Empirical Reasoning AI Systems
Assignment 2 Report

Reykjavik University - School of Computer Science, Menntavegi 1, IS-101 Reykjavík, Iceland

Damiano Buzzo
Matteo Machella

15. November 2025

Contents

1 Last assignment in narsese

In this task, we have created a narsese file to execute the previous assignment made with handwritten NAR's calculations. Thanks to this, we have discovered some little errors of computations so we were able to adjust it. See the quote for reference:

> After trying the narsese, for revising the assignment, we've found out that the judgements J_{16}, J_{18} and J_{24} couldn't be computed using the abduction rule, due to a mistake in defining the abduction formula.
>
> In detail, we have stated that $f_x \geq 0.5$ is a premise for doing the abduction, whereas the correct form is $f_y \geq 0.5$, making the calculations impossible.
>
> Knowing this, we acknowledge the fact that there are no other way to compute $apple \rightarrow plant, apple \rightarrow fruit, fruit \rightarrow plant$ other than the original J_1, J_2, J_4.

2 Knowledge transfer via shared properties

To complete this task, we have encountered a lot of obstacles. First of all, the "similarities" inference it's done using the FIFO modality because ONA will stop at the first shared property to determine if something is similar or not. Basically, if cube and sphere share two properties, and cube and cylinder only share one, sphere and cylinder will have the exact same similarity to the cube because ONA will ignore the second shared property between cube and sphere.

Things are also worse if you create the three objects as an instance of `object`, because ONA will then see that every object share the inheritance of `object` and therefore making them similar.

We made a lot of tentatives, and in most of them, the robot can pick the sphere and cannot pick the cylinder. We've tried giving the experience and also not giving the experience but also with some general rule. We have also tried to do both of them, but in most cases the output was the same.

In the end, we've decided to make the `.nal` file very clean and simple, removing experience and using only a general rule and then prompting if other objects can be held, resulting in having the sphere grabbed but not the cylinder.

```
*setopname 1  go
*setopname 2  pick
*motorbabbling=false
*volume=0


<{cube} -> [blue]>.
<{cube} -> [heavy]>.
<{cube} -> [hard]>.

<{sphere} -> [blue]>.
<{sphere} -> [heavy]>.
<{sphere} -> [soft]>.

<{cylinder} -> [red]>.
<{cylinder} -> [light]>.
<{cylinder} -> [hard]>.

<<$1 -> {cube}> ==> <(<$1 -> [at]> -> pick) =/> <$1 -> [is_held
   ]>>>.

<?1 =/> <{sphere} -> [is_held]>>?

<?1 =/> <{cylinder} -> [is_held]>>?
```

3 Knowledge transfer via identical relations

For this task, we've decided to build two general rules addressing the `MoreMassiveThen`, `Attracts` and `RevolvesAround`. Afterwards, we have given as an input for the knowledge base instances to those relations and finally the queries are asked. ONA is completely able to infer that earth revolves around the sun, and that the planet revolves around the sun, even with a lower confidence.

When we decide to add properties to the sun, we observe that the output of the query "is earth revolving around the sun?" is the exact same of the original, therefore it was not influenced by the new properties. Probably because in the general rule, that the machine uses for the inference, there are no references for the properties to judge those relations.

If the general rule is not implemented, ONA will fail to infer the queries and therefore it won't work as expected. This is because the machine doesn't have enough reasoning material to work with to understand all the causal relation behind those relations.

```
*motorbabbling=false

<<($1 * $2) -> MoreMassiveThan> ==> <($1 * $2) -> Attracts>>.
<<($1 * $2) -> Attracts> ==> <($2 * $1) -> RevolvesAround>>.

<(nucleus * electron) -> MoreMassiveThan>.
<(nucleus * electron) -> Attracts>.
<(electron * nucleus) -> RevolvesAround>.

<(sun * earth) -> MoreMassiveThan>.
<(sun * earth) -> Attracts>.

// expected: Answer: <(earth * sun) -> RevolvesAround>. creationTime=10
    Stamp=[6,2,7,1] Truth: frequency=1.000000, confidence=0.836900
<(earth * sun) -> RevolvesAround>?

<earth -> planet>.

// expected: Answer: <(planet * sun) -> RevolvesAround>. creationTime
   =13 Stamp=[8,6,2,7,1] Truth: frequency=1.000000, confidence=0.429618
<(planet * sun) -> RevolvesAround>?

<sun -> [yellow]>.
<sun -> [hot]>.

// these queries has the same answers as the original ones :c
<(earth * sun) -> RevolvesAround>?

<(planet * sun) -> RevolvesAround>?
```

4 Knowledge transfer via comparison

With the last task, we've first decided to make the people's names an instance of the `person` entity. After that, we've established the `TallerThan` relation between `Quinn` and `Ari`, and then between `Robin` and `Quinn`.

We then define a general rule for which, if x is taller than y, then x can play basketball. This then leads to the statement that `Quinn` plays basketball and thereby the machine is able to infer that also `Robin` can play it, since they are taller than `Quinn`.

Interestingly, since every one of them is a person, the machine is able to infer that also `Ari` is able to play basketball, but with a lower confidence, which in our minds makes sense. If eventually, the relationships with `person` are removed, the machine won't be able to infer that also `Ari` can play basketball. We think that it's because of the general rule is all based on the tallness of people, and therefore if a person is the shortest, `None` is given as an answer as it will never fall inside the general rule.

```
1
2  *motorbabbling=false
3
4  <{Ari} -> person>.
5  <{Quinn} -> person>.
6  <{Robin} -> person>.
7
8  <({Quinn} * {Ari}) -> TallerThan>.
9  <({Robin} * {Quinn}) -> TallerThan>.
10
11 <(<($1 * $2) -> TallerThan> && <$2 -> PlayBasketball>) ==> <$1 ->
       PlayBasketball>>.
12
13 <{Quinn} -> PlayBasketball>.
14
15 // expected: Answer: <{Ari} -> PlayBasketball>. creationTime=11 Stamp
       =[7,2,1] Truth: frequency=1.000000, confidence=0.402762
16 <{Ari} -> PlayBasketball>?
17
18 // expected: Answer: <{Robin} -> PlayBasketball>. creationTime=11 Stamp
       =[5,7,6] Truth: frequency=1.000000, confidence=0.729000
19 <{Robin} -> PlayBasketball>?
```