Solutions to TD10

1 Reachability in Petri Nets

1.1 No transition can be fired twice

Suppose on the contrary that some transition t can be fired twice in some run

$$m_1 \xrightarrow{t} m_2 \xrightarrow{\sigma}^* m'_1 \xrightarrow{t} m'_2.$$

Let Q be the set of places and transitions from which t is reachable. Note that $t^\bullet \cap Q = \emptyset$ else there exists a cycle.

We modify $m_2 \xrightarrow{\sigma}^* m'_1$ by taking out the transitions not in Q to get a sequence $m_2 \xrightarrow{\sigma'}^* m''_1$. Note that t can still be fired from m''_1. Consider now the run

$$m_1 \xrightarrow{t} m_2 \xrightarrow{\sigma'}^* m''_1 \xrightarrow{t} m''_2.$$

As $t^\bullet \cap Q = \emptyset$ and σ' consists only of transitions in Q, we must have $m''_1(p) = m_2(p) = 1$ for $p \in t^\bullet$. After firing t from m''_1, we have $m''_2(p) = 2$ for $p \in t^\bullet$, contradicting 1-safeness of N.

1.2 Reachability for $\mathcal{A}$ is in NP

Given a firing sequence of polynomial length, at each step, one only needs to compute the new marking from the current marking which can be done in linear time. Thus it is possible to decide in polynomial time whether the firing sequence satisfies the given instance of the reachability problem.

1.3 Reachability for $\mathcal{A}$ is NP-hard

The gadgets used for building the 1-safe Petri net from a given SAT formula $\varphi = C_1 \wedge \cdots \wedge C_k$ where $C_i = y_{i1} \vee \cdots \vee y_{in_i}$ are shown in Figure 1. In order to build the Petri net, we join the transition $t_{y_{ij}}$ to the place y_{ij} for all $i \leq k$ and $j \leq n_i$. Note that every place y_{ij} has only one incoming transition $t_{y_{ij}}$ and every transition t_{x_i} may be fired at most once, so y_{ij} can only ever contain at most one token. It is easy to see that the other places are also 1-safe.

One can show that φ is solvable if and only if the marking where every C_i has a token is reachable.

2 Traffic Lights

2.1 Part (a)

See Figure 2.

2.2 Part (b): Street Intersection

We will assume that our traffic lights are manufactured in France (where they follow the rules $\texttt{R} \to \texttt{G}, \texttt{G} \to \texttt{Y}$ and $\texttt{Y} \to \texttt{R}$.). See Figure 3.

The two black places on the bottom ensure that once a traffic light goes from yellow to red, it cannot be greedy and must allow the other traffic light to go from red to green (instead of going from red to green itself).

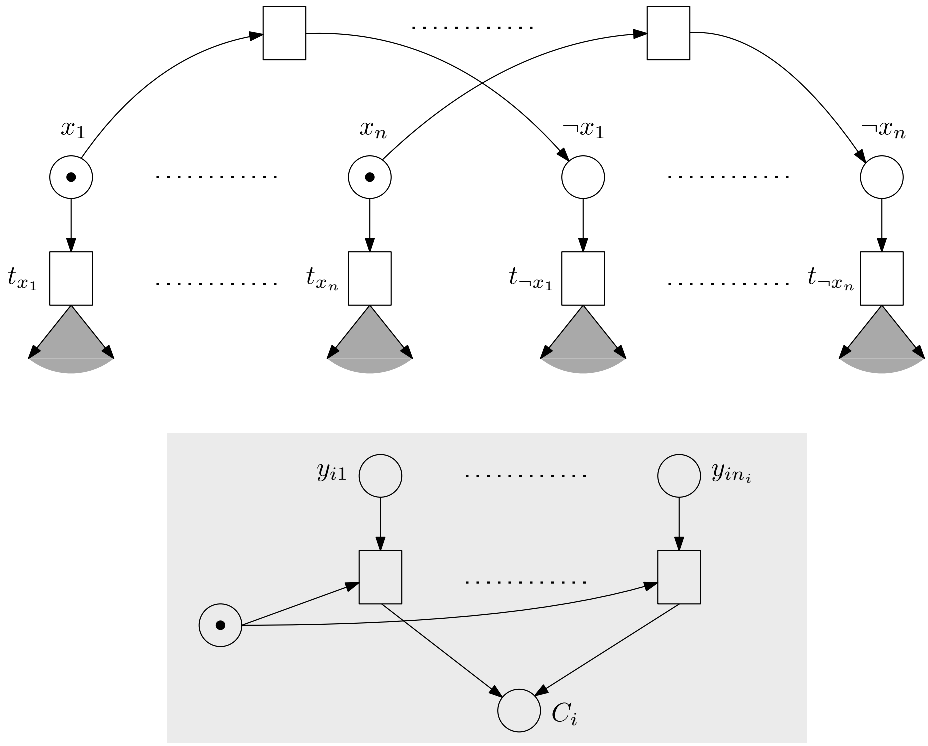

Figure 1: Gadgets for the literals and clauses

3 Producer/Consumer

3.1 Part (a)

See Figure 4.

3.2 Part (b)

Figure 5 shows a Petri net where the first producer to deliver to the channel gets priority by blocking the other producer from using its inhibitor arc. Once it is done, it empties the tokens from the newly added place, allowing the inhibitor arc to be fired by the other producer.

4 Model Checking Petri Nets

4.1 1-safe Petri Nets

Let M denote the set of possible markings of the 1-safe Petri net $\mathcal{N}$. We can construct the Büchi automaton $\mathcal{B} = (Q, \Sigma, \Delta, I, F)$ with state space $Q = \{\iota\} \cup M$, alphabet $\Sigma = M$, initial state space $I = \{\iota\}$ and final state space $F = Q$ where $\iota \notin M$ is a special state. We define the transition set Δ as follows:

$$\iota \xrightarrow{m_0}_\Delta m_0,$$
$$m \xrightarrow{m'}_\Delta m' \quad \text{if } m \xrightarrow{t}_\mathcal{N} m'.$$

Remark: The main point is to have the transitions in $\mathcal{B}$ labeled by markings in $\mathcal{N}$.

To check reachability, one can construct the cross-product of $\mathcal{B}$ and $\mathcal{B}_\varphi$ (the Büchi automaton constructed from the given LTL formula φ) on-the-fly while doing the following. One guesses the path leading to a lasso, and once the lasso is reached, one remembers the starting point and

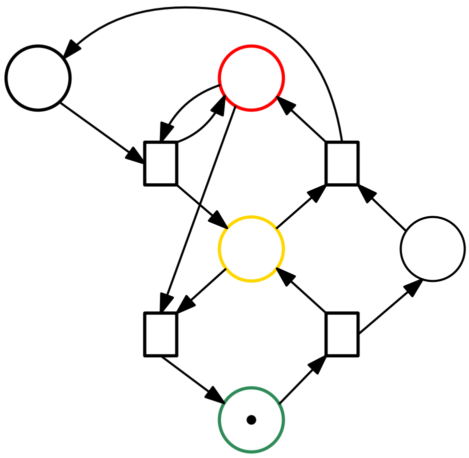

Figure 2: Better traffic lights

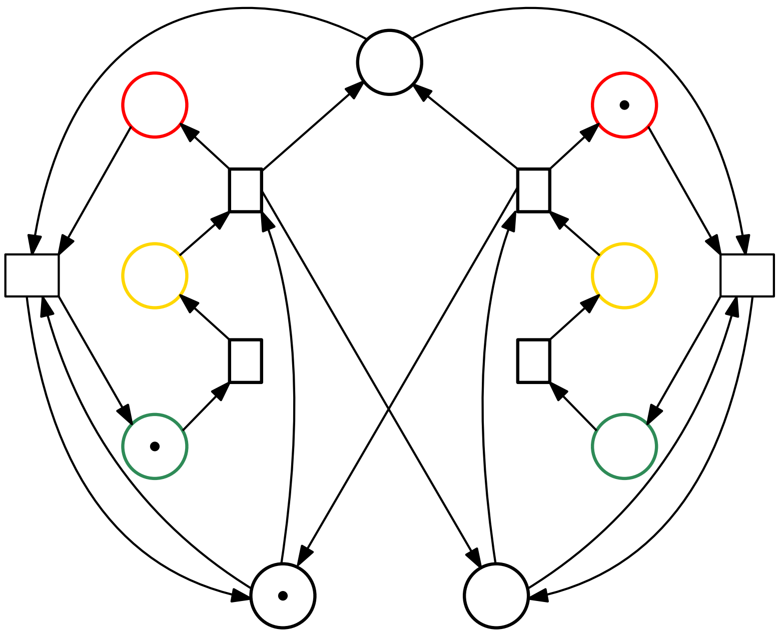

Figure 3: Street intersection

then guesses a loop back to the starting point. This can be done in polynomial space, showing that reachability is in NPSPACE and thus in PSPACE by Savitch's theorem.

4.2 Petri Nets with at least two unbounded places

The 2-counter Minsky machine that we consider has three operations: increment, decrement and zero test. Increment is executable unconditionally, decrement is executable only if the counter is not 0 and zero test is executable only when the counter is zero. If Q is the set of states of the machine and c_1, c_2 are the counters, the places of the Petri net $\mathcal{N}$ to which we want to reduce the counter machine are given by $Q \cup \{c_1, c_2\}$. We can construct $\mathcal{N}$ by using the gadgets in Figure 6 and placing a token at the initial state of the machine.

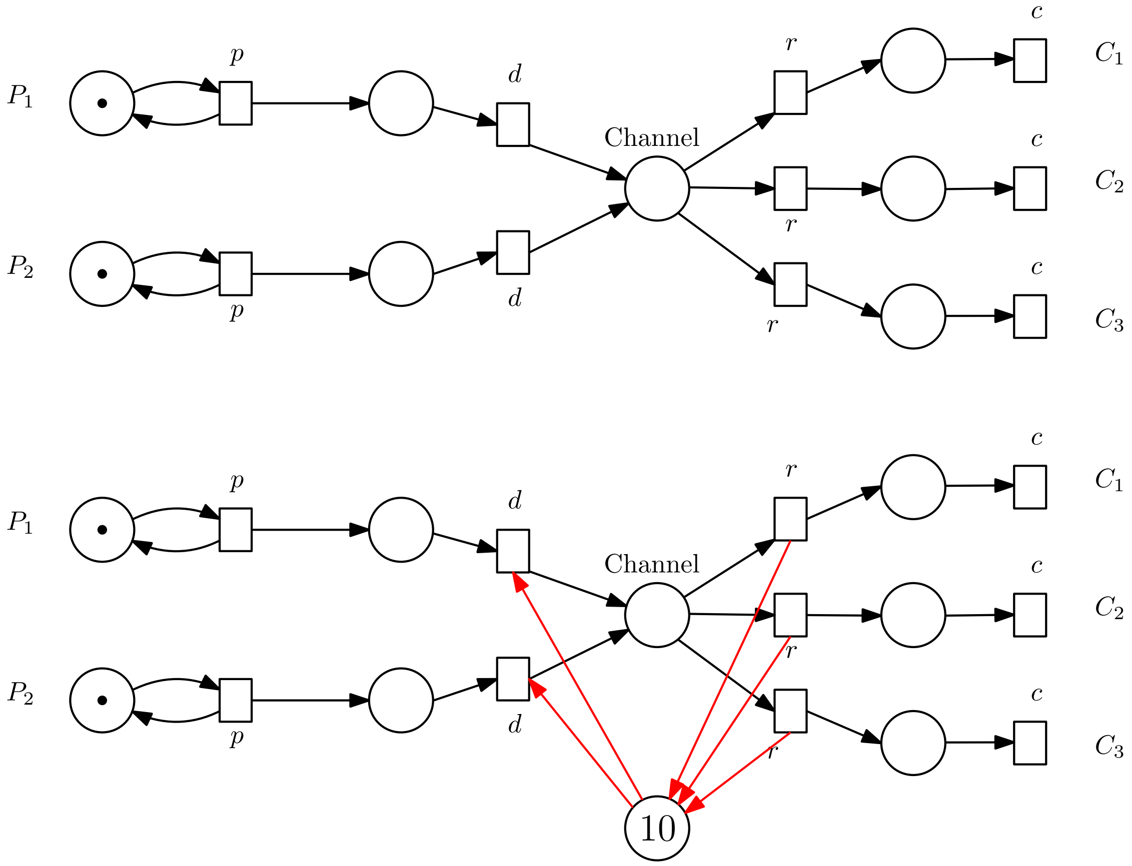

Figure 4: Unbounded channel and channel with limit of 10 tokens

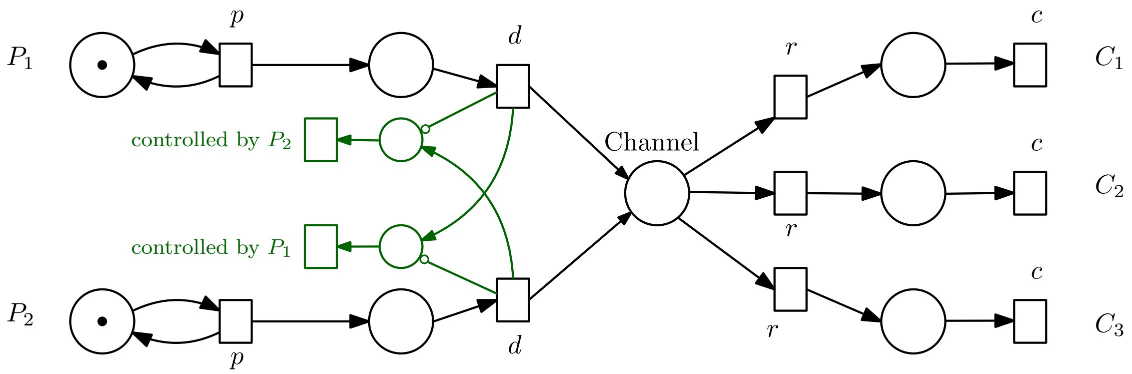

Figure 5: With priority for first producer

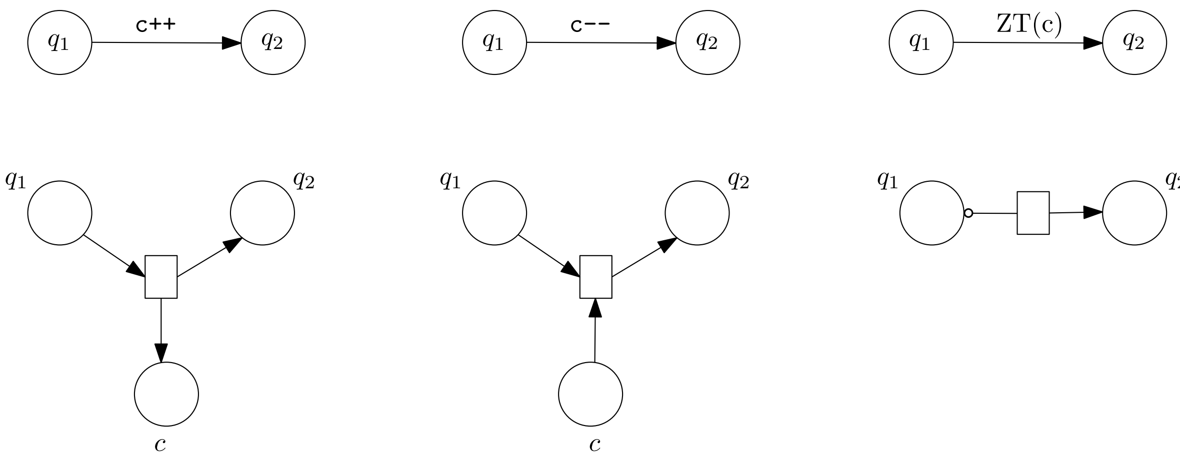

Figure 6: Gadgets used to reduce given 2-counter machine to $\mathcal{N}$