

7 Disjunctions again

After going through the conjunctions based approach, we have decided to go back to an enumeration of disjunctions. Firstly, we enumerate all possible disjunctions and take the move vectors based on this enumeration. Let k be the number of agents. Let $number(D)$ be the ordering of a disjunction D . If

$$P \Rightarrow \langle\langle \mathcal{A} \rangle\rangle \bigcirc D$$

is in the set of clauses, we keep the edges labelled by $\langle a_1, \dots, a_k \rangle$, where $a_i = number(D)$, for all $i \in \mathcal{A}$, from s to s' such that

$$s \models P \text{ and } s' \models D$$

and for all $j \notin \mathcal{A}$

$$s' \models \bigwedge_{j \notin \mathcal{A}} number^{-1}(a_j).$$

That is, if a state s satisfies the left-hand side of a step clause and all the agents in the coalition are voting to satisfy the disjunction on the right-hand side, then the successor should satisfy this disjunction. Moreover, the other agents should not vote for a disjunction which is not satisfied at the successor.

If we start with a full graph, we remove all edges from s to s' labelled by $\langle a_1, \dots, a_k \rangle$, where $a_i = number(D)$, for all $i \in \mathcal{A}$, and where a_j is arbitrary, for all $j \notin \mathcal{A}$, such that

$$s \models P \text{ and } s' \not\models D$$

and, for all $j \notin \mathcal{A}$, we also remove all edges from s to s' such that

$$s' \not\models \bigwedge_{j \notin \mathcal{A}} number^{-1}(a_j).$$

7.1 Examples

Suppose we are given the following set of clauses:

1. **true** $\Rightarrow \langle\langle 1 \rangle\rangle \bigcirc p$
2. **true** $\Rightarrow \langle\langle 2 \rangle\rangle \bigcirc (\neg p \vee q)$

We start by enumerating all possible disjunctions:

| disjunction | ordering |
|----------------------|----------|
| p | 0 |
| q | 1 |
| $\neg p$ | 2 |
| $\neg q$ | 3 |
| $p \vee q$ | 4 |
| $p \vee \neg q$ | 5 |
| $\neg p \vee q$ | 6 |
| $\neg p \vee \neg q$ | 7 |

The choices for each agent are the same in every state (which later will give rise to $d(s, a)$) is, then, $\mathcal{L} = \{0, 1, 2, 3, 4, 5, 6, 7\}$. The move vectors are taken from \mathcal{L}^2 , as we have two agents. In this particular example, we

have 64 different move vectors. As we start with the full graph, each edge goes to every state. Table 2 shows all the edges in the graph.

Now, suppose we add the first clause, that is,

$$1. \text{ true} \Rightarrow \langle\langle 1 \rangle\rangle \bigcirc p$$

All states satisfy the left-hand side of this clause. The ordering of the disjunction on the right-hand side is 0. Therefore, we have to look at the edges that are of the form $(0, *)$, where $*$ is one of the possible moves for agent 2. The first condition says that there is an edge of the form $(0, *)$ if, and only if, the successor state satisfies p . Therefore, we remove the edges that violate this condition. The result is shown in Table 3, where the removed edges are shown in red. Furthermore, if agent 1 is voting to satisfy 0, agent 2 should not vote for a clause which is not satisfied in the successor. The removed edges are shown in yellow in Table 3.

Now, suppose we add the second clause, that is,

$$2. \text{ true} \Rightarrow \langle\langle 2 \rangle\rangle \bigcirc (\neg p \vee q)$$

All states satisfy the left-hand side of this clause. The ordering of the disjunction on the right-hand side is 6. Therefore, we have to look at the edges that are of the form $(*, 6)$, where $*$ is one of the possible moves for agent 1. The first condition says that there is an edge of the form $(*, 6)$ if, and only if, the successor state satisfies $(\neg p \vee q)$. Therefore, we remove the edges that violate this condition. The result is shown in Table 4, where the removed edges are shown in blue. Furthermore, if agent 2 is voting 6 to satisfy $(\neg p \vee q)$, agent 2 should not vote for a clause which is not satisfied in the successor. The removed edges are shown in green in Table 4.

Now, note that the resolvent from 1 and 2 is satisfied by the move vector $(0,6)$. Note as well that we can build a model for this set of clauses, as there are edges labelled by all move vectors in \mathcal{L}^2 leaving from each node.

Suppose, however, that we want to add the following clause:

$$3. \text{ true} \Rightarrow \langle\langle 1, 2 \rangle\rangle \bigcirc q \quad [1, 2, \text{SRES}]$$

This would remove the edges labelled by $(1, 1)$ from each state to a state that does not satisfy q . The result is shown in Table 5, where the removed edges are shown in pink. No other restriction applies, as the other agents are not voting, that is, the empty conjunction is satisfied by all states.

Finally, suppose we want to add the following clause

$$4. \text{ true} \Rightarrow \langle\langle \emptyset \rangle\rangle \bigcirc \neg q$$

then, we observe that the first condition says that will keep the edges where all agents in the coalition vote for the disjunction on the right-hand side of this clause. As the coalition is empty, the condition applies to all edges. Therefore, any edges leading to states that do not satisfy $\neg q$ are removed. This is shown in Table 6, where the removed edges are shown in purple.

Note that there is no edge labelled by $(0, 6)$ leaving any state. All states are deleted. The graph is empty.

[illegible]Table 2: Vector moves for $\mathcal{P} = \{p, q\}$ and $\Sigma = \{1, 2\}$

[illegible]

Table 3: Vector moves for $\mathcal{P} = \{p, q\}$ and $\Sigma = \{1, 2\}$ after adding the first clause

[illegible]

Table 4: Vector moves for $\mathcal{P} = \{p, q\}$ and $\Sigma = \{1, 2\}$ after adding the first and the second clauses

[illegible]

Table 5: Vector moves for $\mathcal{P} = \{p, q\}$ and $\Sigma = \{1, 2\}$ after adding the first and the second clauses, as well as their resolvent

[illegible]

Table 6: Vector moves for $\mathcal{P} = \{p, q\}$ and $\Sigma = \{1, 2\}$ after adding clauses 1, 2, 3 and 4