We need to force that \( t_r \) is used infinitely often in any infinite run \( R \) (it was automatic only for unbounded runs); we thus employ a further idea:

(TO BE MADE NICER:) We add places \( S, S' \); every transition takes from \( S \) and puts to \( S' \). And \( t_r \) does not give in \( s_0 \) but into an intermediate \( p_b \) which enables to move tokens from \( K' \) to \( K \) (also resetting everything else) and then move to \( s_0 \). \( \square \)

Now we show that BP is undecidable for Reset-PNs.

Remark. (MOVE ELSEWHERE ??) The undecidability of boundedness for Reset PNs might be a bit surprising at first sight, especially when coverability (CP) is decidable. Coverability and boundedness at PNs are naturally associated with coverability trees, and have the same complexity in many instances. We have already discussed (reference !!!) that there was a natural conjecture that the boundedness problem for Reset nets is decidable.

Interesting is also the fact that we need three reset arcs for demonstrating the undecidability. To show that this is substantial, we add Section ?? which proves that the boundedness problem for Reset PNs with (at most) two reset arcs is decidable. Thus the decidability boundary is here between 2 and 3, and not between 1 and 2 as is ‘more usual’.

To show the undecidability, we again use 2-counter Minsky machines but now the second problem in Fact 3.2.

**Lemma 7.6** Given any 2-counter machine \( M \), we can effectively (in polynomial time) construct a Reset net \( (N, m_0) \) with 3 reset arcs so that \( M \) is bounded for inputs \( (0,0) \) iff \( (N, m_0) \) is bounded.

**PROOF.** We again start with the (basic) Reset net \( N_M \). Instead of generating the capacity in advance (which would automatically result in an unbounded net), we implement an idea of allowing capacity increasing ‘on demand’, which will result in real increasing only when no cheating has occurred, i.e., when no token from the sum of \( c_1, c_2, K \) has been lost so far.

The implementation is depicted on Fig. ??.

Though there are more than three reset arcs, the only resettable places are \( c_1, c_2, K \); \( N_M \) can thus serve for our aim due to Lemma 7.2.

(CORRECT \( e_i \) ... to \( s_i \) in the figure)

Let us now look at Figure 8, assuming that the sum of tokens in \( c_1, c_2, K \) is \( \ell \), there is a token in \( s_i \) and 0 elsewhere. If now \( t_{s_i} \) is performed, emptying \( K \) (if it was nonempty), then transitions \( r_{c_1} \) and \( r_{c_2} \) successively transfer (some)
tokens from $c_1$ and $c_2$ to $K$, and finally $t_{rs}$ empties $c_1$, $c_2$, adds a (new) token into $K$ and starts a new simulation: there is 1 token in $s_0$, $\ell'$ tokens in $K$ and 0 elsewhere.

We can easily verify that $\ell' \leq \ell + 1$, and we can get $\ell' = \ell + 1$ if and only if $K$ was empty before performing $t_{si}$.

It remains to verify that $M$ is bounded iff $(N, m_0)$ is bounded, where $m_0$ puts 1 token in $s_0$ and 0 tokens elsewhere. We first note that the set \{s_0, s_1, \ldots, s_n, pc\} is a place-invariant, i.e. the sum of tokens in these places is constant – it is 1 in our case. Any reachable marking can thus be naturally presented as $(s, x_1, x_2, x_3)$ where $s \in \{s_0, s_1, \ldots, s_n, pc\}$, and $x_1, x_2, x_3$ are the values of $c_1, c_2, K$ respectively; hence $m_0 = (s_0, 0, 0, 0)$.

- Suppose $M$ is unbounded for 0,0. Then $(N, m_0)$ can simulate the computation of $M$, and when an increment instruction is not performable because $K$ is empty then the ‘restarting segment’ beginning with the relevant $t_{si}$ is performed. The simulation of the computation of $M$ (on 0,0) can then be performed again, now from $(s_0, 0, 0, 1)$ (i.e., with a bigger capacity); etc. In this way we get an unbounded run in $(N, m_0)$ (going through $(s_0, 0, 0, 0), (s_0, 0, 0, 1), (s_0, 0, 0, 2), \ldots$); hence $(N, m_0)$ is unbounded.

- Suppose $(N_M, m_0)$ is unbounded, which means that there is an unbounded run $R$ from $m_0$. Since the only transition which can increase the number of tokens in $N_M$ is $t_{rs}$ – increasing by 1, $t_{rs}$ occurs infinitely often in $R$. This means that $R$ necessarily visits markings $(s_0, 0, 0, 0), (s_0, 0, 0, 1), \ldots, (s_0, 0, 0, \ell), (s_0, 0, 0, \ell + 1), \ldots$. Recalling our previous observations, it is obvious that $(s_0, 0, 0, \ell + 1)$ is reachable from $(s_0, 0, 0, \ell)$ only if the sum of $c_1$ and $c_2$ reaches $\ell$ in the computation of $M$ (on 0,0). This implies that the computation of $M$ must be unbounded.

\[ \square \]