The Genesis Manifesto: Story Understanding and Human Intelligence Draft of December 13, 2016

Patrick Henry Winston* and Dylan Holmes
Massachusetts Institute of Technology

Genesis supports steps toward story understanding

We now describe the Genesis Story Understanding System, whose development emerged from a desire to take sound steps toward an account of our human story understanding competence. Our purpose is to exhibit, in some detail, the representations and computations that we believe any such story-understanding system needs if it is to read text, absorb what it reads, make inferences, extract conceptual content, and answer basic *why* questions in a humanlike way.

In previous papers, Winston introduced the Genesis System, emphasized methodological steps, articulated the Strong Story Hypothesis, and saluted earlier work (**??**), particularly the pioneering work of Schank and his colleges and students, documented in numerous articles and books **????**.

In this section of this paper, we add detail via an explanation of the elements shown in figure 1. After exhibiting representative stories, we describe essential representations, comprised of classification hierarchies, case frames, and constraining connections. Then, we introduce common-sense rules and concept patterns, and we show how those inference rules and concept patterns enable Genesis to perform basic story understanding.

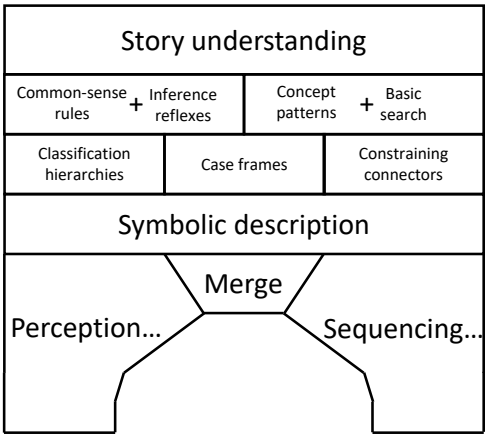

Figure 1: Genesis's story understanding rests on a small number of surprisingly simple representations and computations. At the representation level, the Genesis system requires classification hierarchies, case frames, and constraining connectors. At the computational level, much is done with inference reflexes and concept discovery via basic search. All these are enabled by what appears to be our uniquely human, merge-enable keystone ability to build symbolic descriptions, along with many other widely shared enablers.

*Corresponding author, *Email address:* phw@mit.edu *Postal Address:* 32 Vassar Sreet, Cambridge MA 01742, USA

The computations embodied in Genesis, along with the enabling foundation of inference rules and concept patterns, constitute a evolving model of human story understanding.

Genesis reads simple, concise stories

The following short summary of *Macbeth* is the anvil on which we have hammered out many ideas:

> Macbeth is a thane and Macduff is a thane. Lady Macbeth is evil and greedy. Duncan is the king, and Macbeth is Duncan's successor. Duncan is an enemy of Cawdor. Macduff is an enemy of Cawdor. Duncan is Macduff's friend. Macbeth defeated Cawdor. Duncan becomes happy because Macbeth defeated Cawdor. Witches had visions and danced. Macbeth talks with Witches. Witches make predictions. Witches astonish Macbeth. Macbeth becomes Thane of Cawdor. Duncan rewarded Macbeth because Duncan became happy. Macbeth wants to become king because Lady Macbeth persuaded Macbeth to want to become the king. Macbeth invites Duncan to dinner. Duncan goes to bed. Duncan's guards become drunk and sleep. Macbeth murders Duncan. Macbeth murders guards. Macbeth becomes king. Malcolm and Donalbain flee. Macbeth's murdering Duncan leads to Macduff's fleeing to England. Then, Macduff's fleeing to England leads to Macbeth's murdering Lady Macduff. Macbeth hallucinates at a dinner. Lady Macbeth says he hallucinates often. Everyone leaves. Macbeth's murdering Duncan leads to Lady Macbeth's becoming distraught. Lady Macbeth has bad dreams. Lady Macbeth thinks she has blood on her hands. Lady Macbeth kills herself. Burnham Wood is a forest. Burnham Wood goes to Dunsinane. Macduff's army attacks Macbeth's castle. Macduff curses Macbeth. Macbeth refuses to surrender. Macduff kills Macbeth.

Shakespeare tells us a great deal about the human condition, so as we expected, the infrastructure and much of the knowledge developed to deal with *Macbeth* transferred over to other kinds of conflict, including, for example the Estonia-Russia cyber war of 2007:

> Estonia built Estonia's computer networks. Estonia insulted Russia because Estonia relocated a war memorial. Russia wanted to harm Estonia. 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.

In both stories, harm causes anger: in one story because a person harms a person, and in the other story, because a country harms another country. Both situations are handled by a single inference rule like those described in . Similarly, in both stories, harm leads to harm, indicating revenge, in one story because people harm each other and in the other story, because countries harm each other. Both situations are handled by noting an instance of the *revenge* concept pattern described in .

Genesis uses case frames extensively

Genesis uses Boris Katz's START system to translate simple English into a collection of descriptive triples (**?**), which Genesis further processes into descriptions of story elements describing classifications, properties, relations, actions, and events.

Actions are expressed as case frames in the style of Charles Fillmore (**?**). Then, depending on the action, there are various role players, such as the agent, object, co-agent, beneficiary, instrument, or conveyance. When the action involves motion, role players may include, for example, a source, destination, and direction.

Genesis uses entities, functions, relations, and sequences as a universal substrate for expressing story elements. An entity consists of a name and a distinguishing index so as to ensure that two different entities with the same name are kept separate. In Macbeth, for example, there are three witches, `witch-1`, `witch-2`, and `witch-3`.

Functions are entities plus a subject slot filled by an entity or an entity subclass. Relations are functions plus an object slot filled by an entity or an entity subclass. Sequences are entities that hold either an ordered list or an unordered set of elements, each of which is an entity or an entity subclass.

Consider, for example, the sentence "A bird flew to a tree." When translated into a case frame, the action is *fly*, the bird is the *agent* and the tree is the *destination*.

When the case frame is expressed in the universal substrate of entities, functions, relations, and sequences, there are entities corresponding to the *bird* and the *tree*. The `tree` entity is the subject of a `to` function indicating a destination role. The `to` function is the sole element in a sequence holding a set of the role fillers. A `fly` relation connects the role-filler sequence to the subject, which by convention is taken to contain the agent role.

One way of displaying such a case frame follows. In section , we explain why we do not replace the *to* preposition with a destination-indicating symbol. Note that distinguishing indexes are not shown:

```
(relation fly
          (entity bird)
          (sequence roles (function to (entity tree))))
```

Genesis connects causes to consequents and means to actions

Genesis uses the same entity-function-relation-sequence apparatus to connect causing elements and caused elements, as in *a bird flew to a tree because a cat appeared.* In this case, there is just one causing element, the translation of *a cat appeared* and one element caused, the translation of *a bird flew to a tree*. All the causing elements are bundled together into a sequence, in this case expressing a set of just one causing element. Then, the sequence of causing elements is tied to the element caused with a *cause* relation:

```
(relation cause
          (sequence conjunction (relation appear (entity cat)))
          (relation fly
                    (entity bird)
                    (sequence roles (function to (entity tree)))))
```

Similarly, the entity-function-relation-sequence apparatus is used to connect means to actions. The following expresses the case frame for *In order to become the king, Macbeth murdered Duncan and blamed the guards*:

```
(relation means
          (sequence recipe
                    (relation murder
                              (entity macbeth)
                              (sequence roles (function object (entity duncan))))
                    (relation blamed
                              (entity macbeth)
                              (sequence roles (function object (entity guards)))))
          (function appear
                    (relation position
                              (entity macbeth)
                              (sequence roles (function object (entity king))))))
```

Thus, stories are sequences of story elements, primarily represented as case frames and various kinds of causal connections that either appear explicitly in the English or that are inferred by the inference reflexes described in section .

Genesis uses classification threads to capture class membership

Each entity and entity subclass also includes one or more classification sequences obtained by Genesis from WordNet (**?**). The following, for example, are portions of two of the classification sequences obtained from WordNet for *hawk*:

```
thing entity physical-entity object whole living-thing ... bird bird-of-prey hawk
thing entity physical-entity object whole living-thing ... adult militarist hawk
```

We keep each classification sequence separate, in *classification threads* rather than merging them into a classification tree, anticipating that we will want eventually to make use of ideas introduced by Richard Greenblatt and Lucia Vaina (**?**). They note, for example, that you might want to consider a *boy* to be first a child, and then a male, or first a male, and then a child, depending on circumstances, suggesting a need for flexibility not to be found in a fixed classification tree.

Genesis uses inference reflexes to elaborate on what is written

Genesis uses actions and other story elements, together with common sense, to build an *elaboration graph*, as shown in figure 2. Elements in yellow are established by common-sense deduction rules as indicated by black connecting lines. The story itself supplies the elements in white.

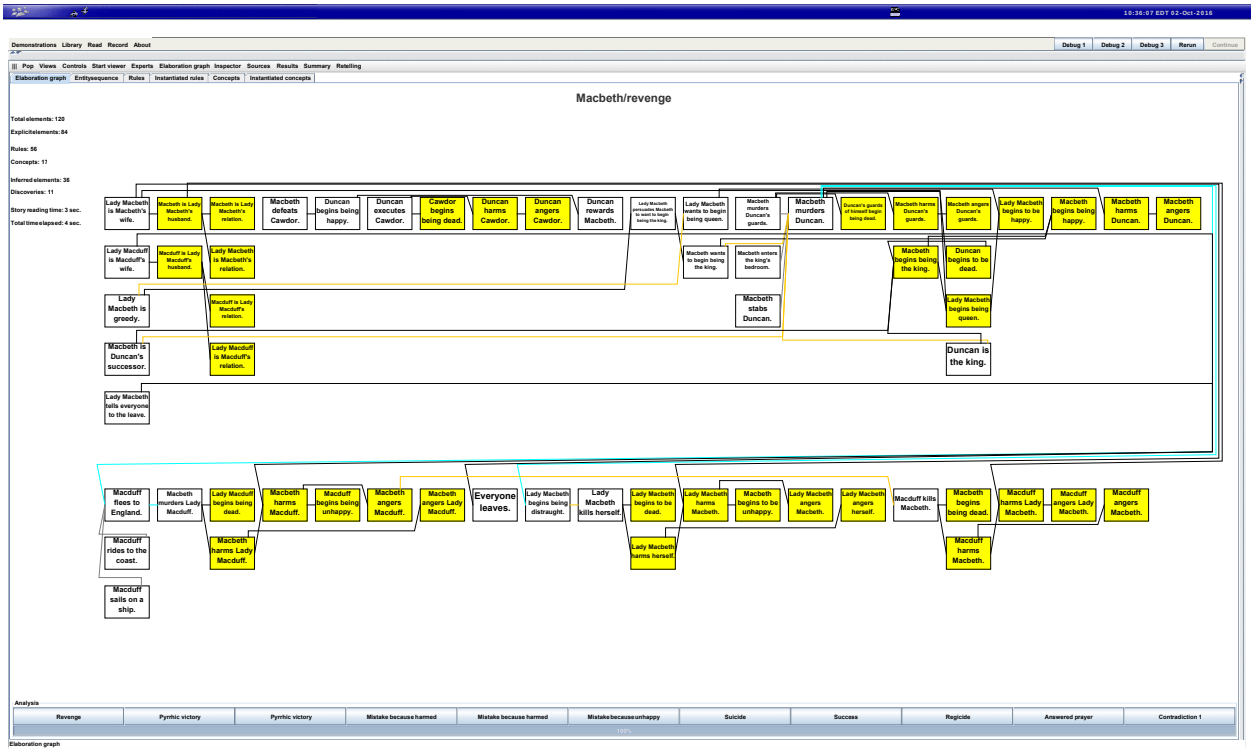

Figure 2: Genesis produces elaboration graphs as shown for a summary of Macbeth. Inference reflexes connect explicit and inferred elements of the story. Note that although the story is told as a sequence of elements, the inference reflexes form long-distance connections. (This figure is included at high resolution in the electronic version of this paper.)

We provide Genesis with deduction rules explicitly, expressing each in simple English, as in the following example:

```
If X kills y, then Y becomes dead.
```

Here is the same deduction rule, translated from the English outer language into the Genesis inner language and expressed in the entity-function-relation-sequence substrate:

```
(relation cause
          (sequence conjunction
                    (relation kill (entity x)
                    (sequence roles (function object (entity y))))))
          (function appear
                    (relation property (entity y)
                    (sequence roles (function object (entity dead))))))
```

Whenever all the antecedents of a deduction rule appear in a story, Genesis asserts the consequent. Genesis uses deduction rules extensively, but if all Genesis had were always-true deduction rules, Genesis would seem quite stupid, because human thinking is not Aristotelian logic. We have found we need many rule types to model how humans digest stories.

For example, in reading a story, we humans seek explanations, and if none is offered, we assume connections that may hold, but not with sufficient regularity to be added by deduction rules. In *Macbeth*, the story itself supplies no explicit reason why Macbeth murders Duncan and no deduction rule supplies a reason, but an *explanation rule* connects the murder to Macbeth's wanting to be king, Macbeth's being Duncan's successor, and Duncan's being king.

Thus, Genesis does *not* assert the consequent of an explanation rule whenever the antecedents appear in a story; explanation rules make connections, but only if both the antecedents and consequent have already appeared and there is no known cause for the consequent.

We express explanation rules in English using what you can view as an idiomatic use of the word *may*, as in the follow example:

```
If X is king and Y wants to be king and Y is X's successor,
then Y may murder X.
```

Another explanation rule connects anger to killing; fortunately, we do not always kill people who anger us, but it is a possibility:

```
If X angers Y, Y may kill X.
```

A *post-hoc-ergo-propter-hoc rule*, also known as a *right-together rule*, is similar to an explanation rule, but post-hoc-ergo-propter-hoc rules make a connection only if the antecedent and consequent elements appear right together in a story. Such a connection is an error in logic, but perfectly natural in story reading. We express such rules using yet another an idiomatic expression:

```
If X becomes Y and Z immediately becomes angry, then assume implication.
```

Such a rule would make a connection if a story read: "John became rich. George became angry." There would be no such connection if the story read: "John became rich. It was a sunny day. Birds sang. George became angry."

In reading a story, we may reach conclusions by way of cultural influence. Some people consider unexplained murder to indicate insanity. We capture such thinking in an *abduction rule*, using a *must* idiom:

```
If X murders Y, then X must be insane.
```

Such a rule ensures that if there is a murder in a story, then the murder is a consequence of insanity. That is, if *John murders Peter* appears in a story, then the result is as if the story explicitly included *John murders peter because John is insane*.

Note that abduction rules can specify antecedent actions, not just antecedent characteristics:

```
If X hates Y, then Y must anger X.
```

A presumption rule, like an abduction rule, assumes a particular cause, indicated by a *can be* idiom:

```
X can be greedy because X is evil.
```

With such a rule in place, if *John is greedy* appears in a story, and there is no explicit reason, or reason put in place by a deduction rule, explanation rule, or abduction rule, then the result is as if the story explicitly included *John is greedy because John is evil*.

We also have occasional need for an *ennoblement rule*, which supplies essential prerequisites to an action. Enablers appear in *enables* idioms:

```
X's having a knife enables X's stabbing Y.
```

Given such a rule, whenever a stabbing occurs, Genesis concludes that the stabbing person must have a knife, and that the having and stabbing are connected by an *enables* relation.

A seventh kind of rule, a *censor rule* prevents overuse, as when a rule might otherwise make a dead person unhappy. A *cannot* idiom identifies this kind of rule:

```
If X becomes dead, X cannot become unhappy.
```

Thus, if the antecedent of a censor rule is present, the consequent cannot be asserted by any deduction, abduction, or presumption rule.

We note that other conventions would work just as well as idioms to identify rule types, such as using explicit markers without any idiomatic use of words:

```
Deduction:  If X kills Y, then Y is dead.

Explanation:  If X angers Y, Y kills X.

Right-Together:  If X becomes Y, then Z becomes angry.
```

In summary, we have, so far, six rule types that can establish six kinds of connection, loosely considered kinds of cause, and one rule type that prevents connection. Each is expressed in an idiom, with the exact form of the idiom jointly constrained by what the front-end START parser can handle and by a desire to have all knowledge in human-readable form.

- Deduction rules, when antecedents are present, assert a conclusion and construct a causal connection between the antecedents and the conclusion.
- Explanation rules, when both antecedents and a conclusion are present, construct a explanation connection between the antecedents and the conclusion.
- Post-hoc-ergo-propter-hoc rules, when an antecedent and a conclusion are present and next to each other, construct a proximity connection between the antecedent and the conclusion.
- Abduction rules, when a conclusion is present, assert antecedents and construct an abduction connection between the antecedents and the conclusion.
- Presumption rules assert antecedents and construct a presumption connection between the antecedents and the conclusion when a conclusion is present but no explicit cause is present, nor has any deduction, explanation, or abduction produced an explicit cause.
- Enablement rules, when an consequent is present, assert antecedents that must be true for the action to occur and construct an ennoblement connection between the antecedents and the consequent.
- Censor rules prevent inappropriate application of other rules.

Each such rule type was discovered when working to model human reaction to particular stories, not through a design exercise disconnected from any specific case. Take away any rule type, and some story could not be properly understood. Accordingly, each rule type constitutes a *computational imperative*.

Each rule does its work the moment it can, and because each application is a sort of knee jerk in response to circumstance, we call each application an *inference reflex*.

> **Inference reflex:** The automatic addition to an inner story of an element or connection between elements using an inference rule.

Inference rules retain prepositional markers

Note that if all we care about is how a story element matches a rule's antecedents or consequent, and if a rule is described with the same case-marking prepositions that appear in a story, and if a rule contains antecedents that constrain what kind of things are matched, then we can defer role interpretation from read time to inference time.

Consider, for example, *Peter killed Paul with Mary*, and *Peter killed Paul with a wrench*. In the first sentence, Mary is a co-agent; in the second, the wrench is an instrument. The preposition *with* can introduce either.

Now consider these rules:

```
If W is a living-thing and X kills Y with W, then W is an accomplice.
If W is a artifact and X kills Y with W, then W is a weapon.
```

Mary is a person, and according to WordNet, a person is a `living-thing`; a wrench is a tool and a tool is an `artifact`. Accordingly, the first rule makes only Mary an accomplice and the second rule makes only the wrench a weapon. The right action can be sorted out by matching at inference-reflex time because the inference rules specify what should match.

Explicit connections also contribute to basic understanding

Of course, a story may itself exhibit a causal connection, as in an *explicit cause* statement:

```
Ducan became happy because Macbeth defeated Cawdor.
```

Alternatively, the connection may work through a chain of causes, with only the first and final elements mentioned, in a *leads to* statement:

```
Macbeth's murdering Duncan leads to Macduff's fleeing to England.
```

When working with Native American Crow creation myths, Wolfgang Yarlott noted that many leads-to statements come with an explicit indication that you will never understand the details, as in *Old Man Coyote made the world from a handful of mud and you will never understand how* (**?**). Such an *unknowable leads-to* statement is expressed using a *strangely* idiom:

```
Strangely, Macbeth's murdering Duncan leads to Macbeth's hallucinating.
```

Still another connection expresses how an event occurs. We call these *means* expressions; they appear in *in order to* idioms:

```
In order to murder Duncan, Macbeth stabbed Duncan.
```

Finally, we sometimes want to force a post-hoc-ergo-propter-hoc connection even in the absence of a right-together rule. We do so with a semicolon idiom:

```
Macduff kills Macbeth; Macduff is happy.
```

Genesis reflects on its reading, searching for concepts

Once Genesis builds the elaboration graph, Genesis uses ordinary search to find instances of concept patterns (**?**). Search instructions are supplied as leads-to statements in concept patterns we provide for Genesis's benefit. Here is the specification for *revenge*:

```
Start description of "Revenge".
X is an entity.
Y is an entity.
X's harming Y leads to Y's harming X.
```

In figure 3, Genesis notes a *Revenge* pattern because Genesis finds a path between Macbeth's harming Macduff and Macduff's harming Macbeth.

> **Concept discovery:** The affirmation that a story contains concept-specified story elements and concept-specified, potentially long-distance connections between story elements, with both kinds of requirements specified in named, storylike descriptions.

Other concept patterns specify more elaborate searches, such as the following for *Pyrrhic victory*:

```
Start description of "Pyrrhic victory".
X is an entity.
Y is an entity.
A is an action.
X's wanting A leads to X's becoming happy.
X's wanting A leads to Y's harming X.
Y harms X after X becomes happy.
```

In figure 4, Genesis notes a *Pyrrhic victory* pattern because Macbeth's wanting to be king leads not only make him happy, but also leads later to his own harm.

Most concepts, but not all, involve causal connections between entities, which may be immediate or over a long distance, as in the Pyrrhic victory example. Thus, concept identification generally requires search, which takes concept identification outside of the reach of inference rules as ordinarily used.

An optional concept pattern element, the *sometimes* element, specifies that an entity may or may not be present in a story, but if it is, it becomes part of the recognized concept. The following, for example, is another version of *Revenge*; there may or may not be hating between the participants:

```
Start description of "Revenge".
X is an entity.
Y is an entity.
X'S harming Y leads to Y's harming X.
```

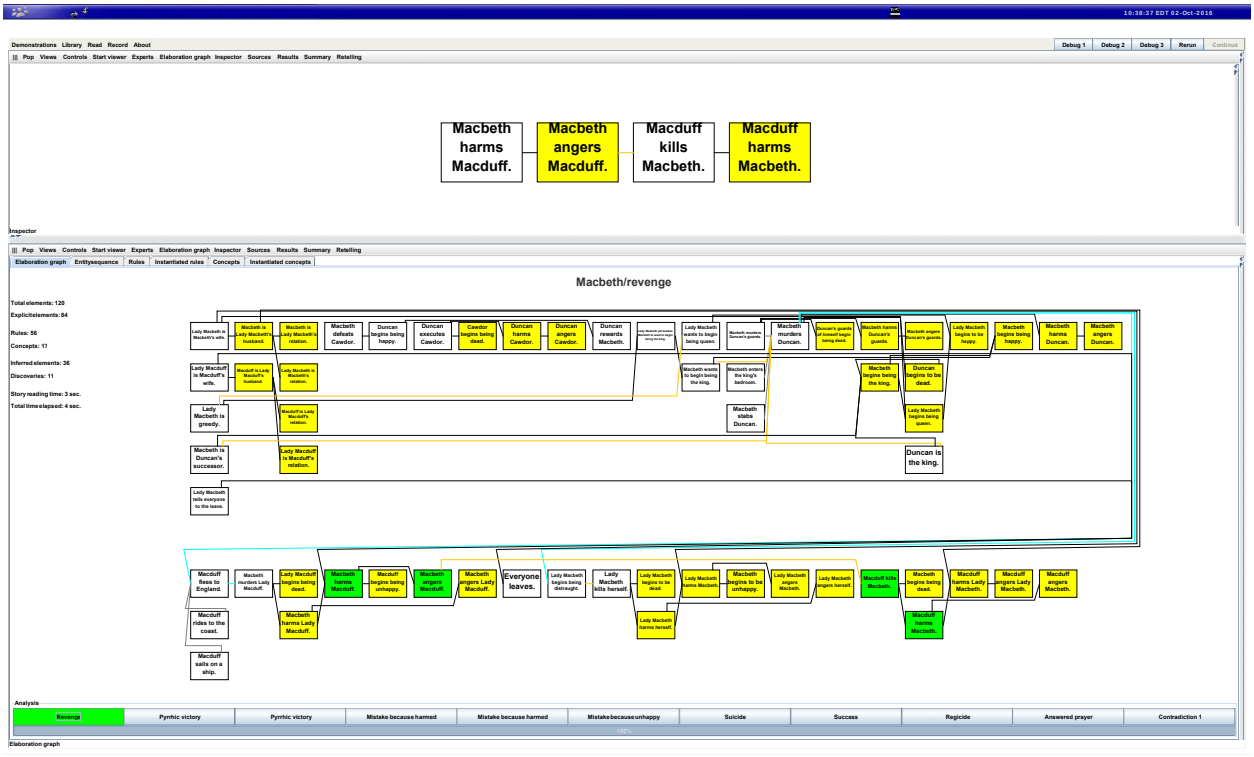

Figure 3: Genesis finds concept patterns by searching the elaboration graph. Here, Genesis highlights revenge elements in the elaboration graph. The inspector panel provides a close-up view.

```
Sometimes X hates Y.
Sometimes Y hates X.
```

Another optional concept pattern element, the *consequently* element, specifies an entity that is to be inserted back into a story as a by-product of noting a concept is present. The following emerged in work with Native American Crow creation myths (**?**):

```
Start description of "Violated belief - Medicine Man".
X is an entity.
Y is a thing.
X transforms into Y.
Consequently, X has strong medicine because X transforms into Y.
```

Concept patterns enable abstraction

Note that *revenge* is an abstraction identified with harming events. The particular kind of harming event is unimportant; it may involve a mild insult or a vicious killing. As long as two harming events are connected, with the harms going in opposite directions, there is revenge in a story. The word *revenge* or a synonym need not appear, so no system that only looks at words can reliably identify revenge.

Summary

Genesis's essential representational foundation consists of classification threads to capture classification information, case frames to express properties, relations, and actions, and various kinds causal connections to establish constraint.

Genesis uses six kinds of inference rules to make causal connections, uses five kinds of explicit causal connections, and uses censor rules that prevent assertion and connection. Genesis uses concept patterns specifying entities that must be present and entities that must be causally connected. The concept patterns may exhibit two kinds of optional elements.

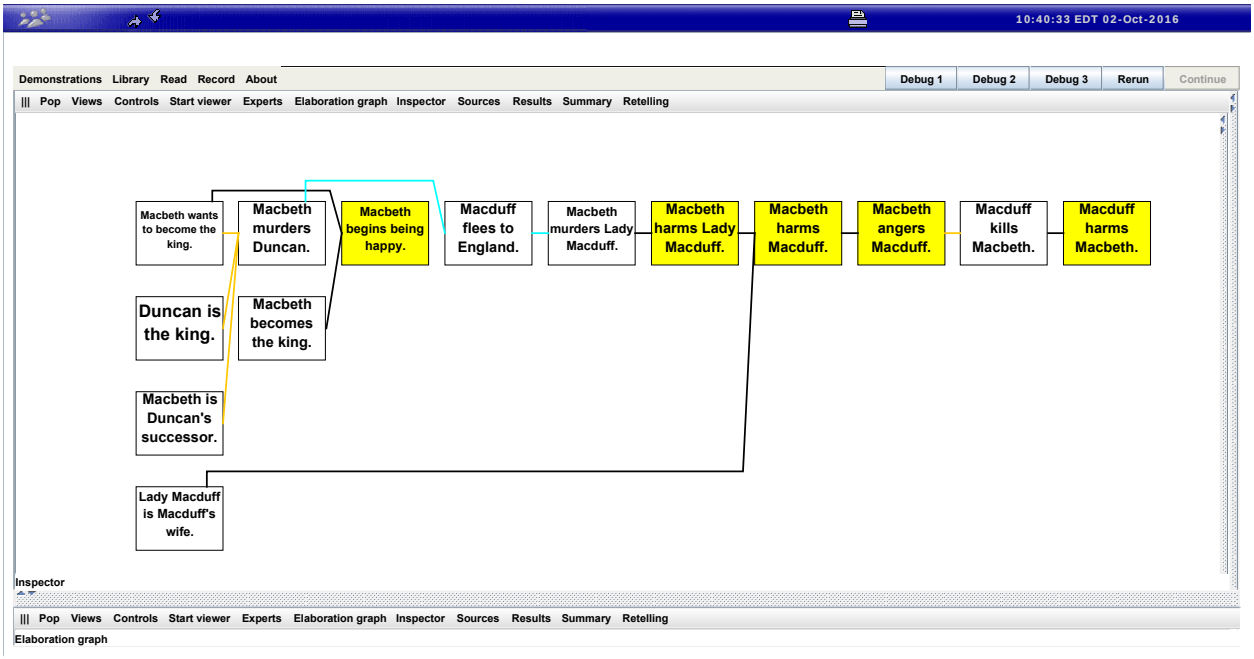

Figure 4: Genesis extracts the Pyrrhic victory elements from the full elaboration graph.

The explicit elements translated from a story into Genesis's inner language, augmented by elements produced by various kinds of inference rules and concept patterns, constitute a Genesis inner story.

Over time, more representational, inference rule, and concept pattern types will be discovered, but what has already been demonstrated suggests that the number of types needed in an account of human story understanding is not implausibly large.