

A Cutoff Rule For A Special Class Of Parameterized Distributed Protocols

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 an 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 $\text{SRPL}(E, G)$ 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 $\text{SRPL}(E, G)$ 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 $\text{SRPL}(E, G)$ 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, \dots, x_n by the following production rules:

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

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

The following is an example of a $\text{SRPL}(E, \text{sample})$ formula:

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

where $A \in (E \rightarrow \{\text{true}, \text{false}\})$ 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 $\text{SRPL}(E, G)$ formula and $H \subseteq E$ such that $H \neq \emptyset$. Then we define $\psi(E \mapsto H)$ by the following rules on the $\text{SRPL}(E, G)$ grammar:

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

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 instances that mirrors the notation described in [1]. Let ψ be a $\text{SRPL}(E, G)$ formula and $k > 0$ be given. Then $\psi(k) := \psi(E \mapsto \{e_1, \dots, e_k\})$ where each $e_i \in E$ is arbitrary and distinct.

Definition 4 (Subinstance). Let ψ be a $\text{SRPL}(E, G)$ 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 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(E, G) 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 $|H| = |\{e_1, \dots, e_k\}| = k$. 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 ψ be a SRPL(E, G) formula. Additionally, let $R \subseteq \mathcal{V}$ be the variables that occur in ψ that quantify over E , and let $\rho : R \rightarrow E$ 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($[arg, arg_list], \rho$) | $:= \text{ToEGround}(arg, \rho), \text{ToEGround}(arg_list, \rho)$ | |
| ToEGround($[Q x \in E, G(arg_list)], \rho$) | $:= G(\text{ToEGround}(arg_list, \rho))$ | for any $x \in \mathcal{V}$ |
| ToEGround($[Q x \in E, srpl], \rho$) | $:= \text{ToEGround}(srpl, \rho)$ | 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 $g = \text{ToEGround}(\psi, \rho)$. 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 $\psi := \forall x \in E, A[x] \rightarrow (\exists y \in B[x], y = 0)$. Assume $E = \{e_1, \dots\}$ and let $\rho(x) = e_1$, then:

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

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:

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

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(E, G) 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:

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

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

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

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

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 $\text{SRPL}(E, G)$. 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 $\text{SRPL}(E, G)$ 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 $H \subseteq E$ and $H \neq \emptyset$ 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 $\text{SRPL}(E, G)$ formula. Then ψ is an inductive invariant iff $I \rightarrow \psi$ and $\psi \wedge \Delta \rightarrow \psi'$ are valid formulas.

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

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

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 $T(E \mapsto H)$.

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

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

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

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

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 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:

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

Proof. Suppose that $s \rightarrow \psi(E \mapsto H)$. We can write $\psi(E \mapsto H) = \forall x_1 \in H, \dots, \forall x_m \in H, F_G(x_1, \dots, x_m)$ where F_G is a formula generated by the input grammar G . Then $s \rightarrow F_G(e_1, \dots, e_m)$ for arbitrary $e_1, \dots, e_m \in H$. However $F_G(e_1, \dots, e_m)$ is an arbitrary formula in $\text{EGr}(\psi, H)$, which implies that $\forall g \in \text{EGr}(\psi, H), s \rightarrow g$.

Now suppose that $\forall g \in \text{EGr}(\psi, H), s \rightarrow g$. Further suppose, for the sake of contradiction, that $\neg(s \rightarrow \psi(E \mapsto H))$, or equivalently $s \wedge \neg\psi(E \mapsto H)$. Because ψ is restricted to universal quantification, we can write $\psi(E \mapsto H) = \forall x_1 \in H, \dots, \forall x_m \in H, F_G(x_1, \dots, x_m)$ where F_G is a formula generated by the input grammar G . Thus we see $\neg\psi(E \mapsto H) = \exists x_1 \in H, \dots, \exists x_m \in H, \neg F_G(x_1, \dots, x_m)$. Let $e_1, \dots, e_m \in H$ witness the existentials, i.e. the formula $\neg F_G(e_1, \dots, e_m)$ holds, or equivalently, $F_G(e_1, \dots, e_m) \rightarrow \text{false}$. However, $F_G(e_1, \dots, e_m) \in \text{EGr}(\psi, H)$ and thus, by our initial assumption, we see $s \rightarrow F_G(e_1, \dots, e_m) \rightarrow \text{false}$ which implies $\neg s$ which is a contradiction. \square

5 Lemmas For Restricting The Sort Domain

In this section we present two lemmas on the subinstances of a SRPL formula. We show that, in two particular cases, a SRPL formula ψ will remain invariant even when we remove elements from E . This invariance is key to proving the M-N Theorem because it allows us to conclude that $\psi(k) \rightarrow \psi(m+n)$ for $k > m+n$.

The first lemma, Lemma 3, considers a state s that implies a property ψ that is restricted to universal quantification, i.e. in the form $\psi(E \mapsto H) = \forall x_1 \in H, \dots, \forall x_m \in H, F_G(x_1, \dots, x_m)$. The lemma states the intuitive fact that s also implies any subinstance of $\psi(E \mapsto H)$ as well. It is worthwhile to point out that this fact holds even when F_G contains quantifiers.

The second lemma, Lemma 4, considers an E -ground formula g that implies a property ψ such that $g \in \text{EGr}(\psi, H)$. In this case, ψ is restricted to existential quantification, i.e. it is in the form $\psi(E \mapsto H) = \exists x_1 \in H, \dots, \exists x_m \in H, F_G(x_1, \dots, x_m)$. Essentially, the lemma states that if $g \rightarrow \psi(E \mapsto H)$, then g also implies any subinstance of $\psi(E \mapsto H)$ whose sort domain contains the sort elements that occur in g . The reason this fact holds is because the sort elements that occur in g witness the existentials of ψ in the formula $g \rightarrow \psi(E \mapsto H)$. It is worthwhile to point out that this fact holds as well when F_G contains quantifiers.

These two lemmas also help motivate the disallowance of the sort E in SRPL input grammars. The idea is that, even if we alter the elements of E , the inner formula F_G does not change. Any domains that are quantified over in F_G remain constant when E changes, which explains why we can allow an arbitrary grammar with quantification (that does not refer to E) as an input grammar to SRPL.

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:

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

Proof. Suppose that $s \rightarrow \psi(E \mapsto H_1)$, it suffices to show that $s \rightarrow \psi(E \mapsto H_2)$. We know that $\psi(E \mapsto H_2)$ is in the form $\psi = \forall x_1 \in H_2, \dots, \forall x_m \in H_2, F_G(x_1, \dots, x_m)$, where F_G is a formula generated by the input grammar G . Then $s \rightarrow \psi(E \mapsto H_2)$ holds iff $s \rightarrow F_G(e_1, \dots, e_m)$ holds for arbitrary $e_1 \in H_2, \dots, e_m \in H_2$. However, this formula must hold by our assumptions that $H_2 \subseteq H_1$ and $s \rightarrow \psi(E \mapsto H_1)$ where ψ is universally 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, \dots, e_m be the elements of H_1 that occur in g . Then for any sort $H_2 \supseteq \{e_1, \dots, e_m\}$:

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

Proof. Suppose that $g \rightarrow \psi(E \mapsto H_1)$, then it suffices to show that $g \rightarrow \psi(E \mapsto H_2)$. We know that ψ is of the form $\psi = \exists x_1 \in H_2, \dots, \exists x_m \in H_2, F_G(x_1, \dots, x_m)$, where F_G is a formula generated by the input grammar G . Because $g \rightarrow \psi(E \mapsto H_1)$, it must be the case that e_1, \dots, e_m witness the existential quantifiers of $\psi(E \mapsto H_1)$. However, $\{e_1, \dots, e_m\} \subseteq H_2$, and hence $g \rightarrow \psi(E \mapsto H_2)$. \square

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 $I(m) \rightarrow \Phi(m)$ is valid, $I(k) \rightarrow \Phi(k)$ is also valid for all $k > m$.

Proof. Coming soon. \square

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 $[\Phi \wedge \Delta \rightarrow \Phi'](m+n)$ is valid. Let $k > m+n$ be given, we want to show that $[\Phi \wedge \Delta \rightarrow \Phi'](k)$ is also valid. Let $H = \{e_1, \dots, e_k\} \subseteq E$ be an arbitrary finite instance of E of size k . Let $s \in \text{States}(H)$ such that $s \rightarrow \Phi(E \mapsto H)$ and let $\delta \in \text{EGr}(\Delta, H)$ such that $\delta \rightarrow \Delta(E \mapsto H)$. 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 $(s \wedge \delta) \rightarrow \Phi'(E \mapsto H)$. 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 $(s \wedge \delta) \rightarrow \phi'$.

Let $\alpha_1, \dots, \alpha_i$ be the unique elements of $\{e_1, \dots, e_k\}$ 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 β_1, \dots, β_j such that $\{\beta_1, \dots, \beta_j\} \subseteq (\{e_1, \dots, e_k\} - \{\alpha_1, \dots, \alpha_i\})$ (define $\{\beta_1, \dots, \beta_j\} = \emptyset$ in the case where $j = 0$). Notice that $|\{\alpha_1, \dots, \alpha_i, \beta_1, \dots, \beta_j\}| = m+n$, and hence, by our initial assumption:

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

must be a valid formula.

Now, $s \rightarrow \Phi(E \mapsto \{\alpha_1, \dots, \alpha_i, \beta_1, \dots, \beta_j\})$ due to Lemma 3 because Φ is restricted to universal quantification on E . Furthermore, $\delta \rightarrow \Delta(E \mapsto \{\alpha_1, \dots, \alpha_i, \beta_1, \dots, \beta_j\})$ by Lemma 4 because Δ is restricted to existential quantification on E . Thus we see:

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

\square

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. \square

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 $k \in \{1, \dots, m+n\}$, then Φ is an inductive invariant for T .

Proof. Suppose that $\Phi(k)$ is an inductive invariant for all $k \in \{1, \dots, m+n\}$. 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.