THESIS IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN ARTIFICIAL INTELLIGENCE

Making Meaningful Movements

A computational model of nonverbal communication interpretation

Stefan A. van der Meer

(s0541796)

Department of Artificial Intelligence
Radboud University Nijmegen

September 27, 2010

Supervisors:

Dr. I. van Rooij	(Radboud University Nijmegen)
Dr. F.A. Grootjen	(Radboud University Nijmegen)

External examiner:

Dr. H.T. Wareham	(Memorial University of Newfoundland)

Acknowledgements

I would like to thank my supervisors, Iris van Rooij and Franc Grootjen, for their feedback, suggestions and support over the past year, all of which contributed immensely to this thesis. I would also like to thank Todd Wareham for lending his expertise on structure-mapping theory, offering valuable insights and suggestions. A substantial amount of thanks goes to my friends and to my family, to be allocated pro rata at a future occasion[1]. Inevitably, the largest slice will end up with my parents, whose support has been incredibly important throughout my academic career.

[1] Exact date to be determined. Contact my future secretary for scheduling.

Abstract

The Tacit Communication Game (TCG) is a task used by cognitive neuroscientists to study the basic principles of human communication (de Ruiter et al., 2007, 2010). In this task, a Sender player must communicate goals nonverbally to a Receiver player by moving a token on a 3-by-3 grid. Both players are assigned a token in each trial, which can vary in shape and can differ between the players. The Sender player must design and perform a sequence of movements that signals the goal location and orientation of the Receiver's token, allowing the Receiver to place it correctly.

An architecture for a computational model of the task was recently developed by van Rooij et al. (2009) at the Donders Institute for Brain, Cognition and Behavior. A key hypothesis of the architecture is that players assign meaning to movements through analogy and re-representation. This thesis describes the first implementation of a core part of said architecture in the form of a Receiver model. Building on established concepts from analogy research such as structure-mapping theory (Gentner, 1983), the implemented Receiver model is capable of correctly interpreting movement sequences resulting from common strategies used by human Sender players.

The capabilities of the model show that analogy and re-representation can be sufficient for the successful interpretation of signals used by human players. Both the strengths and shortcomings of the implementation are analyzed in context of how they can inform future work on a TCG-playing model. A number of possible improvements are discussed, as are several key problems that future research will have to solve in order to develop a fully sufficient TCG-playing model.

Contents

Chapter 1

Introduction

In this chapter, the problem of communicating intentions through actions is introduced, followed by a description of an experimental task designed to study that area: the Tacit Communication Game (TCG). Subsequently, the aims of this thesis are described, which relate to a computational cognitive model of the Receiver role in the TCG.

1.1 Communicating intentions

At first glance, human communication appears to be fairly straightforward. Considered in light of the well-known *mathematical theory of communication* proposed by Shannon (1948), it appears to be a problem of reliably transmitting signals. Information *encoded* by a Sender is transmitted (using sound waves, for example) through a noisy channel and a Receiver *decodes* it (see Figure 1.1). However, the complexity of human communication lies not in data transmission, but in *intention recognition* and *recipient design* (de Ruiter, Noordzij, Newman-Norlund, Newman-Norlund, Hagoort, Levinson & Toni, 2010). These problems relate to the communication of *intentions* from the viewpoint of the Receiver and the Sender respectively.

Intention recognition refers to the problem the Receiver must solve in order to extract the communicative intention that motivated the Sender of the signal to transmit it. In a non-linguistic context, this requires the more basic ability to discern certain behaviors as being intentional or goal-driven (such as a human picking up a lottery ball to show its number) rather than just effects of non-intentional mechanisms (a lottery machine selecting a ball). In addition, the Receiver must be able to perceive whether an action has a communicative goal (wave to hail a cab) or an instrumental goal (wave to chase a fly away). These abilities allow the disambiguation of behavior of others by casting it in light of hypotheses concerning their goals and intentions. An agent with the ability to attribute mental states to others, and form predictions based on those states, is typically referred to as possessing a 'theory of mind' (TOM) (Premack & Woodruff, 1987).

In natural settings, multiple communicative and instrumental actions can occur simultaneously without clear boundaries. Therefore, a more fundamental ability is required before a TOM can be used: the ability to parse overlapping sequences of movement into goal-directed actions. Once an action parsed from behavior can be recognized as (potentially) having a communicative intention, intention recognition involves the interpretation of said action in order to extract the Sender's intention.

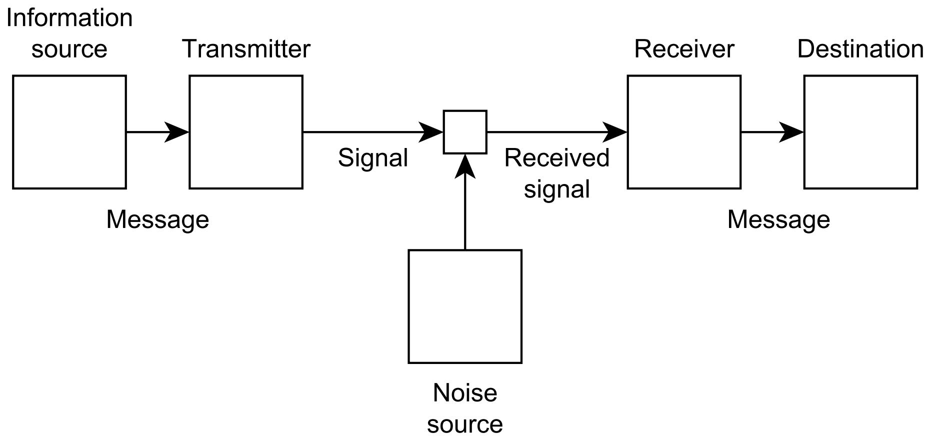


Figure 1.1: Diagram of the 'mathematical theory of communication', from (Shannon, 1948).

Recipient design is the counterpart to intention recognition. Senders must produce signals such that the Receiver's chance of success in the complex task of intention recognition is maximized. For example, Senders do not use a signal they do not expect the Receiver to understand. One way in which actions can be designed to be transparent to the intended recipient is through a simulation of their recognition process (Levinson, 2006).

Figure 1.2 illustrates the basic idea of where intention recognition and recipient design would be positioned in Shannon's diagram. Of course, it is still far from a complete model of human communication, lacking for example an account of the corrective feedback a Receiver will typically provide in many real-world scenarios. However, excluding such factors in an experiment would allow one to examine intention recognition and recipient design more closely. An experimental task that attempts to capture the processes of intention recognition and recipient design is the Tacit Communication Game.

1.2 The Tacit Communication Game

The Tacit Communication Game (TCG) is a non-verbal communication task developed by de Ruiter, Noordzij, Newman-Norlund, Hagoort & Toni (2007) in order to study human communication. The task aims to facilitate experimentally controlled study of communication by avoiding the additional complexities of linguistic communication (being non-verbal) and pre-existing communicative conventions (being sufficiently different from day-to-day communication). At the same time, it captures much of the core non-linguistic complexities of real-world communication, including intention recognition and recipient design. The game has been used in both behavioral (de Ruiter et al., 2010) and fMRI research (Noordzij, Newman-Norlund, de Ruiter, Hagoort, Levinson & Toni, 2009, de Ruiter et al., 2007).

The TCG is a two player game played on a 3 x 3 grid. One player takes on the role of Sender, while the other is the Receiver (here referred to using the female and male

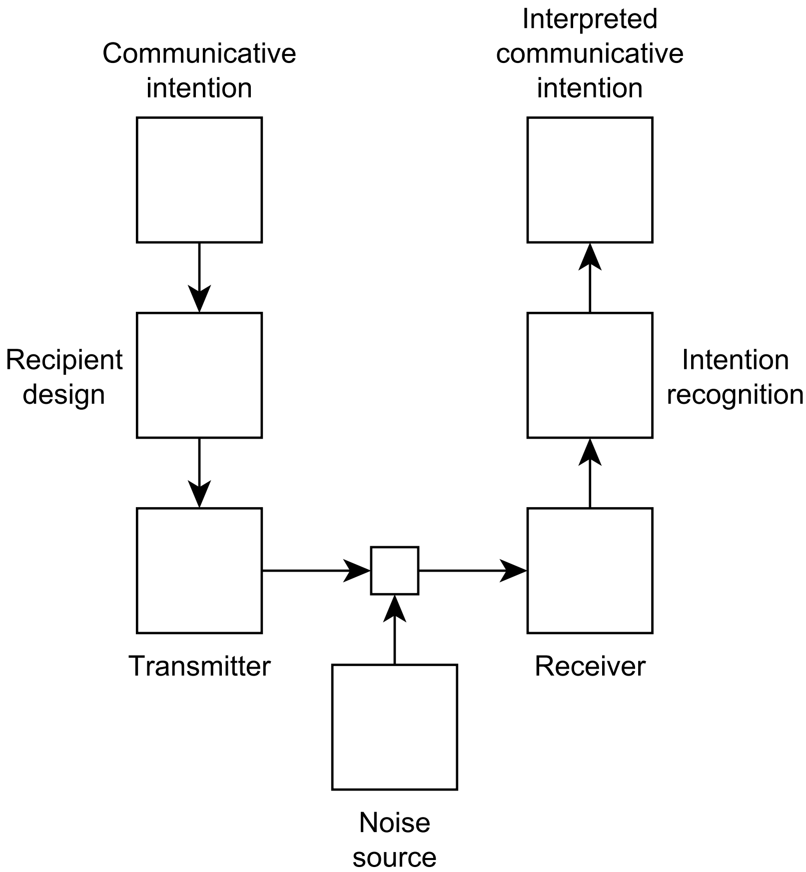


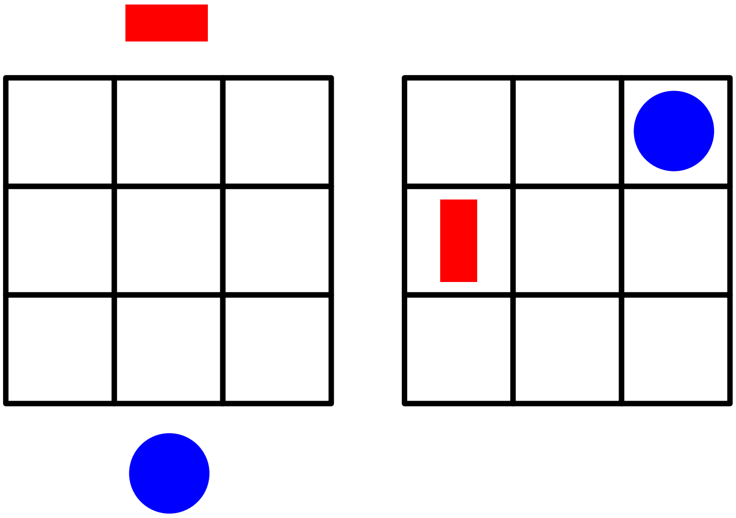
Figure 1.2: A simple extension of Figure 1.1 to illustrate how intention recognition could be positioned in Shannon's diagram.

pronoun, respectively). Each player controls one token, which can have the shape of a rectangle, a circle, or an (equilateral) triangle. Tokens can be translated and rotated (stepwise) by the player during their turn.[1]

A trial is played successfully when both players have moved their tokens into the same position and orientation as their respective target tokens. In a typical trial, only the Sender is shown the positioning of the target tokens (the goal configuration). To complete the trial correctly, the Sender must communicate to the Receiver how his token should be positioned on the board, while also positioning her own token correctly.

As per the rules, the Sender can only communicate by moving her token around the board. Movement is restricted to horizontal and vertical steps to adjacent board positions. Tokens can be rotated on the spot, though the circle token has no visible orientation. Once the Sender has completed her movements, the Receiver moves his token to the position and orientation he has understood to be his goal state.

[1]Demonstration videos of the TCG are available at: http://youtu.be/klx2M7v0hyc and http://youtu.be/OarfUc4nans

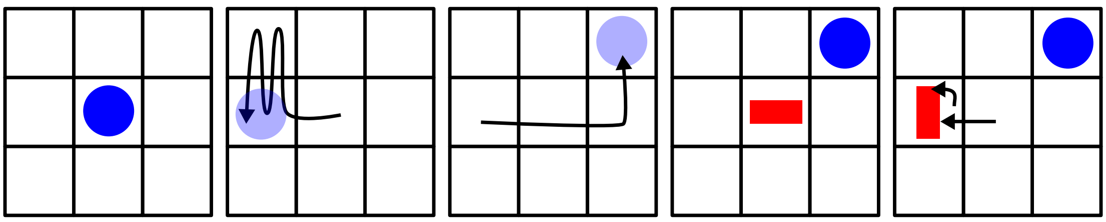


(a) A TCG board. **(b)** A goal state.

Figure 1.3: Examples of the initial state of a TCG board from the Sender's perspective. The Receiver's token is the rectangle positioned above the grid, while the Sender's circle is below it. Each player sees their own token below the grid, i.e., the Receiver's view of the game horizontally flipped compared to the Sender's view shown here (in practice, subjects each see the game on their own computer monitor). The second example shows a possible goal state as shown to the Sender. It indicates how the two tokens should be positioned in order to correctly complete the trial.

The constraints of the game create several problems the players must solve. The Sender must send information using the same means she must use to reach her own token's goal state. Hence, she must perform recipient design in order to make sure that (a) her communicative movements can be discerned from her instrumental movements (and vice versa), and (b) her movements communicate the goal state in such a way that the Receiver can successfully interpret them.

The Receiver faces the counterparts of these issues. He must (a) parse which actions are communicative and should be analyzed, and which actions can be considered purely instrumental, and (b) understand the signaled goal state embedded in the communicative movements.



(a) **(b)** **(c)** **(d)** **(e)**

Figure 1.4: A trial of the TCG for the goal configuration shown in Figure 1.3.

Figure 1.4 illustrates how a trial with the goal configuration shown in Figure 1.3 might proceed:

(a) Start of the Sender's turn, after she has reviewed the goal configuration and planned her movements. When she signals her readiness, her token is placed in the center of the grid and she can begin executing her movements.

(b) The Sender moves to the left, and proceeds to step repeatedly between two positions to signal the orientation of the Receiver's rectangle in the goal configuration.

(c) After a number of repetitions, the Sender moves to her own goal position.

(d) Start of the Receiver's turn. His token is placed in the center and he receives control of it.

(e) The Receiver has understood the Sender's signal, and moves to his goal position and rotates his token to align with the Sender's repeated steps.

Trials in which Sender and Receiver have a differently shaped token, such as in the above example, have a lower success rate than trials in which their tokens are the same shape (de Ruiter et al., 2010). This is caused by the difficulty of communicating the orientation of the Receiver's token, when the Sender's token cannot be put in that orientation. For example, a circle has no orientation, and can therefore only be used to signal an orientation by performing a sequence of movements. Such a signal is inherently more complex than directly showing an orientation by putting one's token in that orientation. The issue is not limited to the circle, as the rectangle can also show fewer orientations than the triangle.

These cases are classified by de Ruiter et al. (2010) as 'hard' trials: the trials in which indicating an orientation is problematic because the Sender's token has fewer visible orientations than the Receiver's token. Besides observing that such trials had a significantly lower success rate than easier types (though still far above chance levels), they analyzed how subjects used their constrained capabilities to communicate goals.

For 'easy' trials, the Sender would most commonly move to the Receiver's target location, rotate their token to match the Receiver's target orientation, pause, and then move to their own target position. The pause serves to indicate which position and orientation is the goal state. This strategy would result in success in 95 percent of the easy trials.

For 'hard' trials, de Ruiter et al. describe a more varied set of strategies. The following three strategies were used by the Sender in nearly 80% of the trials in total[2]:

A. Move to the Receiver target position and pause. Then move one square in the direction the Receiver's triangle is "pointing in", or if a rectangle, oriented along. Move back to the Receiver target position, pause again, and move to the Sender goal position. This strategy is also shown in Figure 1.4. It was used in 40% of the trials, succeeding in 75% of those attempts.

B. The strategy also used in the easy trials left unmodified, simply not communicating the Receiver's orientation. All hard trials required the Receiver's token to be rotated. As a result, this strategy never resulted in a successful trial. Surprisingly, it was still used by subjects in over a quarter of the hard trials.

[2]The remaining 20% of the trials featured a variety of largely ineffective strategies.

C. As B, but attempting to orient the Sender's rectangle such that it matches the Receiver's triangle (hence only applying in trials with that configuration of tokens). Obviously this only transmits half the required information about the orientation of a triangle, and in the 10% of the trials it was used in, only 25% were successful.

Using such strategies, a Sender can communicate a goal state to the Receiver with varying success, despite the limitations of the shape of her token. A question remains, however: how does the Receiver assign meaning to a communicative action performed by the Sender? What cognitive processes allow him to solve the problem of intention recognition when observing and interpreting the movements that make up these strategies? The following section will discuss how this thesis aims to explore these and related questions.

1.3 Aims of this thesis

Van Rooij, Toni & Haselager (2009) have proposed a general architecture of a computational cognitive model of the Sender and Receiver players in the TCG. A core hypothesis of the architecture is that communicative movements are assigned a meaning through the use of analogy and re-representation.

A simple use of analogy can be seen in the common strategy on easy trials. Intuitively, the state of the Sender's token during her pause is analogous to the goal state of the Receiver's token, despite the Sender not treating it as a goal state herself. A less direct mapping occurs in hard trials using the strategy A outlined in the previous section. The Sender uses the direction of movement as analogous to the orientation of the Receiver's token. For this mapping to be possible, the Receiver has to re-represent a series of observed movements as repeated, directed movements that can match the orientation of his token.

By creating an implementation of this model and examining its performance and behavior, insight can be gained into the set of cognitive abilities (such as those relating to analogy and re-representation) that are sufficient and/or necessary for explaining human communication. A sufficient model of communication could be used to test if lesions in specific cognitive abilities give results that approximate human communicative deficits. In addition, a fully sufficient model would allow an artificial agent to play the TCG with a human, effectively engaging in open-ended nonverbal communication.

This thesis describes a first step towards such a fully sufficient model. It describes the implementation and performance of a computational model of the intention recognition process of a Receiver in the TCG task. It aims to test whether the abilities of analogy and re-representation could be sufficient for the interpretation of strategies used by human Senders, as well as informing future development of the model. It does *not* aim to provide a complete implementation of the proposed model architecture, lacking the Sender system and learning abilities included therein (see Figure C.1, p. 66), but focuses on modeling core Receiver abilities such as analogy and re-representation.

These aims can be phrased as the following core questions:

- Are the model's abilities of analogy and re-representation sufficient for the interpretation of movement sequences that apply communicative strategies as used by human Sender players?

- What strategies or signals does the implemented model interpret successfully, and where does it fail? Why is this the case, and what does it mean for future modeling efforts?

- What problems remain that must be solved in order to develop a full, computationally sufficient model of human TCG-playing behavior?

Answers to these questions would substantially advance our knowledge in the area of intention recognition in human communication. They would also supply future modeling efforts with critical information on the strengths and weaknesses of both the model architecture and the implementation described in this thesis, providing a significant stepping stone towards a fully sufficient model.

The remainder of this thesis is structured as follows. First, the implementation of the model is discussed in detail in Chapters 2 and 3. Chapter 2 covers the Parsing system, which processes movements into goal-oriented actions, and hypothesizes which actions are communicative. Chapter 3 describes the Meaning-mapping system, in which such communicative actions are mapped to goals using analogy and re-representation. In Chapter 4, a qualitative analysis of the model's performance on a variety of signals is given, showing that it is capable of successfully interpreting most common strategies. Lastly, the resulting conclusions will be discussed in Chapter 5, as well as opportunities for future research.

Chapter 2

Parsing

This chapter describes the representations and algorithms used in the implementation of the *Parsing system* in the Receiver model. The Parsing system is responsible for processing a sequence of 'raw' movements into higher-level actions. For each action the system must then hypothesize whether it has only an instrumental goal (e.g., simply reaching a certain position), or a non-instrumental communicative goal (e.g., signaling that a position is the Receiver's goal position). Actions that are hypothesized to be communicative will be further analyzed by the Meaning-mapping system, discussed in Chapter 3.

Figure 2.1 shows the architecture on which the model is based (van Rooij et al., 2009). In this architecture, the mapping of actions to possible instrumental goals is informed by the Receiver's theory of mind, as well as a history of previous action-to-goal mappings.

The history of mappings is not implemented in the Receiver model described here. The Receiver's theory of mind about the Sender finds its way into the implementation through certain assumptions about the Sender. The model hypothesizes that the Receiver makes these assumptions when parsing a Sender's movements.

These aspects will be discussed further in this chapter, after describing the relevant representations used in the Parsing system.

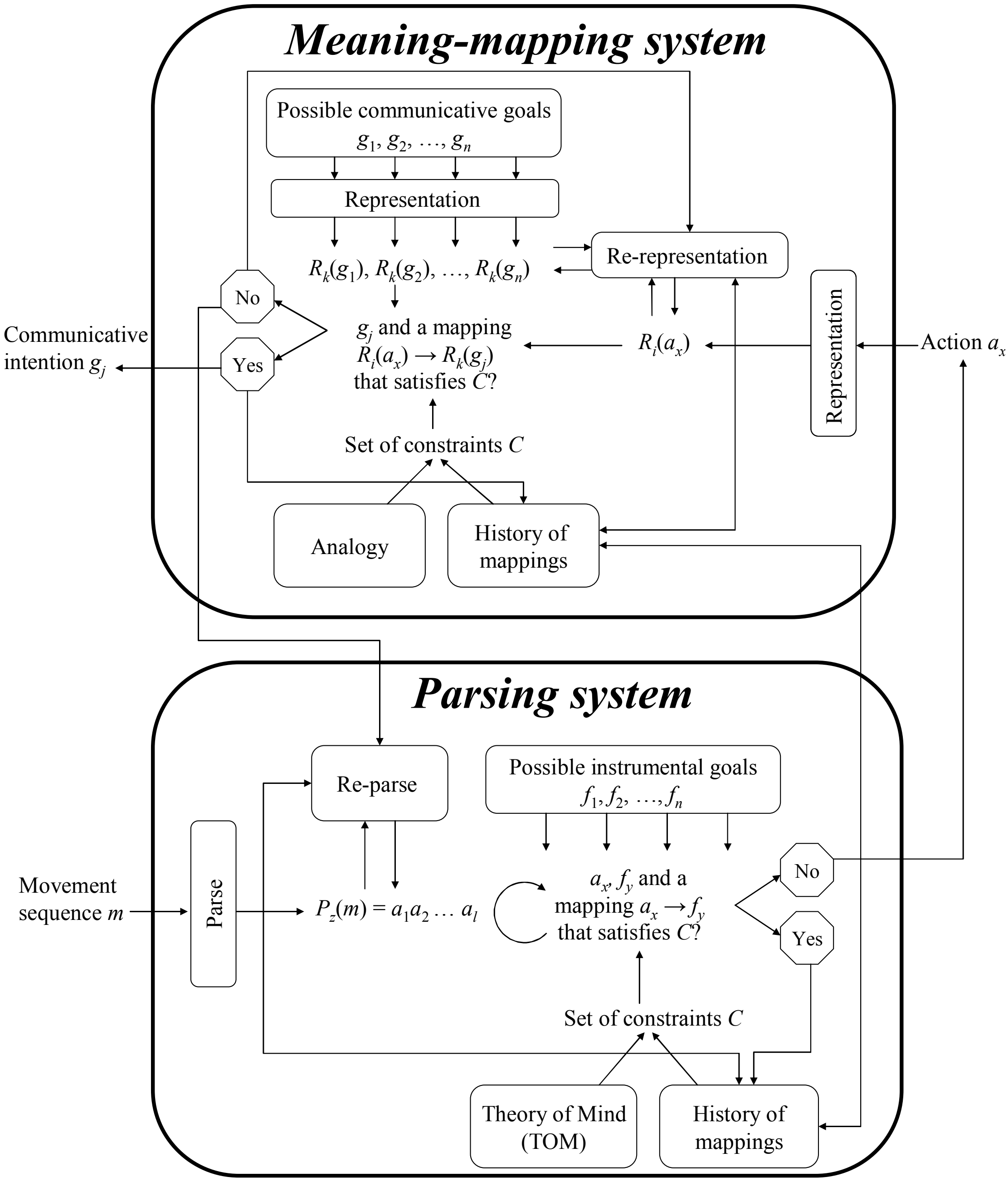


Figure 2.1: Detailed view of the Receiver architecture proposed by van Rooij et al. (2009), on which the model described in this thesis is based. A movement sequence received from a Sender is parsed into actions, which are then mapped to instrumental goals. If an action cannot be mapped to such a goal, it is sent to the Meaning-mapping system. There, attempts are made to map the action to a communicative goal using different (re-)representations. If such a mapping is found, the goal is returned as the communicative intention of the movement. If no mapping is found, the movement is re-parsed into actions and the process repeats. The process of testing different action representations and the re-parsing of the movement can occur in parallel.

2.1 Representation of movements and actions

This section discusses the representations of movements and actions as used by the Parsing system in the model.

2.1.1 Movements

Movements are modeled as transitions from a TCG game state to the next. Specifically, a 'movement' is considered to be a change in the position on the board of the player's token, a change in orientation of the token, or an explicit lack of change (in case of a pause). If a change in position occurs, a change in orientation cannot occur in the same movement, and vice versa. The position of a token can only be changed by a single step on the 3 by 3 board along the vertical or horizontal axis. The orientation of a token can only be changed by a single rotation of 45 degrees clockwise or counterclockwise[1]. To achieve larger turns or reach a further position, multiple movements must be performed.

Time is implicitly represented in a sequence of movements: every movement occurs at a discrete time step. As a simplifying assumption, varying delays between movements are not represented. The only relevant delay is the explicit pause, which is simply represented as a movement resulting in no change in position or orientation.

One could describe these concepts more formally as follows: a movement m is a tuple $\langle s_i, s_{i+1} \rangle$, where s_i and s_{i+1} are the game states before and after the movement has been performed, respectively. These states are elements of a consecutive sequence of game states S forming the entirety of a player's turn in the TCG game. Similarly, the sequence of movements M includes all consecutive movements performed by a player in their turn.

A game state s can be fully described by a triple $\langle x, y, \rho \rangle$, where x and y form a coordinate on the 3 by 3 TCG board from $(0, 0)$ (top left) to $(2, 2)$ (bottom right). This is the location on the board where the current player's token was placed. The positioning of the token within that grid cell is described in ρ. This cannot be described unambiguously by, for example, an angle due to the nature of the shapes in the TCG (such as the equilateral triangle). This issue is revisited in the discussion of the shape representations used in later stages of processing. For now it suffices to assume the presence of a sufficiently descriptive representation in state s.

2.1.2 Actions

Actions are represented simply as sequences of movements, in the order that they occurred. Conceptually, an action is a series of movements achieving a certain instrumental or communicative goal.

Formally, an action A is a (sub)sequence of one or more consecutive movements from the entire movement sequence M, such that $A \subseteq M$.

[1] The amount of rotation per step differs between TCG experiments. For example, de Ruiter et al. (2010) used 90 degree increments.

2.2 Parsing movements to actions

The goal of the parsing process is to discern those parts of the movement sequence that are potentially signaling information from those that are clearly instrumental. The meaning-mapping process can then extract the communicated information without wasting significant computational effort on interpreting movements that are without communicative intentions.

2.2.1 Discerning the communicative from the instrumental

As the Sender in the TCG must reach a goal position herself, the Receiver cannot interpret all movements as being communicative, and must in fact assume the opposite. Unless a series of movements (i.e., an action) is somehow observed as clearly *not* instrumental, the Receiver cannot discern it from an instrumental action.

Luckily, the Receiver has a way to discern non-instrumental actions from the rest. A (well-intentioned) Sender will always perform her instrumental actions as efficiently as possible. When travelling between two locations on the board, she will use an optimal route, rather than taking a detour. She will do this to avoid creating noise for the Receiver: instrumental actions that appear to be informative.

At the same time, the Receiver knows that the Sender is avoiding noise. He could therefore assume that every action that is fully efficient in reaching its end position from its start position must be instrumental. On the other hand, *inefficient* actions are likely to contain a communicative signal. There is no reason to pause while moving from one point to another, hence that pause is likely to signal something and should be interpreted in more detail. The importance of efficiency is a core assumption of the parsing algorithm, as we will see in the remainder of this chapter

2.2.2 Determining efficiency

As described, it is assumed that an instrumental action will be efficient: it will achieve its goal state from its starting state in the smallest possible number of movements. One way to consider all possible states is a graph with a node for every possible combination of a board position and token orientation, and an edge for every valid movement between nodes (including self-loops for pauses). An instrumental action will follow the shortest path on the graph from its starting node to its end node.

Though it is useful to reason about states using such a graph, it is not a requirement for determining the efficiency of an action. An efficient action will only consist of movements that change the position and orientation of a token. The optimal length of an action can be found by taking the minimum number of steps required to reach the end position from the starting position, and add this to the minimum required number of rotations to reach the end state's orientation from the starting orientation.

Both can be found using simple operations. As the board is a grid and no diagonal movement is allowed, the minimum number of moves is found by taking the Manhattan distance between the start and end positions. The minimum number of rotations can be found by dividing the difference between the start and end rotations by the amount a player can rotate his token in a single step.

2.2.3 Generating parsings

In order to determine whether certain actions are communicative or instrumental, the movement sequence must first be divided into actions. One such a division is referred to as *a parsing* of the movement sequence. For a parsing to be valid, all movements must be included in at least one action, as otherwise they are left unexplained. Actions may be of any length, as a single pause can be a communicative action, and actions may overlap, as part of one communicative action may be required to successfully interpret another. As a result, it is computationally intractable to exhaustively generate all possible parsings for all but the shortest movement sequences. The number of possible unique parsings grows super-exponentially to the length of the movement sequence.

The constraints under which the Sender operates can help us reduce the number of parsings. In order to perform a communicative action, a Sender must first navigate to a starting position from which that action can be performed. Similarly, after completing it she must navigate to her own goal state. This suggests that many movement sequences will be structured as a instrumental action, followed by a communicative action, in turn followed by another instrumental action.

Additionally, the Sender will avoid noise. Though in principle instrumental actions can overlap with communicative actions, the Sender will be aware that this makes her signal more difficult to parse and interpret, and will therefore aim to avoid it.

Working from these assumptions about the behavior of the Sender, we can arrive at a restricted set of parsings sufficient to interpret most, or even all, movement sequences a Sender is likely to perform. Given the set U, containing all unique subsequences of M, we take the Cartesian power[2] U^N. Each element of the resulting set is a tuple of N subsequences that are hypothesized communicative actions of a (currently incomplete) parsing.

Each parsing is completed by taking each subsequence that is not part of the N hypothetical communicative actions in that parsing, and hypothesizing it to be an instrumental action. We remove from each parsing any communicative actions that are wholly contained within another communicative action in the same parsing, as such fine-grained subdivisions in the signal should be handled by the meaning-mapping process. The parsing now forms a full 'explanation' of the movement sequence: every movement is hypothesized to be instrumental or communicative. Figure 2.2 shows examples of completed parsings.

Finally, we remove from the set of parsings every parsing where every hypothetical communicative action is in fact fully efficient, and can therefore not be communicative. In the next section, the selection process deciding which parsings to meaning-map is described.

The parameter N for the number of hypothetical communicative subsequences is the primary source of complexity in this approach and should therefore be small. In practice, $N = 1$ is already sufficient to successfully interpret most signals.

[2] The Cartesian power is defined as $U^N = \underbrace{U \times \cdots \times U}_{N}$

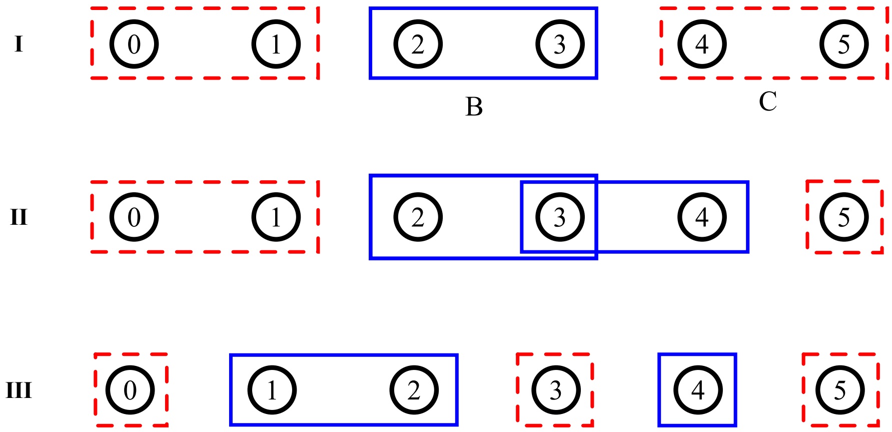


Figure 2.2: Three examples of different possible parsings for a single movement sequence. The numbered circles represent movements. The solid blue rectangles indicate hypothesized communicative actions, while the dashed red rectangles indicate instrumental ones. Parsing **I** shows a basic example where $N = 1$, **II** shows a parsing with overlapping communicative actions with $N = 2$, and **III** shows a parsing where the two communicative actions (via $N = 2$) are not contiguous, resulting in an extra instrumental action in between.

2.3 From Parsing to Meaning-mapping

As the meaning-mapping process can be time-consuming and a source of computational complexity, we assume parsed actions are sent to the Meaning-mapping system in order of expected utility, in order to minimize the number of actions that are processed by the Meaning-mapping system before a result is found.

This requires the definition of an ordering on the set of parsings that is generated from a movement sequence. We can intuitively say that an action is more likely to result in a successful match if it includes only that part of the movement sequence that is communicative. As our means of determining that an action is (hypothesized to be) communicative uses inefficiency, we are interested in the action that is as short as possible while still containing as many of the inefficient movements as possible.

First, the set of parsings is ordered by the number of its instrumental actions that contain inefficiency. It is possible that these inefficiencies are mistakes the Sender made, which are indeed part of an instrumental action and should not be interpreted as communication. However, it is more likely that this is in fact a communicative signal that should be interpreted as such, and the parsing is wrong. Hence, we order the parsings such that those with little or no inefficiency in instrumental actions are considered first.

Those parsings that are equal in the number of inefficient instrumental actions are then sorted on a second criterion, which is the total length of their communicative actions. Parsings where less of the movement sequence is considered communicative, while not missing out on any possible signals (inefficiencies), will be easier to interpret. After all, these parsings will contain less noise in the form of wrongly parsed instrumental movements. Ideally, a parsing should be *minimal*, with its communicative actions only containing movements that signaling information.

The parsings in the sorted collection of parsings are sent to Meaning-mapping in-order, until a successful match is found. If there are multiple communicative actions in a selected parsing, the smallest is interpreted first.

2.4 Summary

Following the proposed architecture (see Figure 2.1, page 10), the Receiver model consists of a Parsing system and a Meaning-mapping system. The Parsing system takes a sequence of discrete movements of equal duration as input, and generates parsings of that sequence into actions. Every such parsing consists of actions that are hypothesized to have either an instrumental goal (i.e., simply reaching their final position from their starting position) or a communicative goal (signaling information about a goal state).

In order to discern actions likely to have a communicative goal from those more likely to have an instrumental goal, the algorithm uses as its core assumption that a Sender will perform instrumental actions as efficiently as possible. Any actions that do not further an instrumental goal of the Sender are hypothesized to have a communicative goal.

It is intractable for a Receiver to consider every possible division of the input sequence into (possibly overlapping) subsequences. Therefore, a structure is assumed in which there are only a limited number of communicative parts, typically one. The parts of the movement sequence not in a (hypothesized to be) communicative subsequence is hypothesized to be instrumental. All possible parsings are generated and sorted, based on how much of the inefficiencies in the movement sequence are captured in the communicative action(s), and the total length of the communicative action(s). As a result, the parsing that is first sent to the Meaning-mapping system is the parsing that captures the most inefficiencies in the communicative part, while keeping that part as short as possible.

Chapter 3

Meaning-mapping

In the Receiver architecture proposed by van Rooij et al. (2009), actions found to be potentially communicative by the Parsing system are sent on to the *Meaning-mapping system*. Much like the Parsing system, the Meaning-mapping system attempts to map actions to goals, informed by previous successful mappings.

However, here the mapping is subject to analogical constraints, as the model hypothesizes that the Sender will use analogy to communicate. Hence, an action that is indeed communicative must be *analogous* to a communicative goal (that is, one of the Receiver's possible goal states). If a mapping cannot be found, both the action and the possible goals can be re-represented. If further re-representation is not considered useful, the Meaning-mapping system can trigger the Parsing system to re-parse the movement sequence.

Often, multiple distinct pieces of information are required, such as the position of the token on the TCG board on one hand, and its orientation on the other. These bits of information may be signaled in different ways, requiring different representations of the action and goal to match, perhaps even a different parsing.

As a result, the model performs the search (including parsing *and* meaning-mapping) for these two aspects independently. In the Parsing system, this is not visible in the process itself, as parsing is performed identically for both. In the meaning-mapping phase, it does have an effect. When attempting to match the action to possible goals, only *relevant* goals are considered. For the orientation search, these are the goals representing all possible goal orientations of the Receiver's token. For the positional search, a set of goals representing all nine possible board positions is used.

The assumption that the search for these distinct elements of information can be split into mutually independent (possibly parallel) searches greatly simplifies the search process. Without it, the search would have to cover irrelevant goals, and might perform re-representation of the action that benefits the search for one element of information, while harming another.

In the following sections, the Meaning-mapping system of the model is described. First, analogy construction is discussed, followed by a description of the algorithm used by the model for that task. Then, the base representations of actions and goals as used by the Meaning-mapping system are described. The remaining two sections discuss the re-representation algorithm and the set of re-representation operators currently implemented in the model.

3.1 Analogy construction

The ability to construct and recognize analogies is considered an important contributor to human intelligence (Gentner & Colhoun, 2008; Gentner, 2003). A good explanatory analogy highlights common relational structure between a *base* and a *target* analog, and allows one to infer new knowledge about an unfamiliar target using existing knowledge from the base.

For example, the analogy “An atom is like our solar system” may result in matches on relational structure: an electron *orbits* the nucleus, like a planet *orbits* the sun, and the nucleus *attracts* the electron like the sun *attracts* the planet.

Assuming one has additional knowledge about the solar system, such as the fact that sun *attracting* the planet *causes* it to *orbit* the sun, one can infer that the same ‘*cause*’ relation exists between the nucleus *attracting* the electron and it *orbiting* around that nucleus. At the same time, properties such as the sun being yellow should not be mapped to the nucleus.

Gentner & Colhoun (2008) distinguishes between several analogical processes:

- *Retrieval*: given a current concept (in working memory), an analogous example may be retrieved from long-term memory.
- *Mapping*: given two cases, mapping aligns their representational structures to find commonalities between them and project inferences from one to the other.
- *Evaluation*: given an analog, evaluate its quality and (primarily) that of its inferences.
- *Abstraction*: abstract the commonalities in structure between the analogs.
- *Re-representation*: adapt the representations of one or both analogs to improve a match.

Of these processes, mapping and re-representation are most relevant for this thesis. Re-representation is discussed in Section 3.4 (page 27), while the remainder of this section will cover mapping. Mapping is the core of analogy construction, and has received most research attention. Structure-mapping theory (Gentner, 1983) is the most influential work in this area, and has been applied in a wide range of contexts (French, 2002).

Gentner’s structure-mapping theory (SMT) of analogy defines an analogy as specifying a mapping between two conceptual structures. In SMT, these conceptual structures are represented as predicate-structures (or *concept graphs*), which consist of a set of *objects* and a set of *predicates*. The objects correspond to entities (`Sun`, `Planet`). Predicates specify *relations* among objects (Attracts(`Sun`, `Planet`)) and among predicates (Cause(Attracts(`Sun`, `Planet`), Orbits(`Planet`, `Sun`))), or express *attributes* of objects (Mass(`Planet`)).

An analogy “T is (like) a B” defines a mapping from B, the *base*, to T, the *target*. The base serves as the source of knowledge in the analogy, and the target is the domain to which knowledge is transferred. The mapping must satisfy three constraints (Gentner & Markman, 1997): structural consistency, relational focus, and systematicity.

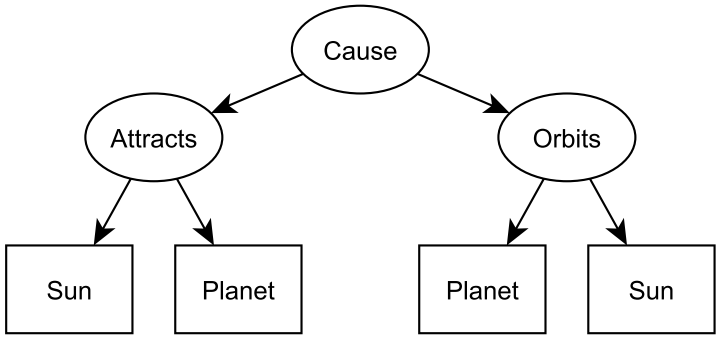


Figure 3.1: The predicate-structure Cause(Attracts(`Sun`, `Planet`), Orbits(`Planet`, `Sun`)).

1. **Structural consistency**: The mapping must be *structurally consistent*, meaning it must observe *parallel connectivity* and *one-to-one correspondence*.

   Parallel connectivity requires matching relations to have matching arguments. For example, given Cause(Attracts(...), Orbits(...)) as base and Cause(Gravity(...), Attracts(...)) as target, the two Cause relations can not be matched because their arguments cannot match. As a result of this constraint, mappings will always include objects that are descendants of matching predicates, preventing analogies consisting only of predicates without being grounded in matching objects.

   One-to-one correspondence requires that any element in the base may only match one element in the target, and vice versa. In other words, a predicate or object that is matched to an element in the analogy, can not also match a second without creating an inconsistent analogy.

2. **Relational focus**: The analogical mapping *must* involve predicates with matching name, number of arguments and order of arguments, but does not have to involve entities with matching names. For example, Attracts(`Sun`, `Planet`) can match Attracts(`Nucleus`, `Electron`), as the two predicates are identical and their arguments can match despite their different names, being objects. However, the same relation cannot match Warms(`Sun`, `Planet`) because the predicates differ in name, despite the objects being matchable. See Figure 3.2.

3. **Systematicity**: The mapping tends to match connected systems of relations, that is, deeply nested interconnected substructures involving higher-order predicates. Matching relations that are interconnected by higher-order relations form a better analogy than an equal number of otherwise unconnected matches.

As evidenced by these constraints, analogies in SMT are based purely on structure. Any object in the base can match any object in the target, as long as there is already a match in the system of relations in which the objects participate. The content of the match, the entities to which the objects correspond, is ignored. The object `Sun` can match `Nucleus`, despite referring to very different entities, if they play corresponding roles in a common relational structure.

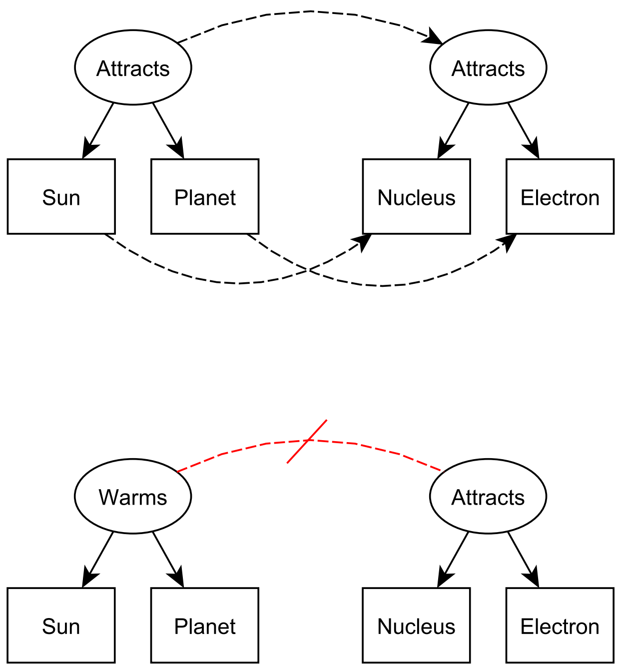


Figure 3.2: Relational focus examples. In the first figure, the two Attracts predicates match on name and the number and order of their arguments. If the match is to succeed, their arguments must also match (as per parallel connectivity). This is the case here, as the arguments are objects, allowing them to match despite differing names. In the second figure, the predicates cannot match due to the differing names. As a result, their arguments are also not matched as they do not participate in a matching relation.

3.2 The Structure-Mapping Engine

The Structure-Mapping Engine (SME) (Falkenhainer, Forbus & Gentner, 1989) is an implementation of analogy derivation as described in SMT. It has been widely used as a module in various analogy-related models and systems (e.g., Ferguson, 1994; Friedman, Taylor & Forbus, 2009; Forbus, Gentner & Law, 1995; Yan, Forbus & Gentner, 2003). Given a base and a target structure, SME finds all structurally consistent analogical mappings between those structures.

In order to perform analogical matching in the Receiver model, the algorithm SME uses to construct mappings between structures was reimplemented, with certain modifications. The SME algorithm will be summarized here, as well as the areas in which the reimplementation diverges from the original[1].

3.2.1 Overview of the algorithm

A mapping found by SME is referred to as a *global mapping*, or *gmap*. As per SMT, only structural criteria are used to construct mappings. A set of *match rules* encodes these criteria, specifying which pairwise matches are valid.

[1]The original SME algorithm is described in great detail in (Falkenhainer et al., 1989).

The algorithm consists of three stages:

1. **Local match construction**: Match rules are applied to find all pairs of base and target items that can potentially match. For each such pair, a *match hypothesis* represents the possibility that this local match is part of a global mapping.

2. **Gmap construction**: Match hypotheses are combined into maximally consistent collections.

3. **Candidate inference construction**: Inferences are derived for each gmap.

4. **Match evaluation**: Evaluation scores are computed for each gmap.

The latter two stages are not relevant for this thesis: the model does not use inference construction, and the mappings are evaluated in the re-representation process, using a method that differs from SME (see Section 3.4.3, p. 29).

The first two stages, both concerning the construction of a mapping, will now be described in more detail.

3.2.2 Local match construction

SME begins by detecting potential matches between the items in the base and the target. Two types of match rules are used to perform this task efficiently: *filter* and *intern* rules. A filter rule is applied to each pair of predicates from the base and target, resulting in an initial set of match hypotheses. For example, a filter rule might hypothesize a match for each pair of predicates with a matching name. An intern rule is applied only to the pair of items of each newly created match hypothesis, creating additional matches suggested by the given hypothesis.

Hypothesizing matches between every pair of objects would create combinatorial explosion, but an intern rule can be used to create match hypotheses for entities in corresponding argument positions of other match hypotheses. As a result, hypothesized object matches are only created for cases where the required structural consistency exists, preventing an intractable number of match hypotheses.

The following match rules are used:

1. Filter rule: If the predicate names are equal, create a match hypothesis.

2. Intern rule: If the given hypothesis matches two predicates, create match hypotheses between any corresponding arguments that are entities.

3. Intern rule: As the previous rule, but applied only to commutative predicates, and entities do not have to be in corresponding argument positions (as the arguments are not ordered).

The resulting collection of match hypotheses can be interpreted as a directed acyclic graph with one or more roots, much like the base and target structures being matched.

3.2.3 Global match construction

Once local match hypotheses have been constructed, the SME algorithm combines them into collections of internally consistent global matches (gmaps). Gmaps are collections of hypotheses that are *maximal* and *structurally consistent*.

The concept of structural consistency comes from SMT, and can be translated directly: one-to-one correspondence requires that none the match hypotheses in the collection assign the same base item to multiple targets or vice versa. In other words, every base and every target item can only be used in a single hypothesis. Parallel connectivity requires that for a hypothesis in the collection, all match hypotheses that pair its arguments are also in the collection.

A collection of match hypotheses is maximal if adding any additional match hypothesis would result in the collection becoming structurally inconsistent.

The construction of gmaps is performed in two steps:

1. Compute consistency relationships: For each hypothesis, compute information used to determine gmap consistency in later steps.
2. Combine match hypotheses: Gmaps are computed by combining hypotheses as follows:
   (a) Combine the descendants of the highest-order structurally consistent hypotheses (roots) into an initial set of gmaps.
   (b) Merge gmaps that have overlapping structure in the base items of the hypotheses, and are structurally consistent with each other.
   (c) Complete the gmaps by merging all gmaps from the previous step, subject to structural consistency and keeping only the maximal results.

When computing consistency relationships, the SME algorithm generates the information required for a number concepts that are subsequently used to build gmaps. Given a match hypothesis $MH_i(b, t)$ involving base b and target t, they are defined as follows:

- $Emaps(MH(b, t))$: An *emap* is a match hypotheses involving two entities. The *Emaps* set for a hypothesis represents the set of emaps implied by that MH. Per the parallel connectivity constraint, this is simply the set of emaps among the descendants of the hypothesis.
- $Conflicting(MH(b, t))$: This set is the set of match hypotheses that postulate alternative matches of b or t. Per the one-to-one correspondence constraint, these alternative hypotheses can never be in the same gmap.
- $NoGood(MH_i))$: The set of hypotheses that can never be present in the same gmap as MH_i. It is defined recursively as follows: if MH_i is an emap, it is equal to $Conflicting(MH_i)$. Else, it is the union of $Conflicting(MH_i)$ with the *NoGood* set of all of its descendants:

$$NoGood(MH_i) = Conflicting(MH_i) \cup \bigcup_{MH_i \in Args(MH_i)} NoGood(MH_i)$$

- $Inconsistent(MH_i)$): A hypothesis is *inconsistent* if the emaps supported by some of its descendants conflict with those implied by other descendants, i.e.,

$$Inconsistent(MH_i) \iff Emaps(MH_i) \cap NoGood(MH_i) \neq \emptyset$$

SME's global match construction step uses these concepts in collecting sets of consistent match hyptheses. An initial set of gmaps is formed working downward from the roots, as gmaps are defined to be maximal. If a root is consistent, the subgraph descending from it must also be consistent, and can therefore form a gmap. Typically, several roots exist, leading to several initial gmaps. These must then be merged into larger, maximal collections of structurally consistent match hypotheses in order to obtain proper, maximal gmaps.

Two gmaps are consistent with each other if no element in either gmap is part of the *NoGood* set of the other gmap. The set *NoGood*(*Gmap*) is simply the union of the *NoGood* sets of all hypotheses in the gmap.

After forming the initial gmaps, the SME algorithm performs a second step in which gmaps that have some connection in their base structure that does not exist in the target are merged. Then, the final step performs successive unions on the gmaps, keeping only combinations that are maximal and consistent.

On these second and third steps of gmap construction, the reimplementation differs from SME. They are replaced by a single step in which *all* maximal and consistent combinations of the initial gmaps are generated. This method is simpler, avoiding the heuristic involving deep structural comparisons, while still guaranteeing all mappings of interest are generated.

3.3 Base representations of actions and goals

The core of the Meaning-mapping system is the analogical matching component. As this component employs structure-mapping, the representations used must be concept graph structures. The base representation of an action or goal is the fundamental concept graph representation, before any *re*-representation has occurred. The base representations for actions and goals are discussed in this section.

3.3.1 Action representation

An action is represented as a chain of `Position` objects, linked by Before predicates. The first `Position` argument of a Before predicate indicates the starting position of the token on the board, and the second argument indicates the ending position. The location on the board is represented by means of an x and a y value. Figure 3.3 shows an example of a simple action.

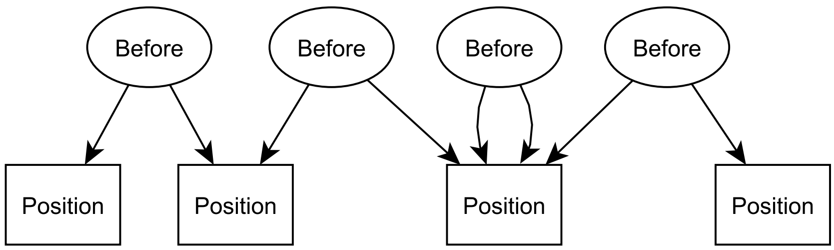


Figure 3.3: A concept graph for a simple action, consisting of two steps, a pause (one time step in length), and another step.

In the terminology adopted in the Parsing section, the Before predicate is the movement m, and the two `Position` objects describe the game states s_i and s_{i+1}. Hence, each Before relation describes a single time step.

3.3.2 Goal representations

As discussed earlier, two types of goals exist: positional goals, which concern the location on the board where the Receiver must place his token, and orientation goals, which concern the specific orientation of the token within that board location.

Position

We have seen how the base action representation is effectively a sequence of positions in which the Sender placed her token. To a human Receiver, most of those positions are obviously not analogous to their goal position. Instead, an explicit pause is used by Senders to indicate that goal. Not all pauses are goal positions, however. For example, the Sender may use a pause to show the Receiver that she is about to perform a communicative action that signals an orientation. If the Sender later performs a longer pause at a different location to indicate the goal position, Receivers will correctly ignore the earlier, shorter pause(s) and interpret only that longer pause as signaling the goal.

Based on these observations, some requirements for the base representation of goals are clear: it should not match with the Before structures of the base action representation, as then it would match movements that are not pauses. Even for explicit pauses, not every pause is performed equally: some pauses are more significant than others, through duration, order of occurrence, or other factors. The Receiver can therefore be assumed to perform a number of reasoning steps before concluding which pauses in a given action are signaling a goal position (if any). The model hypothesizes that these reasoning steps take the form of re-representation operators being applied to the base representation. This process is discussed in more detail later in this chapter (see Section 3.4, p. 27).

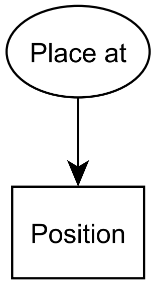


Figure 3.4: Concept graph for a positional goal.

In the model, a positional goal is represented as a very simple graph structure, shown in Figure 3.4, containing only a Place-at predicate whose single argument is a `Position` object. This cannot match with a Before relation as found in the base action representation due to the difference in predicate, fulfilling the first requirement identified above. The aspects relating to pauses will be discussed in the description of the relevant re-representation operators (see Section 3.5, p. 31).

Orientation

Goals concerning the orientation of the token represent the shape of Receiver's token in a relatively detailed way. While the most common and successful strategies use only a specific aspect of the Receiver's token (such as where it is 'pointing', as in Figure B.1), more complex signals can still be understood by human subjects (e.g., Figure B.2). The representations of goal orientations in the model should therefore be rich enough to even allow analogies based on the actual shapes of the tokens.

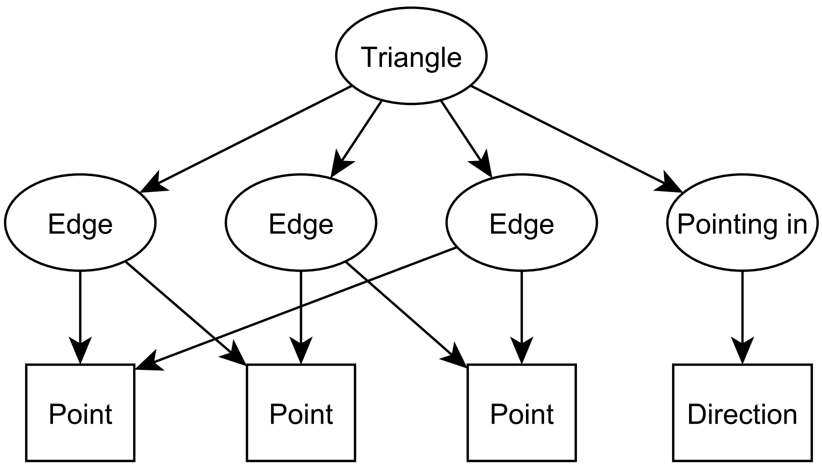


Figure 3.5: Concept graph for a triangle shape.

For a triangle piece, the concept graph includes a Triangle predicate with three Edge predicates as arguments, each of which has two `Point` object arguments. These `Point` objects describe locations inside a grid cell on the TCG board[2]. Such a location does not describe an absolute position on the board, but a point in a local coordinate system, where the origin is the center of a grid cell. Hence, a `Point` object does not represent a specific position on the TCG board, but a relative position invariant across grid cells.

For example, the representation of a triangle token 'pointing' to the east is identical no matter where the token is located on the TCG board. This allows a Receiver to easily identify two shape orientations as being identical, regardless of their location. This makes the Receiver's task trivial when both players have the same shape, leading the Sender to simply position it as the Receiver should. The goal orientation that matches the orientation signaled by the sender will be obvious due to their fully identical representations.

In addition to the edges of the triangle shape, the Triangle predicate has a fourth argument, which is a Pointing-in predicate that has a single `Direction` argument. The `Direction` object describes an angle, in this case the angle in which the Receiver perceives this triangle to be oriented. The underlying assumption is that although the triangle token is equilateral and therefore ambiguous with regards to its orientation, a human observer will nevertheless *perceive* it as 'pointing' in a certain direction, depending on geometric context (Palmer & Bucher, 1981) and a variety of other external factors such as motion and texture (Attneave, 1968; Bucher & Palmer, 1985; Palmer, 1980; Palmer & Bucher, 1982). The full concept graph for a triangle is shown in Figure 3.5.

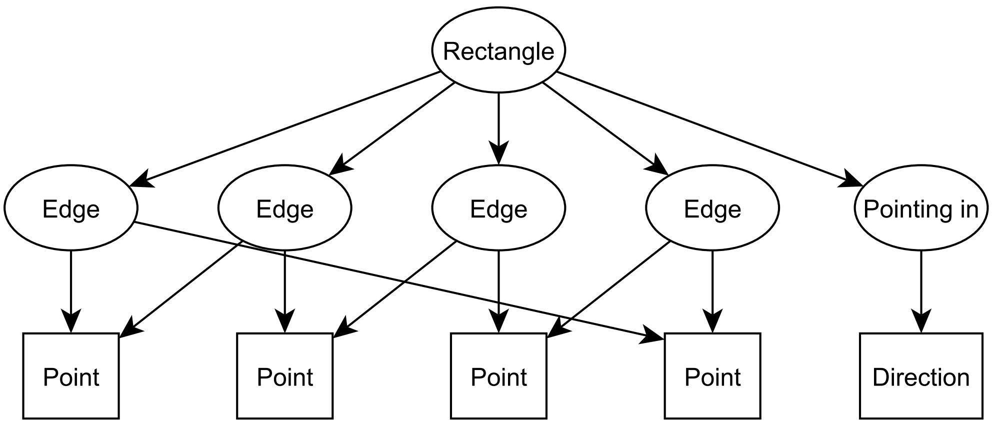


Figure 3.6: Concept graph for a rectangle shape.

A rectangle shape is represented similarly. Of course, the Rectangle predicate requires four Edge predicates as its arguments rather than three. The concept graph for a rectangle is shown in Figure 3.6.

When the token the Receiver must position is a circle, he does not require an orientation, and will consequently not look for that information in a Sender's signal. Nevertheless, the concept graph representation of a circle shape is given in Figure 3.7 for

[2]Recall that the TCG playing board consists of a 3 by 3 grid of cells of identical size.

completeness.

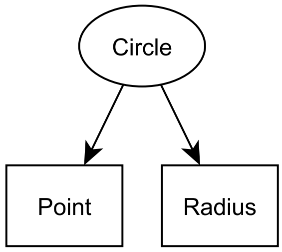


Figure 3.7: Concept graph for a circle shape.

The base representations of the action cannot be analogically matched to any of the goal representations, because there is no matching structure. The Receiver must first analyze the action in order to robustly identify the relevant information that is analogous to a goal. The model hypothesizes that this is a process of *re-representation*, and the following section will discuss this in more detail.

3.4 Re-representation

Re-representation transforms the representation of one or both analogs in order to enable or improve an analogical match. Yan et al. (2003) describe a theory of re-representation in a structure-mapping context, along with one of the few implementations of re-representation. In their model, the re-representation process occurs after the mappings produced by the analogical matcher (SME) have been evaluated. If the mappings are not of sufficient quality, re-representation is one of the ways in which the reasoning process can continue, along with more drastic alternatives such as selecting a different base analog or abandoning the reasoning line in question.

Yan et al. describe a set of *re-representation opportunities* that can be detected in the mappings produced by SME. The detection process involves both the base and the target, as it attempts to find specific issues in a match related to the structural consistency constraints of SMT. Re-representation methods are applied to these opportunities in order to generate re-representation suggestions. One or more of these suggestions is applied, transforming the analog(s) involved. The analogical match is then retried using the modified concept structures. These steps repeat until a match of sufficient quality is found (or the process is aborted).

However, their approach is not sufficient for the Receiver model. It is based on detecting re-representation opportunities in an existing analogical mapping, and applying the appropriate re-representation strategy for that opportunity to improve the mapping. However, the problem that re-representation needs to solve in the meaning-mapping process is not one of improving an existing analogical match.

Instead, a more fundamental problem must be tackled: that of making an analogical match between base and target possible in the first place. Initially, the base representations of the action and the possible goals cannot match at all. Only by re-representing them can an analogical match be found.

The re-representation method implemented in the model is described in the remainder of this section. In short, a search is performed through the space of possible (re)representations of both the action and the potential goals. Re-representations are generated by *re-representation operators* that perform some reasoning step in order to transform a representation, adding inferred knowledge about the action or goal. After each re-representation, an analogical match is attempted between the action and goals. If a good match is found, the matching goal is returned as the goal signaled by the communicative action.

First the concept of a re-representation operator will be specified in more detail, followed by a description of the algorithm.

3.4.1 Re-representation operator

Re-representation is performed via the application of re-representation operators. A re-representation operator r is a tuple $\langle p, t \rangle$, in which t is a function that transforms a representation into a different one, and p is a predicate function that takes a representation and returns whether this operator can re-represent it successfully. For a given operator r, its t and p parts will be referred to here as r^t and r^p respectively. The application of an operator to a graph[3] G can be written as $r(G)$:

$$r(G) = \begin{cases} r^t(G) & \text{if } r^p(G) = \text{true} \\ \emptyset & \text{if } r^p(G) = \text{false} \end{cases}$$

The application $r(G)$ returns a set of graphs resulting from G being re-represented by the operator r. This can be an empty set if no re-representation is possible using that operator. Given a set of re-representation operators R and a graph G, the set of all possible re-representations of G is given by:

$$R(G) = \left\{ r^t(G) \mid \langle r^p, r^t \rangle \in R \wedge r^p(G) \right\}$$

3.4.2 Re-representation process

The process of re-representing until a match between a goal and the action is found is essentially a breadth-first search of the tree of possible re-representations of the action. The root of the tree is formed by the action in its base representation, and each node of the tree is a representation resulting from the application of a re-representation operator. Hence, nodes that are further down the tree are the result of multiple applications of operators. Theoretically it is possible that operators can be infinitely chained this way, requiring an upper limit on the *re-representation depth*, i.e., the number of consecutive operator applications. Figure 3.8 illustrates the concept of a representation tree.

[3]Note that in the meaning-mapping system, all representations are concept graphs (predicate-structures).

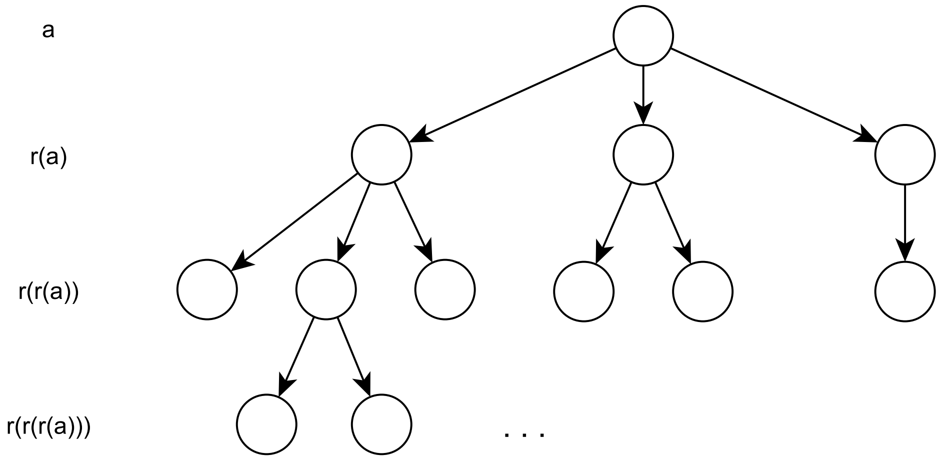


Figure 3.8: A sketch of the tree of representations formed by consecutive application of re-representation operators. Each node in the tree is a concept graph (predicate structure) representing the base action a that forms the root of the tree. An edge indicates a successful application of a re-representation operator. On the left are the levels of re-representation, starting at the base action a, and progressing to $r(r(r(a)))$, which means that level of the tree is reached through 3 operator applications. Note that these do not have to be applications of the same operator, and are in fact likely to be different operators.

As the search process examines an action representation (a node of the tree), a similarly structured re-representation search is performed for every available goal. Each representation of every goal is tested for an analogical match with the given action representation.

Once the search has covered a level of the action representation tree – where a *level* is a set of representations with the same re-representation depth – the results of the analogical matching up to that point are examined. If no mappings have been found, the search continues to the next level. If a single mapping is found, the search completes and the mapping is returned. If multiple mappings are found, they are compared to determine if one of the mappings scores better than the rest. If so, the results are not ambiguous as there is one clear result. If there are multiple mappings with equal quality, the results are ambiguous. In the first case, the best mapping is returned as the result of the search, while in the second case the search continues until an unambiguous result is found (or the depth limit is reached).

A sketch of this process is given in Algorithm 1.

3.4.3 Evaluating analogical matches

When multiple analogical mappings have been found, they are first compared on *structure*. If that does not find one mapping to be better than all others, they are compared on *content*.

This is implemented as follows. All results are grouped according to the systematicity of the analogical maps. The systematicity is measured using the structural evaluation score (SES) also implemented in SME. If there is a single mapping that scores better

Algorithm 1 Sketch of the re-representation search process.

A = set with only the base action representation
$M = \emptyset$
repeat
  for all $a \in A$ **do**
    B = set of base representations of goals
    repeat
      for all $b \in B$ **do**
        if analogical match between b and a **then**
          $M = M \cup$ analogical mapping(s) found
        end if
      end for
      $B = \{r(b) \mid b \in B \wedge r \in R\}$
    until $B = \emptyset$ **or** unambiguous result $\in M$
  end for
  $A = \{r(a) \mid a \in A \wedge r \in R\}$
until $A = \emptyset$ **or** unambiguous result $\in M$
return M

than all others, it forms the unambiguous best result. If there exist multiple mappings of equal SES score, these are then grouped according to their object content match score. This simply counts the number of objects in the mapping that match in content. Again, if a single mapping scores better than all others on this measure, it is considered to be an unambiguous result and the search ends. If not, the current results are considered ambiguous and the search process continues.

The two scoring measures used for these comparisons will now be discussed in more detail.

Structural Evaluation Score

The structural evaluation score rewards mappings involving deeply-nested structures, under the assumption that such mappings form a better analogical match. A simplified *trickle-down* method (Forbus & Gentner, 1989) is used[4]. The SES is computed by assigning a score to each match hypothesis (MH) and summing those scores. The score includes a value inherited from the parents of the MH, which is incremented and passed on to the children of this MH.

As a result, this score increases as one travels from the predicates of the mapping to the object, rewarding deep nesting. Equations 3.1 and 3.2 define the measure in recursive form[5] where $C(x)$ is a function returning the children of a match hypothesis,

[4]The simplification of trickle-down as used here lies in its parameter configuration. Each MH has a base score of 1, and the value it "trickles down" is not scaled (effectively multiplied by 1). Though not as finely tuned as the parameters used in (Forbus & Gentner, 1989) or SME, the results of this more general approach are sufficient for the analogical mappings found by this model.

[5]Note that the score-mh function in Equation 3.1 computes only part of the score of the MH, namely that part that is based on the parent it is receiving the d value from. In computing the score for a mapping, score-mh will be computed multiple times with differing parameters for a MH with multiple

and Roots(x) returns all root match hypotheses of a given mapping.

$$\text{score-mh}(x, d) = \begin{cases} d + \sum_{c \in C(x)} \text{score-mh}(c, d+1) & \text{if } C(x) \neq \emptyset \\ d & \text{if } C(x) = \emptyset \end{cases} \tag{3.1}$$

$$\text{score-map}(x) = \sum_{r \in \text{Roots}(x)} \text{score-mh}(r, 0) \tag{3.2}$$

Object content match score

The object content match score is defined as the number of objects mapped by a match hypothesis that match in type and values. For example, given a MH that maps two `Point` objects, where the `Point` objects contain an x and a y value, it will contribute 1 point to the score if those two objects have equal x and y values.

This content-based score proves necessary when an action concept graph is matched with multiple structurally identical goals, such as graphs representing a triangle shape in different orientations. Without considering object content, each match is of equal quality, when in actuality one of the goals may be more similar or even identical to the action graph. Performing the final disambiguation step of comparing on content matches allows one to select that goal as the best result.

3.5 Re-representation operator definitions

As discussed in the previous section, over the course of the re-representation process the action and goal representations are transformed by re-representation operators. In order to interpret common types of TCG signals employed by human players, a set of operators is required that allows the Meaning-mapping system to re-represent actions such that the analogical match intended by the Sender can be made.

As part of the re-representation process, the operators should be based on the reasoning or inference steps human players are hypothesized to make when they attempt to find an analogical match between a communicative action and possible goals. However, detailed study of human TCG strategies is necessary to develop operators that are well-supported by experimental data, and has as of yet none has been performed. De Ruiter et al. (2010) did enumerate the general strategies used by Senders, but they did not perform more detailed analysis of the reasoning and re-representation involved. As the primary goal of this thesis is to provide some validation of analogy and re-representation in context of the model as a whole, rather than specific operators, such analysis is outside the scope of this research as well.

The operators that have been designed strive to be plausible, in that they perform relatively small, simple steps of analysis and transformation. To simplify the complex task of designing a coherent set of operators capable of performing the reasoning steps required for common strategies, they are domain-specific rather than general operations. Clearly a comprehensive set of highly general re-representation operators would be valuable, but more research is required to inform their design.[6]

parents.

[6] See Chapter 5 for more discussion on the topic of operator generality and future research.

As discussed previously, TCG trials can be divided into 'easy' and 'hard' trials. Strategies used in easy trials are relatively simple, but in hard trials human Senders use more varied strategies. These can be broadly categorized into three types, in order to clarify which strategies the model is intended to be able to interpret.

The most common strategy used by human subjects (per de Ruiter et al. 2010, see also Section 1.2, p. 6) is to perform a movement that signals a specific aspect of the Receiver's token in its goal state: the direction it 'points in'. This type of signal will be referred to as an *aspect*-based signal (for an example see Figure B.1, page 62).

A different type is the *act*-based signal, intended to signal one or more actions the Receiver should perform to reach his goal state, rather than information concerning the goal state itself. If the Sender's token can show rotation, she might for example rotate it clockwise twice to signal that the Receiver should do the same. If her token is not capable of rotating, more abstract signals would be required to communicate the same information (see Figure B.3, page 64).

The last type discussed here is the *shape*-based signal. This strategy involves the Sender performing a movement that signals the Receiver's token as a whole, oriented in accordance with the goal state. For example, the Sender might move in a rectangular or triangular path over the board, 'drawing' the Receiver's shape in the correct orientation (see Figure B.2, page 63).

The model hypothesizes that each of these strategy types requires a different re-representation strategy (and operators capable of executing it) as each uses a different facet of the Receiver's representations. For example, in order to interpret the Sender's path as a shape that is analogous to a goal state (i.e., a shape signal), the action must first be re-represented as (potentially) representing a shape. Without re-representation operators that can perform the steps required to reach that action representation, the Receiver is unable to understand the signal.

Currently, the model's re-representation operators cover only the aspect-based signal type shown to be by far the most common type used by human players. Act-based signals were also found by de Ruiter et al. (2010), but were far less common than aspect-based signals. No Sender player used a shape-based signal in their experiment. Nevertheless, the base representations were designed to be rich enough to plausibly allow re-representation that lets the model interpret other signal types as well, if the required operators are implemented in the future.

The operators that are currently implemented in the model are described below. For each operator r, the transformation that the operator performs on a concept graph G by applying $r^t(G)$ is given, as well as the requirements G must fulfill in order to be re-represented by the operator, tested via $r^p(G)$.

3.5.1 Mark pauses

This operator explicitly marks pauses in the action, allowing other re-representation operators to easily detect them and reason with and about these pauses for future transformations.

Requirements G must contain one or more Before relations that represent a pause, meaning the two `Position` arguments are identical.

Transformation Every pause in G is transformed from a Before relation to a Pause at predicate with a single `Position` argument. The result of applying this transformation to the graph shown in Figure 3.3 is shown in Figure 3.9.

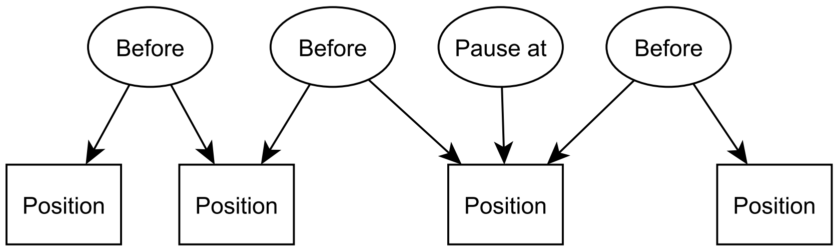


Figure 3.9: The graph shown in Figure 3.3 (p. 24), re-represented by the `Mark pauses` operator.

3.5.2 Extract most significant pause

As discussed in the introduction of this thesis, pauses are an important tool for the Sender. Simply pausing in a certain location can communicate to the Receiver that he should place his token there. In 'easy' trials with matching tokens for both players, it can even be used to signal the orientation of the token.

However, pauses are also used by Senders in other ways, such as delimiting a certain part of the movement sequence as being communicative. The Receiver can therefore not assume all pauses indicate a goal state. Only re-representing pausing movements as explicit pauses (using the MARK PAUSES operator) is not sufficient, some additional evidence is needed before a pause is considered to signal for a goal position.

The assumption underlying this operator is that, when multiple pauses are present, the most 'important' or *significant* pause is taken to signal the goal position (and/or orientation). Importance is assigned to pauses that are repeated more than other pauses: by pausing for a longer period of time in a certain location, the Sender leaves no doubt the pause is intentional and communicative of important information.

If that distinction cannot be made, the first pause is considered more important than later pauses. The assumption is that the Sender can be expected to send the most easily signaled information first, in order to guarantee its successful transfer even if later signaling of other information fails.

Requirements G must contain at least one Pause at relation.

Transformation The operator counts the frequency of each distinct `Position` in G on which the Sender paused. If one occurs more often than any other, the Pause at predicate in that pause relation is replaced by a Place at predicate. The presence of this predicate allows the action to match with goal positions, which also contain a Place at predicate. In fact, the other relations in the action graph are no longer relevant and can safely be removed. The resulting action graph allows a very simple match to the positional goal graph that contains the same `Position`.

If no pause `Position` occurs more than others, the same transformation is performed using the first pause.

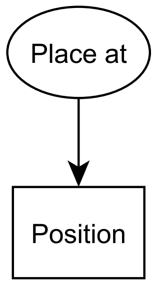


Figure 3.10: The graph shown in Figure 3.9, re-represented by the Extract most significant pause operator. Note that the resulting graph shown here can easily be matched to the goal position representation shown in Figure 3.4.

3.5.3 Fold repetitions

This operator allows the Receiver to extract a repetition present in the action for further analysis in the search process. Recall that a common strategy used by human Sender players to signal the orientation of the Receiver's token uses a repeated movement in the direction the token should 'point to'. This strategy had a high success rate in the experiment described by de Ruiter et al. (2010), showing that human Receivers are adept at understanding the signal.

The repetition taken as a whole can be seen as ambiguous: the Sender moves in one direction, and then in the exact opposite direction. Human Receivers successfully understand that the direction of the first step is the intended signal, and the following step is simply a step back required to perform the signaling step again. The operator implements the same concept.

Requirements G must contain multiple Before relations, and not yet contain a Repeated predicate (which would show it has been re-represented by this operator already).

Transformation Given the sequence of Before relations in G, the operator finds all unique subsequences that are repeated one or more times. Figure 3.11 illustrates the repetitions that are detected. Note that the subsequence $[2, 3]$, for example, is not considered: The operator assumes a parsing where the first movement is always a significant part of the repetition, and therefore only subsequences including that movement are considered. The subsequence $[2, 3]$ does not, and can therefore be ignored. Hence, if the first movement of the sequence is not repeated, this operator will not perform any transformation.

For every repetition that was found, a separate graph G' is generated with only those Before relations that are part of that repetition. A Repeated predicate is added, with three arguments: the two Before predicates that form the start and end of the

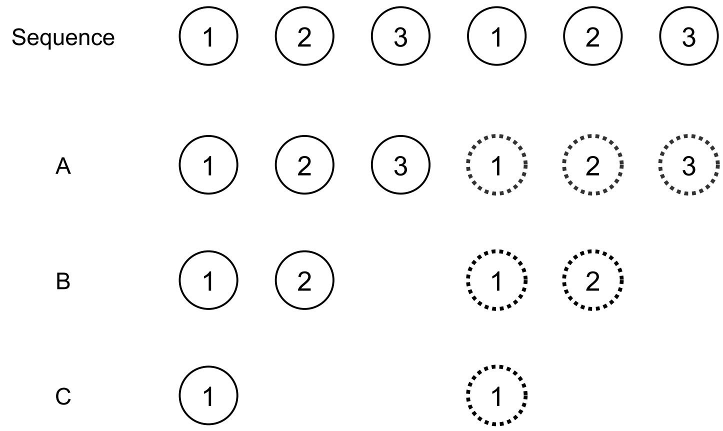


Figure 3.11: A sequence of Before relations, represented by numbered circles. Below the original sequence, three repeated subsequences are shown (A, B, and C), with their repetition shown in dotted outline. Note that B and C are themselves subsequences of another repeated subsequence, but are considered separately.

repetition[7], as well as a `Repetitions` object that contains the number of repetitions. If multiple graphs are generated by the this operator, they are simply added to the set of representations to be considered by the search process, as would a single graph.

3.5.4 Find repetition direction

As discussed earlier, one of the most common strategies in 'hard' trials[8] involves the use of repeated movements to signal an orientation. This operator allows the Receiver model to perform the analysis required to interpret a repeated movement as 'pointing in' a direction, much like certain shapes 'point in' a direction. It simply computes the average direction in which the repeated movements are made.

Requirements G must include a Repeated predicate, which means the graph has a repetition that has been re-represented by the `Fold repetitions` operator. In addition G must *not* include a Indicating direction predicate, as that indicates it has been re-represented by this operator before.

Transformation Given the set of Before relations forming the repetition in G, i.e., a set of movement representations, the mean vector of those movements is computed. The vector of a movement is computed by subtracting its starting coordinate from its ending coordinate. For example, a Before predicate with its starting `Position` representing $(1, 1)$ and its ending `Position` representing $(2, 1)$ can be represented as the vector $[1, 0]$.

[7] The Repeated predicate could take every Before that is part of the repetition as argument, but this would require support for variadic predicates (i.e., with varying numbers of arguments) in the structure mapping implementation. Support for such predicates currently does not exist, because the two-argument representation proved to be sufficient for this research.

[8] 'Hard' trials being trials in which the Sender has a token not capable of taking on the precise orientation the Receiver's token must be positioned in.

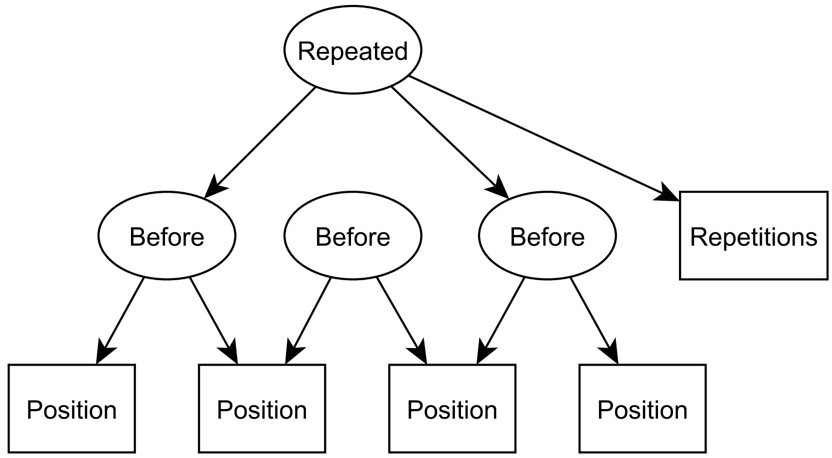


Figure 3.12: A graph resulting from an application of the `Fold repetitions` operator. The original graph contained a repetition of three movements, which has been folded into one such sequence augmented by information about the repetition via the Repeated predicate.

If the mean vector of the repetition is equal to the null vector, the repetition does not indicate a direction and no transformation is performed. Else, an Indicating direction predicate is added that takes the existing Repeated predicate as its first argument, and a new Pointing in predicate as its second argument. The Pointing in predicate has a `Direction` as its only argument, which contains the angle of the mean vector. This angle has been clamped to the angle of the nearest possible goal orientation, as those are the only meaningful angles the Sender can signal. Any variation is likely to be caused by the constraints of the TCG, such as the lack of diagonal movement.

The resulting graph, illustrated in Figure 3.13, can be successfully matched to orientation goals. These goals also have a Pointing in relation, as can be seen in Figure 3.5 for example. These relations will be matched in structure mapping, and the different matching goals can be disambiguated by comparing the content of the `Direction` objects, in order to arrive at the goal that represents the same orientation as the one signaled in the repetition.

3.5.5 Extract shape from pause

In the common strategy of signaling the Receiver's position and orientation simply by pausing in his goal position, with the Sender's token oriented appropriately, the goal position can be found by looking only at the pause. In order to find the matching goal orientation, the actual shape representation of the Sender's token is required, as that is how the possible orientation goals are represented as well. This operator implements the step from reasoning about the pause at a grid cell in itself, to reasoning about the exact positioning of the Sender's token within the cell during that pause. Then, the Receiver can analogically match that shape (the action representation) to their own token shape (the possible goal orientations).

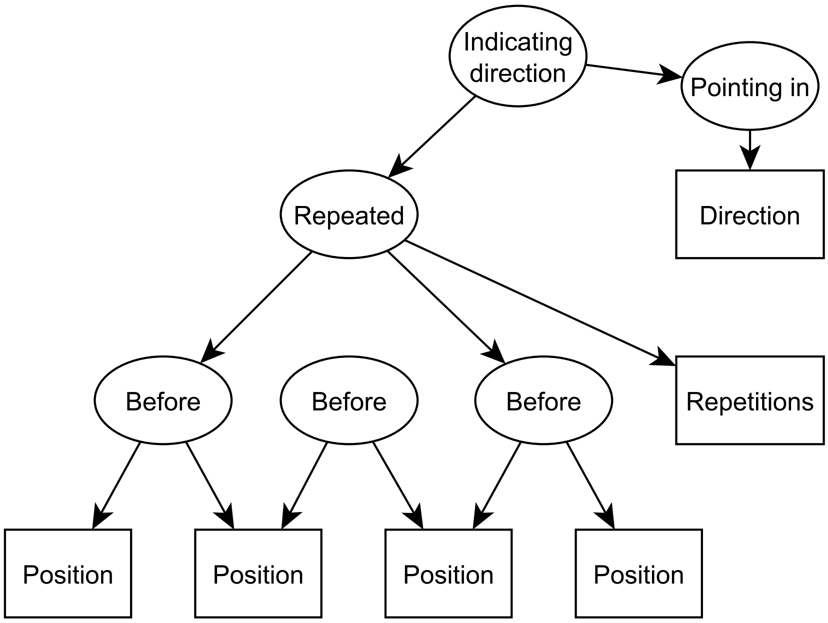


Figure 3.13: The graph shown in Figure 3.12 after applying the `Find repetition direction` operator. The original structure remains, two predicates and an object are added to represent the direction signaled by the repetition.

Requirements G must include a Place at predicate, for similar reasoning as can be found in the MOST SIGNIFICANT PAUSE operator description: when confronted with multiple different pauses, Receivers only examine the pause they consider most likely to signal the goal orientation (as with the goal position).

Transformation The concept graph representation of the shape as it was positioned during the pause is retrieved[9], and returned as the result of the transformation. For example, if the Sender was using a rectangle token and made a (significant) pause at the given `Position`, the concept graph representation of her shape at that point in time (the structure of which is shown in Figure 3.6) forms the result of this transformation.

3.6 Summary

The core hypothesis behind the Meaning-mapping system is that the Sender in the TCG communicates goal states to the Receiver using analogy. Therefore, the Meaning-mapping system must find an analogical match between the communicative action received from the Parsing system and one of the possible goals. The leading structure-mapping theory (SMT) of analogy supplies the analogical constraints on this match,

[9]The shape representation is stored in the `Position` objects, but does not play a role in testing the equality of two `Position` objects for the content match score (described on p. 31). This only occurs when comparing mappings to possible goal positions, in which case the orientation or shape of the token is irrelevant.

implemented in the Structure Mapping Engine (SME) matching algorithm.

The base representations of the action and the goals do not allow a match, and will have to be re-represented. The re-representation process searches the space of possible representations with the aim of finding an unambiguous match. Re-representations are generated by applying operators that encode reasoning steps, analyzing a representation and inferring new knowledge and/or transforming it into a representation that may match better to a goal. Once a match is found of better quality than other matches found so far, the goal that participates in that match is returned as the goal signaled by the communicative action that formed the input of the Meaning-mapping system.

Chapter 4

Results

As described in Section 3.5 (p. 31), the model only possesses re-representation operators designed to interpret aspect-based signals. The much less common act-based and shape-based strategies require re-representation steps the model can therefore not yet perform. For that reason, only aspect-based strategies for 'hard' trials will be covered in this section, along with the typical strategy used on 'easy' trials.

The Parsing system design and implementation makes a simplifying assumption about the movement sequence that forms its input: the sequence must consist of discrete movements of equal duration. This greatly simplifies the parsing process, as it removes the need to handle time as a continuous aspect. The downside is that it makes quantitative analysis using movement data gathered from human subjects in previous experiments difficult. Instead, the following sections consist of a qualitative analysis of the model's performance and its interpretation process.

In all trials discussed here, the parameters of the model were kept constant in each trial, unless explicitly indicated otherwise. The number of hypothesized communicative actions for each parsing generated by the Parsing system was 1. The goal configurations were made up of four possible orientations, with increments of 90 degrees[1]. The re-representation depth limit of both the action and the goal representations was 10 operator applications (not reached in practice with the current set of operators).

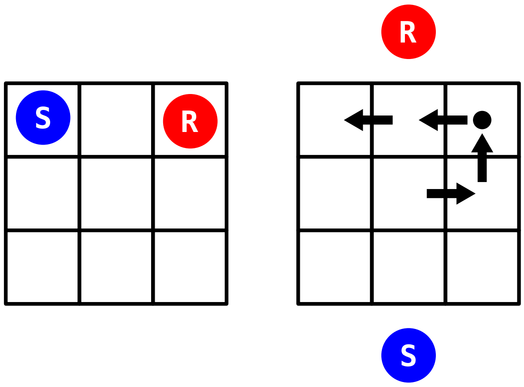
4.1 Easy trials

The first two cases discussed here are 'easy' trials. Case 1 has both players using a circle token (no orientation), while Case 2 gives the Sender a triangle and the Receiver a rectangle (Sender token has more orientations). Case 1 is discussed first, including a look at how variations of the trial's goal configuration would vary the result. This is followed by the results of Case 2.

[1] Research using TCG with human subjects often does not specify the rotation increments. 'Diagonal' goal orientations that occur with 45 degree increments are significantly more difficult to signal due to the TCG constraints not allowing diagonal movement. Perhaps for this reason, de Ruiter et al. (2010) used 90 degree increments. The trials described here will do the same.

4.1.1 Case 1: same shape

The simplest combination of Sender and Receiver shapes is one where both have a circle. Only a position has to be communicated to the Receiver, which is trivial using the typical easy trial strategy of pausing on the goal position. Figure 4.1 shows such a trial, including the movement sequence that will serve as input to the Receiver model.



(a) Goal state. **(b)** Sender's route.

Figure 4.1: Same shape case, where both Sender and Receiver have a circle. In this case, the Sender applies the standard strategy for easy trials, moving her token to the top right (from her perspective), pausing (indicated by a dot), and moving on to her goal position.

Because the Receiver will not be looking for a goal orientation in the Sender's movement. Normally, the process of parsing and meaning-mapping is performed separately for different types of goal information: once to find the goal position, and once to find the goal orientation. Seeing as the orientation is not relevant, only the position-step is required for this trial.

Parsing

Befitting the simple trial and movement sequence, the parsing process is straightforward. The Parsing system generates all unique subsequences of the input movement sequence, and considers each as a hypothetical communicative action in a parsing. Those parsings where the communicative part is in fact fully efficient are removed, as they cannot be communicative. What remains are only the parsings in which the hypothetical communicative action includes the pause performed by the Sender. To determine the order in which these parsings will be further analyzed by the Meaning-mapping system, this set of parsings is sorted on the length of the communicative part.

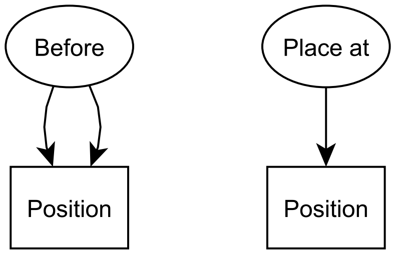
The parsing in which the communicative part consists of only the pause obviously has the shortest possible length while still including all inefficient movements (which, in this case is only the pause). This parsing consists of three actions: first an instrumental action consisting of the first two steps the Sender performs, then the communicative action in the form of the pause, and lastly an instrumental action of the two steps to the Sender's goal position.

The communicative part of this parsing is sent to the Meaning-mapping system, and as we will see in the following section, it is successfully mapped to a goal. Hence, the model has no reason to consider the remaining parsings.

Meaning-mapping

The structure of the base representation of the action is shown in Figure 4.2. The action representation consists of a single Before relation with the same `Position` object as its start and end arguments, representing a time step with no change in position or orientation.

As the Receiver is looking for his goal position, the set of goals consists of the possible goal positions.



(a) Action **(b)** Goals

Figure 4.2: The structure of the base representations of the action and the goals in the circle-circle case.

First, the search process (as shown in Algorithm 1, page 30) will attempt to analogically match this base representation of the action to the different goals. This will not succeed, as the Before and Place at predicates cannot match: they differ in name and in the number of arguments (violating the relational focus constraint). Re-representing the goals is also unsuccessful, as there exist no operators capable of re-representing the graph structure of the goals in question.

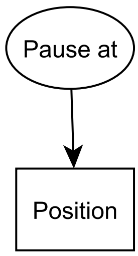


Figure 4.3: The graph of the action after the first re-representation step, in which the MARK PAUSES operator has been applied.

Therefore, the search moves on to action re-representation. Only the MARK PAUSES operator can be successfully applied, resulting in the Pause at relation shown in Figure 4.3. An analogical match to any of the goals is still not possible as the predicates cannot match. Without matches, there is obviously no unambiguous best match, and the search process continues.

In the subsequent re-representation step, there is again only one operator capable of re-representing the action. This time the operator is Most significant pause, which re-represents the Pause at relation that appears to be the most significant as a Place at relation. After this re-representation, the structure of the action's graph re-representation is identical to that of the goals (shown in Figure 4.2).

As a result, when matching the action graph to the different goals, a match is found with every goal. The two Place at predicates match, and of course so do their arguments, as those are objects. The structural evaluation score (SES) for all these matches is obviously equal (at 3).

The ambiguity can be removed, however, by considering the object content match score. The `Position` object of the action representation describes the board position $\langle x = 2, y = 0\rangle$. Only a single goal has a `Position` object that matches this: the goal representing that same board position as a goal position. Of all nine action-goal mappings, that goal has the highest object match score (of 1). This means the search process considers it an unambiguous best match, and returns that goal as its result.

In other words, the Receiver model finalizes its interpretation of the movement sequence as signaling the goal position $\langle x = 2, y = 0\rangle$. This is the intended interpretation of the Sender, and the trial is completed successfully.

Differences in actions or goal configurations

Trials with different goal configurations will have different Sender movements than the sequence discussed here. However, the result will be the same for any sequence of movements with the same general structure of moving to the Receiver's goal position, pausing, and moving to one's own goal position. The exact paths may differ, but as long as they are efficient the Parsing system will correctly interpret them as being instrumental, and they will not affect further reasoning. Similarly, the exact goal location may differ, but this only affects which of the goals ends up being the best match.

Even if the Sender makes a mistake, resulting for example in a longer than necessary path to or from the Receiver's goal position, the model will find the correct interpretation. Though the erroneous path will be seen as inefficient and hence potentially communicative by the Parsing system, the Meaning-mapping system will not find a viable analogical match between those (re-represented) movements and a positional goal. At the same time, the re-representation path as outlined above will also be followed, and will lead to the correct result.

When will the correct goal position *not* be found? The simplest case is one where the Sender does not pause at all, never communicating the required information to the Receiver, or pauses in the wrong location without realizing her error.

A more realistic case would have the Sender pause in the wrong location, and attempt to correct this. In the re-representation process, both pauses will be re-represented as Pause at relations. If both the erroneous pause and the correcting pause are of equal length, the Most significant pause operator will re-represent the first pause as the Place at relation that will match a goal. This is based on the assumption that the Sender will first signal the information that is easiest to communicate, which is typically that which can be signaled through a pause. Later pauses are then more likely to be part of a more complex action signaling orientation, for example.

A human Receiver could conceivably understand that the Sender has no reason to pause twice when no orientation has to be signaled, and conclude one of the pauses was in error. The two pauses could then reasonably be interpreted as an erroneous pause followed by a correcting one. The Receiver model does not currently attempt to handle Sender errors in such a way. The Sender is assumed to execute her planned actions perfectly.

Nevertheless, as the earlier example showed, the Parsing and Meaning-mapping systems are able to handle movement sequences with certain kinds of errors regardless. The case of an erroneous pause will also succeed, as long as the correcting pause has a longer duration than the error. The MOST SIGNIFICANT PAUSE operator assumes Senders know multiple pauses can be ambiguous to the Receiver, and disambiguate them by adding 'weight' to the most significant one by making it longer. Hence, if one pause is longer than any other pause, the MOST SIGNIFICANT PAUSE operator will re-represent only that pause as a Place at relation. Only if no such pause exists does it fall back to weaker assumption of selecting the first pause.

4.1.2 Case 2: different shape with more orientations

The first 'easy' trial concerned a circle-circle configuration where both Sender and Receiver had the same shape as their token. In the second case discussed here, we look at the other type of easy trial. Here, the Sender has a token capable of showing equal or more orientations than the Receiver's token. The Sender will have a triangle, and the Receiver a rectangle.

The Receiver will have to interpret the movement sequence for both the position and orientation of his goal. This means the model will perform two searches, one for each piece of goal information.

Goal position: parsing step

Similar to Case 1, parsing is simple: there is only one inefficient part in the movement sequence, which is the pause. The parsing in which the pause is the only action hypothesized to be communicative will be sent to the Meaning-mapping system.

Goal position: meaning-mapping step

The Meaning-mapping system will process this action in much the same way as the action in case 1. The only difference lies in the board position represented by the action's `Position` object, and consequently the goal position that is found to be the best match. For this trial, this is the location $\langle x = 0, y = 0 \rangle$.

Goal orientation: parsing step

This step is identical to the parsing step in the goal position search, returning the same best parsing and sending the pause action to the Meaning-mapping system.

Goal orientation: meaning-mapping step

As the Receiver is searching for a signaled orientation, the set of goals in the Meaning-mapping system consists of all possible orientations. These are represented as concept

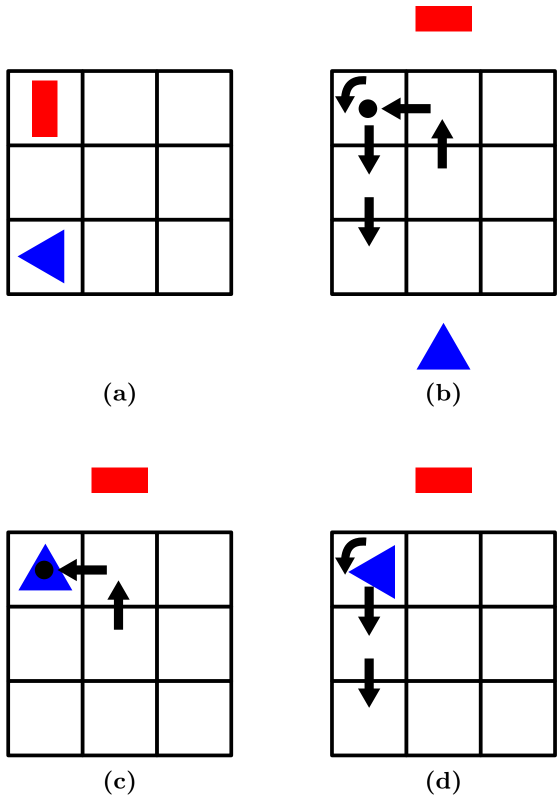


Figure 4.4: Case with differing shape but more visible orientations for the Sender. The goal configuration is shown in (a), and (b) shows the Sender's route in its entirety. She moves to the Receiver's goal location and pauses, as shown in (c), and then proceeds to rotate to her goal orientation and move to her own goal location (d).

graphs representing the Receiver's shape as a whole, oriented in a given direction (as discussed in the previous chapter). Figure 4.5 shows the concept graph for a rectangle shape. The various orientations differ in the values represented by the `Point` and `Direction` objects.

The action is initially represented as a concept graph of the movement (as in Figure 4.2). Much like such a graph cannot match a goal position graph, it cannot match a goal orientation graph. The search process will at first follow the same process of attempting to match and re-representing upon failure to do so, as described in Case 1. It diverges from that case once the action graph has been re-represented to consist of a Place at relation. When searching for position, this would match a goal, but this is not the case when searching for an orientation. After all, the graph structure shown in Figure 4.5 cannot match an action structured as in Figure 4.3.

Therefore, the action graph must be re-represented further. There is only one operator that can re-represent the current graph, which is the EXTRACT SHAPE FROM PAUSE operator. When applied to the graph it transforms it into a graph of the shape the Sender's token had (including its orientation) when she performed that move. In this case, this is a triangle shape 'pointing' north, the structure of which is shown in Figure 4.6.

With this re-representation, the action can match to multiple goals. Though much

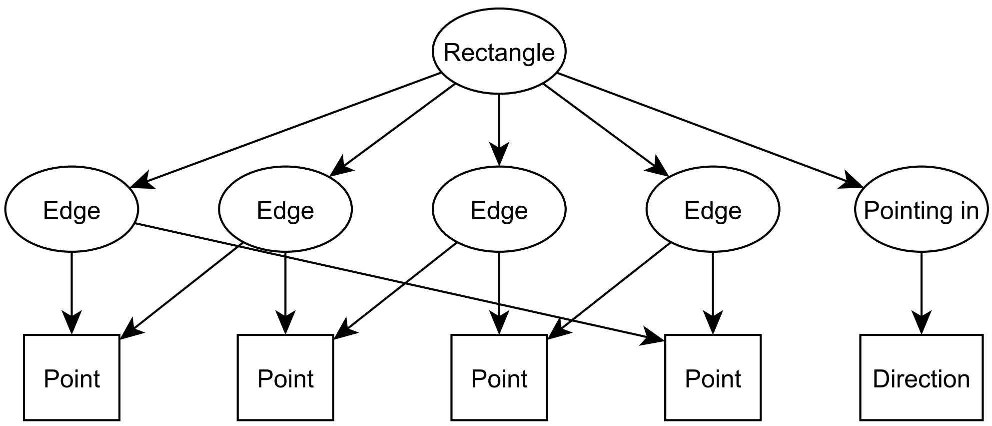


Figure 4.5: Goal orientations in this trial are represented as concept graphs of a rectangle shape.

of the two structures differ, both have a `Pointing-in(Direction)` subgraph. Every goal in the current set of possible goals can match to the action now. The SES of each of these matches is identical, as they match the same structures. Only the values represented by the `Direction` objects differ. Therefore, the object content match score once again disambiguates the matches. Only the rectangle graph oriented in the same direction as the Sender's triangle graph has a matching `Direction` object. This is the graph representing the north-pointing goal, which is the goal returned as the result of the search[2].

If the Receiver also has a triangle shape, the process is very similar. The difference lies in the analogical mappings. Where in the triangle-rectangle case only the `Pointing-in` subgraph can match, in a triangle-triangle case the entire concept graph can be matched. Rather than only the `Direction` object matching in content, all `Point` objects will also match for the matching orientation goal.

4.2 Hard trials

The third and last case is that of a Sender having a token only capable of showing fewer orientations than the Receiver's token. Certain failure cases of similar trial configurations are also discussed.

4.2.1 Case 3: different shape with fewer orientations

In 'hard' trials, the Sender cannot signal the Receiver's goal orientation through the orientation of her own token, because her token has fewer discernible orientations. This is the case for trials where the Sender has a circle token and the Receiver a rectangle or

[2]Note that for a rectangle shape, the orientations 'north' and 'south' are not different in reality. However, the model does represent them separately. This could be rectified by allowing a `Direction` object to contain 'vertical' and 'horizontal' as orientations, which are then made to match the two angles they represent.

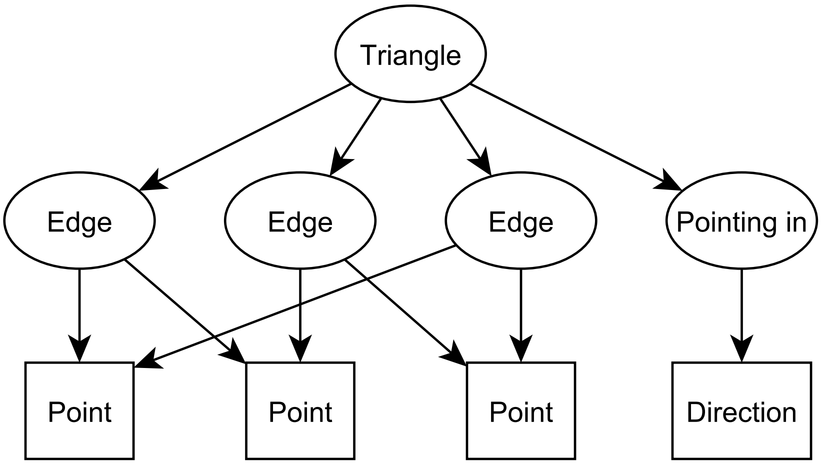
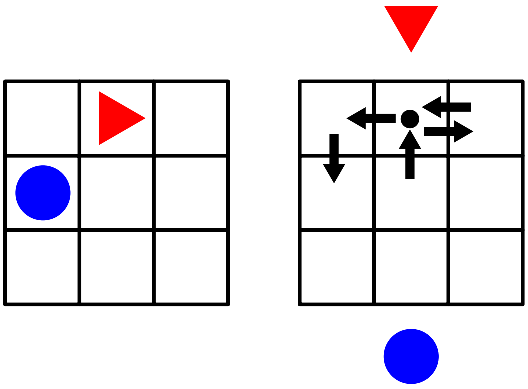


Figure 4.6: Graph structure of the shape of the Sender's token.

triangle, but also for trials where the shapes are a rectangle and triangle respectively. A case of the circle-triangle configuration, shown in Figure 4.7, will be discussed first.



(a) Goal state. **(b)** Sender's route.

Figure 4.7: Case where the Sender cannot signal the Receiver's goal orientation directly. Hence, the Sender uses a different strategy. First she moves to the Receiver's goal position and pauses. Then, she moves in the direction of the Receiver's goal orientation, and steps back to repeat that movement. After several repetitions, she pauses again at the Receiver's goal position and then moves on to her own goal.

The Receiver is looking for a goal position and orientation, but only the orientation search will be discussed here. The position search process for this trial is largely identical to Case 1 and 2.

Goal orientation: parsing step

The Sender has clearly indicated the communicative part of her movement sequence by starting and ending it with a pause. The parsing where the hypothesized communicative part contains everything from the first to the last pause will also contain all inefficient movements while being as short as possible. Therefore, that parsing will be the first to be sent to the Meaning-mapping system.

If the Sender had not performed the second pause, the last step of the repetition back to the Receiver's goal position (i.e., from $\langle x = 2, y = 0\rangle$ to $\langle x = 1, y = 0\rangle$) would *not* be inefficient. After all, that step is part of an efficient route from $\langle x = 2, y = 0\rangle$ to the Sender's own goal at $\langle x = 0, y = 1\rangle$. A parsing could leave that step out of the communicative part and still have it cover all inefficient movement, causing it to be shorter and therefore better by the Parsing system.

This would not affect the interpretation of the Meaning-mapping system in this trial, as the most important part of the repetition (the step in the direction of the Receiver's goal orientation) will still be part of the hypothesized communicative action. Nevertheless, it illustrates how the Sender must take into account how a Receiver will interpret an action without knowing its intended goal.

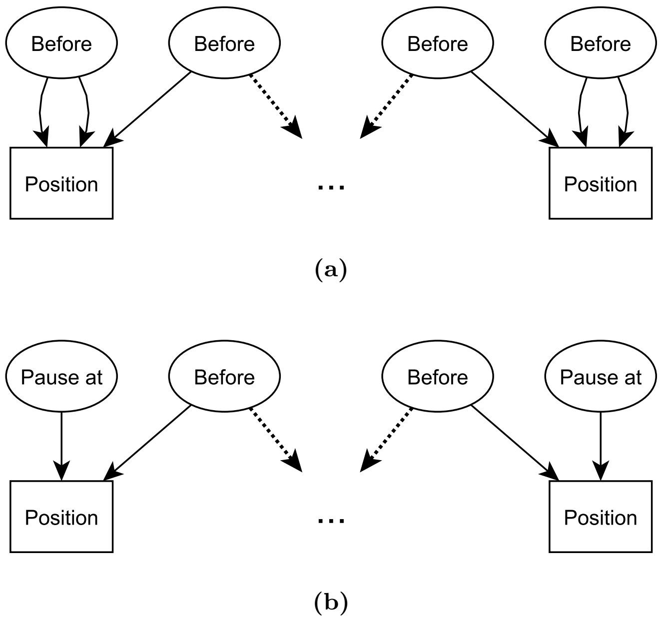
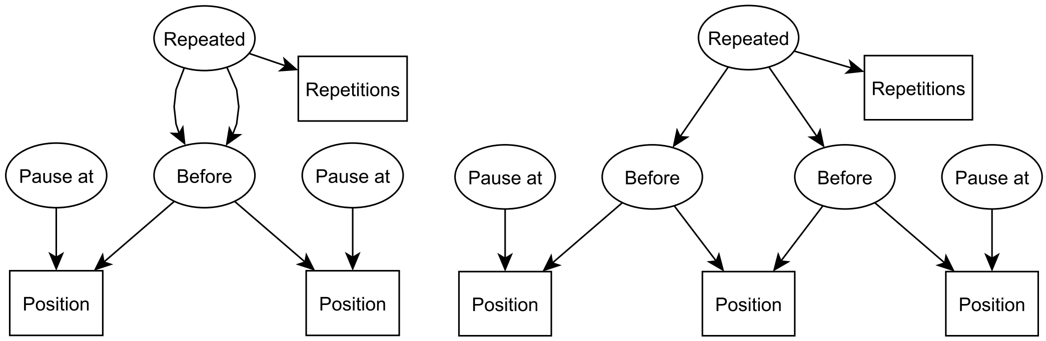


Figure 4.8: Action representations for meaning-mapping phase in Case 3. Figure (a) shows the base representation, (b) shows the result of the MARK PAUSES operator. Repeated Before relations are not shown.

Goal orientation: meaning-mapping step

Initially the FOLD REPETITIONS operator cannot re-represent the action successfully, as the pause at the start is not repeated. The operator requires the first movement to be part of the repetition. The MARK PAUSES operator *can* be applied, re-representing the first and the last Before relations as Pause at.

In the next re-representation step, the FOLD REPETITIONS operator can be applied. The pause at the start is no longer seen as a standard movement represented by a Before relation, but is explicitly marked as a pause. The FOLD REPETITIONS operator only considers Before relations when searching for repetition, so it will ignore the pause. As a result, it now (correctly) considers the Sender's first step to the right to be the start of the repetition. This allows the operator to find two repetitions: the first is one movement long, and consists only of that first step to the right. The second also includes the step back. It returns two re-representations of the original graph: one for each repetition.



(a) **(b)**

Figure 4.9: Action representations in Case 3 resulting from the application of Fold repetitions. In (a) the repetition consists of only one step, in (b) the step back is also included.

At this point in the search process, there is still no action representation that can match to a goal orientation. After another unsuccessful attempt to find an analogical match, the subsequent re-representation step will allow further re-representation. The Find repetition direction operator can be applied to the concept graphs produced by the previous step's Fold repetition application. For the repetition that includes the step back, the operator will not find a clear direction: the direction of the first step is canceled out by the second step. The other repetition graph can be successfully re-represented, because it does not include the second step. The operator adds an Indicating direction relation with the repetition as its first argument, and a Pointing in(`Direction`) relation as its second argument.

In the subsequent analogical matching step, the action can now match the goals on the Pointing in predicate. As we saw in Case 2, every goal orientation is a concept graph of the triangle shape, and each such graph has a Pointing in structure. Once again, the matches are of equal size, and the object content match score disambiguates by giving the orientation goal that matches the repetition's direction a better score than the goals that do not match. This allows the Receiver to conclude that the Sender is indicating he should position his triangle as pointing to the 'right'.

Failure case: separate position and orientation signaling

There are goal configurations possible in which the Sender must signal a direction without room to move in that direction from the Receiver's goal position. For example, when the goal position is the top left cell of the board, and the goal orientation is facing the top or left of the board. The Sender cannot move in those directions, and can therefore

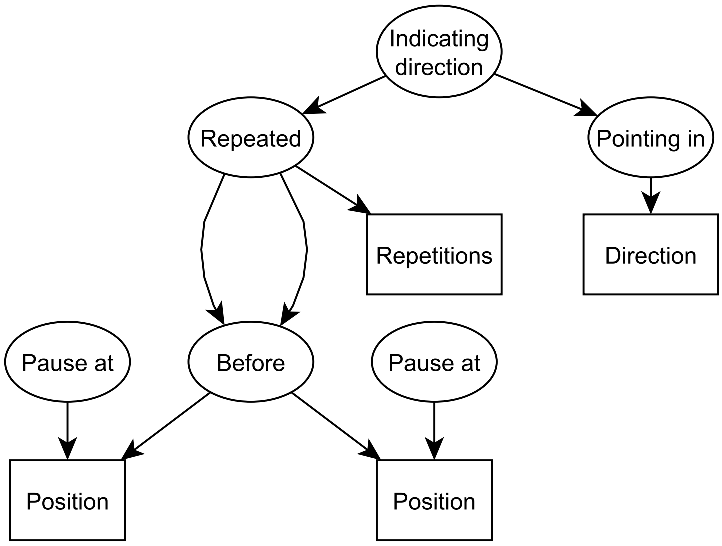


Figure 4.10: The final action representation in Case 3. After Find repetition direction has been applied, the representation contains the Pointing in substructure that can match to goal orientations.

not signal the orientation from that position. She can work around this and signal it successfully by moving to a different location to signal the orientation via a repetition, while still making it clear to the Receiver where the goal position is.

This results in a significantly more complex movement sequence compared to Case 3. The Receiver must discern between pauses to determine which pause signals the goal position, and which pauses are only indicating the start and end of the orientation signal. In Case 3, these pauses were all performed in the same location, causing no ambiguity. Additionally, the Receiver must identify the movements from the goal position to the start of the orientation signal as instrumental, which requires more detailed parsing.

In the Receiver model, the problem of discerning between pauses is accounted for by the Most significant pause operator. As long as the Sender pauses in a certain location more than in any other location, that location will be the one that is re-represented as a Place at structure. As discussed in Case 1, the operator aims to re-represent the most significant pause as the goal position, and a pause that is repeated more than other pauses is considered more significant. Therefore, the Sender can first move to a more open area on the board, perform the repetition to indicate a direction, and then move to the goal position to perform a long pause. According to the model, Receivers are able to interpret this correctly.

However, there are specific cases where the instrumental movements between the orientation signal and the goal position can be seen as part of that orientation signal, interfering with its interpretation. Figure 4.11 shows such a case. In the movement shown in Figure 4.11(b), the Sender performs an instrumental movement that overlaps with the repetition she will perform. If the Receiver considers this movement part of a communicative action, he could come to the faulty conclusion that the repetition is signaling the direction of that movement, as it can be interpreted as the first step of

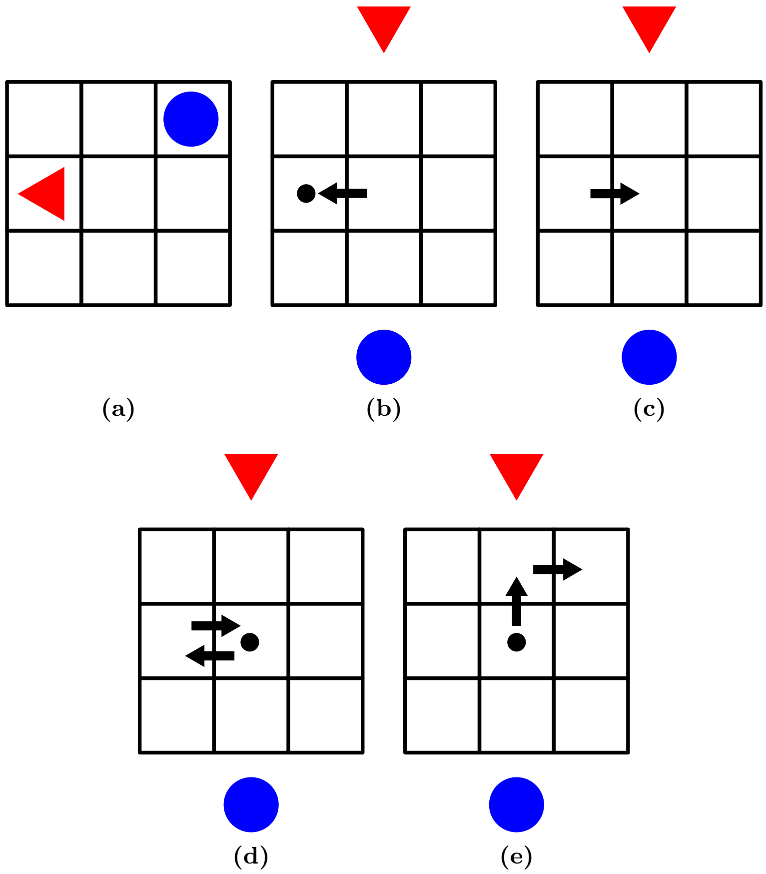


Figure 4.11: Failure case in separate position and orientation signaling. The goals are shown in (a). The Sender first performs a long pause at the goal position in (b). In (c), she steps to a position in which she can signal an orientation. Note that this is an instrumental movement. In (d), she performs the repetition, and moves to her own goal in (e).

the repetition.

That is exactly the mistake made by the Receiver model. The issue lies in the Parsing system, which is configured to assume a single communicative action. Recall that the parsing it considers best has a communicative part that includes as much of the inefficiencies in the movement sequence as possible. Hence, the communicative action covers the goal position signal, the orientation signal, *and* the movement in between.

If we configure the appropriate parameter in the Parsing system so that it assumes two communicative actions, the error is avoided: the first communicative part contains only the long pause, and the second only the repetition.

What this specific failure case shows is that the Parsing system must be able to use multiple communicative parts, in order to handle complex signals. This begs the question: how many communicative parts is sufficient for a given sequence? The number of possible parsings increases super-exponentially with the number of communicative parts. It seems implausible (and perhaps intractable) to assume the Receiver simply parses every movement sequence assuming a large number of communicative actions. Either the Parsing system must be able to detect when more communicative parts are necessary, using them only when required, or a different (and likely more complex) search approach must be used to hypothesize which parts of the movement sequence are communicative.

Failure case: rectangle Sender with a triangle Receiver

An aspect of the representation of the rectangle shape has been touched upon before: it includes the same `Direction` object that other representations use to represent a direction or orientation. In the case of the rectangle, however, this representation is insufficient: humans do not perceive a rectangle as 'pointing in' a single direction. A rectangle suggest an axis or a line that runs parallel to its long side. In context of the TCG, a rectangle can be seen as 'pointing' in two directions. A Receiver using a rectangle token will understand the Sender indicating either of those directions as meaning a single orientation of his shape. Vice versa, a Sender with a rectangle token pausing in a certain position to indicate a goal orientation sends an ambiguous signal. To the Receiver, it could be one of two possible orientations.

Due to the insufficient rectangle representation, the Receiver model fails to account for this. When interpreting a pause with a rectangle shape, the model will interpret this as a single specific orientation with no ambiguity. As a result, for a Sender with a rectangle performing the movements shown in Figure 4.7, the search process can find an orientation based on the rectangle's orientation during the first pause before it finds the orientation signaled by the repetition. For a human Receiver, the orientation of the rectangle during the pause is ambiguous, so he would not interpret it as signaling a goal when there is an unambiguous alternative in the form of the repetition signal.

4.3 Summary

For 'easy' trials, in which a Sender is able to show the Receiver's goal orientation through the orientation of her own piece, the Receiver model is able to correctly interpret the intended goal state from movement sequences that apply the predominant strategy used by human Senders on such trials. Variations in the goal state and the resulting movement sequence of the Sender do not affect performance. Only when the Sender makes a significant error in executing the strategy (such as pausing in a location she does not intend to signal) the model will in certain cases fail to find the correct goal state. Though human Receiver players might fail similarly in such cases, the Receiver model is not designed to model such errors.

For 'hard' trials, in which the Sender must signal the goal orientation via some other means, the model correctly interprets the most common strategy used by human Senders, with two exceptions. The first exception is the case where the Sender has a rectangle shape and the Receiver a triangle. The rectangle representation used by the model does not sufficiently encode the ambiguity in the orientation of the shape. As a result, the model interprets an (incorrect) orientation using a source of information that is unambiguous to the model, but ambiguous to a human Receiver, who would not use it for that reason. The second exception involves a movement sequence in which the position and orientation signals are separated by instrumental movements. For such an input, a parsing must consist of two communicative parts to reliably arrive at the correct interpretation, while the default parameters of the model specify a single communicative part.

In all cases, the object content match score is the deciding factor that disambiguates which matched goal is the goal intended by the Sender. As there is no structural difference between the representations of possible goals, structure-based measures such as SES are not useful for that purpose.

Chapter 5

Discussion

This chapter discusses the conclusions that can be drawn concerning the model and future research. First the Parsing and Meaning-mapping implementations are discussed, followed by conclusions concerning the sufficiency of analogy and re-representation. Finally, several areas of interest are identified for the development of a fully sufficient model.

5.1 Parsing

The Parsing system as implemented is sufficient for interpreting all but the most complex signals. The assumption that inefficiency in an action indicates a potential communicative signal, and the assumptions regarding the structure of the movement sequence (such as the absence of overlapping instrumental actions), allow the parsing process to remain tractable even for long movement sequences. However, complex multi-part signals create difficulties, due to the algorithm not being capable of scaling up the number of assumed communicative parts when the movement sequence does contain them. This causes errors in the generated parsings, as instrumental movements are hypothesized to be communicative. Such movements can function as 'red herrings' for the Meaning-mapping system, resulting in interpretation errors one would not expect a human player to make.

This a complex issue, as generating parsings with more hypothetical communicative parts is undesirable, because the number of possible parsings rises quickly with the number of communicative parts. Exhaustively generating and testing every possible parsing would become intractable for longer movement sequences. A more advanced approach that dynamically increases the number of communicative subsequences based on certain criteria seems more feasible, if potentially difficult to design.

5.2 Meaning-mapping

Similar to the Parsing system, the Meaning-mapping system is capable of interpreting communicative actions resulting from common strategies, save certain notable exceptions. In successful cases, the re-representation algorithm performs the reasoning steps required to find an action representation that can be matched to the correct goal using structure-mapping.

5.2.1 Re-representation

The re-representation process is limited by the available operators, and for this research the set of operators was limited to those required to interpret the most common strategies. The model is unable to account for other strategies that need different representations to succeed, as it simply lacks the ability to perform the required reasoning steps. Adding additional operators is therefore an obvious area for future work.

Alternatively, the existing operators may be abstracted to more general forms. A set of highly general re-representation operators could provide representations allowing analogical matches for a much greater set of signals than the current operators. However, designing such operators is not a trivial task. It could be much helped by research on the representations and reasoning used by human players.

Strategies for which the model has sufficient operators were shown to have certain failure cases where the error occurs in the Meaning-mapping system. The issue lies in the insufficient representation of the orientation of a rectangle shape. For human observers, a rectangle does not normally ‘point’ in a single, unambiguous direction. Nevertheless, the representation did assume this was the case by employing the same representational structure as the triangle shape concept graph. This allowed for simpler matching of rectangle and triangle graphs, but also results in the model interpreting certain signals in ways human Receivers can not. The concept of ambiguity must be introduced to the rectangle’s orientation representation to rectify this. One potential method is the introduction of an `Ambiguous direction` object that can match the content of a `Direction` object in one angle, but also its opposite, guaranteeing an ambiguity of two equally valid matches when trying to match a rectangle representation to a triangle.

The implementation of re-representation in the Receiver model is a significant contribution to re-representation research, as very few implementations exist. The implementation by Yan et al. (2003) is perhaps the best known. Yan et al. propose an approach aimed at analogy construction in the general case, integrated with SME. Much like the research discussed here, they have applied it on a relatively limited scale. Hence, extending or enhancing the set of available operators and representations in the Receiver model is also an opportunity to further the state of the art in re-representation research.

One outstanding question of particular interest is whether re-representation is computationally tractable in a real-world setting. TCG provides an excellent opportunity for research in this area due to its restricted scope. An additional topic is whether the tractability of re-representation differs between the approach used in the Receiver model discussed here, and the method used by Yan et al.. Similarly, it is not known what effect a set of more generalized operators would have on the tractability of the Receiver model.

5.2.2 Structure-mapping

The Receiver model shows that structure-mapping can be successfully applied to the type of low-level analogies used in TCG strategies (compared to analogies involving more abstract concepts, often used as examples in SMT literature). The content of entities that participate in matched relations plays a significant role, however. The structures involved are smaller and simpler than those typically used in SMT research, and there are often multiple matches of identical structure that differ only on content. The amount of matching content is therefore important in discerning between matches to select the match intended by the Sender.

It is possible that more complex TCG strategies involve more high-level relations than those seen in this research, but the low-level nature of TCG, in which the players are dealing with specific positions and orientations rather than abstract relations, guarantees content will always play an important role. Only by moving content into structure – that is, representing object content such as positioning information as predicate relations – can SMT independently find an unambiguous structural match on a specific goal position or orientation. Of course, this runs counter to the fundamental SMT concept of matching on structure rather than content.

5.3 Analogy and re-representation

This research has shown that communicative signals commonly used by human TCG players can be interpreted using the mechanisms of a leading theory of analogy, combined with the concept of re-representation. This result suggests that the cognitive abilities of analogy and re-representation are sufficient to explain the intention recognition abilities of human Receivers, at least for the signals the model can interpret.

However, one could point out some issues. The re-representation operators used here were fairly specific and 'tailored' to the TCG task. They have shown to be sufficient for common signals, but a set of more general operators would provide a more robust validation.

With regards to analogy, one could argue that after the parsing phase, interpreting certain signals (such as a pause at a location) is simply a case of detecting the literal similarity between the action representation and the matching goal (as they represent one and the same location). This is however not an argument against the use of analogy, as such similarity is effectively a less selective form of analogy (Gentner & Colhoun, 2008), and SMT has been shown to extend to literal similarity (Gentner & Markman, 1997).

All in all, the results described in this thesis offer no definite reason why analogy and re-representation would be insufficient for the full range of TCG strategies. Instead, the capabilities of the model show that the approach could quite plausibly account for all signals, given further research and development of all systems that make up the Receiver model. Regardless of issues that exist in this early attempt, it is a promising first step towards a fully sufficient model.

5.4 Towards a fully sufficient model

Based on the current limitations of the model, several areas can be identified that offer opportunities for future work on a TCG model.

5.4.1 Parsing algorithm

As discussed, for the parsing process to be sufficient for explaining all strategies, a method for escalating the number of hypothesized communicative parts is necessary to correctly model the interpretation of complex strategies. Challenges lie in retaining the tractability and optimality of the parsing process.

5.4.2 Re-representation

More re-representation operators should be added to account for more signal types. When designing additional operators, opportunities may present themselves for generalizing operators. Compared to the re-representation implementation by Yan et al. (2003), the current set of operators performs more specialized transformations of graphs, rather than generic transformations on individual predicates. The approach of Yan et al. may allow the Receiver model to generalize better. On the other hand, it is possible that additional operators that combine with the existing set will already facilitate generalization. The current operators may also be generalizable without modifying the re-representation algorithm. The relative computational complexity of the approach by Yan et al. versus the approach used here could inform a decision on this, but analysis is required to establish the complexity of the various approaches first.

Data on human re-representation strategies and/or reasoning steps performed by players when interpreting a TCG movement sequence would be valuable in designing operators. However, such data seems very difficult to gather, outside of potentially unreliable introspection of human players.

5.4.3 Learning

An element that is currently missing from the Receiver model implementation is a mechanism by which the history of previous mappings of actions to goals influences new cases. The proposed architecture on which the current implementation is based leaves this largely undefined (see Figure C.2, p. 67), but one possible area in which learning most likely plays a role is re-representation. If a certain re-representation strategy is successful, human Receivers would most likely attempt to use that strategy again when there is an opportunity to do so. After all, if the Sender knows a certain signal was successfully interpreted, she is more likely to use that strategy again compared to other strategies that might not be understood. If the Receiver can exploit this by attempting to use the successful re-representation strategy earlier in the search process, he can avoid an extensive search that has a chance of finding an incorrect interpretation.

This could be implemented by allowing the model to track whether a certain series of operator applications is often successful, and combining these operators into a single, new re-representation operator. This learned operator can then be applied like any other, applying the transformations of the original operators in sequence. Its position is higher up in the re-representation tree, at the level of the first operator in the original sequence, allows it to reach the resulting re-representation before the normal search would find it. This allows the Receiver to find a good match earlier in the re-representation process. Figure 5.1 illustrates how a re-representation path would be combined resulting in a new operator in the re-representation tree.

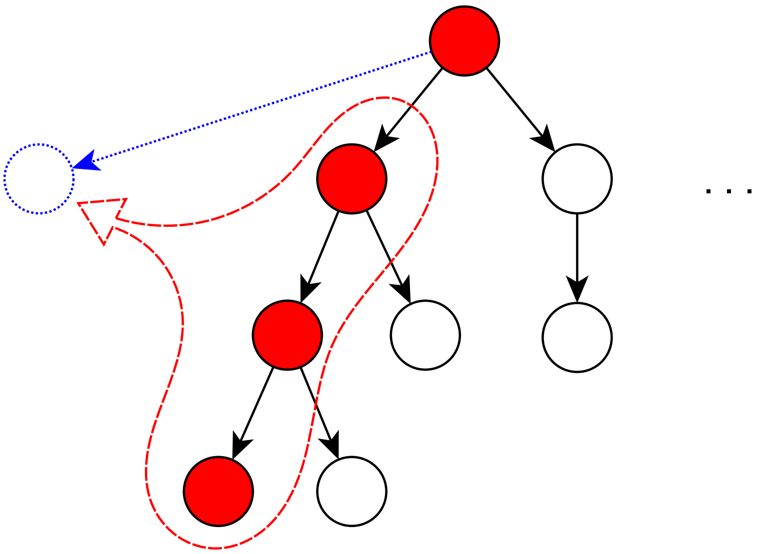


Figure 5.1: Learning re-representation strategies. A successful path (indicated by filled red circles) through part of the re-representation tree can be identified and folded into a single operator (dotted blue circle). This operator becomes a part of the tree, positioned at the level of the first operator of the original path. Note that the original operators remain unchanged, as they may be necessary for other re-representation paths.

5.4.4 Errors and reaction times

To fully model human TCG-playing behavior, errors human players make should also be made by the model. The implemented Receiver model does not attempt to do this yet. In experiments with human players, even trials in which both players have the same shape failed 10% of the time, while different-shape trials had a failure rate of 24% (de Ruiter et al., 2010). Clearly errors by both the Sender and Receiver play an important role in human performance. Before this aspect can be modeled, however, more investigation is necessary on the nature of the mistakes that are made, and where in sending or interpretation process they can be hypothesized to occur.

Besides errors, a complete model should also be able to predict (relative) reaction times. These have been found to differ significantly between ‘easy’ and ‘hard’ trials, and between Sender and Receiver (de Ruiter et al., 2010). Based on the model, factors such as re-representation depth or parsing complexity would be hypothesized to be a source of delay.

5.5 Summary

The model described in this thesis shows how core systems involved in human TCG-playing can be modeled. The results have shown that analogy and re-representation can be sufficient for the successful interpretation of signals. Though limited to relatively simple (but common) strategies, a broad range of opportunities for future research and modeling in this area can be identified based on where this model succeeds, and where it fails. The road from this first attempt to a fully sufficient model is long, but it nevertheless presents a unique opportunity to gain a deeper understanding of human interactional intelligence.

Appendix A

Software implementation

I implemented the model described in this thesis from scratch for this research. It is written in the Clojure (Hickey, 2008) programming language. In addition to parsing and re-representation subsystems (relatively) specific to the model, it includes a full (re-)implementation of the analogy construction algorithms of the Structure-Mapping Engine. As a whole, the implementation represents several months of intensive development.

Besides the model itself, a simple GUI front-end was developed. It allows one to easily create movement sequences as input for the model, as well as viewing the resulting output. The displayed output consists of the interpreted goals and data on the analogical match, as well as the parsing and the re-representation path leading up to that match, including basic visualizations of the concept graphs involved.

The model and its interface may be of interest to anyone looking to verify the results found in this thesis, or to experiment with (modifications of) the model in general. In addition, to my knowledge no publicly available, open-source implementations exist outside of the original SME written in Common Lisp. Hence, the new implementation in the model might be of some general interest.

The software has been released under a liberal, open-source license. At the time of this writing, it can be found at http://svandermeer.ruhosting.nl/. Should this no longer be the case at some future date, the software is available upon request.

Appendix B

Figures of signals

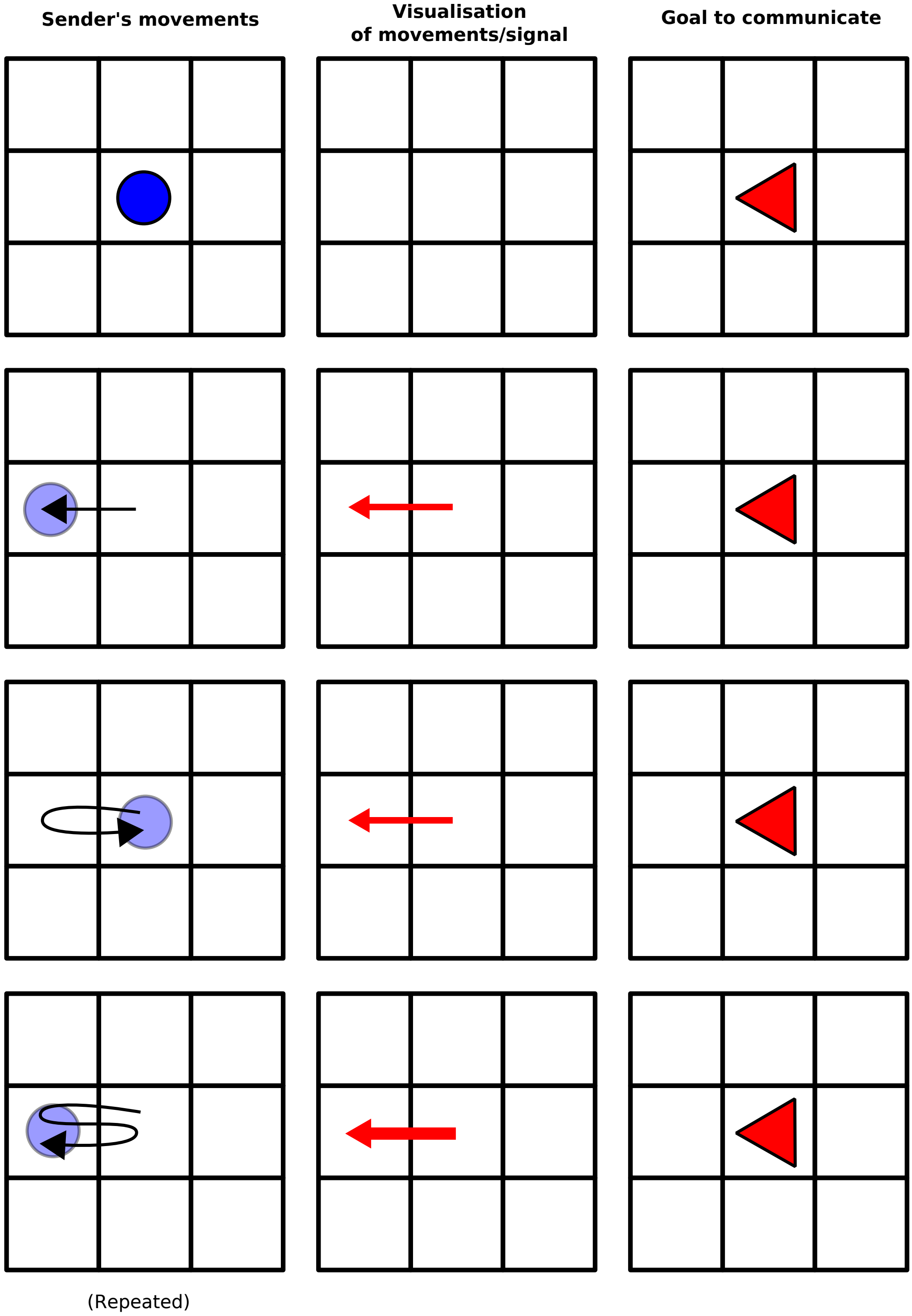


Figure B.1: Example of an aspect-based signal. The Sender's movements intend to signal only one specific aspect of the Receiver's goal, namely the direction his token should 'point in'.

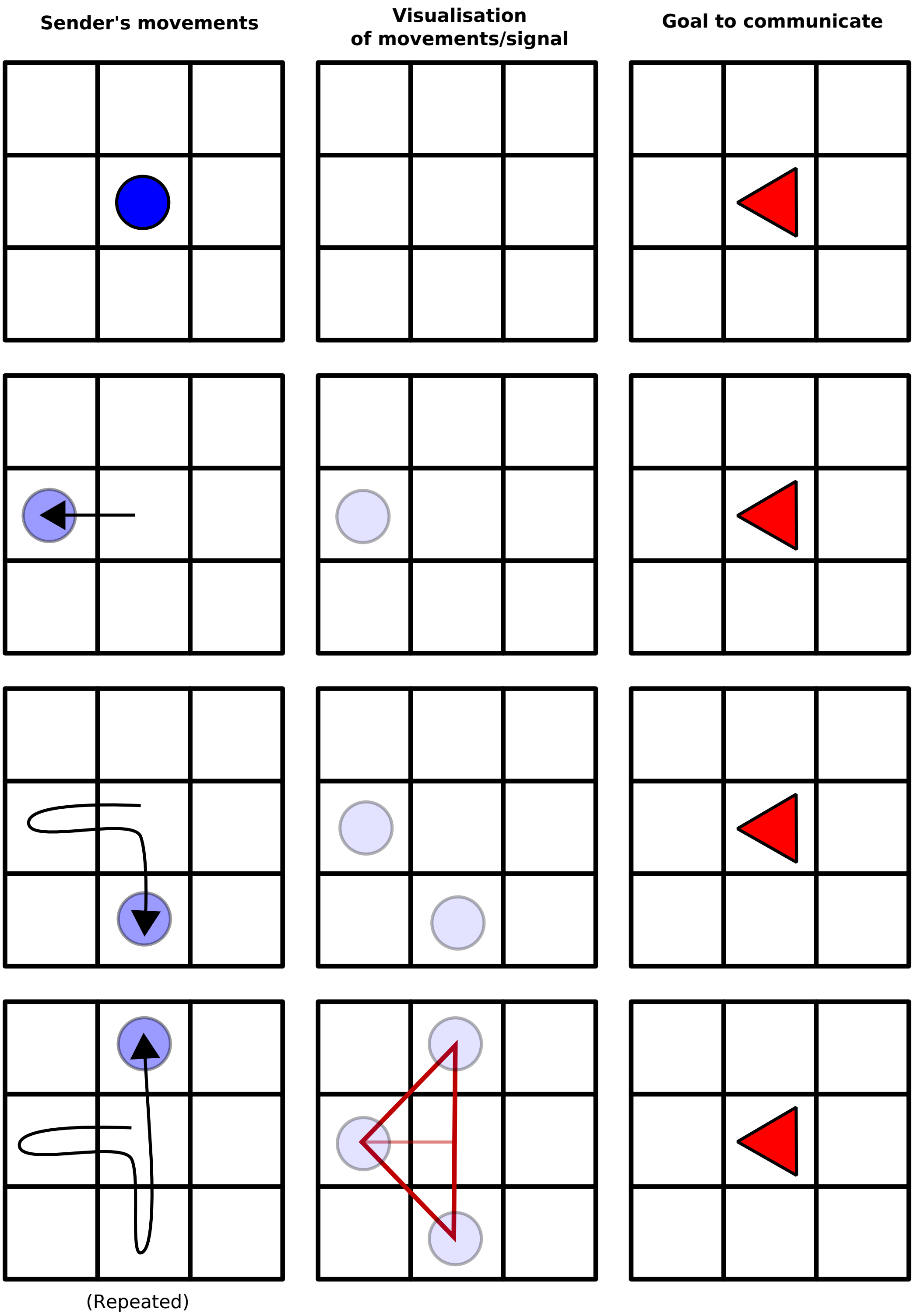


Figure B.2: Example of a shape-based/iconic signal, in which the Sender signals the shape the Receiver's token has when oriented in accordance with his goal state.

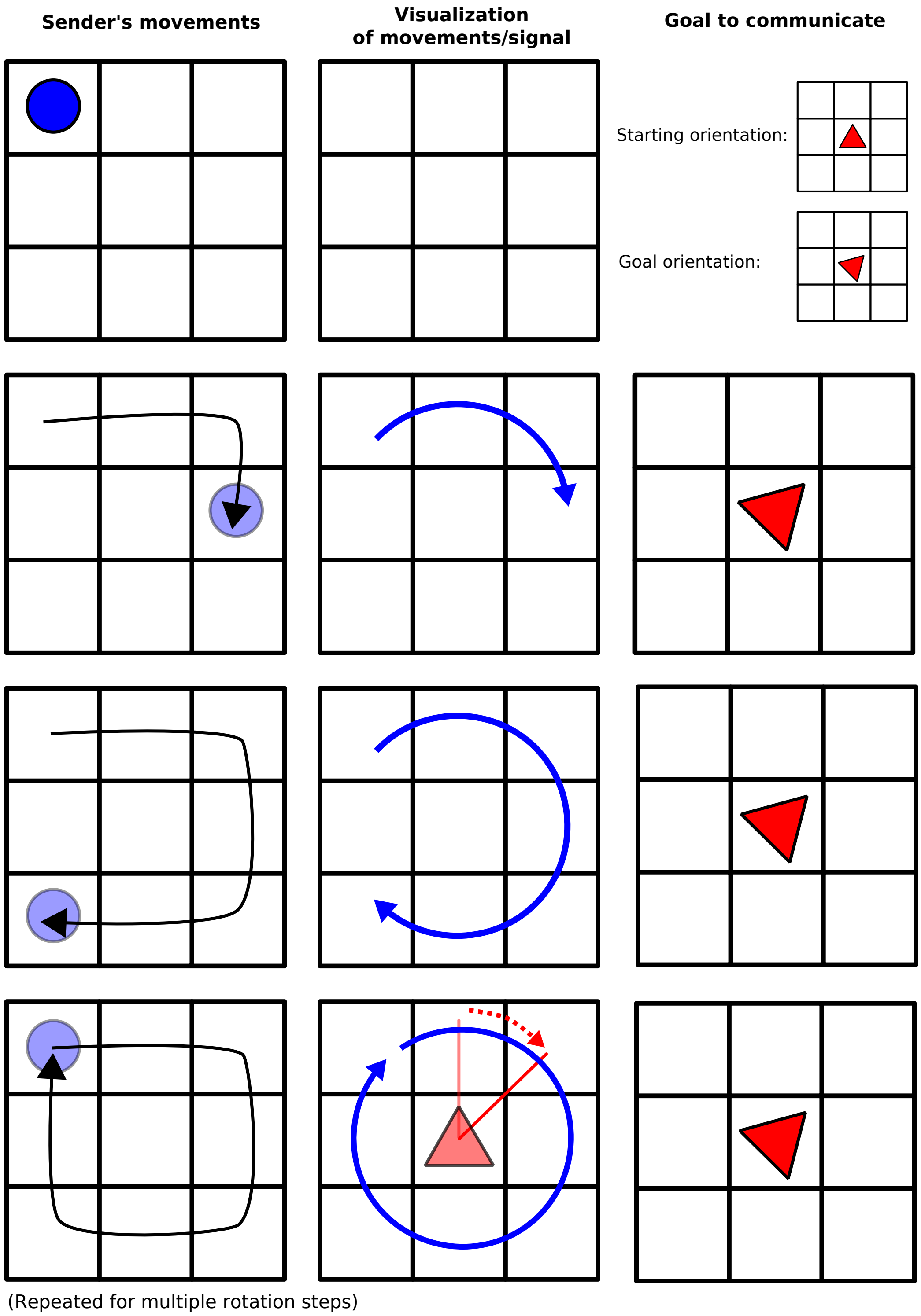


Figure B.3: Example of an act-based signal. The Sender's token cannot show rotations. Despite this, she intends to signal how many times the Receiver should perform a rotation act to reach his goal orientation. The Sender can attempt to circumvent this by navigating in a circular path around the board to indicate a rotation step.

Appendix C

Figures of model architecture

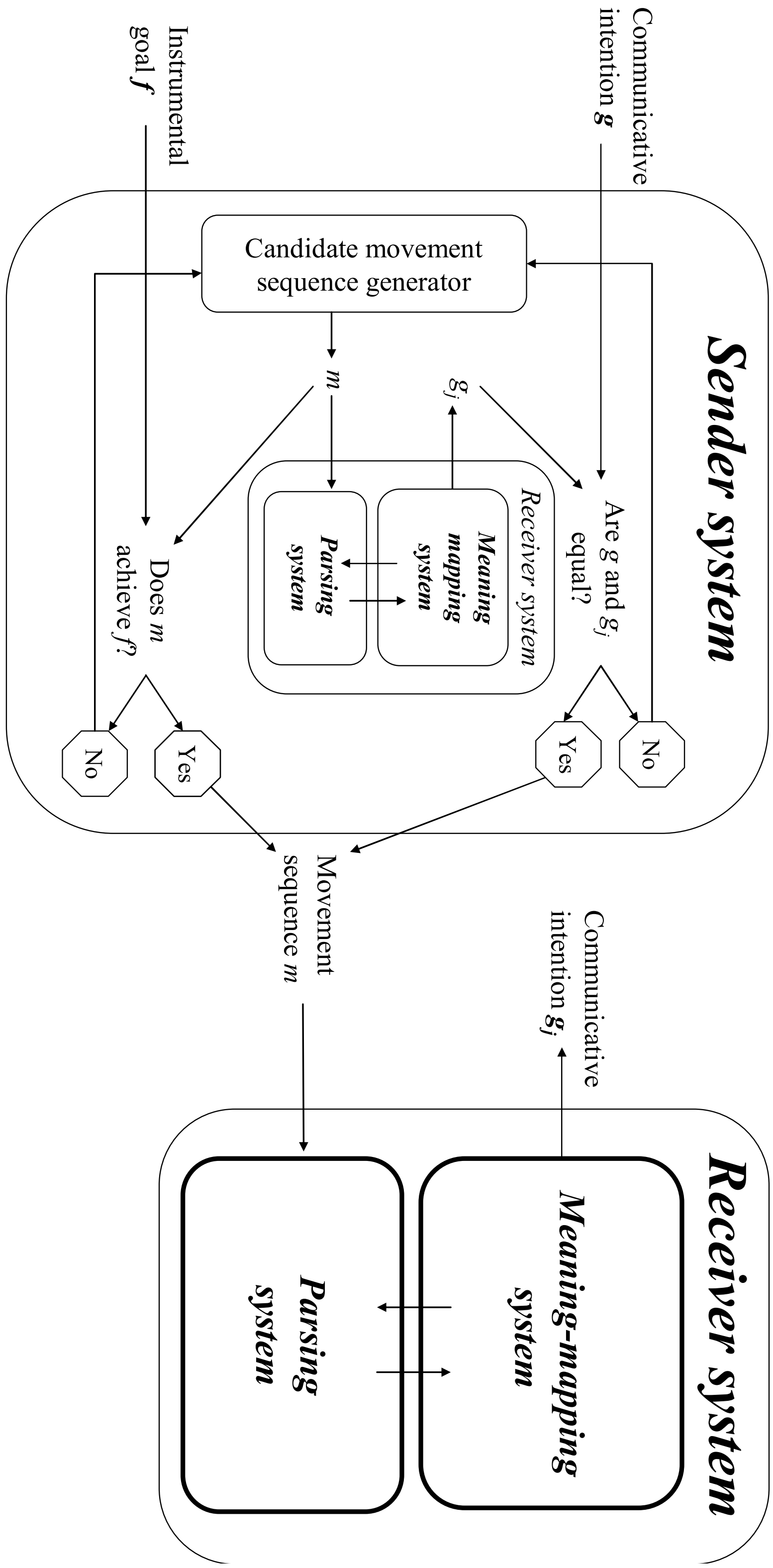


Figure C.1: Architecture of Sender and Receiver models as described and illustrated by van Rooij et al. (2009). The Sender uses its internal Receiver system for recipient design.

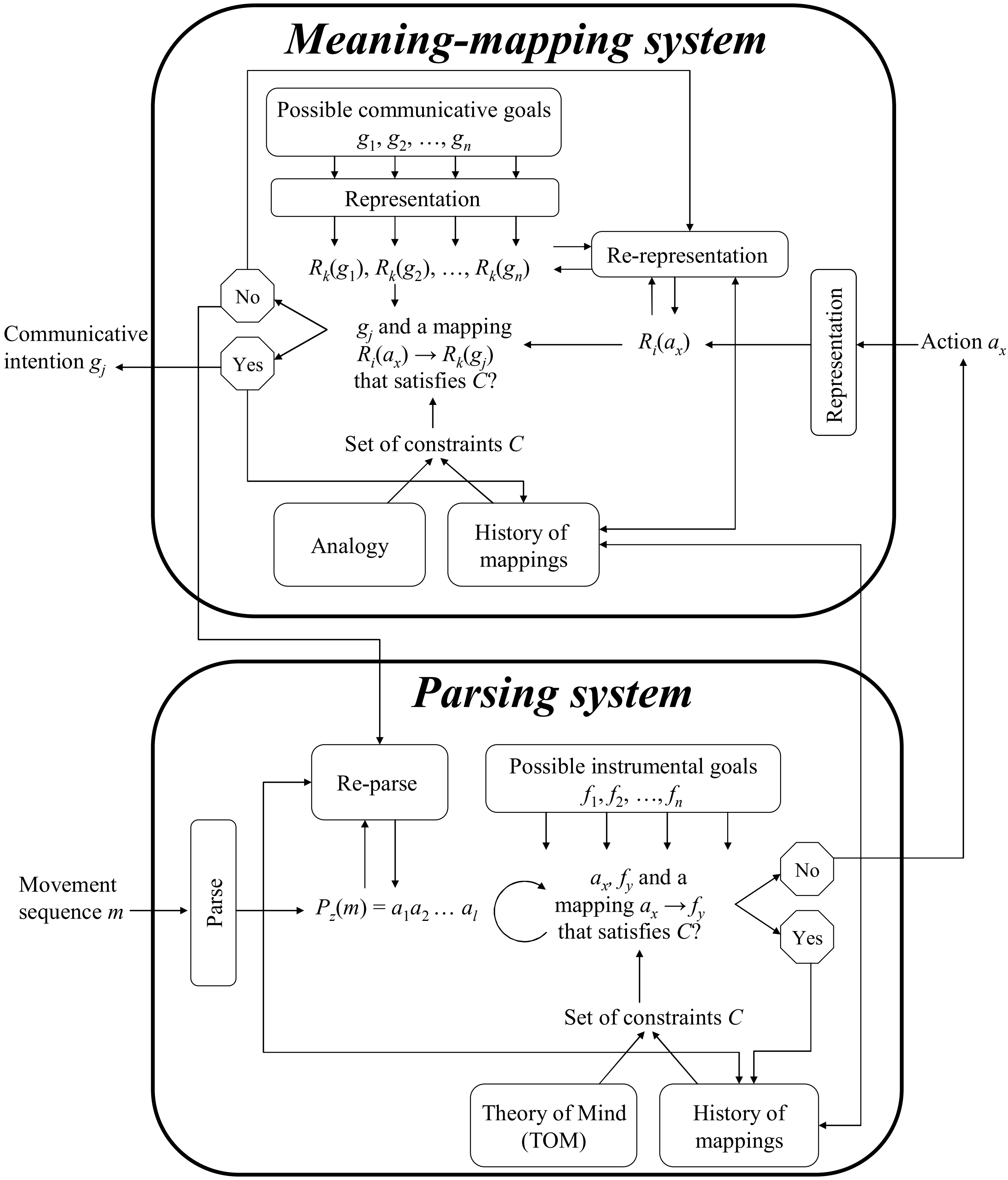


Figure C.2: Detailed view of the Receiver system proposed by van Rooij et al. (2009). Larger version of Figure 2.1.

Bibliography

Attneave, F. (1968). Triangles as ambiguous figures. *The American Journal of Psychology*, *81*(3), 447–453.

Bucher, N. & Palmer, S. (1985). Effects of motion on perceived pointing of ambiguous triangles. *Perception & Psychophysics*, *38*(3), 227–236.

de Ruiter, J., Noordzij, M., Newman-Norlund, S., Hagoort, P., & Toni, I. (2007). On the origin of intentions. In P. Haggard, Y. Rosetti, & M. Kawato (Eds.), *Attention and Performance XXII: Sensorimotor foundation of higher cognition* (pp. 593–610). Oxford: Oxford University Press.

de Ruiter, J., Noordzij, M., Newman-Norlund, S., Newman-Norlund, R., Hagoort, P., Levinson, S., & Toni, I. (2010). Exploring the cognitive infrastructure of communication. *Interaction Studies*, *11*(1), 51–77.

Falkenhainer, B., Forbus, K., & Gentner, D. (1989). The structure-mapping engine: algorithm and examples. *Artificial Intelligence*, *41*, 1–62.

Ferguson, R. (1994). MAGI: Analogy-based encoding using regularity and symmetry. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, (pp. 283–288)., Hillsdale, NJ. Erlbaum.

Forbus, K. & Gentner, D. (1989). Structural evaluation of analogies: What counts? In *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society*, (pp. 341–348)., Hillsdale, NJ. Erlbaum.

Forbus, K., Gentner, D., & Law, K. (1995). MAC/FAC: A model of similarity-based retrieval. *Cognitive Science*, *19*(2), 141–205.

French, R. (2002). The computational modeling of analogy-making. *Trends in Cognitive Sciences*, *6*(5), 200–205.

Friedman, S., Taylor, J., & Forbus, K. (2009). Learning naive physics models by analogical generalization. In *Proceedings of the 2nd International Analogy Conference*, (pp. 145–154)., Sofia, Bulgaria.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, *7*, 155–170.

Gentner, D. (2003). Why we're so smart. In D. Gentner & S. Goldin-Meadow (Eds.), *Language in mind: Advances in the study of language and thought* (pp. 195–236). Cambridge, MA: MIT Press.

Gentner, D. & Colhoun, J. (2008). Analogical processes in human thinking and learning. In A. von Müller, E. P. (series), B. Glatzeder, V. Goel, & A. von Müller (volume) (Eds.), *On thinking: Vol. 2. Towards a theory of thinking.* Berlin Heidelberg: Springer-Verlag.

Gentner, D. & Markman, A. (1997). Structure mapping in analogy and similarity. *American Psychologist*, *52*(1), 45–56.

Hickey, R. (2008). The Clojure programming language. In *Proceedings of the 2008 symposium on Dynamic languages*. ACM New York, NY. See also http://www.clojure.org/.

Levinson, S. (2006). On the human interactional engine. In N. Enfield & S. Levinson (Eds.), *Roots of Human Sociality* (pp. 39–69). Berg Publisher.

Noordzij, M., Newman-Norlund, S., de Ruiter, J., Hagoort, P., Levinson, S., & Toni, I. (2009). Brain mechanisms underlying human communication. *Frontiers in Human Neuroscience*, *3*(14), 1–13.

Palmer, S. (1980). What makes triangles point: local and global effects in configurations of ambiguous triangles. *Cognitive Psychology*, *12*, 285–305.

Palmer, S. & Bucher, N. (1981). Configural effects in perceived pointing of ambiguous triangles. *Journal of Experimental Psychology: Human Perception and Performance*, *7*(1), 88–114.

Palmer, S. & Bucher, N. (1982). Textural effects in perceived pointing of ambiguous triangles. *Journal of Experimental Psychology: Human Perception and Performance*, *8*(5), 693–708.

Premack, D. & Woodruff, G. (1987). Does the chimpanzee have a theory of mind? *The Behavioral and Brain Sciences*, *4*, 515–526.

Shannon, C. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, *27*, 379–423, 623–656.

van Rooij, I., Toni, I., & Haselager, W. (2009). How movements mean. Unpublished manuscript.

Yan, J., Forbus, K., & Gentner, D. (2003). A theory of rerepresentation in analogical matching. In *Proceedings of the Twenty-fifth Annual Meeting of the Cognitive Science Society*, (pp. 1–7).