginning and work through the tutorial step by step. Learning to make theoretical models is no harder or easier than learning to carefully design experiments or to make sound statistical inferences. We are convinced that social psychologists who are interested in picking up the theoretical modeler's sculpting tools also *can* master them.

Did you work through this tutorial step by step? Then you'll have discovered that with just some basics in mathematics (set theory, functions, and some logic notation) *and* your own intuitions and domain knowledge, you can already develop computational-level models of non-trivial social psychological phenomena. We hope that the theoretical modeler's sculpting tools will become standard additions to many a social psychologist's scientific toolbox.

Acknowledgements

We thank Hans Ijzerman and Kai Epstude for inviting and inspiring this tutorial. We also thank Danaja Rutar, Bob van Tiel, and Vasco Brazão for providing invaluable feedback on an earlier version. Mark Blokpoel was supported by Gravitation Grant 024.001.006 of the Language in Interaction Consortium from Netherlands Organization for Scientific Research.

References

Baker, C. L., Tenenbaum, J. B., & Saxe, R. R. (2007). Goal inference as inverse planning. *Proceedings of the Annual Meeting of the Cognitive Science Society*, *29*.

Blokpoel, M. (2018). Sculpting computational-level models. *Topics in Cognitive Science*, *10*(3), 641–648. doi:10.1111/tops.12282

Blokpoel, M., Dingemanse, M., Woensdregt, M., Kachergis, G., Bögels, S., Toni, I., & van Rooij, I. (2019). *Pragmatic communicators can overcome asymmetry by exploiting ambiguity*. Open Science Framework. doi:10.31219/osf.io/q56xs

Blokpoel, M., Kwisthout, J., van der Weide, T. P., Wareham, T., & van Rooij, I. (2013). A computational-level explanation of the speed of goal inference. *Journal of Mathematical Psychology*, *57*(3), 117–133. doi:10.1016/j.jmp.2013.05.006

Cooper, R. P., & Guest, O. (2014). Implementations are not specifications: Specification, replication and experimentation in computational cognitive modeling. *Cognitive Systems Research*, *27*, 42–49. doi:10.1016/j.cogsys.2013.05.001

Cummins, R. (1985). *The Nature of Psychological Explanation*. MIT Press.

Cummins, R. (2000). How does it work?" versus" what are the laws?": Two conceptions of psychological explanation. In *Explanation and cognition* (pp. 117–144). MIT Press.

De Houwer, J., & Moors, A. (2015). Levels of analysis in social psychology. In *Theory and explanation in social psychology* (pp. 24–40). New York, NY, US: Guilford Press.

Guest, O., & Martin, A. E. (2020). How computational modeling can force theory building in psychological science. PsyArXiv. doi:10.31234/osf.io/rybh9

Klapper, A., Dotsch, R., van Rooij, I., & Wigboldus, D. H. (2018). Social categorization in connectionist models: A conceptual integration. *Social Cognition*, *36*(2), 221–246. doi:10.1521/soco.2018.36.2.221

Knuth, D. E. (1968). *The Art of Computer Programming: Sorting and Searching*. Addison-Wesley Publishing Company.

Macal, C. M., & North, M. J. (2010). Tutorial on agent-based modelling and simulation. *Journal of Simulation*, *4*(3), 151–162. doi:10.1057/jos.2010.3

Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. New York.

Mitchell, J. P. (2006). Mentalizing and Marr: An information processing approach to the study of social cognition. *Brain Research*, *1079*(1), 66–75. doi:10.1016/j.brainres.2005.12.113

Navarro, D. J. (2019). Between the devil and the deep blue sea: Tensions between scientific judgement and statistical model selection. *Computational Brain & Behavior*, *2*(1), 28–34. doi:10.1007/s42113-018-0019-z

Otworowska, M., Blokpoel, M., Sweers, M., Wareham, T., & van Rooij, I. (2018). Demons of ecological rationality. *Cognitive Science*, *42*(3), 1057–1066. doi:10.1111/cogs.12530

Perfors, A., & Navarro, D. J. (2019). Why do echo chambers form? The role of trust, population heterogeneity, and objective truth. In *Proceedings of the 41st Annual Conference of the Cognitive Science Society* (pp. 918–923).

Read, S. J., & Miller, L. C. (1998). *Connectionist Models of Social Reasoning and Social Behavior*. Psychology Press.

Rich, P., Blokpoel, M., de Haan, R., Otworowska, M., Sweers, M., Wareham, T., & van Rooij, I. (2019). Naturalism, tractability and the adaptive toolbox. *Synthese*. doi:10.1007/s11229-019-02431-2

Rich, P., Blokpoel, M., de Haan, R., & van Rooij, I. (2020). How intractability spans the cognitive and evolutionary levels of explanation. *Topics in Cognitive Science*. doi:10.1111/tops.12506

Ross, L. (1977). The intuitive psychologist and his [sic] shortcomings: Distortions in the attribution process. *Advances in Experimental Social Psychology*, *10*, 173–220. doi:10.1016/S0065-2601(08)60357-3

Smith, E. R., & Conrey, F. R. (2007). Agent-based modeling: A new approach for theory-building in social psychology. *Personality and Social Psychology Review*, *11*, 87–104.

Szollosi, A., Kellen, D., Navarro, D. J., Shiffrin, R., van Rooij, I., Van Zandt, T., & Donkin, C. (2020). Is preregistration worthwhile? *Trends in Cognitive Sciences*, *24*(2), 94–95. doi:10.1016/j.tics.2019.11.009

Thagard, P. (2000). *Coherence in Thought and Action*. The MIT Press.

Thagard, P. (2006). *Hot Thought*. The MIT Press.

Thagard, P., & Kunda, Z. (1998). Making sense of people: Coherence mechanisms. In *Connectionist models of social reasoning and social behaviour.* Hillsdale, NJ: Erlbaum. Retrieved from http://cogprints.org/669/1/Making_Sense.html

van Rooij, I. (2008). The Tractable Cognition Thesis. *Cognitive Science: A Multidisciplinary Journal*, *32*(6), 939–984. doi:10.1080/03640210801897856

van Rooij, I., & Baggio, G. (2020). Theory before the test: How to build high-verisimilitude explanatory theories in psychological science. PsyArXiv. doi:10.31234/osf.io/7qbpr

van Rooij, I., Blokpoel, M., Kwisthout, J., & Wareham, T. (2019). *Cognition and Intractability: A Guide to Classical and Parameterized Complexity Analysis*. doi:10.1017/9781107358331

Woensdregt, M., Cummins, C., & Smith, K. (2020). A computational model of the cultural co-evolution of language and mindreading. *Synthese*. doi:10.1007/s11229-020-02798-7