A Supplementary Material

A.1 Remarks to preliminaries

Df. 1 implicitly defines destructors *cond*, *sync*, *act* mapping a label to its counterpart.

For $\alpha \in \mathit{Act}$, $V \subseteq \mathit{Var}$, $c \in \mathit{dom}(V)$ the substitution $\alpha[V = c]$ cannot generally be performed in a straightforward manner, i.e. via replacement of all free occurrences of v by c in atomic assignments $\alpha^{(1)}\alpha^{(2)} \dots \alpha^{(m)}$ associated with α. This is because updates of α are executed sequentially and might be internally dependent (i.e., $\mathit{lhs}(\alpha^{(i)})$ appearing in $\mathit{rhs}(\alpha^{(j)})$ for some $i > j$).

In Df. 3 effect function $\mathit{Effect}^i : \mathit{Eval}(\mathit{Var}) \times \mathit{Act}^i \mapsto \mathit{Eval}(\mathit{Var})$ is a natural extension of an original one from agent graph G^i, s.t. $\mathit{Var}^i \subseteq \mathit{Var}$ and $\big(\mathit{Effect}^i(\alpha, \eta) = \eta'\big) \Rightarrow \big(\forall v \notin \mathit{Var}^i \,.\, \eta =_v \eta'\big)$.

A.2 General Variant of the Abstraction

The general variant of $\mathcal{A}^*_F$, where $* \in \{\mathit{may}, \mathit{must}\}$, $F = \{(f_1, \mathit{Sc}_1), \dots, (f_m, \mathit{Sc}_m)\}$ is described in Alg. 4. For each pair (f_i, Sc_i) of mapping and its scope, it transforms the edges in $G = G_j$ in the following way:

- edges entering or inner the Sc_i have their actions appended with (1) update of the target variable z_i and (2) update which sets the values of the source variables X_i to their defaults (resetting those),
- edges leaving or within the Sc_i an have actions prepended with (1) update of source variables X_i (a temporarily one to be assumed for original action) and (2) update which resets the values of the target variable z_i.

Note that due to the introduction of a scope, the variables from $\mathit{Args}_R(F)$ are not genuinely removed - those must be accessible for query and present in AP - removal is mimicked by not having edges that would change their evaluation within states having location label coming from the scope.

A.3 Proof of Theorem 4

In general case, the construction of $\mathcal{A}^{\mathit{may}}_F(\mathit{MG})$ implies that $\mathit{Var}_2 = \mathit{Var}_1 \cup Z$ and $\hat{g_0} = (g_0[V{=}\eta_0(V)]) \wedge g_Z$, where $g_Z \cong \bigwedge_{z \in Z} z{=}\hat{\eta_0}(z)$. Therefore for $\langle l, \eta_1\rangle \in I_1$ it holds that $\eta_2 \in \mathit{Sat}(g_Z) \Rightarrow \eta_2[\overline{V}{=}\eta_1(\overline{V})] \models \hat{g_0}$ and $\langle l, \eta_2[\overline{V}{=}\eta_1(\overline{V})]\rangle \in I_2$, meaning Df. 7(i) holds.

Now we show that Df. 7(ii) holds as well. By construction, each concrete $(l, (g, ch, \alpha), l') \in \hookrightarrow$ from *MG* will have (at least one) matching abstract edge $(l, (\hat{g}_\eta, ch, \hat{\alpha}_\eta), l') \in \hat{\hookrightarrow}$, where: for $J = \{i \mid l \in \mathit{Sc}_i\}$ and $K = \{i \mid l' \in \mathit{Sc}_i\}$ $\hat{g}_\eta = g[X'{=}c]$, $\hat{\alpha}_\eta = \beta \circ \alpha \circ \gamma$, where $X' = \bigcup_{i \in J} \mathit{Args}_R(f_i)$, $X'' = \bigcup_{i \in K} \mathit{Args}_R(f_i)$, $Z' = \bigcup_{i \in J} \mathit{Args}_N(f_i)$, $Z'' = \bigcup_{i \in K} \mathit{Args}_N(f_i)$, $\gamma \equiv (X'{=}\eta(X'))(Z'{=}\eta_0(Z'))$, $\beta \equiv (X''{=}\eta_0(X''))(Z''{=}F_K(\eta(X'')))$ for some $\eta \in d^+(l)$.

Therefore, for any $\langle l, \eta_1\rangle \mathcal{R} \langle l, \eta_2\rangle$ and $\langle l, \eta_1\rangle \longrightarrow_1 \langle l', \eta_1'\rangle$ that was induced by an edge $(l, (g, ch, \alpha), l') \in \hookrightarrow$, where $\eta_1 \models g$ and $\mathit{Effect}(\mathit{act}(\mathit{labl}_1), \eta_1) = \eta_1'$, there must exist an edge $(l, (\hat{g}_\eta, ch, \hat{\alpha}_\eta), l') \in \hat{\hookrightarrow}$ with $\eta \in d^+(l)$ s.t. $\eta =_{(\overline{V})} \eta_1$, that induces $\langle l, \eta_2\rangle \longrightarrow_2 \langle l', \eta_2'\rangle$, and by Lem. 2 and Lem. 3 $\eta_2' =_{\overline{V}} \eta_1'$ and concludes $\langle l', \eta_1'\rangle \mathcal{R} \langle l', \eta_2'\rangle$.

Algorithm 4: General abstraction

```
1  ComputeAbstraction(G, d, F)
2      X := Args_R(F)
3      Z := Args_N(F)
4      Sc := ⋃_{(f_i,Sc_i)∈F} Sc_i
5      g_0 := g_0 ∧ (Z=f(η_0(X)))
6      ↪_a := ∅
7      foreach l --g:ch α--> l' do
8          F_1 := {f_i | (f_i, Sc_i) ∈ F ∧ l ∈ Sc_i}
9          F_2 := {f_i | (f_i, Sc_i) ∈ F ∧ l' ∈ Sc_i}
10         if {l, l'} ∩ Sc = ∅ then
11             ↪_a := ↪_a ∪{l --g:ch α--> l'}
12         else
13             foreach η ∈ d(l) do
14                 W_1 := ⋃_{f∈F_1} Args_R(f)
15                 W_2 := ⋃_{f∈F_1} Args_N(f)
16                 Y_1 := ⋃_{f∈F_2} Args_R(f)
17                 Y_2 := ⋃_{f∈F_2} Args_N(f)
18                 g' := g[W_1 = η(W_1)]
19                 α' := (W_2 := η_0(W_2)).α
20                 α' := (W_1 := η(W_1)).α'
21                 α' := α'.(Y_2 := F_2(η(W_2)))
22                 α' := α'.(Y_1 := F_1(η_0(W_1)))
23                 ↪_a := ↪_a ∪{l --g':ch α'--> l'}
24     ↪ := ↪_a
25     Var := Var ∪ Z
26     return G
```

A.4 Proof of Theorem 7

The construction of $\mathcal{A}^{\mathit{must}}_F(\mathit{MG})$ implies that $\mathit{Var}_2 = \mathit{Var}_1 \cup Z$ and $\hat{g_0} = (g_0[V = \eta_0(V)]) \wedge g_Z$, where $g_Z \cong \bigwedge_{z \in Z} z = \hat{\eta_0}(z)$. Therefore for $\langle l, \eta_2\rangle \in I_2$ it holds that for any $\eta_1 \in \mathit{Eval}(\mathit{Var}_1)$ $\eta_1[\overline{V} = \eta_2(\overline{V})] \models g_0$ and $\langle l, \eta_1[\overline{V} = \eta_1(\overline{V})]\rangle \in I_1$, which means that Df. 7(i) holds.

As the abstraction is constructed in the same way, the reasoning about requirement Df. 7(ii) as above can be applied, except that by Lem. 6 each abstract edge $(l, (\hat{g}_\eta, ch, \hat{\alpha}_\eta), l') \in \hat{\hookrightarrow}$ is now matched by exactly one $(l, (g, ch, \alpha), l') \in \hookrightarrow$ from *MG*.

A.5 Experiments for Must-Abstraction

In the experiments for under-approximation, we used

$$\varphi_{dispatch} \equiv \mathsf{AG}\ \mathsf{coll\_vts}\ \mathit{imply}\ \big(\textstyle\sum_{\mathsf{j}=1}^{NV} \mathsf{pack\_sent[j]} = NV\big)$$

expressing that the election packages must be eventually dispatched to all the voters.[9] The results of the experiments are shown in Tab. 3. In all the completed cases, the verification of the abstract model was conclusive (i.e., the output was "false" for all the instances presented in Tab. 3).

In case when a AG formula is not satisfied by the model, its verification is in fact equivalent to finding a witness for a EF formula, which is often easy in practice. And it becomes even more so if model checker utilizes on-the-fly techniques (as is the case of UPPAAL) and examines the model simultaneously with generation of states. This is confirmed by experimental results for model checking of $\varphi_{dispatch}$. However, it is worth noting that despite the lack of notable gains

[9] The aforementioned formula is equivalent to $\mathsf{AF}(\sum_{\mathsf{j}=1}^{NV} \mathsf{pack\_sent[j]} = NV)$ in our model; the former variant is used due to UPPAAL's non-standard interpretation of the AF.

conf	Concrete		Abstract 1			Abstract 2			Abstract 3		
NV,NC	#St	tv (sec)	ta (sec)	#St	tv (sec)	ta (sec)	#St	tv (sec)	ta (sec)	#St	tv (sec)
1,1	23	0	0.03	15	0	0.07	14	0	0.08	14	0
1,2	27	0	0.03	15	0	0.05	14	0	0.06	14	0.01
1,3	31	0	0.03	15	0	0.05	14	0	0.04	14	0
2,1	241	0	0.01	81	0	0.04	70	0	0.04	70	0
2,2	369	0	0.01	81	0	0.02	70	0	0.03	70	0
2,3	529	0	0.03	81	0	0.02	70	0	0.04	70	0
3,1	2987	0	0.01	459	0	0.03	368	0	0.03	368	0
3,2	6075	0	0.02	459	0	0.03	368	0	0.03	368	0
3,3	1.09e+4	0	0.02	459	0	0.03	368	0	0.03	368	0
4,1	3.98e+4	0	0.01	2673	0	0.03	2002	0.01	0.04	2002	0
4,2	1.06e+5	0	0.01	2673	0	0.05	2002	0	0.03	2002	0
4,3	2.36e+5	0	0.01	2673	0	0.04	2002	0	0.03	2002	0
5,1	5.46e+5	0	0.01	1.58e+4	0	0.04	1.11e+4	0	0.06	1.11e+4	0
5,2	1.90e+6	0	0.01	1.58e+4	0	0.06	1.11e+4	0	0.05	1.11e+4	0
5,3	5.16e+6	0	0.02	1.58e+4	0	0.07	1.11e+4	0	0.05	1.11e+4	0.01
6,1	7.58e+6	0	0.01	9.40e+4	0	0.15	6.30e+4	0	0.09	6.30e+4	0
6,2	3.41e+7	0.01	0.01	9.40e+4	0	0.14	6.30e+4	0	0.10	6.30e+4	0
6,3	1.13e+8	0	0.01	9.40e+4	0	0.09	6.30e+4	0	0.09	6.30e+4	0
7,1	1.06e+8	0	0.01	5.62e+5	0.01	0.28	3.60e+5	0	0.24	3.60e+5	0
7,2	$\gg$1e+8	0	0.01	5.62e+5	0	0.34	3.60e+5	0	0.21	3.60e+5	0
7,3	$\gg$1e+8	0.01	0.01	5.62e+5	0	0.35	3.60e+5	0	0.23	3.60e+5	0
8,1	$\gg$1e+8	0	0.01	3.37e+6	0	0.90	2.08e+6	0	0.69	2.08e+6	0
8,2	$\gg$1e+8	0	0.02	3.37e+6	0	1.03	2.08e+6	0	0.63	2.08e+6	0
8,3	$\gg$1e+8	0	0.01	3.37e+6	0	0.86	2.08e+6	0	0.55	2.08e+6	0
9,1	$\gg$1e+8	0	0.01	2.02e+7	0	4.41	1.21e+7	0	2.43	1.21e+7	0
9,2	$\gg$1e+8	0	0.01	2.02e+7	0	2.80	1.21e+7	0	2.03	1.21e+7	0
9,3	$\gg$1e+8	0	0.01	2.02e+7	0	2.69	1.21e+7	0	1.99	1.21e+7	0
10,1	$\gg$1e+8	0	0.01	1.21e+8	0	9.61	7.03e+7	0	7.49	7.03e+7	0.01
10,2	$\gg$1e+8	0	0.01	1.21e+8	0	7.83	7.03e+7	0	8.02	7.03e+7	0
10,3	$\gg$1e+8	0	0.01	1.21e+8	0	8.99	7.03e+7	0	7.71	7.03e+7	0

Table 3: Experimental results for model checking of $\varphi_{dispatch}$ in must-abstractions of postal voting

in terms of verification time from must-abstraction, the reduction in state space could be of immense importance when model is generated prior to its exploration.