Tutorial questions: embedded agents[1]

1. A purely reactive agent is defined by the function $Ag^r : E \to Ac$.

   A standard agent is defined by the function $Ag^s : \mathcal{R}^E \to Ac$.

   (Recall: E is the set of all possible environment states; Ac is the set of all available actions; $\mathcal{R}^E$ is the set of all possible runs over E and Ac that end with an environment state.)

   (a) Is the following statement true? "For every purely reactive agent, there is a behaviourally equivalent standard agent."

   If true, explain how you can define a standard agent that is behaviourally equivalent of any given purely reactive agent.

   If not true, give an example of a purely reactive agent and explain why it is not possible to define a behaviourally equivalent standard agent.

   (b) Is the following statement true? "For every standard agent, there is a behaviourally equivalent purely reactive agent."

   If true, explain how you can define a purely reactive agent that is behaviourally equivalent of any given standard agent.

   If not true, give an example of a standard agent and explain why it is not possible to define a behaviourally equivalent purely reactive agent.

   (Recall: Two agents Ag_1 and Ag_2 are behaviourally equivalent if, for any environment Env, $\mathcal{R}(Ag_1, Env) = \mathcal{R}(Ag_2, Env)$.)

   Answer

   (a) "For every purely reactive agent, there is a behaviourally equivalent standard agent."

   This is true!

   A purely reactive agent takes as input the current environment state, and decides what to do based on that.

   A standard agent takes as input the current run in the environment, and decides what to do based on that.

   We can define a standard agent so that, whatever the current run is, it performs the action that the purely reactive agent will perform when in the last state of that run. They will then behave equivalently.

   Formally:

   Let $Ag^r : E \to Ac$ be a purely reactive agent.

   We can define a behaviourally equivalent standard agent $Ag^s : \mathcal{R}^E \to Ac$ as follows:

   For any run $r = (e_0, \alpha_0, \ldots, \alpha_{u-1}, e_u) \in \mathcal{R}^E$, $Ag^s(r) = Ag^r(e_u)$.

[1] Many thanks to Michael Rovatsos for inspiring these questions.

(b) "For every standard agent, there is a behaviourally equivalent purely reactive agent."

This is false!

Here is a counter example.

Let Ag^s be a standard agent, in an environment Env with states $E = \{e_0, e_1, e_2\}$ and possible actions α_0 and α_1, such that

- $Ag^s((e_0)) = \alpha_0$,
- $Ag^s((e_0, \alpha_0, e_0)) = \alpha_0$,
- $Ag^s((e_0, \alpha_0, e_1)) = \alpha_0$,
- $Ag^s((e_0, \alpha_0, e_0, \alpha_0, e_1)) = \alpha_0$, and
- $Ag^s((e_0, \alpha_0, e_1, \alpha_0, e_1)) = \alpha_1$.

Let:

- $\tau((e_0, \alpha_0)) = \{e_0, e_1\}$,
- $\tau((e_0, \alpha_0, e_0, \alpha_0)) = \{e_1\}$,
- $\tau((e_0, \alpha_0, e_1, \alpha_0)) = \{e_1\}$,
- $\tau((e_0, \alpha_0, e_0, \alpha_0, e_1, \alpha_0)) = \{e_2\}$, and
- $\tau((e_0, \alpha_0, e_1, \alpha_0, e_1, \alpha_1)) = \{e_2\}$.

Thus we have:

- $(e_0, \alpha_0, e_0, \alpha_0, e_1, \alpha_0, e_2)) \in \mathcal{R}(Ag^s, Env)$, and
- $(e_0, \alpha_0, e_1, \alpha_0, e_1, \alpha_1, e_2)) \in \mathcal{R}(Ag^s, Env)$.

Since there are only two possible actions, for any purely reactive agent Ag^r we design it has to be the case that either $Ag^r(e_1) = \alpha_0$ or $Ag^r(e_1) = \alpha_1$.

Let's assume that $Ag^r(e_1) = \alpha_0$, but then it cannot be the case that:

$$(e_0, \alpha_0, e_1, \alpha_0, e_1, \alpha_1, e_2) \in \mathcal{R}(Ag^s, Env)$$

Let's assume that $Ag^r(e_1) = \alpha_1$, but then it cannot be the case that:

$$(e_0, \alpha_0, e_0, \alpha_0, e_1, \alpha_0, e_2) \in \mathcal{R}(Ag^s, Env)$$

It follows then that $\mathcal{R}(Ag^s, Env) \neq \mathcal{R}(Ag^r, Env)$ (and hence the agents are not behaviourally equivalent).

2. Rock-paper-scissors is a 2-player game where each agent simultaneously selects from one of three moves: rock (R), paper (P), or scissors (S). The outcome is determined as follows:

   - R beats S;
   - S beats P;
   - P beats R;
   - all other combinations result in a draw.

   The set of actions available to the agent are $Ac = \{R, P, S\}$.

   The utility of winning is 1, the utility of losing is -1, and the utility of a draw is 0.

   If we consider a single round of the game we can define the possible environment states as

   $$\{e_0, RR, RS, RP, SR, SS, SP, PR, PS, PP\}$$

   where:

   - e_0 is the initial state before game play has begun, and
   - each two-letter combination XY represents the state in which the agent has played X and its opponent has played Y (e.g., SP means the state in which the agent played scissors and its opponent played paper).

   Assume our opponent (the "environment") plays:

   - P with probability 0.4,
   - S with probability 0.5, and
   - R with probability 0.1.

   Thus for any $\alpha \in Ac$:

   - $\tau((e_0, \alpha)) \in \{RR, RS, RP, SR, SS, SP, PR, PS, PP\}$,
   - the probability that $\tau((e_0, \alpha)) \in \{PP, RP, SP\}$ is 0.4,
   - the probability that $\tau((e_0, \alpha)) \in \{PS, RS, SS\}$ is 0.5, and
   - the probability that $\tau((e_0, \alpha)) \in \{PR, RR, SR\}$ is 0.1.

   We can define a non-deterministic *stochastic* agent

   $$Ag^{st} : \mathcal{R}^E \to 2^{Ac \times [0,1]}$$

   such that $(\alpha, p) \in Ag^{st}(r)$ means that p is the probability the agent will perform action α as the next action of the run r, and the following conditions hold:

   for all $r \in \mathcal{R}^E$:

   - if $(\alpha_i, p_i) \in Ag^{st}(r)$ and $(\alpha_i, p_j) \in Ag^{st}(r)$, then $p_i = p_j$ (i.e., there can be at most one occurrence of each action); and
   - $\sum_{\forall(\alpha_i, p_i) \in Ag^{st}(r)} p_i = 1$ (i.e., the sum of the probabilities associated with all the possible actions must equal 1).

   Specify an *optimal* stochastic agent Ag^{st} for this problem (i.e., one that maximises the expected utility).

Answer

Specify an *optimal* stochastic agent for this problem (i.e., one that maximises the expected utility).

Winning states (utility 1) are: PR, RS, SP.

Drawing states (utility 0) are: RR, PP, SS.

Losing states (utility -1) are: RP, SR, PS.

We can specify the agent in terms of variables p and r, such that $Ag^{st}(e_0) = \{(P,p),(R,r),(S,1-p-r)\}$, thus we need to find values of p and r that maximise our expected utility.

$$\begin{aligned}
\text{Expected utility} &= u(\text{win}) \cdot P(\text{win} \mid Ag^{st}, Env) \\
&\quad +u(\text{lose}) \cdot P(\text{lose} \mid Ag^{st}, Env) \\
&\quad +u(\text{draw}) \cdot P(\text{draw} \mid Ag^{st}, Env) \\
&= 1 \cdot (p \cdot 0.1 + r \cdot 0.5 + (1-p-r) \cdot 0.4) \\
&\quad +(-1) \cdot (r \cdot 0.4 + p \cdot 0.5 + (1-p-r) \cdot 0.1) \\
&\quad +0 \cdot (r \cdot 0.1 + p \cdot 0.4 + (1-p-r) \cdot 0.5) \\
&= 0.1p + 0.5r + 0.4 - 0.4p - 0.4r \\
&\quad -(0.4r + 0.5p + 0.1 - 0.1p - 0.1r) \\
&= 0.1p + 0.5r + 0.4 - 0.4p - 0.4r \\
&\quad -0.4r - 0.5p - 0.1 + 0.1p + 0.1r \\
&= 0.3 + (0.1 - 0.4 - 0.4 + 0.1)p + (0.5 - 0.4 - 0.4 + 0.1)r \\
&= 0.3 - 0.7p - 0.2r
\end{aligned}$$

Since the expected utility is decreasing in both p and r, the optimal strategy is $p = 0$ and $r = 0$ (i.e., $Ag^{st}(e_0) = \{(S,1)\}$).

Tutorial questions: topic 3

1. Write each of the following statements as a temporal logic formula.

   (a) If a person never dies then they are immortal. (Use the predicates $dies(person)$ and $immortal(person)$.)

   (b) Unless you study agents, you won't know how great they are. (Use the predicates $study(agents, you)$ and $knowGreat(agents, you)$.)

   Answer

   Note, as is often the case with logic, there may be other correct answers.

   (a) If a person never dies then they are immortal. (Use the predicates $dies(person)$ and $immortal(person)$.)

   $$(\blacksquare\neg dies(person) \wedge \Box\neg dies(person)) \rightarrow immortal(person)$$

   This says "If it is true at every point in the past that a person never dies and it is true now and at all points in the future that a person never dies, then the person is immortal".

   Further discussion about this question

   Here is another answer that students have suggested in previous years that isn't quite right.

   $$\Box\neg dies(person) \rightarrow immortal(person)$$

   This says "If there's no point in the future where a person dies then they are now immortal". This isn't quite right, since this means that someone who died last year is considered immortal.

   Similarly, the answer below isn't quite right.

   $$immortal(person)\mathcal{W}dies(person)$$

   This says "A person is immortal at every point in the future unless they die". This isn't quite right, since this means that someone who dies next year is immortal until the point that they die.

   Students in previous years have asked: is the following formula correct/do we need the $\Box$ on the right of the implication?

   $$(\blacksquare\neg dies(person) \wedge \Box\neg dies(person)) \rightarrow \Box immortal(person)$$

   This says "If it is true that the person never dies in the past, the person doesn't die now and never dies in the future, then it is true that now and at all points in the future the person is immortal." This is correct, but the $\Box$ on the right of the implication (in front of $immortal(person)$) isn't necessary here.

   To understand why the $\Box$ isn't necessary on the right of the implication, consider again our original solution:

$$(\blacksquare\neg dies(person) \wedge \Box\neg dies(person)) \rightarrow immortal(person)$$

At any point where the left side of the implication is true, the right side of the implication is true. And if the left side of the implication is true at any point, then it must be true at all points: the left side essentially says that it is true at all points in the past and future that $\neg dies(person)$, if this is true at any time point then it must be true at all time points.

According to the formula above, if there is ever a point where the person never dies in the past and never dies now or in the future, then they are immortal at that point. But we have just seen that if there is ever a point where the person never dies in the past and never dies now or in the future then it must be true that at all points the person never dies in the past and never dies now or in the future, thus our implication must be satisfied at each time point. Hence the implication above is sufficient to infer that, if there is ever a point where the person never dies in the past and never dies now or in the future, then the person is immortal at all time points in the future (and in fact in the past) and this is why we don't need to add the $\Box$ in front of $immortal(person)$.

(b) Unless you study agents, you won't know how great they are. (Use the predicates $study(agents, you)$ and $knowGreat(agents, you)$.)

$$\blacklozenge(\blacksquare\neg knowGreat(agents, you) \wedge \neg knowGreat(agents, you)\mathcal{W}study(agents, you))$$

This says "It is true at some point in the past that: (1) at every point before that you didn't know how great agents were, and (2) it will always be true going forward that you don't know how great agents are, unless there comes a point in the future where you study agents".

Further discussion about this question

What about the following?

$$\neg knowGreat(agents, you)\mathcal{W}study(agents, you)$$

This says "it will always be true that you don't know how great agents are, unless there comes a point in the future where you study agents". This doesn't allow for the fact that someone might have studied agents in the past.

What about the following?

$$\Diamond(knowGreat(agents, you)\mathcal{S}study(agents, you))$$

This says "It will be true at some point in the future that at some point in the past you studied agents and you have known how great they are ever since.". This isn't quite right, since it doesn't allow for the fact that you might never study agents.

What about the following?

$$\neg study(agents, you) \rightarrow \Box\neg knowGreat(agents, you)$$

This says "If it's not true (now) that you study agents, then you will never know how great they are". This isn't quite right, since it doesn't allow for the fact that you might study agents in the future.

2. The following specification is adapted from a famous "Snow White" example in Concurrent MetateM:

$$
\begin{array}{ll}
snowWhite(ask(X))[give(X)]: & \\
& \circledcirc ask(X) \rightarrow \lozenge give(X) \\
& start \rightarrow \Box\neg(give(X) \wedge give(Y) \wedge X \neq Y)
\end{array}
$$

$$
\begin{array}{ll}
eager(give(eager))[ask(eager)]: & \\
& start \rightarrow ask(eager) \\
& \circledcirc give(eager) \rightarrow ask(eager)
\end{array}
$$

$$
\begin{array}{ll}
greedy()[ask(greedy)]: & \\
& start \rightarrow \Box ask(greedy)
\end{array}
$$

[Note that X and Y are variables. An agent that listens for messages of type $give(X)$ will hear all messages of the form $give(X)$ (where X can be instantiated to anything) (same for an agent who listens for messages of type $ask(X)$). For example, Snow White will listen to messages $ask(eager)$ and $ask(greedy)$.]

Trace a run of this system (i.e. give a run of the system) for the first 6 time steps (use a table like the one we saw in the example from the core material).

Answer

Note, this is only one possible run of the system. Others are possible, depending on the order that snowWhite chooses to satisfy her commitments.

$snowWhite$		$eager$		$greedy$	
	$\Box\neg(give(X) \wedge give(Y) \wedge X \neq Y)$	$ask(eager)$		$ask(greedy)$	$\Box ask(greedy)$
$ask(eager)$, $ask(greedy)$				$ask(greedy)$	
$ask(greedy)$, $give(eager)$	$\lozenge give(eager)$, $\lozenge give(greedy)$			$ask(greedy)$	
$ask(greedy)$, $give(greedy)$	$\lozenge give(greedy)$	$give(eager)$		$ask(greedy)$	
$ask(greedy)$, $give(greedy)$	$\lozenge give(greedy)$	$ask(eager)$		$ask(greedy)$	
$ask(greedy)$, $ask(eager)$ $give(greedy)$	$\lozenge give(greedy)$			$ask(greedy)$	

Timestep 1:

snowWhite's second rule fires, causing her to add a commitment: from now and every point forward, she cannot give to two different agents at the same time point.

Eager's first rule fires, and so it updates its propositions to include ask(eager) (which is broadcast, since this is something that Eager broadcasts).

Greedy's rule fires, and so it updates its commitments to include that it will always ask, and adds ask(greedy) to its propositions to satisfy this commitment (which is broadcast, since this is a message type that Greedy broadcasts).

Timestep 2:

SnowWhite's environment propositions are updated to reflect that she has received the broadcasts of eager's and greedy's asks in the previous timestep. None of its rules fire, since there is no ask(X) true in its propositions for the previous timestep.

Eager: only listens out for give(eager) messages, and none of these were broadcast in the previous timestep. None of its rules fire, since give(eager) does not appear in its propositions for the previous timestep.

Greedy: doesn't listen for any messages, and no rules fire. It broadcasts an ask(greedy) message to satisfy its commitment from the first timestep (note that this commitment persists, since it is a $\Box$ commitment).

Timestep 3:

SnowWhite's environment propositions are updated to reflect that she has received the broadcast of greedy's ask in the previous timestep; her first rule fires twice, generating the commitment to give to eager at some point now or in the future and the commitment to give to greedy now or at some point in the future. Now, snowWhite cannot satisfy both of these commitments at once, becuase of her commitment from the first time step. Since neither commitment is older than the other, she is free to select either of these. Let's say she selects to give to eager, and so she adds give(eager) to her propositions, satisfying the commitment to give now or at some point in the future to eager.

Eager: no messages received, no rules fire.

Greedy: no rules fire, broadcasts an ask message to satisfy its commitment from first timestep.

Timestep 4:

SnowWhite's environment propositions are updated to reflect that she has received the broadcast of greedy's ask in the previous timestep; her first rule fires generating the commitment to give to greedy at some point now or in the future. SnowWhite also sends a give to greedy message to satisfy both her commitment from timestep 3 and her new commitment from this timestep.

Eager: updates its propositions to reflect that it received the give(eager) that snowWhite broadcast in the previous timestep. No rules fire.

Greedy: no rules fire, broadcasts an ask message to satisfy its commitment from first timestep.

Timestep 5:

SnowWhite's environment propositions are updated to reflect that she has received the broadcast of greedy's ask in the previous timestep; her first rule fires generating the commitments to give to greedy now or at some point in the future; she sends a give to greedy message to satisfy this commitment.

Eager: no new messages received. Eager's second rule fires and so it broadcasts an ask(eager) message.

Greedy: no rules fire; broadcasts an ask message to satisfy its commitment from timestep 1.

Timestep 6:

SnowWhite's environment propositions are updated to reflect that she has received the broadcast of greedy's ask and the broadcast of eager's ask in the previous timestep; her first rule fires generating the commitment to give to greedy now or at some point in the future and she broadcasts a give greedy message to satisfy this commitment.

Eager: no messages received and no rules fire.

Greedy: no rules fire; broadcasts an ask message to satisfy its commitment from timestep 1.

Additional discussion about this question

In previous years, students had questions about whether it is possible with this example that snowWhite never ends up giving to eager.

Consider the run of the system below. This is the same as the run given above, but extended by three timesteps.

$snowWhite$		$eager$		$greedy$	
	$\Box\neg(give(X) \wedge give(Y) \wedge X \neq Y)$	$ask(eager)$		$ask(greedy)$	$\Box ask(greedy)$
$ask(eager)$, $ask(greedy)$				$ask(greedy)$	
$ask(greedy)$, $give(eager)$	$\lozenge give(eager)$, $\lozenge give(greedy)$			$ask(greedy)$	
$ask(greedy)$, $give(greedy)$	$\lozenge give(greedy)$	$give(eager)$		$ask(greedy)$	
$ask(greedy)$, $give(greedy)$	$\lozenge give(greedy)$	$ask(eager)$		$ask(greedy)$	
$ask(greedy)$, $ask(eager)$ $give(greedy)$	$\lozenge give(greedy)$			$ask(greedy)$	
$ask(greedy)$, $give(greedy)$	$\lozenge give(greedy)$ $\lozenge give(eager)$			$ask(greedy)$	
$ask(greedy)$, $give(eager)$	$\lozenge give(greedy)$			$ask(greedy)$	
$ask(greedy)$, $give(greedy)$	$\lozenge give(greedy)$	$give(eager)$		$ask(greedy)$	

Timestep 7:

SnowWhite's environment propositions are updated to reflect that she has received the broadcast of greedy's ask made in the previous timestep. Her first rule fires twice, giving two new commitments at this timestep: to give to greedy now or at some point in the future and to give to eager now or at some point in the future. She cannot satisfy both of these commitments at once, and since neither is older than the other she selects one at random to satisfy and decides to give to greedy at this time step. (Note, at this timepoint, all of the earlier $\lozenge$ commitments are satsified, so she can choose either $\lozenge give(greedy)$ or $\lozenge give(eager)$ to satisfy at this timestep.)

Eager: no messages received and no rules fire.

Greedy: no rules fire; broadcasts an ask message to satisfy its commitment from timestep 1.

Timestep 8:

SnowWhite's environment propositions are updated to reflect that she has received the broadcast of greedy's ask made in the previous timestep. Her first rule fires generating a commitment to give to greedy now or at some point in the future. It is important to note that snowWhite has an older unsatsisfied commitment than this: the commitment from the previous timestep to give now or at some point in the future to eager. Since agents are obliged to give precedence to older unsatisfied commitments, snowWhite does not have a choice here but has to give to eager at this timestep in order to satisfy this older commitment.

Eager: no messages received and no rules fire.

Greedy: no rules fire; broadcasts an ask message to satisfy its commitment from timestep 1.

Timestep 9:

SnowWhite's environment propositions are updated to reflect that she has received the broadcast of greedy's ask made in the previous timestep. Her first rule fires generating a commitment to give to greedy now or at some point in the future. Her oldest unsatisfied commitment at this point is the $\lozenge give(greedy)$ from the previous timestep, so she must now give to greedy (which also satisfies the $\lozenge give(greedy)$ from the current timestep.)

Eager: updates propositions to reflect receipt of the $give(eager)$ message that was broadcast in previous timestep; no rules fire.

Greedy: no rules fire; broadcasts an ask message to satisfy its commitment from timestep 1.

I hope you can see from the above why snowWhite will have to give to eager at some point. (Note that eager will always broadcast an ask in the first timestep, so snowWhite will always have a commitment at the third timestep to give to eager now or at some point in the future. This will, either in the third or fourth timestep, mean that snowWhite gives to eager to satisfy this commitment. This will trigger eager's rule to ask, and so on.)

Tutorial questions: topic 4

1. The pseudo-code below defines the control loop for `agent1`, a practical reasoning agent.

```
B := B0;          /* B0 is initial value of B */
I := I0;          /* I0 is initial value of I */
while true do
      get next percept ρ;
      B:= brf(B, ρ);
      D:= options(B, I);
      I:= filter(B, D, I);
      π:= plan(B, I);
      while not (empty(π) or succeeded(I, B)
                   or impossible(I, B)) do
            α:= hd(π);
            execute(α);
            π:= tail(π);
            get next percept ρ;
            B:= brf(B, ρ);
            if reconsider(I, B) then
                 D:= options(B, I);
                 I:= filter(B, D, I);
            end-if
            if not sound(π, I, B) then
                 π:= plan(B, I);
            end-if
       end-while
 end-while
```

(a) With reference to the pseudo-code that defines `agent1` identify and explain the role of the variables and functions that ensure the *deliberation* of the agent and the *means-ends reasoning* of the agent.

Deliberation is carried out via two functions. The `options` *function takes the agent's current beliefs* `B` *and current intentions* `I`*, and on the basis of these produces a set of desires* `D`*. The* `filters` *function selects from these competing desires* `D`*, taking into account its current beliefs* `B` *and current intentions* `I` *and returning a new set of intentions.*

Means-ends reasoning is carried out via the `plan` *function, which, on the basis of the current beliefs* `B` *and current intentions* `I`*, determines a plan* π *to achieve the current intentions* `I`*.*

(b) One of the aims of the pseudo-code that defines `agent1` is to ensure that the agent is committed to plans just as long as it is rational to be committed to them. With reference to the pseudo-code, explain how the control loop achieves this. In your answer, be sure to clearly identify all the circumstances under which an agent reconsiders its plan.

After each action that is performed (line 11) the agent will check the environment (line 13), update its beliefs (line 14), possibly reconsider its intentions (lines 15 - 17) and then if its plan is no longer sound (line 19) it will replan (line 20).

The agent will also replan if it finds at any point that its intentions are already succeeded or are impossible (line 9). If this is the case it will observe the environment again and update its beliefs (lines 4 and 5), reconsider its intentions (lines 6 and 7) and replan (line 8).

(c) Another aim of the pseudo-code that defines `agent1` is to ensure that the agent is committed to intentions as long as it is rational to be committed to them. With reference to the pseudo-code, explain how the control loop achieves this. In your answer, be sure to clearly identify all the circumstances under which an agent reconsiders its intentions.

After each action that is performed (line 11) the agent will check the environment (line 13), update its beliefs (line 14), check based on these new beliefs and its intentions whether it should reconsider its intentions (line 15, if reconsider(I,B) returns true) and if so will reconsider its intentions.

The agent will also reconsider its intentions if it finds at any point that they are already succeeded or impossible (line 9). If this happens it will observe the environment again and update its beliefs (lines 4 and 5), and then reconsider its intentions (lines 6 and 7).

2. Using examples to illustrate your answer, explain how an agent's behaviour is affected by its intentions and the attitudes you can expect an agent to have towards its intentions. The examples you use can be about human agents or artificial agents.

Agent will devote resources into trying to work out how to bring about its intention (means end reasoning). Agent will devote resources to trying to achieve its intentions — it doesn't just give up if attempts fail and there are alternative ways it can try. It won't adopt inconsistent intentions. It won't have an intention that it believes is impossible or inevitable, but also agents are conscious that they may not achieve their intentions. Agents don't necessarily need to intend all the expected side effects of their intentions.

For example, if a student has an intention to have an academic career, you would expect them to spend some resources on working out how they're going to try to achieve it (apply for phd positions, get a phd, make contacts, write papers, apply for post doc positions, apply for fellowships etc). If the student doesn't get a particular job they apply for you wouldn't expect them to immediately give up on their intention to have an academic career. Instead you would expect them to apply for other jobs. There's no point me having an intention to have an academic career, since I already have one. Equally, it wouldn't be rational for someone in their 90s who has worked in the fashion industry all their life to have an intention to have a academic career in science, since this is clearly not achievable. A student wouldn't have an intention both to have an academic career and a career in the fashion industry, since you can't simultaneously do both. And it's rational for me to intend to travel up to stay with my family over xmas even though I believe that journeys around xmas are always busy and horrible, and I don't have an intention to suffer through a horrid busy journey.

Tutorial questions: topic 5

1. Think of a problem that you think would be well suited to collaborative distributed problem solving.

   (a) Explain how you would decompose the problem into sub-tasks.

   Search and rescue in a collapsed building. There is a controller agent and a range of different agents with different capabilities: small robots with heat sensors and infrared; small robots with cameras; large robots that are able to move debris and people; robots that are able to cut through debris. The controller agent has a map of the building and splits the overall task into smaller tasks to explore the different rooms of the building and rescue any casualties. For each of these sub-tasks (to search and rescue in one room), the controller will break it down into even smaller sub-tasks, that need to be performed sequentially:

   - *explore the room in order to: identify and locate any casualties; identify and locate any obstacles;*
   - *remove any obstacles;*
   - *rescue any casualties.*

   (b) Take one of the sub-tasks from the problem you have described for question 1a and assume that an agent is using the Contract Net protocol to identify a contractor for the sub-task. When making the task announcement, explain

   i. what information the agent would ask to be included in bids and any constraints it would provide in its task announcement; and

   ii. how it would evaluate the bids it receives.

   Taking the sub-task: explore the room in order to: identify and locate any casualties; identify and locate any obstacles.

   The agent would include the constraints that: bids must be received within 3 minutes; bidding agents must have either an infrared sensor, a heat sensor or a camera; and bidding agents must be less than 1m tall. It will ask the agents to include in their bids: details of the different sensing equipment they have, including the range of that equipment; the amount of battery the agent has remaining.

   To evaluate the bids, the agent will ideally want a robot that has the most amount of sensors with the largest range, and with the most amount of battery. There may be trade-offs to be considered (e.g., the agent with the most battery may not have the best sensors). If it turns out that there is no single agent with a good set of sensors available, the controller may decide to allocate the task to multiple agents.

2. Give an example of a pair of actions that should be coordinated (this should be different from the examples in the core material). Explain why these actions should be coordinated.

   I want to order some equipment for working at home. A precondition of this is that I have to have approval from my head of department. So I request my head of department to achieve this precondition. I need my head of department's approval in order to achieve my goal of having equipment for working at home, and I cannot achieve this on my own. I have to request that my head of department performs this action.

3. Give an example of prescriptive norm (different from the ones in the core material) and give three different reasons why an agent might choose to violate the norm. $\langle \textit{all-people}, \textit{obliged}, \textit{in-UK-shop}, wear\text{-}face\text{-}mask, fine\text{-}100\rangle$

   All people are obliged to wear a face mask when they're in a shop in the UK, and if they get caught whiteout one then they will be fined £100.

   Reasons they might violate this:

   They don't care if they get a fine.

   They find wearing masks so uncomfortable that they're willing to risk the fine.

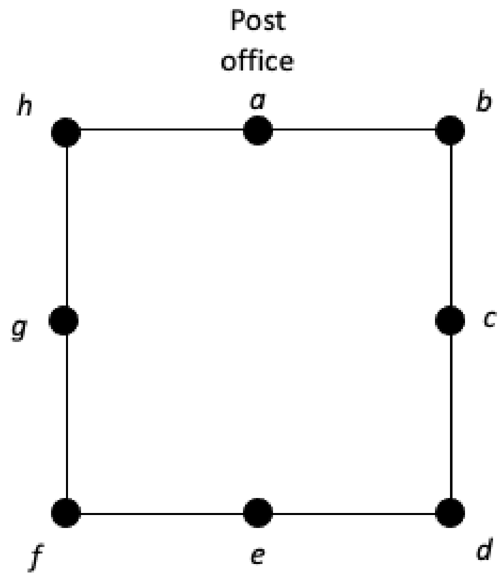
   They are exempted from the obligation because they have a medical condition. This means there is a permission norm as follows:

   $\langle \textit{all-people}, \textit{permitted}, \textit{have-medical-condition}, \textit{not } wear\text{-}face\text{-}mask, \_\rangle$

Tutorial questions: topic 6

Consider the following task oriented domain. There are two postal workers $Ag1$ and $Ag2$ who each start at the post office and must deliver some letters to particular houses. So a task relates to a particular letter and is represented by the house it must be delivered to. The cost of a set of tasks is the minimum distance the postal workers must travel to visit each of the houses he/she must deliver to and to return to the post office.

1. A specific scenario is shown in the figure below. We see that there are eight possible houses that the agents may need to visit (the nodes in the figure: $a, b, \ldots, h$) and the post office is at the same location as house a. We assume that the distance between each node is 1 (e.g., the minimum distance from a to b is 1, the minimum distance from a to g is 2).



   (a) Consider we have the following initial encounter: $\langle \{b, f\}, \{e\} \rangle$
   So there are three tasks that need performing (to deliver a letter to b, e, and f). Initially, Ag_1 has been allocated the tasks to deliver to both b and f, and Ag_2 has been allocated the task to deliver to e. The cost of this encounter to Ag_1 is 8, and the cost of this encounter to Ag_2 is also 8. (Remember, the cost of a set of tasks is the minimum distance an agent must cover in order to travel from the post office, visit each of the houses, and return to the post office.)

      i. What are all the possible deals for this scenario?
      $(\{\}, \{b, e, f\})$, $(\{b\}, \{e, f\})$, $(\{e\}, \{b, f\})$, $(\{f\}, \{b, e\})$, $(\{b, e\}, \{f\})$, $(\{b, f\}, \{e\})$, $(\{e, f\}, \{b\})$, $(\{b, e, f\}, \{\})$

      ii. Which deals are in the negotiation set?
      $(\{\}, \{b, e, f\})$, $(\{b, e, f\}, \{\})$

      iii. What happens if the agents each use the Zeuthen strategy with the monotonic concession protocol to try to redistribute their tasks? What is the *expected utility* that each agent can expect to gain from this negotiation? (Reminder: expected utility is equal to the summation over all possible outcomes of the probability that each outcome will occur times the utility of that outcome.) (Reminder: if each agent is equally willing to risk the conflict deal, flip a fair coin to decide who should make a concession.)

First round: Ag_1 proposes $(\{\}, \{b, e, f\})$; Ag_2 proposes $(\{b, e, f\}, \{\})$.
Ag_1's willingness to risk the conflict deal: $\frac{8-0}{8} = 1$.
Ag_2's willingness to risk the conflict deal: $\frac{8-0}{8} = 1$.
Since each equally willing to risk conflict, flip a coin to decide who concedes: 0.5 probability that Ag_1 will concede, 0.5 probability that Ag_2 will concede. Whoever concedes will match the other agent's proposal in the next round, so there is 0.5 probability they will agree on $(\{\}, \{b, e, f\})$ and 0.5 probability they will agree on $(\{b, e, f\}, \{\})$.
This means that each agent has an expected utility of $(0.5 \times 0) + (0.5 \times 8) = 4$.

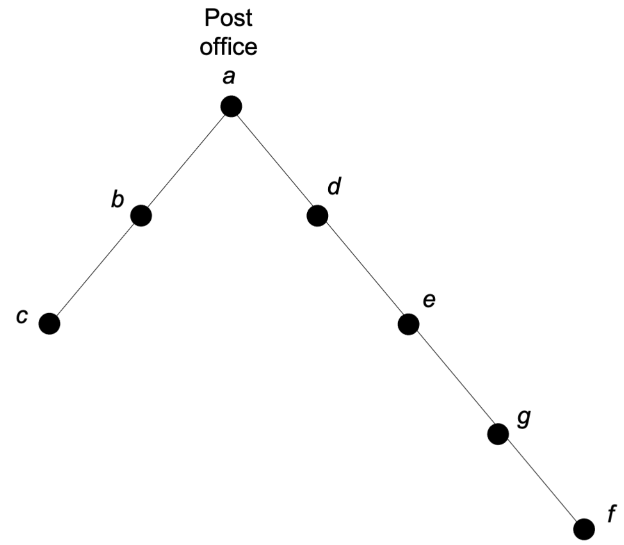
(b) Now consider that $Ag1$ deceives $Ag2$ by hiding the fact that it has been allocated the task to deliver to b in the initial allocation.

The (untruthful) initial encounter is thus presented as $\langle\{f\}, \{e\}\rangle$, and so the agents will negotiate over the tasks e and f. Whatever the result of the negotiation is, $Ag1$ will also have to perform task b (since this task has been allocated to $Ag1$ in the truthful initial encounter, and $Ag1$ has no chance of giving it away since it has hidden this task from the other agent).

i. Which deals are in the negotiation set, given $Ag1$'s deception here?
$(\{\}, \{e, f\})$

ii. What happens if the agents each use the Zeuthen strategy with the monotonic concession protocol to try to redistribute their tasks? What is the expected utility that each agent can expect to gain from this negotiation? (Hint: don't forget that $Ag1$ will also have to perform the task that it hid, to deliver to b.)
Since there is only one member of the negotiation set, both agents will propose this in the first round and they will agree to: $(\{\}, \{e, f\})$.
Thus, after the negotiation, Ag_1 will have to carry out task b, and Ag_2 will need to carry out tasks e and f. The expected utility of this to Ag_1 is 6, the expected utility of this to Ag_2 is 0.

2. A different postal office domain scenario is shown in the figure below. Here there are six possible houses that the agents may need to visit. Again, we assume that the distance between each node is 1.



(a) Consider we have the following initial encounter: $\langle\{c, d\}, \{c, d\}\rangle$.

So there are four letters that need to be delivered (two to house c and two to house d) and in the initial allocation each agent has to visit both house c and house d. The cost of this encounter to Ag_1 is 6, and the cost of this encounter to Ag_2 is also 6.

i. Which deals are in the negotiation set?
$(\{c, d\}, \{\})$, $(\{c\}, \{d\})$, $(\{d\}, \{c\})$, $(\{\}, \{c, d\})$

ii. What happens if the agents each use the Zeuthen strategy with the monotonic concession protocol to try to redistribute their tasks? What is the *expected utility* that each agent can expect to gain from this negotiation? (Reminder: expected utility is equal to the summation over all possible outcomes of the probability that each outcome will occur times the utility of that outcome.) (Reminder: if each agent is equally willing to risk the conflict deal, flip a fair coin to decide who should make a concession.)
First round: Ag_1 proposes $(\{\}, \{c, d\})$; Ag_2 proposes $(\{c, d\}, \{\})$.
Ag_1's willingness to risk the conflict deal: $\frac{6-0}{6} = 1$.
Ag_2's willingness to risk the conflict deal: $\frac{6-0}{6} = 1$.
Since each equally willing to risk conflict, flip a coin to decide who concedes: 0.5 probability that Ag_1 will concede, 0.5 probability that Ag_2 will concede.
Let's assume that Ag_1 loses the coin toss and has to concede. The smallest concession that Ag_1 can make is to say it will take on task d. This would give the following values of the agents' willingness to risk the conflict deal (if Ag_1 concedes to $(\{d\}, \{c\})$ and Ag_2 sticks with their proposal $(\{c, d\}, \{\})$).
For Ag_1: $\frac{4-0}{4} = 1$. For Ag_2: $\frac{6-2}{6} = \frac{2}{3}$. So this is enough to change the balance of risk, and Ag_2 will then need to concede.
The smallest concession that Ag_2 can make is to say it will take on task d. This would give the following values of the agents' willingness to risk the conflict deal (if Ag_1 sticks with their proposal of $(\{d\}, \{c\})$ and Ag_2 concedes to $(\{c\}, \{d\})$).

For Ag_1: $\frac{4-2}{4} = \frac{1}{2}$. For Ag_2: $\frac{4-2}{4} = \frac{1}{2}$. So this is enough to change the balance of risk, and they need to flip a coin to see who concedes next.
Whoever loses the coin toss will need to concede to the proposal on offer by the other agent, so there is a 0.5 chance they will end up agreeing to $(\{a\}, \{b\})$ and a 0.5 chance they will end up agreeing to $(\{b\}, \{a\})$. Since the situation is symmetric, we get the same thing if we assume that Ag_2 loses the initial coin toss.
The expected utility to each agent is $(0.5 \times 2) + (0.5 \times 4) = 3$.

(b) Now consider that $Ag1$ deceives $Ag2$ by pretending that it has also been allocated a task to deliver a letter to house f in the initial allocation.

The (untruthful) initial encounter is thus presented as $\langle\{c, d, f\}, \{c, d\}\rangle$.

i. Which deals are in the negotiation set, given $Ag1$'s deception here?
$(\{c, d, f\}, \{\})$, $(\{d, f\}, \{c\})$

ii. What happens if the agents each use the Zeuthen strategy with the monotonic concession protocol to try to redistribute their tasks? What is the expected utility that each agent can expect to gain from this negotiation? (Hint: whatever the result of the negotiation is, no agent will actually have to visit house f since this task does not actually exist.)
First round: Ag_1 proposes $(\{d, f\}, \{c\})$; Ag_2 proposes $(\{c, d, f\}, \{\})$.
Ag_1's willingness to risk the conflict deal: $\frac{4-0}{4} = 1$.
Ag_2's willingness to risk the conflict deal: $\frac{6-2}{6} = \frac{2}{3}$.
Ag_2 has to concede and they end up agreeing to $(\{d, f\}, \{c\})$.
The expected utility to Ag_1 is 4. The expected utility to Ag_2 is 2.

Tutorial questions: making group decisions

1. Consider the following voter profile representing the *truthful* preferences of Alice, Bob, and Charlie:

Alice:	$B \succ A \succ C \succ D$
Bob:	$C \succ A \succ B \succ D$
Charlie:	$B \succ A \succ C \succ D$

Alice, Bob, and Charlie decide to hold a vote which will be decided according to the *Borda count* protocol, with ties decided randomly.

Is it in anyone's interest to vote strategically and *falsely* report their preference (assuming they know the truthful preferences of the other two voters)? If your answer is yes, then specify the person as well as the ordering they should report instead of their true preference.

Hint. A voter v has an incentive to vote strategically if and only if it is the case that:

- the winner if they submit their true preferences is ω_x
- the winner if they submit some false preferences is ω_y, and
- v prefers ω_y to ω_x

Answer. If all parties report their true preferences the winner is: B.

$$\begin{aligned} A &= (3 \times 0) + (2 \times 3) + (1 \times 0) = 6 \\ B &= (3 \times 2) + (2 \times 0) + (1 \times 1) = 7 \\ C &= (3 \times 1) + (2 \times 0) + (1 \times 2) = 5 \\ D &= (3 \times 0) + (2 \times 0) + (1 \times 0) = 0 \end{aligned}$$

Alice and Charlie want this outcome and so don't have an incentive to vote strategically.

Bob would prefer C or A to B. He could falsely report his preferences as $A \succ C \succ D \succ B$, which would give the following, meaning A would win:

$$\begin{aligned} A &= (3 \times 1) + (2 \times 2) + (1 \times 0) = 7 \\ B &= (3 \times 2) + (2 \times 0) + (1 \times 0) = 6 \\ C &= (3 \times 0) + (2 \times 1) + (1 \times 2) = 4 \\ D &= (3 \times 0) + (2 \times 0) + (1 \times 1) = 1 \end{aligned}$$

It is in Bob's interest to falsely report: $A \succ C \succ D \succ B$.

2. The plurality voting protocol is an example of a positional scoring rule voting protocol. What is the scoring vector that captures the plurality procedure?

Answer. The plurality voting protocol is a positional scoring rule voting protocol with the scoring vector $(1, 0, ..., 0)$. That is, a candidate only receives a point from a particular voter if that voter has listed the candidate in first place.

3. Consider the following voter profile:

3 voters:	$A \succ B \succ C$
6 voters:	$B \succ C \succ A$
4 voters:	$C \succ A \succ B$
3 voters:	$C \succ B \succ A$

Suppose that (s_1, s_2, s_3) is an arbitrary *scoring vector*, where $s_1 \geq s_2 \geq s_3 \geq 0$ and $s_1 > s_3$. Show that the corresponding *positional scoring rule* does not select the Condorcet winner.

Hint. This is essentially an algebraic manipulation question. You can represent the scores received by each of the three candidates in terms of s_1, s_2 and s_3. Let $score(\omega_i)$ be the score received by candidate ω_i. If you can show that $score(\omega_i) - score(\omega_j) < 0$ then you know that $score(\omega_i) < score(\omega_j)$. If you can show that $score(\omega_i) - score(\omega_j) \leq 0$ then you know that $score(\omega_i) \leq score(\omega_j)$. Don't forget that you know $s_1 \geq s_2 \geq s_3 \geq 0$ and $s_1 > s_3$.

Answer. To determine the Condorcet winner, we have to consider pairwise majority elections.

- A vs B. A gets 7. B gets 9. B wins.
- A vs C. A gets 3. C gets 13. C wins.
- B vs C. B gets 9. C gets 7. B wins.

B is the Condorcet winner.

Under the positional scoring rule protocol, we get:

$score(A) = 3s_1 + 4s_2 + 9s_3$

$score(B) = 6s_1 + 6s_2 + 4s_3$

$score(C) = 7s_1 + 6s_2 + 3s_3$

Now let's manipulate this algebraically to determine the winner under the positional scoring rule protocol.

$$\begin{aligned} score(A) - score(B) &= -3s_1 - 2s_2 + 5s_3 \\ &= 3(s_3 - s_1) + 2(s_3 - s_2) \end{aligned}$$

Since we know that $s_1 > s_3$, we know that $s_3 - s_1 < 0$, hence $3(s_3 - s_1) < 0$.

Since we know that $s_1 \geq s_2 \geq s_3 \geq 0$, we know that $s_3 - s_2 \leq 0$, hence $2(s_3 - s_2)0$.

It follows that $3(s_3 - s_1) + 2(s_3 - s_2) < 0$.

So $score(A) - score(B) < 0$, therefore $score(A) < score(B)$.

$$score(C) - score(B) = s_1 - s_3$$

Since we know that $s_1 > s_3$, we know that $s_1 - s_3 > 0$.

So $score(C) - score(B) > 0$, therefore $score(C) > score(B)$.

Hence $score(C) > score(B) > score(A)$ and C wins under the positional scoring rule protocol with any arbitrary scoring vector (s_1, s_2, s_3) where $s_1 \geq s_2 \geq s_3 \geq 0$ and $s_1 > s_3$.

As the Condorcet winner for this example is B, we have shown that the positional scoring rule protocol violates the Condorcet principle with any scoring vector (s_1, s_2, s_3) where $s_1 \geq s_2 \geq s_3 \geq 0$ and $s_1 > s_3$.

There are other ways you might have approached this. For example, instead of determining $score(C) - score(B)$, you might have done:

$$\begin{aligned} score(A) - score(C) &= -4s_1 - 2s_2 + 6s_3 \\ &= 4(s_3 - s_1) + 2(s_3 - s_2) \end{aligned}$$

Since we know that $s_1 > s_3$, we know that $s_3 - s_1 < 0$, hence $4(s_3 - s_1) < 0$.

Since we know that $s_1 \geq s_2 \geq s_3 \geq 0$, we know that $s_3 - s_2 \leq 0$, hence $2(s_3 - s_2)0$.

It follows that $4(s_3 - s_1) + 2(s_3 - s_2) < 0$

So $score(A) - score(C) < 0$, therefore $score(A) < score(C)$.

Tutorial questions: allocating scarce resources

1. Auctions on eBay are ascending price open cry auctions but with a proxy bidding system, which proceeds as follows.

   - The auctioneer sets a reserve price (so at the start of the auction the current highest bid is equal to the reserve price), an increment value, and an ending time.
   - Potential buyers use a proxy bidding system. They submit a maximum value to the proxy bidding system (this must be higher than the current highest bid price).
   - Whenever:
     - the current highest bid plus increment is less than or equal to the submitted maximum value, and
     - the buyer is not the current highest bidder,

     the proxy system will bid one increment more than the current highest bid on behalf of the buyer.
   - The potential buyers are able to increase their maximum value at any point during the auction, as long as it is higher than the current highest bid.
   - At the end of the auction, the proxy bidding system ensures that the bidder who submits the highest maximum value wins the auction and the price they pay is the the second highest maximum value submitted plus the increment. If more than one bidder submits the highest maximum value, the bidder who submitted first will win, and the amount they will pay is their maximum value.

   Consider the following three possible bidder strategies.

   Maxing: At the start of the auction, submit to the proxy bidding system a maximum value that is equal to your true valuation of the item and then wait for the auction to close.

   Increments: If you are not the highest bidder and the current highest bid plus increment value is less than your true valuation of the item, submit to the proxy bidding system a maximum value that is the current auction price plus the increment value.

   Sniping: Wait until the last moment before the end of the auction at which it is possible to submit a new maximum value to the proxy bidding system and then submit to the proxy bidding system a maximum value that is equal to your true valuation of the item.

   Assume there are only two potential buyers for an eBay auction, BuyerA and BuyerB. BuyerA's true valuation of the item is £50 and BuyerB's true valuation of the item is £60. The reserve price is £40 and the increment value is £1. What would be the outcome of the auction in the following situations? Say who would win the auction and what price they would end up paying.

   a) BuyerA uses maxing and BuyerB uses maxing.

   Answer. Both bidders submit their true value at the start of the auction, so BuyerB wins and pays £51

   b) BuyerA uses increments and BuyerB uses sniping.

   Answer. At the start of the auction, BuyerA submits the lowest possible maximum value which is £41. At the last moment, BuyerB submits a maximum value of £60. BuyerB wins and pays £42.

c) BuyerA uses increments and BuyerB uses maxing. (Assume that BuyerA is always first to submit their maximum value.)

Answer. At the start of the auction, BuyerA submits the lowest possible bid as their maxiumum value, which is £41, and BuyerB submits their true value as their maximum value, which is £60. The proxy system will bid £41 on behalf of BuyerA, followed by £42 on behalf of BuyerB. At this point, BuyerA will increase their maximum value to £43, the proxy system will bid £43 on behalf of BuyerA, followed by £44 on behalf of BuyerB. This continues until we reach a bid of £49 on behalf of BuyerA, followed by £50 on behalf of BuyerB. BuyerA is not prepared to go higher than 50 so BuyerB wins and pays £50.

d) BuyerA uses increments and BuyerB uses increments. (Assume that BuyerA is always the first to submit their maximum value.)

Answer. At the start of the auction, BuyerA and BuyerB both submit the lowest possible bid as their maximum value, which is £41. As BuyerA was first, the proxy system will bid £41 on behalf of BuyerA, at which point BuyerB will increase their maximum value to £42. The proxy system will bid £42 on behalf of BuyerB, at which point BuyerA will increase their maximum value to £43 and the proxy system will bid £43 on behalf of BuyerA. This continues until we reach a bid of £50 on behalf of BuyerB. BuyerA is not prepared to go higher than 50 so BuyerB wins and pays £50.

e) BuyerA uses sniping and BuyerB uses maxing.

Answer. At the start of the auction, BuyerB submits their true value as their maximum value, which is £60, and the proxy system bids £41 on behalf of BuyerB. At the last possible minute, BuyerA submits their true value of £50 as their maximum value. BuyerB wins and pays £51 (since the proxy system ensures that the bidder who submits the maximum value wins the auction and the price they pay is the maximum value of the second highest bidder plus the increment).

f) BuyerA uses sniping and BuyerB uses sniping.

Answer. At the last possible moment, BuyerA submits their true value as their maximum, which is £50, and BuyerB submits their true value as their maximum, which is £60. BuyerB wins and pays £51 (since the proxy system ensures that the bidder who submits the maximum value wins the auction and the price they pay is the maximum value of the second highest bidder plus the increment).

2. Suppose we have two goods a and b and two agents ag_1 and ag_2. The agents declare the following true valuation functions:

$$\hat{v}_1(\{a\}) = v_1(\{a\}) = 8$$

$$\hat{v}_1(\{b\}) = v_1(\{b\}) = 7$$

$$\hat{v}_1(\{a, b\}) = v_1(\{a, b\}) = 14$$

$$\hat{v}_2(\{a\}) = v_2(\{a\}) = 6$$

$$\hat{v}_2(\{b\}) = v_2\{b\}) = 4$$

$$\hat{v}_2(\{a, b\}) = v_2\{a, b\}) = 12$$

a) Which allocation is made using the VCG mechanism?

Answer. The allocation that is made is the one that maximises social welfare according to the valuations that were declared: $(\{a, b\}, \emptyset)$.

b) How much does each agent pay to the mechanism?

Answer.

ag_1: If ag_1 had declared a valuation of 0 for every possible allocation, then ag_2 would have been allocated $\{a, b\}$ for total social welfare (excluding ag_1) of 12. The total value received by all agents except ag_1 from the actual allocation is 0 (ag_2 receives nothing in the actual allocation). Thus the amount ag_1 must pay is $p_1 = 12 - 0 = 12$.

ag_2: If ag_2 had declared a valuation of 0 for every possible allocation, then ag_1 would have been allocated $\{a, b\}$ for total social welfare (excluding ag_2) of 14. The total value received by all agents except ag_2 from the actual allocation is 14. Thus the amount ag_2 must pay is $p_2 = 14 - 14 = 0$.

c) What utility does each agent get?

Answer.

ag_1: ag_1 receives $\{a, b\}$, which it values at 14 and it has to pay 12. Thus the utility gained by ag_1 is $14 - 12 = 2$.

ag_2: ag_2 receives $\emptyset$, which it values at 0 and it has to pay 0. Thus the utility gained by ag_2 is $0 - 0 = 0$.

d) Can ag_2 improve its utility by lying and declaring valuation $\hat{v}_2' \neq v_2$ (assuming ag_1 remains truthful)?

Hint. Think about the possible cases that can occur. In the allocation made when the agents submitted their true valuations we had that ag_2 ends up with nothing, what other possible allocations might have been made and how would these have affected the utility gained by ag_2?

Answer. There are three cases to consider. (Note, we don't need to consider the cases where one or more of the items are not allocated, since these cases will never maximise social welfare.)

Case 1, allocation $(\{a\}, \{b\})$ is made. In this case ag_2 would have to pay $p_2 = 14 - 8 = 6$. The utility for ag_2 in this case is thus $4 - 6 = -2$ (since they would receive b which they value at 4 but would have to pay 6 for it). Therefore in this case ag_2 would be worse off.

Case 2, allocation $(\{b\}, \{a\})$ is made. In this case ag_2 would have to pay $p_2 = 14 - 7 = 7$. The utility for ag_2 in this case is thus $6 - 7 = -1$ (since they would receive a which they value at 6 but would have to pay 7 for it). Therefore in this case ag_2 would be worse off.

Case 3, allocation $(\{a, b\}, \emptyset)$ is made. In this case ag_2 would have to pay $p_2 = 14 - 14 = 0$. The utility for ag_2 in this case is thus $0 - 0 = 0$ (since they would receive $\emptyset$ which they value at 0 and would have to pay 0 for it). Therefore in this case ag_2 would be no better or worse off.

Case 4, allocation $(\emptyset, \{a, b\})$ is made. In this case ag_2 would have to pay $p_2 = 14 - 0 = 14$. The utility for ag_2 in this case is thus $12 - 14 = -2$ (since they would receive $\{a, b\}$ which they value at 12 but would have to pay 14 for it). Therefore in this case ag_2 would be worse off.

Thus, for all possible allocations that might be made if ag_2 falsely reports its valuation of the items, ag_2 would be either no better than or worse off than if it reported its true valuation, and so ag_2 cannot benefit from lying about its valuation.

Tutorial questions: reasoning with arguments

1. Below is some knowledge that an agent has about monsters, where X is a variable that can be instantiated by any constant and constants all start with lower-case letters.

   $Angry(cedric)$

   $Strong(cedric)$

   $Sleeping(cedric)$

   $Smiling(cedric)$

   $Smiling(X) \rightarrow \neg Angry(X)$

   $Angry(X) \land Strong(X) \rightarrow RunAwayFrom(X)$

   $Sleeping(X) \rightarrow \neg RunAwayFrom(X)$

   (a) Using the knowledge above, it is possible to construct two arguments that **rebut** one another. Identify two such arguments, giving the support and claim of each.

   Answer:

   Two possible answers:

   $\langle\{Angry(cedric),$
   $Strong(cedric),$
   $Angry(cedric) \land Strong(cedric) \rightarrow RunAwayFrom(cedric)\},$
   $RunAwayFrom(cedric)\rangle$

   and

   $\langle\{Sleeping(cedric),$
   $Sleeping(cedric) \rightarrow \neg RunAwayFrom(cedric)\},$
   $\neg RunAwayFrom(cedric)\rangle$

   rebut each other.

   $\langle\{Angry(cedric)\},$
   $Angry(cedric)\rangle$

   and

   $\langle\{Smiling(cedric),$
   $Smiling(cedric) \rightarrow \neg Angry(cedric)\},$
   $\neg Angry(cedric)\rangle$

   rebut each other.

(b) Using the knowledge above, it is possible to construct two arguments such that one **undercuts** the another. Identify two such arguments, giving the support and the claim of each and making clear which argument undercuts the other.

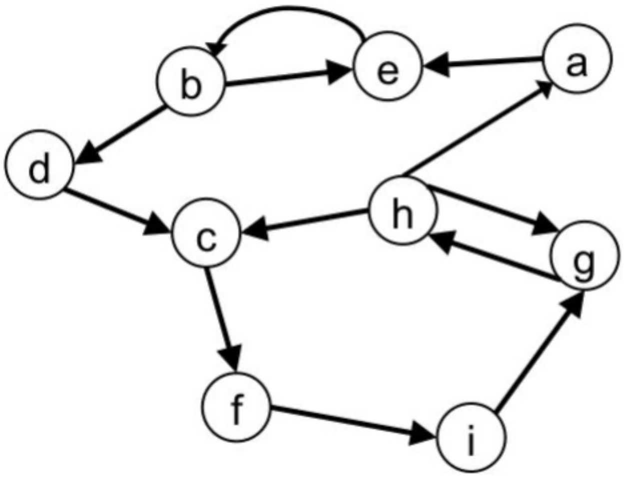
Answer:

$\langle\{Smiling(cedric),$
$Smiling(cedric) \rightarrow \neg Angry(cedric)\},$
$\neg Angry(cedric)\rangle$

undercuts

$\langle\{Angry(cedric),$
$Strong(cedric),$
$Angry(cedric) \wedge Strong(cedric) \rightarrow RunAwayFrom(cedric)\},$
$RunAwayFrom(cedric)\rangle$

2. Consider the following abstract argumentation graph.



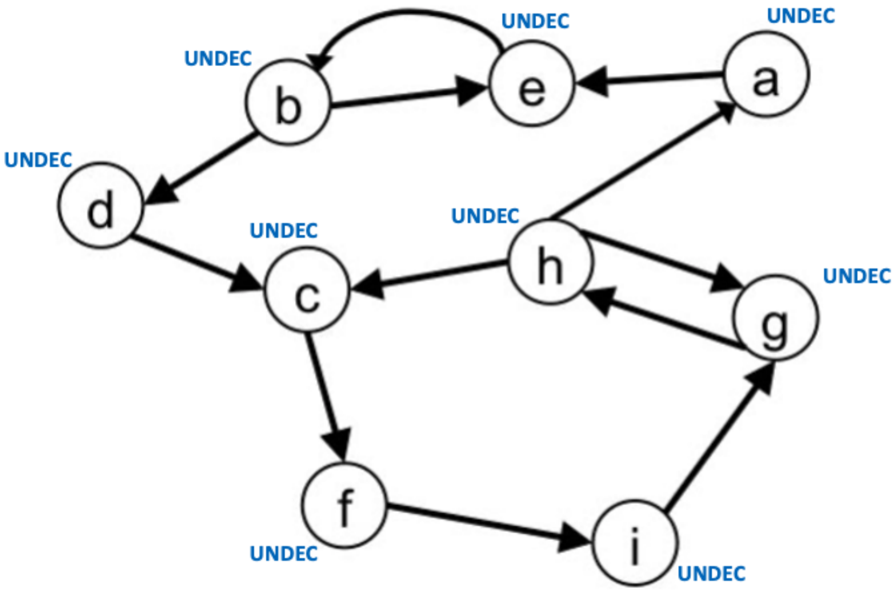
Which arguments are acceptable under:

(a) the grounded semantics;

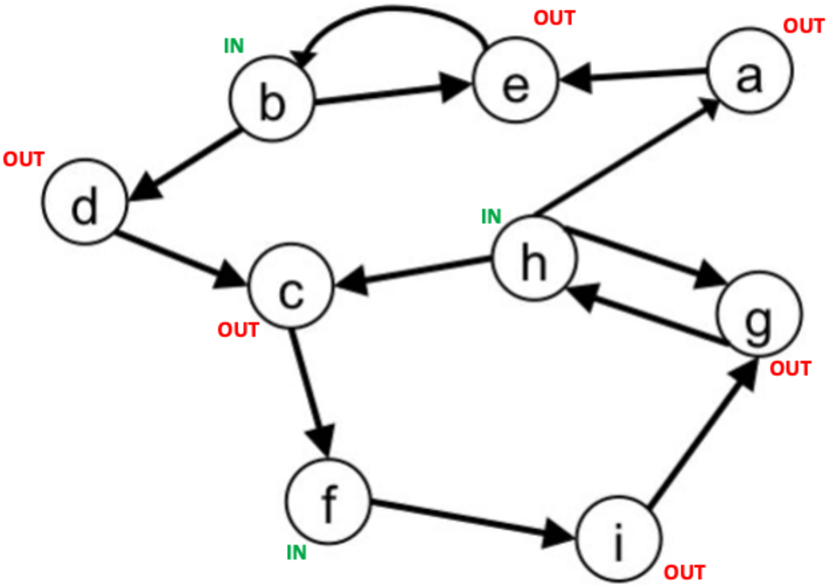
(b) the preferred credulous semantics;

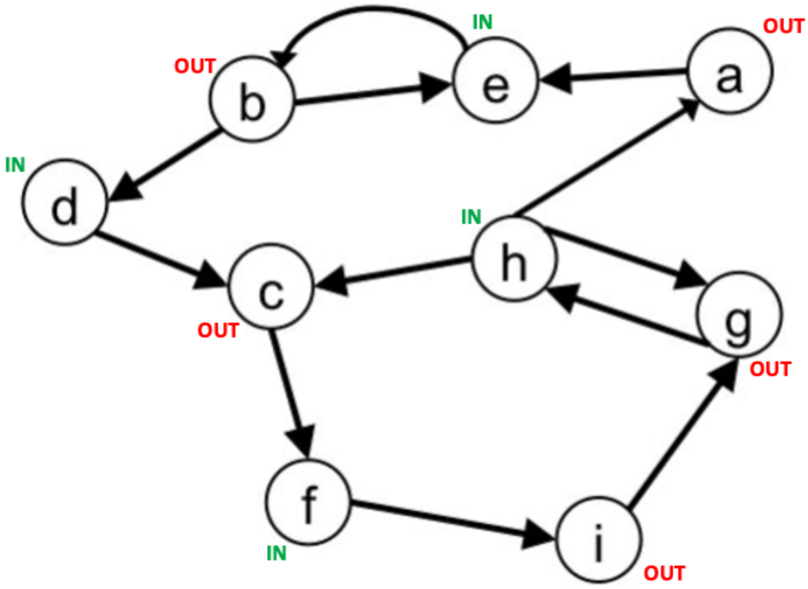
(c) the preferred sceptical semantics.

No arguments are acceptable under the grounded semantics. The valid labelling under the grounded semantics is:



Under the preferred semantics, there are two valid labellings:





The arguments acceptable under the preferred credulous semantics are: b, e, h, f, d.

The arguments acceptable under the preferred sceptical are: h, f.

Tutorial questions, topic 10: argument-based dialogues

1. Consider the following simple dialogue system with two participating agents. Each agent has its own knowledge base (its beliefs), which is a finite set of arguments about beliefs.

   Moves:

   An agent can ASSERT an argument.

   An agent can make a CLOSE move.

   Protocol:

   - The agents take it in turn to make a move.
   - The first agent to move ASSERTs an argument, which causes that argument to become the TOPIC of the dialogue.
   - For all other turns:
     - An agent can ASSERT an argument as long as it not previously been asserted and it attacks some argument that has been previously asserted during the dialogue. An agent can always make a CLOSE move.

   Termination rules:

   A dialogue terminates if each agent makes a CLOSE move in succession (i.e. one agent makes a CLOSE move and then the other agent makes a CLOSE move immediately afterwards).

   Dialogue outcome:

   If the TOPIC is acceptable under the grounded semantics given the abstract argumentation framework that is constructed from all of the arguments that have been asserted during the dialogue, then the outcome of the dialogue is Acceptable(TOPIC).

   If the TOPIC is not acceptable under the grounded semantics given the abstract argumentation framework that is constructed from all of the arguments that have been asserted during the dialogue, then the outcome of the dialogue is NotAcceptable(TOPIC).

   Let's consider a few different strategies.

   Strategy 1:

   If it is permissible to ASSERT an argument from your beliefs, then ASSERT some such an argument;

   otherwise make a CLOSE move.

   Strategy 2: If it is permissible to ASSERT an argument from your beliefs and adding that argument to the abstract argumentation framework that is constructed from all off the arguments that have been asserted so far during the dialogue causes the label (i.e. IN, OUT, or UNDEC) of the TOPIC to change, then ASSERT such an argument; otherwise make a CLOSE move.

Strategy 3:

If it is permissible to ASSERT an argument from your beliefs and adding that argument to the abstract argumentation framework that is constructed from all off the arguments that have been asserted so far during the dialogue causes the label of the TOPIC to change to IN, then ASSERT such an argument;

otherwise make a CLOSE move.

Strategy 4:

If it is permissible to ASSERT an argument from your beliefs and adding that argument to the abstract argumentation framework that is constructed from all off the arguments that have been asserted so far during the dialogue causes the label of the TOPIC to change to OUT, then ASSERT such an argument;

otherwise make a CLOSE move.

(a) Assume both of the agents are using Strategy 1. Can you say what type of dialogue (from Walton and Krabbe's dialogue typology) this would produce? If you assume an omniscient perspective where you can see inside the knowledgebase of each agent, can you say what the outcome of the dialogue would be?

Answer:

If both agents are using strategy 1, we'll end up with each agent asserting all of their arguments that are connected in some way to the topic argument. This leads to an (inefficient) inquiry dialogue, where the outcome of the dialogue is sound and complete in relation to reasoning with the union of the two agents' knowledge bases. The outcome will be Acceptable(TOPIC) if and only if the TOPIC argument is acceptable under the grounded semantics applied to the union of the two agents' knowledge bases. The outcome will be NotAcceptable(TOPIC) if and only if the TOPIC argument is not acceptable under the grounded semantics applied to the union of the two agents' knowledge bases.

(b) Assume both of the agents are using Strategy 2. Can you say what type of dialogue (from Walton and Krabbe's dialogue typology) this would produce? If you assume an omniscient perspective where you can see inside the knowledgebase of each agent, can you say what the outcome of the dialogue would be?

Answer:

If both agents are using strategy 2, the dialogue will continue until neither agent has any arguments that they could add to the argumentation framework constructed from the asserted arguments that would cause the status (IN/OUT/UNDECIDED) of the TOPIC argument to change under the grounded semantics. This leads to a (more efficient) inquiry dialogue, where the outcome of the dialogue is sound and complete in relation to reasoning with the union of the two agents' knowledge bases. The outcome will be Acceptable(TOPIC) if and only if the TOPIC argument is acceptable under the grounded semantics applied to the union of the two agents' knowledge bases. The outcome will be NotAcceptable(TOPIC) if and only if the TOPIC argument is not acceptable under the grounded semantics applied to the union of the two agents' knowledge bases.

(c) Assume the agent who starts the dialogue was using Strategy 3 and the other agent is using Strategy 4. Can you say what type of dialogue (from Walton and Krabbe's dialogue typology) this would produce?

Answer:

The agent using Strategy 3, will only assert arguments that, when added to the argumentation framework that is constructed from the arguments asserted during the dialogue, will cause the status (IN/OUT/UNDECIDED) of the TOPIC argument to change to IN under the grounded semantics. It is therefore trying to persuade the other agent that the TOPIC argument is acceptable.

The agent using Strategy 4, will only assert arguments that, when added to the argumentation framework that is constructed from the arguments asserted during the dialogue, will cause the status (IN/OUT/UNDECIDED) of the TOPIC argument to change to OUT under the grounded semantics. It is therefore trying to persuade the other agent that the TOPIC argument is not acceptable.