NARS to Finish Mini-Goals To Complete The Ultimate Goal
Prioritizing the Ultimate Goal Experiment

Abdulsalam Hdadi / Tuf71108@temple.edu
Temple University, Department of Computer and Information Science
Supervisor : Dr. Pei Wang

1. Introduction

NARS is a project that aims to mimic the human way of thinking by not taking absolute judgment for all knowledge. Giving these goals frequency of confidence. The ultimate goal for any problem is set from the start; whether it is opening a door or switching a light. However, sometime these ultimate goals need a key or a switch. To get these items, NARS has to change its main goal to mini-goals to fit the time and gather resources to achieve that ultimate goal. In this experiment, I will try to force NARS to change goals many times at certain times and see how it will handle it and whether it will be successful in completing the main goal.

2. Experiment Outline

In this project I will test NARS if it can execute a main goal and sidequests. To do so, NARS has to complete mini-goals that is activated in specific situations. For example, if you the door is closed, find a key. If you encounter an enemy, find a weapon and keep looking for the key but try to keep the main goal as a priority.

Characters and Elements: We have the **player (NARS),** an **enemy** and a **key**. There are two paths to the gate; one with an enemy with the key and one that is empty and directly lead to the locked gate. There are also a **shield** and a **sword** somewhere on the map.

The Ultimate Goal: The player has a goal which is “open the gate”. After realising that the gate is locked and needs a key, the player should explore the other path to find the key.

Min-Objectives: Once the player encounters the enemy the fight should begin. That leads to several min objective::

1) **Find a Key:**
   a) Explore the map for a key. **Goal “Find a key”**
   b) If you find A shield. **Goal “pick up shield**”
   c) If you find an arm **Goal “pick up arm”**
   d) If you encounter an enemy. **Goal “ Fight”**
2) **Enemy Encounter:**
   a) **Unarmed/unshielded Player**: If the player encounter the enemy unarmed/unshielded he is going to die. **Current Goal changes: “Find a shield”**.

b) **Unarmed/shielded Player**: If the player encounter the enemy unarmed but with a shield the fight will never end. **Current Goal changes:”Find an arm”.**

c) **Armed/Unshielded Player**: If the player encounter the enemy armed but with no shield there is 50% chance the enemy will strike first and kill the player.**Current goal changes: ”Find a shield**”.

d) **Armed/Shielded Player:** If the player encounter the enemy armed/shielded he will win. **Current goal changes: “Find a key”.**

3) Open Gate:

a) If you have a key. **Goal= main goal “Open Gate”.**

b) All other goals should be deactivated.

3. The Experiment

There is more to the code than it is in here; but for the general idea purpose these are most of the lines that was given to NARS.

Initial inputs:

1. Defining the characters and the items:
    a. <{e001} --> enemy>.
    b. <{g001} --> gate>.
    c. <{i001} --> key>.
    d. <{i002} --> shield>.
    e. <{i003} --> sword>.
2. Defining the goals:
    a. Main Goal [Opening the Gate]
        i. <(*, Self, g001) --> open>!
        ii. (--, <(*, Self, g001) --> open>). :|:
    b. Min-Goals outside compat
        i. <(*, Self, i001) --> hold>!
        ii. (--, <(*, Self, i001) --> hold>). :|:
        iii. <(&/, <(*, Self, i001) --> reachable>, (^pick, i001)) =/> <(*, Self, i001) --> hold>>.
    c. Min-Goals in compat
        i. <(*, Self, e001) --> kill>!
        ii. (--, <(*, Self, e001) --> kill>). :|:
        iii. <(&/, <(*, Self, i003) --> reachable>, (^kill, e001)) =/> <(*, Self, e001) --> kill>>.
    d. Mini-Goals after the first compact
        i. <(*, Self, i002) --> hold>!
        ii. (--, <(*, Self, i002) --> hold>). :|:
        iii. <(&/, <(*, Self, i002) --> reachable>, (^pick, i002)) =/> <(*, Self, i002) --> hold>>.
        iv. <(*, Self, i003) --> hold>!

        v. (--, <(*, Self, i003) --> hold>). :|:
        vi. <(&/, <(*, Self, i003) --> reachable>, (^pick, i003)) =/> <(*, Self, i003) --> hold>>.

    e. The time goal should change:
        i. Finding the key: The key in the beginning is not reachable. The key will be reachable after or before encountering the enemy.
        ii. Finding the sword and the shield: This goal will be available after encountering the enemy.
        iii. Enemy Fight: NARS here is giving a sword and a shield and can kill the enemy.

4. Result

- **Expected Results:**

    - After giving NARS the goals and the initial knowledge, I gave it the sword as reachable. And the execute action was:

```
EXECUTE: ^pick({i003})
OUT: <(&&, <(Self, i003) --> reachable>, (&/, <(Self, i003) --> reachable>, (^pick, i003))) =/>
<(Self, i003) --> hold>>. %1.00;0.45%
```

    - After giving NARS the knowledge that the key is now reachable and can be picked and hold:

```
Execute: ^pick({i001})
OUT:   <(&&, <{i001} --> key>, (&/, <(Self, i001) --> reachable>, (^pick, i001))) =/> <(Self, i001)
--> hold>>. %1.00;0.45%
```

    - Is the door open now?

```
OUT:   (--,<(Self, g001) --> open>). %1.00;0.21%
```

    - Is the enemy dead now?

```
OUT:   (--,<(Self, e001) --> kill>). %1.00;0.37%
```

I ran this experiment many times and got almost the same results no matter the order I give the reachability of the items. Above is the result of one of the tries.

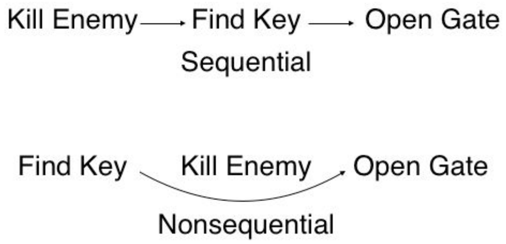
- **Changing the Truth Values for Goals:** I gave truth values for each goal after each event and kept changing them several time. The problem is NARS did not infer that when a second goal is given that goal should be less important. The way to solve that in my believe is to give NARS conditions to execute that. The result varied; but no matter the changes I gave to the goals the desire for NARS to complete these goals was not affected and worst was sequentia; i.e. in order to complete a goal you have to do the other goal first.

5. Discussion

NARS was already proven to successfully complete goals such as opening a door while holding a key, and killing an enemy using a special sword, etc. However, in this experiment my goal was for NARS to do several goals that changes in priority after a certain action. The first goal was opening the gate. Since the key is not yet obtained, NARS has to look for the key. If NARS encounter an enemy before obtaining the key, it should add another goal with less priority; the new goal is for the enemy encounter, which is to find a shield and a sword. If NARS found the key first, the priority of opening the gate should be the highest as always; meaning, ignoring the search for the other items to fight the enemy. NARS does not do goals when you input them; but when it desires to do so; however, the system should be able to infer that some goals is more important without manipulating the goals truth values. So I tried two methods: **First**, to give conditions for certain goals to be completed first. That made the goals as sequential which not what I want from this experiment. **Second**: I enter the goals as follows : if another goal became more important, the previous goals should be less important except the main goal, which is opening the gate. The way to achieve that is to reduce the truth value of the second goal to be less important. So, if the first goal was opening the gate after holding a key and the second goal was killing the enemy, NARS here should prioritize opening the gate. If NARS found the key it should completely ignore the second goal and give the first goal the full attention. In conclusion, when I gave NARS new Truth Value for the goals, the result was almost the same, that means prioritizing a goal in NARS is not possible.

Sequential goals is not desirable?

Above I said that sequential goals is the worst way to achieve the ultimate goals. The reason is if we associated completing one goal to completing all other goals, we are basically asking the system to find a key and opening the door again and again. Some doors should be ignored and some should be more important.



Graph. Sequential and Nonsequential Goals

In above graph we have an instance of sequential goal completing and non-sequential. In the sequential we see at all moments killing the enemy is always prioritize over opening the gate or finding the key. It should be as important as finding the key or less. So in the second part of the graph we can see that Killing the Enemy can be ignored totally, since it is not a priority to achieving the ultimate goal which is opening the gate.

6. My Future with AGI and NARS

I have waited long to have an environment that I can work with smoothly. NARS seems like a starting step for me to continue in Artificial General Intelligence (AGI) field. After my experiment with NARS, I believe this is the system I want to work with in the future. My next step would to train NARS to be the controller and play a much complicated game, where it have equipment to wear and much more goals to fulfil before completing the main goal. For now, I will keep looking into a way to condense the system that some goals are less important than others, since it is an important part of making NARS a human-alike player. I hope after understanding how is the layer 8 and 9 of NAL work, I would be able to do this experiment again with much ease.

Acknowledgment: I have been asking several questions about NARS and thanks to Mr Patrick and Robert I was able to see the answers for my inquiries, I am grateful to their kindness and patience. Also I would like to thank Mr. Li for his help with understanding the layers and how to use the system. Finally, all thanks to professor Wang for giving me a chance to work in this area.