

# A Cutoff Rule For Parameterized Distributed Protocols in Prenex Normal Form

Ian Dardik

February 25, 2022

## 1 Introduction

In this note, we consider the verification problem of a transition system  $T = (I, \Delta)$  parameterized by a single sort  $P$  of identical elements. We assume that a candidate inductive invariant  $\Phi$  (which implies our key safety property) is given, and that both  $\Delta$  and  $\Phi$  are in Prenex Normal Form (PNF). We adopt the convention of [2] where  $T(P)$  is the template of  $T$ , and  $T(|P|)$  is a finite instantiation. We also will consider the prime ( $'$ ) symbol to be an operator that can be recursively applied to a formula, only affecting (sticking to) state variables.

In this note, we will build several lemmas that lead to an interesting result:  $\Phi(P)$  is an inductive invariant for  $T(P)$  iff  $\Phi(m+n)$  is an inductive invariant for  $T(m+n)$ , where  $m$  is the number of variables that  $\Phi$  quantifies over and  $n$  is the number of variables that  $\Delta$  quantifies over. This result is useful for the verification problem laid out above because it reduces the burden to model checking the single finite instance  $T(m+n)$ .

## 2 Finitely Instantiated Properties (FIPs)

In this section we introduce the FIP, a key tool for proving the M-N Theorem.

### 2.1 FIP Basics

**Definition 1** (FIP). Let  $M_\Phi$  and  $M_\Delta$  be the respective matrices of  $\Phi$  and  $\Delta$ . A Finitely Instantiated Property (FIP) of  $[\Phi \wedge \Delta](|P|)$  is a formula  $(M_\Phi \wedge M_\Delta)[v_i \mapsto e_i]$ , where each free variable  $v_i$  has been substituted for a concrete element  $e_i \in P$ .

**Example 1.** Let  $\Phi = \forall p, q \in P, M_\Phi(p, q)$  and  $\Delta = \exists p \in P, M_\Delta(p)$ . Then  $M_\Phi(1, 3) \wedge M_\Delta(2)$  and  $M_\Phi(1, 1) \wedge M_\Delta(1)$  are both FIPs of  $[\Phi \wedge \Delta](3)$  (i.e. for the case when  $P = \{1, 2, 3\}$ ).  $\square$

*Remark 1.* We will often write a FIPs of  $[\Phi \wedge \Delta](|P|)$  in the abstract as  $\phi \wedge \delta$ . We specifically choose “ $\phi \wedge \delta$ ” to show the syntactic correspondence of  $\phi$  to  $\Phi$  and  $\delta$  to  $\Delta$ . In other words, if  $M_\Phi$  is the matrix of  $\Phi$  and  $M_\Delta$  is the matrix of  $\Delta$ , then it is always the case that  $\phi = M_\Phi[v_i \mapsto e_i]$  and  $\delta = M_\Delta[v_i \mapsto e_i]$ .

**Example 2.** Let  $\phi \wedge \delta = M_\Phi(1, 3) \wedge M_\Delta(2)$  be a FIP of  $[\Phi \wedge \Delta](3)$ . By Remark 1,  $\phi = M_\Phi(1, 3)$  and  $\delta = M_\Delta(2)$ .  $\square$

### 2.2 Protocol Assumptions

Now that we have formally defined FIPs, we can describe our requirement that “the elements of sort  $P$  are identical” in precise terms.

**Assumption 1** (Sort Elements Are Identical). Let  $\phi \wedge \delta$  be an arbitrary FIP of  $[\Phi \wedge \Delta](|P|)$  and  $g : P \rightarrow P$  be a bijective function. Then we assume:

$$\phi \leftrightarrow \phi[P \mapsto g(P)]$$

and

$$\phi \wedge \delta \leftrightarrow (\phi \wedge \delta)[P \mapsto g(P)]$$

*Remark 2.* In the context of Assumption 1,  $(\phi \wedge \delta)[P \mapsto g(P)]$  is said to be a *permutation* of  $\phi \wedge \delta$ , and:

$$(\phi \wedge \delta)[P \mapsto g(P)] = g(\phi \wedge \delta)$$

**Example 3.** Let  $\Phi = \forall p \in P, M_\Phi(p)$  and  $\Delta = \exists p, q \in P, M_\Delta(p, q)$ . Consider  $M_\Phi(2) \wedge M_\Delta(1, 3)$  and  $M_\Phi(3) \wedge M_\Delta(1, 2)$  which are both FIPs of  $[\Phi \wedge \Delta](3)$ . Let  $g$  be the permutation:

$$g = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}$$

Then these two FIPs are permutations of each other because

$$g(M_\Phi(2) \wedge M_\Delta(1, 3)) = M_\Phi(3) \wedge M_\Delta(1, 2)$$

and

$$M_\Phi(2) \wedge M_\Delta(1, 3) = g(M_\Phi(3) \wedge M_\Delta(1, 2))$$

□

Because each sort element is identical, we can make the following assumption without loss of generality:

**Assumption 2.** For any sort  $P$ ,  $P = \{1, \dots, |P|\}$ .

This leads to the trivial result:

**Lemma 1.** Let  $P$  and  $Q$  be sorts, then  $|P| \leq |Q| \leftrightarrow P \subseteq Q$ .

*Proof.* Suppose  $|P| \leq |Q|$ . Then by Assumption 2:

$$P = \{1, \dots, |P|\} \subseteq \{1, \dots, |Q|\} = Q$$

Now suppose that  $P \subseteq Q$ . Then by Assumption 2:

$$\{1, \dots, |P|\} = P \subseteq Q = \{1, \dots, |Q|\}$$

Hence it follows that  $|P| \leq |Q|$ . □

## 2.3 FIP Equivalency

In this section we introduce the notion of *equivalency* for FIPs and prove some useful lemmas.

**Definition 2** (Equivalent). Let  $\phi_1 \wedge \delta_1$  and  $\phi_2 \wedge \delta_2$  be FIPs, then  $\phi_1 \wedge \delta_1 \equiv \phi_2 \wedge \delta_2$  iff  $\phi_2 \wedge \delta_2$  is a permutation of  $\phi_1 \wedge \delta_1$ . In this case,  $\phi_1 \wedge \delta_1$  and  $\phi_2 \wedge \delta_2$  are said to be *equivalent*.

**Example 4.** Let  $F_1 = M_\Phi(1, 2) \wedge M_\Delta(1)$ ,  $F_2 = M_\Phi(2, 3) \wedge M_\Delta(2)$  and  $F_3 = M_\Phi(2, 2) \wedge M_\Delta(2)$  be FIPs of  $[\Phi \wedge \Delta](3)$ . Let  $g = (1 \ 2 \ 3)$  be a permutation (using cycle notation) on  $P$ . Then  $F_1 \equiv F_2$  because  $g(F_1) = F_2$ , however,  $F_3$  is a permutation of neither  $F_1$  nor  $F_2$  and hence is not equivalent to either. □

The notion of equivalency is important because it partitions a formula  $\Phi \wedge \Delta$  into distinct classes of FIPs. In the example above,  $F_1$  and  $F_2$  describe the same class of properties/actions because each element of  $P$  is interchangeable for one another. We now present several basic lemmas about FIPs.

**Lemma 2** (Equivalency Is Commutative). Let  $\phi_1 \wedge \delta_1$  and  $\phi_2 \wedge \delta_2$  be FIPs for  $[\Phi \wedge \Delta](|P|)$ , then

$$(\phi_1 \wedge \delta_1 \equiv \phi_2 \wedge \delta_2) \leftrightarrow (\phi_2 \wedge \delta_2 \equiv \phi_1 \wedge \delta_1)$$

*Proof.* Suppose that  $\phi_1 \wedge \delta_1 \equiv \phi_2 \wedge \delta_2$ . Then there exists a permutation  $g$  such that  $g(\phi_1 \wedge \delta_1) = \phi_2 \wedge \delta_2$ . Permutations are bijective and hence are invertible. Then

$$\phi_1 \wedge \delta_1 = g^{-1}(g(\phi_1 \wedge \delta_1)) = g^{-1}(\phi_2 \wedge \delta_2)$$

But  $g^{-1}$  is a permutation itself, and hence  $\phi_1 \wedge \delta_1$  is a permutation of  $\phi_2 \wedge \delta_2$ .

We omit the proof in the other direction because the argument is nearly identical.  $\square$

**Lemma 3.** Let  $\phi_1 \wedge \delta_1$  and  $\phi_2 \wedge \delta_2$  both be FIPs of  $[\Phi \wedge \Delta](|P|)$ . Then:

$$((\phi_1 \wedge \delta_1) \equiv (\phi_2 \wedge \delta_2)) \rightarrow ((\phi_1 \equiv \phi_2) \wedge (\delta_1 \equiv \delta_2))$$

*Proof.* Suppose  $(\phi_1 \wedge \delta_1) \equiv (\phi_2 \wedge \delta_2)$ . Then there exists a permutation  $g$  such that  $g(\phi_1 \wedge \delta_1) = \phi_2 \wedge \delta_2$ . But:

$$g(\phi_1) \wedge g(\delta_1) = g(\phi_1 \wedge \delta_1) = \phi_2 \wedge \delta_2$$

From Remark 1, we see that  $\phi_1$  and  $\phi_2$  are identical up to a finite instantiation, and hence  $g(\phi_1) = \phi_2$ ; a similar argument shows that  $g(\delta_1) = \delta_2$ .  $\square$

**Lemma 4.** Let  $\phi_1$  and  $\phi_2$  be quantifier-free properties parameterized by the sort  $P$ . Then:

$$(\phi_1 \equiv \phi_2) \leftrightarrow (\phi_1' \equiv \phi_2')$$

*Proof.* Suppose that  $\phi_1 \equiv \phi_2$ . Then there exists a permutation  $g$  such that  $g(\phi_1) = \phi_2$ . Recall that the prime operator only affects state variables, and not  $P$  or its elements; on the other hand,  $g$  can only affect the elements of  $P$ . Thus it is the case that  $g(\phi_1)' = g(\phi_1')$ . We now see:

$$\phi_2' = g(\phi_1)' = g(\phi_1')$$

Showing that  $\phi_1' \equiv \phi_2'$  by definition. We omit the proof in the other direction since it is nearly identical.  $\square$

**Lemma 5.** Let  $\phi_1$  and  $\phi_2$  be quantifier-free properties parameterized by the sort  $P$ . Suppose that  $\phi_1 \equiv \phi_2$ . Then:

$$\phi_1 \leftrightarrow \phi_2$$

*Proof.* Because  $\phi_1 \equiv \phi_2$ , there exists a permutation  $g$  such that  $g(\phi_1) = \phi_2$ . However  $g$  is bijective, and hence by Assumption 1:

$$\phi_1 \leftrightarrow (\phi_1)[P \mapsto g(P)] = \phi_2$$

$\square$

**Lemma 6.** Let  $\phi_1 \wedge \delta_1$  and  $\phi_2 \wedge \delta_2$  both be FIPs for  $[\Phi \wedge \Delta](|P|)$ . Suppose that  $\phi_1 \wedge \delta_1 \equiv \phi_2 \wedge \delta_2$ . Then:

$$(\phi_1 \wedge \delta_1) \leftrightarrow (\phi_2 \wedge \delta_2)$$

*Proof.* Because  $\phi_1 \wedge \delta_1 \equiv \phi_2 \wedge \delta_2$ , there exists a permutation  $g$  such that  $g(\phi_1 \wedge \delta_1) = \phi_2 \wedge \delta_2$ . However  $g$  is bijective, and hence by Assumption 1:

$$\phi_1 \wedge \delta_1 \leftrightarrow (\phi_1 \wedge \delta_1)[P \mapsto g(P)] = \phi_2 \wedge \delta_2$$

$\square$

## 2.4 The FIPS Operator

In this section we introduce the FIPS operator:

**Definition 3.** Let  $\Phi$  and  $\Delta$  be PNF properties with respective matrices  $\phi$  and  $\delta$ . Suppose that  $\Phi$  quantifies over  $m \in \mathbb{N}$  variables while  $\Delta$  quantifies over  $n \in \mathbb{N}$  variables. Then:

$$\text{FIPS}(\Phi \wedge \Delta, |P|) := \{\phi(v_1, \dots, v_m) \wedge \delta(w_1, \dots, w_n) \mid v_1, \dots, v_m, w_1, \dots, w_n \in P\}$$

The FIPS operator simply contains every possible FIP for a given formula  $[\Phi \wedge \Delta](|P|)$ . Next, we prove an intuitive result:

**Lemma 7.**  $\text{FIPS}(\Phi \wedge \Delta, |P|) \subseteq \text{FIPS}(\Phi \wedge \Delta, |Q|) \leftrightarrow |P| \leq |Q|$

*Proof.* We begin by showing that  $|P| \leq |Q| \rightarrow \text{FIPS}(\Phi \wedge \Delta, |P|) \subseteq \text{FIPS}(\Phi \wedge \Delta, |Q|)$ . Suppose  $|P| \leq |Q|$ , and then it follows that  $P \subseteq Q$  by Lemma 1. Then:

$$\begin{aligned} \text{FIPS}(\Phi \wedge \Delta, |P|) &= \{\phi(v_1, \dots, v_m) \wedge \delta(w_1, \dots, w_n) \mid v_1, \dots, v_m, w_1, \dots, w_n \in P\} \\ &\subseteq \{\phi(v_1, \dots, v_m) \wedge \delta(w_1, \dots, w_n) \mid v_1, \dots, v_m, w_1, \dots, w_n \in P\} \cup \\ &\quad \{\phi(v_1, \dots, v_m) \wedge \delta(w_1, \dots, w_n) \mid v_1, \dots, v_m, w_1, \dots, w_n \in Q \setminus P\} \\ &= \{\phi(v_1, \dots, v_m) \wedge \delta(w_1, \dots, w_n) \mid v_1, \dots, v_m, w_1, \dots, w_n \in Q\} \\ &= \text{FIPS}(\Phi \wedge \Delta, |Q|) \end{aligned}$$

Next we show that  $\text{FIPS}(\Phi \wedge \Delta, |P|) \subseteq \text{FIPS}(\Phi \wedge \Delta, |Q|) \rightarrow |P| \leq |Q|$ . Suppose that  $\text{FIPS}(\Phi \wedge \Delta, |P|) \subseteq \text{FIPS}(\Phi \wedge \Delta, |Q|)$ , then:

$$\begin{aligned} &\{\phi(v_1, \dots, v_m) \wedge \delta(w_1, \dots, w_n) \mid v_1, \dots, v_m, w_1, \dots, w_n \in P\} \\ &\subseteq \{\phi(v_1, \dots, v_m) \wedge \delta(w_1, \dots, w_n) \mid v_1, \dots, v_m, w_1, \dots, w_n \in Q\} \end{aligned}$$

Which implies that  $P \subseteq Q$ , which in turn implies that  $|P| \leq |Q|$  by Lemma 1. □

## 3 Intuition

We will build intuition by proving the M-N Theorem for small examples. Coming soon.

## 4 M-N Theorem

**Lemma 8** (FIP Saturation). Suppose that  $\Phi$  quantifies over  $m \in \mathbb{N}$  variables and  $\Delta$  quantifies over  $n \in \mathbb{N}$  variables. Then every FIP of  $[\Delta \wedge \Phi](k)$ , for  $k > m + n$ , has an equivalent FIP in  $[\Delta \wedge \Phi](m + n)$ .

*Proof.* Let  $k = |P| = m + n + z$  where  $z \in \mathbb{Z}_{>0}$ , and then  $P = \{1, \dots, k\}$  by Assumption 2. Let  $\phi \wedge \delta$  be an arbitrary FIP of  $[\Phi \wedge \Delta](k)$ . Then, because  $[\Phi \wedge \Delta](k)$  quantifies over exactly  $m + n$  variables, there must be at least  $z$  elements in  $P$  that do not appear in  $\phi \wedge \delta$ . Let  $u \leq m + n$  be the number of elements of  $P$  that are used in  $\phi \wedge \delta$ , and let  $\{e_1, \dots, e_u\} \subseteq P$  be the set of elements that are used. Consider the following permutation:

$$g = \begin{pmatrix} e_1 & \dots & e_u \\ 1 & \dots & u \end{pmatrix}$$

Notice that  $g(\phi \wedge \delta)$  contains only the elements  $1 \dots u$ . Since  $u \leq m + n$ , it must be the case that  $g(\phi \wedge \delta)$  is a FIP of  $[\Phi \wedge \Delta](m + n)$ . □

**Theorem 1** (M-N). Suppose that  $\Phi$  quantifies over  $m \in \mathbb{N}$  variables and  $\Delta$  quantifies over  $n \in \mathbb{N}$  variables. Then  $\Phi(P)$  is an inductive invariant for  $T(P)$  iff it is an inductive invariant for the finite instantiation  $T(m+n)$ .

*Proof.* It is clear that if  $\Phi(P)$  is an inductive invariant, then it must be an inductive invariant for  $T(m+n)$ . We prove the opposite direction in the remainder of the proof.

First, consider the case when the finite instantiation is less than or equal to  $m+n$ . Suppose that  $[\Phi \wedge \Delta](m+n) \rightarrow \Phi(m+n)'$ . Let  $k \leq m+n$ , then we must show that  $[\Phi \wedge \Delta](k) \rightarrow \Phi(k)'$ . Consider an arbitrary FIP  $\phi \wedge \delta$  of  $[\Phi \wedge \Delta](k)$ . By Lemma 7,  $\phi \wedge \delta$  is also a FIP of  $[\Phi \wedge \Delta](m+n)$ , and therefore it follows that  $\phi'$  holds.

We will now focus on the case when the finite instantiation is larger than  $m+n$ . Suppose that  $[\Phi \wedge \Delta](m+n) \rightarrow \Phi(m+n)'$ . Let  $k > m+n$ , then we must show that  $[\Phi \wedge \Delta](k) \rightarrow \Phi(k)'$ . Consider an arbitrary FIP  $\phi \wedge \delta$  of  $[\Phi \wedge \Delta](k)$ . By Lemma 8, there exists a FIP  $\phi_2 \wedge \delta_2$  of  $[\Phi \wedge \Delta](m+n)$  such that  $\phi_2 \wedge \delta_2 \equiv \phi \wedge \delta$ . By Lemma 6, we see that  $\phi_2 \wedge \delta_2$  holds because  $\phi \wedge \delta$  holds. However, because  $\phi_2 \wedge \delta_2$  holds, and we know that  $[\Phi \wedge \Delta](m+n) \rightarrow \Phi(m+n)'$ , it follows that  $\phi_2'$  holds. Now Lemma 3 implies that  $\phi \equiv \phi_2$ , and Lemma 4 implies that  $\phi' \equiv \phi_2'$ , so we have  $\phi_2'$  and  $\phi' \equiv \phi_2'$ . Thus by Lemma 5 and Lemma 2, we can conclude that  $\phi'$  holds.  $\square$

## 5 Case Studies

In this section we visit several (more coming soon) distributed protocols that are parameterized by a single sort and adhere to the property of Assumption 1.

### 5.1 Peterson's Mutex Protocol

Peterson's Mutex Protocol can be encoded with a transition function  $\Delta$  in PNF that quantifies over two variables. A sample inductive invariant candidate is given in [1] that quantifies of two variables and works for  $|P| = 2$ :

```
Phi == \A p,q \in ProcSet :
  /\ pc[p] \in {"a3","a4","cs"} => flag[p]
  /\ (p#q /\ pc[p] = "cs" /\ pc[q] = "a4") => turn = p
  /\ (p # q) => ~(pc[p] = "cs" /\ pc[q] = "cs")
```

However, by the M-N Theorem, we must show that  $\Phi$  is an inductive invariant for the case when  $|P| = 4$ . In fact, we easily see that  $\Phi$  fails to be inductive in the case:

$\wedge \text{turn} = 1$	$\wedge \text{turn} = 2$
$\wedge \text{pc}[1] = \text{"cs"}$	$\wedge \text{pc}[1] = \text{"cs"}$
$\wedge \text{pc}[2] = \text{"a4"}$	$\wedge \text{pc}[2] = \text{"a4"}$
$\wedge \text{pc}[3] = \text{"a3"}$	$\wedge \text{pc}[3] = \text{"a4"}$
$\wedge \text{a3}(3,2)$	

$\rightarrow$

This example uses states to describe the counterexample, but we can also describe it using the FIP  $M_\Phi(1,2) \wedge M_\Delta(3,2)$  from  $[\Phi \wedge \Delta](4)$ . When this FIP is true, both  $M_\Phi(1,2)$  and  $M_\Phi(1,3)$  fail to hold in the next state, showing that  $M_\Phi(1,2)$ —and hence  $\Phi$ —is not inductive.

This example shows how a FIP describes a specific relationship between  $\Phi$  and  $\Delta$ ; in this case the specific relationship leads to a counterexample. It is important to note that it is only possible to describe this particular counterexample using a FIP with a minimum of three elements in  $P$ , which is precisely why we do not detect the counter example in Peterson's Protocol when  $|P| = 2$ .

It is also worthwhile to note that we could derive the same counterexample using an equivalent FIP, say  $M_\Phi(3,2) \wedge M_\Delta(1,2)$ . This shows how FIP equivalency partitions a formula  $\Phi \wedge \Delta$  into classes of specific relationships that a transition system can exhibit.

## References

- [1] Parametric Peterson’s Mutex Protocol. [https://github.com/iandardik/iinf/blob/master/ii\\_cutoff/mn\\_thm/PetersonParametric.tla](https://github.com/iandardik/iinf/blob/master/ii_cutoff/mn_thm/PetersonParametric.tla), 2022.
- [2] 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.