

MODULE *ReachabilityProofs*

This module contains several lemmas about the operator *ReachableFrom* defined in module *Reachability*. Their proofs have been checked with the *TLAPS* proof system. The proofs contain comments explaining how such proofs are written.

Lemmas *Reachable1*, *Reachable2*, and *Reachable3* are used to prove correctness of the algorithm in module *Reachable*. Lemma *Reachable0* is used in the proof of lemmas *Reachable1* and *Reachable3*. You might want to read the proofs in module *Reachable* before reading any further.

All the lemmas except *Reachable1* are obvious consequences of the definition of *ReachableFrom*.

EXTENDS *Reachability*, *NaturalsInduction*

This lemma is quite trivial. It's a good warmup exercise in using *TLAPS* to reason about data structures.

LEMMA *Reachable0* \triangleq

$\forall S \in \text{SUBSET } \text{Nodes} :$

$\forall n \in S : n \in \text{ReachableFrom}(S)$

Applying the Decompose Proof command to the lemma generates the following statement.

$\langle 1 \rangle$ SUFFICES ASSUME NEW $S \in \text{SUBSET } \text{Nodes}$,

NEW $n \in S$

PROVE $n \in \text{ReachableFrom}(S)$

OBVIOUS

By definition of *Reachable*, we have to show that there exists a path from some node m in S to n . We obviously want to use n for m .

$\langle 1 \rangle 1$. *ExistsPath*(n, n)

To convince *TLAPS* that there exists a path from n to n , we have to give it the path. That path is obviously $\langle n \rangle$. A convenient way to tell *TLAPS* to use that path is with the statement:

$\langle 2 \rangle$ WITNESS $\langle n \rangle \in \text{Seq}(\text{Nodes})$

We can use this statement because the current goal is *ExistsPath*(n, n) which by definition of *ExistsPath* and *IsPathFromTo* equals $\exists p \in \text{Seq}(\text{Nodes}) : F(p)$, with the obvious meaning of $F(p)$. The body of this WITNESS statement is an abbreviation for:

$\langle 2 \rangle$ SUFFICES $F(\langle n \rangle)$

$\langle 3 \rangle 1$. $\langle n \rangle \in \text{Seq}(\text{Nodes})$

OBVIOUS

$\langle 3 \rangle 2$. QED

BY $\langle 3 \rangle 1$

The WITNESS statement takes no proof. Since correctness of the equivalent SUFFICES step depends on the definitions of *ExistsPath* and *IsPathFromTo*, we need to tell *TLAPS* to use those definitions by putting the following USE statement before the WITNESS step.

$\langle 2 \rangle$ USE DEF *ExistsPath*, *IsPathFromTo*

$\langle 2 \rangle$ WITNESS $\langle n \rangle \in \text{Seq}(\text{Nodes})$

$\langle 2 \rangle$ QED

OBVIOUS

$\langle 1 \rangle 2$. QED

PROOF BY $\langle 1 \rangle 1$ DEF *ReachableFrom*, *ExistsPath*

The following lemma lies at the heart of the correctness of the algorithm in module *Reachable*. The lemma is not obviously true. To write a proof that *TLAPS* can check, we need to start with an informal proof and then formalize that proof in TLA+. A mathematician should be able to devise an informal proof of this lemma in her head. Others will have to write it down. The informal proof that I came up with appears as comments placed at the appropriate points in the TLA+ proof. I devised the informal proof before I started writing the TLA+ proof. But it's easier to read that informal proof by using the higher levels of the TLA+ proof to give it the proper hierarchical structure. The best way to read the proof hierarchically is in the *Toolbox*, clicking on the little + and - icons beside a step to show and hide the step's proof. Start by executing the Hide Current Subtree command on the lemma.

LEMMA *Reachable1* \triangleq

$\forall S, T \in \text{SUBSET } \text{Nodes} :$
 $(\forall n \in S : \text{Succ}[n] \subseteq (S \cup T))$
 $\Rightarrow (S \cup \text{ReachableFrom}(T)) = \text{ReachableFrom}(S \cup T)$

An informal proof usually begins by implicitly assuming the following step.

(1) SUFFICES ASSUME NEW $S \in \text{SUBSET } \text{Nodes}$, NEW $T \in \text{SUBSET } \text{Nodes}$,
 $\forall n \in S : \text{Succ}[n] \subseteq (S \cup T)$
 PROVE $(S \cup \text{ReachableFrom}(T)) = \text{ReachableFrom}(S \cup T)$

OBVIOUS

The goal is that two sets are equal. The most common way to prove this is to prove that each set is a subset of the other.

(1)1. $(S \cup \text{ReachableFrom}(T)) \subseteq \text{ReachableFrom}(S \cup T)$

This is pretty obvious from the definitions. I realized that it follows immediately from two easily proved facts:

- $\text{ReachableFrom}(S \cup T) = \text{ReachableFrom}(S) \cup \text{ReachableFrom}(T)$
- $S \subseteq \text{ReachableFrom}(S)$

However, I tried to see if *TLAPS* could prove it more directly. It couldn't prove it directly from the definitions, but it could when I told it to first prove step (2)1. I then noticed that the same step occurred in the proof of lemma *Reachable3*, which I had already proved. (It's a good idea to prove the simplest theorems first.) So, I pulled that step and its proof out into lemma *Reachable0*.

(2)1. $\forall n \in S : n \in \text{ReachableFrom}(S)$

BY *Reachable0*

(2)2. QED

BY (2)1 DEF *ReachableFrom*

(1)2. $\text{ReachableFrom}(S \cup T) \subseteq (S \cup \text{ReachableFrom}(T))$

To prove that a set U is a subset of a set V , we prove that every element of U is an element of V . This is proved by letting n be any element of U and proving that it's an element of V . This leads to the following reduction of what has to be proved.

(2) SUFFICES ASSUME NEW $n \in \text{ReachableFrom}(S)$

PROVE $n \in S \cup \text{ReachableFrom}(T)$

BY DEF *ReachableFrom*

The assumption that n is in $\text{ReachableFrom}(S)$ tells us that there exists an element m in S and a path p from m to n . We need to prove that the existence of such an m and p implies that n is in S or in $\text{ReachableFrom}(T)$, using the assumption that $\text{succ}[m]$ is a subset of $S \cup T$ (which follows from the lemma's hypothesis).

A lot of thought convinced me that the only way of proving this is by induction. In general, there are many ways to reason by induction. For example, if S is a finite set, we can prove our goal by induction on S . However, there's no need to assume that S or T are finite. So, the obvious approach was then induction on the length of the path p . We can do that by defining

$$R(i) \triangleq \text{For any } m \text{ in } S \text{ and } q \text{ in } Nodes, \text{ if there is a path of length } i \text{ from } m \text{ to } q \text{ then } q \text{ is in } S \cup ReachableFrom(T)$$

and then proving that $R(i)$ holds for all positive integers by proving $R(1)$ and $R(i) \Rightarrow R(i+1)$. However, the *NaturalInductions* module contains an induction rule for proving a result about all natural numbers by proving it first for 0. So we define $R(i)$ as follows so that $R(0)$ is the assertion for paths of length 1.

(2) DEFINE $R(i) \triangleq$
 $\forall m \in S, q \in Nodes :$
 $(\exists p \in Seq(Nodes) : \wedge IsPathFromTo(p, m, q)$
 $\wedge Len(p) = i + 1)$
 $\Rightarrow (q \in S \cup ReachableFrom(T))$
 (2)1. $\forall i \in Nat : R(i)$

Level (3) is the obvious decomposition for an induction proof.

(3)1. $R(0)$

TLAPS has no problem proving this.

(4) SUFFICES ASSUME NEW $m \in S$, NEW $q \in Nodes$,
 NEW $p \in Seq(Nodes)$,
 $\wedge IsPathFromTo(p, m, q)$
 $\wedge Len(p) = 0 + 1$
 PROVE $q \in S \cup ReachableFrom(T)$

OBVIOUS

(4) QED

BY DEF $IsPathFromTo$

(3)2. ASSUME NEW $i \in Nat$, $R(i)$
 PROVE $R(i + 1)$

The proof of $R(i + 1)$ is decomposed as usual.

(4) SUFFICES ASSUME NEW $m \in S$, NEW $q \in Nodes$,
 NEW $p \in Seq(Nodes)$,
 $\wedge IsPathFromTo(p, m, q)$
 $\wedge Len(p) = (i + 1) + 1$
 PROVE $q \in S \cup ReachableFrom(T)$

BY DEF R

Since m is in S and $p[2]$ is in $Succ[m]$, the lemma's hypothesis implies that $p[2]$ is in $S \cup T$. The proof that q is in $S \cup ReachableFrom(T)$ is split into the two cases $p[2] \in S$ and $p[2] \in T$. If $p[2]$ is in S , then the result follows from the induction hypothesis, since $Tail(p)$ is a path of length $Len(p) - 1$ from an element of S to q . If $p[2]$ is in T , then $Tail(p)$ is a path from an element of T to q , so q is in $ReachableFrom(T)$.

Step (4)1 asserts some simple facts that I found were needed to get *TLAPS* to prove the first case. I then found they helped *TLAPS* prove the second case too, so I moved them before the case split.

$\langle 4 \rangle 1. \wedge Tail(p) \in Seq(Nodes)$
 $\wedge IsPathFromTo(Tail(p), p[2], q)$
 $\wedge Len(Tail(p)) = i + 1$
 BY DEF *IsPathFromTo*

This step isn't necessary because *TLAPS* can figure out that the two cases are exhaustive from the usable facts and the definition of *PathFromTo*, but I think it makes the proof easier to read.

$\langle 4 \rangle 2. p[2] \in S \cup T$
 BY DEF *IsPathFromTo*

TLAPS easily proves the two cases. However, it needs to be told to split the proof into cases because it's not good at figuring out by itself when to use a case split.

$\langle 4 \rangle 3. \text{CASE } p[2] \in S$
 BY $\langle 3 \rangle 2, \langle 4 \rangle 1, \langle 4 \rangle 3$
 $\langle 4 \rangle 4. \text{CASE } p[2] \in T$
 BY $\langle 4 \rangle 1, \langle 4 \rangle 4$ DEF *ReachableFrom, ExistsPath*
 $\langle 4 \rangle 5. \text{QED}$
 BY $\langle 4 \rangle 2, \langle 4 \rangle 3, \langle 4 \rangle 4$
 $\langle 3 \rangle \text{ HIDE DEF } R$
 $\langle 3 \rangle 3. \text{QED}$
 BY $\langle 3 \rangle 1, \langle 3 \rangle 2, \text{NatInduction}$

Proving $q \in S \cup ReachableFrom(T)$ from $\langle 2 \rangle 1$ is straightforward.

$\langle 2 \rangle 2. \text{PICK } m \in S, p \in Seq(Nodes) :$
 $IsPathFromTo(p, m, n)$
 BY DEF *ReachableFrom, ExistsPath*

We have to tell *TLAPS* to apply $\langle 2 \rangle 1$ with $i = Len(p) - 1$.

$\langle 2 \rangle 3. R(Len(p) - 1) \Rightarrow n \in S \cup ReachableFrom(T)$
 BY $\langle 2 \rangle 2$ DEF *IsPathFromTo*

Hiding the definition of *R* makes it easier for *TLAPS* to prove the result.

$\langle 2 \rangle \text{ HIDE DEF } R$

The definition of *IsPathFromTo* is needed for *TLAPS* to deduce $Len(p) > 0$, so $Len(p) - 1$ is in *Nat*.

$\langle 2 \rangle 4. \text{QED}$
 BY $\langle 2 \rangle 1, \langle 2 \rangle 2, \langle 2 \rangle 3$ DEF *IsPathFromTo*
 $\langle 1 \rangle 3. \text{QED}$
 BY $\langle 1 \rangle 1, \langle 1 \rangle 2$

The proof of this lemma is straightforward.

LEMMA *Reachable2* \triangleq

$\forall S \in \text{SUBSET } Nodes : \forall n \in S :$
 $\wedge ReachableFrom(S) = ReachableFrom(S \cup Succ[n])$
 $\wedge n \in ReachableFrom(S)$
 $\langle 1 \rangle \text{ SUFFICES ASSUME NEW } S \in \text{SUBSET } Nodes,$
 NEW $n \in S$
 PROVE $\wedge ReachableFrom(S) = ReachableFrom(S \cup Succ[n])$

$\wedge n \in \text{ReachableFrom}(S)$

OBVIOUS

$\langle 1 \rangle 1. \text{ReachableFrom}(S) = \text{ReachableFrom}(S \cup \text{Succ}[n])$

We decompose the proof of equality of two sets to proving the two subset relations.

$\langle 2 \rangle 1. \text{ReachableFrom}(S) \subseteq \text{ReachableFrom}(S \cup \text{Succ}[n])$

This subset relation is trivial because $S \subseteq T$ obviously implies $\text{ReachableFrom}(S) \subseteq \text{Reachable}(T)$

BY DEF *ReachableFrom*

$\langle 2 \rangle 2. \text{ReachableFrom}(S \cup \text{Succ}[n]) \subseteq \text{ReachableFrom}(S)$

We reduce the proof $U \subseteq V$ to proving that $u \in V$ for every u in U .

$\langle 3 \rangle$ SUFFICES $\text{ReachableFrom}(\text{Succ}[n]) \subseteq \text{ReachableFrom}(S)$

BY DEF *ReachableFrom*

$\langle 3 \rangle$ SUFFICES ASSUME NEW $m \in \text{Succ}[n]$, NEW $o \in \text{Nodes}$,

$\text{ExistsPath}(m, o)$

PROVE $\text{ExistsPath}(n, o)$

BY DEF *ReachableFrom*

$\langle 3 \rangle 1.$ PICK $p \in \text{Seq}(\text{Nodes}) : \text{IsPathFromTo}(p, m, o)$

BY DEF *ExistsPath*

$\langle 3 \rangle$ DEFINE $q \triangleq \langle n \rangle \circ p$

$\langle 3 \rangle 2. (q \in \text{Seq}(\text{Nodes})) \wedge \text{IsPathFromTo}(q, n, o)$

BY $\langle 3 \rangle 1, \text{SuccAssump}$ DEF *IsPathFromTo*

$\langle 3 \rangle 3.$ QED

BY $\langle 3 \rangle 2$ DEF *ExistsPath*

$\langle 2 \rangle 3.$ QED

BY $\langle 2 \rangle 1, \langle 2 \rangle 2$

Here's where we need *Reachable0*.

$\langle 1 \rangle 2. n \in \text{ReachableFrom}(S)$

BY *Reachable0*

$\langle 1 \rangle 3.$ QED

BY $\langle 1 \rangle 1, \langle 1 \rangle 2$

This lemma is quite obvious.

LEMMA *Reachable3* $\triangleq \text{ReachableFrom}(\{\}) = \{\}$

BY DEF *ExistsPath, ReachableFrom*

\ * Modification History
 \ * Last modified Sat Apr 13 18:07:57 PDT 2019 by lamport
 \ * Created Thu Apr 11 18:19:10 PDT 2019 by lamport