Unified Perception-Prediction Model for Cognitive Agents

Sergio Arzola[1] and Claudia Zepeda[1]

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
sinrotulos@gmail.com, czepedac@gmail.com

Abstract. In this work, we adapt the unified perception-prediction model for cognitive agents in order to solve related perception and prediction problems. Furthermore, we present an approach based in logic programming. The Unified Perception-Prediction Model is based on how the brain works according to neuroscience research.
Key words Cognitive science, Unified perception-prediction model, artificial intelligence, logic programming, stable semantics.

1 Introduction

The research about how the brain works has increased over the last years. Neuroscience started to provide mechanistic biological oriented explanation for every aspect of behavior, such as psychology, economics, anthropology and so on. In particular, we will focus on perception and prediction studies at an abstract level in order to explain the Unified Perception-Prediction Model [7]. Though this model has been used for Context Aware Text Recognition, we will use this model in order to model intelligent agents. The main purpose of using the UPP Model as reference is to create a better understanding of the physical world for the agent for acting properly. With this model the agent can be capable of make inferences of the model it has of the world with perceptions and predictions. In addition, we present a logic approach of this model for intelligent agents using the stable semantics [4].
In [3] it is remarked the importance of perception in the brain as well as prediction in order to interact within the physical world. There exists an infinite loop of interaction between perception and prediction because the brain can not perceive what it does not expect to perceive as well as the brain can not predict what it does not have information about [3].

2 Background

In this section we summarize some basic concepts and definitions used to understand this paper.

2.1 Logic programs

A *signature* $\mathcal{L}$ is a finite set of elements that we call *atoms*, or *propositional symbols*. The language of a propositional logic has an alphabet consisting of *proposition symbols*: $p_0, p_1, \ldots$; *connectives*: $\wedge$, $\vee$, $\leftarrow$, $\neg$; and *auxiliary symbols*: (,). Where $\wedge$, $\vee$, $\leftarrow$ are 2-place connectives and $\neg$ is a 1-place connective. Formulas are built up as usual in logic. A *literal* is either an atom a, called *positive literal*; or the negation of an atom $\neg a$, called *negative literal.* The formula $F \equiv G$ is an abbreviation for $(F \leftarrow G) \wedge (G \leftarrow F)$. A *clause* is a formula of the form $H \leftarrow B$ (also written as $B \rightarrow H$), where H and B, arbitrary formulas in principle, are known as the *head* and *body* of the clause respectively. The body of a clause could be empty, in which case the clause is known as a *fact* and can be denoted just by: $H \leftarrow$. In the case when the head of a clause is empty, the clause is called a *constraint* and is denoted by: $\leftarrow B$. A *normal* clause is a clause of the form $H \leftarrow \mathcal{B}^+ \cup \neg\mathcal{B}^-$ where H consists of one atom, $\mathcal{B}^+$ is a conjunction of atoms $b_1 \wedge b_2 \wedge \ldots \wedge b_n$, and $\neg\mathcal{B}^-$ is a conjunction of negated atoms $\neg b_{n+1} \wedge \neg b_{n+2} \wedge \ldots \wedge \neg b_m$. $\mathcal{B}^+$, and $\mathcal{B}^-$ could be empty sets of atoms. A finite set of normal clauses P is a *normal program.*

Finally, we define $RED(P, M) = \{H \leftarrow B^+, \neg(B^- \cap M) \mid H \leftarrow B^+, \neg B^- \in P\}$. For any program P, the positive part of P, denoted by $POS(P)$ is the program consisting exclusively of those rules in P that do not have negated literals.

2.2 Stable semantics

From now on, we assume that the reader is familiar with the notion of classical minimal model [5]. We give the definitions of the stable semantics for normal programs.

Definition 1. *[6] Let P be a normal program and let $M \subseteq \mathcal{L}_P$. Let us put $P^M = POS(RED(P, M))$, then we say that M is a* stable model *of P if M is a minimal classical model of P^M.*

3 The Unified Perception-Prediction Model for Cognitive Agents

In this section we establish a cognitive agent model based on the UPP Model.

This model consists in three main parts: the Perception part, the Prediction part, and World's Model.
The way that this model works is analogous as the brain works [3]. However, for agent programming we consider the following assumptions:
The World's Model must have a background knowledge of the problem, which is intended to solve.
Perceptions must be in the same terms in which data is stored.

Predictions must be made based on the previous knowledge the agent has or the current perceptions.

The Perception part receives signals of the world through its sensors. The Prediction part makes inferences from information stored in the Worlds Model part or Perception. Both parts store the information in the Worlds Model part, which is always being updated.
It is easy to see the loop between Perception and Prediction as well as the World's Model as a storage of both parts.

Perceptions depend upon a prior belief [3], which is located in the world's model. Thus, the process start from the inside, the model brings predictions based on the world's model, in this way the agent can predict what perceptions the agent should be receiving. These perceptions are compared with the signals sensed in order to compare them, if the signals are equal, a reinforcement is made, otherwise, it indicates that some errors have occurred and the world's model must be updated. These errors bring the agent a better understanding of the world by a trial and error technique. Thus, the agent develops a better world's model.

4 Example

In this section we present an example in order to show how this model works.

This example illustrates how inferences of the world can be done through perception as well as the previous knowledge of it.

Example 1. Imagine the following scenario:
There are two agents in the same building but, they are in different rooms. The agents can perceive arbitrary sounds and know that a sound can be produced by another agent.

Suppose that the agent A perceives in time 1 an arbitrary sound.
Then, the agent A by this perception, can infer that agent B may produce this sound.
However, this inference may be wrong because it is not explicitly said that agent B makes the sound. But the agent A can valid this inference by asking the agent B.

We can model this problem as well as agent A inference into logic programming clauses. The first part is about the previous knowledge the agent has, which corresponds to the scenario. This information corresponds to the part of the World's Model of the Unified Perception-Prediction Model. The second part is about the perceptions the agent A has. This is according to perception part of the Unified Perception-Prediction Model. The third part is about the inference the agent A makes with the previous knowledge and perception. This inference should be stored into the world's model.

Knowledge:

```
agent(a). agent(b). sound(arbitrary).
emits(X,Y) ← agent(X), sound(Y).
```

Perception:

```
perception(P, T) ← sound(P), time(T).
perception(arbitrary,1).
```

Inference:

```
emits(b,arbitrary).
```

As we can see, the model is useful for agents. Furthermore, logic programming makes this task more easily.

5 Conclusions and Future Work

Here we introduced a new model for intelligent agents, which is based upon cognitive science. There are few works which involve logic programming and cognitive science [2] [1] however, there should be more research between these areas.
This model pretends to open a new framework for logic programs and to solve problems where the agent needs to perceive or predict the environment in order to act.
There is a lot of work ahead, for creating perception or prediction rules, as well as, create an implementation framework of this model.

References

1. H. T. Ahn and L. M. Pereira. Intention-based decision making with evolution prospection. *XV Portuguese Conference on Artificial Intelligence*, 2011.
2. M. Balduccini and S. Girotto. Asp as a cognitive modeling tool: Short-term memory and long-term memory. *Symposium on Constructive Mathematics in Computer Science*, pages 360–381, 2010.
3. C. Frith. *Making up the mind, How the brain creates our mental world.* Blackwell Publishing, 2007. 234 pages. ISBN: 1405160225.
4. M. Gelfond and V. Lifschitz. The stable model semantics for logic promgramminglogics with common weak completions. *5th Conference on Logic Programming*, pages 1070–1080, 1988.
5. J. W. Lloyd. *Foundations of Logic Programming.* Springer, Berlin, second edition, 1987.
6. M. Osorio, J. Arrazola, and J. L. Carballido. Logical weak completions of paraconsistent logics. *Journal of Logic and Computation, doi: 10.1093/logcom/exn015*, 2008.
7. Q. W. Qinru Qiu and R. Linderman. Unified perception-prediction model for context aware text recognition on a heterogeneous many-core platform. *International Joint Conference on Neural Networks*, pages 1714 – 1721, 2011.