Artifical Intelligence Programming Paradigms
Assignment 2

Antoine Carpentier

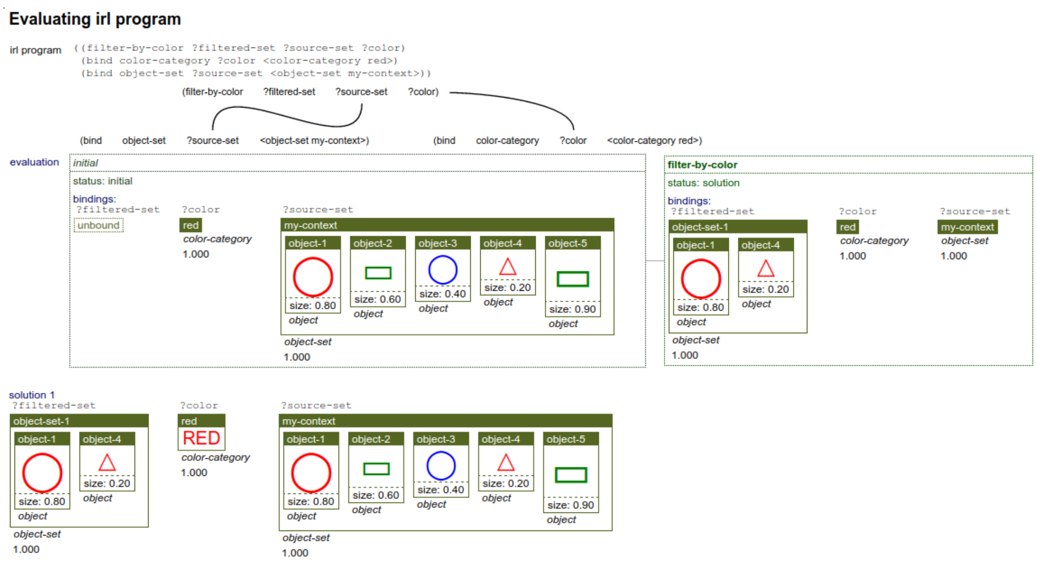
November 4, 2016

1 Introduction

In this assignment, we demonstrate the use of the Incremental Recruitment Language (IRL) framework to filter geometric forms by color. The primitive we introduce is named ”filter-by-color”. It can both filter an object set given a color and retrieve a color given a source set and a filtered set. We use an ontology in the primitive to retrieve the context and the colors.
We reuse some of the code from ”irl-tutorial.lisp” to filter by shape and show several combinations of the two primitives. We also use the context defined in ”irl-tutorial.lisp” and add colors to objects.
We introduce 3 color categories : red, green and blue and we think of a similarity measure to introduce continuous color values in RGB.
We also think of a way to chunk this primitive for composition and matching.
The code is available in the ”assigment2.lisp” source file.

2 Stand-alone use

In 1, we filter all red objects from the context using the primitive.


Evaluating irl program

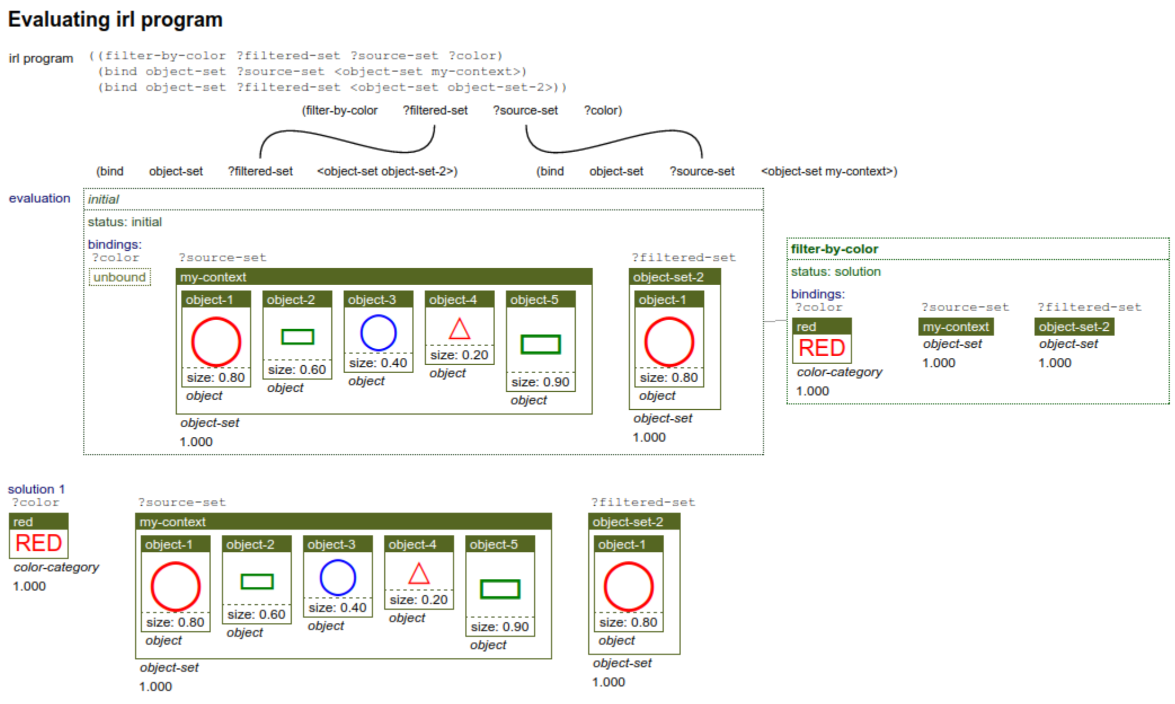
irl program
```
((filter-by-color ?filtered-set ?source-set ?color)
 (bind color-category ?color <color-category red>)
 (bind object-set ?source-set <object-set my-context>))
```

Figure 1: Filter red objects from context

In 2, we retrieve the red color from the context and a filtered set containing the red circle.
In 3, we try to retrieve yellow objects and there is no solution in this context.

3 Use in combination with shape

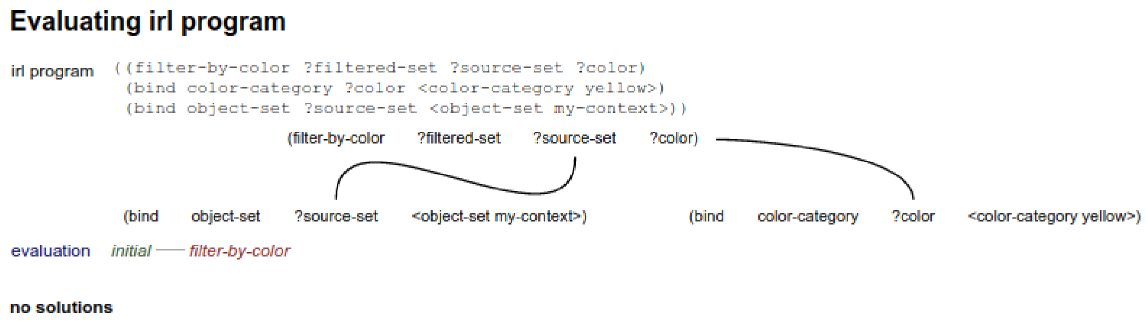
In 4, we retrieve all red circle objects from the context using ”filter-by-color” and ”filter-by-shape”.



Evaluating irl program

irl program
```
((filter-by-color ?filtered-set ?source-set ?color)
 (bind object-set ?source-set <object-set my-context>)
 (bind object-set ?filtered-set <object-set object-set-2>))
```

Figure 2: Retrieve red color from context and red circle



Evaluating irl program

irl program
```
((filter-by-color ?filtered-set ?source-set ?color)
 (bind color-category ?color <color-category yellow>)
 (bind object-set ?source-set <object-set my-context>))
```

(filter-by-color ?filtered-set ?source-set ?color)
(bind object-set ?source-set <object-set my-context>)
(bind color-category ?color <color-category yellow>)
evaluation
initial
filter-by-color
no solutions

Figure 3: Try to retrieve yellow objects

In 5, we retrieve the red color and the circle shape from the context and a filtered set containing the red circle.

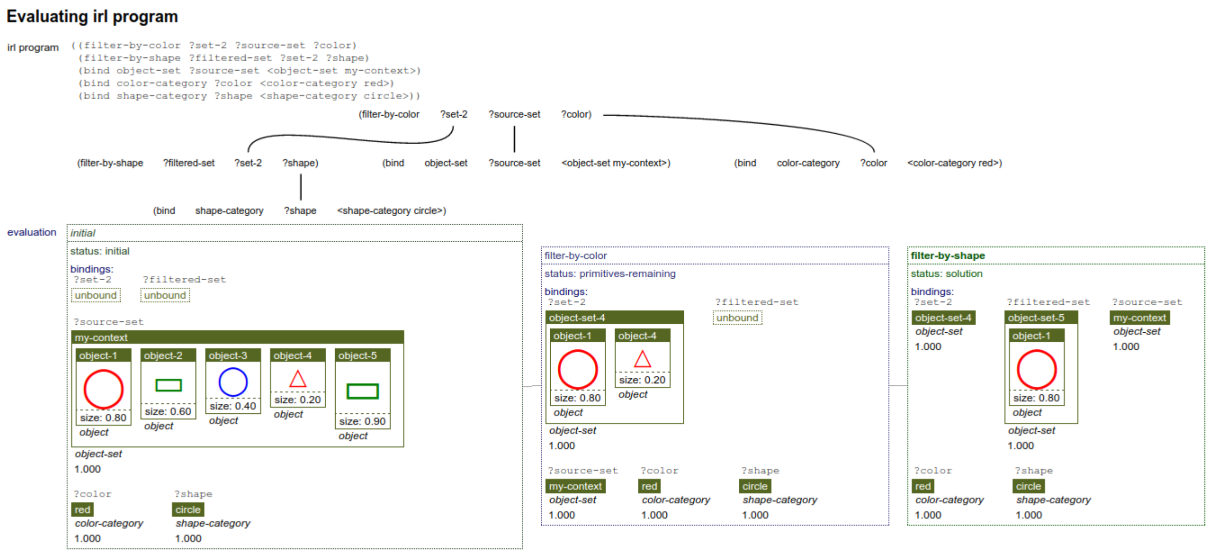
In 6, we try to retrieve red rectangles and there is no solution in this context.

4 Continuous color values

In order to introduce continuous color values, we could code them as a triplet of RGB colors, each ranging from 0 to 1. The similarity measure would be the euclidian distance in the 3-dimensional space. Color categories would be defined as a triplet in the same way and in order to categorize a color in such a category, we would choose the category with the highest similarity.

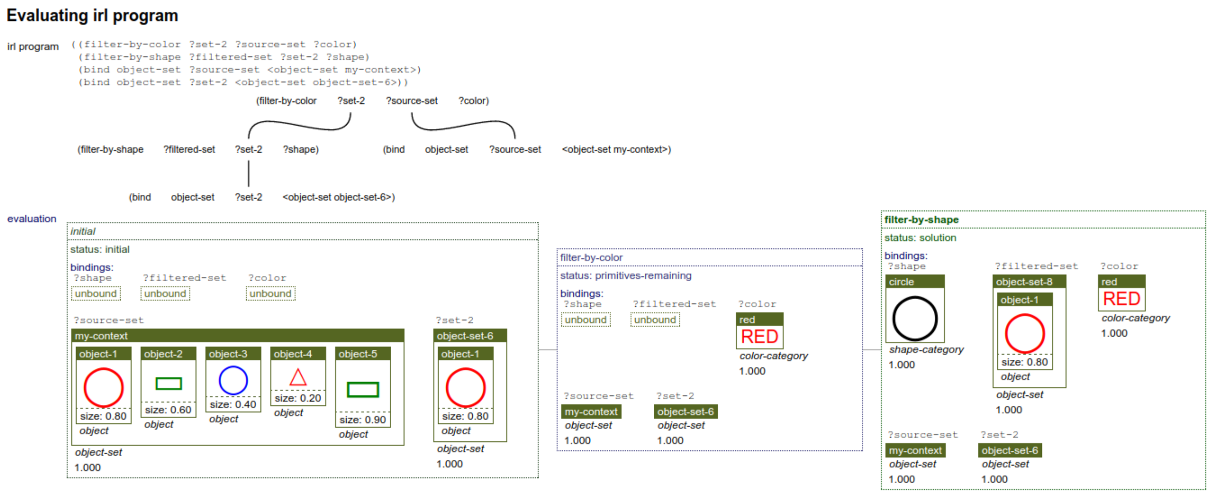
5 Chunk

In 7, we use use a chunk to retrieve all objects of a given color. The chunk uses the primitives ”get-context” and ”get-color-category” to get data from the ontology.


Evaluating irl program

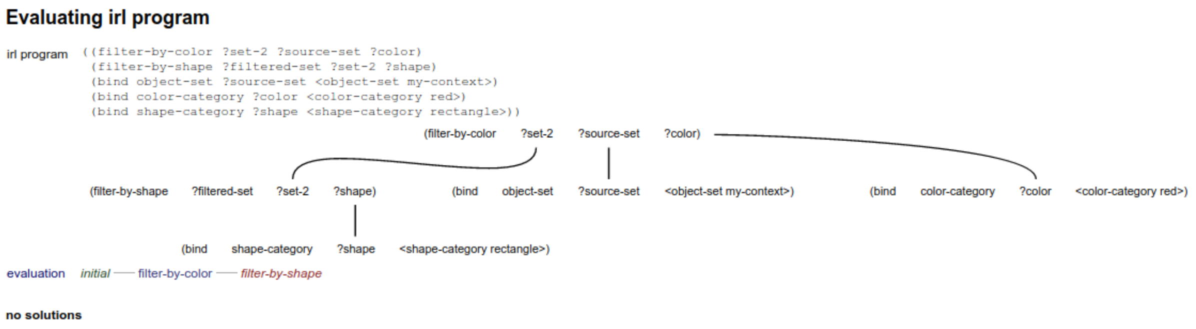
irl program
```
((filter-by-color ?set-2 ?source-set ?color)
 (filter-by-shape ?filtered-set ?set-2 ?shape)
 (bind object-set ?source-set <object-set my-context>)
 (bind color-category ?color <color-category red>)
 (bind shape-category ?shape <shape-category circle>))
```

Figure 4: Filter red circle objects from context


Evaluating irl program

irl program
```
((filter-by-color ?set-2 ?source-set ?color)
 (filter-by-shape ?filtered-set ?set-2 ?shape)
 (bind object-set ?source-set <object-set my-context>)
 (bind object-set ?set-2 <object-set object-set-6>))
```

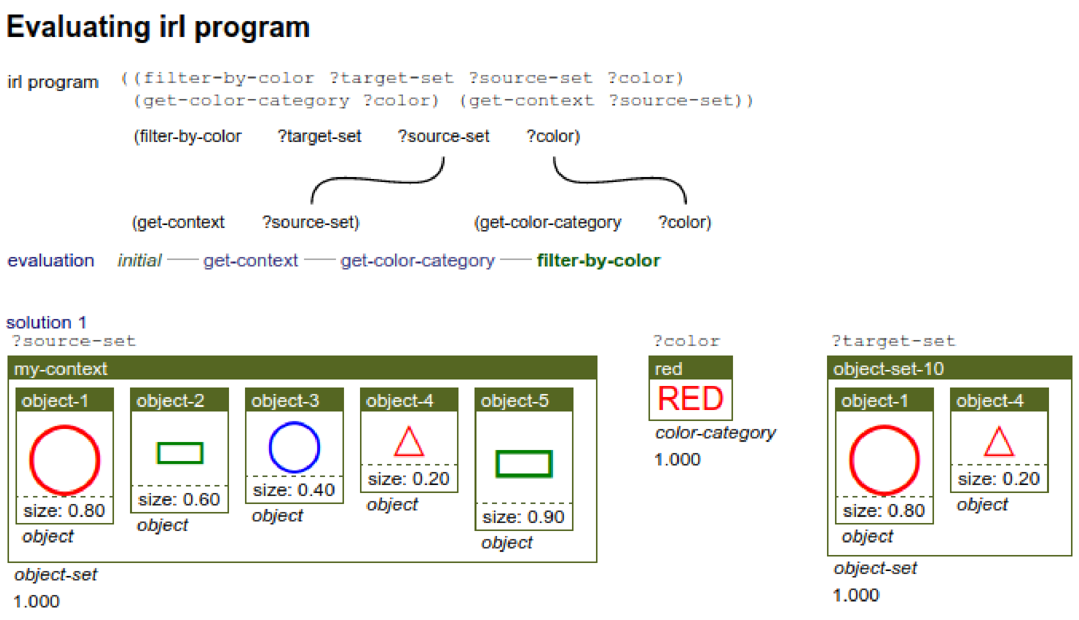
Figure 5: Retrieve red color and circle shape from context and red circle


Evaluating irl program

irl program
```
((filter-by-color ?set-2 ?source-set ?color)
 (filter-by-shape ?filtered-set ?set-2 ?shape)
 (bind object-set ?source-set <object-set my-context>)
 (bind color-category ?color <color-category red>)
 (bind shape-category ?shape <shape-category rectangle>))
```

no solutions

Figure 6: Try to retrieve red rectangles



```
((filter-by-color ?target-set ?source-set ?color)
 (get-color-category ?color) (get-context ?source-set))
```

Figure 7: Retrieve red objects using a chunk