An Investigation of Actions, Change and Space

Paulo E. Santos*
Elect. Eng. Dep.
FEI, São Paulo, Brazil.
E-mail: `psantos@fei.edu.br`

Pedro Cabalar†
Department of Computer Science,
University of Corunna, Spain.
E-mail: `cabalar@udc.es`

Abstract

This work investigates the spatial knowledge and automated solution of a domain composed of non-trivial objects such as *strings* and *holed objects*.

Formalising the Fisherman's Folly Puzzle

The work presented in (Cabalar and Santos 2011; Santos and Cabalar 2008) has concentrated on the formalisation and automated solution of the Fisherman's Folly puzzle (shown in Figure 3), whose goal is to release a ring from an entanglement of objects, maintaining the objects' physical integrity. The elements of the Fisherman's Folly puzzle are a holed post ($Post$) fixed to a wooden base ($Base$), a string (Str), a ring ($Ring$), a pair of spheres ($Sphere1, Sphere2$) and a pair of disks ($Disk1, Disk2$). The spheres can be moved along the string, whereas the disks are fixed at each string endpoint. The string passes through the post's hole in a way that one sphere and one disk remain on each side of the post. The spheres are larger than the post's hole, thus the string cannot be separated from the post without cutting either the post, or the string, or destroying one of the spheres. The disks and the ring can pass through the post's hole.

(a) S_0 (b) Goal (c) Diagram of S_0

Figure 1: A spatial puzzle: the Fisherman's Folly.

In the initial state (Fig. 1(a)) the post is in the middle of the ring, which in its turn is supported on the post's base. The goal of this puzzle is to find a sequence of (non-destructive) transformations that, when applied on the domain objects, frees the ring from the other objects, regardless their final configuration. Fig. 3(b) shows one possible goal state.

A simple planning system capable of finding a solution to the Fisherman's Folly puzzle was reworked in (Cabalar and Santos 2011), where the states of the puzzle were represented as lists containing the sections of a long object between hole crossings (Fig. 1(c)). In (Santos and Cabalar 2008), a mereotopological representation of the domain objects was presented. The work in (Cabalar and Santos 2011) developed a representation of the puzzle actions in a Situation Calculus (Reiter 2002) framework developed in Quantified Equilibrium Logic (Pearce and Valverde 2008), where we were interested in a solution that was tolerant to elaborations, showing its applicability to other similar puzzles.

In a nutshell the formalisation of the Fisherman's Folly puzzle presented in (Cabalar and Santos 2011) relies on the definition of a list data structure named $chain(X)$. This data structure represents the sequence of all hole crossings on a long object X, when traversing X from its negative tip to its positive one. For instance, the state shown in Fig. 1(c) is represented by the following two chains: $chain(P) = [Ring^+]$ (for the post (P) object) and $chain(Str) = [Sphere1^+, PostH^+, Sphere2^+]$ (for the string (Str) object). The former represents that the long object $Post$ (P) crosses the ring hole and the latter states that the string crosses the hole on the sphere 1, the post hole and the hole on the sphere 2, respectively. Note that, for brevity, only the outgoing hole faces are shown, following the direction negative to positive tip.

An action $pass$ was defined to represent the movements of puzzle objects: $pass(Obj, Hole^i)$ represents the action of passing an object Obj towards the i face of a hole $Hole$ ($i \in \{+, -\}$)). The effects of $pass$ either add or delete hole crossings from the $chain$ on which it is applied. Using these definitions, a solution to the Fisherman's Folly puzzle can be represented by the sequence of chains shown on Fig. 2.

The main contribution of (Cabalar and Santos 2011) was showing how to work with the puzzle at two levels: one using a logical representation which is more flexible and allows talking about different fluents and their inertial behaviour, and a second one based on the list structure $chain(X)$ which is less flexible but more practical for planning purposes, while still correct with respect to the logical

*Paulo E. Santos acknowledges financial support from FAPESP grant 2012/04089-3, and CNPq "bolsa de produtividade em pesquisa" grant 303331/2011-9.

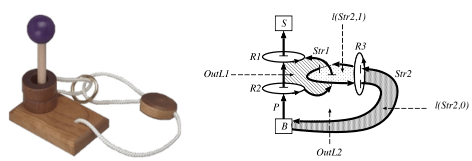
†Pedro Cabalar was partially supported by Spanish MEC project TIN2009-14562-C05-04.

state	$chain(P)$	$chain(Str)$
S_0	$[Ring^+]$	$[Sphere1^+, PostH^+, Sphere2^+]$
s_1	$[Ring^+]$	$[Sphere1^+, PostH^+, Sphere2^+, PostH^-]$
s_2	$[\,]$	$[Sphere1^+, Ring^-, PostH^+, Ring^+, Ring^-, Sphere2^+, Ring^+, PostH^-]$
s_3	$[\,]$	$[Sphere1^+, Ring^-, PostH^+, Sphere2^+, PostH^-, Ring^+]$
s_4	$[\,]$	$[Sphere1^+, PostH^+, Ring^-, Sphere2^+, Ring^+, PostH^-]$
s_5	$[\,]$	$[Sphere1^+, PostH^+, Sphere2^+, PostH^-]$

Figure 2: A formal solution for the Fisherman's puzzle.

representation. Still, from the planning perspective, the prototype planner considered in that paper consisted in a simple blind search, iterative deepening strategy, capable of solving the five steps of Fisherman's Folly, but unable to deal in a reasonable time with other similar problems with solutions of a dozen steps.

Solving puzzles with string loops



(a) Puzzle (b) Diagram

Figure 3: Easy does it.

In none of our previous formalisations, however, the notion of loop was taken into account. This is due to the fact that the object "loop" was not relevant to the solution of the Fisherman's Folly puzzle or its close relatives. The question that naturally follows from this is whether these previous solutions are tolerant to this elaboration. Our current investigation aims to extend the framework presented in (Cabalar and Santos 2011) towards the automated solution of puzzles that involve reasoning about string loops, such as the Easy-does-it puzzle (Fig. 3(a)). In the Easy-does-it domain, loops in the string form holes that are represented with two faces + and -, as done previously for holes in rigid objects. Therefore, in this case, actions on a string create a distinct kind object that accept a distinct set of actions from those that can be applied on the original string itself.

Interesting issues in planning

The investigation of string puzzles, such as those considered in this paper, brings up a number of issues of interest not only to the knowledge representation field, but also to researchers in the planning community. Below we summarise a few of the most important open issues in this respect.

- **Complexity:** how an efficient planner can deal with the large amount of possible outcomes (infinite in fact) of applying a deformation action on a piece of string, let alone the possibility of tying knots or tight entangling the string with the other objects.
- **Intentionality:** The complexity issue mentioned above makes us wonder why a human playing with such puzzles is not baffled by the existence of infinite chains of actions and outcomes of actions (and why most of (or some of) us do not get bored when trying to solve it). There seems to be an *intentionality* behind the actions people choose to apply on the domain. This intentionality is not only related to *reducing* the complexity of the entanglement, but also on a prediction of the possible states that the action would lead to, and how they related to the solution. These *intentional heuristics* are somewhat very elusive to define.
- **Learning domain constraints:** In our previous work, the domain constraints were handwritten (i.e. we had to explicitly represent the cases where an object could not pass through a hole). We believe that these constraints could be automatically learnt within the reasoning procedures to solve the puzzle.
- **Real application domains:** the importance of reasoning about strings and holes goes beyond pure theoretical interest, embracing application domains such as: *cabling*, where reasoning about flexible and perforated objects is needed for optimising the spatial arrangements of networks of cables; *autonomous maintenance of mechanical machines*, from which devices have to be extracted without damaging the surrounding parts; *robotic surgery*, where autonomous machines have to deal (and plan how to handle) sutures in situations involving very distinct and delicate structures.
- **Obtaining specialist knowledge:** related to the application domains cited in the previous item is the issue of how to obtain specialist knowledge, and how this could interact with the planning task. This is particularly relevant in the (semi)autonomous robotic surgery domain, where a human surgeon can interfere, guide or prohibit certain actions as planned by the robotic agent.
- **Proving physical feasibility:** last but not least, how to guarantee physical feasibility of the sequence of actions obtained by the planner (without considering an entire 3D model of the domain in the planning process) is a challenging open issue.

Conclusion

In this paper we discussed the challenging problem of formally describing flexible objects such as strings. We outlined an initial formalisation based on previous investigations on an automated solution for spatial puzzles. There is, however, still a long way to go before deploying this initial formalisation in a real application setting.

References

Cabalar, P., and Santos, P. E. 2011. Formalising the fisherman's folly puzzle. *Artif. Intell.* 175(1):346–377.

Pearce, D., and Valverde, A. 2008. Quantified equilibrium logic and foundations for answer set programs. In *Proc. of the 24th ICLP*, volume 5366 of *LNCS*. Springer. 546–560.

Reiter, R. 2002. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. U.S.: MIT Press.

Santos, P. E., and Cabalar, P. 2008. The space within fisherman's folly: Playing with a puzzle in mereotopology. *Spatial Cognition & Computation* 8(1-2):47–64.