STORY-ENABLED HYPOTHETICAL REASONING

by

Dylan Alexander Holmes

B.S., Wichita State University (2012)

Submitted to the
Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering and Computer Science
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2017

Signature of Author .....
Department of Electrical Engineering and Computer Science
May 19, 2017

Certified by .....
Patrick H. Winston
Ford Professor of Artificial Intelligence and Computer Science
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

STORY-ENABLED HYPOTHETICAL REASONING

by
Dylan Alexander Holmes

Submitted to the Department of Electrical Engineering and Computer Science
on May 19, 2017, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

Story understanding is a central competence that illuminates all other aspects of human intelligence. In this work, I demonstrate how our story understanding ability sheds light on our ability to think in terms of hypothetical situations. Using the Genesis story understanding system as a substrate, I develop a story-enabled hypothetical reasoning system that models several high-level human abilities, including judging actions in terms of moral alternatives, contextualizing stories by considering what could have otherwise happened, and deliberating about personality to decide what characters will do next. In developing this system, I built many new computational mechanisms and representations, including a program for answering what-if questions, a side-by-side story comparator, rules for making presumptive inferences, heuristics for evaluating personality fit, and a problem-solving approach for evaluating moral character. Together, they take Genesis's story understanding capabilities to another level and advance our understanding of human intelligence.

Thesis Supervisor: Patrick H. Winston
Title: Ford Professor of Artificial Intelligence and Computer Science

Acknowledgements

First, I express my gratitude and admiration for the friends whose lives have intersected with mine at MIT.

In this particular endeavor, I am especially grateful to Adam and Jessica for providing incredible writing advice.

Thanks as always to Robert and to my family, who have been a part of this adventure since the very beginning.

And thanks to Patrick Winston—for truly heroic mentorship, for leading by example, for compassion and inspiration, and for setting all our sights on the big picture. I am honored to be part of your team.

Financial sponsorship for the work in this thesis was supplied in part by the U.S. Air Force Office of Scientific Research under contract FA9550-17-1-0081.

This material is also based upon work supported in part by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216.

Contents

List of Figures

The main ideas

Our intelligence springs from our facility with possibilities, impossibilities, and constraints. When you go for a walk in the woods, you perceive countless *possibilities* for action: sticks and how you might grasp them, stepping stones and how you might navigate them, and things stacked on and covering each other and how you might reveal them (Gibson, 1986). Or if you observe a child over the first two years of life, you will see that child become acquainted with fundamental categories of things and the *impossibilities* associated with them: solid objects cannot pass through one another, inanimate objects do not move on their own, and so on—impossibilities that are key to their understanding of the world (Spelke and Kinzler, 2007; Sloman, 2015). Finally, in everyday life, you can improvise reasonable solutions to unfamiliar problems; for example, if asked how candy canes get their stripes, you might suggest a paintbrushing mechanism, but would probably rule out a spraying mechanism because it couldn't produce sharp-edged lines—your knowledge of the *constraints* and requirements of the problem help guide your search toward solutions that make sense (Magid et al., 2015) and often help you avoid even *thinking* of bad solutions (Minsky, 1994).

If we are to understand how we perform feats like these, we must understand how we think in terms of possibilities, impossibilities, and constraints—what I call *hypothetical reasoning*. In this thesis, I have taken steps toward understanding hypothetical reasoning through its connection to our ability to understand stories—where the term *stories* can broadly include fables, legal arguments, descriptions of physical processes, and more. In particular, I propose that many kinds of human-level hypothetical reasoning can be grounded in our ability to think in terms of stories, and moreover that we understand stories more deeply because we can consider hypothetical alternatives. To develop these ideas concretely, I implemented several computational hypothetical reasoning capabilities using the using the Genesis Story Understanding System as a foundation[1].

[1] See Section 2 for more background on Genesis.

In particular, I—

1. Introduced **presumption rules** to encode fragile default assumptions about potential causes and effects. These rules extend Genesis's existing rule types with a form of knowledge that is normally latent until the system is asked a question.
2. Developed a **what-if question-answering system** which uses presumption rules to determine what would change if an element of the story were removed. For example, in a courtroom scenario, "What if the assailant didn't brandish a knife?"
3. Developed a **side-by-side comparator** to automatically summarize the differences between the original and a what-if variant story at different levels of granularity.
4. Introduced **means-ends rules** so as to extend the what-if question answering system to reason about goals, alternative methods, and morality.
5. Developed a novel **computational model for judgments of character** using all of the preceding apparatus—what-if questions, presumption rules, side-by-side comparisons, and means-ends rules—as a foundation.

Together, these computational capabilities constitute a powerful story-enabled hypothetical reasoning system which allows Genesis to understand new kinds of stories and which takes Genesis's existing story understanding capabilities to another level. With this hypothetical reasoning system, Genesis can now:

- Describe how a legal defense hinges on whether the assailant brandished a knife.
- Judge a character's moral actions in terms of alternative strategies the character could have used—but didn't.
- Predict a character's actions based on whether the character makes choices that evoke personality types such as Conformist, Thief, Robin Hood, or Opportunist.

By shedding light on how varieties of hypothetical reasoning can be usefully grounded in story understanding, I

pave the way for understanding the many other varieties of hypothetical reasoning. From a scientific perspective, these new hypothetical reasoning capabilities help us to model what makes humans so intelligent. From an engineering perspective, they demonstrate the powerful prospect of systems equipped to reason about possibilities, impossibilities, and constraints—inventing solutions that conform to specifications, explaining their decisions, and anticipating problems before they occur.

A program for answering what-if questions

Chapter Summary

In this chapter, I describe the three ingredients of a what-if question answering system: a framework of knowledge for encoding default assumptions and latent possibilities, a program for answering "What would happen if the story were different?" questions by removing elements of the story and filling in gaps using presumption rules, and an automatic comparator for summarizing how the stories diverge—identifying the differences that make a difference.

For example, the program reads a courtroom-inspired story about a break in and initially decides that the victim's retaliation counts as self-defense. When the user asks how the self-defense verdict depends on the assailant brandishing a knife, the program generates and considers a variant story in which the knife is unmentioned. By comparing its analysis of the original and variant stories, the program concludes that *it believes* that the knife is essential to the self-defense verdict, because it views self-defense as unjustified in the second story (Figure 1).

By considering the effect of these variations, this program can consider how the story interpretation would be different if events in the story were different, if the reader's knowledge and conceptual framework were different, or even if the reader's cultural background were different, paving the way for applications as diverse as planning, moral reasoning, and conflict resolution.

With this program, my aim is to model how humans understand more about stories than what is explicitly stated. We are always supplying background when we read, from commonsense assumptions about how the world works, to context about what could have happened or what might happen next. Consider how our ability to feel suspense, surprise, or poignancy depend on our ability to conceive of such variations ("What is this character going to find in the basement?", "I did not expect the explorer to be able to jump across the chasm in

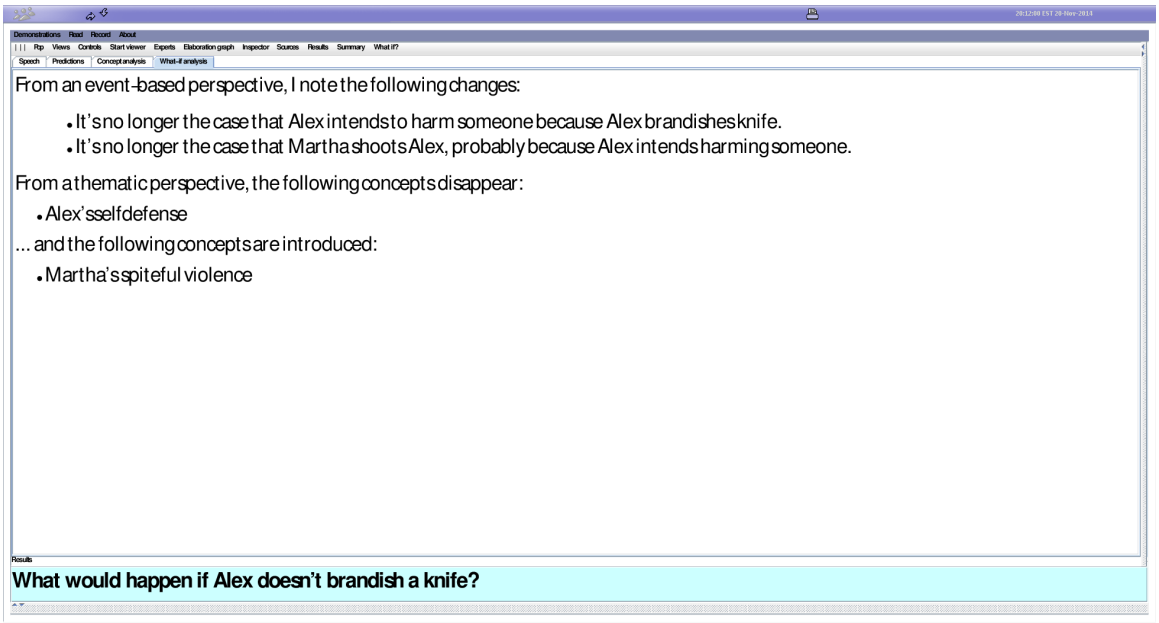

Figure 1: Having read a story about self-defense, the program developed in this chapter answers the user's question "What would happen if Alex didn't brandish a knife?". Using a suite of latent background knowledge and an automatic side-by-side comparison routine, it produces a distilled summary of the expected differences between the original and variant stories, shown here.

time!", "If only Romeo had learned Juliet's death was a ruse!"). Thus, from a scientific perspective, this program is valuable because it provides a model of a skill that humans use to read capably. From an engineering perspective, the discipline of considering hypothetical variations helps to develop representations and processes that are robust to change.

1 The ramifications of removed story elements

I was initially inspired by the idea of a program that can analyze a story using hypothetical reasoning, comparing the story against variations in what could otherwise have plausibly happened or what might happen next. Such a program, for example, would be able to register its expectations and highlight live alternatives (that is, alternatives that could reasonably happen, requiring data and processes for figuring out *plausible* outcomes) and would consequently be equipped to ex-

perience reader reactions such as surprise, suspense, anticipation, poignancy, disappointment, and so on. Furthermore, such a program would have the groundwork for imagining how circumstances might be different, building a foundation for skills such as making plans and anticipating bad side effects, or tailoring what to say based on models of how different listeners might react.

For the scope of this project, I chose to develop a subset of these capabilities. Rather than requiring the program to invent its own plausible variations—which could require, among other things, a capacity to locate pivotal events in the story, a store of knowledge about how to vary them, and the ability to judge alternatives and manage its attention—I elected to have the *user* tell the program what part of the story to focus on. Second, I elected to consider variations of the specific form "How would the story change if this element of the story were removed?" rather than arbitrary hypothetical questions such as "What would have happened if this character had triumphed in the second act?", "How would the container's effectiveness change if we used a different polymer?", "What would change in *Macbeth* if Lady Macbeth were less ambitious?"—questions which rely on a broad and heterogeneous collection of complex knowledge, skills, and creative processes. Even with these simplifications, however, this program represents an important step toward the overall goal of developing a hypothetical reasoning program.

At a high level, the program works as follows. First, this program reads a short story in simple English, typically consisting of around twenty to thirty lines. The program uses Genesis's existing story understanding capabilities to draw inferences, detect thematic concepts in the story (such as Self-defense or Pyrrhic victory), and perform other analyses. Then, when the user directs the program to remove an element of the story through a question such as "What would happen if the assailant didn't brandish a knife?", the program removes the element and evaluates the modified story. Importantly, because I have supplied the program with detailed commonsense knowledge, the revised evaluation can be quite different, as the program will make guesses about how to fill in gaps in the story. The program is then equipped with procedures for

comparing the before and after analyses; it describes these differences and thereby reflects on which story differences make a difference.

Previously, Genesis could only analyze a story as written, i.e., the explicit events in the story as augmented by commonsense knowledge. This program takes Genesis's story understanding capability to another level by allowing Genesis to judge a story in light of its *possible variations*. The essential idea is that our understanding of what has happened in the story is sharpened by what doesn't happen, what could happen, and what hasn't happened yet.

2 The Genesis story-understanding substrate

Before delving into what's new, here I briefly review the components of the Genesis story-understanding system upon which this thesis is built. For a more detailed background of the motivations for and capabilities of the Genesis system, I refer the reader to our foundational documents (Winston, 2011, 2012a,b; Winston and Holmes, 2017).

The **Genesis story-understanding system**, developed by Patrick Winston's research group at MIT, is a computational architecture which models how humans understand and tell stories. The overall vision of the group is that human intelligence is uniquely distinguished from the intelligence of other species by our ability to use and manipulate deeply nested symbolic descriptions—stories, broadly construed—and that if we are to understand and model human intelligence, we must understand and model the mechanisms that enable these story understanding capabilities.

In a typical use case, Genesis reads a text file of about twenty to thirty lines containing a story in simple English. After the story is processed, it is shunted to a variety of different agents—an arrangement inspired by propagation networks (Radul, 2009). These agents are specialized for a variety of intelligent tasks such as identifying questions, representing movement through space, accumulating knowledge, forming models of what characters know, judging tone, and forming a self model, among many others. The agents dispatch on the incoming sentences, construct their own internal represen-

tations, and pass messages to one another throughout this cognitive system so as to assemble a detailed comprehension of the story along many different dimensions.

One of the more fundamental representations used by Genesis is the **elaboration graph** (Figure 2). The elaboration graph is a structured representation of the elements of the story as a directed graph, with arrows indicating causal connections or inferential connections. (Causal connections include, for example, a problem precipitating a character's response; inferential connections include, for example, the conclusion that if a character is in the kitchen of a house, the character is consequently also in the house.)

Such commonsense connections are essential to understanding the story in a humanlike way, but are hardly ever expressed explicitly in the stories people encounter. Hence, we ourselves furnish Genesis with the necessary kind of background knowledge that even young children know. As a result of this general commonsense knowledge, provided through an auxiliary text file similarly expressed in simple English, Genesis can discover many more causal and inferential connections than are expressed explicitly in the story, providing a richly connected graph.

There are two major structures for representing this commonsense information. The first is a family of different **commonsense rules**. Genesis possesses an arsenal of rule types each with a specialized behavior developed to meet a particular engineering need. Genesis possesses deduction rules, abduction rules, explanation rules, and unknowable-leads-to rules, among others. Instantiated rules are matched against the story, and when they fire, they may add new information to the story (such as deductive consequences or more detailed information about how an action could be carried out) or new connections (such as causal connections). These new story elements and connections are accumulated in the elaboration graph.

The second structure comprises **narrative concept patterns**. In the Genesis system, a concept pattern is a constellation of events in a story which together represent a high-level narrative theme such as *Success through adversity* or *Escalating violence* (See (Lehnert, 1981) for related work on plot

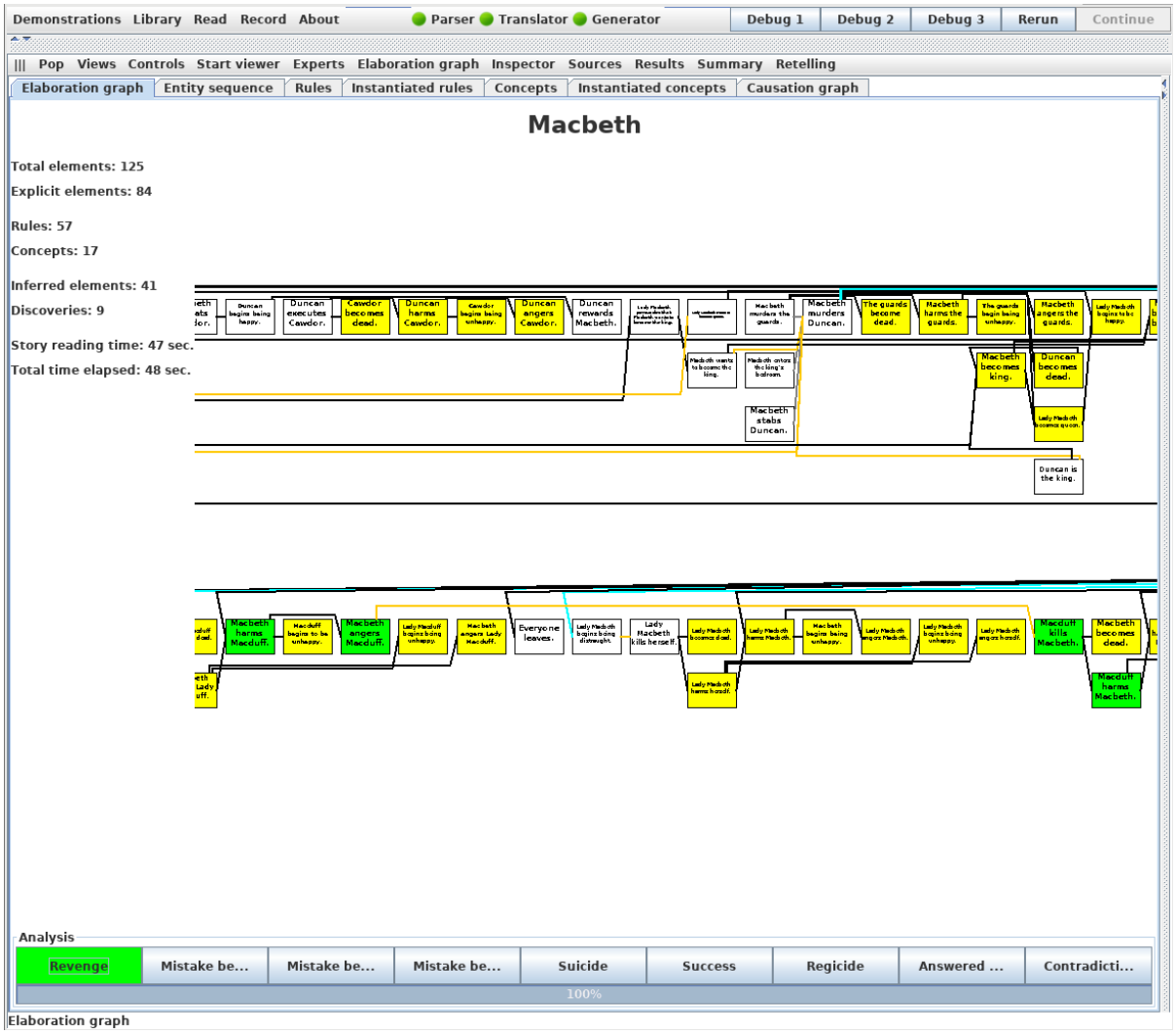

Figure 2: The *elaboration graph* shown here depicts the events in a simplified version of *Macbeth*, including deduced facts and conjectured causal connections. Concept patterns such as Revenge (highlighted in green) emerge from chains of such causes or inferences in the narrative.

units). Many, but not all, concept patterns involve *leads-to* relationships; that is, relationships that emerge from an unbroken chain of events and inferences in a story. For example, the concept pattern Revenge occurs whenever one act of harm is connected to a reciprocal act of harm through any number of intervening story elements (Figure 2).

3 Presumption rules fill in gaps

I have focused on a specific type of question: what would happen if we remove a particular element of the story. Enabling Genesis to simply remove an element and re-analyze the story is technically straightforward—most of the challenge there involves parsing the question, matching it against the story, and then re-running the story with Genesis's existing story understanding apparatus. The true technical challenge arises from the fact that a story with a missing element is not *simply* a shorter story. Consider, for example, the widespread ramifications of removals like these:

1. What if the assailant did not have a knife?
2. What if this character were not selfish?
3. What if the reader did not have a particular cultural background?
4. What if the sidekick had not left in the second act?

Each of these questions may drastically alter the outcome and interpretation of the story. And though these questions all have the same superficial form, they differ widely with respect to the kind of information they affect and the skills required to respond competently. All of the interesting things happen in the omission, so if you don't have the right latent background information—beyond the information you used to understand the original scenario—you will not be able to describe the alternative scenario intelligently!

What kind of background information is necessary, and how do we process it? For demanding what-if questions such as "What would happen if the sidekick had not left in the second act?", we might rely on an extensive and varied range of information, and the process for manipulating it might involve a lot of search and evaluation to find a plausible answer. For certain kinds of questions—which I call *gap-filling questions*—we seem to fill in gaps more or less automatically: we reflexively fill in missing information using commonsense knowledge and cognitive biases. Though aspects of this kind of unreflective filling-in can have unwelcome consequences—prejudice, functional fixedness (where we overlook new uses for familiar objects) (Duncker and Lees, 1945), hackneyed tropes,

and jumping to conclusions, for example—it nevertheless lies at the core of our ability to make a story or a visual scene coherent by hallucinating missing details: using "the stereotypes of what we expected" (Minsky, 2006, Chapter 4), we can process a visual scene before we've seen every detail and we can read stories without needing every connection to be explicitly laid out.

To model this kind of reflexive gap-filling, I introduced **presumption rules** into the Genesis story understanding system. I extended Genesis's rule-matching system (which searches the story for elements that match the commonsense rules in its database, then adds inferences and causal connections to the story accordingly) to handle this new rule type and new behavior.

A presumption rule encodes fragile default knowledge about what to assume in the absence of evidence to the contrary. Hence, you can express many default assumptions as presumption rules such as the following:

- If someone enters the kitchen, then presumably that person wants to eat.
- If an adult stands at the front of a lecture hall, that person is presumably the instructor.
- There is smoke presumably because there is fire.

Declaring a presumption rule

The presumption rule type supplements Genesis's existing family of rule types in that presumption rules introduce genuinely *new, presumptive facts* into the story being read. This behavior is importantly different from the behavior of *deduction rules*, which only add completely certain conclusions ("If you kill someone, that person becomes dead.") and *explanation rules*, which can introduce new connections between existing events, but cannot introduce new events ("A character may kill someone because that character is angry.", interpreted as meaning "If both events occur in the story, tentatively add a causal connection between them").

The syntax for declaring presumption rules is consistent with the standard syntax for declaring other rule types. Presumption rules are signaled by the idiom "presumably" or

"can" (which here means "could potentially"). The keywords "can" and "presumably" can be used interchangeably, and they can be used for rules formatted either as "if xx then yy" or as "yy because xx". For example, Genesis would recognize all of the following rules as presumption rules:

```
xx can enter the kitchen because xx wants to
eat.

If xx stands in front of the lecture hall, then
xx is presumably the instructor.

If there is smoke, there can be fire.
```

As an aside, I note that our everyday language maintains subtly different rules for when *can* or *presumably* are the right word: in an inference rule, *can* has a connotation of being "one presumption among many good alternatives", while *presumably* has a connotation of being "the one obvious presumption to make". For the purposes of this thesis, however, the two keywords can be used interchangeably.

Overshadowing

Because presumption rules encode *default* knowledge, they will only introduce a new event into the story if no other explanation exists. As such, presumption rules can become "overshadowed" by earlier rules that compete to provide an explanation. Presumption rules can be overshadowed by explicit sentences, or inferences, or explanation rules.

For example, consider the presumption rule "xx shoves yy presumably because xx dislikes yy". In a story, the following sentences would *match* this presumption rule because shoving occurs, but would preclude the rule from firing and introducing an explanation because in each case an explanation already exists:

- Riley shoves Casey because Riley and Casey are actors.
- Riley may shove Casey because Casey is in harm's way.

Conversely, presumption rules can introduce connections that overshadow explanation rules. Hence by controlling the order in which presumption rules and other rules fire, you

can change which kind of explanation will dominate—the default presumption explanation or an alternate explanation. Such rule precedence provides a potential way to model differences in how attached people are to their presumptions. Some presumptions may be easily overridden; others, as in certain forms of psychopathology, are so firmly embedded that little can override them.

Future work for presumption rules

For the work in this thesis, the "presumptive" nature of presumption rules appears in two ways: first, the fact that they only fire to fill in explanatory gaps; second, the fact that they are intended to be used for abductive inference—inference that is provisional and uncertain. This is the extent of the presumptive nature; once the presumption rules have fired, Genesis itself does not currently treat them any differently than other rule types. In particular, Genesis does not yet have the capacity to discard presumptions in light of new information.

In future work, however, I envision developing a more elaborate system for managing presumptions: not only introducing new presumptions, but comparing them against existing facts, choosing between competing presumptions, presuming large frameworks of knowledge rather than individual events, and revoking presumptions in light of new evidence. Such extensions would solidify the role of presumption rules as encoding fragile default knowledge.

4 How to argue for self-defense

Let us now look at an example story to see how these presumption rules enable Genesis to reason hypothetically. In the following court-inspired story about a break-in, we want to understand the presumptive reason for a particular character's actions and whether consequently those actions count as self-defense:

> **Martha's response**. Alex and Martha have despised each other for a long time. George is Martha's spouse. The hour is late; George and Martha are asleep.

> Martha wakes up because Alex breaks a window.
> Alex begins shouting, then Alex brandishes a knife.
> Martha shoots Alex; Alex dies.

In addition to the text of the story, I supply background commonsense rules and concept patterns to capture what a typical person might know or believe in addition to what is explicitly in the story. The following rules and concepts constitute the relevant subset:

1. `If xx brandishes a knife, then presumably xx intends to harm someone.`
2. `xx may shoot yy because yy intends to harm someone.`
3. `xx can shoot yy because xx despises yy.`
4. `yy's intending to harm someone leads to xx's shooting yy.` (Self defense)
5. `xx's despising yy leads to xx's shooting yy.` (Spiteful violence)

As you will see, I have embedded within these rules a reader mentality that prefers a self-defense explanation over the alternative. Through hypothetical reasoning, Genesis will begin to discover this tendency for itself.

With these rules and concepts in place as context, Genesis evaluates Martha's actions in the story as an instance of *self-defense* (Figure 3). In detail, Genesis arrives at the self-defense characterization through the following steps: First, Genesis reads in the story that Alex brandishes a knife. Genesis matches and fires the first rule, presuming that Alex intends to harm someone. When Genesis reaches the event "Martha shoots Alex", Genesis matches the second and third rules; due to Genesis's built-in precedence of explanation rules over presumption rules, however, Genesis fires only the earlier match, adding a putative causal link between "Martha shoots Alex" and "Alex intends to harm someone." Because a cause has been established for why Martha shoots Alex, the presumptive cause "Martha shoots Alex because Martha despises Alex" is overshadowed and is not added.

We can determine the relevance of Alex's knife to this evaluation by asking Genesis "What would happen if Alex did not

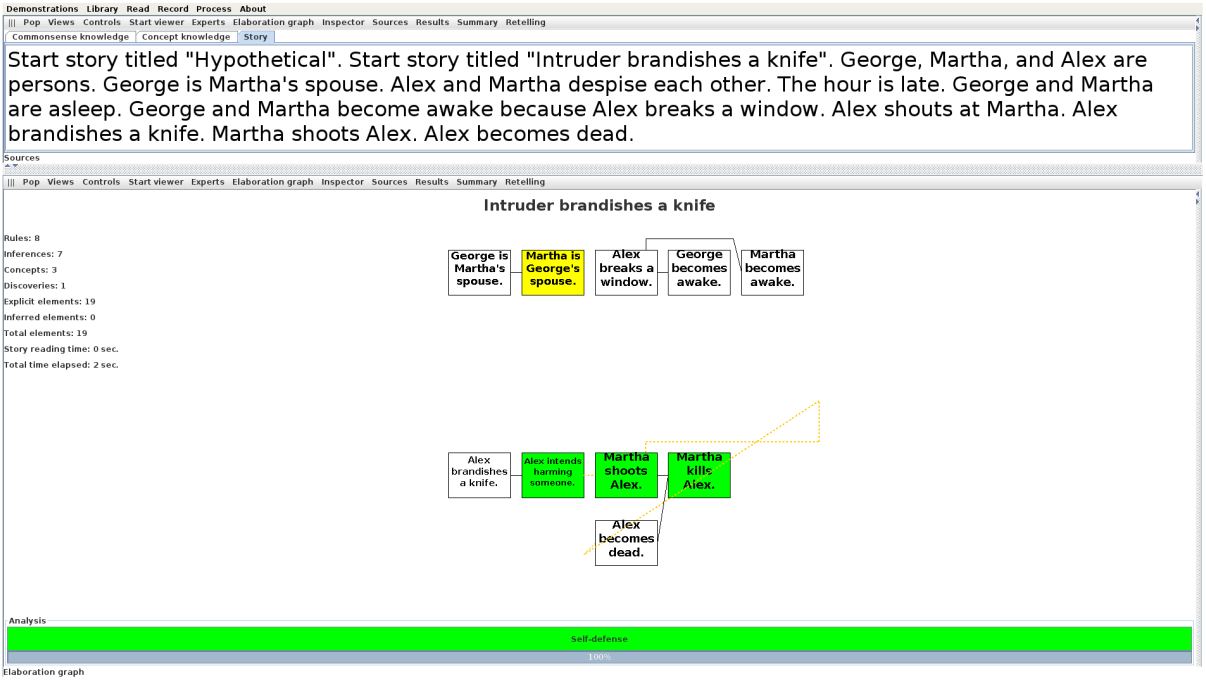

Figure 3: The initial setup of the story. Through a sequence of tentative (explanation rule) and deductive (deduction rule) inferences, Genesis detects the *self-defense* concept pattern (highlighted in green.)

brandish a knife?". Accordingly, Genesis removes the knife from the story and reads the story anew (Figure 4). The fragile default cause-and-effect knowledge I have provided in the form of rules fills in the gaps.

Whereas in the first story, Genesis interprets Martha's actions as self-defense, in the variant story Genesis interprets Martha's actions as *spiteful violence* instead. At an intuitive level, this difference arises because in the absence of a knife, Genesis considers the pre-existing antipathy between Martha and Alex to dominate as an explanation for Martha's response. At a mechanistic level, this difference arises because in the absence of a knife, the presumption that Alex intends to harm someone is no longer introduced, and so the explanation of self-defense no longer applies. Because there is no obvious explanation for Martha's shooting Alex, Genesis searches the story and finds that the presumption rule ("Martha can shoot Alex because Martha despises Alex") provides an explanation.

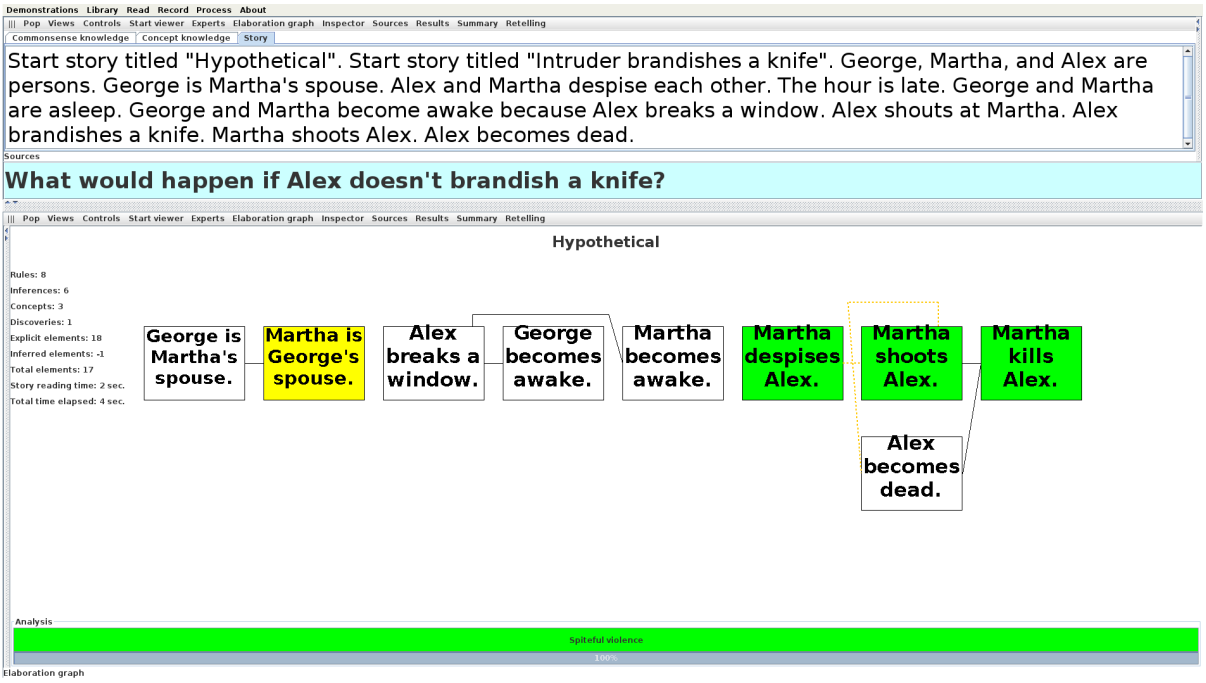

Figure 4: When Genesis analyzes a hypothetical alternative story, without a perceived threat of harm, Genesis discovers an alternative explanation, *spiteful violence*.

The presumption rule, previously overshadowed, fires, and the concept of spiteful violence emerges as a result.

In this way, Genesis is able to use presumptive defaults to stitch together the details of a new story after an element is removed. At this point, some readers may object that they do not agree with Genesis's analysis of this story: some will say that surely the frightening midnight break-in was reason enough for Martha's violent response in both cases? Or surely violence is a wrongheaded way to resolve any conflict, even in times of great danger? Or surely we should consider the consequences of not using a gun, the prior history between Alex and Martha, whether Martha was defending someone else, and so on.

These are all cogent reactions to Genesis's particular analysis of the story. But though you might disagree with Genesis's particular model, this is not a threat to the mechanisms that we are trying to develop; the purpose of presumption rules and other rule types (and more generally the purpose of Genesis's commonsense knowledge) is *not* to definitively

model a full, accurate, objective, uncontroversial analysis of the story—whatever that could mean. Instead, the purpose of this commonsense information is to provide a framework and a language for describing the knowledge, beliefs, and biases of many different kinds of cognitively plausible readers. I argue that presumption rules advance this purpose by adding a much-needed capability, and that using presumption rules in concert with existing rule types, we *can* model such a wide variety of perspectives, including a more territorial, more pacifist, or more precedent-driven viewpoint than the one Genesis embodies here.

5 Simulate-and-inspect is a brute-force approach

At this point, it is helpful to take a step back to discuss the general approach I take in this thesis to answering hypothetical reasoning questions. I call it the simulate-and-inspect approach.

In the preceding section, I described the "simulate" component, where the program removed an event from the story and then re-read the story from the top. In the next section, I shall describe the "inspect" component, where the program discovers and summarizes the resulting differences between the two runs.

Of course, the simulate-and-inspect approach differs from human approaches to solving hypothetical reasoning problems, in that it requires reading the entire story again—Genesis does not yet have enough knowledge to selectively revisit only the parts of the story that might reasonably change, or include only the rules that may become relevant as a result of the change, for example. As a result, the mechanisms that Genesis uses to arrive at its answers are different from the mechanisms that humans use.

Nonetheless, the simulate-and-inspect approach is useful for modeling the kinds of *answers* that humans characteristically give and the *knowledge* required to give them. It is, therefore, a model at the competence level rather than the performance level (Chomsky, 1965), or at the computational

level rather than the algorithmic level (Marr, 1982). This work constitutes a first step toward enabling Genesis to analyze the differences in the story as well as the differences in its commonsense knowledge that make a difference.

In future work, this simulate-and-inspect-based approach could be extended to include many other strategies that humans use to answer what-if questions, such as remembering a previous answer, reasoning by analogy with another problem, reasoning using abstract domain knowledge, using shared memory to avoid duplicating events from both stories, using a K-line approach (Minsky, 1980) to locate relevant points of difference, and so forth.

6 Automating side-by-side comparisons

Having introduced presumption rules and enabled Genesis to answer gap-filling hypothetical questions, I now describe a procedure by which Genesis can compare the original and variant stories and summarize the relevant differences automatically. This kind of comparison procedure is a useful tool for the human operator asking what-if questions because it produces a distilled summary of the essential differences. This procedure is also useful beyond the scope of this thesis because it is capable of comparing *any* two stories, not just stories produced by the what-if question answerer. In particular, in future work, this procedure could also be useful for enabling Genesis to reflect on its own knowledge—because the differences are expressed in Genesis's own internal language, Genesis could potentially use such summaries to monitor and learn about its own story-processing procedures, telling and analyzing its own story.

The procedure starts by comparing two stories at the level of individual events, producing an exhaustive account of fine-grained differences. These differences consist of events that were introduced or removed between the two stories.

In order to compare the events of the two stories, I employed Genesis's implementation (Fay, 2012) of the algorithm from biology known as the Needleman-Wunsch alignment algorithm (Needleman and Wunsch, 1970). This algorithm, originally developed to align nucleotide sequences, takes two lists

of symbols and produces a description of how to transform one sequence into the other through an (optimally short) succession of additions, removals, and matched pairs. (In a typical Genesis use case, we use the algorithm to align stories that differ in only a few places.)

With the original story and the variant story having been aligned, the program prints out an exhaustive list of differences (where both stories, of course, have been elaborated with presumption rules and other commonsense information so as to expose a nontrivial amount of difference—a story with a removed element is not *simply* a shorter story). In the case of the self-defense story (Section 4) where Genesis is asked "What would happen if Alex did not brandish a knife", the program reports two differences, both removals (See Figure 5):

- "It's no longer the case that 'Alex intends to harm someone, presumably because Alex brandishes [a] knife' "
- "It's no longer the case that 'Martha shoots Alex presumably because Alex intends harming someone.' "

This fine-grained analysis provides a detailed account of the differences between two stories—a way of judging their Hamming distance, in other words. But because the procedure does not currently have a method for filtering or summarizing those differences for different purposes, the list of differences tends to become cumbersomely large for longer stories, burying key differences beneath a deluge of detail.

To remedy this problem for longer stories and to provide the ability to align stories which are very different in their particulars but similar in their overarching themes, I introduced a more coarse-grained analysis—analysis at the level of *narrative theme*. To compare stories thematically, the program collects the list of concept patterns from both stories. The pair of lists—consisting of the names of instantiated concept patterns such as "Macbeth's revenge", or "Martha's self-defense" or "Dorothy's success through adversity"—are themselves compared using the Needleman-Wunsch algorithm to determine which themes differ between the two stories. These thematic differences are reported alongside the fine-grained event-level differences to provide analysis at two different resolutions.

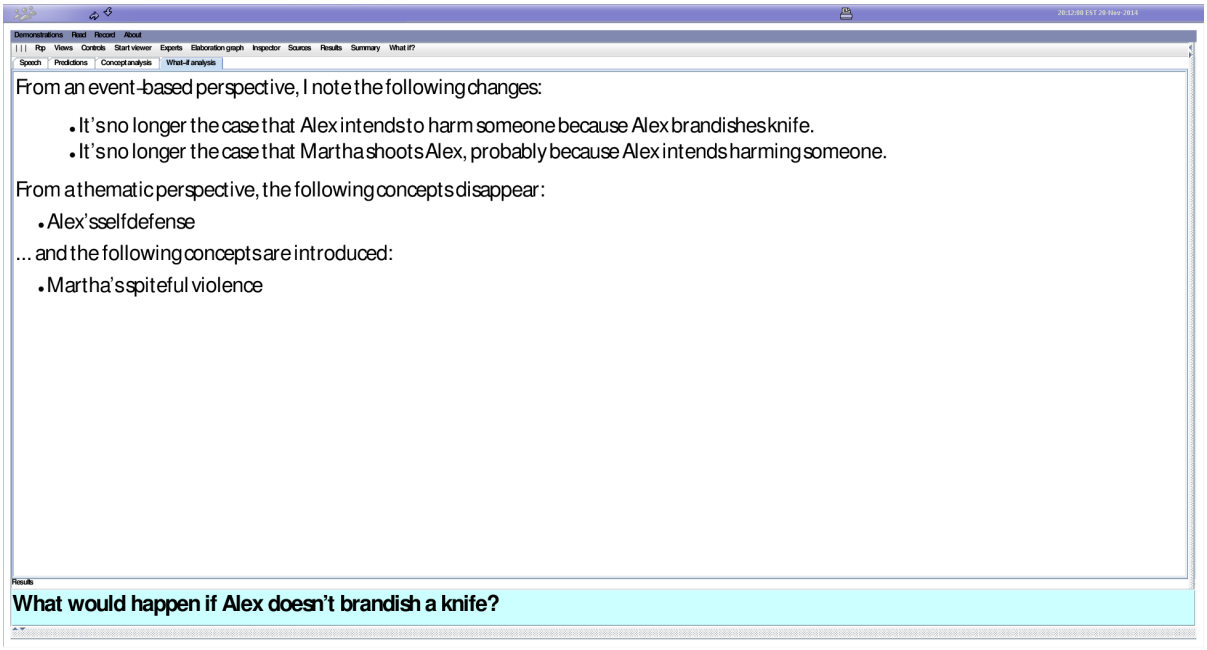

Figure 5: The side-by-side comparator automatically summarizes the differences between the original "self-defense" story and the hypothetical variant at two levels of granularity. At the fine level, it reports differences at the level of individual elements. At the coarse level, it reports differences at the level of thematic concepts.

As an aside, in developing this project, I found it necessary to develop a shorthand naming convention for instantiated concept patterns so that I could print them as text onto the screen and compare them symbolically using the Needleman-Wunsch algorithm. The convention—an *ad hoc* trick, but workable in practice—consists of combining the name of the generic concept pattern ("Pyrrhic victory", "Aggression of a bully", etc.) with the name of one of the participants. The choice of participant is a subtle one in general, because in different circumstances you may variously want to focus on the subject or the object of a sentence or both. (In the play *Othello*, do we want to refer to Iago's nefarious plot as *Iago's betrayal* of Othello or *Othello's betrayal* by Iago? The right focal point will differ based on the analytical context.) Despite this subtlety, I opted to simply use the first name mentioned in the concept pattern. This naïve policy works generally well, producing names such as "Martha's self-defense" and "Mac-

beth's murder". In the future, an extension of this program could vary the policy according to context so that it produces naturalistic descriptions for a wider variety of concepts.

Put together, the event-based analysis and thematic analysis produce descriptions as shown previously in Figure 5.

7 The program handles what-if questions about viewpoints, too

You have just seen how the gap-filling program can read a story about a break-in and answer questions such as "What would happen if Alex didn't brandish a knife?", summarizing the major differences automatically. But the surprise is that, as a result of the unified way in which Genesis represents story elements, the gap-filling program can not only vary explicit facts in the story, but also, without any additional code, vary *aspects of the reader* such as the reader's commonsense knowledge, cultural background, or political stances. Experiments with lesioning this kind of knowledge and cultural background allow the user—and Genesis itself—to reason about how Genesis's conclusions arise explicitly from particular elements of the reader's background.

In the next example, I'll demonstrate this surprisingly broad capacity with a news story about the 2007 cyberwar between Russia and Estonia, long a part of the Genesis story repertoire (Winston, 2012b). Before I developed the hypothetical reasoning framework for Genesis, Genesis's capability amounted to interpreting the story two different ways as a result of differences in the mental models of readers (as Russia-sympathetic or Estonia-sympathetic). The differences in sympathies result in characterizations of the same act of aggression as bullying or justified retaliation.

The question-answering system I have developed takes this existing capability to another level: it enables the user to generate on-the-fly interpretations from different viewpoints by asking what-if questions and automatically summarizes the differences using the story comparison procedure.

Here is the text of the story, written by default from an Estonian point of view:

> **Cyberwar**. I am from Estonia. I have bias. Estonia built Estonia's computer networks. Estonia insulted Russia because Estonia relocated a war memorial. Russia wanted to harm Estonia. Russia is bigger than Estonia. Estonia relocated the war memorial because Estonia did not respect Russia. Someone attacked Estonia's computer networks after Estonia harmed Russia. The attack on Estonia's computer networks included the jamming of the web sites. The jamming of the sites showed that someone did not respect Estonia. Estonia created a center to study computer security. Estonia believed other states would support the center.

As before, I developed a set of interacting rules, some of which are overshadowed. In particular, the relevant subset is:

1. `If I am from Estonia, then I am Estonia's friend.`
2. `I may have bias because I am Estonia's friend.`
3. `I can have bias because I am Russia's friend.`
4. `I am xx's friend. xx's angering yy leads to yy's harming xx.` (Aggression of a bully)
5. `I am yy's friend. xx's angering yy leads to yy's harming xx.` (Teaching a lesson)

Because this story contains a cue to Genesis about the reader ("I am from Estonia"), Genesis concludes that the reader is Estonia's friend and that consequently this allegiance is the source of the reader's bias. ("If I am from Estonia, then I am Estonia's friend", "I may have bias because I am Estonia's friend.") According to these rules, there are exactly two potential sources of bias: Estonia-centered bias (rule 2), and Russia-centered bias (rule 3).

Because the story includes the cue that Genesis (as story reader) is from Estonia ("I am from Estonia"), Genesis views the leads-to sequence "Estonia insults Russia. . . Russia attacks Estonia's computer networks" as embodying the concept pattern "Aggression of a bully".

If, however, we ask the gap-filling program "What would happen if I am not from Estonia?", a different pattern emerges:

Genesis uses the presumption rule to conclude that it has allegiance with Russia instead ("I can have bias because I am from Russia"), whereby Genesis characterizes the same sequence of events as the concept pattern "Teaching a lesson" (Figure 6).

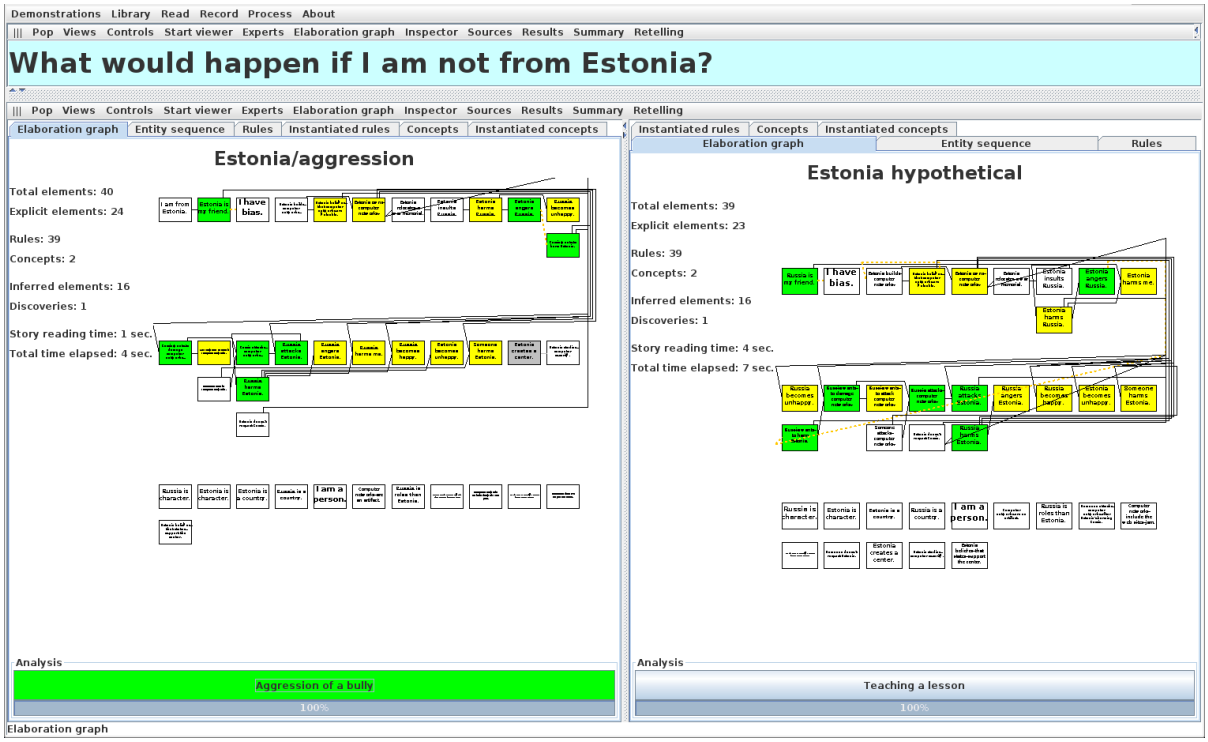

Figure 6: Here we see the original (Estonia-sympathetic) interpretation and hypothetical variant (Russia-sympathetic) interpretation of the cyberwar story. The same event is perceived as Aggression of a bully in the Estonian view and Teaching a lesson in the Russian view. Presumption rules elicit a Russia-sympathetic bias when the Estonian bias is removed.

The differences between the two stories, as found by automatic side-by-side comparison, are summarized in Figure 7.

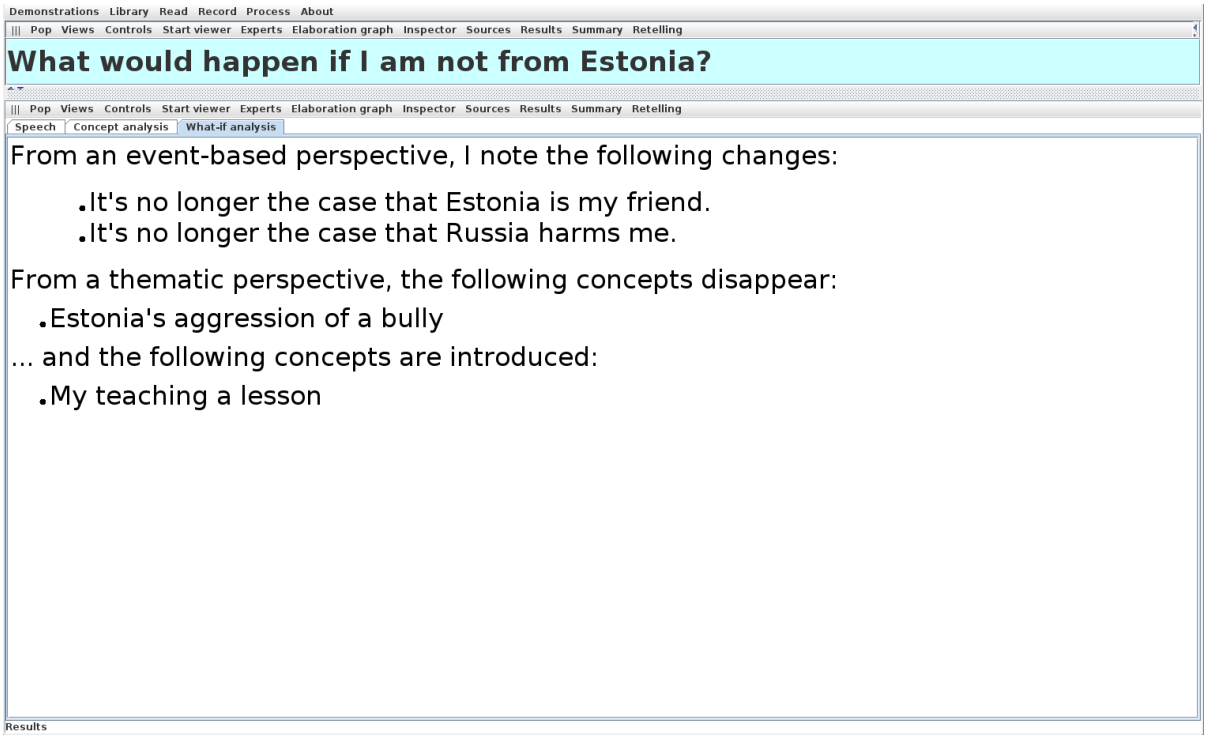

Figure 7: The side-by-side comparator produces an automatic summary of the differences between the Estonian and Russian view of the conflict. The individual differences are noted first, followed by the major thematic differences (Aggression becomes Teaching a lesson).

Applications of what-if questions to moral reasoning

Chapter Summary

Having developed a toolkit of representations and processes for hypothetical reasoning in the previous chapter—namely, presumption rules, a what-if question answering system, and an automatic side-by-side comparator—I proceed to apply the toolkit to the realm of moral reasoning.

I describe **means-ends rules** which empower Genesis to reason about what people want and how they acquire it. Informed by means-ends rules, Genesis is able to **morally judge characters' actions** in light of hypothetical alternative methods they could have used to accomplish their goals. One surprising consequence is that Genesis tends to forgive bad actions performed by characters with few alternatives, perhaps in the same way that people do.

In an extension of this means-ends reasoning process, Genesis infers characters' moral constraints. I introduce personality-like representations called **personas** which cohesively describe characters' goals, how they achieve them, and the moral constraints they observe when they act.

Finally, I capture aspects of how humans think about characters' personalities with a **problem-solving approach to assigning personality**, a theory which uses hypothetical reasoning in a fundamental way. Combining all of the components developed so far, I implemented this theory on top of the Genesis system in a program I call PERSONATE (Figure 8). The program PERSONATE is able to read a short story, infer characters' goals, evaluate their choices in light of their available alternatives, assign personalities, and finally answer general questions about how those characters will react to new situations by using their personality as a useful constraint.

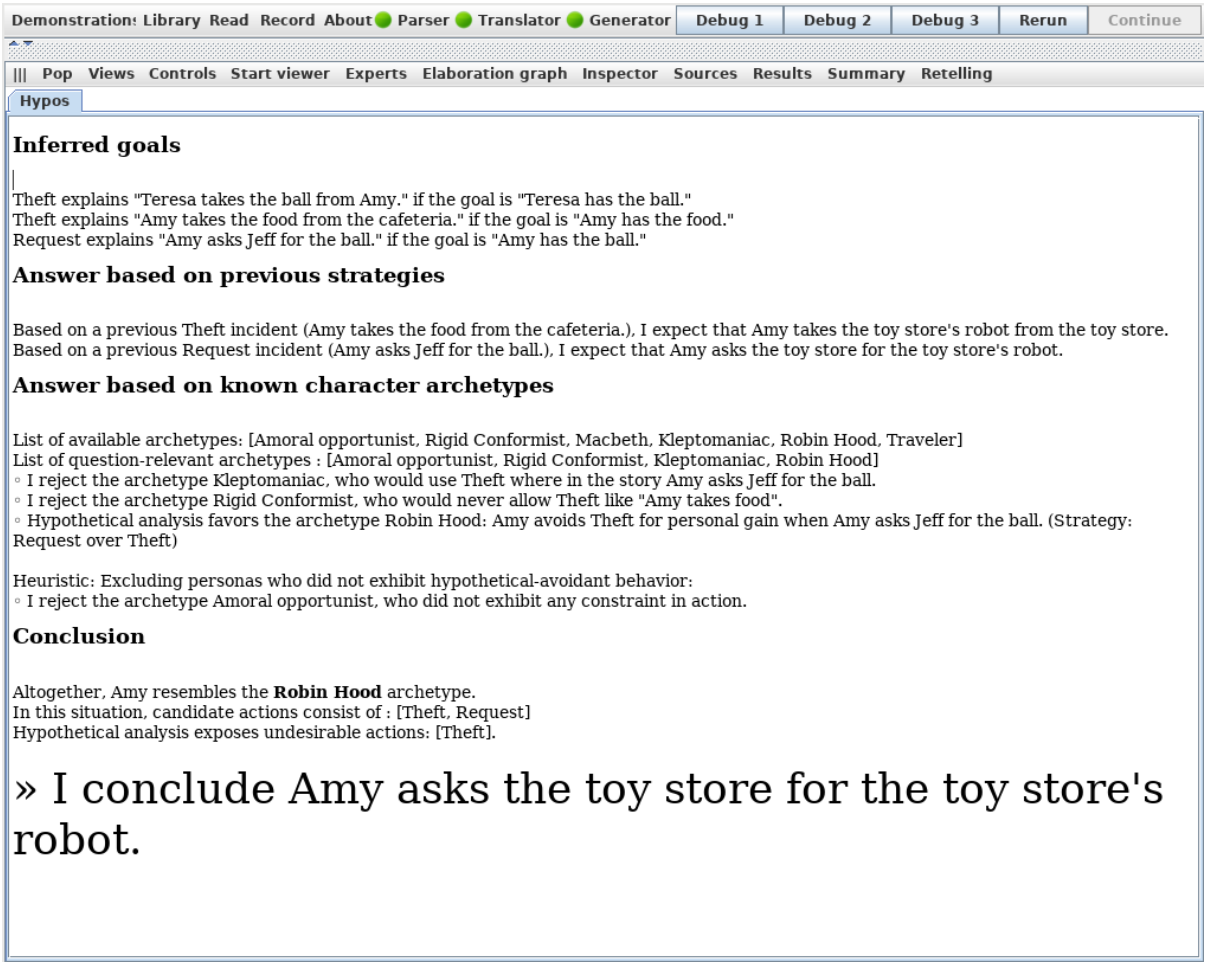

Figure 8: PERSONATE embodies a problem-solving approach to assigning personalities to characters in a story. PERSONATE augments the hypothetical reasoning tools developed in the previous chapter with specialized representations for goals, personalities, and moral values, yielding a program that can reason hypothetically in a human-like way about what characters would and would not do.

8 Means-ends rules support reasoning about characters' actions and intentions

Structured knowledge for moral hypotheticals

Our human moral reasoning capability rests on a cohesive framework of knowledge about goals and strategies. Consider how our basic ability to reason morally depends on mastering a huge number of interrelated concepts including: ac-

cidents, agency, agreement, alternatives, apologies, arbitration, attempts, bias, blame, coercion, commensuration, conflict, constraints, conventions, costs, crimes, culpability, culture, debt, deception, decisions, dependence, deterrents, distractions, domination, duress, duty, escalation, excuses, exoneration, failures, fairness, false beliefs, forgiveness, freedom, goals, goodness, identity, ignorance, impairment, impartiality, innocence, intervention, justifications, mental models, mercy, mistakes, moral rules, norms, paragons, passion, persons, plans, preferences, prohibitions, punishment, recklessness, reparations, reputation, retaliation, shame, side-effects, strategies, temptation, tort, trust, universals, utility, values, vengeance, virtues, and will.

These concepts can only be defined and understood with respect to one another. As such, together they represent more than just a collection of disconnected cause-and-effect knowledge; they represent an interwoven *theory*. It is because we humans possess rich moral domain knowledge consisting of interrelated theoretical concepts, processes, and constraints that we are able to reason so expertly about why people do what they do. Hence if we are to develop programs that reason morally the way humans do, we must supply them with a similarly rich framework of moral domain knowledge.

I have already described presumption rules, which encode fragile assumptions about possible causes and effects. Next, I'll introduce another specialized rule type, the means-ends rule, which encodes possibilities about what people want and how they obtain it. Means-ends rules constitute an essential part of this moral reasoning framework.

Means-ends rules relate actions to intentions

A **means-ends rule** is a new rule type for Genesis which represents goals that people typically have and what methods they use to achieve them. Means-ends rules consist of a name, a goal to be achieved, a means of achieving that goal, and potentially some prerequisites which must be satisfied in order for the strategy to be reasonable. (By this definition, means-ends rules are reminiscent of STRIPS operators (Fikes and Nilsson, 1971), which have long been used to represent steps in plan-

ning algorithms.)

As a concrete example, the two means-ends rules shown in Figure 9 describe two different strategies for obtaining some physical object that someone else has: the "Theft" rule describes how you could take it from that person, while the "Request" rule describes how you could ask for it. (Of course, neither strategy is guaranteed to work—means-ends rules are intended to represent possible *strategies*, not surefire techniques.)

```
name: "Theft"
goal: xx has zz.
prerequisites: yy has zz.
method: xx takes zz from yy.
```

```
name: "Request"
goal: xx has zz.
prerequisites: yy has zz.
method: xx asks yy for zz.
```

Figure 9: Means-ends rules encode possible *means* for attempting to achieve a particular *end*, possibly assuming certain *prerequisite conditions* are met.

Means-ends rules support several kinds of inference. For example, you can search a story for places where a character's action matches the *means* part of a rule; if the rule's *prerequisites* also appear in the story, you might infer that the character has that particular *goal* in mind. Or, having guessed a possible *goal*, you can use your library of means-ends rules to look up alternative means of accomplishing that same goal; the fact that a character chose one method over another may tell you something about what the character values or what strategies the character knows.

9 Judging the actions a character didn't take

A playground example

The program described in this section is based on this idea of considering hypothetical means-ends context. The program, built on top of the Genesis story understanding system, looks for alternative (perhaps morally better or worse) ways that a character could have acted and judges the character accordingly. Within this paradigm, for example, we might consider a person to be *vicious* if that person chooses violent means to achieve their ends when we know that there are other effective

alternatives, or *noble* if the character avoids stealing despite having an obvious motive and opportunity.

In detail, by using a library of hand-coded means-ends rules, the program infers the putative goals of each character in a short story. Then, the program evaluates the characters' choice of action, not only with respect to the consequences of those choices, but also with respect to other alternative actions the characters presumably could have chosen for achieving their same goals.

The following mini scenario serves as an illustration:

> **Patrick's incivility**. Patrick and Boris are at the playground. Boris has a ball. Patrick takes the ball from Boris and plays with the ball.

If the program knows at least the *Theft* and *Request* means-ends rules defined previously, then the program will be able to infer that Patrick takes the ball because Patrick wants the ball, and that moreover Patrick could have asked for the ball, instead. Hence the program can evaluate Patrick's stealing the ball in contrast to asking for it.

In order to compare two strategies such as Theft and Request, we need a method for deciding which action is better. Here, in particular, we are interested in which actions are better *morally* speaking— the word "better" acts as a higher-order function in that there can be many dimensions for evaluating and comparing the goodness of actions—more efficient, less risky, more diplomatic, faster, etc. (Sloman, 1969). The problem of how to represent the consequences and costs of actions is interesting and complex. We might assign numerical scores to the goodness or badness of consequences, analogous to the Goldstein scale (Goldstein, 1992) which assigns numerical scores to international acts of aggression and conciliation. Or we might use a more qualitative method for comparing actions, grouping the ones that are approximately equivalently harmful, and ordering actions by which are comparatively "better" or "worse" than others.

For our purposes in this simplified world, I have adopted an all-or-none model of morality: an event in the story is either blameworthy or neutral. We can use even this simple "dumb-

bell" model of morality (Minsky, 2006, Chapter 9) to make hypothetical judgments about actions. Hence, in this case, the program considers Patrick's stealing to be blameworthy because (through intervening rules) it results in Boris becoming sad—an event marked as morally wrong. Furthermore, using hypothetical reasoning, the program concludes that Patrick's action was moreover *unpardonable*, because Patrick could have instead asked for the ball without any negative consequence (Figure 10).

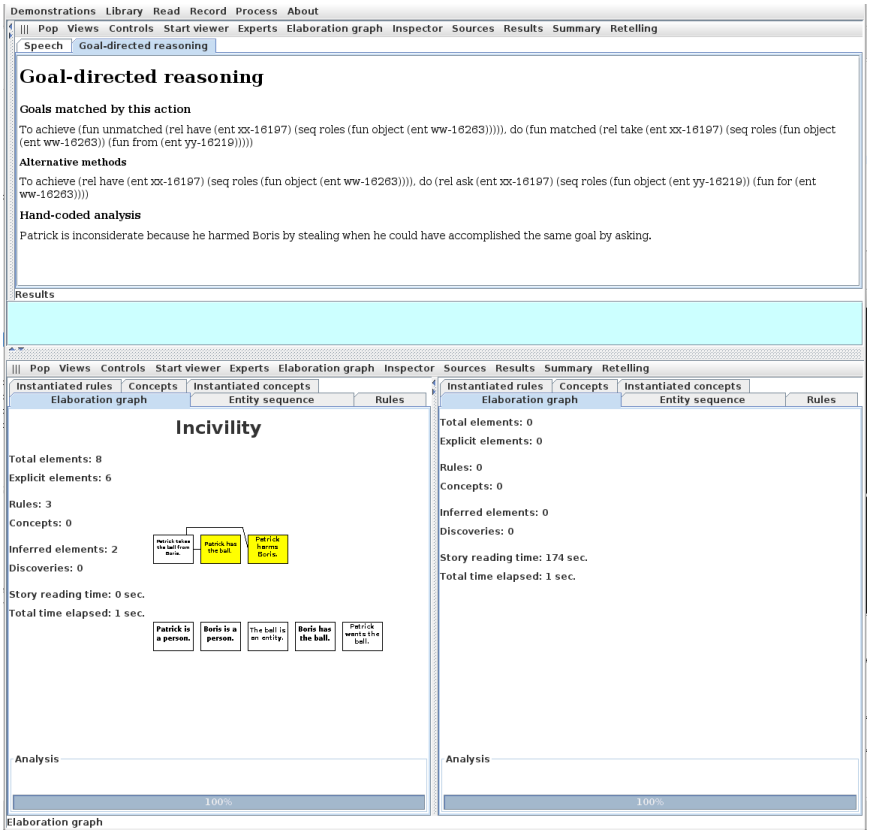

Figure 10: In the first setup, the evaluating program is given knowledge of several goals, methods for achieving them, and side-effects of each method. Hence the program concludes that Patrick is inconsiderate for taking the ball when asking might have caused less harm.

Contrast this analysis with a control version of the story in which Patrick *does* ask for the ball (Figure 11).

Patrick's variant civility. Patrick and Boris are at

the playground. Boris has a ball. Patrick *asks Boris for the ball.* Boris gives Patrick the ball, and Patrick plays with the ball.

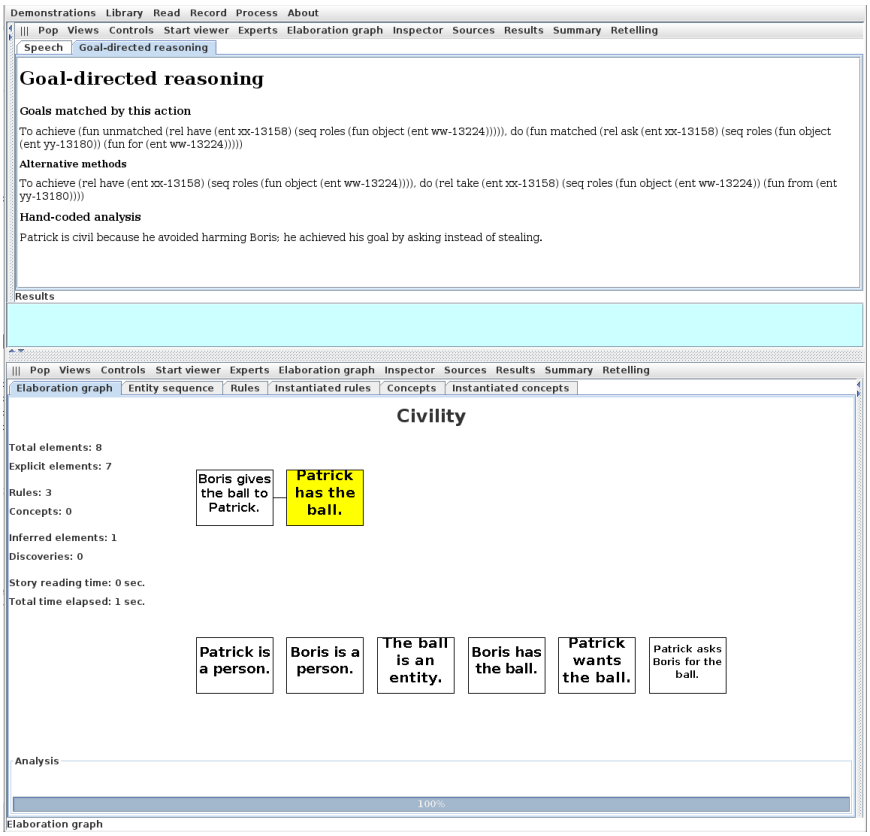

Figure 11: In the second version of the story, Patrick is characterized as civil because Patrick's actions have no known negative side-effect (whereas the alternative—stealing—does).

The limits of imagination affect moral judgment

The surprise comes when, having read the original story where Patrick steals the ball from Boris, we use Genesis's new what-if capability to ask "What if I forget the request strategy?", in which case Genesis (interpreting the question as a special idiom) temporarily removes the Request means-ends rule from its database and re-reads the story (Figure 12).

As with the original story, the program correctly infers Patrick's goal by aligning what happened in the story ("taking the ball") with a related goal in the database, represented

as a template story ("If xx has yy and zz wants yy, zz may take yy from xx".) Also as before, the program acknowledges the negative consequences of such an action ("If xx has yy and zz takes yy from xx, then xx presumably becomes sad.")

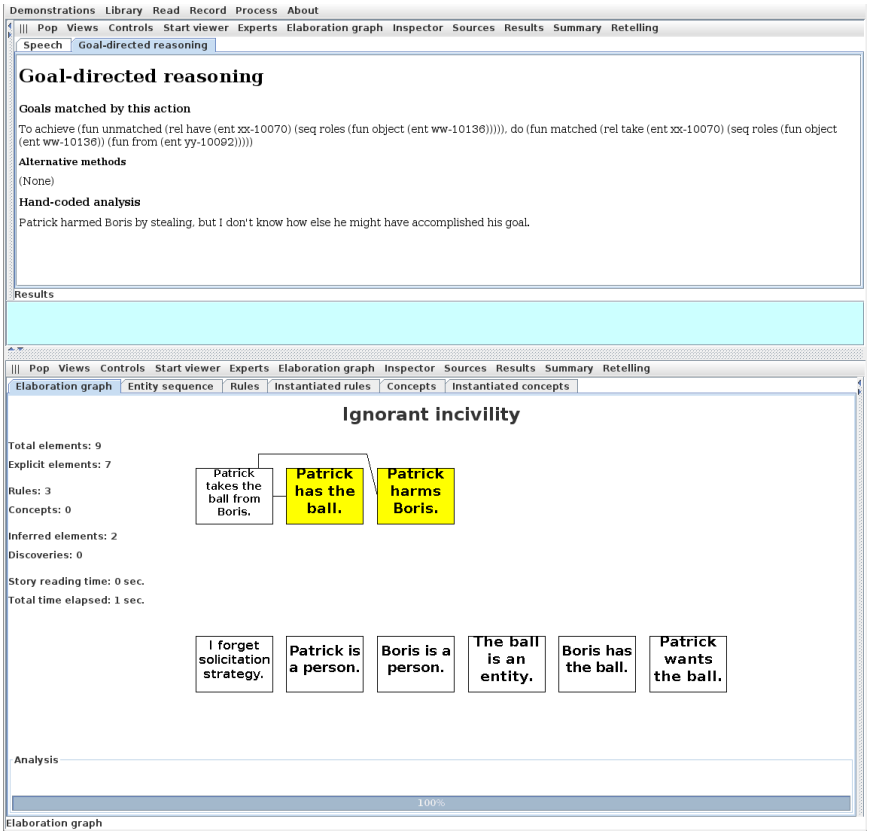

Figure 12: Without knowledge of multiple means, the program becomes confounded: Patrick harmed Boris by taking the ball, but the program knows of no other method for achieving the same goal. This sort of ambivalent reasoning imitates how humans seem to make judgments about morally exigent circumstances.

The difference is that with restricted knowledge of alternatives, the program can no longer find a better alternative than stealing. It is essentially in the following position: "I believe that Patrick behaved wrongly—but I know of no other way the character could have achieved this goal." From the program's point of view, the act was a kind of unavoidable cost of achieving the goal. This, I think, is a realistic parallel with how humans generally regard people who make choices with bad consequences in desperate circumstances. Moreover, there

is some psychological evidence that suggests that this way of thinking about morality in terms of intent and alternatives is a cognitive universal (Saxe, 2016).

Putting it all together

This program uses our toolkit of hypothetical reasoning capabilities to model several interesting aspects of moral reasoning. First, it reasons about goals and motives in a feedback loop in which character actions imply certain means and ends, and those actions are subsequently judged based on alternative means. Second, it assigns moral judgments based on what viable alternatives were available. Such hypothetical reasoning was made possible by a library of commonsense information deployed in the form of means-ends rules. Third, it constitutes a simple model of child morality. The system is childlike because it does not yet possess the reflective capability to question goals themselves or to perform sophisticated cost-benefit analysis. Moreover, it mimics the crude and insensitive behavior of children who are still learning about prosocial ways of achieving their goals. In human adults, such methods may be replaced, on reflection, with more diplomatic means. In this system, additional knowledge provides additional possibility—and additional responsibility. (As the number of viable alternatives increases, the characters' culpability in choosing one of the unethical alternatives grows.) Fourth and finally, this program highlights the exculpatory nature of extreme circumstances: when the knowledge base of the system is artificially limited, it produces the same kind of excuse ("it was wrong, but unavoidable") that we often use to describe choices we make in dire circumstances.

10 Personality as hypothetical problem-solving

Using means-ends rules to find precedents

The moral hypothetical reasoning apparatus I have explained so far can evaluate characters' actions in light of alternatives they could have taken and the consequences thereof. In this

section, I'll show how this same idea can be extended to make a program that infers characters' *behavioral constraints* by considering the actions they could have taken but didn't; these constraints feed into models of *personality* that help the program answer questions about how characters will behave in new situations.

The program **PERSONATE**, which I describe here, predicts character behavior by putting together all of the tools developed so far—including means-ends rules and hypothetical reasoning about moral alternatives—along with personas, partial models of personality. As an illustration of the kinds of problems PERSONATE can solve, consider the following story and associated question:

> **Amy and the robot**. Amy is at the playground. Jeff is playing with the ball. Amy asks Jeff for the ball, so Jeff gives the ball to Amy. Amy plays with the ball. Teresa steals the ball from Amy and plays with the ball. Then, Amy goes to the cafeteria. Kate is Amy's friend. Kate doesn't have food. Amy steals food from the cafeteria and gives it to Kate. Then, Amy walks home. Amy passes a toy store. The toy store has a robot.
>
> *What would happen if Amy wants the robot?*

How might humans answer questions like these, and how could we build a computational model of this process? To start, means-ends rules give a rough-and-ready method to cite evidence in the story as precedent: by matching means-ends rules against the story, Genesis can make goal inferences like the ones shown in Figure 13 and replicated as a list below. (Detail: For matching purposes, an idiomatic transformation finesses "Amy wants the robot" into "Goal: Amy has the robot".)

- Amy probably asks Jeff for the ball because Amy wants the ball. (Request strategy)
- Teresa probably steals the ball from Amy because Teresa wants the ball. (Theft strategy)

- Amy probably steals food from the cafeteria because Amy wants the food[2]. (Theft strategy)

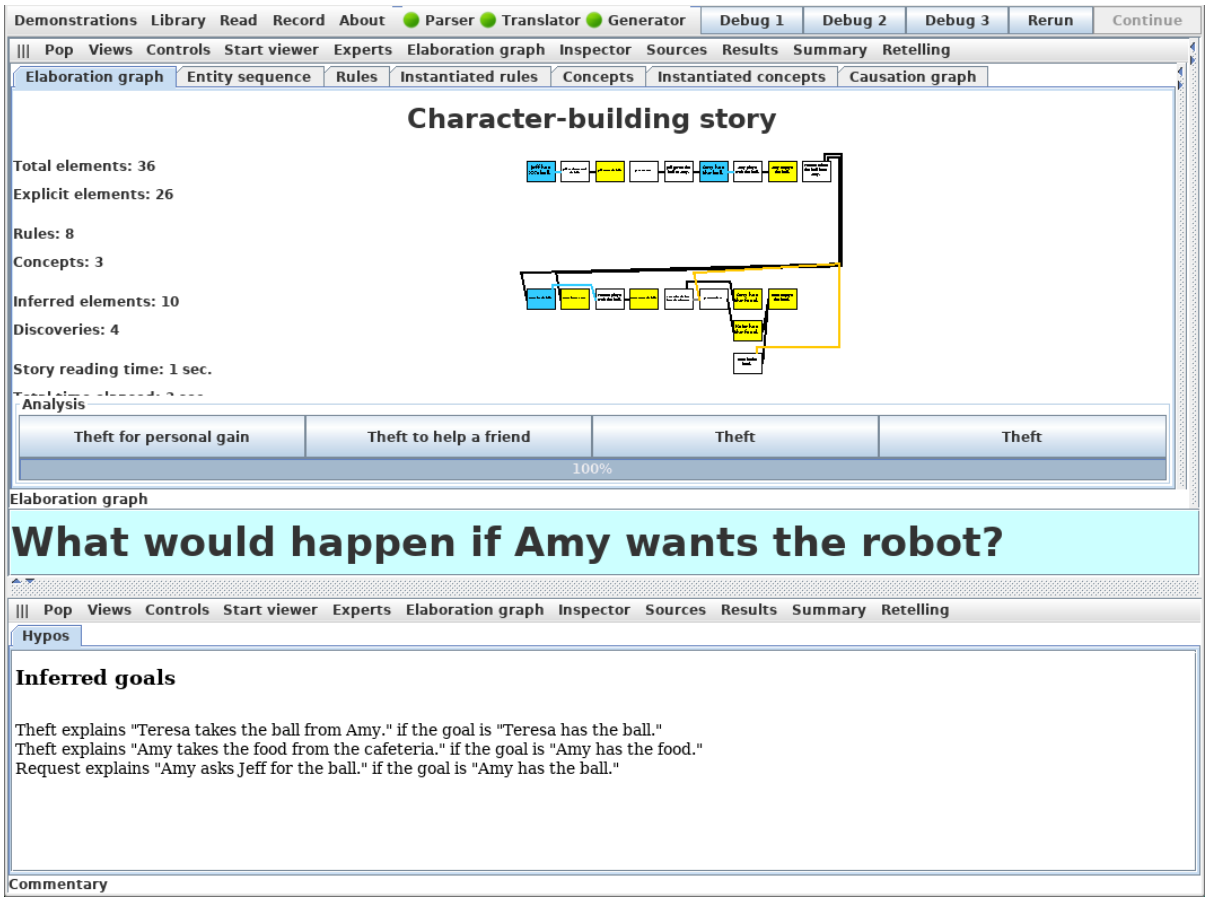

Figure 13: Genesis processes the story about Amy and the robot. In the first stage of processing, narrated in the bottom pane, the program aligns character actions with a database of means-ends rules, imputing goal-directed motives to the actions of characters in the story. These goals provide explanation as well as precedent for future actions.

Hence, owing to of its knowledge of goals, Genesis can see *relevant precedents* in the story, despite the fact that nothing related to "wanting" is mentioned explicitly anywhere and

[2]This inference is probably faulty, as Amy was intending to steal the food for a friend. As implemented, means-ends rules are shortsighted and only consider one-step plans, not two-step plans such as stealing food for the express purpose of giving it to someone else. However, the program's reflexive assignment of goals need not be correct.

the connection between the sentences "Amy wants the robot", "Amy steals food from the cafeteria", and "Amy asks Jeff for the ball" is not overtly apparent. These precedents provide a policy for guessing what a character will do next by enumerating all of the relevant methods that character has used to attain similar goals in the past:

> Amy might *steal* the robot (because Amy stole food).
> Amy might *ask for* the robot (because Amy asked for the ball).

Models of personality provide useful constraint

As a computational model of how humans make reflexive predictions about characters in a story, means-ends based precedent works reasonably well. (In an informal survey, a majority suggested that Amy might steal the robot given that she had stolen before). But these guesses are crude, providing no principle for deciding *which* of these prior methods a character will be more likely to use. In contrast, humans, when pressed, can provide a much deeper analysis than a simple list of past actions: through deliberation, we can determine aspects of a person's *personality* or *moral character*. Here, personality consists of a partial cognitive model for how people choose what to do—their methods, motives, and behavioral constraints. And these constraints, broadly speaking, support our ability to form a cohesive society by allowing us to reliably predict what we ourselves and others will do. ("I trust this person to be punctual and efficient when it counts", "That person can be crass, but is always sincere", "I know that I'm capable of doing this.")

By augmenting this program to reason about personalities in the way humans do, I enable it to refine its predictions about character actions by using personality as a strong constraint: "Of course Amy stole in the story—but Amy would never steal except to help someone else", or "That character is deeply loyal to friends and family and would probably behave differently in different groups." From a problem-solving perspective (Langley et al., 2005), domain knowledge, in the form of personality, entails less search and more realistic assessments.

In the scenario about Amy and the robot, we can ask what we know about Amy's *character*, what information we use to make such decisions, and how we decide between competing alternative characterizations.

How do we decide which personalities fit best?

Now we have a puzzle: How do we humans decide which personalities fit best? For example, some people reading about Amy might conclude that Amy is a kind of *Robin Hood* character: Although she steals some of the time, it is never for personal gain. But there are many other ways to characterize Amy's behavior: why do we not consider Amy to be an inveterate thief, stealing whatever she wants—after all, we have positive evidence that Amy *does* steal food—and what would constitute opposing evidence? Why should we bother to conclude that Amy is operating under a moral constraint (avoiding theft for personal gain on principle), rather than simply acting opportunistically, stealing whenever the mood strikes?

Evidently, we must have a method for evaluating and comparing different personality types. My solution, as embodied in the program PERSONATE, is that by considering actions that characters could have otherwise taken (building upon the moral reasoning approach described earlier), we can infer characters' motives, methods, and behavioral constraints. So informed, we use a series of heuristic rules to reject or promote certain personality ascriptions.

Personas consist of means, motives, and moral constraints

To begin to implement the computational theory I've described, PERSONATE needs to have a representation for aspects of personality, in particular a representation for a character's means, motives, and behavioral constraints. I have already introduced means-ends rules as a way of encoding particular methods (means) for accomplishing particular goals (motives). As for behavioral constraints, there is a vast design space of possible implementations. For example, you could imagine using entire decision algorithms as the basis of the representation; the algorithm would describe how the charac-

ter would choose to act in response to any given situation. Or you could encode a character's hierarchy of goals so that some goals (such as being considerate) take precedence over others (such as acquiring food as efficiently as possible) depending on mood.

For my own purposes, I found it compelling to use *moral constraints* to represent a model of how characters choose to act. In my terminology, moral constraints describe generic situations that a particular character will tend to avoid—constraints such as *avoid causing conflict*, *do not steal*, *avoid being cruel*, *do not eat meat*, or *do not prioritize convenience over duty*.

Moral constraints are a compelling choice because they cleanly separate character-specific information (each character's own moral principles) from the decision procedure used to act (avoid constraints, perhaps with certain priorities or tiebreaking rules). Hence we do not need to describe *in detail* how a character will respond to any given situation; we simply describe what the character generally tries to avoid. These sparse descriptions may sometimes result in extreme situations where we do not have any good idea of how the character will behave—but this seems to be a realistic feature, rather than a bug.

Moral constraints are moreover compelling from an engineering standpoint because you can search for them in a story. In fact, I was able to appropriate the existing *narrative concept pattern* apparatus, previously used by Genesis for story-understanding, as a representation of moral constraints. Hence, we can use concept pattern idioms to define moral constraints like these:

- `xx steals zz from yy` (Lawbreaking)
- `xx's stealing zz from yy leads to xx's enjoying zz` (Theft for personal gain)

From a certain point of view, using concept patterns for this purpose makes good sense, as there is undoubtedly some overlap between moral constraints and the themes of our best narratives. I use the term **forbidden concept patterns** to refer to this special moral-constraint usage of narrative concept patterns.

Putting means-ends rules together with forbidden concept patterns, we arrive at a rudimentary model of personality—a persona.

A **persona** consists of a library of known strategies (in the form of means-ends rules) and a list of moral constraints (in the form of forbidden concept patterns). Figure 14 shows some examples—though not all of them: I have defined other personas, including Pathfinder (a GPS-inspired[3] persona (Newell et al., 1959) that knows several different strategies for traveling from one place to another depending on how far away they are) and Macbeth (which knows strategies for becoming a monarch by dispatching one's predecessor), though only the four personas in Figure 14 are relevant for the worked example I discuss in this thesis.

```
Persona: "Conformist"
Means-ends: Theft, Request
Forbidden concepts: Lawbreaking.

Persona: "Thief"
Means-ends: Theft
Forbidden concepts: None.

Persona: "Opportunist"
Means-ends: Theft, Request
Forbidden concepts: None.

Persona: "Robin Hood"
Means-ends: Theft, Request
Forbidden concepts: Theft for personal gain.
```

Figure 14: A catalogue of personas used by PERSONATE in this thesis. A persona consists of several means-ends rules which encode possible goal-directed strategies, along with a list of forbidden concept patterns which encode the constraints controlling how the persona will deploy those strategies.

As a point of clarification: a persona's means-ends strategies are intended to include all strategies that the persona *knows*, not necessarily the ones that the persona will use.

[3]That is to say, a persona inspired by the General Problem Solver model of Newell, Shaw, and Simon.

For example, the Conformist knows the Theft strategy but will never use it. The Thief represents a kind of child-like person whose only known method for acquiring something is stealing it. And a caveat: personas are only intended to capture fine-grained, task-specific aspects of a person's behavior (such as acquiring a ball). A fuller description of a character's behavior would require multiple personas, variously employed based on task and changing mood. For example, you might temporarily model an angry character using a persona with fewer moral constraints.

In the next section, I'll describe the program PERSONATE, which makes a reasoned human-like argument using hypothetical reasoning heuristics to compare and evaluate the personas in this list as fitting descriptions of Amy from the scenario above. Such reasoning ultimately leads PERSONATE to conclude, through argument, that Amy best resembles the Robin Hood archetype, laying the groundwork for a behavioral prediction and an answer to the question "What would happen if Amy wants the robot?"

11 Four principled heuristics determine personality fit

The program PERSONATE embodies a computational theory of how humans use personality traits to answer prediction questions such as "What would happen if Amy wants the robot?". In particular, PERSONATE uses a human-like procedure for deciding which personality types fit best. PERSONATE captures our intuitive assessments using a small number of principled heuristics for promoting or eliminating candidate personas. These heuristics often centrally involve hypothetical reasoning about characters' available alternatives and avoided outcomes.

The theory, components of which have been explained already in this thesis, is as follows:

1. **How do we recognize relevant incidents?** We link the question and the story through knowledge of means and ends. In this particular story, we look for all actions that have *acquisition* as an implicit goal. This allows

us to link Amy's *wanting* the robot, *asking* for Jeff's ball, and *stealing* food from the cafeteria. Characters' previous actions reveal their goals, their methods, and their moral constraints (or lack thereof).

2. **What do we consider when we deliberate?** We dig up additional evidence, often weighing counterfactual alternatives and highlighting implicit moral constraints. We use this (often hypothetical) evidence to form a mental model of people's motives, methods, and moral constraints—an aspect of *personality*. We use personality models to predict behavior and answer the original query.

3. **Which personalities fit best?** We assign personalities by aligning characters with a subset of known personas that fit best. To assign personalities, we must have methods for evaluating and comparing the fitness of different personality types.

   In this story, to see Amy as a Robin Hood character (who steals but never for personal gain), we must rule out other apparently plausible accounts: for example, that Amy is simply a thief (as precedent suggests), or that Amy is an opportunist (operating without any moral constraint).

   - **Why not just think that Amy is a thief?** We note that Amy asked for Jeff's ball, rather than stealing it. When a character achieves the same kind of goal through different means in different situations, we may resist a one-sided characterization.
   - **Why believe that Amy operates under constraint?** Amy steals food to benefit a friend, but does not steal a ball to benefit herself. When a character could have chosen a constraint-violating strategy but did not, we may infer that the character heeds the constraint.

   Hence, a small number of principled heuristics like these can guide our sense of fit. These heuristics crucially consider hypothetical alternatives to rule out (or promote) certain personality types.

4. **How do we predict behavior using personalities?** Once we know a character's goals, methods, and constraints, we can simulate their possible moves and eliminate the forbidden. When we decide that Amy is a Robin Hood character, we conclude that Amy would refuse to steal the robot, and would be more likely to ask for it instead.

In the following sections, I explain how PERSONATE works in detail, using Amy's story as an example. To start, PERSONATE populates an initial list of candidate personas for the character in the question (in our case, Amy). This list of candidates is extremely permissive: it includes any persona that knows a means-ends strategy the character used in the story. The initial list is permissive in the following way: it will in general include personas whose moral constraints are explicitly violated in the story, as well as personas who account for only a subset of the strategies the character knows. Both kinds of candidate should be immediately disqualified. In fact, the purpose of the first two heuristics I define later is to weed out the two types of candidate, respectively.

Algorithm 1 shows in pseudocode how this initialization procedure takes place, first finding means-ends rules whose *means* match actions in the story, then matching the *ends* of those rules against the ends of persona strategies to establish a list of possible persona candidates.

For example, when PERSONATE searches for means-ends precedents in the story, PERSONATE finds that Amy uses the Request strategy (asking for the ball) and Theft strategy (stealing the food) in the story. Hence PERSONATE will look up any persona that has *either* the Request or Theft strategy (or both) in its repertoire. In our case, the search process finds all four of the personas listed previously in Figure 14: Conformist, Thief, Robin Hood, Opportunist. And these candidates embody the speculative, intuitive questions we have asked about Amy: Is Amy a conformist who always obeys the law? Is Amy a thief who only ever steals to get what she wants? Is Amy a Robin Hood character who steals occasionally but never for personal gain? Is Amy an amoral opportunist who steals or asks without principle or restraint? PERSONATE embodies a theory of how we humans weigh such alternatives.

Algorithm 1: Initialize the list of candidate personas

```
input : a named person character,
        a story sequence story,
        a list of known personas personaTemplates,
        a list of means-ends rules strategyTemplates

/* Infer all strategies used in the story                    */
employedStrategies ← [ ];
foreach strategy in strategyTemplates do
    foreach element in story where character appears in element do
        binding = tryToMatch(strategy.means, element);
        if binding exists then
            employedStrategies.append( populate(binding, strategy)
             );
/* Initially allow any persona that knows a strategy used in
   the story                                                 */
matchedPersonas ← [ ] ;
foreach persona in personaTemplates do
    foreach personaStrategy in persona.knownStrategies do
        foreach strategy in employedStrategies do
            binding = tryToMatch(strategy.goal,
             personaStrategy.goal);
            if binding exists then
                matchedPersonas.append( populate(binding,
                 persona) );
                next persona;
return employedStrategies, matchedPersonas
```

Note that PERSONATE avoids doing too much unnecessary work by ascribing a personality only to the person mentioned in the query—a kind of question-directed search.

After populating the initial list, PERSONATE heuristically shortens this list by eliminating unlikely candidates. This process resembles a kind of near-miss learning (Winston, 1970) of personality type, using counter-examples in the story to hone an emerging model. I use Amy's story as a concrete example to demonstrate these personality-evaluating heuristics in practice.

Heuristic 1: Check forbidden concepts Of the four candidate personas in our story, the most straightforward to eliminate is the Conformist (who never breaks the law): Our human intuition is that Conformist is a bad fit because Amy steals food from the cafeteria. PERSONATE's corresponding heuristic

is to eliminate all candidate personas whose forbidden concept pattern (here, Lawbreaking) appears explicitly in the story.

Heuristic 2: Reject oversimplified personas Given that Amy does steal food from the cafeteria, should we conclude that Amy is simply a thief who steals whatever she wants? Our intuition is to resist such a one-sided characterization. As justification, we might cite the fact that Amy gets the ball from Jeff by asking for it—she does not steal the ball in that case, although she could have. PERSONATE's corresponding heuristic is to consider all methods that a character employs in service of each particular goal. Then, if a character knows more methods for achieving a goal than the candidate persona does, we reject the persona as being too simplistic. (In this case, PERSONATE rejects the Thief persona, which cannot account for how Amy gets the ball from Jeff by asking for it.) Note that this deliberative process imitates how humans, having made a reflexive judgment based on means-ends precedent, later reflect on the character's other actions and adopt a more nuanced characterization.

Heuristic 3: Reward actively-avoided constraints Now we consider why we might be justified in explaining Amy's behavior as operating under a Robin Hood constraint (stealing only to help others), rather than simply doing whatever she wants. To be sure, both remaining candidate personas—Robin Hood and the Opportunist—account for Amy's previous behavior equally well. That is to say, both contain all of the means-ends strategies Amy employed during the story. Moreover, the Opportunist is arguably a simpler model, as it includes no constraints. Why conclude that Amy is deliberately avoiding theft for personal gain?

Our intuition is that Amy steals *only* to benefit others: for one thing, Amy steals food from the cafeteria for Kate, not herself. More interestingly, Amy gets the ball from Jeff by asking rather than stealing it: When she wants something for *herself*, evidently Amy does not steal it. This kind of argument requires hypothetical reasoning, thinking about the actions Amy *could* have taken and what their consequences would have been.

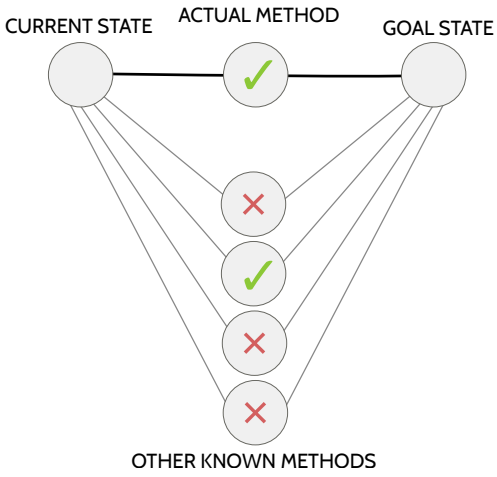

Figure 15: A schematic cartoon depicting how Heuristic 3 works in the general case. According to Heuristic 3, there is suggestive evidence that the character is deliberately conforming to a persona's constraints if the character's chosen method for achieving a goal does not violate the persona's constraints (✓), but many of the persona's alternative methods would have (×). This is exactly the behavior we would expect of a character who knows the same methods as the persona and furthermore observes the same constraints when picking actions. In such a case, we promote the persona as a close behavioral match.

PERSONATE's corresponding heuristic (Figure 15) is to check whether the character consistently chooses methods that avoid violating the persona's constraints. In detail, PERSONATE considers all the means-ends strategies the character uses in the story. For each strategy, PERSONATE considers other known methods of achieving the same goal. For each alternative method, PERSONATE checks whether that alternative method would have activated a forbidden concept pattern. If so, PERSONATE concludes that the character may have avoided the forbidden concept *on principle*. When we find that a persona has complicated constraints, and yet we note character's choices adhere to those constraints, we consider the persona to be a more falsifiable, more predictive model of behavior. We prefer personas with more "actively avoided" constraints in the story. In this case, Amy avoids the "Theft for personal gain" concept pattern when asking Jeff for the ball

rather than stealing it, so PERSONATE prefers the Robin Hood characterization over the Opportunist.

This concludes our analysis of Amy: using these general heuristics, PERSONATE decides that Amy is most like a Robin Hood character. As a final addendum, I demonstrate how these heuristics would interact in other stories and other questions. For example, consider the same story with the question "What would happen if *Teresa* wants the robot?" In a derivation similar to the one you've just seen for Amy, PERSONATE will initially consider all four personas: Conformist, Thief, Opportunist, Robin Hood. Because Teresa steals the ball for herself, Teresa exhibits the concept pattern "Lawbreaking" and "Theft for personal gain"; thus, Teresa cannot be a Conformist or a Robin Hood character (according to the first heuristic.) Instead, Teresa must either be a Thief or an Opportunist—yet none of the remaining heuristics can help us choose between them.

Heuristic 4: Prefer parsimony as a constraint of last resort At this point, we have no policy for choosing between the Thief and Opportunist characterizations of Teresa. Hence, we may reasonably decide that we lack enough evidence to choose one persona over the other. If pressed to pick only one persona, however, we might consider using a parsimony heuristic: prefer models with fewer components; that is, fewer means-ends rules and fewer concept patterns. This heuristic, potentially undesirably, would prefer characterizing Teresa as a Thief rather than an Opportunist.

To summarize this section, PERSONATE implements our human judgments using the following heuristics:

1. **Check forbidden concepts**. If a character participates in a persona's forbidden concept pattern, reject that persona.
2. **Reject oversimplified personas**. If a character knows more means to the same end than a persona, reject that persona.

3. **Reward actively avoided constraints**. If a character's unused alternatives trigger a persona's forbidden concept pattern, prefer that persona. The more avoided concepts, the better.
4. **Prefer parsimony as a constraint of last resort**. When pressed, prefer personas with fewer means-ends rules and fewer constraints.

The "reject oversimplified" and "reward actively avoided" heuristics are explicitly hypothetical: They consider alternative methods for achieving the same end and evaluate the consequences of those alternatives.

12 Models of personality circumscribe behavior

I have now described how PERSONATE identifies goal-seeking behavior in the story and uses a suite of heuristics and hypothetical reasoning capabilities to integrate those behaviors into a model of personality. Intuitively, once we have a model of the character's personality, we can use that model to predict behavior and answer questions such as "What would happen if Amy wants the robot?". The character's available methods constitute possible actions, and the character's constraints help us determine which of those actions the character will choose in a novel situation.

PERSONATE's corresponding behavior starts with the final list of candidate personas produced in the previous section. If there is only one candidate remaining, the next step is straightforward: consider every method (means-ends rule) the persona has for achieving the goal in question. If any of those methods activate the persona's forbidden concept patterns, eliminate them. Report that the character may use any of the remaining means. For example, in our story, Amy fits the Robin Hood persona best. The Robin Hood persona has two methods for acquiring the robot: stealing it, or asking for it. Stealing it would constitute "Theft for personal gain", hence PERSONATE concludes that Amy will not steal the robot. PERSONATE reports that, upon reflection, Amy will ask for the

robot instead. This represents the culmination of PERSONATE's ability to predict behavior from personality (Figure 16).

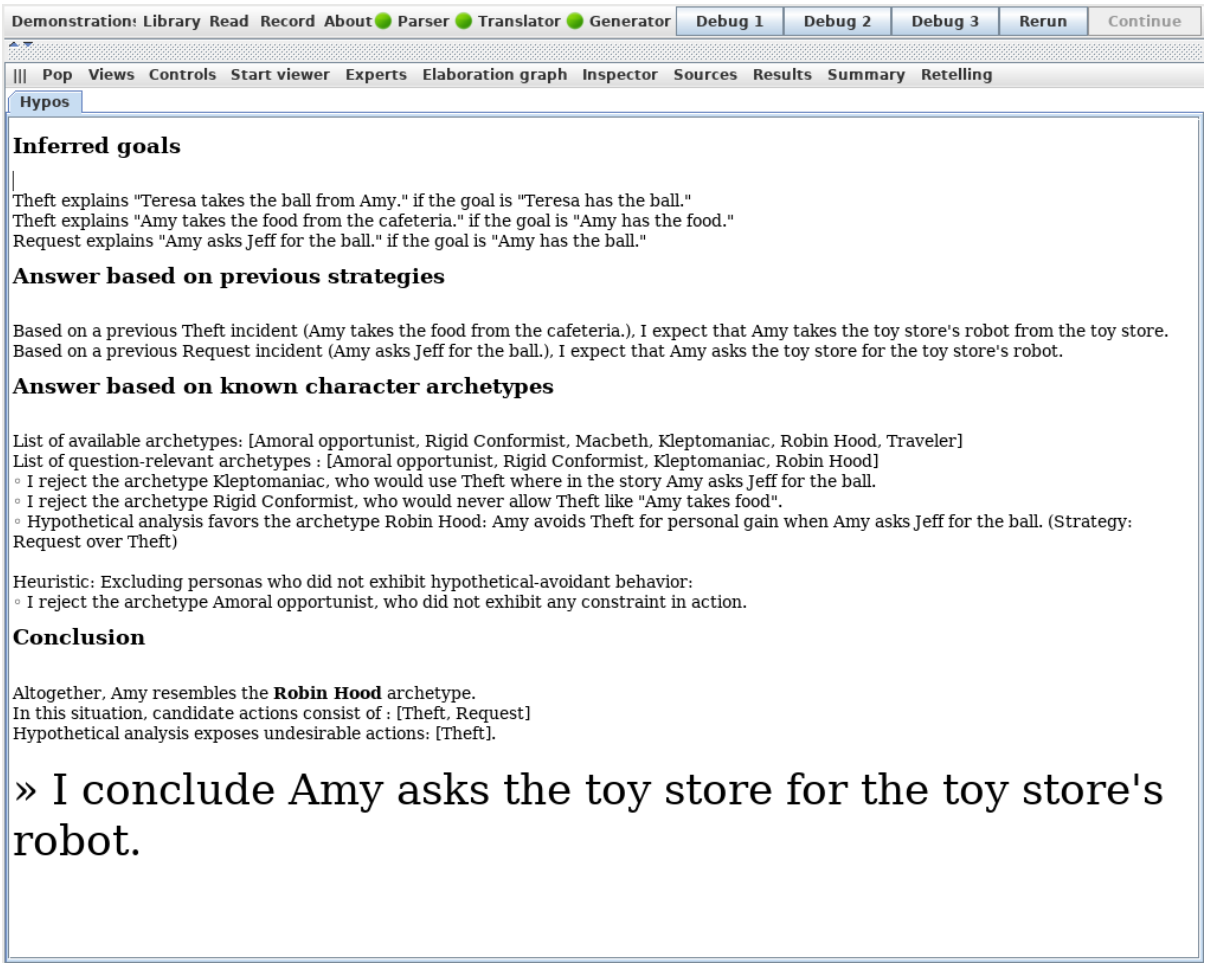

Figure 16: A complete trace of PERSONATE answering the question "What would happen if Amy wanted the robot?". PERSONATE infers character goals using means-ends rules, proposes an initial answer based on means-ends precedent, uses heuristic methods to match Amy's behavior to the Robin Hood persona, and predicts Amy's action using the Robin Hood persona as a constraint.

If there is more than one candidate persona remaining, it is less clear what PERSONATE ought to do. For our purposes, PERSONATE uses a straightforward, if cumbersome, generalization of the one-persona strategy. PERSONATE considers all possible methods available to the remaining personas and eliminates those that are constraint-violating for every remaining persona. PERSONATE reports that all remaining methods

are possible. In future work, an extension of PERSONATE could include more sophisticated methods for choosing between strategies.

Contributions

If we are to understand human intelligence and if we are to build machines whose flexibility and ingenuity rival that of human beings, we must model the many facets of human story understanding. One such facet, as I have argued in this thesis, is our ability to think in terms of possibilities, impossibilities, and constraints—to reason *hypothetically*.

Hypothetical reasoning empowers our diverse everyday activities—navigating a crowded forest, learning new concepts, solving unfamiliar problems. As you have seen, it empowers us to engage with stories more deeply, feeling suspense about what could happen, surprise at what did happen, or poignancy about what might have happened. It empowers us to reason about right and wrong, about intentions and accidents, about the case at hand and its near-miss variations. Moreover, hypothetical reasoning pervades our lives and our behavior, from the gap-filling reflexes that supply missing details to our most exalted forms of abstract cognition—imagining, comparing, adjudicating, reasoning.

In this thesis, I have demonstrated that many varieties of human-level hypothetical reasoning can be usefully grounded in story understanding.

Story-enabled hypothetical reasoning

First, I argued that varieties of hypothetical reasoning pervade human intelligence and asked how we might model them. I proposed that our story-understanding mechanisms enable us to reason hypothetically and that hypothetical reasoning enriches our understanding of stories.

A program for answering what-if questions

Second, I shed light on aspects of hypothetical reasoning by developing a suite of programs for answering different kinds of what-if questions. This led to the development of a suite of story-enabled hypothetical reasoning capabilities as described in the first half of this thesis, including:

- A **what-if question-answering program** that removes an event from a story and re-analyzes it, giving Genesis brand-new contextual understanding of stories in terms of near-miss variations. These variations can include not only variations in factual knowledge, but moreover variations in the temperament and background knowledge of the reader.
- **Presumption rules**, a new rule type that encodes fragile default knowledge. Presumption rules enable Genesis to make uncertain, provisional inferences and supply latent information, which what-if questions can expose.
- An automatic **side-by-side comparator** which summarizes the differences between two similar stories, both at a fine-grained and at a thematic level of granularity. This side-by-side comparator enables Genesis to reflect on how its own analyses change depending on circumstance, laying the groundwork for Genesis to chart its own self-knowledge in future work.

Applications of what-if questions to moral reasoning

With these tools in place, in the second half of my thesis, I showed how applications of story-enabled hypothetical reasoning could take moral and personality-based reasoning to another level. In detail, I introduced

- **Concept patterns as moral constraints**, a new use-case which links narrative patterns and moral codes.
- **Means-ends rules**, which link character actions to intentions and supply a framework of knowledge for evaluating character choices.
- **Means-ends analysis of morality**, a strategy for inferring character goals from character actions, then character values from character choices.
- **Personas**, micro-stories that represent aspects of personality pertaining to goal-directed behavior, thereby taking Genesis's mental models to another level.
- **A problem-solving approach to personality**, a theory (and associated model, PERSONATE) which treats the

problem of predicting personality analogously to how a scientist might debug a theory: by looking for hypothetical near-miss counterexamples to rule out (falsify) candidate interpretations.

- **Four heuristics for personality assignment**, a collection of crystallized principles which capture how humans intuitively evaluate and compare personalities.
- **PERSONATE**, a program that predicts how characters will respond to new situations by inferring their methods, motives, and moral constraints, then assigning a persona based the four hypothetical-reasoning heuristics. PERSONATE integrates all the computational components of this thesis to perform its various functions.

The work in this thesis is just the beginning—there is much more to explore. As we learn more about how humans think hypothetically and we build systems to test our cognitive theories, we will be able to develop engineering applications that are as broadly applicable as hypothetical reasoning itself.

Field of analysis	How would the analysis change if . . .
Case-based reasoning in medicine	. . .the patient's T-cell count were diminished?
Case-based reasoning in morality and law	. . .the suspect did not have a weapon?
Social psychology	. . .I look for situational explanations, rather than trait-based explanations?
Conflict resolution, empathy, diplomacy	. . .I read the story with this particular cultural outlook?
Moral development, self-modeling, child psychology	. . .I steal this toy when no one is looking?
Story trope analysis, personality traits, story-generation	. . .Red Riding Hood were the villain?
Literary analysis, reasoning from precedent, analogical alignment	. . .I compare this novel to *The Great Gatsby*?
Planning, naïve physics, on-the-fly safety analysis	. . .I run down the street with a full bucket of water?

Bibliography

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT press, Cambridge, MA.

Duncker, K. and Lees, L. S. (1945). On problem-solving. *Psychological monographs*, 58(5):i.

Fay, M. P. (2012). *Enabling Imagination through Story Alignment*. PhD thesis, Electrical Engineering and Computer Science Department, MIT, Cambridge, MA.

Fikes, R. E. and Nilsson, N. J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4):189–208.

Gibson, J. J. (1986). *The Ecological Approach to Visual Perception*. Psychology Press.

Goldstein, J. S. (1992). A conflict-cooperation scale for WEIS events data. *Journal of Conflict Resolution*, 36(2).

Langley, P., Magnani, L., Schunn, C., and Thagard, P. (2005). An extended theory of human problem solving. In *Proceedings of the Cognitive Science Society*, volume 27.

Lehnert, W. G. (1981). Plot units and narrative summarization. *Cognitive Science*, 5(4):293–331.

Magid, R. W., Sheskin, M., and Schulz, L. E. (2015). Imagination and the generation of new ideas. *Cognitive Development*, 34:99–110.

Marr, D. (1982). *Vision*. W.H. Freeman, San Francisco, CA.

Minsky, M. (1980). K-Lines: A theory of memory. *Cognitive science*, 4(2):117–133.

Minsky, M. (1994). Negative expertise. *International Journal of Expert Systems*, 7(1):13–19.

Minsky, M. L. (2006). *The Emotion Machine*. Simon and Schuster, New York, NY.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453.

Newell, A., Shaw, J. C., and Simon, H. A. (1959). Report on a

general problem solving program. In *IFIP congress*, volume 256, page 64. Pittsburgh, PA.

Radul, A. (2009). *Propagation networks: A flexible and expressive substrate for computation*. PhD thesis, Electrical Engineering and Computer Science Department, MIT, Cambridge, MA.

Saxe, R. (2016). Moral status of accidents. *Proceedings of the National Academy of Sciences*, 113:4555–4557.

Sloman, A. (1969). How to derive "better" from "is". *American Philosophical Quarterly*, 6(1):43–52.

Sloman, A. (2015). Impossible objects. URL: http://www.cs.bham.ac.uk/research/projects/cogaff/misc/impossible.html. Accessed: 2017-02-23.

Spelke, E. S. and Kinzler, K. D. (2007). Core knowledge. *Developmental science*, 10(1):89–96.

Winston, P. H. (1970). *Learning structural descriptions from examples*. PhD thesis, Electrical Engineering and Computer Science Department, MIT, Cambridge, MA.

Winston, P. H. (2011). The strong story hypothesis and the directed perception hypothesis. *AAAI*.

Winston, P. H. (2012a). The next 50 years: a personal view. *Biologically Inspired Cognitive Architectures*.

Winston, P. H. (2012b). The right way. *Cognitive Systems Foundation*.

Winston, P. H. and Holmes, D. (2017). The Genesis manifesto: Story understanding and human intelligence. In preparation.