M-N Without Permutations

Ian Dardik

April 4, 2022

1 Introduction

Finding an inductive invariant is key for proving safety of a distributed protocol. As such, a considerable amount of effort has been dedicated to aid engineers and researchers in finding and proving an inductive invariant for a given system. Ivy, for example, will interactively guide a user towards discovering an inductive invariant, while many other tools attempt to synthesize an inductive invariant with little to no help. Many tools operate within the confines of a decidable fragment of FOL which makes it possible to *prove* that the output is, indeed, an inductive invariant. However, tools that may accept protocols and properties outside of a decidable FOL fragment–such as IC3PO–offer no theoretical guarantees that the output is a correct inductive invariant, and may rely on heuristics instead.

In this note, we choose to focus exclusively on verifying that a candidate is inductive invariant, assuming that a candidate is already provided. We have discovered a syntactic class of protocols that lie outside of a decidable logic fragment, but 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.

We begin by introducing the Sort-Quantifiers Restricted to Prenex Normal Form 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-Quantifiers Restricted to PNF Language

In this section we will define SRPL(E, G) as a grammar parameterized by a sort E and an *input grammar* G.

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

arg	$::= x$	for any $x \in \mathcal{V}$
arg_list	$::= arg$	
arg_list	$::= arg, arg\_list$	
Q	$::= \forall \mid \exists$	
$srpl$	$::= Q\, x \in E,\ G(arg\_list)$	for any $x \in \mathcal{V}$
$srpl$	$::= Q\, x \in E,\ srpl$	for any $x \in \mathcal{V}$

We will often refer to a grammar SRPL(E, G) and implicitly assume that E and G are either given or previously defined.

We now provide an example of an input grammar to illustrate a potential use case of the SRPL grammar.

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 $x_1, ..., x_n$ and is by the following production rules:

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

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 $\lor$, $\exists$, $\rightarrow$, etc. as syntactic sugar in *sample* formulas, defined in the expected way.

The following is an example of a SRPL(E, *sample*) formula:

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

where $A \in (E \rightarrow \{true, false\})$ and $B \in (E \rightarrow \mathcal{P}(\mathbb{N}))$ are state variables, and $\mathcal{P}$ denotes the power set.

The sort E in a SRPL grammar is assumed to be countably infinite, however, we are particularly interested in verifying SRPL formulas for arbitrary *finite* sized subsets of E, since the sort presumably models a real world system with finite resources. Hence, we will be primarily concerned with a *finite instance* of a formula. We formally introduce this concept below.

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

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

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 may use a special shorthand for finite instaces that mirrors the notation described in [1]. Let ψ be a SRPL(E, G) 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.

Definition 4 (Subinstance)**.** Let ψ be a SRPL(E, G) formula and $\psi(E \mapsto H_1)$ be an instance of ψ. Then for any $H_2 \subseteq H_1$ where $H_2 \neq \emptyset$, we call $\psi(E \mapsto H_2)$ a *subinstance* of $\psi(E \mapsto H_1)$.

We now define validity of a SRPL formula in terms of finite instances.

Definition 5 (Valid SRPL Formula)**.** Let E be a sort, G be a valid SRPL input grammar, and ψ be a SRPL(E, G) formula. Then ψ is valid iff every finite instance of ψ is valid.

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

Proof. Notice that, for a given $k > 0$, $\psi(k)$ is valid iff $\psi(E \mapsto H)$ for arbitrary H such that $|H| = |\{e_1, ..., e_k\}| = k$. Thus it follows that every $\psi(k)$ is valid iff every finite instance of ψ is valid; our desired result follows immediately. $\square$

3 E-Ground Formulas

In this section we introduce E-ground formulas, an important tool that will be used in the proof for the M-N Theorem. We begin by introducing the ToEGround operator which we then use to formally define an E-ground formula.

Definition 6 (ToEGround)**.** Let E be a sort, G be a valid SRPL input grammar, and ψ be a SRPL(E, G) formula. Next, let $R \subseteq \mathcal{V}$ be the variables that occur in ψ that quantify over E, and let $\rho : R \to E$ be given. Then we define ToEGround(ψ, ρ) by the following rules on the SRPL(E, G) grammar:

ToEGround(x, ρ)	$:= \rho(x)$	for any $x \in R$
ToEGround$([arg, arg_list], \rho)$	$:=$ ToEGround(arg, ρ), ToEGround(arg_list, ρ)	
ToEGround$([Q\, x \in E,\ G(arg_list)], \rho)$	$:= G($ToEGround$(arg_list, \rho))$	for any $x \in \mathcal{V}$
ToEGround$([Q\, x \in E,\ srpl], \rho)$	$:=$ ToEGround$(srpl, \rho)$	for any $x \in \mathcal{V}$

Here we assume that each quantifier for E in ψ gets a unique variable name. This assumption comes without loss of generality since we can always alpha-rename duplicate quantifier variables.

Definition 7 (EGround)**.** A formula g is an *E-ground formula* iff there exists a SRPL formula ψ and a mapping ρ such that $g =$ ToEGround(ψ, ρ). Moreover, we call g a *ground instance* of ψ.

Notice that E-ground formulas are not necessarily vanilla ground formulas, that is, formulas without quantifiers. We illustrate this in the following example.

Example 2. Recall the following SRPL$(E, sample)$ formula from the previous example $\psi := \forall x \in E,\ A[x] \to (\exists y \in B[x], y = 0)$. Assume $E = \{e_1, ...\}$ and let $\rho(x) = e_1$, then:

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

is an E-ground formula. However, it is not a ground formula because it contains a quantifier. Notice that we can also take the ToEGround of a finite instance:

$$\text{ToEGround}(\psi(E \mapsto \{e_1, e_2, e_3\}), \rho) = A[e_1] \to (\exists y \in B[e_1], y = 0)$$

We next introduce the EGr operator. This operator offers a simple notation for describing the set of all possible E-ground instances for a given SRPL formula.

Definition 8 (EGr)**.** Let ψ be a SRPL formula and let $\psi(E \mapsto H)$ be a finite instance. Let $R \subseteq \mathcal{V}$ be the variables that occur in ψ that quantify over E. Then:

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

In other words, EGr(ψ, H) is the set of all possible E-ground formulas of the finite instance $\psi(E \mapsto H)$.

Example 3. Let $\psi := \forall x \in E,\ A[x] \to (\exists y \in B[x], y = 0)$ from the previous example, then:

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

4 Transition System

Let a sort E be given along with a valid SRPL input grammar G. We encode a protocol as a transition system $T = (I, \Delta)$ where I is the initial constraint and Δ is the transition relation, both formulas encoded in SRPL(E, G). We assume that I is restricted to universal quantification over E while Δ is restricted to existential quantification over E. Further assume that an inductive invariant candidate Φ is given in SRPL(E, G) 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 $H \subseteq E$ to denote an *instance* of T.

For the remainder of this note we will refer to E, T, I, Δ, and Φ as defined above. In particular, we will no longer think of E as a generic input sort to a SRPL grammar; it is now *the* sort of T that we specifically use as input to the SRPL grammar that is used to encode I, Δ, and Φ.

Definition 9 (Inductive Invariant)**.** Let ψ be a SRPL(E, G) formula. Then ψ is an inductive invariant iff $I \to \psi$ and $\psi \wedge \Delta \to \psi'$ are valid formulas.

Definition 10 (States)**.** Let H be a nonempty, finite subset of E. Then:

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

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

Example 4. Recall the running example with state variables $A \in (E \to \{true, false\})$ and $B \in (E \to \mathcal{P}(\mathbb{N}))$. Here are several examples of "states":

$$\begin{aligned}
&A[e_1] = true \wedge B[e_1] = \{1, 2, 9\} && \in \mathrm{States}(\{e_1\})\\
&A[e_1] = false \wedge B[e_1] = \{0\} && \in \mathrm{States}(\{e_1\})\\
&A[e_1] = true \wedge A[e_2] = true \wedge B[e_1] = \emptyset \wedge B[e_2] = \emptyset && \in \mathrm{States}(\{e_1, e_2\})\\
&A[e_1] = true \wedge A[e_2] = false \wedge B[e_1] = \{1, 2, 9\} \wedge B[e_2] = \emptyset && \in \mathrm{States}(\{e_1, e_2\})\\
&A[e_1] = false \wedge A[e_2] = false \wedge B[e_1] = \{0, ..., 74\} \wedge B[e_2] = \{0, 2, 4\} && \in \mathrm{States}(\{e_1, e_2\})
\end{aligned}$$

This example showcases the power of using formulas to describe states; it allows us to reason about states across different finite instances. For example:

$$\begin{gathered}(A[e_1] = true \wedge A[e_2] = false \wedge B[e_1] = \{1, 2, 9\} \wedge B[e_2] = \emptyset) \to\\ (A[e_1] = true \wedge B[e_1] = \{1, 2, 9\})\end{gathered}$$

Here, the first state is stronger than the second state.

We now show a key lemma that shows that a state that satisfies a SRPL formula restricted to universal quantification on E can be described equivalently in terms of the formulas E-ground formulas.

Lemma 2. Let ψ be a SRPL(E, G) formula restricted to universal quantification on E, and $\psi(E \mapsto H)$ be a finite instance. Let $s \in \text{States}(H)$, then:

$$(s \to \psi(E \mapsto H)) \leftrightarrow (\forall g \in \text{EGr}(\psi, H), s \to g)$$

Proof. TODO need to be cleaned up with latest notation.

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. $\square$

5 Lemmas For Restricting The Sort Domain

The following two lemmas show two interesting ways a SRPL formula can remain true if elements are removed from the sort E. These two properties are key to proving the M-N Theorem because they show how a property ψ that holds as $\psi(k)$ may also hold for $\psi(m + n)$ where $k > m + n$.

The first lemma shows that any state that satisfies a property ψ–restricted to universal quantificaiton over E–will also satisfy any subinstance (maybe add this as a technical def?) of ψ. While this result is rather intuitive if ψ is restricted to quantification only on E, this result is perhaps less clear when additional quantifiers are included in the formula. The key reason this result holds is because a subinstance of ψ will only affect the domain of sort E, leaving all domains of the input grammar unaffected.

The second lemma essentially states that, for a property ψ restricted to existential quantification on E, any ground term that implies ψ will also imply the subinstance of ψ when E is restricted to the sort elements in the ground term of ψ. This result is also intuitive if ψ is restricted to quantification only on E because the sort elements in the ground term witness the existentials of ψ, and less clear when additional quantifiers are included in the formula. The key reason this result holds is the same–all domains of the input grammar are unaffected in any subinstance of ψ. In fact, the lemma goes a step further and shows that the ground term implies *any* instance of ψ that contains its sort elements.

Lemma 3. Let ψ be a SRPL(E, G) formula restricted to universal quantification on E and let $\psi(E \mapsto H_1)$ be an instance of ψ. Now let $\psi(E \mapsto H_2)$ be a subinstance of $\psi(E \mapsto H_1)$, then:

$$(s \to \psi(E \mapsto H_1)) \to (s \to \psi(E \mapsto H_2))$$

Proof. Suppose that $s \to \psi(E \mapsto H_1)$, it suffices to show that $s \to \psi(E \mapsto H_2)$. We know that $\psi(E \mapsto H_2)$ is in the form:

$$\psi = \forall x_1 \in H_2, ..., \forall x_m \in H_2,\ F_G(x_1, ..., x_m)$$

where F_G is a formula generated by the input grammar G. Then $s \to \psi(E \mapsto H_2)$ holds iff $s \to F_G(e_1, ..., e_m)$ holds for arbitrary $e_1 \in H_2, ..., e_m \in H_2$. However, this formula must hold by our assumptions that $H_2 \subseteq H_1$ and $s \to \psi(E \mapsto H_1)$ where ψ is unversally quantified over E. $\square$

Lemma 4. Let ψ be a SRPL(E, G) formula restricted to existential quantification on E and let $\psi(E \mapsto H_1)$ be a finite instance. Now let $g \in \text{EGr}(\psi, H_1)$ and $e_1, ..., e_m$ be the elements of H_1 that occur in g. Then for any sort $H_2 \supseteq \{e_1, ..., e_m\}$:

$$(g \to \psi(E \mapsto H_1)) \to (g \to \psi(E \mapsto H_2))$$

Proof. Suppose that $g \to \psi(E \mapsto H_1)$, then it suffices to show that $g \to \psi(E \mapsto H_2)$. We know that ψ is of the form:

$$\psi = \exists x_1 \in H_2, ..., \exists x_m \in H_2,\ F_G(x_1, ..., x_m)$$

where F_G is a formula generated by the input grammar G. Because $g \to \psi(E \mapsto H_1)$, it must be the case that $e_1, ..., e_m$ witness the existential quantifiers of $\psi(E \mapsto H_1)$. However, $\{e_1, ..., e_m\} \subseteq H_2$, and hence $g \to \psi(E \mapsto H_2)$. □

6 The M-N Theorem

In this section, we will establish initiation and consecution in two separate lemmas. The M-N Theorem is then easily proved from these two lemmas.

Lemma 5 (M-N Initiation)**.** Let m be the number of quantifiers over E in I. Then if $I(m) \to \Phi(m)$ is valid, $I(k) \to \Phi(k)$ is also valid for all $k > m$.

Proof. Coming soon. □

Lemma 6 (M-N Consecution)**.** Let m be the number of quantifiers over E in Φ and n be the number of quantifiers over E in Δ. 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 an E-ground 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 restricted to universal quantification on E, it suffices to show that $(s \wedge \delta) \to \phi'$.

Let $\alpha_1, ..., \alpha_i$ be the unique elements of $\{e_1, ..., e_k\}$ that occur 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 $\{\beta_1, ..., \beta_j\} \subseteq (\{e_1, ..., e_k\} - \{\alpha_1, ..., \alpha_i\})$ (define $\{\beta_1, ..., \beta_j\} = \emptyset$ in the case where $j = 0$). 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, $s \to \Phi(E \mapsto \{\alpha_1, ..., \alpha_i, \beta_1, ..., \beta_j\})$ due to Lemma 3 because Φ is restricted to universal quantification on E. Furthermore, $\delta \to \Delta(E \mapsto \{\alpha_1, ..., \alpha_i, \beta_1, ..., \beta_j\})$ by Lemma 4 because Δ is restricted to existential quantification on E. 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)**.** Let m be the number of quantifiers over E in Φ and n be the number of quantifiers over E in Δ. Then if $\Phi(m+n)$ is an inductive invariant, $\Phi(k)$ is also an inductive invariant for any $k > m+n$.

Proof. This follows immediately from Lemma 5 and Lemma 6. □

Perhaps even more important than the M-N Theorem itself, is the following corollary:

Corollary 1. Let m be the number of quantifiers over E in Φ and n be the number of quantifiers over E in Δ. Then if $\Phi(k)$ is an inductive invariant for all $k \in \{1, ..., m+n\}$, then Φ is an inductive invariant for T.

Proof. Suppose that $\Phi(k)$ is an inductive invariant for all $k \in \{1, ..., m+n\}$. By the M-N Theorem, we know that $\Phi(k)$ is also an inductive invariant for all $k > 0$. The result then follows from Lemma 1. (TODO: a bit more is need here) $\square$

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.