Dining Philosophers Problem (DPP)

Consider $n$ philosophers sitting around a table. There is a fork on the table between every two philosophers, and if each philosopher was able to eat with one fork, they would be able to eat their meal. However, our philosophers need two forks in order to be able to eat. How can we design a protocol such that no philosopher starves?

We can express this problem in CTL, using the atoms $e_i$ to mean that philosopher $i$ is eating, and $f_i$ to mean that philosopher $i$ has just finished eating.

The following is a statement of strong fairness, meaning that each philosopher gets infinitely many turns to eat:

$$\text{AG}( \bigwedge_{i \in [1,\ldots,n]} \text{AF}e_i)$$

The following is a statement of weak fairness, meaning that each philosopher gets to eat at least once:

$$\bigwedge_{i \in [1,\ldots,n]} \text{AF}e_i$$
The following are several statements that do not actually imply fairness:

1. $\text{AF} (\bigwedge_{i \in \{1, \ldots, n\}} e_i)$ says that eventually, all philosophers must eat simultaneously, which is impossible.

2. $\bigwedge_{i \in \{1, \ldots, n\}} \text{EF} e_i$ does not even guarantee that everyone eats at all.

All deterministic strategies for the dining philosophers fail to provide fairness, e.g. because of deadlocks. However, randomization allows the design of a deadlock-free protocol, as follows: each philosopher...

- Attempts to pick up either his right or his left fork, at random.
- If he succeeds, he attempts to pick up the other fork.
  - If he succeeds, he eats!
  - If he fails, he puts down the first fork and starts over.
- If he fails, he starts over.

See Handout 22 for details.

**CTL Model Checking**

CTL model checking works from the inside out, checking the smallest sub-WFFs first. Consider the following example: $\phi = \text{AXEF} \psi$, where $\psi = (\neg p \land q \land \neg r) \lor (p \land \neg q \land r)$. $\phi$ is checked starting with $(\neg p \land q \land \neg r)$ and $(p \land \neg q \land r)$ individually, continuing with $\text{EF}$, and finishing with $\text{AX}$. See Handout 23 for details.

**Markov Processes / PCTL**

PCTL stands for Probabilistic Computation Tree Logic. It is a variant of CTL with probabilistic path quantifiers.

**Definition 1.** A Markov Chain is a 5-tuple

$$\mathcal{M} = (S, \mathcal{P}, \text{init}, AP, L),$$

where

- $S$ denotes a non-empty, finite or countably infinite set of states.
- $\mathcal{P}$ denotes a transition probability function - $\mathcal{P} : S \times S \rightarrow [0, 1]$, where $[0, 1]$ is a closed interval of rational numbers. It is required that for all states $s \in S$, it hold that
  \[ \sum_{s' \in S} \mathcal{P}(s, s') = 1. \]

- init denotes a initial state distribution - init : $S \rightarrow [0, 1]$. It is required that
  \[ \sum_{s \in S} \text{init}(s) = 1. \]

- AP is the set of atomic propositions in the model.

- $L$ is a labeling function - $L : S \rightarrow \mathcal{P}(AP)$, where $\mathcal{P}$ denotes the power set.

As an example, consider the following model $\mathcal{M}$:

Since the only possible init state is $s_1$ (meaning $\text{init}(s_1) = 1$, $\text{init}(s_i) = 0$ for $i \neq 1$), $\mathcal{M}$ would have associated initial state distribution init:

\[
\text{init} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = (1, 0, 0, 0)
\]

$\mathcal{M}$ would have the following associated transition probability matrix $\mathcal{P}$:

<table>
<thead>
<tr>
<th></th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$s_2$</td>
<td>0</td>
<td>0</td>
<td>1/10</td>
<td>9/10</td>
</tr>
<tr>
<td>$s_3$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$s_4$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The transition probability matrix is a right-stochastic matrix, meaning that all of its rows add up to 1, but no such restriction is placed upon the columns.

Note that if $\mathcal{M}$ had multiple possible actions, instead of just one, each action would then have its own transition probability matrix.
Syntax of PCTL

There are two equivalent syntaxes for PCTL:

1. $a \in AP$
   - $\phi ::= \bot | a | \neg \phi | \phi \land \psi | P^{\triangledown p} \psi$
   - $\psi ::= X\phi | \phi U \phi$
   - $\triangledown \in \{<, >, \leq, \geq, =, \neq\}$
   - $p \in [0, 1]$ (where $[0, 1]$ is a closed interval of rational numbers)

   $M, s \models P^{\triangledown p} \psi$ iff $\Pr[M, s \models \psi] \triangleright p$.

2. $a \in AP$
   - $\phi ::= \bot | a | \neg \phi | \phi \land \psi | P^{\triangledown p} \psi$
   - $\psi ::= X\phi | \phi U \phi | \phi U \leq n \phi$

   $M, \pi \models \phi_1 U \leq n \phi_2$ iff $\phi_1$ in path $\pi$ holds until $\phi_2$ holds within at most $n$ steps.

Note that $\{\neg, \land\}$ is adequate for prepositional logic, $\{X, U\}$ is adequate for LTL, and $\{AU, EU, EX\}$ is adequate for CTL.

Instead of using the $P^{\triangledown p}$ notation, we will use the following shorthand: $X^{\triangledown p} \phi$ to denote $P^{\triangledown p} X \phi$, and $\phi_1 U^{\triangledown p} \phi_2$ to denote $P^{\triangledown p} (\phi_1 U \phi_2)$. Informally, this notation describes the probability with which events occur. For instance, $X^{\leq .5} \phi$ means “$\phi$ should be true in the next state with probability at most .5”. More formally,

- $M, s \not\models \bot$
- $M, s \models a$ iff $a \in L(s)$
- $M, s \models X^{\triangledown p} \phi$ iff $\Pr[\{\pi | \pi = t_1 t_2 \ldots \text{ and } t_1 = s \text{ and } M, t_2 \models \phi\}] \triangleright p$
- $M, s \models \phi_1 X^{\triangledown p} \phi_2$ iff $\Pr[\{\pi | \exists j \geq 1 \text{ s.t. } M, \pi^j \models \phi_2 \text{ and } \forall i < j, M, \pi^i \models \phi_1\}] \triangleright p$

References:

4. Professor Kfoury’s lecture notes