

ShallowX

Tommy Petersen
tp@aia.dk
Copyright 2015 AIA
All rights reserved

July 22, 2015

Abstract

This article presents a basic theory for an explorative agent.

1 Introduction

The ability to explore seems crucial to many species.

Two features characterize exploration:

- **Action** Something new has to happen and if it does not, an action is chosen to make it so.
- **Mapping** During exploration the environment is mapped in order to be understood.

An explorative agent explores the environment on behalf of it's clients. This article presents a basic theory for an explorative agent. The theory can be used as a design framework for a general explorative agent. Both the theory and the general agent are called **ShallowX**.

2 ShallowX data flow diagram

A diagram representing the general data flow around ShallowX is shown in figure 1.

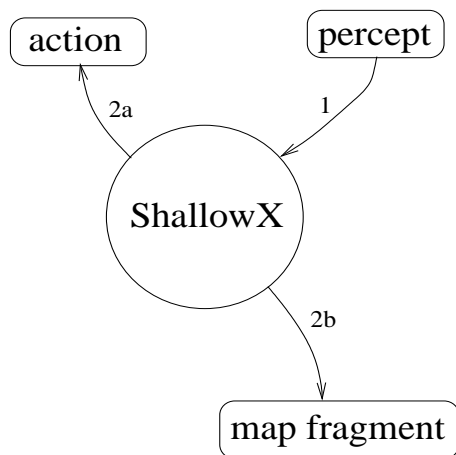


Figure 1: ShallowX data flow diagram.

The data named “**action**”, “**percept**” and “**map fragment**” are all finite symbol sequences and should be understood as follows:

- **percept** Sensory data from the environment.
- **action** Chosen action.
- **map fragment** Part of a map of the environment.

ShallowX continually acts and maps as a result of a never ending stream of percepts from the environment.

3 Context concept

In order to describe what is happening in the ShallowX data flow diagram (figure 1), some new concepts are defined.

First, the **experience** is defined as a finite symbol sequence. The experience represents the internal state of ShallowX.

Second, the **potential** is defined.

Definition 3.1 (*potential*)

The potential is a finite symbol sequence defined as

$$potential := experience.percept$$

where the operator '.' is concatenation. The right hand side is kept unconcatenated, so that it is clear which part of the potential is the experience and which part is the percept.

The **concatenated potential** is then the concatenation of the experience and the percept.

The percept is used as a **context** in which to structure the experience. A sequence in the concatenated potential matching the percept is an **instance** of the context. Example:

$$experience = 11010$$

$$percept = 110$$

$$potential = 11010.110$$

In the concatenated potential, the context and the instance are as follows:

$$\begin{array}{c} \boxed{110}10|110 \\ \uparrow \quad \uparrow \\ \text{instance} \quad \text{context} \end{array}$$

The concepts of context and instance now make it possible to define the two exploration features **action** and **mapping** as will be done in the following sections.

4 Overlapping instances

Context instances are not necessarily separated from each other, they can overlap. ShallowX uses this situation to choose an action.

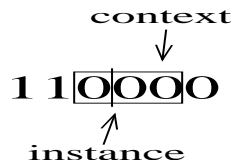
The following example shows a situation where an instance overlaps with the context itself:

$$experience = 110$$

$$percept = 000$$

$$potential = 110.000$$

In the concatenated potential, the context and the instance are as follows:



The instance overlaps the context by the sequence 00.

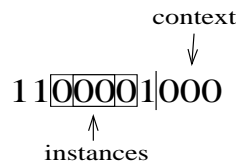
There might also be overlaps earlier on, as shown by the following example:

experience = 1100001

percept = 000

potential = 1100001.000

In the concatenated potential, the context and the instance are as follows:



The two instances overlap by the sequence 00.

A combination of the above examples is also possible.

4.1 Chosen action

This section shows how ShallowX uses overlapping instances to choose an action.

First, a lemma, to help argue about what can be expected when instances overlap.

Lemma 4.1 (*the overlap lemma*)

If two instances overlap, then their union will end with a pattern which repeats itself at least two times.

Proof.

See figure 2. ■

The overlap lemma shows that ShallowX can expect a future input of periodic characteristic. But a periodic input is not desired for an explorative agent as ShallowX is. It should find new things not repetitions. Since the overlap seems to be a problem, the choice is to return the concatenation of some selected instance overlaps as ShallowX's chosen action. It is still a research topic which overlaps to select. The selected overlaps are removed from ShallowX's concatenated potential as they have been used to resolve the possible input period and hence should not keep indicating periodic input. Using the previous example with

experience = 110

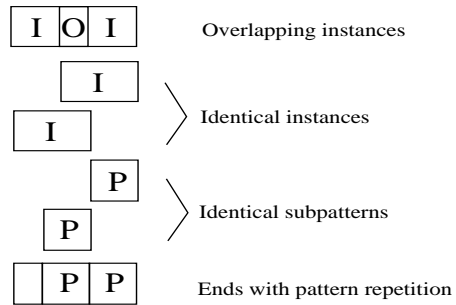


Figure 2: Proof of the overlap lemma.

$$percept = 000$$

$$potential = 110.000$$

and by selecting the latest instance overlap (which is the only overlap in the example), the overlap is 00 and after it is removed from the concatenated potential this becomes

$$concatenated\ potential = 1100$$

After having chosen an action ShallowX updates it's experience to be the concatenated potential and then awaits a new percept.

5 Non overlapping instances

As implied earlier, it is possible to have situations where context instances do not overlap. It will be described how ShallowX uses some of these situations to map it's environment.

6 No previous instance

There might be no previous instance so that the context itself is the only instance. Example:

$$experience = 10001$$

$$percept = 011$$

$$potential = 10001.011$$

The concatenated potential:

$$\begin{array}{c} \text{context} \\ \downarrow \\ 10001|011 \end{array}$$

In this case, ShallowX cannot map it's environment and therefore does nothing but take the concatenated potential to be the new experience:

$$experience = 10001011$$

ShallowX then awaits a new percept.

7 Previous instance

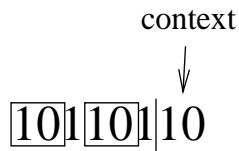
As the following example shows there can be one or more non overlapping instances:

$$experience = 101101$$

$$percept = 10$$

$$potential = 101101.10$$

The concatenated potential:



Here, the context is matched by instances in the experience giving information about what has happened earlier in that particular context. This information can be used to map the environment.

8 Map fragment

In order to describe how ShallowX maps it's environment, the concept of a map fragment is defined.

Definition 8.1 (*map fragment*)

A map fragment is a pair of the form

$$(C, I)$$

where C and I both are finite symbol sequences. C is interpreted as a context and I is interpreted as information about that context.

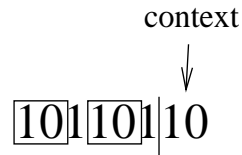
In the case of previous non overlapping instances, the context information can be used as a map fragment. This is done by letting C be the context and by letting I be the concatenation of the part of the experience that is between some selected context instances. It is still a research topic which context instances to select. Consider again the example:

$$experience = 101101$$

$$percept = 10$$

$$potential = 101101.10$$

The concatenated potential:



By selecting all instances the map fragment is (10,11) with C equal to 10 and I equal to 11.

In general, a map fragment should contain the newest available information about the context. Hence the part of the experience that was used for the map fragment is removed from the experience. Then, the next time a map fragment is made in the same context, it is not the previously mapped information that is used but newest available information. In the above example, only the context remains after removal of used information:

$$\text{updated experience} = 10$$

ShallowX sends the map fragment to it's clients and then awaits a new percept.

9 Flow

Having described the details of the ShallowX data flow diagram in figure 1, the flow can now be summarized as in figure 3:

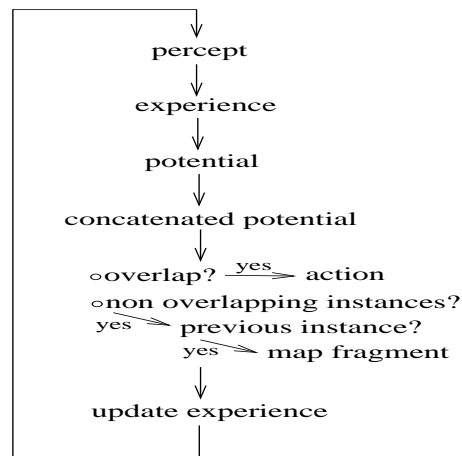


Figure 3: Flow.

It is still a research topic whether it should be possible to return both an action and a map fragment at the same time, or only one of them should be returned.