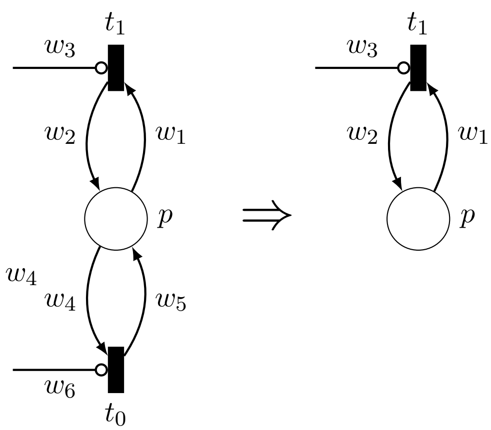
Rule L: Dominated Transition

Rule L removes transitions that have the same effect as another transition, but with more preconditions. Since both transitions lead to the same state, we can therefore remove the one with the higher preconditions and use the other instead. See the formal description in Figure 1.

where
$w_3 \geq w_6$
$w_1 \leq w_4$
$w_2 - w_1 = w_5 - w_4$



Precondition	Update
Fix transition t_1 and t_0 s.t.: L1) $I(t_1) \geq I(t_0)$ L2) $\boxminus(t_1) \leq \boxminus(t_0)$ L3) $E(t_1) = E(t_0)$	UL1) Remove t_0

Figure 1: Rule L: Dominated Transition

Theorem 1 *Rule L in Figure 1 is correct for CTL*.*

Rule M: Effectively dead places and transitions

The Rule M finds and removes effectively dead places and transitions. We define an effectively dead place to be a place that will never gain nor lose tokens. Effectively dead transitions are transitions that are initially disabled (and/or inhibited) by a place that cannot gain (and/or lose) tokens. These places and transitions are found using fixed-point iteration as defined in Algorithm 1.

Algorithm 1: Rule M: Effectively dead places and transitions

Input: A net $N = \langle P, T, \boxminus, \boxplus, I \rangle$, initial marking M_0 and CTL* formula φ

Output: A reduced net N' and its initial marking M'_0

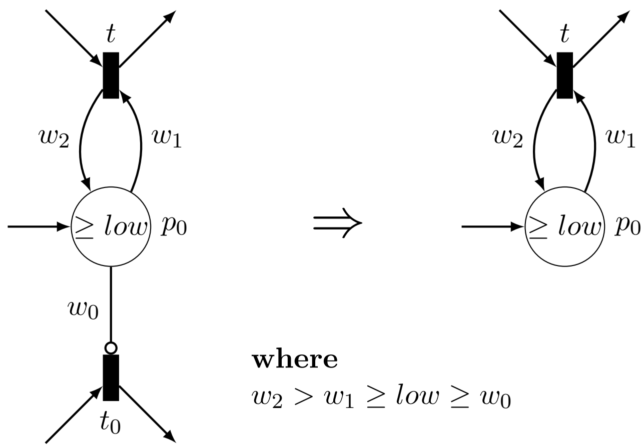
$S_\leq := P$ `/* Places that cannot gain tokens */`
$S_\geq := P$ `/* Places that cannot lose tokens */`
$F := T$ `/* Transitions that cannot fire */`
do
  `/* Find transitions that may fire and update sets accordingly */`
  foreach $t \in F$ *where* $\forall p \in P.(\boxminus(p,t) \leq M_0(p) \vee p \notin S_\leq) \wedge (I(p,t) > M_0(p) \vee p \notin S_\geq)$ **do**
    $F := F \setminus \{t\}$
    $S_\leq := S_\leq \setminus t^{\boxplus}$
    $S_\geq := S_\geq \setminus {}^{\boxminus}t$
until $S_\leq$, $S_\geq$, *and* F *do not change*
$P' := P \setminus (S_\leq \cap S_\geq \setminus places(\varphi))$
$T' := T \setminus F$
return $N' = \langle P', T', \boxminus, \boxplus, I \rangle$ *and* M_0

Theorem 2 *Rule M in Algorithm 1 is correct for CTL*.*

Theorem 3 *Rule M supercedes Rule E.*

Rule O: Inhibited transition

We can find the lower bound of tokens at a place p_0. Any inhibitor arc from p_0 with a weight smaller than the lower bound always inhibits the given transition, which means that the transition can be removed. See Figure 2 for a formal description of Rule O.



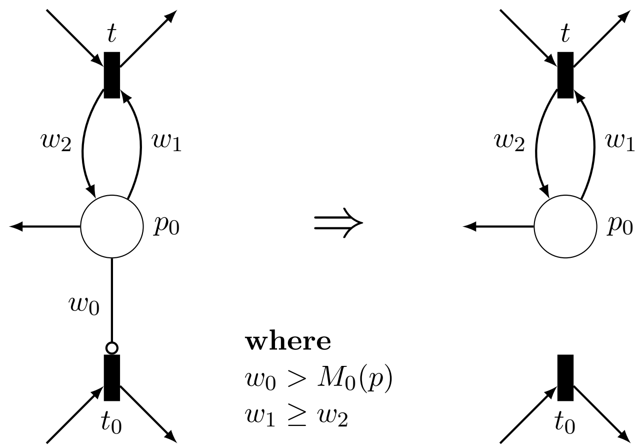
Precondition	Update
Fix place p_0 and transition t_0 s.t.: O1) $t_0 \in p_0^{\circ}$ O2) $I(p_0, t_0) \leq low$ where $low = \min\{M_0(p_0)\} \cup \{\boxplus(p_0, t) \mid t \in p_0^{\boxminus}\}$	UO1) Remove t_0.

Figure 2: Rule O: Inhibited transition

Theorem 4 *Rule O in Figure 2 is correct for CTL**

Rule P: Redundant inhibitor arc

Sometimes we can find an upper bound on the number of tokens at a place p_0. This upper bound is given by the initial marking if all transitions have a non-positive effect on p_0. Any inhibitor arc from p_0 with a weight higher than the upper bound of p_0 therefore never inhibits, which means the inhibitor arc can be removed. See Figure 3 for a formal description of Rule P.



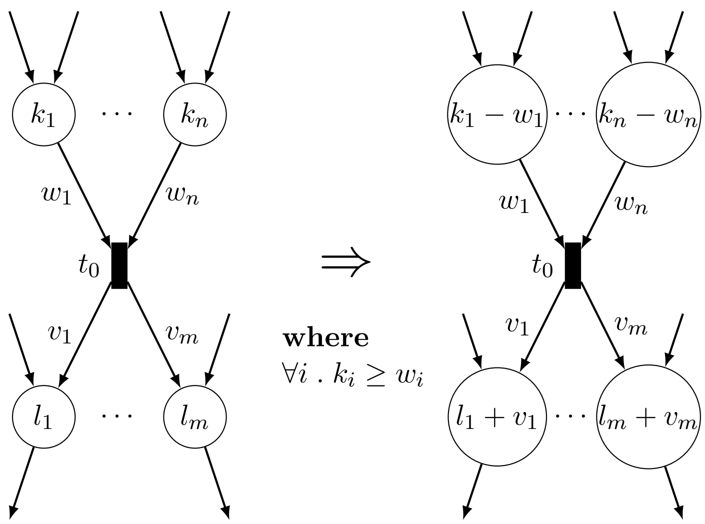
Precondition	Update
Fix place p_0 and transition t_0 s.t.: P1) $t_0 \in p_0^\circ$ P2) $I(p_0, t_0) > M_0(p_0)$ P3) $^{\boxplus}p_0 = \emptyset$	UP1) $I(p_0, t_0) = \infty$.

Figure 3: Rule P: Redundant inhibitor arc

Theorem 5 *Rule P in Figure 3 is correct for CTL*.*

Rule Q: Preemptive transition firing

Rule Q evaluates transitions that are initially enabled and are the only consumer of all places in its pre set. The formal description of Rule Q can be found in Figure 4. Remark that Rule Q can potentially put tokens into places which will prevent other reductions. Furthermore, it can be applied infinitely if $\boxminus(t_0) \leq \boxplus(t_0)$, or if the Petri net contains a loop.



Precondition	Update
Fix transition t_0 s.t.: Q1) $({}^\bullet t)^\bullet = \{t_0\}$ Q2) $\boxminus(t_0) \leq M_0 < I(t_0)$ Q3) $({}^\bullet t_0 \cup t_0^\bullet) \cap places(\varphi) = \emptyset$ Q4) $({}^\bullet t_0)^\circ = (t_0^\bullet)^\circ = \emptyset$	UQ1) $M_0 := M_0 + E(t_0)$.

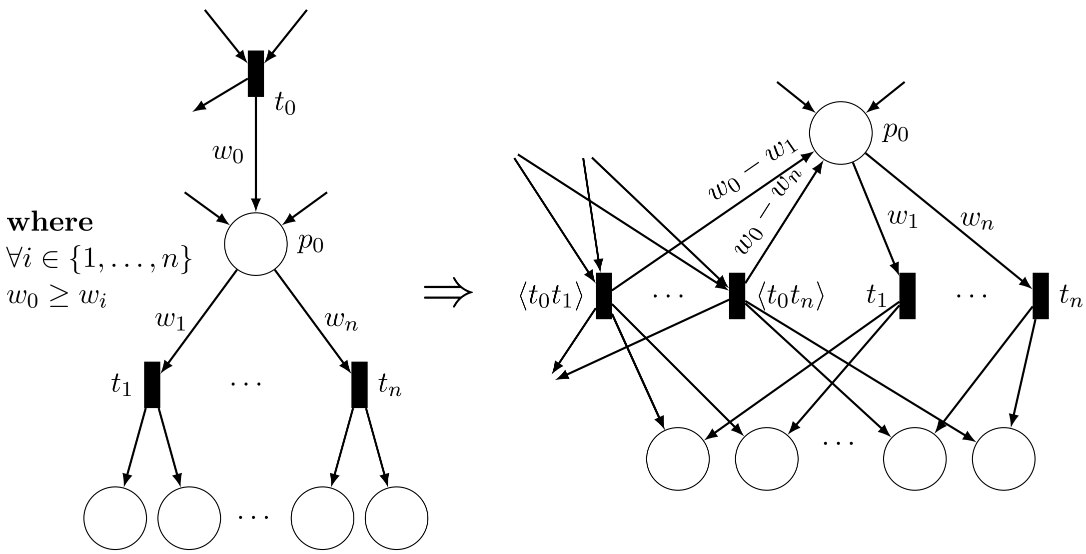
Figure 4: Rule Q: Preemptive transition firing

Theorem 6 *Rule Q in Figure 4 is correct for CTL\X.*

Rule R: Atomic post-agglomerable producer

Rule R is similar to a post agglomeration rule and a formal description of Rule R is in Figure 5. In Rule R we look for a place p_0 with a producer t_0 such that t_0 can always be followed by a firing of any consumer of p_0 without inhibiting other transitions or affecting places in $places(\varphi)$. The producer t_0 is then replaced with new transitions, one for each consumer, and these new transitions combine the effect of firing t_0 and the given consumer. Similarly to an agglomeration rule, Rule R removes interleavings despite potentially increasing the size of the Petri net. However, Rule R is more general, since it only operates on one producer at a time and leaves p_0 untouched, allowing tokens in p_0 in the initial marking, which a post agglomeration does not. Additionally, Rule R does not require the weights of the arcs to and from the agglomerated place to be equal, making R usable in many cases.

Theorem 7 *Rule R in Figure 5 is correct for LTL\X.*



Precondition	Update
Fix place p_0 and transition t_0 s.t.: R1) $t_0 \in {}^\bullet p_0 \wedge p_0^\bullet \neq \emptyset$ R2) ${}^\bullet p_0 \cap p_0^\bullet = \emptyset$ R3) $p_0^\circ = {}^\circ(p_0^\bullet) = ((p_0^\bullet)^\bullet)^\circ = \emptyset$ R4) $(\{p_0\} \cup (p_0^\bullet)^\bullet) \cap places(\varphi) = \emptyset$ R5) ${}^\bullet(p_0^\bullet) = \{p_0\}$ R6) $\boxplus(t_0, p_0) \geq \boxminus(p_0, t)$ for all $t \in p_0^\bullet$	UR1) For each transition $t \in p_0^\bullet$ create a transition $\langle t_0 t \rangle$ with the following arcs: $\boxminus(\langle t_0 t \rangle) = \boxminus(t_0)$ $\boxplus(\langle t_0 t \rangle) = \boxplus(t_0) + \boxplus(t) - \boxminus(t)$ $I(\langle t_0 t \rangle) = I(t_0)$ UR2) Remove t_0

Figure 5: Rule R: Atomic post-agglomerable producer