Exploring Regression-Based Narrative Planning

Stephen G. Ware and Mira Fisher
Narrative Intelligence Lab, University of Kentucky
Lexington, Kentucky, 40506
sgware@cs.uky.edu, ocfi222@uky.edu

Abstract

Valid narrative plans need to meet at least two requirements: the author's goal must be satisfied by the end, and every action must make sense based on the intentions and beliefs of the characters who take them. Many narrative planners are based on progression, or forward search through the space of possible states. When reasoning about intentions and beliefs, progression can be wasteful, because either the planner needs to satisfy the author's goal first and then explain actions, which may fail, or explain actions as they are taken, which may waste effort explaining actions that are not relevant to the author's goal. We propose that regression, or backward search from goals, can address this problem. Regression ensures that every action sequence is intentional and only reasons about the agent beliefs needed for a plan to make sense.

Introduction

Narrative planning algorithms search for a sequence of actions that tell a story and that make sense for each character involved in the actions. Many search strategies have been adapted from classical planning research, including partial-order causal-link planning (Young 1999; Riedl and Young 2010; Ware and Young 2011), constraint satisfaction (Thue et al. 2016), and answer set programming (Dabral and Martens 2020; Siler and Ware 2020), to name just a few, but as in the classical planning community, many narrative planners are based on forward heuristic search though the space of states (Charles et al. 2003; Teutenberg and Porteous 2013; Ware and Young 2014; Thorne and Young 2017).

Forward search (or progression) starts at the initial state of the problem and checks which actions are possible in that state. Those actions are applied to generate the possible next states. Then any actions which are possible in those states are applied, and so on, until a valid story is discovered. Plans are constructed from start to end in order.

Narrative planning is a challenging because it places complex constraints on what action sequences are considered valid stories, and these constraints may be defined in terms of the whole sequence, or even in terms of the space of possible sequences. Consider intentionality. Narrative planners often require that every action taken by an agent contribute to a sequence of actions to achieve that agent's goal. Because goals are achieved at the end of the sequence, it is difficult to know at the beginning whether the actions an agent is taking will contribute or not.

In this paper, we propose a regression-based narrative planning algorithm that starts at the goals of the problem and works backwards to the initial state. Regression planning was described as early as 1975 (Waldinger 1975), but is rarely used in classical planners. We propose it is a good fit for narrative planners for two reasons:

1. Intentions are goal-directed, so searching backwards from goals ensures the planner does not spend effort considering actions that don't contribute to goals.
2. When we allow for a theory of mind (what x believes y believes, etc.), belief propositions can be infinitely nested. Regression can limit the planner to reasoning only about the beliefs that are relevant to the plan.

We begin with a description of the narrative planning formalism. We then present our regression algorithm and explain why it is promising. We conclude with a fully worked example to demonstrate the process.

Narrative Planning

Narrative planners have modeled many kinds of story phenomena (see Young et al. (2013) for a survey). In this paper, we build on a version of narrative planning described by Shirvani, Farrell, and Ware (2018) with these features:

- There is a system-level *author goal* that must be achieved by the end of the story.
- Agents have (possibly wrong) *beliefs* about the world and other agents. Beliefs can be arbitrarily nested, meaning there is no depth limit on the theory of mind.
- Agents have *intentions*, or personal goals. For an agent to take an action, the agent must believe the action can contribute to achieving their goal (whether or not it will).

In this section, we formally define our model of narrative planning, modifying Shirvani, Farrell, and Ware's definitions slightly to include an explicit representation of the author as an agent. We introduce our own version of the *Trea-*

sure Island problem as a running example in Figure 1, which is a simplified plot of Robert Louis Stevenson's 1883 novel.

In the story, protagonist Jim Hawkins (H) finds a map that gives the location of treasure (T) buried by Captain Flint. Antagonist Long John Silver (S) is Flint's former first mate, but does not know where the treasure is buried. Hawkins lets it be known that he has the map, prompting Silver to recruit a pirate crew and sail to Treasure Island with Hawkins. There, Hawkins digs up the treasure. Both Hawkins and Silver hope to take the treasure for themselves, and Hawkins eventually succeeds.

Formally, a narrative planning problem is a tuple $\langle C, F, G, s_0, A\rangle$. C is a set of agents, F a set of state fluents, G a goal function, s_0 the initial state, and A a set of actions that change the state. Each of these is defined in the sections below.

Agents, Fluents, and Goals

C is a set of objects that represent the *agents*, (i.e. characters) in the story. All domains include the special author agent c_A that represents the author of the story. For *Treasure Island*, $C = \{c_A, H, S\}$.

F is a finite set of state *fluents*, each with an associated finite domain D_f. Each fluent $f \in F$ is like a variable that can be assigned exactly one value from D_f at any moment in time. The proposition $f = v$ means that fluent f has value $v \in D_f$. In Figure 1, the fluent T represents the treasure's location, which can be buried on the island (B), unknown (N), dug up on the island (I), or in the possession of Hawkins (H) or Silver (S). We use the shorthand TB to mean "the treasure is buried on the island." The constant N, for unknown, is simply a value and has no special semantics here.

We define a simple logical language which allows three kinds of propositions p, expressed by this grammar:

$$p := f = v \mid b(c,p) \mid p \wedge p$$

The first kind, $f = v$, is defined above. The modal proposition $b(c,p)$ means that some non-author agent $c \in C$ believes proposition p to be true (where p can be any proposition, including another belief). We also allow conjunctions, $p \wedge p$. We assume this equivalency:

$$b(c, p \wedge q) \leftrightarrow b(c,p) \wedge b(c,q)$$

These three kinds of propositions are sufficient to describe our model, though our implementation (currently under development) includes additional features like negation, disjunction, first order quantifiers, and conditional effects.

G is a function $\forall c \in C : G(c) \rightarrow p$ that defines the *goal* proposition of every agent. $G(c_A)$ is the *author's goal*, a proposition which must be true at the end of the story. For *Treasure Island*, $G(c_A) = TH$, meaning Hawkins has the treasure. Hawkins and Silver both want the treasure; $G(H) = TH$ and $G(S) = TS$.

For simplicity, we define every agent to have exactly one goal for the whole story, though in our implementation agents can have multiple goals which can be adopted or dropped during the story.

States and Actions

A state is a data structure that can determine the truth value of any proposition. It must define a value for every fluent, plus every agent's beliefs about the values of every fluent, plus their beliefs about others' beliefs, and so on infinitely.

A *state* is a function s such that $\forall f \in F : s(f) \rightarrow v \in D_f$. For every state s, and for every agent $c \in C$, there exists exactly one state $\beta(c,s)$ that represent agent c's beliefs in s. That is, when the world is in state s, agent c believes the world is actually in state $\beta(c,s)$. To evaluate an epistemic proposition $b(c,p)$ in state s, we evaluate p in $\beta(c,s)$. For the special author agent c_A we define $\beta(c_A, s) = s$ for all states.

Note that β is a function, which implies that every agent commits to a specific (but possibly wrong) belief about every fluent. This requirement simplifies problems significantly, but means we cannot represent uncertainty (where an agent could hold one of several sets of beliefs). We have found this a useful tradeoff in practice, though others have found it valuable to model uncertainty (Mohr, Eger, and Martens 2018).

s_0 is the *initial state* of the narrative planning problem. It describes the initial values of all fluents and all initial agent beliefs.

In *Treasure Island*, the treasure is initially buried on the island, TB, and Hawkins believes this. Using Shirvani, Farrell, and Ware (2018)'s extension to the closed world assumption, we do not need to explicitly state $b(H, TB)$; this is assumed because TB is true and Hawkins has no explicitly stated belief that contradicts it. Silver does not know the treasure's location, so $b(S, TN)$ must be explicitly stated. Hawkins believes Silver does not know where the treasure is, $b(H, b(S, TN))$, but this also is assumed by the closed world assumption and does not need to be stated. It is equivalent to say that $b(S, TN)$ holds in s_0 and to say that TN holds in $\beta(S, s_0)$.

The set A is all the *actions* that could be taken in a narrative planning problem. Every action $a \in A$ has a precondition, $\text{PRE}(a)$, a proposition that must hold in the state immediately before a occurs, and an effect, $\text{EFF}(a)$, a proposition becomes true in the state immediately after a occurs.

Action preconditions and effects should not be contradictions. For example, an action may not have the precondition $TB \wedge TN$, since a fluent may only have one value at a time. This rule also applies to beliefs. For example, an action cannot have the precondition $b(S, TB) \wedge b(S, TN)$.

Actions also define $\text{CON}(a)$, a set of 0 to many *consenting agents*, who must have a reason to take the action. Not every agent involved in an action is necessarily a consenting agent. Consider the *rumor* action. Silver's beliefs are modified, so he is involved, but he is a passive participant. Only Hawkins needs a reason to take this action, so $\text{CON}(rumor) = \{H\}$. Actions that happen by accident (i.e. actions agents cannot anticipate) should have only the special author agent c_A as the consenting character, which means only the author needs a reason for it to occur.

Finally, every action a defines $\text{OBS}(a)$, a set of 0 to many *observing agents*, which are non-author agents who see the action occur and update their beliefs accordingly. Because

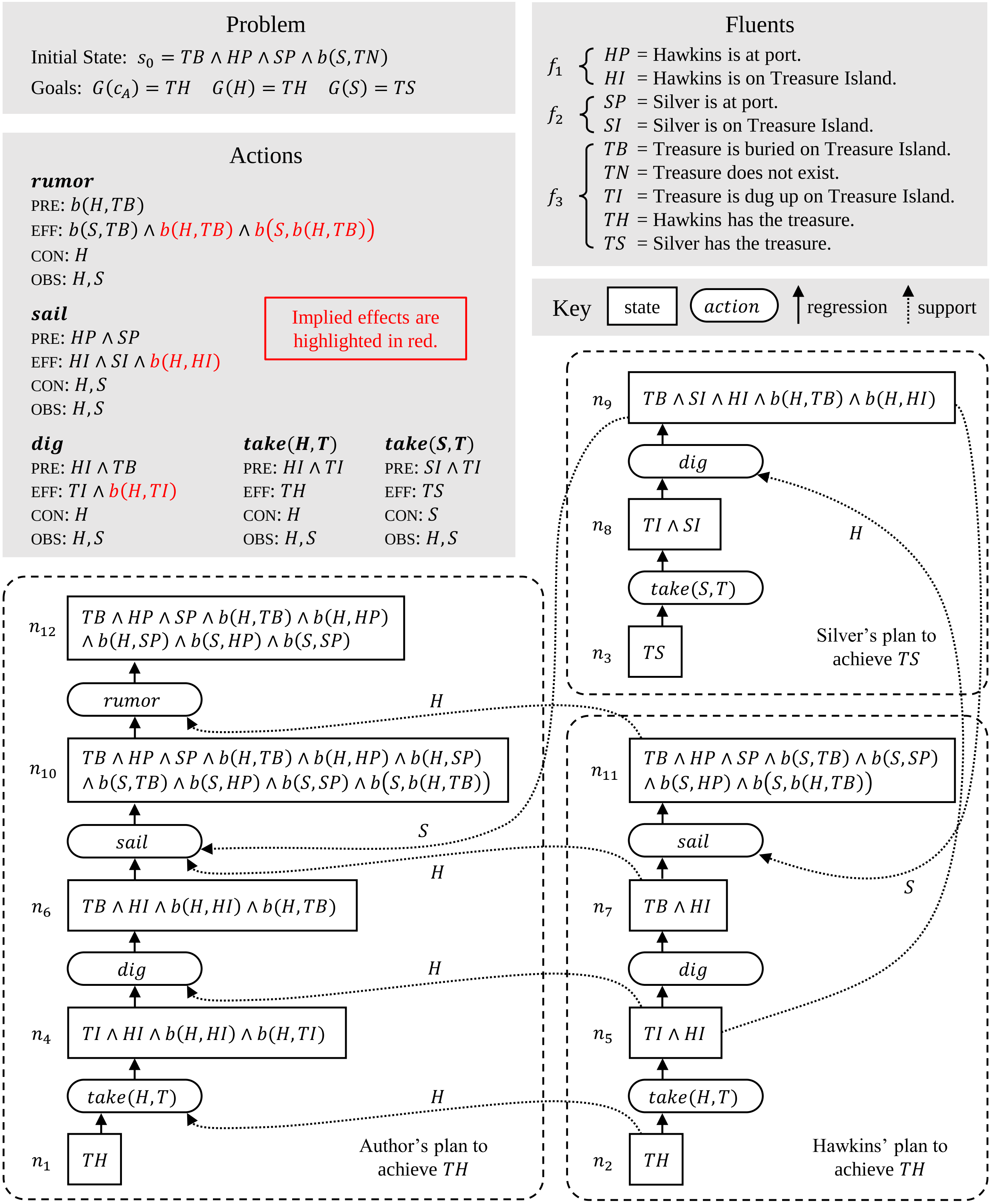

Figure 1: An example problem and example regression search space.

$\beta(c_A, s) = s$ by definition, the author effectively observes every action.

Belief propositions can be explicitly stated in preconditions and effects. Consider the $rumor$ action. Its precondition is that Hawkins believe the treasure is buried on the island, $b(H, TB)$, and its effect is that Silver now believes the treasure is buried on the island, $b(S, TB)$. See Shirvani, Ware, and Farrell (2017) for full details on how effects are imposed on states.

Actions can have implied effects which are not explicitly authored but which still result from the action. This can happen in two ways.

The first implied effects are from *surprise actions*. It is possible for agents to observe actions they do not believe are possible. For example, if Silver does not know the treasure's location (i.e. be believes $\text{PRE}(dig)$ is false), he would be surprised to see Hawkins dig it up. When a surprise action happens, agents first update their beliefs to correct wrong beliefs and then observe the effects. We accomplish this by copying any preconditions that remain unchanged into the effects of an action. Formally,

$$\forall a, p : p \in \text{PRE}(a) \wedge (p \wedge \text{EFF}(a)) \text{ is not a contradiction} \\ \rightarrow p \in \text{EFF}(a)$$

Consider the $rumor$ action. Its precondition is $b(H, TB)$, and Hawkins' belief about the treasure is not changed by the action's effect, so this action implicitly also has the effect $b(H, TB)$. This is important, because when Silver hears the rumor, he not only believes the treasure is buried on the island, he also believes Hawkins believes this.

The second kind of implied effects are from observations. When a character observes an action, they believe its effects have occurred. Consider $sail$. It has the effect that Hawkins is on the island, HI, and Hawkins observes this action, so it implicitly has the effect $b(H, HI)$. Formally:

$$\forall c, a, p : c \in \text{OBS}(a) \wedge p \in \text{EFF}(a) \rightarrow b(c, p) \in \text{EFF}(a)$$

Valid Narrative Plans

We use the function α to denote the state after a sequence of actions. In state s, let $\alpha(\{a_1, a_2, ..., a_n\}, s)$ denote the state of the world after taking those n actions from state s. α is only defined if the preconditions of those actions are satisfied immediately before they occur; that is $\text{PRE}(a_1)$ holds in s, and $\text{PRE}(a_2)$ holds in $\alpha(\{a_1\}, s)$, etc.

A sequence of actions is a valid story when it achieves the author's goal and when every action can be explained by the beliefs and intentions of the agents who take them.

In a state s, an action a_1 is *explained for* agent c iff there exists a sequence of actions $\{a_1, a_2, ..., a_n\}$ such that:

1. $\alpha(\{a_1, a_2, ..., a_n\}, \beta(c, s))$ is defined.
2. $G(c) \subseteq \alpha(\{a_1, a_2, ..., a_n\}, \beta(c, s))$.
3. All actions after a_1 are explained.
4. Unless $c = c_A$, no action has c_A as a consenting agent.
5. No strict subsequence of those actions also meets these same 5 criteria.

In other words, it makes sense for agent c to take action a_1 if and only if, according to c's beliefs about what the current state is, c can imagine a reasonable sequence of actions starting with a_1 that achieves c's goal (items 1 to 3). Item 4 means that actions intended only by the author (e.g. unexpected events or accidents) can only be explained for the author; agents cannot plan for them to happen. Item 5 expresses the idea that the plan the agent imagines should not contain unnecessary or redundant actions.

Note that the explanatory action sequence only needs to exist; it does not actually have to occur in the story. In *Treasure Island*, Silver is willing to sail to the island because he hopes to take the treasure, even if he never actually succeeds in executing this plan. This is Ware and Young's (2014) model of conflict. It is important to note that explaining an action is, itself, a planning problem. The high cost of explaining actions is one of the motivations to use regression planning, which we discussion in the following sections.

In a state s, an action a_1 is *explained* (in general) iff it is explained for every agent $c \in \text{CON}(a_1)$. In other words, an action makes sense when it makes sense for every agent who takes it.

Finally, we can define that a sequence of actions $\{a_1, a_2, ..., a_n\}$ as a valid solution to the narrative planning problem iff:

- $\alpha(\{a_1, a_2, ..., a_n\}, s_0)$ is defined.
- $G(c_A) \subseteq \alpha(\{a_1, a_2, ..., a_n\}, s_0)$.
- All actions are explained.

Progression

Progression, or forward search, begins at the initial state s_0 and generates possible futures until a state is discovered where the author's goal $G(c_A)$ holds. A classical planner is finished once this node is discovered because any path to the goal is a valid solution.

Progression is difficult for narrative planners because solutions must meet two requirements: the author's goal is achieved *and* every action is explained. Not every path to the goal is a solution. Planners like Glaive (Ware and Young 2014) first search for sequences that achieve the author's goal and then try to explain the actions in the sequence. Significant work is wasted when an action cannot be explained. Glaive's heuristic tries to account for the number of yet-unexplained actions in its calcuations, but this is only effective in some cases.

Recent work on the density of narrative planning solutions (Siler and Ware 2020) suggests it may be valuable to do progression the other way—the planner tries to explain an action immediately after taking it, and when it cannot be explained, that branch of the search can be pruned. This guarantees that any path to the author's goal is a solution, but this approach risks wasting significant work by explaining actions that are not relevant to achieving the author's goal. IMPRACTical (Teutenberg and Porteous 2013) uses an explain-first approach, but actions are explained using heuristics, so it cannot guarantee every action in the final solution will be explained.

Regression

Regression, or backward search, starts at the goal $G(c)$ and generates plans from end to start until one is found that can be executed in the initial state s_0.

Consider Hawkins' goal, TH, represented by node n_2 in Figure 1. Only the $take(H,T)$ action has the effect TH. We can regress Hawkins' goal TH over $take(H,T)$ by calculating a new proposition which, if it were true in some state, would mean that Hawkins could take that action and achieve his goal. We do this by removing the action's effects from the proposition and adding the action's preconditions. The result is n_5, whose goal proposition is $TI \wedge HI$. In other words, if we can find a state where the treasure is dug up and Hawkins is on the island, Hawkins would have a way to achieve his goal—the plan $take(H,T)$.

A nodes in the regression search space is a 2-tuple $\langle c,p \rangle$, where c is an agent and p is a proposition. In Figure 1, nodes inside the dashed boxes all have the same agent, and each node is labeled with its proposition. Nodes must be *valid* and *supported*. An edge $\langle c,p \rangle \xleftarrow{a} \langle c,q \rangle$ exists between two nodes for the same agent c and is labeled with an action a. An edge indicates that we regress proposition q over action a to get p for agent c.

Formally, a node $\langle c,p \rangle$, which was generated by the regression of $\langle c,q \rangle$ over action a, is *valid* iff:

- p is not a contradiction
- a can be taken in a state satisfying p: $\textsc{pre}(a) \subseteq p$
- $\textsc{eff}(a)$ partially satisfies q: $\exists l \in \textsc{eff}(a) : l \in p$
- q can hold after applying $\textsc{eff}(a)$ to p: $\forall r \in \textsc{eff}(a) : r \wedge q$ is not a contradiction.

Between nodes of the same agent, an edge represents a step in their plan to achieve their goal. These edges are drawn as solid arrows in Figure 1. Between nodes of distinct agents, an edge represents an expectation of consent. These edges are drawn as dotted arrows in Figure 1.

Consider node n_{10}. This node provides a valid regression for a node also owned by the author, n_6. It also contains the necessary beliefs to be *supported* by nodes n_7 and n_9.

Formally, a node $\langle c,p \rangle$ generated by expanding a node with action a is *supported* if a regression can be found for at least one node for every agent in the consenting set except for c. That is, given γ is the regression function, defined in Algorithm 1:

$$\forall c_{other} \in (\textsc{con}(a) - \{c\})(\exists \langle c_{other}, p_{other} \rangle : (b(c_{other}, \gamma(a, p_{other}) \subseteq p))$$

Algorithm

The regression of a single proposition over an action is given by the function $\gamma(a,p)$ in Algorithm 1. This function returns the simplest proposition required for the action to be acceptable for any plan continuing from that point, or it signals failure.

The regression search, given in Algorithm 2, starts with the set of nodes $\{\langle c, G(c) \rangle : c \in C\}$ (line 3). The search is an iterative expansion of the search space which proceeds by choosing a node to expand (line 5) and an action to expand it with (line 9), then choosing the consenting agent nodes to establish support for the action (line 12). All chooses are non-deterministic.

Each expansion produces nodes which describe the conditions under which the plan—the chain of actions leading back to the node $\langle c, G(c) \rangle$ for that same agent—will succeed, and which explain participation of all consenting agents for each action to be taken. The search concludes when a node is found which is both owned by the author and satisfied by the initial state (line 7).

Algorithm 1 $\gamma(a,p)$

Let a be an action, p is a proposition.
if $(\exists q : q \in \textsc{eff}(a) \wedge q \in p) \wedge (\forall r \in \textsc{eff}(a) : r \wedge p$ is not a contradiction$)$ **then**
  Let q be $\textsc{pre}(a)$.
  $\forall l \in p$: Let q be $q \wedge l$ iff $l \notin \textsc{eff}(a)$
  if q is a contradiction **then**
    return failure
  else
    return q
  end if
else
  return failure
end if

Algorithm 2 SEARCH(C, G, A, s_0)

1: C is the set of agents, G is a function of agents to agent goals, A is the set of actions, and s_0 is the initial state.
2: Let X be $\emptyset$
3: $\forall c \in C$: Let X be $X \cup \langle c, G(c) \rangle$
4: **loop**
5:   Choose a node $\langle c,p \rangle \in X$.
6:   **if** $(c = c_A) \wedge (p \subseteq s_0)$ **then**
7:     **return** the path from $\langle c,p \rangle$ to $\langle c_A, G(c_A) \rangle$
8:   **else**
9:     Choose an action $a \in A$.
10:     Let p_{new} be $\gamma(a,p)$.
11:     **for** $c_{other} \in \textsc{con}(a) : c_{other} \neq c$ **do**
12:       Choose a node $\langle c_{other}, p_{other} \rangle \in X$ such that $\gamma(a, p_{other})$ does not fail.
13:       Let p_{new} be $p_{new} \cup b(c_{other}, \gamma(a, p_{other}))$
14:     **end for**
15:     **if** $\langle c, p_{new} \rangle$ not redundant for $\langle c_A, G(c_A) \rangle$ **then**
16:       Let X be $X \cup \{\langle c, p_{new} \rangle\}$
17:     **end if**
18:   **end if**
19: **end loop**

Recall that the sequence used to explain an an action should not contain unnecessary or redundant actions (e.g. sailing back and forth to the island before digging up the treasure). For now, we define a node $\langle c,p \rangle$ to be redundant when it has an ancestor node $\langle c,q \rangle$ such that $q \subseteq p$. In other words, a plan is redundant when it ends with a sequence of actions that would also achieve the goal and that could be taken in all of the same states (and possibly more).

As an example, consider regressing node n_{12} over $rumor$. This represents the obviously redundant story:

$$\{rumor, rumor, sail, dig, take(H,T)\}$$

Hawkins spreading the rumor that he has the map twice is possible, but unnecessary, because the proposition produced by this regression would be exactly the same as the proposition for n_{12}.

Note that a node $\langle c, p\rangle$ is *not* redundant when it has an ancestor node $\langle c, q\rangle$ such that $p \subseteq q$. The proposition for node n_{12} is a strict subset of the proposition for n_{10}, but spreading the rumor is not necessarily redundant, because the plan represented by node n_{12} may apply in some states where n_{10} does not apply, e.g. any state where $b(S, TN)$.

This definition of redundant plans is not as robust as ones used in some progression planners like Glaive (Ware and Young 2014). Improving this check is an area for future work.

Worked Example

Looking at Figure 1 in more detail, we can see how the algorithm takes shape. Initially, we begin our search at the goals for each agent: Silver, Hawkins, and the author. Any of these would be effective choices for our first expansion, but we choose to expand the author's goal, n_1: Hawkins has the treasure.

We compute the regression of TH over $take(H,T)$: $\gamma(take(H,T), TH) = TI \wedge HI$. If the treasure is on the island, and so is Hawkins, we can use $take(H,T)$ to accomplish the author's goal. The resulting node is *valid*, but we must also ensure that the node is supported by finding a regression over $take(H,T)$ from a node owned by the consenting agent of $take(H,T)$, Hawkins. n_2 serves our purpose, and the regression is also $TI \wedge HI$. However, from the perspective of the author, this is our expectation of what the consenting character needs to think to take the action, as opposed to the true state of the world. Therefore, this proposition is added as a belief: $b(H, TI \wedge HI) = b(H, TI) \wedge b(H, HI)$. These are unified to get the final result. Regardless of whether he is correct, Hawkins believes that n_4 will put him in the position to take the treasure. Since he is correct, the author can accomplish that goal as well.

The next regression in the author's sequence will be the regression of the proposition for n_6 over the action dig, but we can only expand a node if we can find a regression for it *and* for a node from every consenting character as well as the current one. In this case, we must first expand n_2 (Hawkins' goal to have the treasure) to get n_5 (Hawkins' belief that he can eventually get the treasure if he is on the island and it is too) and now we have everything necessary to produce n_6 in the same way that we did for n_4. When preforming this regression, we must be sure to remove the implied effect of dig, $b(H, TI)$, as we preform the regression on the proposition in n_4 over dig.

The process continues as we consider the dig actions for the Author and Hawkins. Then prior to being able to consider the $sail$ action, which requires Silver's consent, we must expand upon Silver's plan until he has a proposition which can be regressed over the $sail$ action. We find that we can perform a regression of his goal over $take(S,T)$, and then regress over the action dig. Hawkins is the only agent who must consent to dig, so Silver must expect that Hawkins will have reason to dig. This is an instance of what Shirvani, Ware, and Farrell (2017) call anticipation. Anticipating the dig action provides an explanation for why Silver should consent to a $sail$ action, if it left the world in a state fitting n_9.

The most complicated proposition for this example is the result of the regression of $TB \wedge HI \wedge b(H, HI) \wedge b(H, TB)$ over $sail$. $sail$ requires consent from both Hawkins and Silver, so we must retrieve their regression results as well, and add their beliefs. The final proposition is given by: $\gamma(sail, TB \wedge HI \wedge b(H, HI) \wedge b(H, TB)) \wedge b(S, \gamma(sail, TB \wedge SI \wedge HI \wedge b(H, TB) \wedge b(H, HI))) \wedge b(H, \gamma(sail, TB \wedge HI)))$. Included in this, as an example of nested belief, is Silver's belief that Hawkins believes the treasure is buried—and therefore Hawkins will seek to dig up the treasure and give Silver the chance to take it. n_{11} is determined in much the same way, but only needs consideration of Hawkins' and Silver's goals, not the author's. n_{12} is expanded in the same way as the others.

At every step the algorithm compares expanded author nodes against the initial state, though we have left this step out until now. When n_{12} is compared with the initial state, we see that we have satisfied the needs of the problem—keeping in mind that, unless explicitly stated otherwise in the initial state, we assume that each agent has an accurate belief of the world.

We propose that regression planning has three major advantages:

- By searching backward from goals, we ensure action sequences are intentional. There is still a risk that search effort will be wasted exploring sequences which can never be possible, but regression addresses the two criteria problem described in the previous section. The heuristic search can prioritize sequences that can reach the initial state, and once such a sequence is found, it is guaranteed to be a solution, with no additional constraint checking required afterwards.
- With no limit imposed on the model's theory of mind, it can be difficult to know which beliefs are relevant to an agent's plan. Shirvani, Ware, and Farrell's (2017) model, on which we build, spends much effort generating all changes to beliefs that result from actions, many of which are not relevant. Regression reasons only about the beliefs which are needed to make a plan work.
- Narrative planners are often used in interactive systems where the narrative is replanned frequently. A regression plan expresses only the requirements needed to ensure it will work, so plans found this way can be easily reused in many states. Consider node n_5 in Figure 1. Hawkins has a plan to get the treasure in any state where the proposition $TI \wedge HI$ holds, which might be multiple states during the lifetime of an interactive story.

Conclusions and Future Work

The algorithm we detail here presents a method to manage intention and belief in narrative planning problems in a single search process, with no requirement to check that actions are explained after reaching the author goal. By the nature of the search space, nodes are only added to the search if the action being used for the regression is fully explained.

Our implementation of the algorithm is in development, and will be tested a suite of benchmark narrative planning problems to determine the experimental performance of the method. We also intend to develop and test heuristics to guide the regression effectively. Heuristics like the one used by Glaive are complicated because they attempt to account for the number of yet-unexplained steps in a plan. Since every node produced by our regression planner represents a valid plan, a heuristic only needs to estimate the distance between the initial state and a node's proposition.

References

Charles, F.; Lozano, M.; Mead, S.; Bisquerra, A. F.; and Cavazza, M. 2003. Planning formalisms and authoring in interactive storytelling. In *Proceedings of the conference on Technologies for Interactive Digital Storytelling and Entertainment*.

Dabral, C., and Martens, C. 2020. Generating explorable narrative spaces with answer set programming. In *Proceedings of the 16th AAAI international conference on Artificial Intelligence and Interactive Digital Entertainment*. (forthcoming).

Mohr, H.; Eger, M.; and Martens, C. 2018. Eliminating the impossible: a procedurally generated murder mystery. In *Proceedings of the 5th Experimental AI in Games workshop at the 14th AAAI international conference on Artificial Intelligence and Interactive Digital Entertainment*.

Riedl, M. O., and Young, R. M. 2010. Narrative planning: balancing plot and character. *Journal of Artificial Intelligence Research* 39(1):217–268.

Shirvani, A.; Farrell, R.; and Ware, S. G. 2018. Combining intentionality and belief: revisiting believable character plans. In *Proceedings of the 14th AAAI international conference on Artificial Intelligence and Interactive Digital Entertainment*, 222–228.

Shirvani, A.; Ware, S. G.; and Farrell, R. 2017. A possible worlds model of belief for state-space narrative planning. In *Proceedings of the 13th AAAI international conference on Artificial Intelligence and Interactive Digital Entertainment*, 101–107.

Siler, C., and Ware, S. G. 2020. A good story is one in a million: solution density in narrative generation problems. In *Proceedings of the 16th AAAI international conference on Artificial Intelligence and Interactive Digital Entertainment*. (forthcoming).

Teutenberg, J., and Porteous, J. 2013. Efficient intent-based narrative generation using multiple planning agents. In *Proceedings of the 2013 international conference on Autonomous Agents and Multiagent Systems*, 603–610.

Thorne, B. R., and Young, R. M. 2017. Generating stories that include failed actions by modeling false character beliefs. In *Proceedings of the 10th workshop on Intelligent Narrative Technologies at the 13th AAAI international conference on Artificial Intelligence and Interactive Digital Entertainment*.

Thue, D.; Schiffel, S.; Árnason, R. A.; Stefnisson, I. S.; and Steinarsson, B. 2016. Delayed roles with authorable continuity in plan-based interactive storytelling. In *Proceedings of the 9th International Conference on Interactive Digital Storytelling*, 258–269.

Waldinger, R. 1975. Achieving several goals simultaneously. Technical report, Stanford University.

Ware, S. G., and Young, R. M. 2011. CPOCL: a narrative planner supporting conflict. In *Proceedings of the 7th AAAI international conference on Artificial Intelligence and Interactive Digital Entertainment*, 97–102.

Ware, S. G., and Young, R. M. 2014. Glaive: a state-space narrative planner supporting intentionality and conflict. In *Proceedings of the 10th AAAI international conference on Artificial Intelligence and Interactive Digital Entertainment*, 80–86.

Young, R. M.; Ware, S. G.; Cassell, B. A.; and Robertson, J. 2013. Plans and planning in narrative generation: a review of plan-based approaches to the generation of story, discourse and interactivity in narratives. *Sprache und Datenverarbeitung, Special Issue on Formal and Computational Models of Narrative* 37(1-2):41–64.

Young, R. M. 1999. Notes on the use of plan structures in the creation of interactive plot. In *Proceedings of the AAAI Fall Symposium on Narrative Intelligence*, 164–167.