Combining Answer Set Programs for Adaptive and Reactive Reasoning

Tony Ribeiro, Katsumi Inoue, Gauvain Bourgne

The Graduate University for Advanced Studies, Tokyo, Japan

National Institute of Informatics, Tokyo, Japan

University Pierre et Marie Curie, Paris, France

ICLP Technical communications, August 27th 2013

Outline

Answer Set Programming

Answer set programming (ASP) is a form of declarative programming introduced by (Gelfond and Lifschitz 1988).

ASP has been successively used in many works:

- Constraint programming (Niemela 1999)
- Knowledge representation and reasoning (Baral 2003/2008)
- Multi-agent systems (Nieuwenborgh et al. 2006)

Outline

Motivation

Our interest is to help the construction of efficient ASP based reasoning systems.

Example (For a Developer: Modeling)

The indivisible nature of ASP programs is causing increasing difficulties as program instances tend to grow in real applications.

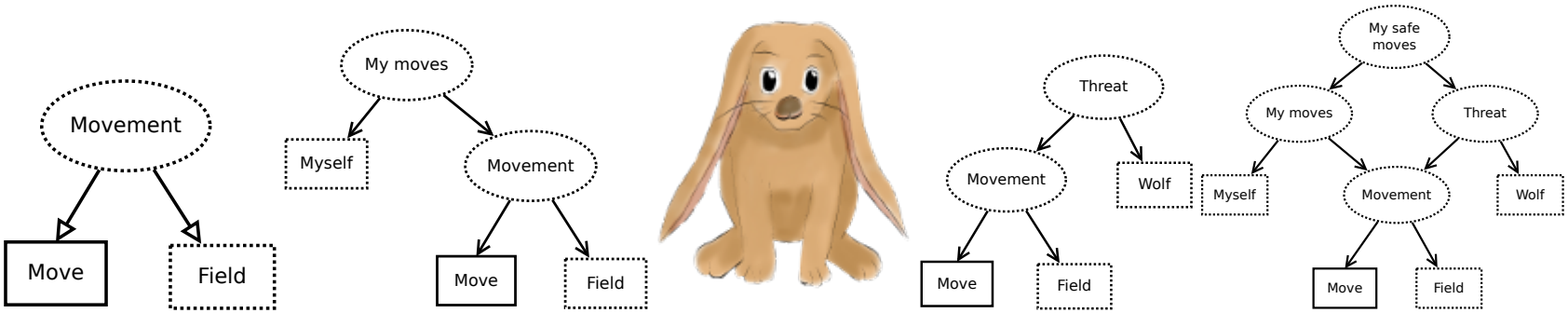
Example (For an ASP based System: Reasoning)

Reactivity of an ASP based system is very dependant on the quantity of knowledge, i.e. size of the progam, used for reasoning.

Outline

Proposal: design patern

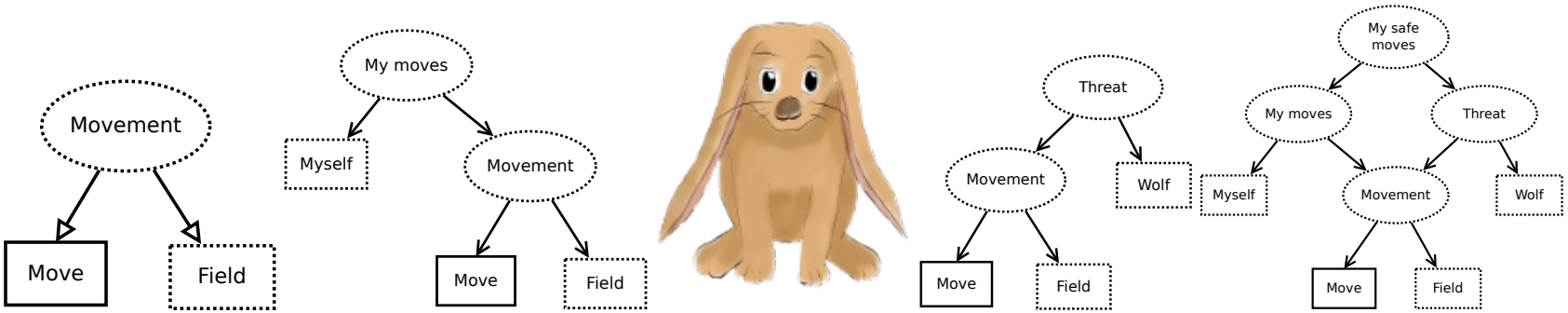
What we propose is a design patern to model reasoning by multiple ASP programs.



Our idea is quite similar to (Minsky 1991):
Is a mind composed of smaller and smaller minds, until the pieces become so small that they are no longer mind like?

Proposal: design patern

What we propose is a design patern to model reasoning by multiple ASP programs.

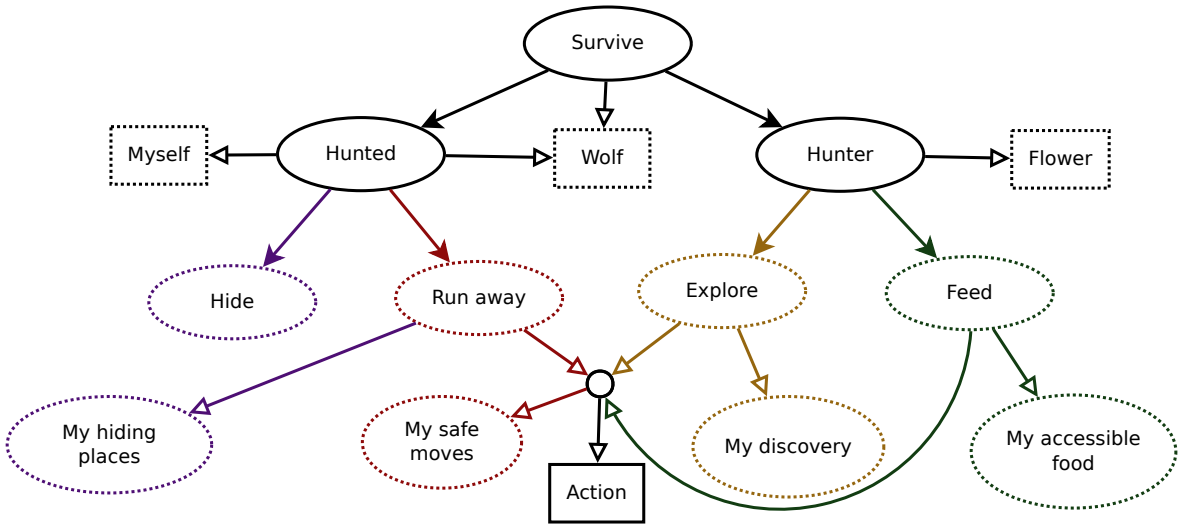


Example (Modeling)

It is usually easier to model a system as a set of small interacting components rather than a huge and obscure monolithic system.

Proposal: Framework

A framework to reason by combinations of ASP programs.



Example (Reasoning)

Modular division of ASP programs can allows to reduce the quantity of knowledge used for reasoning.

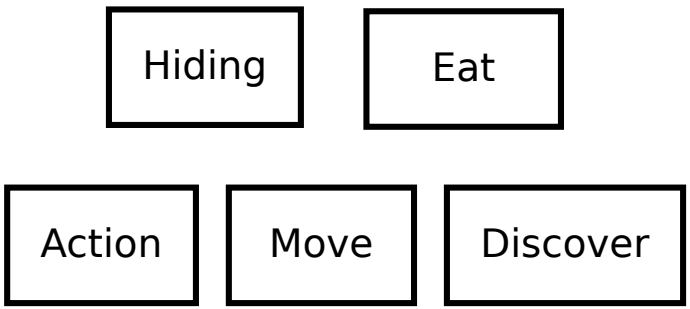
Outline

Theory and Observation Modules

We first propose to separate background knowledge and observations.

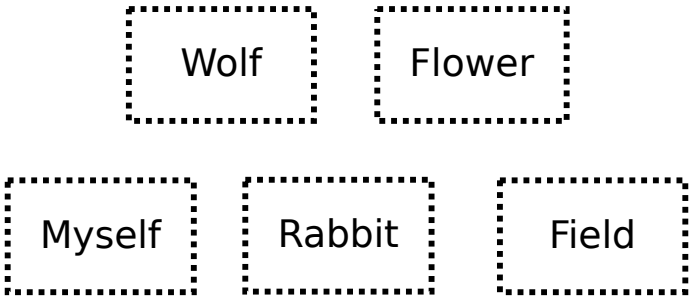
Theory Module

- Background knowledge
- Given



Observations Module

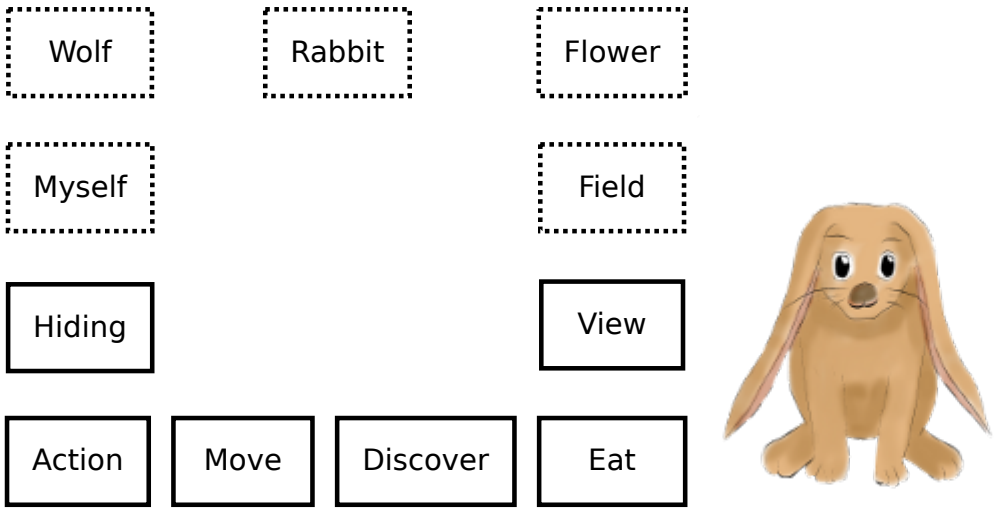
- Consistent observations
- Acquired



Outline

Module Combination

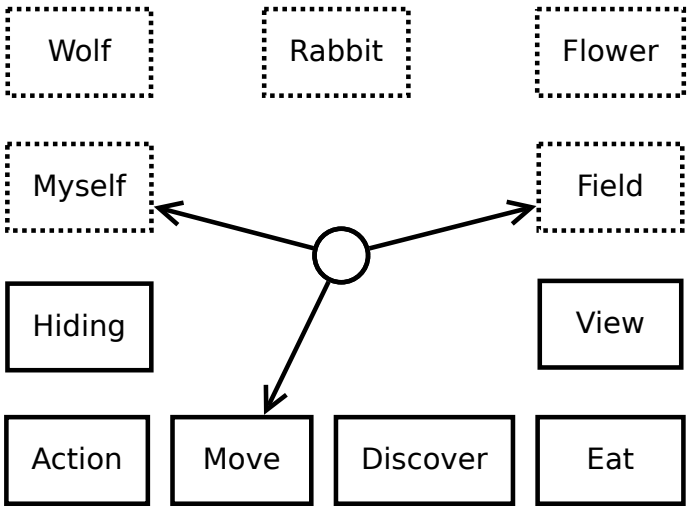
The core idea of our reasoning framework is similar to contextual logic programming (Monteiro and Porto 1989).



Modules combination

Module Combination

The core idea of our reasoning framework is similar to contextual logic programming (Monteiro and Porto 1989).



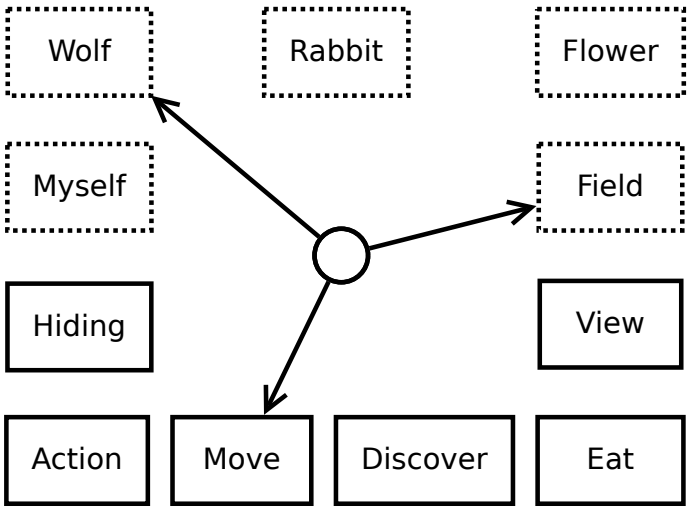
Modules combination

- My moves possibilities

We define a reasoning system consisting of independent logic programs which can be combined together regarding context needs.

Module Combination

The core idea of our reasoning framework is similar to contextual logic programming (Monteiro and Porto 1989).



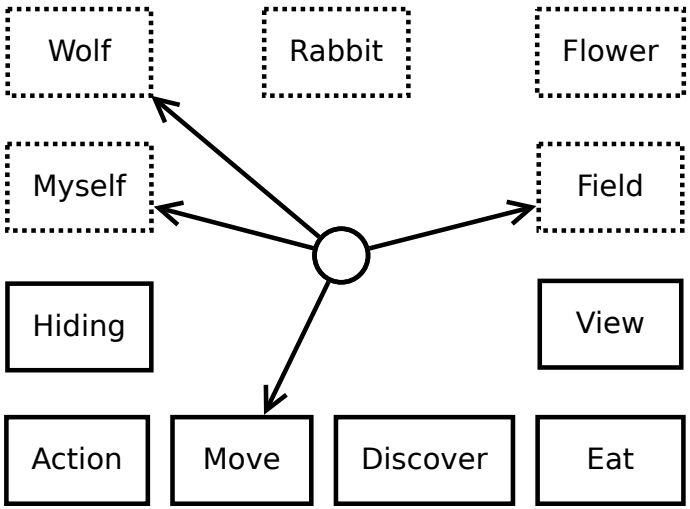
Modules combination

- My moves possibilities
- Wolves moves possibilities

We define a reasoning system consisting of independent logic programs which can be combined together regarding context needs.

Module Combination

The core idea of our reasoning framework is similar to contextual logic programming (Monteiro and Porto 1989).



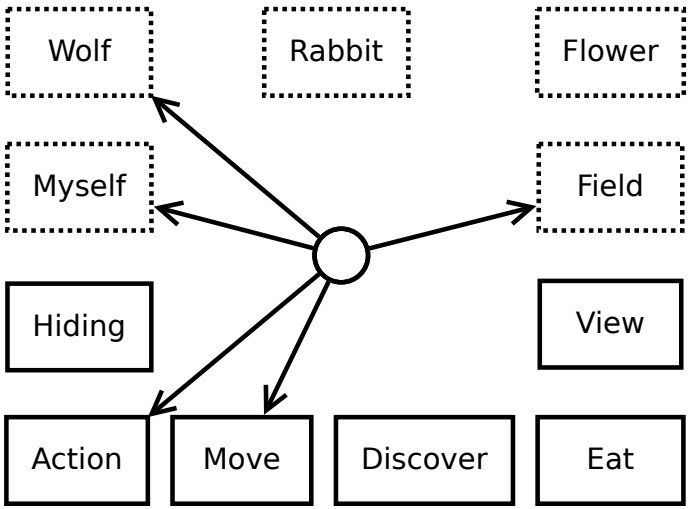
Modules combination

- My moves possibilities
- Wolves moves possibilities
- My safe moves

We define a reasoning system consisting of independent logic programs which can be combined together regarding context needs.

Module Combination

The core idea of our reasoning framework is similar to contextual logic programming (Monteiro and Porto 1989).



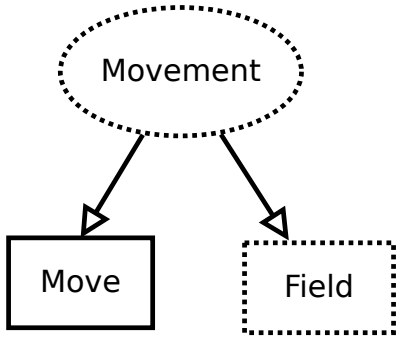
Modules combination

- My moves possibilities
- Wolves moves possibilities
- My safe moves
- Actions to run away

We define a reasoning system consisting of independent logic programs which can be combined together regarding context needs.

Combination and Meta-Knowledge

These programs combinations can be known by the agent as meta-knowledge about its own knowledge.



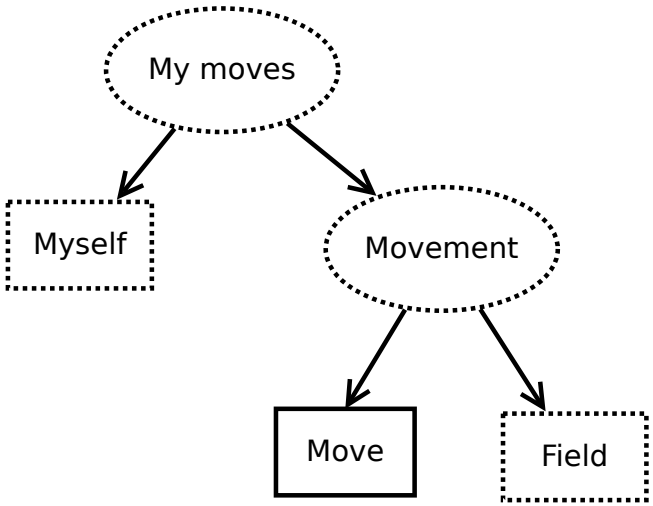
Meta-knowledge

- Movements

In our module typology we use meta-knowledge module to represent knowledge about module combination (dotted circle).

Combination and Meta-Knowledge

These programs combinations can be known by the agent as meta-knowledge about its own knowledge.



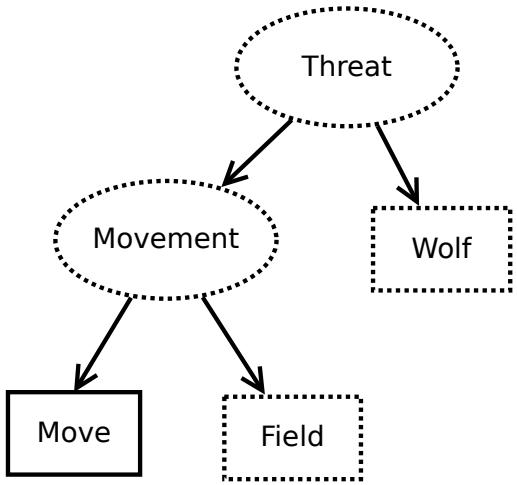
Meta-knowledge

- Movements
- My moves

In our module typology we use meta-knowledge module to represent knowledge about module combination (dotted circle).

Combination and Meta-Knowledge

These programs combinations can be known by the agent as meta-knowledge about its own knowledge.



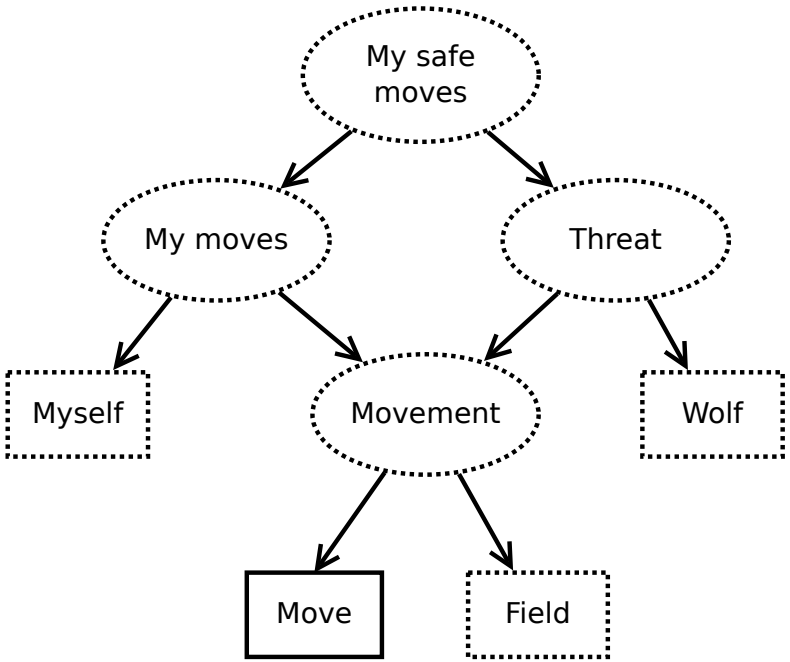
Meta-knowledge

- Movements
- My moves
- Threat

In our module typology we use meta-knowledge module to represent knowledge about module combination (dotted circle).

Combination and Meta-Knowledge

These programs combinations can be known by the agent as meta-knowledge about its own knowledge.

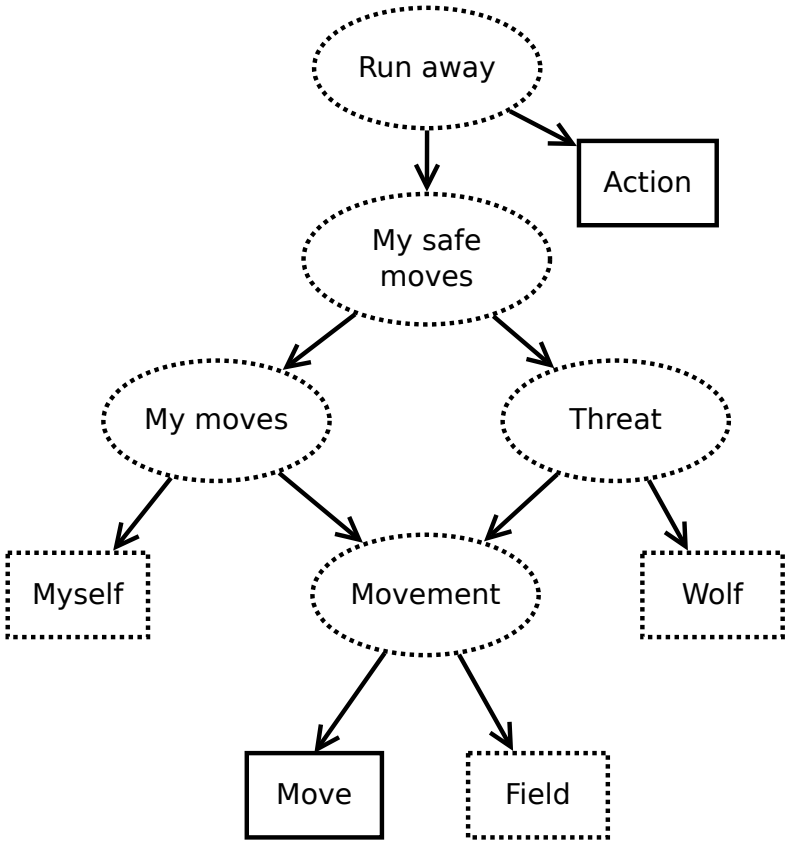


Meta-knowledge

- Movements
- My moves
- Threat
- Safe moves

Combination and Meta-Knowledge

These programs combinations can be known by the agent as meta-knowledge about its own knowledge.



Meta-knowledge

- Movements
- My moves
- Threat
- Safe moves
- Run away

Outline

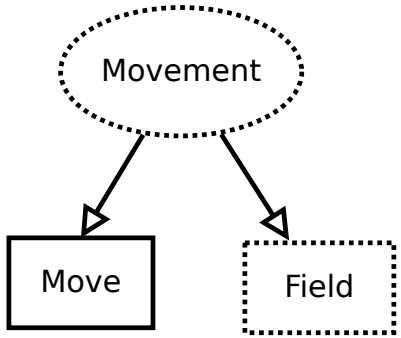
Keywords

Our framework parses ASP programs and answer set to interprete respectively two simples keywords:

- *include('module name')*

Movements

```
%#include "Move".
%#include "Field".
```



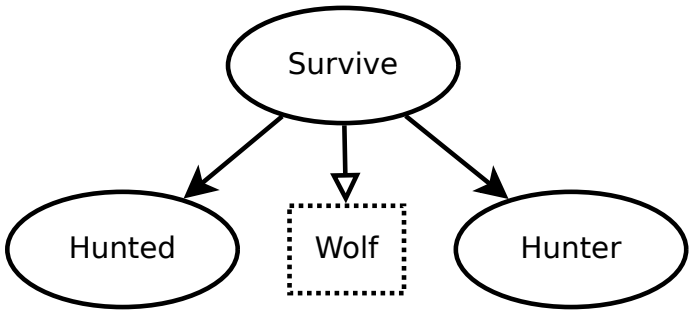
Keywords

Our framework parses ASP programs and answer set to interprete respectively two simples keywords:

- *include('module name')*
- *next('module name')*

Survive

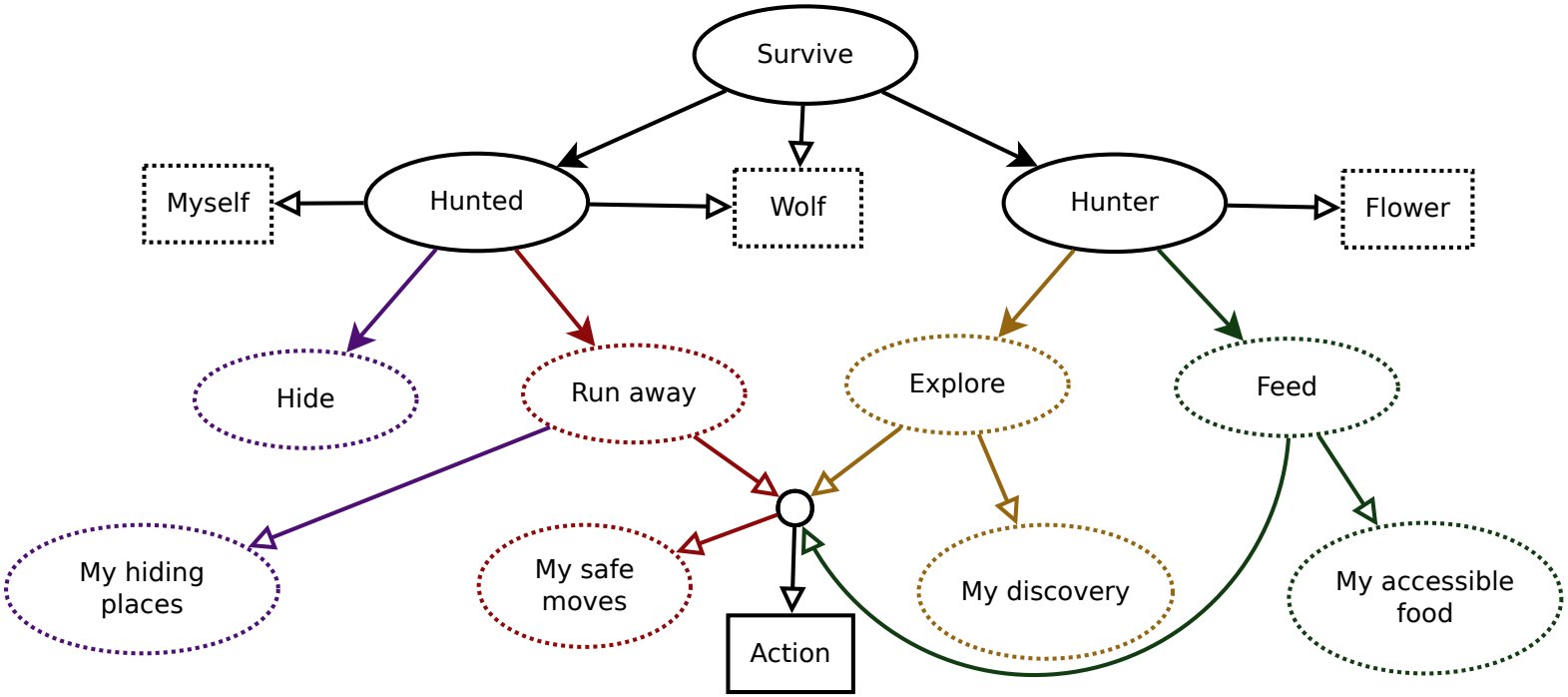
```
%#include "Wolf".
%#extern position/2.

wolf :- position(wolf,Position).
next("Hunted") :- wolf.
next("Hunter") :- not wolf.
```



Outline

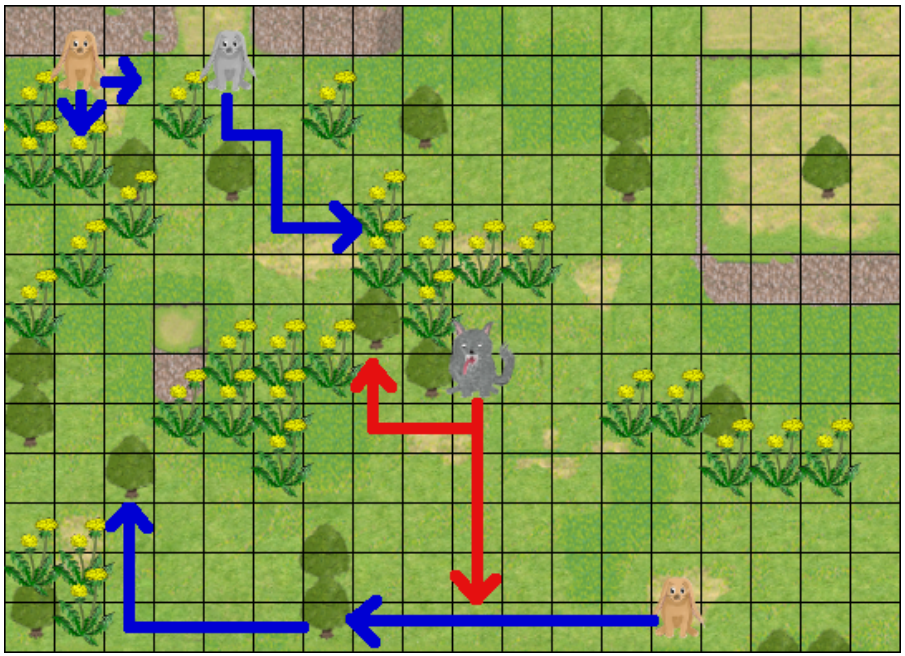
Knowledge Division

In this example, we propose a possible division of the reasoning of a rabbit agent into 4 independent parts.



Empty arrows represent inclusions, plain ones represent decisions.

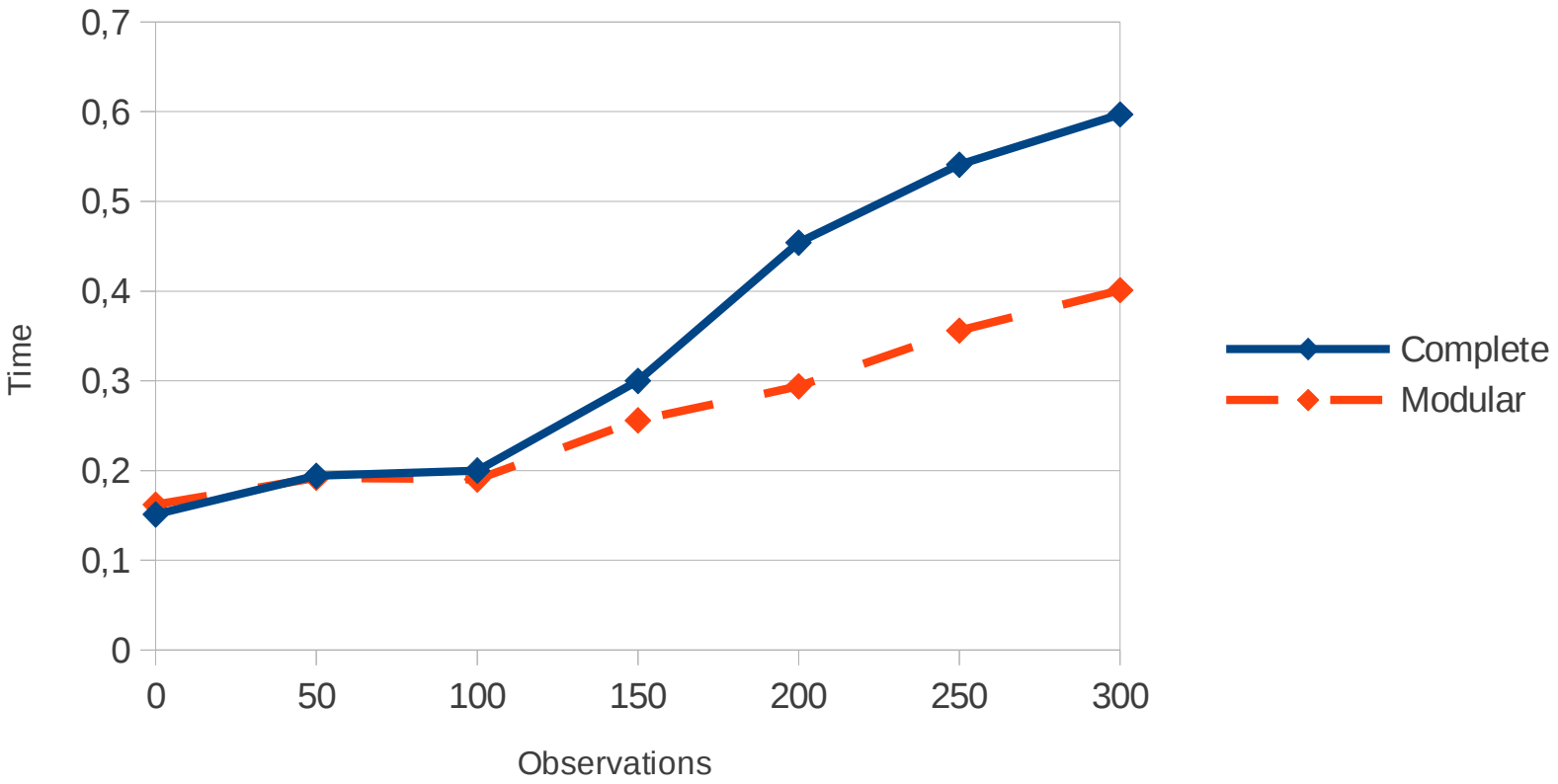
Experiments



Experimental application based on the survival game example. Arrows represent some movements that agents are considering in order to explore/feed/hide/escape.

Results

Reasoning time evolution regardings number of observations.



Experimental results, for each method it shows rabbit reasoning time average of 1000 runs.

Outline

Contributions

- Contributions
  - A design pattern to represent knowledge as multiple ASP programs.
  - A framework to use this representation within an ASP based reasoning system.
- Outlook
  - Dynamic learning of module and combinations
    - Experiment based
  - Generic methods to divide a monolithic ASP program
    - Rules dependancy


Baral, C. 2003.
Knowledge representation, reasoning and declarative problem solving.
Cambridge university press.


Baral, C. 2008.
Using answer set programming for knowledge representation and reasoning: Future directions.
In *ICLP*. 69–70.


Gelfond, M. and Lifschitz, V. 1988.
The stable model semantics for logic programming.
In *ICLP/SLP*. 1070–1080.


Minsky, M. 1991.
Society of mind: A response to four reviews.
Artif. Intell. 48, 3, 371–396.


Monteiro, L. and Porto, A. 1989.
Contextual logic programming.
In *ICLP*. 284–299.


Niemelä, I. 1999.
Logic programs with stable model semantics as a constraint programming paradigm.
Ann. Math. Artif. Intell. 25, 3-4, 241–273.


Nieuwenborgh, D. V., Vos, M. D., Heymans, S., and Vermeir, D. 2006.
Hierarchical decision making in multi-agent systems using answer set programming.
In *CLIMA*. 20–40.