TAC Game Report

Tristan Claverie
950418P612
trcl16@student.bth.se

Damien Duvacher
941206P633
dadu16@student.bth.se

I. Agent Architecture

An architecture for an agent define the global work methods of the agent.
It exists many architectures but 2 are more famous:

- Reactive Agent
- Reasoning Agent

The first one is when there are many behaviours developped for and agent and then when an event occurs, the agent react to this event by chosing one of his behaviour. So the agent just react when events occurs.
Whereas the reasoning agent will look at his environement and his goal then he will deliberate to find a solution to reach his goal regarding his starting state.
The second one is more intelligent but it's not very effective for simple tasks which need quick reaction.
These 2 architectures are different but sometimes you can mix both to create an hybrid agent. So you can adapt to many situations.
In this part we will describe the architecture for the 2 kind of agent in the philosophy dinner problem, for the naive agent (whithout cooperation) and for the smart agent(with cooperation).

A. Naive Agent

This kind is self fish and will act to satisfy his needs.
Basically, the goal is to eat when the philosopher is hungry. Then when the philosopher become hungry he will just try to get the 2 forks to eat and if he has the 2 forks, he starts to eat until he becomes full.
This strategy is really simple because the agent (or the philosopher) just look at the forks, if they are available or not and then take it to eat wihtout consider other agent needs. So for this kind of philosopher it's a reactive agent that is used because this behaviour is just a series of if. So the agent just react to different situation, there is no deliberation to reach his goal.

B. Smart Agent

This philosopher is a little bit more smart because he will look at the marks on the forks and will mark the forks that he need. The goal is the same: eat when hungry. But this timethe agent be careful of other agents needs.
Despite this difference the architecture for this smart philospoher is the same than the naive: a reactive agent.
Because the difference is about the marks on the forks, and the marks are managed by a "if series" too, indeed, if a philospoh has a fork marked and he doesn't have the other fork he will put down the fork because the mark mean that an other philosoph need it.
As you can see before it's just manage by many "if", he doesn't think about his environment and how he can reach his goal.

II. Agent Communication and Interaction

In this part we wiil talk about the interaction and the communication between the different agents in this multi-agent system. We will first see the intercation between naive agents and then for the smart agent.

A. Naive agent

For the naive version of the agent, the communication is not really advanced because the philospohers don't communicate between them. One philospoh will just communicate with the fork, he will just look if they are available or not. And the interaction is not really complicated neither because a philosopher can just take or release forks. If we want to find a little part of interaction between philosopher it's that if a philosopher have 2 forks another one can't have his 2 forks.
So for the naive version there is no lot of interaction and communication.

B. Smart agent

This kind of philosopher is more interesting because smart philosopher are supposed to cooperate to help each other to reach their goal: eat and think.
Here there is a communication between philosopher because they mark a fork, that means they want this fork and if a philosopher have a marked fork but doesn't have the other one he will release this fork because someone else need it.
It's a kind of communication not directly but by the forks, they send informations to each other marking forks they need.
The interaction between philosopher is the same that the naive agent.