M-N Without Permutations

Ian Dardik

April 1, 2022

1 Introduction

Finding an inductive invariant is key for proving the correctness of a distributed protocol with respect to a safety property. As such, a considerable amount of effort has been dedicated to finding and proving an inductive invariant for a given system. For example, Ivy will guide a user to interactively find an inductive invariant within the confines of a decidable fragment of FOL. In the past few years there has also been a host of research into inductive invariant synthesis for parameterized distributed protocols. The synthesis tools that remain within the bounds of a decidable logic fragment are able to guarantee that they produce an inductive invariant, however, any tool that produces a candidate inductive invariant for a system that falls outside of a decidable fragment offers no guarantee that the candidate is indeed correct. In this note, we assume that a candidate inductive invariant is *given* and we exclusively focus on the verification step.

We have discovered a syntactic class of protocols which exhibit a *cutoff* for the number of finite protocol instances which need to be verified. We have captured this result in the M-N Theorem.

In this note we begin by introducing the Sort-Restricted to PNF Language (SRPL), the logic language that we use to encode our class of protocols. We then introduce our encoding of protocols as a transition system in SRPL. Next, we will prove some key lemmas before finally presenting and proving the M-N Theorem.

2 Sort-Restricted to PNF Language

In this section we will define SRPL as a parameterized grammar. SRPL formulas are parameterized by a non-sorted grammar *ns* as well as single sort E of indistinguishable elements.

Definition 1. Let $\mathcal{V}$ be a countable set of variables, E be an infinitely countable sort of indistinguishable elements, and *sv* be an input grammar that may not refer to E. A formula in SRPL is defined by the grammar for the production rule of *srpl*:

$$\begin{array}{lll} arg & ::= x & \text{for any } x \in \mathcal{V} \\ arg\_list & ::= arg & \\ arg\_list & ::= arg, arg\_list & \\ Q & ::= \forall \mid \exists & \\ srpl & ::= Q\, x \in E,\ ns(arg\_list) & \text{for any } x \in \mathcal{V} \\ srpl & ::= Q\, x \in E,\ srpl & \text{for any } x \in \mathcal{V} \end{array}$$

The input grammar *ns* has a single requirement–that it cannot explicitly refer to E–and therefore is quite general. We now provide an example of an input grammar to illustrate a potential use case.

Example 1. Let $\mathcal{S}$ be a finite set of state variables, $\mathcal{A}$ be a countable set of constants, and let $\mathcal{V}$ be a countable set of variables. We define the grammar *sample* that is parameterized on the variable symbols $x_1, ..., x_n$ by the following production rules:

$$\begin{array}{lll}
prim(x_1,...,x_n) & ::= v & \text{for any } v \in \mathcal{S} \\
prim(x_1,...,x_n) & ::= y & \text{for any } y \in \mathcal{V} \\
prim(x_1,...,x_n) & ::= a & \text{for any } a \in \mathcal{A} \\
prim(x_1,...,x_n) & ::= x_i & \text{for any } 1 \leq i \leq n \\
prim(x_1,...,x_n) & ::= prim(x_1,...,x_n)[prim(x_1,...,x_n)] & \\
sample(x_1,...,x_n) & ::= prim(x_1,...,x_n) = prim(x_1,...,x_n) & \\
sample(x_1,...,x_n) & ::= \neg sample(x_1,...,x_n) & \\
sample(x_1,...,x_n) & ::= sample(x_1,...,x_n) \wedge sample(x_1,...,x_n) & \\
sample(x_1,...,x_n) & ::= \forall x \in sample(arg_list(x_1,...,x_n)),\ sample(x_1,...,x_n) & \text{for any } x \in \mathcal{V}
\end{array}$$

Notice that *sample* formulas have no way to refer to the sort E directly, and hence cannot quantify over E nor take its cardinality. We will use $\vee$, $\exists$, $\rightarrow$, etc. as syntactic sugar in *sample* formulas, defined in the expected way.

Definition 2 (Instance)**.** Let ψ be a SRPL formula and let $H \subseteq E$ such that $H \neq \emptyset$. Then we define $\psi(E \mapsto H)$ by the following rules on the SRPL grammar:

$$\begin{array}{lll}
x(E \mapsto H) & := x & \text{for any } x \in \mathcal{V} \\
{[arg, arg_list]}(E \mapsto H) & := arg, arg_list & \\
{[Q\, x \in E,\ ns(arg_list)]}(E \mapsto H) & := Q\, x \in H,\ ns(arg_list) & \text{for any } x \in \mathcal{V} \\
{[Q\, x \in E,\ srpl]}(E \mapsto H) & := Q\, x \in H,\ [srpl(E \mapsto H)] & \text{for any } x \in \mathcal{V}
\end{array}$$

In other words, $\psi(E \mapsto H)$ is the formula ψ with E replaced with H. We call $\psi(E \mapsto H)$ an *instance* of ψ, and when H is finite, we call $\psi(E \mapsto H)$ a *finite instance* of ψ.

Definition 3 (Finite Instance Notation)**.** We use a special shorthand for finite instaces that mirrors the notation described in [1]. Let ψ be a SRPL formula and $k > 0$ be given. Then $\psi(k) := \psi(E \mapsto \{e_1, ..., e_k\})$ where each $e_i \in E$ is arbitrary and distinct. We can also write $E(k) := \{e_1, ..., e_k\}$ where each $e_i \in E$ is arbitrary and distinct.

Definition 4 (Valid SPRL Formula)**.** Let ψ be a SRPL formula. Then ψ is valid iff $\psi(E \mapsto H)$ is valid for every $H \subseteq E$.

Lemma 1. Let ψ be a SRPL formula. Then ψ is valid iff $\psi(k)$ is valid for all $k > 0$.

3 *E*-Ground Formulas

Definition 5 (ToEGround)**.** Let ψ be a SRPL formula, $R \subseteq \mathcal{V}$ be the variables that occur in ψ that quantify over E, let $H \subseteq E$ such that $H \neq \emptyset$, and let $\rho : R \rightarrow H$ be given. Then we define $\text{ToEGround}(\psi, \rho)$ by the following rules on the SRPL grammar:

$$\begin{array}{lll}
\text{ToEGround}(x, \rho) & := \rho(x) & \text{for any } x \in R \\
\text{ToEGround}([arg, arg_list], \rho) & := \text{ToEGround}(arg, \rho), \text{ToEGround}(arg_list, \rho) & \\
\text{ToEGround}([Q\, x \in E,\ ns(arg_list)], \rho) & := ns(\text{ToEGround}(arg_list, \rho)) & \text{for any } x \in \mathcal{V} \\
\text{ToEGround}([Q\, x \in E,\ srpl], \rho) & := \text{ToEGround}(srpl, \rho) & \text{for any } x \in \mathcal{V}
\end{array}$$

Definition 6 (EGround)**.** A formula g is an *e-ground* formula iff there exists a SRPL formula ψ and a mapping ρ such that $g = \text{ToEGround}(\psi, \rho)$. Moreover, we call g a *ground instance* of ψ.

Notice that e-ground terms are not necessarily *ground terms*, that is, terms without quantifiers. We illustrate this in the following example.

Example 2. Consider the following SRPL formula with input grammar *sample*:

$$\psi := \forall x \in E,\ A[x] \to (\exists y \in B[x], y = 0)$$

where $A \in (E \to \{true, false\})$ and $B \in (E \to \mathbb{N})$ are state variables. Let $H = \{e_1, e_2, e_3\}$ and $\rho(x) = e_1$, then:

$$\text{ToEGround}(\psi, \rho) = A[e_1] \to (\exists y \in B[e_1], y = 0)$$

is an e-ground term. However, it is not a ground term because it contains a quantifier.

Definition 7 (EGr)**.** Let ψ be a SRPL formula and let $H \subseteq E$ be finite. Then:

$$\text{EGr}(\psi, H) := \{g \mid \exists \rho,\ g = \text{ToEGround}(\psi, \rho)\}$$

$\text{EGr}(\psi, H)$ is the set of all possible e-ground formulas of the finite instance $\psi(E \mapsto H)$.

Example 3. Recall the SRPL formula with input grammar *sample* from the previous example:

$$\psi := \forall x \in E,\ A[x] \to (\exists y \in B[x], y = 0)$$

Let $H = \{e_1, e_2, e_3\}$, then:

$$\begin{aligned}\text{EGr}(\psi, H) = \{&A[e_1] \to (\exists y \in B[e_1], y = 0),\\ &A[e_2] \to (\exists y \in B[e_2], y = 0),\\ &A[e_3] \to (\exists y \in B[e_3], y = 0)\}\end{aligned}$$

4 Transition System

We encode a protocol as a transition system $T = (I, \Delta)$ where I is the initial constraint restricted to universal quantification over E and Δ is the transition relation restricted to existential quantification over E, and both are encoded in SRPL. We assume that an inductive invariant candidate Φ is given in SRPL, and is restricted to universal quantification over E. We use the notation $T(E \mapsto H) := (I(E \mapsto H), \Delta(E \mapsto H))$ where $H \subseteq E$.

Definition 8 (States)**.**

$$\text{States}(H) := \{s \mid s \text{ is a state of } T(E \mapsto H)\}$$

In this note we consider a "state" $s \in \text{States}(H)$ to be a ground formula. More specifically, s is a conjunction of constraints that describe a single state in $T(E \mapsto H)$.

Definition 9 (Inductive Invariant)**.** Φ is an inductive invariant iff $I \to \Phi$ and $\Phi \wedge \Delta \to \Phi'$ are valid formulas.

5 Lemmas

Lemma 2. Let $k \in \mathbb{N}$ such that $s \in \text{States}(k)$ and F is a universally quantified formula. Then:

$$(s \to F(k)) \leftrightarrow (\forall f \in \text{EGr}(F,k), s \to f)$$

Proof. Suppose that $s \to F(k)$. For an arbitrary formula $f \in \text{EGr}(F,k)$, $F(k) \to f$ and hence we see that $s \to F(k) \wedge F(k) \to f$. It follows that $s \to f$.

Now suppose that $\forall f \in \text{EGr}(F,k), s \to f$. Suppose, for the sake of contradiction, that $\neg(s \to F(k))$. Then it must be the case that $s \wedge \neg F(k)$. We know that F is unversally quantified, so let $F(k) := \forall x_1, ..., x_m \in P, \phi(x_1, ..., x_m)$ where $m \geq 1$. Then, because $\neg F(k)$ holds, it must be the case that $\exists x_1, ..., x_m \in P, \neg\phi(x_1, ..., x_m)$. However, $\phi(x_1, ..., x_m) \in \text{EGr}(F,k)$ which, by our original assumption, implies $\neg s$. Hence we have both s and $\neg s$ and we have reached a contradiction. □

6 The M-N Theorem

In this section, we will establish initiation and consecution in two separate lemmas using similar techniques. The M-N Theorem follows immediately from these two lemmas.

Lemma 3 (M-N Initiation)**.** Let m be the number of variables that I quantifies over. Then if $I(m) \to \Phi(m)$ is valid, $I(k) \to \Phi(k)$ is also valid for all $k > m$.

Lemma 4 (M-N Consecution)**.** Let m be the number of variables that Φ quantifies over and n be the number of variables that Δ quantifies over. Then if $\Phi(m+n)$ is inductive, $\Phi(k)$ is also inductive for any $k > m+n$.

Proof. Assume that $[\Phi \wedge \Delta \to \Phi'](m+n)$ is valid. Let $k > m+n$ be given, we want to show that $[\Phi \wedge \Delta \to \Phi'](k)$ is also valid. Let $H = \{e_1, ..., e_k\} \subseteq E$ be an arbitrary finite instance of E. Let $s \in \text{States}(H)$ such that $s \to \Phi(E \mapsto H)$ and let $\delta \in \text{EGr}(\Delta, H)$ such that $\delta \to \Delta(E \mapsto H)$. Then $(s \wedge \delta)$ is a formula that describes the states reachable from s in one "δ step", and it suffices to show that $(s \wedge \delta) \to \Phi'(E \mapsto H)$. Furthermore, let $\phi' \in \text{EGr}(\Phi', H)$ be arbitrary, then, by Lemma 2 and the fact that Φ' is in PNF and universally quantified, it suffices to show that $(s \wedge \delta) \to \phi'$.

Let $\alpha_1, ..., \alpha_i$ be the unique elements of $\{e_1, ..., e_k\}$ in $(\phi \wedge \delta)$, then we know that $i \leq m+n$ because $\phi \in \text{EGr}(\Phi, H)$ where Φ quantifies over m variables and $\delta \in \text{EGr}(\Delta, H)$ where Δ quantifies over n variables. Let $j = m+n-i$, then we can choose $\beta_1, ..., \beta_j$ such that $\{\beta_1, ..., \beta_j\} \subseteq (\{e_1, ..., e_k\} - \{\alpha_1, ..., \alpha_i\})$. Notice that $|\{\alpha_1, ..., \alpha_i, \beta_1, ..., \beta_j\}| = m+n$, and hence, by our initial assumption:

$$[\Phi \wedge \Delta \to \Phi'](E \mapsto \{\alpha_1, ..., \alpha_i, \beta_1, ..., \beta_j\})$$

must be a valid formula.

Now, $s \to \Phi(E \mapsto \{\alpha_1, ..., \alpha_i, \beta_1, ..., \beta_j\})$ because Φ is in PNF and universally quantified (need lemma). Furthermore, $\delta \to \Delta(E \mapsto \{\alpha_1, ..., \alpha_i, \beta_1, ..., \beta_j\})$ because Δ is in PNF and restricted to existential quantification (need lemma). Thus we see:

$$(s \wedge \delta) \to [\Phi \wedge \Delta](E \mapsto \{\alpha_1, ..., \alpha_i, \beta_1, ..., \beta_j\}) \to \Phi'(E \mapsto \{\alpha_1, ..., \alpha_i, \beta_1, ..., \beta_j\}) \to \phi'$$

□

Next we present the M-N Theorem:

Theorem 1 (M-N)**.** Suppose that Φ is in PNF with only universal quantifiers, while Δ is in PNF with only existential quantifiers. Let m be the number of variables that Φ quantifies over and n be the number of variables that Δ quantifies over. If $\Phi(m+n)$ is an inductive invariant, then $\Phi(k)$ is also an inductive invariant for any $k > m+n$.

Proof. This follows immediately from the previous two lemmas. □

References

[1] Aman Goel and Karem Sakallah. On Symmetry and Quantification: A New Approach to Verify Distributed Protocols. In *NASA Formal Methods Symposium*, pages 131–150. Springer, 2021.