Proof System Overview

In what follows we give a full overview of the components of the proof system. We first define the type of objects used, then list the inference rules used within the proof system and afterwards the basic statements which have to be verified based on semantics. To emphasize that the proof system interprets judgments on a purely syntactical level, we write $S \sqsubseteq S$ instead of $S \subseteq S$ when denoting a judgment in the proof system.

Types We define the object types through a simple grammar:

state set variables	$X := \{I^\Pi\} \mid S_G^\Pi \mid \emptyset \mid X_\mathbf{R}$
state set literals	$L := X \mid \overline{X}$
state set expressions	$S := L \mid (S \cup S) \mid (S \cap S) \mid S[A] \mid [A]S$
action set expressions	$A := A^\Pi \mid a \mid (A \cup A)$
set expressions	$E := S \mid A$

Set expressions are defined separately to enable us to define basic set theory rules that can be applied to both state set expressions and action set expressions. In what follows we denote an object of e.g. type E by simply E,E' or E'' instead of $Z : E$ and also do not mention the type of constant expressions since they are defined above.

Rules The following rules show that state sets are dead:

Empty set **D**ead
$$\frac{}{\emptyset \text{ dead}} \textbf{ ED}$$

Union **D**ead
$$\frac{S \text{ dead} \qquad S' \text{ dead}}{S \cup S' \text{ dead}} \textbf{ UD}$$

Subset **D**ead
$$\frac{S' \text{ dead} \qquad S \sqsubseteq S'}{S \text{ dead}} \textbf{ SD}$$

Progression **G**oal
$$\frac{S[A^\Pi] \sqsubseteq S \cup S' \qquad S' \text{ dead} \qquad S \cap S_G^\Pi \text{ dead}}{S \text{ dead}} \textbf{ PG}$$

Progression **I**nitial
$$\frac{S[A^\Pi] \sqsubseteq S \cup S' \qquad S' \text{ dead} \qquad \{I^\Pi\} \sqsubseteq S}{\overline{S} \text{ dead}} \textbf{ PI}$$

Regression **G**oal
$$\frac{[A^\Pi]S \sqsubseteq S \cup S' \qquad S' \text{ dead} \qquad \overline{S} \cap S_G^\Pi \text{ dead}}{\overline{S} \text{ dead}} \textbf{ RG}$$

Regression **I**nitial
$$\frac{[A^\Pi]S \sqsubseteq S \cup S' \qquad S' \text{ dead} \qquad \{I^\Pi\} \sqsubseteq \overline{S}}{S \text{ dead}} \textbf{ RI}$$

These rules show that the task is unsolvable:

Conclusion **I**nitial
$$\frac{\{I^\Pi\} \text{ dead}}{\text{unsolvable}} \textbf{ CI}$$

Conclusion **G**oal
$$\frac{S_G^\Pi \text{ dead}}{\text{unsolvable}} \textbf{ CG}$$

These rules from basic set theory can be used for both state and action set expressions:

Union **R**ight $\dfrac{}{E \sqsubseteq (E \cup E')}$ **UR**

Union **L**eft $\dfrac{}{E \sqsubseteq (E' \cup E)}$ **UL**

Intersection **R**ight $\dfrac{}{(E \cap E') \sqsubseteq E}$ **IR**

Intersection **L**eft $\dfrac{}{(E' \cap E) \sqsubseteq E}$ **IL**

DIstributivity $\dfrac{}{((E \cup E') \cap E'') \sqsubseteq ((E \cap E'') \cup (E' \cap E''))}$ **DI**

Subset **U**nion $\dfrac{E \sqsubseteq E'' \qquad E' \sqsubseteq E''}{(E \cup E') \sqsubseteq E''}$ **SU**

Subset **I**ntersection $\dfrac{E \sqsubseteq E' \qquad E \sqsubseteq E''}{E \sqsubseteq (E' \cap E'')}$ **SI**

Subset **T**ransitivity $\dfrac{E \sqsubseteq E' \qquad E' \sqsubseteq E''}{E \sqsubseteq E''}$ **ST**

The final rules focus on progression and its relation to regression:

Action **T**ransitivity $\dfrac{S[A] \sqsubseteq S' \qquad A' \sqsubseteq A}{S[A'] \sqsubseteq S'}$ **AT**

Action **U**nion $\dfrac{S[A] \sqsubseteq S' \qquad S[A'] \sqsubseteq S'}{S[A \cup A'] \sqsubseteq S'}$ **AU**

Progression **T**ransitivity $\dfrac{S[A] \sqsubseteq S'' \qquad S' \sqsubseteq S}{S'[A] \sqsubseteq S''}$ **PT**

Progression **U**nion $\dfrac{S[A] \sqsubseteq S'' \qquad S'[A] \sqsubseteq S''}{(S \cup S')[A] \sqsubseteq S''}$ **PU**

Progression to **R**egression $\dfrac{S[A] \sqsubseteq S'}{[A]\overline{S'} \sqsubseteq \overline{S}}$ **PR**

Regression to **P**rogression $\dfrac{[A]\overline{S'} \sqsubseteq \overline{S}}{S[A] \sqsubseteq S'}$ **RP**

Basic Statements

B1 $\bigcap_{L_{\mathbf{R}} \in \mathcal{L}} L_{\mathbf{R}} \subseteq \bigcup_{L'_{\mathbf{R}} \in \mathcal{L}'} L'_{\mathbf{R}}$ with $|\mathcal{L}| + |\mathcal{L}'| \leq r$

B2 $(\bigcap_{X_{\mathbf{R}} \in \mathcal{X}} X_{\mathbf{R}})[A] \cap \bigcap_{L_{\mathbf{R}} \in \mathcal{L}} L_{\mathbf{R}} \subseteq \bigcup_{L'_{\mathbf{R}} \in \mathcal{L}'} L'_{\mathbf{R}}$ with $|\mathcal{X}| + |\mathcal{L}| + |\mathcal{L}'| \leq r$

B3 $[A](\bigcap_{X_{\mathbf{R}} \in \mathcal{X}} X_{\mathbf{R}}) \cap \bigcap_{L_{\mathbf{R}} \in \mathcal{L}} L_{\mathbf{R}} \subseteq \bigcup_{L'_{\mathbf{R}} \in \mathcal{L}'} L'_{\mathbf{R}}$ with $|\mathcal{X}| + |\mathcal{L}| + |\mathcal{L}'| \leq r$

B4 $L_{\mathbf{R}} \subseteq L'_{\mathbf{R}'}$

B5 $A \subseteq A'$

Efficient Verification

In what follows we describe the set of operations we consider. An **R**-formula φ is a particular instance of formalism **R**. It is associated with a set of variables $vars(\varphi)$, which is a superset of (but not necessarily identical to) the set of variables occurring in φ. Furthermore, $vars(\varphi)$ follows a strict total order $\prec$. We denote the size of the representation as $\|\varphi\|$ and the amount of models as $|\varphi|$.

MO (model testing)
Given **R**-formula φ and truth assignment $\mathcal{I}$, test whether $\mathcal{I} \models \varphi$. Note that $\mathcal{I}$ must assign a value to all $v \in vars(\varphi)$ (if it assigns values to other variables not occurring in φ, they may be ignored).

CO (consistency)
Given **R**-formula φ, test whether φ is satisfiable.

VA (validity)
Given **R**-formula φ, test whether φ is valid.

CE (clausal entailment)
Given **R**-formula φ and clause (i.e. disjunction of literals) γ, test whether $\varphi \models \gamma$.

IM (implicant)
Given **R**-formula φ and cube (i.e. conjunction of literals) δ, test whether $\delta \models \varphi$.

SE (sentential entailment)
Given **R**-formulas φ and ψ, test whether $\varphi \models \psi$.

ME (model enumeration)
Given **R**-formula φ, enumerate all models of φ (over $vars(\varphi)$)

$\wedge$BC (bounded conjunction)
Given **R**-formulas φ and ψ, construct an **R**-formula representing $\varphi \wedge \psi$.

$\wedge$C (general conjunction)
Given **R**-formulas $\varphi_1, \ldots, \varphi_n$, construct an **R**-formula representing $\varphi_1 \wedge \cdots \wedge \varphi_n$.

$\vee$BC (bounded disjunction)
Given **R**-formulas φ and ψ, construct an **R**-formula representing $\varphi \vee \psi$.

$\vee$C (general disjunction)
Given **R**-formulas $\varphi_1, \ldots, \varphi_n$, construct an **R**-formula representing $\varphi_1 \vee \cdots \vee \varphi_n$.

$\neg$C (negation)
Given **R**-formula φ, construct an **R**-formula representing $\neg\varphi$.

CL (conjunction of literals)
Given a conjunction φ of literals, construct an **R**-formula representing φ.

RN (renaming)
Given **R**-formula φ and an injective variable renaming $r : vars(\varphi) \to V'$, construct an **R**-formula representing $\varphi[r]$, i.e., φ with each variable v replaced by $r(v)$.

RN$_\prec$ (renaming consistent with order)
Same as **RN**, but r must be consistent with the variable order in the sense that if $v_1, v_2 \in vars(\varphi)$ with $v_1 \prec v_2$, then $r(v_1) \prec r(v_2)$.

toCNF (transform to CNF)
Given **R**-formula φ, construct a CNF formula that is equivalent to φ.

toDNF (transform to DNF)
Given **R**-formula φ, construct a DNF formula that is equivalent to φ.

CT (model count)
Given **R**-formula φ, count how many models φ has.

Theorem 1. *The statement* $\bigcap_{L_i \in \mathcal{L}} L_i \subseteq \bigcup_{L'_i \in \mathcal{L}'} L'_i$ *where* $|\mathcal{L}| + |\mathcal{L}'| \leq r$ *and the involved state set variables are represented with a set of* $\mathbf{R}$*-formulas* Φ *can be verified in polynomial time in* $\|\Phi\|$ *if* $\mathbf{R}$ *efficiently supports one of the options in the corresponding cell:*

	$\mathcal{L}^+ + \mathcal{L}'^- = 0$	$\mathcal{L}^+ + \mathcal{L}'^- = 1$	$\mathcal{L}^+ + \mathcal{L}'^- > 1$
$\mathcal{L}^- + \mathcal{L}'^+ = 0$		***CO***	***CO***, $\wedge$***BC*** ***toDNF***
$\mathcal{L}^- + \mathcal{L}'^+ = 1$	***VA***	***SE***	***SE***, $\wedge$***BC*** ***toDNF***, ***IM***
$\mathcal{L}^- + \mathcal{L}'^+ > 1$	***VA***, $\vee$***BC*** ***toCNF***	***SE***, $\vee$***BC*** ***toCNF***, ***CE***	***SE***, $\wedge$***BC***, $\vee$***BC*** ***toDNF***, ***IM***, $\vee$***BC*** ***toCNF***, ***CE***, $\wedge$***BC***

where $\mathcal{X}^+$ *is the number of non-negated literals in* $\mathcal{X}$ *and* $\mathcal{X}^-$ *the number of negated literals in* $\mathcal{X}$ *for* $\mathcal{X} \in \{\mathcal{L}, \mathcal{L}'\}$*.*

Theorem 2. *The statements* $(\bigcap_{X_i \in \mathcal{X}} X_i)[A] \cap \bigcap_{L_i \in \mathcal{L}} \subseteq \bigcup_{L'_i \in \mathcal{L}'} L'_i$ *and* $[A](\bigcap_{X_i \in \mathcal{X}} X_i) \cap \bigcap_{L_i \in \mathcal{L}} \subseteq \bigcup_{L'_i \in \mathcal{L}'} L'_i$ *where* $|\mathcal{X}| + |\mathcal{L}| + |\mathcal{L}'| \leq r$ *and the involved state set variables are represented with a set of* $\mathbf{R}$*-formulas* Φ *can be verified in time polynomial in* $\|\Phi\|$ *and* $|A|$ *if* $\mathbf{R}$ *efficiently supports one of the options in the corresponding cell:*

$\mathcal{L}^- + \mathcal{L}'^+ = 0$	***CO***, $\wedge$***BC***, ***CL***, ***RN***$_\prec$
$\mathcal{L}^- + \mathcal{L}'^+ = 1$	***SE***, $\wedge$***BC***, ***CL***, ***RN***$_\prec$
$\mathcal{L}^- + \mathcal{L}'^+ > 1$	***SE***, $\vee$***BC***, $\wedge$***BC***, ***CL***, ***RN***$_\prec$ ***toCNF***, ***CE***, $\wedge$***BC***, ***CL***, ***RN***$_\prec$

where $\mathcal{X}^+$ *is the number of non-negated literals in* $\mathcal{X}$ *and* $\mathcal{X}^-$ *the number of negated literals in* $\mathcal{X}$ *for* $\mathcal{X} \in \{\mathcal{L}, \mathcal{L}'\}$*.*

Theorem 3. *The statement* $L \subseteq L'$ *where the two involved state set variables are represented by* $\varphi_{\mathbf{R}}$ *and* $\psi_{\mathbf{R}'}$ *with* $\mathbf{R} \neq \mathbf{R}'$ *can be verified in polynomial time in* $\|\varphi_{\mathbf{R}}\|$ *and* $\|\psi_{\mathbf{R}'}\|$ *in the following cases:*

	$\mathbf{R}$	$\mathbf{R}'$
$\varphi_{\mathbf{R}} \models \psi_{\mathbf{R}'}$ $\neg\psi_{\mathbf{R}'} \models \neg\varphi_{\mathbf{R}}$	***ME***, *ns* ***toDNF*** ***CE*** ***ME***	***MO*** ***IM*** ***toCNF*** ***MO***, *ns*
$\neg\varphi_{\mathbf{R}} \models \psi_{\mathbf{R}'}$ $\neg\psi_{\mathbf{R}'} \models \varphi_{\mathbf{R}}$	***ME***, *ns* ***toCNF*** ***IM*** ***MO***, ***CT***	***MO***, ***CT*** ***IM*** ***toCNF*** ***ME***, *ns*
$\varphi_{\mathbf{R}} \models \neg\psi_{\mathbf{R}'}$ $\psi_{\mathbf{R}'} \models \neg\varphi_{\mathbf{R}}$	***ME***, *ns* ***toDNF*** ***CE*** ***MO***	***MO*** ***CE*** ***toDNF*** ***ME***, *ns*

where "ns" means that the formalism is non-succinct (i.e. the representation size of the formula is in best case linear in the amount of models). If the other involved formula is succinct, it cannot contain variables not mentioned in the non-succinct formula.

If $\mathbf{R}$ *(*$\mathbf{R}'$*) supports* $\neg$***C*** *and* $\neg\varphi_{\mathbf{R}}$ *(*$\neg\psi_{\mathbf{R}'}$*) occurs, we can also reduce the case to* $\varphi_{\mathbf{R}} \models \psi_{\mathbf{R}'}$*.*