# *On the generalization of learned constraints for ASP solving in temporal domains*

*Supplementary Material*

JAVIER ROMERO

TORSTEN SCHAUB

KLAUS STRAUCH

*University of Potsdam, Germany*

*submitted [n/a]; revised [n/a]; accepted [n/a]*

## 1 Additional rules of the Blocksworld Example

The following lines specify the action `unstack(X,Y)`, where '`B`' is a shorthand for the body '`block(X), block(Y)`':

```
        action(unstack(X,Y)) :- B.
pre(unstack(X,Y), handempty) :- B.
pre(unstack(X,Y),  clear(X)) :- B.
pre(unstack(X,Y),    on(X,Y)) :- B.
add(unstack(X,Y),  clear(Y)) :- B.
add(unstack(X,Y),holding(X)) :- B.
del(unstack(X,Y),    on(X,Y)) :- B.
del(unstack(X,Y), handempty) :- B.
del(unstack(X,Y),  clear(X)) :- B.
```

The following lines specify the actions `pick_up(X)` and `out_down(X)`, where '`B`' is a shorthand for the body '`block(X)`':

```
        action(pick_up(X))  :- B.
pre(pick_up(X), handempty)  :- B.
pre(pick_up(X),  clear(X))  :- B.
pre(pick_up(X),ontable(X))  :- B.
add(pick_up(X),holding(X))  :- B.
del(pick_up(X), handempty)  :- B.
del(pick_up(X),  clear(X))  :- B.
del(pick_up(X),ontable(X))  :- B.

        action(put_down(X)) :- B.
pre(put_down(X),holding(X)) :- B.
add(put_down(X),  clear(X)) :- B.
add(put_down(X), handempty) :- B.
add(put_down(X),ontable(X)) :- B.
del(put_down(X),holding(X)) :- B.
```

## 2 Program translations from our conference paper

We present the program translations from our conference paper (**?**). We think they can still be of interest, although the new translation in Section **??** has some advantages, as it is easier to understand and generalizes the nogoods to the complete interval $[0, n]$ instead of $[1, n]$. The size of both translations is linear on the size of the input programs.

Given some temporal logic program $\Pi$, we say that the rules $r \in \Pi$ such that $At(r) \subseteq \mathcal{A}$ are static, and otherwise we say that they are dynamic.

We start with a simple translation $tr^{\lambda}$ that works for temporal programs where all dynamic rules are integrity constraints. Later, we show that all temporal programs can be translated to this form.

We say that a temporal logic program $\Pi$ over $\mathcal{A}$ is in *previous normal form* (PNF) if $At(\Pi \setminus \Pi^i) \cap \mathcal{A}' = \emptyset$, and that a temporal logic problem $(\Pi, I, F)$ over $\mathcal{A}$ is in PNF if $\Pi$ is in PNF. Given a temporal logic program $\Pi$ over $\mathcal{A}$, let $\Pi^{di}$ denote the set $\{r \mid r \in \Pi^i, At(r) \cap \mathcal{A}' \neq \emptyset\}$ of dynamic integrity constraints of $\Pi$. Note that if $\Pi$ is in PNF, then the dynamic rules of $\Pi$ belong to $\Pi^{di}$. The translation $tr^{\lambda}(\Pi)$ tags the rules in $\Pi^{di}$ with a new atom $\lambda$, that does not belong to $\mathcal{A}$ or $\mathcal{A}'$, and extends the program with a choice rule for $\lambda$. Formally, by $tr^{\lambda}(\Pi)$ we denote the temporal logic program:

$$\Pi \setminus \Pi^{di} \cup \big\{\{\lambda\} \leftarrow \big\} \cup \{\bot \leftarrow Bd(r) \cup \{\lambda\} \mid r \in \Pi^{di}\}.$$

It is easy to see that when $\lambda$ is chosen to be true, $tr^{\lambda}(\Pi)$ generates the same transitions as $\Pi$. Then, we can solve temporal programs $(\Pi, I, F)$ by solving temporal problems $(tr^{\lambda}(\Pi), I, F)$, if we consider only solutions that make $\lambda$ true at all steps after the initial one. For convenience, we consider only the case where $\lambda$ is false. This means that, as in Section **??**, we focus on the $\lambda$-normal solutions. The next proposition states the relation between these $\lambda$-normal solutions and the original solutions using $\Pi$.

*Proposition 1*
Let $\mathcal{T}_1 = (\Pi, I, F)$ and let $\mathcal{T}_2 = (tr^{\lambda}(\Pi), I, F)$ be temporal logic problems. There is a one-to-one correspondence between the solutions to $\mathcal{T}_1$ and the $\lambda$-normal solutions to $\mathcal{T}_2$.

The call $\mathit{CDNL\text{-}ASP}(gen(tr^{\lambda}(\Pi), n),\ I[0] \cup F[n] \cup \{\mathbf{F}\lambda_0, \mathbf{T}\lambda_1, \ldots, \mathbf{F}\lambda_n\})$ computes $\lambda$-normal solutions to $\mathcal{T}_2$, enforcing the correct value for $\lambda$ at every time point using assumptions. The solutions to the original problem $\mathcal{T}_1$ can be extracted from the $\lambda$-normal solutions, after deleting the atoms in $\{\lambda\}[1, n]$.

We turn now our attention to the resolvents $\delta$ of the set of nogoods $\Psi_{tr^{\lambda}(\Pi)}[1, n]$ used by the procedure $\mathit{CDNL\text{-}ASP}$. As we will see, just by looking at these resolvents $\delta$, we can approximate the specific interval $[i, j] \subseteq [1, n]$ of the nogoods that were used to prove them.

To this end, we say that the nogoods containing literals of different steps are dynamic nogoods, and they are static nogoods otherwise. All dynamic nogoods in $\Psi_{tr^{\lambda}(\Pi)}[1, n]$ come from the instantiation of some dynamic integrity constraint $\{\bot \leftarrow Bd(r) \cup \{\lambda\} \mid r \in \Pi^{di}\}$ at some time step $i$ and, therefore, they contain some literal of the form $\mathbf{T}\lambda_i$. On the other hand, in $\Psi_{tr^{\lambda}(\Pi)}[1, n]$ there are no literals of the form $\mathbf{F}\lambda_i$. Hence, the literals $\mathbf{T}\lambda_i$ occurring in the dynamic nogoods can never be resolved away. Then, if some dynamic nogood is used to prove a learned nogood $\delta$, the literal $\mathbf{T}\lambda_i$ occurring in that dynamic nogood must belong to $\delta$. This means that the literals $\mathbf{T}\lambda_i$ from a learned nogood $\delta$ tell us exactly the steps $i$ of the dynamic nogoods that have been used to prove $\delta$.

Observe now that two nogoods $\delta_1 \in \Psi_{tr^{\lambda}(\Pi)}[i]$ and $\delta_2 \in \Psi_{tr^{\lambda}(\Pi)}[i+1]$ can only be resolved if $\delta_2$ is a dynamic nogood. Otherwise, the nogoods would have no opposite literals to resolve. Applying the same reasoning, if two nogoods $\delta_1 \in \Psi_{tr^\lambda(\Pi)}[i]$ and $\delta_2 \in \Psi_{tr^\lambda(\Pi)}[j]$, such that $i < j$, are part of the same resolution proof of a learned nogood $\delta$, then the proof must also contain some dynamic nogoods from each step in the interval $[i+1, j]$. Therefore, the learned nogood $\delta$ must contain the literals $\{\mathbf{T}\lambda\}[i+1, j]$.

This implies that, given the literals $\{\mathbf{T}\lambda\}[k, j]$ occurring in a learned nogood $\delta$, we can infer the following about the nogoods from $\Psi_{tr^\lambda(\Pi)}[1, n]$ used to prove $\delta$: dynamic nogoods from all the steps $[k, j]$ were used to prove $\delta$, possibly some static nogoods of the step $k-1$ were used as well, and no nogoods from other steps were used in the proof. It is possible that some static nogoods at steps $[k, j]$ were also used, but no dynamic nogoods at $k-1$ could be used, since otherwise $\delta$ should contain the literal $\mathbf{T}\lambda_{k-1}$.

We formalize this with the function $\mathit{step}^\lambda(\delta)$, that approximates the specific interval $[i, j]$ of the nogoods that were used to prove $\delta$: if $\delta$ contains some literal of the form $\mathbf{T}\lambda_i$ for $i \in [1, n]$, then $\mathit{step}^\lambda(\delta)$ is the set of steps $\{j-1, j \mid \mathbf{T}\lambda_j \in \delta\}$. For example, if $\delta$ is $\{\mathbf{T}a_3, \mathbf{T}\lambda_3\}$ then the value of $\mathit{step}^\lambda(\delta)$ is $\{2, 3\}$. It is clear that $\delta$ was derived using some dynamic nogood of step 3, that added the literal $\mathbf{T}\lambda_3$. And it could also happen that some static nogood of step 2 was used, but we are uncertain about it. That is why we say that *step* is an approximation. To continue, note that it can also be that $\delta$ has no literals of the form $\mathbf{T}\lambda_i$. In this case, $\delta$ must be the result of resolving some static nogoods of a single time step, and we can extract that time step from the unique time step of the literals occurring in the nogood. Hence, in this case we define $\mathit{step}^\lambda(\delta)$ as $\mathit{step}(\delta)$. For example, $\mathit{step}^\lambda(\{\mathbf{T}c_2, \mathbf{T}d_2\}) = \{2\}$. With this, we can generalize a nogood $\delta$ to the shifted nogoods $\delta\langle t\rangle$ whose *step* value fits in the interval $[1, n]$. We state this precisely in the next theorem.

Observe that it excludes the shifted nogoods $\delta\langle t\rangle$ that contain the literal $\mathbf{T}\lambda_1$, since in that case $\mathit{step}^\lambda(\delta\langle t\rangle)$ contains the step $0 \notin [1, n]$. This makes sense because to prove $\delta\langle t\rangle$ we could need some static nogoods at step 0, and they do not belong to $\Psi_{tr^\lambda(\Pi)}[1, n]$.

*Theorem 1*
Let $\Pi$ be a temporal logic program in PNF, and $\delta$ be a resolvent of $\Psi_{tr^\lambda(\Pi)}[1, m]$ for some $m \geq 1$. Then, for every $n \geq 1$, the set of nogoods $\Psi_{tr^\lambda(\Pi)}[1, n]$ entails the generalization

$$\{\delta\langle t\rangle \mid \mathit{step}^\lambda(\delta\langle t\rangle) \subseteq [1, n]\}.$$

*Example 1*
Consider the call $\mathit{CDNL\text{-}ASP}(\mathit{gen}(\mathit{tr}^\lambda(\Pi_1), 4), \emptyset)$, similar to the one that we have seen before using the original program $\Pi_1$. The nogoods $\Psi_{tr^\lambda(\Pi_1)}[1, n]$ are the same as those in $\Psi_{\Pi_1}[1, n]$, except that every dynamic nogood contains one instantiation of the literal $\mathbf{T}\lambda$. Instead of learning the nogood $\{\mathbf{T}a_3\}$ the algorithm would learn the nogood $\delta = \{\mathbf{T}a_3, \mathbf{T}\lambda_3, \mathbf{T}\lambda_4\}$. Then, applying part (i) of Theorem 9 the nogood $\delta$ can be generalized to $\delta\langle -1\rangle = \{\mathbf{T}a_2, \mathbf{T}\lambda_2, \mathbf{T}\lambda_3\}$, but not to $\delta\langle 1\rangle = \{\mathbf{T}a_4, \mathbf{T}\lambda_4, \mathbf{T}\lambda_5\}$ or to $\delta\langle -2\rangle = \{\mathbf{T}a_1, \mathbf{T}\lambda_1, \mathbf{T}\lambda_2\}$ (see Figure 1).

The next step is to show how temporal programs in general can be translated to PNF form. For this, given a temporal logic program $\Pi$ over $\mathcal{A}$, let $\mathcal{A}^* = \{a^* \mid a \in \mathcal{A}\}$, and assume that this set is disjoint from $\mathcal{A}$ and $\mathcal{A}'$. The translation $tr^*(\Pi)$ consists of two parts. The first part consists of the result of replacing in $\Pi$ every atom $a' \in \mathcal{A}'$ by its corresponding new atom $a^*$. The second part consists of the union of the rules

$$\{\{a^*\} \leftarrow; \bot \leftarrow a', \neg a^*; \bot \leftarrow \neg a', a^*\}$$

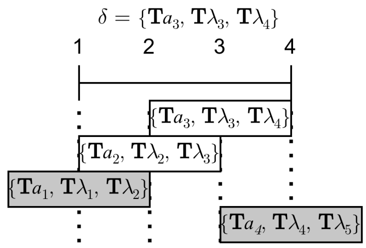


Fig. 1. Representation of different shifted versions of the nogood $\delta = \{\mathbf{T}a_3, \mathbf{T}\lambda_3, \mathbf{T}\lambda_4\}$. The surrounding rectangles cover the interval of their *step* value. For example, the rectangle of $\{\mathbf{T}a_2, \mathbf{T}\lambda_2, \mathbf{T}\lambda_3\}$ covers the interval $[1, 3]$ because $step(\{\mathbf{T}a_2, \mathbf{T}\lambda_2, \mathbf{T}\lambda_3\}) = [1, 3]$.

for every $a \in \mathcal{A}$. The idea of the translation is that the atoms $a' \in \mathcal{A}'$ are confined to integrity constraints by replacing them by new atoms $a^* \in \mathcal{A}^*$, whose truth value is completely determined by the corresponding $a' \in \mathcal{A}'$ atoms by means of the last set of rules.

*Proposition 2*
For any temporal logic program $\Pi$, the program $tr^*(\Pi)$ is in PNF.

The solutions to temporal problems with $\Pi$ are the same as the solutions to the same temporal problems with $tr^*(\Pi)$ where the atoms $a^*[i]$ are false at $i = 0$ and have the truth value of $a[i-1]$ at the other time steps $i$. Just like before, when we use this translation, we have to add to *CDNL-ASP* the correct assumptions to fix the value of the $a^*$ atoms at step $0$.

*Proposition 3*
Let $\mathcal{T}_1 = (\Pi, I, F)$ and let $\mathcal{T}_2 = (tr^*(\Pi), I, F)$ be temporal logic problems. There is a one-to-one correspondence between the solutions to $\mathcal{T}_1$ and the solutions to $\mathcal{T}_2$ that do not contain any atom $p^* \in \mathcal{P}^*$ at step $0$.

This proposition allows us to replace any temporal program $\Pi$ by a temporal program $tr^*(\Pi)$ in PNF. We can then apply the translation $tr^\lambda$ and benefit from Theorem 9. In fact, we can go one step further, and apply the nogoods learned with the program $tr^\lambda(tr^*(\Pi))$ directly to the original problem with $\Pi$. We make this claim precise in the next theorem. We extend our definition of the simplification of a nogood $\delta$, $simp(\delta)$, to accommodate literals over $\mathcal{A} \cup \mathcal{A}^*$. That is, $simp(\delta)$ is the nogood $\{\mathbf{V}a_i \mid \mathbf{V}a_i \in \delta, \mathbf{V} \in \{\mathbf{T}, \mathbf{F}\}, a \in \mathcal{A}\} \cup \{\mathbf{V}a_{i-1} \mid \mathbf{V}a^*_i \in \delta, \mathbf{V} \in \{\mathbf{T}, \mathbf{F}\}, a^* \in \mathcal{A}^*\}$ that results from skipping the $\lambda_i$ literals of $\delta$, and replacing the atoms $a^*_i$ by their corresponding atoms $a_{i-1}$.

*Theorem 2*
Let $\Pi$ be a temporal logic program, and $\delta$ be a resolvent of $\Psi_{tr^\lambda(tr^*(\Pi))}[1, m]$ for some $m \geq 1$. Then, for every $n \geq 1$, the set of nogoods $\Psi_\Pi[1, n]$ entails the generalization

$$\{simp(\delta\langle t\rangle) \mid step^\lambda(\delta\langle t\rangle) \subseteq [1, n]\}.$$

## 3 Proofs

*Lemma 1*
For any temporal logic program $\Pi$, $\Sigma_\Pi[n] = \Sigma_{\Pi[n]}$.

*Proof*

1. **Constraints**
For any constraint $c$ of the form $\bot \leftarrow a_1, ...a_m, not\ b_{m+1}, ..., not\ b_l$ it holds that

$$\begin{aligned}
c[n] &= \bot \leftarrow a_1[n], ...a_m[n], not\ b_{m+1}[n], ..., not\ b_l[n] \\
\Sigma_{c[n]} &= \mathbf{T}a_1[n], ...\mathbf{T}a_m[n], \mathbf{F}b_{m+1}[n], ..., \mathbf{F}b_l[n] \\
\Sigma_c &= \mathbf{T}a_1, ...\mathbf{T}a_m, \mathbf{F}b_{m+1}, ..., \mathbf{F}b_l \\
\Sigma_c[n] &= \mathbf{T}a_1[n], ...\mathbf{T}a_m[n], \mathbf{F}b_{m+1}[n], ..., \mathbf{F}b_l[n]
\end{aligned}$$

Since all constraints have the form of $c$, we can conclude that $\Sigma_{c[n]} = \Sigma_c[n]$ for any constraint.
2. **Body**
For any body $B$ of the form $\{a_1, ...a_m, not\ b_{m+1}, ..., not\ b_l\}$ it holds that

$$\begin{aligned}
B[n] &= a_1[n], ...a_m[n], not\ b_{m+1}[n], ..., not\ b_l[n] \\
\Sigma_{B[n]} &= \{\{\mathbf{T}B[n], \mathbf{F}a_1[n]\}, ..., \{\mathbf{T}B[n], \mathbf{F}b_l[n]\}\} \cup \{\mathbf{F}B[n], \mathbf{T}a_1[n], ..., \mathbf{F}b_l[n]\} \\
\Sigma_B &= \{\{\mathbf{T}B, \mathbf{F}a_1\}, ..., \{\mathbf{T}B, \mathbf{F}b_l\}\} \cup \{\mathbf{F}B, \mathbf{T}a_1, ..., \mathbf{F}b_l\} \\
\Sigma_B[n] &= \{\{\mathbf{T}B[n], \mathbf{F}a_1[n]\}, ..., \{\mathbf{T}B[n], \mathbf{F}b_l[n]\}\} \cup \{\mathbf{F}B[n], \mathbf{T}a_1[n], ..., \mathbf{F}b_l[n]\}
\end{aligned}$$

Since all bodies have the form of $B$, we can then conclude that $\Sigma_{B[n]} = \Sigma_B[n]$.
3. **A set of rules with the same head**
For any set of rules with the same head $\Pi$ of the form $\{a \leftarrow B_1, ..., a \leftarrow B_l\}$ where $B_i$ are bodies and $\Sigma_B$ is the set of all body nogoods it holds that

$$\begin{aligned}
\Pi[n] &= \{a[n] \leftarrow B_1[n], ..., a[n] \leftarrow B_l[n]\} \\
\Sigma_{\Pi[n]} &= \{\{\mathbf{F}B_1[n], ..., \mathbf{F}B_l[n], \mathbf{T}a[n]\}, \{\mathbf{T}B_1[n], \mathbf{F}a[n]\}, ..., \{\mathbf{T}B_l[n], \mathbf{F}a[n]\}\} \cup \Sigma_B \\
\Sigma\Pi &= \{\{\mathbf{F}B_1, ..., \mathbf{F}B_l, \mathbf{T}a\}, \{\mathbf{T}B_1, \mathbf{F}a\}, ..., \{\mathbf{T}B_l, \mathbf{F}a\} \cup \Sigma_B\} \\
\Sigma_\Pi[n] &= \{\{\mathbf{F}B_1[n], ..., \mathbf{F}B_l[n], \mathbf{T}a[n]\}, \{\mathbf{T}B_1[n], \mathbf{F}a[n]\}, ..., \{\mathbf{T}B_l[n], \mathbf{F}a[n]\}\} \cup \Sigma_B
\end{aligned}$$

Since body nogoods are also equal, we can conclude that $\Sigma_{\Pi[n]} = \Sigma_\Pi[n]$.
4. **Choice rules**
Since choice rule nogoods are a subset of normal rule nogoods, we can conclude that $\Sigma_{c[n]} = \Sigma_c[n]$ for any choice rule $c$.
5. **Loops**
For any set of rules $\Pi$ forming a loop of the form $a_1 \leftarrow a_2, B_1, ..., a_n \leftarrow a_1, B_n$ with external Bodies for (some) $a_i$ being labeled $E_i$ and $\Sigma_R$ is the set of all rule nogoods it holds that

$$
\begin{aligned}
\Pi[n] &= \{a_1[n] \leftarrow a_2[n], B_1[n], ..., a_n[n] \leftarrow a_1[n], B_n[n]\} \\
\Sigma_{\Pi[n]} &= \{\{\mathbf{T}a_1[n], \mathbf{F}E_{i_1}[n], ..., \mathbf{F}E_{i_m}[n]\}, ..., \{\mathbf{T}a_n[n], \mathbf{F}E_{i_1}[n], ..., \mathbf{F}E_{i_m}[n]\}\} \cup \Sigma_R \\
\Sigma_{\Pi} &= \{\{\mathbf{T}a_1, \mathbf{F}E_{i_1}, ..., \mathbf{F}E_{i_m}\}, ..., \{\mathbf{T}a_n, \mathbf{F}E_{i_1}, ..., \mathbf{F}E_{i_m}\}\} \cup \Sigma_R \\
\Sigma_{\Pi}[n] &= \{\{\mathbf{T}a_1[n], \mathbf{F}E_{i_1}[n], ..., \mathbf{F}E_{i_m}[n]\}, ..., \{\mathbf{T}a_n[n], \mathbf{F}E_{i_1}[n], ..., \mathbf{F}E_{i_m}[n]\}\} \cup \Sigma_R
\end{aligned}
$$

Since rule nogoods are also equal, we can conclude that $\Sigma_{\Pi[n]} = \Sigma_{\Pi}[n]$.

From items 1, 2, 3, 4, 5 we can say that for any program $\Pi$, $\Sigma_{\Pi[n]} = \Sigma_{\Pi}[n]$. □

*Proposition 1*

If $\Pi$ is a temporal logic program and $n \geq 1$ then $\Sigma_{gen(\Pi,n)} = \Psi_{\Pi}[1, n]$.

*Proof*

Let $C = \{\{a'\} \leftarrow | a \in \mathcal{A}\}$ where $\mathcal{A}$ is the set of atoms ocurring in $\Pi$. Since all the rules in $C$ are choice rules with empty bodies, $\Sigma_{C[n]}$ is comprised of nogoods of the form $\{\mathbf{T}a'[n], \mathbf{F}\emptyset\}$. Given that $\mathbf{F}\emptyset$ is always false the nogoods can be safely removed. Hence, for any program $\Pi$ it holds that $\Sigma_{C[n]} \cup \Sigma_{\Pi} = \Sigma_{\Pi}$.

For a given temporal logic program $\Pi$ we can define $trans(\Pi) = C \cup \Pi$. Additionally, $gen(\Pi, n)$ can be defined as $C[1] \cup \Pi[1, n]$, which means that

$$
\begin{aligned}
\Sigma_{gen(\Pi,n)} &= \Sigma_{C[1]} \cup \Sigma_{\Pi[1,n]} \\
&= \Sigma_{C[1]} \cup \Sigma_{\Pi[1]} \cup ... \cup \Sigma_{\Pi[n]} \\
&= \Sigma_{\Pi[1]} \cup ... \cup \Sigma_{\Pi[n]} && \text{(deleting choice nogoods)}
\end{aligned}
$$

Also,

$$
\begin{aligned}
\Psi_{\Pi}[1, n] &= \Sigma_{trans(\Pi)}[1, n] \\
&= \Sigma_{trans(\Pi)}[1] \cup ... \cup \Sigma_{trans(\Pi)}[n] \\
&= \Sigma_C[1] \cup \Sigma_{\Pi}[1] \cup ... \cup \Sigma_C[n] \cup \Sigma_{\Pi}[n] \\
&= \Sigma_{\Pi}[1] \cup ... \cup \Sigma_{\Pi}[n] && \text{(deleting choice nogoods)} \\
&= \Sigma_{\Pi[1]} \cup ... \cup \Sigma_{\Pi[n]} && \text{(lemma1)} \\
&= \Sigma_{gen(\Pi,n)}
\end{aligned}
$$

□

*Theorem 2*

Let $(\Pi, I, F)$ be a temporal logic problem over $\mathcal{A}$. The pair $(X, n)$ is a solution to $(\Pi, I, F)$ for $n \geq 1$ and $X \subseteq \mathcal{A}[0, n]$ iff $X = S^{\mathbf{T}} \cap \mathcal{A}[0, n]$ for a (unique) solution $S$ for $\Psi_{\Pi}[1, n]$ such that $I[0] \cup F[n] \subseteq S$.

*Proof*

Let $\mathcal{A}$ be the set of atoms occurring in $\Pi$.

By Proposition 1 a solution for the set of nogoods $\Psi_{\Pi}[1, n]$ is a solution for $\Sigma_{gen(\Pi,n)}$. A solution for $\Sigma_{gen(\Pi,n)}$ is a stable model for the generator program $gen(\Pi, n)$. Since a stable model of $gen(\Pi, n)$ consistent with $I$ and $F$ is a solution of $(\Pi, I, F)$, then a solution $S$ of $\Psi_\Pi[1, n]$ consistent with some $I$ and $F$, the pair $(X, n)$ where $X = S^{\mathbf{T}} \cap \mathcal{A}[1, n]$ is a solution for $(\Pi, I, F)$.

Let $(X, n)$ be a solution to the temporal logic problem $(\Pi, I, F)$. By definition, $X$ is a stable model of $gen(\Pi, n)$ consistent with $I$ and $F$. Since a stable model of $gen(\Pi, n)$ is a solution of $\Sigma_{gen(\Pi,n)}$ which is a solution of $\Psi_\Pi[1, n]$ (by Proposition 1), it follows that $S = \{\mathbf{T}a | a \in X\} \cup \{\mathbf{F}a | a \in \mathcal{A}[0, n] \setminus X\}$ is a solution for the temporal logic program $\Psi_\Pi[1, n]$ such that $I[0] \cup F[n] \subseteq S$. $\square$

*Lemma 2*
For any resolvent $\delta$ of $\Psi[i, j]$ it holds that $\delta\langle t\rangle$ is a resolvent of $\Psi[i + t, j + t]$

*Proof*
Recall that if a nogood is a resolvent of $\Psi[i, j]$ then it must have a resolution proof $\mathcal{T}$ where every nogood $\delta_i \in \mathcal{T}$ is either entailed by $\Psi[i, j]$ or the result of resolving some $\delta_j$ and $\delta_k$ where $j < k < i$ and both $\delta_j$ and $\delta_k$ are entailed by $\Psi[i, j]$. Additionally, for a resolution proof $\mathcal{T} = \delta_1, \dots \delta_n$ the result is $\delta_n$. Finally, note that if a nogood $\delta \in \Psi[i, j]$ then $\delta\langle t\rangle \in \Psi[i + t, j + t]$

We now prove the lemma by induction. Let $\mathcal{T}$ be the resolution proof of a nogood $\delta$ that is entailed by $\Psi[i, j]$.

***Induction base 1:*** If $\mathcal{T} = \delta$ then $\delta \in \Psi[i, j]$ holds and, trivially, $\delta\langle t\rangle \in \Psi[i + t, j + t]$.

***Induction base 2:*** If $\mathcal{T} = \delta_1, \delta_2$ then, since there less than two nogoods before $\delta_1$ and $\delta_2$ then both must be in $\Psi[i, j]$. Consequently, $\delta_1\langle t\rangle \in \Psi[i + t, j + t]$ and $\delta_2\langle t\rangle \in \Psi[i + t, j + t]$.

***Induction step n:*** Let $\mathcal{T} = \delta_1, ..., \delta_n$ be a resolution proof for nogood $\delta_n$. If $\delta_n \in \Psi[i, j]$ then, trivially, $\delta_n\langle t\rangle \in \Psi[i + t, j + t]$. If $\delta_n \notin \Psi[i, j]$ then we know by induction that all $\delta_i$ with $0 \leq i \leq n - 1$ are entailed by $\Psi[i, j]$. Since $\delta_n \notin \Psi[i, j]$ then there are some $\delta_k$ and $\delta_l$ where $k < l < i$ that resolve to $\delta_n$. By induction, $\delta_k\langle t\rangle$ and $\delta_l\langle t\rangle$ are entailed by $\Psi[i + t, j + t]$. Consequently, $\delta_n\langle t\rangle$ is entailed by $\Psi[i + t, j + t]$. $\square$

*Theorem 3*
Let $\Pi$ be a temporal logic program, and $\delta$ be a resolvent of $\Psi_\Pi[i, j]$ for some $i$ and $j$ such that $1 \leq i \leq j$. Then, for any $n \geq 1$, the set of nogoods $\Psi_\Pi[1, n]$ entails the generalization

$$\{\delta\langle t\rangle \mid [i + t, j + t] \subseteq [1, n]\}.$$

*Proof*
Let $\delta$ be a resolvent of $\Psi[i, j]$. Then the shifted nogood $\delta\langle t\rangle$ is entailed by $\Psi[i + t, j + t]$ (Lemma 2). Let $t$ be a value where $[i + t, j + t] \subseteq [1, n]$ holds, then $\delta\langle t\rangle$ is entailed by $\Psi[1, n]$ since $\Psi[i + t, j + t] \subseteq \Psi[1, n]$. $\square$

*Theorem 4*
Let $(\Pi, I, F)$ be a temporal logic problem over $\mathcal{A}$, $n \geq 1$, and $X$ be a set of atoms over $\mathcal{A}[0, n]$. Then, the following statements are equivalent:

- The pair $(X, n)$ is a solution to $(\Pi, I, F)$.
- $X = S^{\mathbf{T}} \cap \mathcal{A}[0, n]$ for a solution $S$ for $\Psi_\Pi[1, n]$ such that $I[0] \cup F[n] \subseteq S$.
- There is a path $(X_0, \dots, X_n)$ in $G(\Pi)$ such that $X = \bigcup_{i \in [0,n]} X_i[i]$, the state $X_0$ is consistent with $I$, and the state $X_n$ is consistent with $F$.

*Proof*

The solution to the temporal logic problem $(\Pi, I, F)$ is a stable model of $gen(\Pi, n)$ consistent with $I[0]$ and $F[0]$. We can split $gen(\Pi, n)$ as follows: Let $C = \{\{a\} \leftarrow |a \in \mathcal{A}\}$

$$C[0] \cup \Pi[1] \cup ... \cup \Pi[n]$$

where $\mathcal{A}$ is the set of atoms ocurring in $\Pi$.

From the Splitting Set Theorem (**?**) it follows that we can build every stable model $X$ for $gen(\Pi, n)$ as follows:

$$\begin{aligned}
&L_0 \text{ is a stable model of } C[0]\\
&L_1 \text{ is a stable model of } \Pi[1] \cup L_0\\
&...\\
&L_n \text{ is a stable model of } \Pi[n] \cup L_{n-1}
\end{aligned}$$

where $L_n$ is a stable model of $gen(\Pi, n)$.

It is easy to see that every $L_{i-1} \subseteq L_i$ where $1 \leq i \leq n$. Let $Z_i = L_i \cap \mathcal{A}[i]$ with $0 \leq i \leq n$, then program $\Pi[i] \cup L_{i-1}$ can be rewritten as $\Pi[i] \cup Z_{i-1} \cup ... \cup Z_0$.

$M$ is a stable model of $\Pi[i] \cup Z_{i-1} \cup ... \cup Z_0$ iff $M$ has the form $M_i \cup Z_{i-2} \cup ... \cup Z_0$ for some stable model $M_i$ of $trans(\Pi)[i]$ such that $Z_{i-1} = M_i \cap P[i-1]$. This follows from the fact that $trans(\Pi)[i] = \Pi[i] \cup C[i-1]$. Following the Splitting Set Theorem, we can build a stable model for $trans(\Pi)[i]$ by first getting a model $S$ for $C[i-1]$ and then a model for $\Pi[i] \cup S$. Since $C$ is comprised of choice rules for all atoms, then the assignment formed from $Z_{i-1}$ is a stable model of $C[i-1]$. Thus, a stable model of $\Pi[i] \cup Z_{i-1}$ is a stable model of $trans(\Pi)[i]$.

This also means that $Z_i = M_i \cap P[i]$ is a state in $G(\Pi)$ and that $(Z_i, Z_{i-1})$ is an edge.

Consequently, we can say that the states $Z_0, ..., Z_n$ form a path in the graph $G(\Pi)$. Finally, for any stable model of $gen(\Pi, n)$ consistent with $I[0]$ and $F[n]$, then the states $Z_0, ..., Z_n$ form a path in $G(\Pi)$ and $I[0]$ and $F[n]$ are consistent with $Z_0$ and $Z_n$ respectively. □

*Proposition 2*

Let $\Pi$ be a temporal logic program over $\mathcal{A}$, $n \geq 1$, and let $\delta$ be a (non-temporal) nogood over $\mathcal{A}[0, n]$. Then, the following two statements are equivalent:

- The set of nogoods $\Psi_\Pi[1, n]$ entails $\delta$.
- Every path $(X_0, \ldots, X_n)$ of length $n$ in $G(\Pi)$ does not violate $\delta$.

*Proof*

By Theorem 4 when $I$ and $F$ are empty, the solutions to $\Psi_\Pi[i, j]$ correspond to paths of length $j - i + 1$ in $G(\Pi)$. This means that no path of this length violates a nogood in $\Psi_\Pi[i, j]$.

Since $\delta$ is entailed by $\Psi_\Pi[i, j]$, then no path of length $j - i + 1$ in $G(\Pi)$ violates $\delta$. □

*Proposition 3*

Let $\Pi$ be a internal temporal program. If $(X_0, \ldots, X_n)$ is a path of length $n$ in $G(\Pi)$, then for any $i, j \geq 0$ there is a path $(Y_0, \ldots, Y_{n+i+j})$ of length $n + i + j$ in $G(\Pi)$ such that $(X_0, \ldots, X_n) = (Y_i, \ldots, Y_{n+i})$.

*Proof*
Let $Y_i = X_0, \dots,$ and $Y_{n+i} = X_n$. We have that $(Y_i, \dots, Y_{n+i})$ is a path in $G(\Pi)$. Since $\Pi$ is internal, there is some edge $(Y_{i-1}, Y_i)$ in $G(\Pi)$, and there is also some edge $(Y_{i-2}, Y_{i-1})$ in $G(\Pi)$, and so on. Hence, there is a path $(Y_0, \dots, Y_{n+i})$ of length $n+i$ in $G(\Pi)$. Similarly, since $G(\Pi)$ is internal, in $G(\Pi)$ there are edges $(Y_{n+i}, Y_{n+i+1})$, $(Y_{n+i+1}, Y_{n+i+2})$, and so on. Hence, there is a path $(Y_0, Y_{n+i+j})$ in $G(\Pi)$ of length $n+i+j$ such that $(X_0, \dots, X_n) = (Y_i, \dots, Y_{n+i})$. □

*Theorem 5*
Let $\Pi$ be a temporal logic program, and $\delta$ be a resolvent of $\Psi_\Pi[i, j]$ for $1 \leq i \leq j$. If $\Pi$ is internal, for any $n \geq 1$, the set of nogoods $\Psi_\Pi[1, n]$ entails the generalization

$$\{\delta\langle t\rangle \mid \mathit{step}(\delta\langle t\rangle) \subseteq [0, n]\}.$$

*Proof*
We prove the case where $\delta$ consist of normal atoms, the proof for the general case follows the same lines.

Let $\Pi$ be defined over some set of atoms $\mathcal{A}$. Given that $\delta$ is a resolvent of $\Psi_\Pi[i, j]$, its atoms must belong to some smallest set $\mathcal{A}[k, l]$ such that $0 \leq i-1 \leq k \leq l \leq j$. Then, the integers $t$ such that $\mathit{step}(\delta\langle t\rangle) \subseteq [0, n]$ are exactly the $t$'s such that $-k \leq t \leq n-l$. Hence, to prove this theorem we just have to prove that for every $t$ such that $-k \leq t \leq n-l$ the set of nogoods $\Psi_\Pi[1, n]$ entails $\delta\langle t\rangle$.

Since $\delta$ is a resolvent of $\Psi_\Pi[i, j]$, the shifted version $\delta\langle 1-i\rangle$ is a resolvent of $\Psi_\Pi[1, j+1-i]$, and therefore $\Psi_\Pi[1, j+1-i]$ entails $\delta\langle 1-i\rangle$. By Proposition 2, every path $(X_0, \dots, X_{j+1-i})$ in $G(\Pi)$ does not violate $\delta\langle 1-i\rangle$. We consider two cases: $k < l$ and $k = l$.

*Case 1 ($k < l$).* Since $\Pi$ is internal, we can prove by contradiction that every path $(Y_0, \dots, Y_{l-k})$ in $G(\Pi)$ does not violate $\delta\langle -k\rangle$.

Assume that there is such a path $(Y_0, \dots, Y_{l-k})$. By Proposition 3, there is a path $(X_0, \dots, X_{j+1-i})$ in $G(\Pi)$ such that $(Y_0, \dots, Y_{l-k}) = (X_{k+1-i}, \dots, X_{l+1-i})$. Given that $(Y_0, \dots, Y_{l-k})$ violates $\delta\langle -k\rangle$, the path $(X_0, \dots, X_{j+1-i})$ would violate $\delta\langle -k+(k+1-i)\rangle = \delta\langle 1-i\rangle$, which contradicts one of our previous statements.

If every path $(Y_0, \dots, Y_{l-k})$ in $G(\Pi)$ does not violate $\delta\langle -k\rangle$, then for every path $(X_0, \dots, X_n)$ in $G(\Pi)$ and every $t$ such that $-k \leq t \leq n-l$, the shifted nogood $\delta\langle t\rangle$ is not violated. Then, by Theorem 4, we can conclude that for every $t$ such that $-k \leq t \leq n-l$, the solutions to $\Psi_\Pi[1, n]$ do not violate $\delta\langle t\rangle$, and therefore $\Psi_\Pi[1, n]$ entails $\delta\langle t\rangle$.

*Case 2 ($k = l$).* Note that since all atoms of $\delta$ belong to $\mathcal{A}[k]$, all atoms of $\delta\langle -k\rangle$ belong to $\mathcal{A}[0]$. Given that $\Pi$ is internal, we can prove by contradiction that every path $(Y_0, Y_1)$ in $G(\Pi)$ does not violate $\delta\langle -k\rangle$.

Assume that there is such a path $(Y_0, Y_1)$. Then, by Proposition 3, there is a path of the form $(X_0, \dots, X_{j+1-i}, X_{j+2-i})$ in $G(\Pi)$ such that $(Y_0, Y_1) = (X_{k+1-i}, X_{k+2-i})$. This path violates $\delta\langle -k+(k+1-i)\rangle = \delta\langle 1-i\rangle$. Since the atoms of $\delta\langle 1-i\rangle$ belong to $\mathcal{A}[k+1-i]$ and $k+1-i < j+2-i$, the subpath $(X_0, \dots, X_{j+1-i})$ in $G(\Pi)$ also violates $\delta\langle 1-i\rangle$, which contradicts one of our previous statements.

Given that every path $(Y_0, Y_1)$ in $G(\Pi)$ does not violate $\delta\langle -k\rangle$, it follows that for every path $(X_0, \dots, X_n, X_{n+1})$ in $G(\Pi)$ and every $t$ such that $-k \leq t \leq n-l$, the shifted nogood $\delta\langle t\rangle$ is not violated. Since the atoms of $\delta\langle t\rangle$ belong to $\mathcal{A}[k+t]$ and $k+t < n+1$ for all $t$, then the previous statement also holds for all paths $(X_0, \dots, X_n)$ in $G(\Pi)$. Finally, we can reason as in the previous case, and by Theorem 4 conclude that for all $t$ such that $-k \leq t \leq n-l$, the set of nogoods $\Psi_\Pi[1,n]$ entails $\delta\langle t\rangle$. $\square$

*Proposition 4*

Let $\Pi$ be a temporal logic program and $I$ be a partial assignment such that $\Pi$ is internal wrt $I$. If $(X_0,\ldots,X_n)$ is a path of length $n$ in $G(\Pi)$ and $X_0$ is initial or reachable wrt $I$, then for any $i,j \geq 0$ there is a path $(Y_0,\ldots,Y_{n+i+j})$ of length $n+i+j$ in $G(\Pi)$ such that $(X_0,\ldots,X_n) = (Y_i,\ldots,Y_{n+i})$.

*Proof*

Let $Y_i = X_0$, ..., and $Y_{n+i} = X_n$. We have that $(Y_i,\ldots,Y_{n+i})$ is a path in $G(\Pi)$ where $Y_i$ is initial or reachable wrt $I$. If $Y_i$ is initial wrt $I$, since $\Pi$ is internal wrt $I$, $Y_i$ is also loop-reachable, and therefore there is a path $(Y_0,\ldots,Y_i)$ of length $i$ in $G(\Pi)$, that may go through a loop in $G(\Pi)$ as many times as necessary. Similarly, if $Y_i$ is reachable wrt $I$, since $\Pi$ is internal wrt $I$, there is a path $(Y_0,\ldots,Y_i)$ of length $i$ in $G(\Pi)$ that may go through a loop in $G(\Pi)$ and through some initial state wrt $I$. Both cases imply that there is a path $(Y_0,\ldots,Y_{n+i})$ of length $n+i$ in $G(\Pi)$.

On the other direction, in $G(\Pi)$ there are edges $(Y_{n+i},Y_{n+i+1})$, $(Y_{n+i+1},Y_{n+i+2})$, and so on. These edges must exist because the states occurring in them are reachable wrt $I$, and therefore they must have some outgoing edge. This gives us a path $(Y_0,\ldots,Y_{n+i+j})$ in $G(\Pi)$ of length $n+i+j$ such that $(X_0,\ldots,X_n) = (Y_i,\ldots,Y_{n+i})$. $\square$

*Proposition 5*

A temporal logic program $\Pi$ is internal iff it is internal wrt the empty assignment.

*Proof*

*From left to right.* Assume $\Pi$ is internal. To prove condition (i) of being internal wrt $\emptyset$, take any initial state $X$ wrt $\emptyset$. Since $\Pi$ is internal, $X$ has some predecessor in $G(\Pi)$. Similarly, each predecessor must have another predecessor, and so on. Given that $G(\Pi)$ is finite, at some point one of these states must be repeated, which implies that $X$ is loop-reachable and condition (i) holds. Condition (ii) of being internal wrt $\emptyset$ follows directly from the assumption.

*From right to left.* Assume that $\Pi$ is internal wrt $\emptyset$. The initial states wrt $\emptyset$ are the states of $G(\Pi)$ that have some outgoing edge. By the assumption, these states are loop-reachable. Hence, they have some incoming edge, which implies that they are internal. The reachable states wrt $\emptyset$ are the states of $G(\Pi)$ that have some incoming edge. Again, by the assumption, these states are also internal. This shows that states with incoming or incoming edges are internal, and therefore $\Pi$ is internal. $\square$

*Theorem 6*

Let $\Pi$ be a temporal logic program, $I$ be a partial assignment, and $\delta$ be a resolvent of $\Psi_\Pi[i,j]$ for $1 \leq i \leq j$. If $\Pi$ is internal wrt $I$, then for any $n \geq 1$, the set of nogoods $\Psi_\Pi[1,n] \cup \mathit{nogoods}(I)$ entails the generalization

$$\{\delta\langle t\rangle \mid \mathit{step}(\delta\langle t\rangle) \subseteq [0,n]\}.$$

*Proof*

The proof is similar to the proof of Theorem 5. We prove the case where $\delta$ consist of normal atoms, the proof for the general case follows the same lines.

Let $\Pi$ be defined over some set of atoms $\mathcal{A}$. Given that $\delta$ is a resolvent of $\Psi_\Pi[i,j]$, its atoms must belong to some smallest set $\mathcal{A}[k,l]$ such that $0 \leq i-1 \leq k \leq l \leq j$. Then, the integers $t$ such that $\mathit{step}(\delta\langle t\rangle) \subseteq [0,n]$ are exactly the $t$'s such that $-k \leq t \leq n-l$. Hence, to prove this theorem we just have to prove that for every $t$ such that $-k \leq t \leq n-l$ the set of nogoods $\Psi_\Pi[1,n] \cup \mathit{nogoods}(I)$ entails $\delta\langle t\rangle$.

Since $\delta$ is a resolvent of $\Psi_\Pi[i,j]$, the shifted version $\delta\langle 1-i\rangle$ is a resolvent of $\Psi_\Pi[1,j+1-i]$, and therefore $\Psi_\Pi[1,j+1-i]$ entails $\delta\langle 1-i\rangle$. By Proposition 2, every path $(X_0,\dots,X_{j+1-i})$ in $G(\Pi)$ does not violate $\delta\langle 1-i\rangle$. We consider two cases: $k<l$ and $k=l$.

*Case 1 ($k<l$).* Since $\Pi$ is internal wrt $I$, we can prove by contradiction that every path $(Y_0,\dots,Y_{l-k})$ in $G(\Pi)$ where $Y_0$ is initial or reachable wrt $I$ does not violate $\delta\langle -k\rangle$.

Assume that there is such a path $(Y_0,\dots,Y_{l-k})$. By Proposition 4, there is a path of the form $(X_0,\dots,X_{j+1-i})$ in $G(\Pi)$ such that $(Y_0,\dots,Y_{l-k}) = (X_{k+1-i},\dots,X_{l+1-i})$. Given that $(Y_0,\dots,Y_{l-k})$ violates $\delta\langle -k\rangle$, the path $(X_0,\dots,X_{j+1-i})$ would violate $\delta\langle -k+(k+1-i)\rangle = \delta\langle 1-i\rangle$, which contradicts one of our previous statements.

If every path $(Y_0,\dots,Y_{l-k})$ in $G(\Pi)$ where $Y_0$ is initial or reachable wrt $I$ does not violate $\delta\langle -k\rangle$, then for every path $(X_0,\dots,X_n)$ in $G(\Pi)$ where $X_0$ is initial wrt $I$ and every $t$ such that $-k \leq t \leq n-l$, the shifted nogood $\delta\langle t\rangle$ is not violated. Theorem 4 gives us a correspondence between the paths $(X_0,\dots,X_n)$ in $G(\Pi)$ and the solutions to $\Psi_\Pi[1,n]$. It is easy to see that, if we add to $\Psi_\Pi[1,n]$ the set $\mathit{nogoods}(I)$, we also have a correspondence between the paths $(X_0,\dots,X_n)$ in $G(\Pi)$ where $X_0$ is initial wrt $I$ and the solutions to $\Psi_\Pi[1,n] \cup \mathit{nogoods}(I)$. Hence, we can conclude that the solutions to $\Psi_\Pi[1,n] \cup \mathit{nogoods}(I)$ do not violate $\delta\langle t\rangle$, and therefore $\Psi_\Pi[1,n] \cup \mathit{nogoods}(I)$ entails $\delta\langle t\rangle$.

*Case 2 ($k=l$).* Note that since all atoms of $\delta$ belong to $\mathcal{A}[k]$, all atoms of $\delta\langle -k\rangle$ belong to $\mathcal{A}[0]$. Given that $\Pi$ is internal wrt $I$, we can prove by contradiction that every path $(Y_0,Y_1)$ in $G(\Pi)$ where $Y_0$ is initial or reachable wrt $I$ does not violate $\delta\langle -k\rangle$.

Assume that there is such a path $(Y_0,Y_1)$. Then, by Proposition 4, there is a path of the form $(X_0,\dots,X_{j+1-i},X_{j+2-i})$ in $G(\Pi)$ such that $(Y_0,Y_1) = (X_{k+1-i},X_{k+2-i})$. This path violates $\delta\langle -k+(k+1-i)\rangle = \delta\langle 1-i\rangle$. Since the atoms of $\delta\langle 1-i\rangle$ belong to $\mathcal{A}[k+1-i]$ and $k+1-i < j+2-i$, the subpath $(X_0,\dots,X_{j+1-i})$ in $G(\Pi)$ also violates $\delta\langle 1-i\rangle$, which contradicts one of our previous statements.

Given that every path $(Y_0,Y_1)$ in $G(\Pi)$ where $Y_0$ is initial or reachable wrt $I$ does not violate $\delta\langle -k\rangle$, it follows that for every path $(X_0,\dots,X_n,X_{n+1})$ in $G(\Pi)$ where $X_0$ is initial wrt $I$ and every $t$ such that $-k \leq t \leq n-l$, the shifted nogood $\delta\langle t\rangle$ is not violated. Since the atoms of $\delta\langle t\rangle$ belong to $\mathcal{A}[k+t]$ and $k+t<n+1$ for all $t$, then the previous statement also holds for all paths $(X_0,\dots,X_n)$ in $G(\Pi)$ where $X_0$ is initial wrt $I$. Finally, we can reason as in the previous case, and by Theorem 4 conclude that for all $t$ such that $-k \leq t \leq n-l$, the set of nogoods $\Psi_\Pi[1,n] \cup \mathit{nogoods}(I)$ entails $\delta\langle t\rangle$. □

*Proposition 6*

Let $\Pi$ be a temporal logic program, and let $E$ be the set of edges in its transition graph $G(\Pi)$. The set of edges in $G(\lambda(\Pi))$ is the union of the following four sets:

- $\{(X, Y \cup \{\lambda\}) \mid (X,Y) \in E\}$

- $\{(X \cup \{\lambda\}, Y \cup \{\lambda\}) \mid (X, Y) \in E\}$
- $\{(X, Y) \mid X, Y \subseteq \mathcal{A}\}$
- $\{(X \cup \{\lambda\}, Y) \mid X, Y \subseteq \mathcal{A}\}$

*Proof*

The proof follows the lines of the explanation of the main text, using the Splitting Set Theorem to split the program into the choice rule over $\lambda$ and the rest of the program. □

*Proposition 7*

Let $(\Pi, I, F)$ be a temporal logic problem. There is a one-to-one correspondence between the solutions to $(\Pi, I, F)$ and the $\lambda$-normal solutions to $(\lambda(\Pi), I, F)$.

*Proof*

For any solution S of $(\Pi, I, F)$, by Theorem 4, there is a path $P = (X_0, X_1, \dots, X_n)$ in $G(\Pi) = (V, E)$ where $I \subset X_0$ and $F \subset X_n$. By Proposition 6, if $(X, Y) \in E$ then $(X, Y \cup \{\lambda\}) \in E^\lambda$ and $(X \cup \{\lambda\}, Y \cup \{\lambda\}) \in E^\lambda$ where $G(\lambda(\Pi)) = (V^\lambda, E^\lambda)$. So, we can transform $P$ to a unique $P^\lambda = (X_0, X_1 \cup \{\lambda\}, \dots, X_n \cup \{\lambda\})$ which is a path of $G(\lambda(\Pi))$. By Theorem 4, $P^\lambda$ is a solution of $(\lambda(\Pi), I, F)$. Additionally, since $\lambda$ is true on all states but the first one, it is also a $\lambda$-normal solution.

On the other hand, for a $\lambda$-normal solution $S^\lambda$ of $(\lambda(\Pi), I, F)$, we have a path $P^\lambda = (Y_0, Y_1 \cup \{\lambda\}, \dots, Y_n \cup \{\lambda\})$ (Theorem 4). Since a state $W \cup \{\lambda\} \in V^\lambda$ must satisfy the rules of $\Pi$ then $W \in V$. Hence, $Y_1, \dots, Y_n \in V$ and $(Y_{m-1}, Y_m) \in E$ for $2 \leq m \leq n$. Since the pairs of the form $(X, Y \cup \{\lambda\})$ only exist for $(X, Y) \in E$, then $(Y_0, Y_1) \in E$. Therefore, by Theorem 4 $P = (Y_0, Y_1, \dots, Y_n)$ is a path in $G(\Pi)$. Since $I \in Y_0$ and $F \in Y_n$, then there is a solution that corresponds to this path and is a solution $(\Pi, I, F)$. □

*Proposition 8*

For any temporal program $\Pi$, the program $\lambda(\Pi)$ is internal wrt $\{\mathbf{F}\lambda_0\}$.

*Proof*

Recall the condition for a temporal logic problem to be internal: 1) Every initial state wrt $I$ is loop-reachable, and 2) Every reachable state wrt $I$ is internal.

In this case $I = \{\mathbf{F}\lambda_0\}$. The first condition is satisfied since the initial state has no $\lambda$. By Proposition 6, a state without lambda is connected to itself, meaning that it is a loop.

For the second condition we consider that all possible states are connected to some next state without $\lambda$ (Proposition 6), hence all reachable states have outgoing edges. Additionally, every state with or without $\lambda$ has some previous state connected to it (Proposition 6).

We can then say that the program $\lambda(\Pi)$ is internal wrt $\{\mathbf{F}\lambda_0\}$. □

*Theorem 7*

Let $\Pi$ be a temporal logic program, and $\delta$ be a resolvent of $\Psi_{\lambda(\Pi)}[i, j]$ for some $i$ and $j$ such that $1 \leq i \leq j$. For any $n \geq 1$, the set of nogoods $\Psi_{\lambda(\Pi)}[1, n] \cup \big\{\{\mathbf{T}\lambda_0\}\big\}$ entails the generalization

$$\{\delta\langle t\rangle \mid \mathit{step}(\delta\langle t\rangle) \subseteq [0, n]\}.$$

*Proof*
Proposition 8 tells us that $\lambda(\Pi)$ is internal. We can also think of $\{\mathbf{F}\lambda_0\}$ as a partial assignment $I$. Hence we can directly apply Theorem 6 to get that $\Psi_\lambda(\Pi)[1,n] \cup \{\{\mathbf{T}\lambda_0\}\} \models \delta$. $\square$

*Theorem 8*
Let $\Pi$ be a temporal logic program, and let $\delta$ be a resolvent of $\Psi_{\lambda(\Pi)}[i,j]$ for some $i$ and $j$ such that $1 \leq i \leq j$. For any $n \geq 1$, the set of nogoods $\Psi_\Pi[1,n]$ entails the generalization

$$\{simp(\delta\langle t\rangle) \mid step(\delta\langle t\rangle) \subseteq [0,n], \mathbf{T}\lambda_0 \notin \delta\langle t\rangle\}.$$

*Proof*
For any $\delta$ and $t$ such that $step(\delta\langle t\rangle) \subseteq [0,n], \lambda_0 \notin \delta\langle t\rangle$ it holds that $\Psi_\lambda(\Pi)[1,n] \cup \{\{\mathbf{T}\lambda_0\}\} \models \delta\langle t\rangle$ (Theorem 7).

This means that every path of length $k = j - i$ in $G(\lambda(\Pi))$ does not violate $\delta\langle t\rangle$ where $\lambda_0 \notin \delta\langle t\rangle$. Since every path longer than $k$ has a subpath of length $k$, then every path longer than $k$ must also not violate $\delta\langle t\rangle$. Therefore, for $n \geq k$, every path $(X_0,\ldots,X_n)$ in $G(\lambda(\Pi))$ must not violate $\delta\langle t\rangle$ where $\lambda_0 \notin \delta\langle t\rangle$.

Now we will prove the theorem by contradiction. Take any path $(X_0,\ldots,X_n)$ in $G(\Pi)$ and assume that it violates $simp(\delta\langle t\rangle)$. By Proposition 6 we have a path $(X_0, X_1 \cup \{\lambda\} \ldots, X_n \cup \{\lambda\})$ that will violate $simp(\delta\langle t\rangle) \cup \{\lambda_1,\ldots,\lambda_n\}$. Since $\delta\langle t\rangle \subset simp(\delta\langle t\rangle) \cup \{\lambda_1,\ldots,\lambda_n\}$ the path also violates $\delta\langle t\rangle$. This is a contradiction. Hence, every path of length $n$ of $G(\Pi)$ does not violate $simp(\delta\langle t\rangle)$. Therefore $\Psi_\Pi[1,n] \models \{simp(\delta\langle t\rangle) \mid step(\delta\langle t\rangle) \subseteq [0,n], \mathbf{T}\lambda_0 \notin \delta\langle t\rangle\}$. $\square$

*Proposition 9*
Let $\mathcal{T}_1 = (\Pi, I, F)$ and let $\mathcal{T}_2 = (tr^\lambda(\Pi), I, F)$ be temporal logic problems. There is a one-to-one correspondence between the solutions to $\mathcal{T}_1$ and the $\lambda$-normal solutions to $\mathcal{T}_2$.

*Proof*
For any model $I$ of $\mathcal{T}_1$ of lengh $n$ there is also a model $I^\lambda = I \cup \{\lambda_1,...,\lambda_n\}$ of $\mathcal{T}_2$. Since $tr^\lambda$ only adds a $\lambda$ to the dynamic constraints the only difference in the nogoods of $\mathcal{T}_1$ and $\mathcal{T}_2$ is that $tr^\lambda(\Pi)^{di}$ have an additional $\lambda$. Hence, no nogoods of $tr^\lambda(\Pi) \setminus tr^\lambda(\Pi)^{di}$ is satisfied by $I^\lambda$. Additionally, since in $\lambda$-normal solutions all $\lambda$ are true, the nogoods of $\Pi^{di}$ can be simplified by deleting their $\lambda$. The simplified nogoods of $tr^\lambda(\Pi)^{di}$ are the same as the nogoods of $\Pi^{di}$. This means that $I^\lambda$ does not satisfy any nogood in $tr^\lambda(\Pi)^{di}$. We can then conclude that $I^\lambda$ does not satisfy any nogood of $tr^\lambda(\Pi)$ and is thus a model of $\mathcal{T}_2$.

For any model $I^\lambda$ of $\mathcal{T}_2$ of lengh $n$ there is also a model $I = I^\lambda \setminus \{\lambda_1,...,\lambda_n\}$ of $\mathcal{T}_1$. Since $tr^\lambda$ only adds a $\lambda$ to the dynamic constraints the only difference in the nogoods of $\mathcal{T}_1$ and $\mathcal{T}_2$ is that $tr^\lambda(\Pi)^{di}$ have an additional $\lambda$. Hence, all nogoods of $\Pi \setminus \Pi^{di}$ are not satisfied by $I$. Since in $\lambda$-normal solutions all $\lambda$ are true, the nogoods of $tr^\lambda(\Pi)^{di}$ act the same way as the nogoods of $\Pi^{di}$. Hence, the nogoods of $\Pi^{di}$ are also not satisfied by $I$. We can then conclude that $I$ does not satisfy any nogood of $\Pi$ and is thus a model of $\mathcal{T}_1$.

It follows that for every stable model of $(\Pi, I, F)$ there is a corresponding stable model of $(tr^\lambda(\Pi), I, F)$ and vice versa. $\square$

*Lemma 3*
An interval $[k,l]$ is an *overaproximation* of the interval $[i,j]$ if $k \leq i$ and $j \leq l$ holds. For some resolvent $\delta$ of $\Psi^\lambda_{tr}(\Pi)[i,j]$, $step^\lambda(\delta)$ computes an *overaproximation* of the interval $[i,j]$.

*Proof*

- **case 1**: $\delta$ is a resolvent of $\Psi^{\lambda}_{tr}(\Pi)[i,i]$. By definition, $step^{\lambda}(\delta) = [i,i]$ since there would be no $\lambda$ in $\delta$ and the only timestep in the atoms of $\delta$ would be $i$.
- **case 2**: $\delta$ is a resolvent of $\Psi^{\lambda}_{tr}(\Pi)[i,j]$ with $i < j$ and $\lambda[i,j] \in \delta$. By definition, $step^{\lambda}(\delta) = [i-1,j]$ since the lowest timepoint in any $\lambda$ is $i$.
- **case 3**: $\delta$ is a resolvent of $\Psi^{\lambda}_{tr}(\Pi)[i,j]$ with $i < j$ and $\lambda[i+1,j] \in \delta$. By definition, $step^{\lambda}(\delta) = [i,j]$ since the lowest timepoint in any $\lambda$ is $i+1$.

We can clearly see that for any resolvent $\delta$ the function $step^{\lambda}(\delta)$ computes the exact (cases 1 and 3) or a bigger (case 2) interval. Hence, it is an overapproximation of the interval. □

*Theorem 9*
Let $\Pi$ be a temporal logic program in PNF, and $\delta$ be a resolvent of $\Psi_{tr^{\lambda}(\Pi)}[1,m]$ for some $m \geq 1$. Then, for every $n \geq 1$, the set of nogoods $\Psi_{tr^{\lambda}(\Pi)}[1,n]$ entails the generalization

$$\{\delta\langle t\rangle \mid step^{\lambda}(\delta\langle t\rangle) \subseteq [1,n]\}.$$

*Proof*
If $\delta$ is a resolvent of $\Psi_{tr^{\lambda}(\Pi)}[1,m]$ and $step^{\lambda}(\delta) = [i,j]$ where $0 \leq i \leq j \leq m$ then $\delta$ is a resolvent of $\Psi_{tr^{\lambda}(\Pi)}[i,j]$ (by Lemma 3). For any $t$, $step^{\lambda}(\delta\langle t\rangle) = [i+t, j+t]$ which means $step^{\lambda}(\delta\langle t\rangle)$ is a resolvent of $\Psi_{tr^{\lambda}(\Pi)}[i+t,j+t]$ (by Lemma 2). Consequently, for any $t$ where $[i+t,j+t] \subseteq [1,n]$ then $\delta\langle t\rangle$ is entailed by $\Psi_{tr^{\lambda}(\Pi)}[1,n]$ due to Theorem 3. □

*Proposition 10*
For any temporal logic program $\Pi$, the program $tr^{*}(\Pi)$ is in PNF.

*Proof*
Let $\mathcal{A}$ be a set of atoms ocurring in a logic program $\Pi$ and $\mathcal{A}'$ be the set of atoms that reference the past. Recall that a logic program $\Pi$ is in PNF if for any rule $r \in \Pi^n \cup \Pi^c$ it holds that $Bd(r) \cap \mathcal{A}' = \emptyset$

For any rule $r \in \Pi^n \cup \Pi^c$ where $Bd(r) \cap \mathcal{A}' \neq \emptyset$ it holds that $Bd(r^{*}) \cap \mathcal{A}' = \emptyset$ since any occurrence is substituted by the corresponding $p^{*}$ atom. For any rule $r \in \Pi^n \cup \Pi^c$ where $Bd(r) \cap \mathcal{A}' = \emptyset$ it holds that $Bd(r^{*}) \cap \mathcal{A}' = \emptyset$ since the translation does not change the rule. Hence, for any rule $r \in \Pi^n \cup \Pi^c$ it holds that $Bd(r^{*}) \cap \mathcal{A}' = \emptyset$. Which means that $tr^{*}(\Pi)$ is in PNF. □

*Lemma 4*
For any program $\Pi$ the truth value of $p^{*}$ and $p'$ always conincide in the solutions of $tr^{*}(\Pi)$ where $a^{*} \in \mathcal{A}^{*}$ are the atoms introduced by the $tr^{*}$translation and $a' \in \mathcal{A}'$ are the atoms occuring in $\Pi$ referencing the past.

*Proof*
We label the rules added by the translation $tr^{*}$as follows:

$$\{a^{*}\} \leftarrow \tag{1}$$

$$\bot \leftarrow a', \neg a^{*} \tag{2}$$

$$\bot \leftarrow \neg a', a^{*} \tag{3}$$

- if $a'$ is True then $a^*$ must also be True to not violate rule 2
- if $a'$ is False then $a^*$ must also be False to not violate rule 3
- if $a^*$ is True then $a'$ must also be True to not violate rule 3
- if $a^*$ is False then $a'$ must also be False to not violate rule 2

We can then conclude that the truth value of $a^*$ and $a'$ always conincide in the resulting program $tr^*(\Pi)$. □

*Proposition 11*
Let $\mathcal{T}_1 = (\Pi, I, F)$ and let $\mathcal{T}_2 = (tr^*(\Pi), I, F)$ be temporal logic problems. There is a one-to-one correspondence between the solutions to $\mathcal{T}_1$ and the solutions to $\mathcal{T}_2$ that do not contain any atom $p^* \in \mathcal{P}^*$ at step 0.

*Proof*
We label the rules added by the translation $tr^*$as follows:

$$\{a^*\} \leftarrow \tag{4}$$

$$\bot \leftarrow a', \neg a^* \tag{5}$$

$$\bot \leftarrow \neg a', a^* \tag{6}$$

Let $a^* \in \mathcal{A}^*$ be the atoms added by the $tr^*$ translation and $a' \in \mathcal{A}'$ be the set of atom occuring in $\Pi$ that reference the past.

**Case 1:** Let $\mathcal{A}' \cap At(\Pi) = \emptyset$. Since $tr^*(\Pi) = \Pi$ then $\mathcal{T}_1 = \mathcal{T}_2$ and they have the same solutions.

**Case 2:** Let $\mathcal{A}' \cap At(\Pi) \neq \emptyset$. For any solution $\mathcal{S}_1$ of $\mathcal{T}_1$ there is a solution $\mathcal{S}_2$ of $\mathcal{T}_2$ where $\mathcal{S}_2 = \mathcal{S}_1 \cup \{a^* | a' \in \mathcal{S}_1\}$. Since $a'$ and $a^*$ always have the same truth value (Lemma 4), the evaluation of the nogoods induced by $tr^*(\Pi)$ where $a'$ was substituted by $a^*$ will stay the same regardless of the assignment. Also, none of the nogoods induced by the extra rules (5) and (6) will be satisfied since $a'$ and $\neg a^*$ always have different truth values. We can also ignore rule (4) since it does not induce any nogoods. Hence, $\mathcal{S}_2$ is a stable model of $tr^*(\Pi)$. Finally, given that $\mathcal{S}_1$ is consistent with $I$ and $F$ then $\mathcal{S}_2$ is also consistent with $I$ and $F$. Consequently, it is also a solution to $\mathcal{T}_2$.

On the other hand, For any solution $\mathcal{S}_2$ of $\mathcal{T}_2$ there is a solution $\mathcal{S}_1$ of $\mathcal{T}_1$ where $\mathcal{S}_1 = \mathcal{S}_2 \setminus \{a^* | a' \in \mathcal{S}_2\}$. Since $a'$ and $a^*$ always have the same truth value (Lemma 4), the evaluation of the nogoods induced by $\Pi$ will stay the same regardless of the assignment. Hence, $\mathcal{S}_2$ is a stable model of $\Pi$. Finally, given that $\mathcal{S}_2$ is consistent with $I$ and $F$, then $\mathcal{S}_1$ is also consistent with $I$ and $F$. Consequently, it is also a solution to $\mathcal{T}_1$. □

*Lemma 5*
For a resolvent $\delta$ of $\Psi_{tr^\lambda(tr^*(\Pi))}[1,m]$ it holds that $simp(\delta)$ is entailed by $\Psi_\Pi[1,m]$ for $\lambda$-normal solutions of $\Psi_{tr^\lambda(tr^*(\Pi))}[1,m]$.

*Proof*
Let $\delta^\lambda$ be a resolvent of $\Psi_{tr^\lambda(tr^*(\Pi))}[1,m]$. In $\lambda$-normal solutions the $\lambda$ atoms in the nogoods are always true and thus have no effect in their satisfaction. Hence, the nogood $\delta^* = \delta^\lambda \setminus \lambda[1,m]$ is entailed by $\Psi_{tr^*(\Pi)}[1,m]$ since the nogoods in $\Psi_{tr^*(\Pi)}[1,m]$ are the nogoods in $\Psi_{tr^\lambda(tr^*(\Pi))}[1,m]$ without $\lambda$s.

Let $\delta^*$ be a resolvent of $\Psi_{tr^*(\Pi)}[1,m]$ with its corresponding resolution proof $\mathcal{T}$. Let $C$ be the constraints added by the $tr^*$ translation. Observe that a nogood of $tr^*(\Pi) \setminus C$ can be transformed into a nogood of $\Pi$ simply by substituting all $a^*[i] \in \delta^*$ by their corresponding atom $a[i-1]$ from $\mathcal{A}$ where $a^* \in \mathcal{A}^*$ are the atoms introduced by the translation and $\mathcal{A}$ is the set of atoms ocurring in $\Pi$.

For any nogood in $\mathcal{T}$ containing atoms $a^*[i]$, we can substitute them by the corresponding atom $a[i-1]$ without changing the semantics of the nogoods since they always have the same truth value (by Lemma 4).

Next, recall that the constraints added by the $tr^*$ translation have the form

$$\bot \leftarrow a[i-1], \neg a^*[i]$$
$$or$$
$$\bot \leftarrow \neg a[i-1], a^*[i]$$

for some integer $i$. If we substitute $a^*$ by the corresponding atom we get the constraints

$$\bot \leftarrow a[i-1], \neg a[i-1]$$
$$or$$
$$\bot \leftarrow \neg a[i-1], a[i-1]$$

It is easy to see that any nogood that resolves with the nogoods induced by these constraints would result in the same nogood. Hence, we can remove the nogoods induced by $C$ from $\mathcal{T}$ without affecting its result. Note that the choice rules introduced by the translatation do not induce nogoods. This means that any nogood left in $\mathcal{T}$ is either in or entailed by $\Psi_\Pi[1,m]$. Hence, the result $\delta$ of the resolution proof $\mathcal{T}$ is entailed by $\Psi_\Pi[1,m]$.

It is clear that $\delta = \{\mathbf{V}a_i | \mathbf{V}a_i \in \delta^\lambda, a \in \mathcal{A}\} \cup \{\mathbf{V}a_{i-1} | \mathbf{V}a^*_i \in \delta^\lambda, a^* \in \mathcal{A}^*\}$. In words, $\delta$ is the result of substitung any atom in $\mathcal{A}^*$ with the corresponding atom in $\mathcal{A}$ and ignoring any $\lambda$ atoms. Hence, $\delta = simp(\delta^\lambda)$. Consequently, $simp(\delta^\lambda)$ is a resolvent of $\Psi_\Pi[1,m]$. $\square$

*Theorem 10*
Let $\Pi$ be a temporal logic program, and $\delta$ be a resolvent of $\Psi_{tr^\lambda(tr^*(\Pi))}[1,m]$ for some $m \geq 1$. Then, for every $n \geq 1$, the set of nogoods $\Psi_\Pi[1,n]$ entails the generalization

$$\{simp(\delta\langle t\rangle) \mid step^\lambda(\delta\langle t\rangle) \subseteq [1,n]\}.$$

*Proof*
Since $\delta$ is entailed by $\Psi_{tr^\lambda(tr^*(\Pi))}[1,m]$ then $simp(\delta)$ is also entailed by $\Psi_\Pi[1,m]$ (Lemma 5). By Theorem 9 $simp(\delta)\langle t\rangle$ is entailed by $\Psi_\Pi[1,n]$ for any $t$ where $step^\lambda(\delta\langle t\rangle) \subseteq [1,n]$. $\square$

## 4 Additional results

The following tables show the results of the experiments using the translations from our conference paper (2). The experiments of our conference paper (**?**) have a bug in the multi-shot case. Here, that bug is fixed and the learning approach is no longer worse than the baseline, but it is still not better.

| | | **baseline** | **500** | **1000** | **1500** |
|---|---|---|---|---|---|
| **blocks** | (300) | 0.5 (0) | **0.1** (0) | **0.1** (0) | **0.1** (0) |
| **depots** | (270) | 146.4 (30) | 138.2 (29) | **126.0** (25) | 128.3 (30) |
| **driverlog** | (135) | 14.1 (1) | 12.5 (1) | 12.3 (1) | **10.7** (1) |
| **elevator** | (300) | **3.0** (0) | 3.7 (0) | 3.8 (0) | 4.3 (0) |
| **grid** | (30) | 11.4 (0) | **5.2** (0) | 5.3 (0) | **5.2** (0) |
| **gripper** | (255) | 381.0 (96) | 368.5 (91) | **359.0** (90) | 370.8 (88) |
| **logistics** | (225) | **0.5** (0) | 0.9 (0) | 0.9 (0) | 0.9 (0) |
| **mystery** | (126) | 57.0 (3) | 58.5 (3) | 50.9 (3) | **46.8** (2) |
| **Total** | (1663) | 89.7 (130) | 86.4 (124) | **82.4** (119) | 84.2 (121) |

Table 1. Single shot solving of PDDL benchmarks using translations.

| | | **baseline** | **500** | **1000** | **1500** |
|---|---|---|---|---|---|
| **HanoiTower** | (20) | 160.6 (2) | **97.7** (0) | 101.0 (0) | 118.2 (1) |
| **Labyrinth** | (20) | **247.3** (3) | 355.7 (4) | 355.7 (4) | 356.1 (4) |
| **Nomistery** | (20) | 585.3 (12) | 575.6 (12) | 556.2 (12) | **502.0** (10) |
| **Ricochet Robots** | (20) | 465.3 (9) | **464.7** (9) | 464.8 (8) | **464.7** (8) |
| **Sokoban** | (20) | 458.8 (9) | **441.5** (9) | 458.8 (8) | 453.0 (8) |
| **Visit-all** | (20) | 559.0 (12) | 556.5 (12) | 560.8 (12) | **556.4** (12) |
| **Total** | (120) | 412.7 (47) | 415.3 (46) | 416.2 (44) | **408.4** (43) |

Table 2. Single shot solving of ASP benchmarks using translations.

| | | **baseline** | **500** | **1000** | **1500** |
|---|---|---|---|---|---|
| **blocks** | (20) | 1.3 (0) | **0.7** (0) | **0.7** (0) | **0.7** (0) |
| **depots** | (18) | **148.6** (2) | 257.0 (3) | 189.4 (3) | 221.7 (3) |
| **driverlog** | (9) | 108.9 (1) | **102.0** (1) | 104.9 (1) | 108.5 (1) |
| **elevator** | (20) | **280.3** (5) | 285.7 (5) | 295.0 (5) | 305.4 (5) |
| **freecell** | (16) | 900.0 (16) | 900.0 (16) | 900.0 (16) | 900.0 (16) |
| **grid** | (2) | 5.2 (0) | **4.1** (0) | 4.2 (0) | 4.3 (0) |
| **gripper** | (17) | 848.6 (16) | **847.5** (16) | 849.1 (16) | 847.9 (16) |
| **logistics** | (20) | **225.2** (5) | 225.3 (5) | 225.3 (5) | 225.3 (5) |
| **mystery** | (14) | **321.8** (5) | **321.8** (5) | 321.9 (5) | 321.9 (5) |
| **Total** | (136) | **346.6** (50) | 361.0 (51) | 353.8 (51) | 359.7 (51) |

Table 3. Multi shot solving of PDDL benchmarks using translations.

| | | baseline | 500 | 1000 | 1500 |
|---|---|---|---|---|---|
| **HanoiTower** | (20) | **554.1** (10) | 601.4 (11) | 593.7 (10) | 646.7 (11) |
| **Labyrinth** | (20) | **647.7** (14) | 647.8 (14) | 647.8 (14) | 647.9 (14) |
| **Nomistery** | (20) | **64.2** (1) | 77.0 (1) | 81.0 (1) | 69.3 (1) |
| **Ricochet Robots** | (20) | 527.3 (11) | **518.1** (11) | 519.3 (11) | 521.3 (11) |
| **Sokoban** | (20) | **721.5** (16) | 722.6 (16) | 722.3 (16) | 722.0 (16) |
| **Visit-all** | (20) | **677.5** (13) | 704.0 (13) | 774.6 (15) | 801.6 (16) |
| **Total** | (120) | **532.1** (65) | 545.2 (66) | 556.5 (67) | 568.1 (69) |

Table 4. Multi shot solving of ASP benchmarks using translations.