Understanding Restaurant Stories Using an ASP Theory of Intentions (Extended Abstract)

Daniela Inclezan[1], Qinglin Zhang[1], Marcello Balduccini[2], and Ankush Israney[2]

1 **Miami University, Oxford OH, USA**
inclezd,zhangq7@miamioh.edu
2 **Drexel University, Philadelphia PA, USA**
mb3368,avi26@drexel.edu

Abstract

The paper describes an application of logic programming to story understanding. Substantial work in this direction has been done by Erik Mueller, who focused on texts about stereotypical activities (or scripts), in particular restaurant stories. His system performed well, but could not understand texts describing exceptional scenarios. We propose addressing this problem by using a theory of intentions developed by Blount, Gelfond, and Balduccini. We present a methodology in which we model scripts as activities and employ the concept of an intentional agent to reason about both normal and exceptional scenarios.

1998 ACM Subject Classification I.2.4 Knowledge Representation Formalisms and Methods

Keywords and phrases answer set programming, story understanding, theory of intentions

Digital Object Identifier 10.4230/OASIcs.ICLP TCs.2017.

Overview

This paper describes an application of Answer Set Prolog [3] and its extension [1] to the understanding of narratives. According to Schank and Abelson [7], stories frequently narrate episodes related to *stereotypical activities — sequences of actions normally performed in a certain order by one or more actors, according to cultural conventions.* An example of a stereotypical activity is dining in a restaurant with table service. A story mentioning a stereotypical activity is not required to state explicitly all of the actions that are part of it, as it is assumed that readers are capable of filling in the blanks with their own commonsense knowledge about the activity [7]. Consider, for instance, the following narrative:

▶ **Example 1.** *Nicole went to a vegetarian restaurant. She ordered lentil soup. The waitress set the soup in the middle of the table. Nicole enjoyed the soup. She left the restaurant.*

Norms indicate that customers do not seat themselves when there is table service, but rather wait to be seated by a waiter; they are also expected to pay for their meal. Readers are supposed to know these conventions, and thus such information is missing from the text.

Schank and Abelson [7] introduced the concept of a *script* to model stereotypical activities: a *fixed* sequence of actions that are *always* executed in a specific order. Following these ideas, Erik Mueller conducted substantial work on narratives about stereotypical activities. He focused on restaurant stories [5] and news about terrorist incidents [4]. In the former, Mueller developed a system that can take as an input a text about a restaurant episode, process it using information extraction techniques, and demonstrate an understanding of the narrative by answering questions whose answers are not necessarily explicitly stated in

Technical Communications of the 33rd International Conference on Logic Programming (ICLP 2017).
Editors: Ricardo Rocha and Tran Cao Son; Article No. ; pp. :1–:3

Open Access Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

the text. The system had a good accuracy but the rigidity of scripts did not allow for the correct processing of scenarios describing exceptions (e.g., waiter bringing a wrong dish). To be able to handle such scenarios, all possible exceptions of a script would have to be foreseen and encoded as new scripts by a knowledge engineer, which is an important hurdle.

In this paper, we propose a new representation methodology and reasoning approach, which makes it possible to answer, in both normal and exception scenarios, questions about events that did or did not take place. We overcome limitations in Mueller's work by abandoning the rigid script-based approach. Instead, we view characters in stories about stereotypical activities (e.g., the customer, waiter, and cook in a restaurant scenario), as BDI agents that *intend* to perform some actions in order to achieve certain goals, but may not always need to/ be able to perform these actions as soon as intended. It was instrumental for our purpose to use a *theory of intentions* developed by Blount *et al.* [2] that introduces the concept of an *activity* — a triple consisting of the activity's name, a goal, and the plan aimed at achieving it. An activity (or rather its plan) may be divided into sub-activities with their own sub-goals. In our work, each character role that is relevant to a stereotypical activity had its own activity. For instance, for the customer role in a restaurant episode, we created an activity named $c\_act(C, R, W, F)$, read as "customer C goes to restaurant R where s/he communicates to waiter W an order for food F." The plan for this activity is the sequence

$$[\ enter(C, R),\ lead\_to(W, C, \mathtt{t}),\ sit(C),\ c\_subact\_1(C, F, W),\ eat(C, F),\ c\_subact\_2(C, W),\ stand\_up(C),\ move(C, \mathtt{t}, \mathtt{entrance}),\ leave(C)\ ]$$

in which `t` stands for the customer's table. The activity's goal is $satiated\_and\_out(C)$. Modeling the customer's activity as a nested one with sub-activities allowed reasoning about a larger number of exceptional scenarios compared to its formalization as a flat activity, for instance Example 2. We introduced two sub-activities: $c\_subact\_1(C, F, W)$ – "C consults the menu and communicates an order for food F to W," and $c\_subact\_2(C, W)$ – "C asks W for the bill and pays for it." In Example 2, Nicole does not execute the actions in c_subact_2 as the goal of this sub-activity, being done with payment, is already satisfied as the meal is on the house. Instead, she performs the next action in the overall activity: $stand\_up$.

▶ **Example 2** (Serendipity). *Nicole went to a vegetarian restaurant. She ordered lentil soup. When the waitress brought her the soup, she told her that it was on the house.*

Activities were encoded in Answer Set Prolog via rules like:

$activity(c\_act(C, R, W, F)) \leftarrow customer(C), restaurant(R), waiter(W), food(F).$
$comp(c\_act(C, R, W, F),\ 1,\ enter(C, R)) \leftarrow activity(c\_act(C, R, W, F)).\quad \ldots$
$length(c\_act(C, R, W, F),\ 9) \leftarrow activity(c\_act(C, R, W, F)).$
$goal(c\_act(C, R, W, F),\ satiated\_and\_out(C)) \leftarrow activity(c\_act(C, R, W, F)).$

Blount *et al.* also introduced an architecture ($\mathcal{AIA}$) of an *intentional agent*, an agent that obeys his intentions. According to $\mathcal{AIA}$, at each time step, the agent observes the world, explains observations incompatible with its expectations (diagnosis), and determines what action to execute next (planning). $\mathcal{AIA}$ models the control strategy of an agent capable of reasoning about a wide variety of scenarios, including the serendipitous achievement of its goal by exogenous actions as in Example 2 or the realization that an ongoing activity is futile. In contrast with $\mathcal{AIA}$, which encodes an agent's reasoning process about its own goals, intentions, and ways to achieve them, we represented the reasoning process of a (cautious) reader that learns about the actions of an intentional agent from a narrative. For instance, while an intelligent agent creates or selects its own activity to achieve a goal, in a narrative context, the reader learns about the activity that was selected by the agent from the text. As a consequence, we adapted parts of the $\mathcal{AIA}$ architecture to suit our purposes.

Stories about stereotypical activities do not mention all actions that occur, as they rely on the reader's background knowledge. As a result, the reader needs to fill the story time line with new time points (and thus construct what we call a *reasoning time line*) to accommodate physical and mental actions not mentioned in the text. We complemented the reasoning module adapted from $\mathcal{ALA}$ with reasoning rules below, in which we denote story vs. reasoning time steps by predicates *story_step* and *step*, resp., and introduce predicate *map*(`s`,`i`) to say that story step `s` is mapped into reasoning time step `i`:

$1\{map(S,I) : step(I)\}1 \quad \leftarrow \quad story\_step(S).$

$\neg map(S,I) \quad \leftarrow \quad map(S_1,I_1),\ S < S_1,\ I \geq I_1,\ story\_step(S),\ step(I).$

Observations about the occurrence of actions and values of fluents mentioned in the text, recorded using predicates *st_hpd* and *st_obs*, are translated into observations on the reasoning time line via rules of the type:

$hpd(A,V,I) \ \leftarrow \ st\_hpd(A,V,S),\ map(S,I).$

A reader may be asked questions about the story. We support yes/no, when, who, and where questions related to events. A question is represented by an atom *question*(q), where q is a term encoding the question, e.g., *query_occur*(A) ("did action A occur?"), *query_when*(A) ("when did A occur?"). Answers are encoded by atoms *answer*(q,a), where a is the answer. For example, $answer(occur(pay(nicole,b)),yes)$ states that the answer to question "Did Nicole pay the bill?" is yes. A positive answer about the occurrence of a specific event is encoded by the rule:

$answer(query\_occur(A),\ yes) \ \leftarrow \ physical\_action(A),\ step(I),\ occurs(A,I).$

Answering a definite "no" requires ensuring that the action did not happen at *any* step:

$maybe(A) \ \leftarrow \ physical\_action(A),\ step(I),\ \text{not } \neg occurs(A,\ I).$

$answer(query\_occur(A),\ no) \ \leftarrow \ physical\_action(A),\ step(I),$
$\text{not } answer(query\_occur(A),\ yes),\ \text{not } maybe(A).$

While we exemplified and tested our methodology on restaurant scenarios, our approach is equally applicable to other stereotypical activities. The main task for a new stereotypical activity is defining the different activities, including goals, for each relevant character role. Part of this process can be automated by starting from a rigid and centralized script learned in an unsupervised manner (e.g., [6]). Determining (sub-)goals and splitting activities into sub-activities is a more challenging problem, which deserves substantial attention.

References

1 Marcello Balduccini and Michael Gelfond. Logic Programs with Consistency-Restoring Rules. In *Proceedings of Commonsense-03*, pages 9–18. AAAI Press, 2003.

2 Justin Blount, Michael Gelfond, and Marcello Balduccini. A theory of intentions for intelligent agents. In *Proceedings of LPNMR 2015*, pages 134–142, 2015.

3 Michael Gelfond and Vladimir Lifschitz. Classical Negation in Logic Programs and Disjunctive Databases. *New Generation Computing*, 9(3/4):365–386, 1991.

4 Erik T. Mueller. Understanding script-based stories using commonsense reasoning. *Cognitive Systems Research*, 5(4):307–340, 2004.

5 Erik T. Mueller. Modelling space and time in narratives about restaurants. *Literary and Linguistic Computing*, 22(1):67–84, 2007.

6 Michaela Regneri, Alexander Koller, and Manfred Pinkal. Learning script knowledge with web experiments. In *Proceedings of the ACL '10*, pages 979–988, 2010.

7 R. C. Schank and R. P. Abelson. *Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures*. Lawrence Erlbaum, 1977.