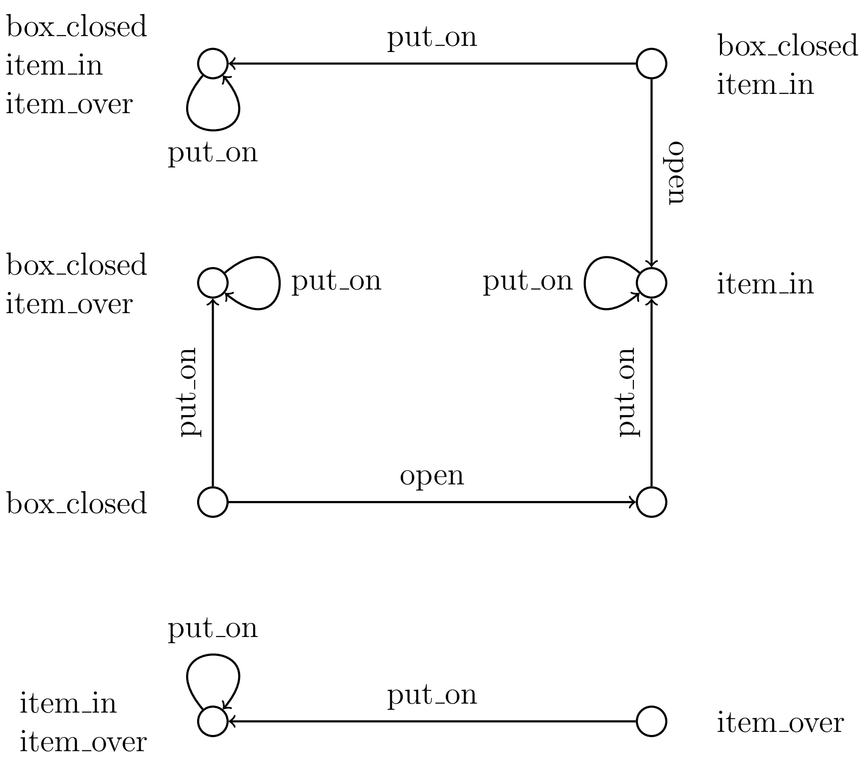
Question 1

Draw the complete search space of the “over” example as a directed graph.

The main characteristic that distinguishes the “over” example from the other ones is the use of conditional effects. In particular, the “put_on” action produces a different effect depending on the variable “box_closed”. However, in this case, it can be managed in a straightforward way. First, we check whether the preconditions are satisfied in a state, then we perform the actual operation computing the positive and the negative effects (this time only if the condition is met in the considered state). This has a direct translation into the search space. Conditional effects are often associated with models of complex situations, for example partial observable (or unobservable) environments and non-deterministic environments, where the most common representations make use of belief states and AND–OR branches.

In the search space, each node is labeled only with the true variables (missing variables are negated in that state). To construct a complete graph, each element (variables and actions) is considered only for its role, ignoring its semantic meaning (for example, an item cannot be over a box if the box is open). This problem must be addressed by designing in the right way both the domain and the problem files.



Question 2

Write a STRIPS problem with two variables and two actions; calculate the h^{max} heuristics on two states.

Elon is a poor guy, and he wants to find a plan to get rich. He comes out with the following STRIPS configuration.

- $P = \{\text{company}, \text{money}\}$
- $O = \{\langle\emptyset, \{\text{company}\}, \emptyset\rangle, \langle\{\text{company}\}, \{\text{money}\}, \{\text{company}\}\rangle\}$
- $I = \emptyset$
- $G = \{\text{company}, \text{money}\}$

We have two fluents $\{\text{company}, \text{money}\}$ and two operators.
The first one is $\text{start\_business} = \langle\emptyset, \{\text{company}\}, \emptyset\rangle$, while the second is $\text{sell} = \langle\{\text{company}\}, \{\text{money}\}, \{\text{company}\}\rangle$. Let's assume that the first action has cost 5 and the second one has cost 4. The optimal plan to reach the goal state from the initial one is $[\text{start\_business}, \text{sell}, \text{start\_business}]$, and the total cost is 14.

Now we can calculate the h^{max} heuristics. First of all, we have to remove the negative effects (delete relaxation). In our case, this means changing the second action, which becomes $\text{sell} = \langle\{\text{company}\}, \{\text{money}\}, \emptyset\rangle$. We first compute the heuristics on the state where both variables are negated, and we consider as goal state the goal of the previous STRIPS problem.

$$
\begin{tikzcd}
\begin{array}{c}\emptyset\\0\end{array} \arrow[r] & \begin{array}{c}\text{start\_business}\\0+5\end{array} \arrow[r] & \begin{array}{c}\text{company}\\5\end{array} \arrow[r] \arrow[d] & \begin{array}{c}\text{dummy\_action}\\max\{5,9\}+0\end{array} \arrow[r] & \begin{array}{c}\text{goal}\\9\end{array} \\
 & & \begin{array}{c}\text{sell}\\5+4\end{array} \arrow[r] & \begin{array}{c}\text{money}\\9\end{array} \arrow[u] &
\end{tikzcd}
$$

The goal state contains two variables and this is why we must introduce a dummy action with zero cost. As any other action with at least two preconditions, the total cost of executing the dummy action is its cost plus the maximum of the preconditions. In fact, both preconditions must be obtained and this cannot be easier than obtaining the "worst" of the two. This is the idea behind the h^{max} heuristics. In general, the algorithm does not know if the preconditions are generated by different paths or the same one. The application of the max is only used in that dummy action because this toy example has too few variables and does not help us to appreciate the proceeding of the method. We can still see all the fundamental steps, the use of the heuristics and an example of the cost underestimation ($9 \leq 14$).

If we compute the h^{max} on the state where only the variable *company* is positive, we obtain

$$\begin{tikzcd} \begin{matrix}\text{company}\\ 0\end{matrix} \arrow[r] \arrow[rrr, bend right] & \begin{matrix}\text{sell}\\ 0+4\end{matrix} \arrow[r] & \begin{matrix}\text{money}\\ 4\end{matrix} \arrow[r] & \begin{matrix}\text{dummy\_action}\\ max\{0,4\}+0\end{matrix} \arrow[r] & \begin{matrix}\text{goal}\\ 4\end{matrix} \end{tikzcd}$$

Question 3

Translate the problem of the previous point into SAT

Preconditions

$\text{sell}_0 \rightarrow \text{company}_0$
$\text{sell}_1 \rightarrow \text{company}_1$

Effects

$\text{company}_1 \equiv \text{start\_business}_0 \vee \text{company}_0 \wedge \neg\text{sell}_0$
$\text{money}_1 \equiv \text{sell}_0 \vee \text{money}_0$
$\text{company}_2 \equiv \text{start\_business}_1 \vee \text{company}_1 \wedge \neg\text{sell}_1$
$\text{money}_2 \equiv \text{sell}_1 \vee \text{money}_1$

Initial State

$\neg\text{company}_0 \wedge \neg\text{money}_0$

Goal State

$(\text{company}_0 \wedge \text{money}_0) \vee (\text{company}_1 \wedge \text{money}_1) \vee (\text{company}_2 \wedge \text{money}_2)$

Actions

$start\_business_0 \lor sell_0$
$\neg start\_business_0 \lor \neg sell_0$
$start\_business_1 \lor sell_1$
$\neg start\_business_1 \lor \neg sell_1$

Actually, we can avoid the single-action constraints. This is because, in this case, we can allow the execution of parallel action through linearization since no action falsifies the precondition of another: start_business does not have negative effects; sell falsifies company but the latter is not the precondition of start_business.

Finally we obtain the propositional formula taking the conjunction of all previous constraints. Finding an interpretation that satisfies that formula means finding a plan for the initial problem. In particular, a solution restricted to only the actions and obtained with the linearization is $[start\_business_0, sell_1, start\_business_1]$. If we don't use linearization we have to add one "epoch", obtaining $[start\_business_0, sell_1, start\_business_2]$. Using mutex, we have to introduce again the constraint $\neg start\_business_1 \lor \neg sell_1$, because the two actions have opposite effects (mutex between effects). This precludes us from the first solution, not from the second.