Application of Frames and Production Systems in Story Representation and Diagnostics

Magahet Mendiola
(mmendiola3@mail.gatech.edu)

February 6th, 2015

Introduction

We will utilize Frames to represent the situations illustrated in a number of story excerpts. We will then create a production system for the diagnosis of robot related troubles. Finally, we will examine the concept of Chunking, and how long-term memory in a production system can be adapted, using episodic memory, to handle impasses.

Frames and Story Representation

The following sentences have been adapted to frame representations:

"I knelt in front of the unmoving blue robot."

```
I
    ako: person
    state: kneeling
    location: in front of Robot

Knelt
    subject: I
    location: in front of Robot

Robot
    state: unmoving
    color: blue
```

"As if brooding, it sat on the floor in the middle of the living room."

```
It
    state: sitting
    location: middle of the living room
    appearance: brooding

Sitting
    subject: it
    location: middle of the living room
    on: floor
```

" 'The Johnsons across the street bought a new robot,' it said finally."

```
Johnsons
    location: across the street
    action: Bought

Bought
    subject: Johnsons
    object: Robot

Robot
    condition: new

It
    action: Said

Said
    subject: It
    statement: Johnsons bought a new robot
```

" 'Yeah,' the husband confirmed from behind me, 'One of those new A-01 models.' "

```
Husband
    action: confirmed
    location: behind me
    statement: Yeah, one of those new A-01 models

Confirmed
    subject: Husband
    statement: Yeah, one of those new A-01 models

A-01
    ako: robot model
    condition: new
```

Production Systems Diagnostic Procedure

The following is an example set of production rules that is designed to diagnose problems observed in the story:

```
Working Memory
    robot name:
    goal: meet robot


Rule 1:
    If:
        goal is to meet robot
        and I perceive robot has no name
    Then:
        address the robot

Rule 2:
    If:
        goal is to meet robot
        and I perceive robot responds with
        his name
    Then:
        add robot name
        add problem is unknown
        suggest goal of diagnose problem


Working Memory
    robot name: Henry
    goal: diagnose problem
    problem: unknown


Rule 3:
If:
    goal is to diagnose problem
    and I perceive problem is unknown
Then:
    say, "Are you functioning correctly"

Rule 4:
If:
    goal is to diagnose problem
    and I perceive problem is unknown
    and I perceive robot's response is
    evasive
Then:
    say, "What's that all about?"
    set problem to unclear


Working Memory

    robot name: Henry
    goal: diagnose problem
    problem: unclear
```

```
Rule 5:
If:
    goal is to diagnose problem
    and I perceive problem is unclear
Then:
    say, "Go on"

Rule 6:
If:
    goal is to diagnose problem
    and I perceive problem is unknown
    and I perceive robot is jealous
Then:
    set goal to encourage robot


Working Memory

    robot name: Henry
    goal: encourage robot
    problem: robot is jealous
```

Chunking

We will now examine the concept of chunking in the context of the previous production system.

Let us suppose that the following rule is missing:

```
Rule 1:
    If:
        goal is to meet robot
        and I perceive robot has no name
    Then:
        address the robot
```

Let us also suppose that the current working memory contains the following:

```
robot name:
goal: meet robot
```

There are no rules in long-term memory that would allow the agent to achieve this goal state. This impasse can be resolved by looking at other cases when the goal state was reached and observe the working memory to determine what rule is needed to continue. For example, a previous case could include this state of the working memory:

```
robot name: Henry
goal: diagnose problem
problem: unknown
```

This indicates that we need to supply the working memory with the robot's name. We could then derive the required rule to get this information from the robot for the purpose of updating our working memory and moving forward with the diagnostic process.

In this case, we see that the slot for robot name has been filled. We can build a rule that if the robot name slot is empty, we should do something to evoke a response. We will therefore, input a rule into long-term memory that if the goal is to meet the robot, and the current working memory does not include the robot's name, that we should address the robot. This new rule will allow the system to handle the instance we observed initially.