Verifying the Soundness of Resource Analysis for LogicGuard Monitors
Part 1*

Temur Kutsia  Wolfgang Schreiner
RISC, Johannes Kepler University Linz
{kutsia,schreine}@risc.jku.at
December 16, 2013

Abstract

In a companion paper (Wolfgang Schreiner, Temur Kutsia. A Resource Analysis for LogicGuard Monitors. RISC Technical report, December 5, 2013) we described a static analysis to determine whether a specification expressed in the LogicGuard language gives rise to a monitor that can operate with a finite amount of resources, notably with finite histories of the streams that are monitored. Here we prove the soundness of the analysis with respect to a formal operational semantics. The analysis is presented for an abstract core language that monitors a single stream.

Contents

1 Introduction 2
2 The Core Language and Resource Analysis 2
3 Operational Semantics 5
4 Soundness of Resource Analysis 9
5 Conclusion 13
A Proofs 14
A.1 Theorem 1: Soundness Theorem . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
A.2 Proposition 1: The Invariant Statement . . . . . . . . . . . . . . . . . . . . . . . . 24
A.3 Lemma 1: Soundness Lemma for Formulas . . . . . . . . . . . . . . . . . . . . 34
A.4 Lemma 2: Equivalence of Left- and Right-Recursive Definitions of n-Step Reductions 44
A.5 Lemma 3: History Cut-Off Lemma . . . . . . . . . . . . . . . . . . . . . . . . . . . . 51
A.6 Lemma 4: n-Step Reductions to done Formulas for TN, TCS, TCP . . . . . . . 64
A.7 Lemma 5: Soundness Lemma for Universal Formulas . . . . . . . . . . . . . . . . 89
A.8 Lemma 6: Monotonicity of Reduction to done . . . . . . . . . . . . . . . . . . . . . 90
A.9 Lemma 7: Shifting Lemma . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 96
A.10 Lemma 8: Triangular Reduction Lemma . . . . . . . . . . . . . . . . . . . . . . . 97

*The project “LogicGuard: The Efficient Checking of Time-Quantified Logic Formulas with Applications in Computer Security” is sponsored by the FFG BRIDGE program, project No. 832207.
1 Introduction

The goal of the LogicGuard project is to investigate to what extent classical predicate logic formulas are suitable as the basis for the specification and efficient runtime verification of system runs. The specific focus of the project is on computer and network security, concentrating on predicate logic specifications of security properties of network traffic. Properties are expressed by quantified formulas interpreted over sequences of messages; the quantified variable denotes a position in the sequence. Using the ordering of stream positions and nested quantification, complex properties can be formulated. Furthermore, to raise the level of abstraction, a higher-level stream may be constructed from a lower-level stream by a notation analogous to classical set builders. A translator generates from the specification an executable monitor.

The main ideas of these developments have been presented in [4] and [3]; in [1], the syntax and semantics of (an early abstract form of) the specification language are given; in [2], the translation of a specification to an executable monitor is described. A prototype of the translator and of the corresponding runtime system have been implemented and are operational.

The current implementation assumes that the whole “history” of a stream is preserved, i.e., that all received messages are stored in memory; thus the memory requirements of a monitor continuously grow. In practice, however, we are only interested in monitors that operate for an indefinite amount of time within a bounded amount of memory.

In [5], we tried to fill this gap by presenting a static analysis that

1. is able to determine whether a given specification can be monitored with a finite amount of history (and that may consequently generate a warning/error message, if not) and that

2. generates corresponding information in an easily accessible form such that after each execution step the runtime system of the monitor may appropriately prune the histories of the streams on which it operates.

One part of [5] was devoted to presenting the main ideas of the analysis by an abstract core language, which is only a skeleton of the real language; in particular it only monitors a single stream and does not support the construction of virtual streams. In this report, we use this language to formalize the operational semantics of the monitor execution and prove the soundness of the analysis presented in this report with respect to that semantics.

This paper is organized as follows: In Sect. 2 we briefly recall the definitions of the core language and the resource analysis from [5]. In Sect. 3 the operational semantics of the core language is described. In Sect. 4 the main result is formulated: soundness of the resource analysis with respect to the operational semantics. This section contains also all the lemmas needed for proving the soundness theorem. The detailed proof of one of the lemmas (Lemma 5) is the subject of the second, forthcoming part of this report. All the other proofs can be found in the Appendix.

2 The Core Language and Resource Analysis

The core language is depicted in Figure 1.

```
M ::= monitor X : F  
F ::= @X | ~F | F1 && F2 | F1 \(\land\) F2 | forall X in B1..B2 : F
B ::= 0 | infinity | X | B + N | B - N
N ::= 0 | 1 | 2 | ... 
X ::= x | y | z | ...
```

Figure 1: The Core Language
A specification in the core language describes a single monitor that controls a single stream of Boolean values where the atomic predicate \( @X \) denotes the value on the stream at the position \( X \), \( \neg X \) denotes negation, \( F_1 \& F_2 \) denotes sequential conjunction (the evaluation of \( F_2 \) is delayed until the value of \( F_1 \) becomes available), \( F_1 \lor F_2 \) describes parallel evaluation (both formulas are evaluated simultaneously until becomes false or both become true) and \( \forall X \in B_1 \ldots B_2 : F \) evaluates \( F \) at all positions in the range denoted by the interval \( B_1 \ldots B_2 \) until one instance becomes false or all instances have become true; the creation of a new instance \( F[n] \) is triggered by the arrival of the message number \( n \) on the stream.

This language is interpreted over a single stream of messages carrying truth values. We assume that a monitor \( M \) in this language is executed as follows: whenever a new message arrives on the stream, an instance \( F[p/X] \) of the monitor body \( F \) is created where \( p \) denotes the position of the message in the stream. All instances are evaluated on every subsequently arriving message which may or may not let the instance evaluate to a definite truth value; whenever an instance evaluates to such a value, this instance is discarded from the set; the positions of instances with negative truth values are reported as “violations” of the monitor.

A formula \( F \) in a monitor instance is evaluated as follows:

- the predicate \( @X \) is immediately evaluated to the truth value of the message at position \( X \) of the stream (see below for further explanation);
- \( \neg F \) first evaluates \( F \) and then negates the result;
- \( F_1 \& F_2 \) first evaluates \( F_1 \) and, if the result is true, then also evaluates \( F_2 \);
- \( F_1 \lor F_2 \) evaluates both \( F_1 \) and \( F_2 \) “in parallel” until the value of one subformula determines the value of the total formula;
- \( \forall X \in B_1 \ldots B_2 : F \) first determines the bounds of the position interval \( [B_1, B_2] \); it then creates for every position \( p \) in the interval, as soon as the messages in the stream reach that position, an instance \( F[p/X] \) of the formula body. All instances are evaluated on the subsequently arriving messages until all instances have been evaluated to “true” (and no more instances are to be generated) or some instance has been evaluated to “false”.

We assume that the monitoring formula \( M \) is closed, i.e., every occurrence of a position variable \( X \) in it is bound by a quantifier \texttt{monitor} or \texttt{forall}. Since by the evaluation strategies for these quantifiers, a formula instance is created only when the messages have reached the position assigned to the quantified variable, every occurrence of predicate \( @X \) can be immediately evaluated without delay.

We are interested in determining bounds for the resources used by the monitor, i.e., in particular in the following questions:

1. From the position where a monitor instance is created, how many “look-back” positions are required to evaluate the formula? This value determines the size of the “history” of past messages that have to be preserved in an implementation of the monitor.
2. How many instances can be active at the same time? This value determines the size that has to be reserved for the set of instances in the implementation of the monitor.

The basic idea for the analysis is a sort of “abstract interpretation” of the monitor where in a top-down fashion every position variable \( X \) is annotated as \( X^{(l,u)} \) where the interval \( [p+l, p+u] \) denotes those positions that the variables can have in relation to the position \( p \) of the “current” message of the stream; in a bottom up step, we then annotate every formula \( F \) with a pair \((h,d)\) where \( h \) is (an upper bound of) the size of the “history” (the number of past messages) required for the evaluation of \( F \) and \( d \) is (an upper bound of) the number of future messages that may be required such that the evaluation of \( F \) may be “delayed” by this number of steps.

The basic idea is formalized in Figures 2 and 3 by a rule system with three kinds of judgements:

3
⊢ \mathit{M} : \mathbb{N}^\infty \times \mathbb{N}^\infty \quad \mathit{Environment} \vdash \mathit{F} : \mathbb{N}^\infty \times \mathbb{N}^\infty \quad \mathit{Environment} \vdash \mathit{B} : \mathbb{Z}^\infty \times \mathbb{Z}^\infty

\frac{[[X]] \mapsto (0,0) \vdash \mathit{F} : (h,d)}{\vdash (\text{monitor } X : \mathit{F}) : (h,d)}

\begin{align*}
\vdash \mathit{a}X : (0,0) & \quad \vdash \mathit{~F} : (h,d) \\
\vdash \mathit{F}_1 : (h_1,d_1), \vdash \mathit{F}_2 : (h_2,d_2) & \quad \vdash \mathit{F}_1 \& \& \mathit{F}_2 : (\max^\infty(h_1,h_2 + \infty d_1), \max^\infty(d_1,d_2)) \\
\vdash \mathit{F}_1 : (h_1,d_1), \vdash \mathit{F}_2 : (h_2,d_2) & \quad \vdash \mathit{F}_1 \wedge \mathit{F}_2 : (\max^\infty(h_1,h_2), \max^\infty(d_1,d_2)) \\
\vdash \mathit{B}_1 : (l_1,u_1), \vdash \mathit{B}_2 : (l_2,u_2) & \quad \vdash \mathit{B} : (l_1,u_1) \vdash \mathit{F} : (h',d') \\
& \quad \vdash \mathit{B}_1 \ldots \mathit{B}_2 : (h,d) \\
\vdash \mathit{X} : (0,0) & \quad \vdash \mathit{X} : (0,0) \\
\vdash \mathit{X} \notin \text{domain}(e) & \quad \vdash \mathit{X} : e[[X]] \\
\vdash \mathit{B} : (l,u) & \quad \vdash \mathit{B} : (l,u) \\
\vdash \mathit{B} \ldots \mathit{B} : (l + \infty \llbracket N \rrbracket, u + \infty \llbracket N \rrbracket) & \quad \vdash \mathit{B} : (l,u) \\
\vdash \mathit{~B} : (l,u) & \quad \vdash \mathit{~B} : (l,u) \\
\vdash \mathit{B} : (l,u) & \quad \vdash \mathit{~B} : (l,u) \\
\vdash \mathit{~B} : (l,u) & \quad \vdash \mathit{~B} : (l,u)
\end{align*}

Figure 2: The Analysis of the Core Language

\begin{align*}
\mathit{Environment} & := \text{Variable} \rightarrow \mathbb{Z}^\infty \times \mathbb{Z}^\infty \\
\mathbb{N}^\infty & := \mathbb{N} \cup \{\infty\}, \mathbb{Z}^\infty := \mathbb{Z} \cup \{\infty, -\infty\} \\
\max^\infty : \mathbb{N} \times \mathbb{N}^\infty & \rightarrow \mathbb{N}^\infty \\
\max^\infty(n_1,n_2) & := \begin{cases} n_2 = \infty & \text{then } \infty \\ \text{else} \max(n_1,n_2) \end{cases} \\
+(\infty : \mathbb{N}^\infty \times \mathbb{N}^\infty) & \rightarrow \mathbb{N}^\infty \\
n_1 + \infty n_2 & := \begin{cases} n_1 = \infty \vee n_2 = \infty & \text{then } \infty \\ \text{else} n_1 + n_2 \end{cases} \\
-\infty : \mathbb{N}^\infty \times \mathbb{N} & \rightarrow \mathbb{N}^\infty \\
n_1 - \infty n_2 & := \begin{cases} n_1 = \infty & \text{then } \infty \\ \text{else} \max(0,n_1 - n_2) \end{cases} \\
-\infty : \mathbb{Z}^\infty & \rightarrow \mathbb{Z}^\infty \\
-\infty i & := \begin{cases} i = \infty & \text{then } -\infty \\ \text{else if } i = -\infty & \text{then } \infty \\ \text{else} -i \end{cases} \\
\mathbb{N} : \mathbb{Z}^\infty & \rightarrow \mathbb{N}^\infty \\
\mathbb{N}(i) & := \begin{cases} i = -\infty \vee i < 0 & \text{then } 0 \\ \text{else} i \end{cases}
\end{align*}

Figure 3: The Semantic Algebras of the Analysis
• $\vdash M : (h, d)$ states that the evaluation of the denoted monitor requires at most $h$ messages from the past of the stream and at most $d$ old monitor instances.

• $e \vdash F : (h, d)$ states that the evaluation of formula $F$ requires at most $h$ messages from the past of the stream and at most $d$ messages from the future of the stream. $e$ denotes a partial mapping of variables to pairs $(l, u)$ denoting the lower bound and upper bound of the interval relative to the position of the “current” message.

• $e \vdash B : (l, u)$ determines the lower bound $l$ and upper bound $u$ for the position denoted by an interval bound $B$.

We have $(h, d) \in \mathbb{N}^\infty \times \mathbb{N}^\infty$ where $\mathbb{N}^\infty = \mathbb{N} \cup \{\infty\}$; a value of $\infty$ indicates that the corresponding resource (history-instance set) cannot be bounded by the analysis. We have $e(X) \in \mathbb{Z}^\infty \times \mathbb{Z}^\infty$ where $\mathbb{Z}^\infty = \mathbb{Z} \cup \{\infty, -\infty\}$; a value of $\infty$ respectively $-\infty$ indicates that the position cannot be bounded from above respectively from below by the analysis. We have $(l, u) \in \mathbb{Z}^\infty \times \mathbb{Z}^\infty$; a value of $\infty$ for $u$ indicates that the corresponding interval has no upper bound; a value of $-\infty$ for $l$ indicates that the interval has no lower bound.

In [5] one can find more detailed illustration of the resource analysis, based on examples.

3 Operational Semantics

In this section we describe formalization of the operational interpretation of a monitor by a translation $T : \text{Monitor} \rightarrow T\text{Monitor}$ from the abstract syntax domain $\text{Monitor}$ to a domain $T\text{Monitor}$ denoting the runtime representation of the monitor. First, we list the domains used in the formalization, together with their definitions:

$$T\text{Monitor} := \text{TM of Variable} \times T\text{Formula} \times \mathcal{P}(T\text{Instance})$$

$$T\text{Instance} := \mathbb{N} \times T\text{Formula} \times \text{Context}$$

$$\text{Context} := (\text{Variable} \rightarrow \mathbb{N}) \times (\text{Variable} \rightarrow \text{Message})$$

$$T\text{Formula} := \text{done of Bool} \mid \text{next of TFormulaCore}$$

$$T\text{FormulaCore} :=$$

$$\text{TV of Variable} \mid$$

$$\text{TN of TFormula} \mid$$

$$\text{TCS of TFormula} \times T\text{Formula} \mid$$

$$\text{TCP of TFormula} \times T\text{Formula} \mid$$

$$\text{TA of Variable} \times \text{BoundValue} \times \text{BoundValue} \times T\text{Formula} \mid$$

$$\text{TA0 of Variable} \times \mathbb{N} \times \mathbb{N}^\infty \times T\text{Formula} \mid$$

$$\text{TA1 of Variable} \times \mathbb{N}^\infty \times T\text{Formula} \times \mathcal{P}(T\text{Instance})$$

$$\text{BoundValue} := \text{Context} \rightarrow \mathbb{N}^\infty$$

Translation. The translation is defined for monitors, formulas, and bounds. Monitors are translated into $T\text{Monitor}$’s (translated monitors), formulas are translated into $T\text{Formula}$’s (translated formulas), and bounds are translated into $\text{BoundValue}$’s:

$$T : \text{Monitor} \rightarrow T\text{Monitor}$$

$$T(\text{monitor } X : F) := \text{TM}(X, T(F), \emptyset)$$

$$T : \text{Formula} \rightarrow T\text{Formula}$$

$$T(@X) := \text{next}(TV(X))$$

$$T(\sim F) := \text{next}(TN(T(F)))$$
\[
T(F_1 \& F_2) := \text{next}(TCS(T(F_1), T(F_2))) \\
T(F_1 \lor F_2) := \text{next}(TCP(T(F_1), T(F_2))) \\
T(\text{forall } X \text{ in } B_1 \ldots B_2 : F) := \text{next}(TA(X, T(B_1), T(B_2), T(F)))
\]

\[T : \text{Bound} \rightarrow \text{BoundValue}\]
\[T(0)(c) := 0\]
\[T(\infty)(c) := \infty\]
\[T(X)(c) := c.1(X)\]
\[T(B + N)(c) := T(B)(c) + \left\lfloor N \right\rfloor\]
\[T(B + N)(c) := T(B)(c) - \left\lfloor N \right\rfloor\]

**One-Step Operational Semantics.** Apart from the quantified position variable \(X\) and the translation \(f = T(F)\) of the body of this monitor, the representation maintains the set \(fs\) of instances of \(f\) which for certain values of \(X\) could not yet be evaluated to a truth value. The execution of the monitor is formalized by an operational semantics with a small step transition relation \(\rightarrow_{n, ms, m, rs}\) where \(n\) is the index of the next message \(m\) arriving on the stream, \(ms\) denotes the sequence of messages that have previously arrived (the stream history), and \(rs\) denotes the set of those positions for which it can be determined by the current step that they violate the specification. In this step, first a new instance mapping \(X\) to the pair \((n, m)\) is created and added to the instance set and all instances in this set are evaluated; \(rs\) becomes the set of positions of those instances yielding “false”, the new instance set \(fs_1\) preserves all those instances that could not yet be evaluated to a definite truth value:

\[
T\text{Monitor} \rightarrow_{n, Message^\omega, Message, P(\text{nat})} T\text{Monitor}
\]
\[fs_0 = fs \cup \{(n, f, [X \mapsto (n, m)])\}\]
\[rs = \{n \in \mathbb{N} \mid \exists g \in T\text{Formula}, c \in \text{Context} : (n, g, c) \in fs_0 \land \]
\[\vdash g \rightarrow_{n, ms, m, c} \text{done(false)}\}
\[fs_1 = \{(n, g_0, c) \in T\text{Instance} \mid \exists g \in T\text{Formula} : (n, g, c) \in fs_0 \land \]
\[\vdash g \rightarrow_{n, ms, m, c} \text{next}(g_1)\}
\[TM(X, f, fs) \rightarrow_{n, ms, m, rs} TM(X, f, fs_1)\]

As one can see from this definition, the monitor operation is based on an operational semantics of formula evaluation. The rules for the latter are given below:

\[
T\text{Formula} \rightarrow_{n, Message^\omega, Message, Context} T\text{Formula}
\]

Atomic formula:
\[
X \in \text{dom}(c, 2) \\
\text{next}(TV(X)) \rightarrow_{(p, ms, m, c)} \text{done}(c.2(X))
\]
\[
X \notin \text{dom}(c, 2) \\
\text{next}(TV(X)) \rightarrow_{(p, ms, m, c)} \text{done}(\text{false})
\]

Negation:
\[
f \rightarrow_{(p, ms, m, c)} \text{next}(f') \\
\text{next}(TN(f)) \rightarrow_{(p, ms, m, c)} \text{next}(TN(\text{next}(f')))
\]
\[
f \rightarrow_{(p, ms, m, c)} \text{done}(\text{true}) \\
\text{next}(TN(f)) \rightarrow_{(p, ms, m, c)} \text{next}(TN(\text{done}(\text{false}))
\]
\[
f \rightarrow_{(p, ms, m, c)} \text{done}(\text{false}) \\
\text{next}(TN(f)) \rightarrow_{(p, ms, m, c)} \text{next}(TN(\text{done}(\text{true})))
\]
Sequential Conjunction:

\[ f_1 \rightarrow_{\text{TCP}(f_1, f_2)} \text{next}(f'_1) \]
\[ \text{next}(\text{TCP}(f_1, f_2)) \rightarrow_{(p, m, s, c)} \text{next}(\text{TCP}(\text{next}(f'_1), f_2)) \]

\[ f_1 \rightarrow_{(p, m, s, c)} \text{done}(\text{false}) \]
\[ \text{next}(f'_1) \rightarrow_{(p, m, s, c)} \text{done}(\text{false}) \]

Parallel Conjunction:

\[ f_1 \rightarrow_{(p, m, s, c)} \text{next}(f'_1) \]
\[ f_2 \rightarrow_{(p, m, s, c)} \text{next}(f'_2) \]
\[ \text{next}(\text{TCP}(f_1, f_2)) \rightarrow_{(p, m, s, c)} \text{next}(\text{TCP}(\text{next}(f'_1), \text{next}(f'_2))) \]

Universal Quantification:

\[ p_1 = b_1(c) \]
\[ p_1 = \infty \]
\[ \text{next}(\text{TA}(X, b_1, b_2, f)) \rightarrow_{(p, m, s, c)} \text{done}(\text{true}) \]

\[ p_1 = b_1(c) \]
\[ p_2 = b_2(c) \]
\[ p_1 \neq \infty \]
\[ \text{next}(\text{TA}(X, p_1, p_2, f)) \rightarrow_{(p, m, s, c)} \text{TA} \]
\[ \text{next}(\text{TA}(X, b_1, b_2, f)) \rightarrow_{(p, m, s, c)} \text{TA} \]

\[ p < p_1 \]
\[ \text{next}(\text{TA}(X, p_1, p_2, f)) \rightarrow_{(p, m, s, c)} \text{next}(\text{TA}(X, p_1, p_2, f)) \]

\[ p \geq p_1 \]
\[ f_s = \{(p_0, f, (c.1[X \mapsto p_0], c.2[X \mapsto ms(p_0 + p - |ms|)])) | p_1 \leq p_0 < \infty \min_{\infty}(p, p_2 + \infty 1)\} \]
\[ \text{next}(\text{TA}(X, p_1, p_2, f_s)) \rightarrow_{(p, m, s, c)} \text{TA} \]
\[ \text{next}(\text{TA}(X, p_1, p_2, f_s)) \rightarrow_{(p, m, s, c)} \text{TA} \]
\begin{align*}
fs_0 &= \text{if } p >^\infty p_2 \text{ then } \fs \text{ else } \fs \cup \{(p, f, c.1[X \mapsto p], c.2[X \mapsto m])\} \\
\exists t \in \mathbb{N}, g \in TFormula, c \in \text{Context} : (t, g, c) \in \fs_0 \land g \nrightarrow (p, ms, m, c) & \text{ done(false)} \\
\text{next}(TAI(X, p_2, f, \fs)) & \rightarrow (p, ms, m, c) \text{ done(false)} \\
\nofs_0 &= \text{if } p >^\infty p_2 \text{ then } \ofs \text{ else } \ofs \cup \{(p, f, c.1[X \mapsto p], c.2[X \mapsto m])\} \\
\exists t \in \mathbb{N}, g \in TFormula, c \in \text{Context} : (t, g, c) \in \ofs_0 \land g \nrightarrow (p, ms, m, c) & \text{ done(false)} \\
\offs_1 &= \{(t, \text{next}(fc), c) \in TInstance | 2g \in TFormula : (t, g, c) \in \ofs_0 \land g \nrightarrow (p, ms, m, c) \text{ next}(fc)\} \\
\text{next}(TAI(X, p_2, f, \fs)) & \rightarrow (p, ms, m, c) \text{ done(true)} \\
\offs_1 &= \emptyset \land p >^\infty p_2 \\
\text{next}(TAI(X, p_2, f, \fs)) & \rightarrow (p, ms, m, c) \text{ next}(TAI(X, p_2, f, \fs_1)) \text{ done(true)} \\
\end{align*}

Finally, we give definitions of \(n\)-step reduction. There are for versions: right- and left-recursive with and without history.

**Definition 1** (Right-Recursive \(n\)-Step Reduction).

**Without history.** \(TFormula \rightarrow^*_r(\infty, \mathbb{N}, \text{Stream}, \text{Environment}) TFormula\), where the first \(\mathbb{N}\) is the number of steps and the second \(\mathbb{N}\) is the current position.

\[
\begin{align*}
Ft & \rightarrow^*_r(0, p, s, c) Ft \\
& \quad n > 0 \\
& \quad c = (e, \{(X, s(e(X))) \mid X \in \text{dom}(e)\}) \\
& \quad Ft \rightarrow (p, s, p, s(p), c) Ft' \\
& \quad Ft' \rightarrow (n-1, p+1, s, c, s|h) Ft'' \\
& \quad \text{next}(Ft) \rightarrow (n, p, s, c) Ft'' \\
\end{align*}
\]

**With history.** \(TFormula \rightarrow^*_r(\infty, \mathbb{N}, \text{Stream}, \text{Environment}, \text{Message}^*) TFormula\), where the first \(\mathbb{N}\) is the number of steps, the second \(\mathbb{N}\) is the current position, and \(\text{Message}^*\) is the history.

\[
\begin{align*}
Ft & \rightarrow^*_r(0, p, s, c, h) Ft \\
& \quad n > 0 \\
& \quad c = (e, \{(X, s(e(X))) \mid X \in \text{dom}(e)\}) \\
& \quad Ft \rightarrow (p, s, p, s(p), c) Ft' \\
& \quad Ft' \rightarrow (n-1, p+1, s, c, s|h) Ft'' \\
& \quad \text{next}(Ft) \rightarrow (n, p, s, c, h) Ft'' \\
\end{align*}
\]

**Definition 2** (Left-Recursive \(n\)-Step Reduction).

**Without history.** \(TFormula \rightarrow^*_l(\infty, \mathbb{N}, \text{Stream}, \text{Environment}) TFormula\), where the first \(\mathbb{N}\) is the number of steps and the second \(\mathbb{N}\) is the current position.

\[
\begin{align*}
Ft & \rightarrow^*_l(0, p, s, c) Ft \\
& \quad n > 0 \\
& \quad Ft \rightarrow (n-1, p, s, c) Ft' \\
& \quad c = (e, \{(X, s(e(X))) \mid X \in \text{dom}(e)\}) \\
& \quad Ft' \rightarrow (p+n-1, s(p+n-1), s(p), c) Ft'' \\
& \quad \text{next}(Ft) \rightarrow (n, p, s, c) Ft'' \\
\end{align*}
\]

**With history.** \(TFormula \rightarrow^*_l(\infty, \mathbb{N}, \text{Stream}, \text{Environment}, \text{Message}^*) TFormula\), where the first \(\mathbb{N}\) is the
number of steps, the second \( N \) is the current position, and \( \text{Message}^* \) is the history.

\[
\begin{align*}
Ft \rightarrow &^{\mathcal{I}_{(0,p,s,e,h)}} Ft \\
n > 0 \quad &Ft \rightarrow^*_{(n-1,p,s,e,h)} Ft' \\
c = (e, \{(X, s(e(X))) \mid X \in \text{dom}(e)\}) \quad &Ft' \rightarrow_{(p+n-1,s,(\max(0,p+n-1-h),\min(p+n-1,h)),s(p+n-1),c)} Ft'' \\
Ft \rightarrow &^*_{(n,p,s,e,h)} Ft''
\end{align*}
\]

4 Soundness of Resource Analysis

In this section we formulate the main result:

**Theorem 1** (Soundness of Resource Analysis for Monitors). The resource analysis of the core monitor language is sound with respect to its operational semantics, i.e., if the analysis yields for monitor \( M \) natural numbers \( h \) and \( d \), then the execution does not maintain more than \( d \) monitor instances and does not require more than the last \( h \) messages from the stream. Formally:

\[
\forall M \in \text{Monitor}, Mt \in \text{TMonitor}, n \in \mathbb{N}, s \in \text{Message}^\omega, rs \in \mathcal{P}(\mathbb{N}), d, h \in \mathbb{N}^\omega : \\
\vdash M : (h, d) \\
\Rightarrow \\
(d \in \mathbb{N} \Rightarrow (\vdash T(M) \rightarrow^{*}_{n,s,rs} Mt \Rightarrow |\text{instances}(Mt)| \leq d)) \land \\
(h \in \mathbb{N} \Rightarrow (\vdash T(M) \rightarrow^{*}_{n,s,rs} Mt \Leftrightarrow \vdash T(M) \rightarrow^{*}_{n,s,rs,h} Mt))
\]

where \( \text{instances}(TM(X,f,fs)) := fs \)

The proof of this theorem uses three lemmas and a statement about an invariant of \( n \)-step reductions of translated monitors. These propositions, for their part, rely on four additional lemmas. Dependencies between these statements, which give an idea of the high-level proof structure, are shown in Fig. 4. Below we formulate these lemmas with some informal explanations. The complete proofs can be found in the appendix.1

Figure 4: Lemma dependencies in the proof of the Soundness Theorem. The triangle indicates the pending proof.

The Invariant Statement asserts essentially the following: For a monitor \( M \) (with the monitoring variable \( X \) and the monitored formula \( F \)), if the analysis yields natural numbers \( h \) and \( d \), and the translated version of \( M \) reduces to another translated monitor \( TM(Y, Ft, It) \) in \( n \) steps, then the following invariant holds:

1At the time of writing this report, the proof of the Statement 4 of Lemma 4 is not finished.
\begin{itemize}
  \item $X$ and $Y$ are the same and $Ft$ is the translation of $F$,
  \item all elements in the set of instances $It$ contain \textit{next} formulas, which have been generated at different steps in the past, but not earlier than $d$ units before from the current step,
  \item the formulas in the elements of $It$ are obtained by reductions of $T(F)$, and they themselves will reduce to a \textit{done} formula in at most $d$ steps from the moment of their creation.
\end{itemize}

More formally, the invariant definition looks as follows:

\textbf{Definition 3 (Invariant).}

\[
\forall X, Y \in \text{Variable}, F \in \text{Formula}, Ft \in T\text{Formula}, It \in \mathbb{P}(T\text{Instance}),
\begin{align*}
&n \in \mathbb{N}, s \in \text{Stream}, d \in \mathbb{N}^\infty : \\
&\text{invariant}(X, Y, F, Ft, It, n, s, d) : \Leftrightarrow \\
&T = Y \land Ft = T(F) \land \text{alldiff}(It) \land \text{allnext}(It) \land \\
&\forall t \in \mathbb{N}, Ft' \in T\text{Formula}, c \in \text{Context} : \\
&(t, Ft', c) \in It \land d \in \mathbb{N} \Rightarrow \\
&c.1 = \{(X, t)\} \land c.2 = \{(X, s(t))\} \land \\
&n - d \leq t \leq n - 1 \land \\
&T(F) \rightarrow^{n-t, t, s, c.1} Ft' \land \\
&\exists b \in \text{Bool}, d' \in \mathbb{N} : \\
&d' \leq d \land \vdash Ft' \rightarrow^{\text{max}(0, t+d'-n), n, s, c.1} \text{done}(b),
\end{align*}
\]

where \text{alldiff}(It) means that $t_1 \neq t_2$ for all distinct elements $(t_1, Ft_1, c_1)$, $(t_2, Ft_2, c_2)$ of $It$, and \text{allnext}(It) denotes the fact that for all $(t, Ft, c) \in It$, $Ft$ is a \textit{next} formula.

Then the Invariant Statement is formulated in the following way:

\textbf{Proposition 1 (Invariant Statement).}

\[
\forall X \in \text{Variable}, F \in \text{Formula}, h \in \mathbb{N}^\infty, d \in \mathbb{N}^\infty, n \in \mathbb{N}, s \in \text{Stream},
\begin{align*}
&rs \in \mathbb{P}(N), Y \in \text{Variable}, Ft \in T\text{Formula}, It \in \mathbb{P}(T\text{Instance}) : \\
&\vdash (\text{monitor } X : F) : (h, d) \land \\
&\vdash T(\text{monitor } X : F) \rightarrow^{n, s, rs} T\text{Instance}(Y, Ft, It) \Rightarrow \\
&\text{invariant}(X, Y, F, Ft, It, n, s, d)
\end{align*}
\]

In the course of proving the Soundness Statement, the reasoning moves from the monitor level to the formula level. Therefore, we need a counterpart of the Soundness Theorem (which is formulated for monitors) for formulas. This is the first Lemma.

\textbf{Lemma 1 (Soundness Lemma for Formulas).}

\[
\forall F, F' \in \text{Formula}, re \in \text{RangeEnv}, e \in \text{Environment}, Ft \in T\text{Formula}, n, p \in \mathbb{N},
\begin{align*}
&s \in \text{Stream}, d \in \mathbb{N}^\infty, h \in \mathbb{N} : \\
&\vdash (re \vdash F : (h, d)) \land \\
&\forall Y \in \text{dom}(e) : re(Y).1 + p \leq e(Y) \leq re(Y).2 + p \Rightarrow \\
&(d \in \mathbb{N} \Rightarrow \\
&\exists b \in \text{Bool}, d' \in \mathbb{N} : \\
&d' \leq d + 1 \land \vdash T(F) \rightarrow^{n, p, s, e} \text{done}(b) \land \\
&(\forall h' \in \mathbb{N} : h' \geq h \Rightarrow \\
&(T(F) \rightarrow^{n, p, s, c} Ft \Leftrightarrow T(F) \rightarrow^{n, p, s, c, h'} Ft)).
\end{align*}
\]

The second lemma states equivalence of left- and right-recursive definitions of $n$-step reductions. This is a technical result which helps to simplify proofs of the Soundness Theorem, Invariant Statement, and in Lemma 4 below.
Lemma 2 (Equivalence of Left- and Right-Recursive Definitions of n-Step Reductions).

(a) \( \forall n, p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, F_1, F_2 \in T\text{Formula} : \)
\[ F_1 \xrightarrow{\ast}_{n, p, s, e} F_2 \Leftrightarrow F_1 \xrightarrow{\ast}_{n, p, s, e} F_2. \]

(b) \( \forall n, p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, F_1, F_2 \in T\text{Formula}, h \in \mathbb{N} : \)
\[ F_1 \xrightarrow{\ast}_{n, p, s, e, h} F_2 \Leftrightarrow F_1 \xrightarrow{\ast}_{n, p, s, e, h} F_2. \]

The next lemma establishes the limit on the number of past messages needed for a single monitoring step to be equivalent to such a step performed with the full history. Both the Soundness Theorem and the Soundness Lemma use it.

Lemma 3 (History Cut-Off Lemma).

\[ \forall F \in \text{Formula}, F_1 \in T\text{Formula}, p, q \in \mathbb{N}, s \in \text{Stream}, h, d \in \mathbb{N}, e \in \text{Environment}, re \in \text{RangeEnv} : \]
\[ \text{let } c := (e, \{ (X, s(e(X))) \mid X \in \text{dom}(e) \}) : \]
\[ \vdash (re \vdash F : (h, d)) \land \]
\[ q \leq p \land \forall Y \in \text{dom}(e) : re(Y).1 + q \leq e(Y) \leq re(Y).2 + q \Rightarrow \]
\[ \forall h' \in \mathbb{N} : h' \geq h \Rightarrow \]
\[ T(F) \xrightarrow{\ast}_{p, s} (p, s(p), c) F_1 \]
\[ \Leftrightarrow \]
\[ T(F) \xrightarrow{\ast}_{p, s} \max(0, p - h'), \min(p, h'), s(p), c F_1 \]

The Soundness Lemma requires yet two auxiliary propositions. The first of them, Lemma 4 below, establishes the conditions of reduction of translated TN (negation), TCS (sequential conjunction), and TCP (parallel conjunction) formulas into done formulas:

Lemma 4 (n-Step Reductions to done Formulas for TN, TCS, TCP).

Statement 1. TN Formulas:

\[ \forall F \in \text{Formula}, n, p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, F_1 \in T\text{Formula} : \]
\[ T(F) \xrightarrow{\ast}_{n, p, s, e} \text{done}(\text{false}) \Rightarrow \text{next}(T(N(T(F)))) \xrightarrow{\ast}_{n, p, s, e} \text{done}(\text{true}) \land \]
\[ T(F) \xrightarrow{\ast}_{n, p, s, e} \text{done}(\text{true}) \Rightarrow \text{next}(T(N(T(F)))) \xrightarrow{\ast}_{n, p, s, e} \text{done}(\text{false}) \]

Statement 2. TCS Formulas:

\[ \forall p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment} : \]
\[ \forall F_1, F_2 \in T\text{Formula}, n \in \mathbb{N} : \]
\[ n > 0 \land F_1 \xrightarrow{\ast}_{n, p, s, e} \text{done}(\text{false}) \Rightarrow \]
\[ \text{next}(T(CS(F_1, F_2))) \xrightarrow{\ast}_{n, p, s, e} \text{done}(\text{false}) \land \]
\[ \forall F_1, F_2 \in T\text{Formula}, n_1, n_2 \in \mathbb{N}, b \in \text{Bool} : \]
\[ n_1 > 0 \land n_2 > 0 \land F_1 \xrightarrow{\ast}_{n_1, p, s, e} \text{done}(\text{true}) \land F_2 \xrightarrow{\ast}_{n_2, p, s, e} \text{done}(b) \Rightarrow \]
\[ \text{next}(T(CS(F_1, F_2))) \xrightarrow{\ast}_{\max(n_1, n_2), p, s, e} \text{done}(b) \]
Statement 3. TCP Formulas:

\[ \forall p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, Ft_1, Ft_2 \in TFormula, n_1, n_2 \in \mathbb{N} : \]

\[ n_1 > 0 \land Ft_1 \rightarrow_{n_1, p, s, c}^{\text{done}(\text{false})} \land Ft_2 \rightarrow_{n_2, p, s, c}^{\text{done}(\text{true})} \Rightarrow \]

\[ \text{next}(TCP(Ft_1, Ft_2)) \rightarrow_{n_1, p, s, c}^{\text{done}(\text{false})} \]

\[ \land \]

\[ n_1 > 0 \land n_2 > 0 \land Ft_1 \rightarrow_{n_1, p, s, c}^{\text{done}(\text{false})} \land Ft_2 \rightarrow_{n_2, p, s, c}^{\text{done}(\text{false})} \Rightarrow \]

\[ \text{next}(TCP(Ft_1, Ft_2)) \rightarrow_{\min(n_1, n_2), p, s, c}^{\text{done}(\text{false})} \]

\[ \land \]

\[ n_1 > 0 \land n_2 > 0 \land Ft_1 \rightarrow_{n_1, p, s, c}^{\text{done}(\text{true})} \land Ft_2 \rightarrow_{n_2, p, s, c}^{\text{done}(\text{true})} \Rightarrow \]

\[ \text{next}(TCP(Ft_1, Ft_2)) \rightarrow_{\max(n_1, n_2), p, s, c}^{\text{done}(\text{true})} \]

\[ \land \]

\[ n_1 > 0 \land n_2 > 0 \land Ft_1 \rightarrow_{n_1, p, s, c}^{\text{done}(\text{true})} \land Ft_2 \rightarrow_{n_2, p, s, c}^{\text{done}(\text{false})} \Rightarrow \]

\[ \text{next}(TCP(Ft_1, Ft_2)) \rightarrow_{n_2, p, s, c}^{\text{done}(\text{false})} \]

The other auxiliary statement needed in the proof of Lemma 1 is Lemma 5 below, which formulates a special case of the soundness statement for universally quantified formulas. Its proof will be given in the forthcoming second part of this report.

Lemma 5 (Soundness Lemma for Universal Formulas).

\[ \forall F \in \text{Formula}, X \in \text{Variable}, B_1, B_2 \in \text{Bound} : \]

\[ R(F) \Rightarrow R(\forall X \in B_1 . . . B_2 : F) \]

where

\[ R(F) : \Leftrightarrow \]

\[ \forall re \in \text{RangeEnv}, e \in \text{Environment}, s \in \text{Stream}, d \in \mathbb{N}^\infty, h \in \mathbb{N} : \]

\[ \vdash (re \vdash F : (h, d)) \land d \in \mathbb{N} \land \]

\[ \forall Y \in \text{dom}(e) : re(Y).1 + p \leq e(Y) \leq re(Y).2 + p \Rightarrow \]

\[ (\forall p \in \mathbb{N}, \exists b \in \text{Bool}, d' \in \mathbb{N} : d' \leq d + 1 \land \vdash (F \rightarrow_b \rightarrow_{d', p, s, c}^{\text{done}(b)}) \]

Proving of Lemma 4 requires a couple of other statements. Besides Lemma 2 above, there are two other lemmas: for monotonicity and for shifting. The Monotonicity Lemma states that if a translated formula reduces to a done formula, then starting from that moment on it will always reduce to the same done formula:

Lemma 6 (Monotonicity of Reduction to done).

\[ \forall Ft \in TFormula, p, k \in \mathbb{N}, s \in \text{Stream}, c \in \text{Context}, b \in \text{Bool} : \]

\[ k \geq p \Rightarrow \]

\[ Ft \rightarrow_{p,s,(p),s(p),c}^{\text{done}(b)} \Rightarrow Ft \rightarrow_{k,s,(k),s(k),c}^{\text{done}(b)}. \]

The Shifting Lemma expresses a simple fact: If a next formula reduced to a done formula in \( n + 1 \) steps starting from the stream position \( p \), then the same reduction will take \( n \) steps if it starts at position \( p + 1 \):

Lemma 7 (Shifting Lemma).

\[ \forall f \in TFormulaCore, n, p \in \mathbb{N}, s \in \text{Stream}, c \in \text{Environment}, b \in \text{Bool} : \]

\[ n > 0 \Rightarrow \]

\[ \text{next}(f) \rightarrow_{n+1, p, s, c}^{\text{done}(b)} \Rightarrow \text{next}(f) \rightarrow_{n+p+1, s, c}^{\text{done}(b)}. \]

Lemma 7 requires a so called Triangular Reduction Lemma, shown below. The latter, for itself, relies on Lemma 6.
Lemma 8 (Triangular Reduction Lemma).

\[ \forall f_1, f_2 \in T_{FormulaCore}, Ft \in T_{Formula}, p \in \mathbb{N}, s \in Stream, c \in Context : \]
\[ \text{next}(f_1) \rightarrow_{p,s\downarrow(p),s(p),c} \text{next}(f_2) \land \text{next}(f_2) \rightarrow_{p+1,s\downarrow(p+1),s(p+1),c} Ft \Rightarrow \]
\[ \text{next}(f_1) \rightarrow_{p+1,s\downarrow(p+1),s(p+1),c} Ft. \]

5 Conclusion

The goal of resource analysis of the core LogicGuard language is two-fold: To determine the maximal size of the stream history required to decide a given instance of the monitor formula, and to determine the maximal delay in deciding a given instance. Ultimately, it determines whether a specification expressed in this language gives rise to a monitor that can operate with a finite amount of resources. This report presents propositions needed to prove soundness of resource analysis of the core LogicGuard language with respect to the operational semantics.

Acknowledgments

The authors thank the project partner companies: SecureGuard GmbH and RISC Software GmbH.

References


A Proofs

A.1 Theorem 1: Soundness Theorem

Soundness Theorem for Monitors:

∀ X ∈ Variable, F ∈ Formula, h ∈ N, d ∈ N, n ∈ N, s ∈ Stream, rs ∈ P(N),

Y ∈ Variable Ft ∈ TFormula, It ∈ P(Instance):

let M = monitor X : F, Mt = TM(Y,Ft,It) :

⊢ M: (h,d) ⇒

(d ∈ N ⇒ (∀ T(M) → *(n,s,rs) Mt ⇒ |It| ≤ d)) ∧

(h ∈ N ⇒ (∀ T(M) → *(n,s,rs) Mt ⇔ ⊢ T(M) → *(n,s,rs,h) Mt))

PROOF:

------

We split the soundness statement into two formulas:

(a) ∀ X ∈ Variable, F ∈ Formula, h ∈ N, d ∈ N, n ∈ N, s ∈ Stream, rs ∈ P(N),

Y ∈ Variable Ft ∈ TFormula, It ∈ P(Instance):

let M = monitor X : F, Mt = TM(Y,Ft,It) :

⊢ M: (h,d) ⇒

(d ∈ N ⇒ (∀ T(M) → *(n,s,rs) Mt ⇒ |It| ≤ d))

and

(b) ∀ X ∈ Variable, F ∈ Formula, h ∈ N, d ∈ N, n ∈ N, s ∈ Stream, rs ∈ P(N),

Y ∈ Variable Ft ∈ TFormula, It ∈ P(Instance):

let M = monitor X : F, Mt = TM(Y,Ft,It) :

⊢ M: (h,d) ⇒

(h ∈ N ⇒ (∀ T(M) → *(n,s,rs) Mt ⇔ ⊢ T(M) → *(n,s,rs,h) Mt))

Proof of (a)

---------------

We take Xf, Ff, Yf, Ft, It, hf, df, nf, sf, rsf arbitrary but fixed.

Assume

(1) ⊢ (monitor Xf : Ff): (hf,df)
(2) df ∈ N
(3) T(monitor Xf : Ff) → *(nf, sf, rsf) TM(Yf, Ft, Itf)

Prove

[4] |Itf| ≤ df

From (1,2,3), we know that

(5) invariant(Xf, Yf, Ff, Itf, nf, sf, df)

holds. That means, we know

(6) Xf = Yf
\[ F_{tf} = T(F_f) \]
\[ \text{alldiffs}(I_{tf}) \]
\[ \text{allnext}(I_{tf}) \]
\[ \forall t \in \mathbb{N}, F_t \in \text{TFormula}, c \in \text{Context}: \]
\[ (t,F_t,c) \in I_{tf} \Rightarrow \]
\[ \{X_f, t\} \land \{X_f, sf(t)\} \land \]
\[ T(F_f) \rightarrow \*) (n-t, t, s, c.1) F_t1 \land \]
\[ nf-df \leq t \leq nf-1 \land \]
\[ \exists b \in \text{Bool} \exists d' \in \mathbb{N} : \]
\[ d' \leq df \land \vdash F_t \rightarrow \*) (\max(0, t+df'-nf), nf, sf, c.1) \text{done}(b) \]

From (10), we know that the tags of the elements of \( I_{tf} \) are between \( nf-df \) and \( nf-1 \) inclusive. From (8), we know that no two elements of \( I_{tf} \) have the same tag. Hence, \( I_{tf} \) can contain at most \( (nf-1)-(nf-df)+1 = df \) elements. Hence, (5) holds.

Proof of (b)
---------------

Parametrization:

\[ Q(n) : \]
\[ \forall X \in \text{Variable}, F \in \text{Formula}, h \in \mathbb{N}_\infty, d \in \mathbb{N}_\infty, s \in \text{Stream}, rs \in \mathbb{P}(\mathbb{N}), \]
\[ Y \in \text{Variable}, F_t \in \text{TFormula}, I_t \in \mathbb{P}(\text{Instance}) : \]
\[ \text{let } M = \text{monitor } X : F, M_t = \text{TM}(Y,F_t,I_t) : \]
\[ \vdash M : (h,d) \Rightarrow \]
\[ (h \in \mathbb{N} \Rightarrow (\vdash T(M) \rightarrow \*) (n,s,rs) M_t \iff \vdash T(M) \rightarrow \*) (n,s,rs,h) M_t) \]

We want to show

\[ \forall n \in \mathbb{N} : Q(n). \]

For this is suffices to show

1. \( Q(0) \)
2. \( \forall n \in \mathbb{N} : Q(n) \Rightarrow Q(n+1) \)

Proof of 1
----------

\[ Q(0) \]
\[ \forall X \in \text{Variable}, F \in \text{Formula}, h \in \mathbb{N}_\infty, d \in \mathbb{N}_\infty, s \in \text{Stream}, rs \in \mathbb{P}(\mathbb{N}), \]
\[ Y \in \text{Variable}, F_t \in \text{TFormula}, I_t \in \mathbb{P}(\text{Instance}) : \]
\[ \text{let } M = \text{monitor } X : F, M_t = \text{TM}(Y,F_t,I_t) : \]
\[ \vdash M : (h,d) \Rightarrow \]
\[ (h \in \mathbb{N} \Rightarrow (\vdash T(M) \rightarrow \*) (0,s,rs) M_t \iff \vdash T(M) \rightarrow \*) (0,s,rs,h) M_t) \]

We take \( X_f, F_f, Y_f, F_{tf}, c_f, I_{tf}, df, hf, sf, rs_f \) arbitrary but fixed.

Assume
(1) ⊢ (monitor X_f : F_f): (hf, df)
(2) hf ∈ N

Prove

[3] ⊢ T(monitor X_f : F_f) →∗(0, sf, rsf) TM(Y_f, F_t_f, I_t_f) ⇔
    ⊢ T(monitor X_f : F_f) →∗(0, sf, rsf, hf) TM(Y_f, F_t_f, I_t_f)

Direction (⇒). Assume

(4) ⊢ T(monitor X_f : F_f) →∗(0, sf, rsf) TM(Y_f, F_t_f, I_t_f)

Prove

[5] ⊢ T(monitor X_f : F_f) →∗(0, sf, rsf, hf) TM(Y_f, F_t_f, I_t_f)

From (4), by the def. of →∗(0, sf, rsf), we get

(6) T(monitor X_f : F_f) = TM(Y_f, F_t_f, I_t_f).

and

(7) rsf = ∅.

From (6, 7) and the def. of →∗(0, sf, rsf, hf) we obtain [5].

Direction (⇐) can be proved analogously.

Hence, Q(0) holds.

========

Proof of 2
----------

Take arbitrary n ∈ N.

Assume Q(n), i.e.,

(1) ∀ X ∈ Variable, F ∈ Formula, h ∈ N∞, d ∈ N∞, s ∈ Stream, rs ∈ P(N),
    Y ∈ Variable, F_t ∈ TFormula, I_t ∈ P(Instance):
    let M = monitor X : F, M_t = TM(Y, F_t, I_t):
    ⊢ M: (h, d) ⇒
    (h ∈ N ⇒ (¬ T(M) →∗(n, s, rs) M_t ⇔ ⊢ T(M) →∗(n, s, rs, h) M_t))

Prove Q(n+1), i.e.,

[2] ∀ X ∈ Variable, F ∈ Formula, h ∈ N∞, d ∈ N∞, s ∈ Stream, rs ∈ P(N),
    Y ∈ Variable, F_t ∈ TFormula, I_t ∈ P(Instance):
    let M = monitor X : F, M_t = TM(Y, F_t, I_t):
    ⊢ M: (h, d) ⇒
    (h ∈ N ⇒ (¬ T(M) →∗(n+1, s, rs) M_t ⇔ ⊢ T(M) →∗(n+1, s, rs, h) M_t))
We take $X_f, F_f, h_f, d_f, s_f, r_s, Y_f, F_{tf}, I_{tf}$ arbitrary but fixed.

Assume

(3) $\vdash (\text{monitor } X_f : F_f) : (h_f, d_f)$
(4) $h_f \in \mathbb{N}$

and prove

[5] $\vdash T(\text{monitor } X_f : F_f) \rightarrow (n+1, s_f, r_s) \ TM(Y_f, F_{tf}, I_{tf})$ $\iff$
$\vdash T(\text{monitor } X_f : F_f) \rightarrow (n+1, s_f, r_s, h_f) \ TM(Y_f, F_{tf}, I_{tf})$

To prove (5), we need to prove

[5.1]
$\vdash T(\text{monitor } X_f : F_f) \rightarrow (n+1, s_f, r_s) \ TM(Y_f, F_{tf}, I_{tf}) \Rightarrow$
$\vdash T(\text{monitor } X_f : F_f) \rightarrow (n+1, s_f, r_s, h_f) \ TM(Y_f, F_{tf}, I_{tf})$.

and

[5.2]
$\vdash T(\text{monitor } X_f : F_f) \rightarrow (n+1, s_f, r_s, h_f) \ TM(Y_f, F_{tf}, I_{tf}) \Rightarrow$
$\vdash T(\text{monitor } X_f : F_f) \rightarrow (n+1, s_f, r_s) \ TM(Y_f, F_{tf}, I_{tf})$.

Proof of [5.1]

----------

Since $T(\text{monitor } X_f : F_f) = \text{TM}(X_f, T(F_f), \emptyset)$, we assume

(6) $\vdash \text{TM}(X_f, T(F_f), \emptyset) \rightarrow (n+1, s_f, r_s) \ TM(Y_f, F_{tf}, I_{tf})$

and prove

[7] $\vdash \text{TM}(X_f, T(F_f), \emptyset) \rightarrow (n+1, s_f, r_s, h_f) \ TM(Y_f, F_{tf}, I_{tf})$.

From (3) and (6), by the invariant statement, we know

(8) $Y_f = X_f, F_{tf} = T(F_f)$

From (6) by the definition of $\rightarrow$ we know that there exist $Y', F_t', I_t'$, $r_s'$ and $r_s''$ such that

(9) $r_s = r_s' \cup r_s''$
(10) $\vdash \text{TM}(X_f, T(F_f), \emptyset) \rightarrow (n, s_f, r_s') \ TM(Y', F_t', I_t')$
(11) $\vdash \text{TM}(Y', F_t', I_t') \rightarrow (n, s_f(n), s_f(n), r_s'') \ TM(X_f, T(F_f), I_{tf})$

From (10), by the definition of $\rightarrow$, (and by the invariant) we have

(12) $Y' = X_f, F_t' = T(F_f)$.

From (10), by (1,3,4), and (12) we get

(13) $\vdash \text{TM}(X_f, T(F_f), \emptyset) \rightarrow (n, s_f, r_s', h_f) \ TM(X_f, T(F_f), I_{tf})$.
From (11) by (12) we have

(14) ⊢ TM(Xf, T(F), I') → (n, sf↓(n), sf(n), rs2') TM(Xf, T(F), If)

From (14), by definition of → for TMonitors we know

(15) rs2' = { t ∈ N | $\exists g \in TFormula, c \in Context: (t, g, c) \in I' \land$
$\vdash g \rightarrow (n, sf↓(n), sf(n), c) \ done(false) }$

(16) Itf = { (t, g1, c) ∈ TInstance | $\exists g \in TFormula: (t, g, c) \in I' \land$
$\vdash g \rightarrow (n, sf↓(n), sf(n), c) \ next(g1) }$

where

(17) I't0 = I' \cup \{(n, T(F), \{(X, n)\}, \{X, sf(n)\})\}

To prove (7), by the definition of →* with h-cutoff for TMonitors, and (12), we need to prove that there exist Y*, F*, I*, rs1* and rs2* such that

(18) rs1*∪rs2*=rsf

(19) TM(Xf, T(F), ∅) →*(n, sf, rs1*, hf) TM(Y*, F*, I*)

(20) TM(Y*, F*, I*) →(n, s↑(max(0, n-hf), min(n, hf)), s(n), rs2*) TM(Xf, T(F), If).

We can take rs1*=rs1', rs2*=rs2', Y*=Xf, F*=F'=T(F), It*'=I'. Then (18) holds due to (9) and (19) holds due to (13). Hence, we need to prove only (20), which after instantiating the variables has the form

(21) TM(Xf, T(F), I') →(n, sf↑(max(0, n-hf), min(n, hf)), sf(n), rs2') TM(Xf, T(F), If).

By definition of → for TMonitors, to prove (21), we need to prove

[22] rs2' = { t ∈ N |
$\exists g \in TFormula, c \in Context: (t, g, c) \in I' \land$
$\vdash g \rightarrow (n, sf↑(max(0, n-hf), min(n, hf)), sf(n), c) \ done(false) }$

and

[23] Itf = { (t, g1, c) ∈ TInstance |
$\exists g \in TFormula: (t, g, c) \in I' \land$
$\vdash g \rightarrow (n, sf↑(max(0, n-hf), min(n, hf)), sf(n), c) \ next(g1) }$

where Itf0 is defined as in (17).

Hence, by (15) and [22], we need to prove

[24] { t ∈ N | $\exists g \in TFormula, c \in Context: (t, g, c) \in I' \land$
$\vdash g \rightarrow (n, sf↓(n), sf(n), c) \ done(false) }$

= { t ∈ N |
$\exists g \in TFormula, c \in Context: (t, g, c) \in I' \land$
$\vdash g \rightarrow (n, sf↑(max(0, n-hf), min(n, hf)), sf(n), c) \ done(false) }$

By (16) and [23], we need to prove
[25] \{ (t,g1,c) \in TInstance \mid \exists g \in TFormula: (t,g,c) \in I_0 \land \\
\vdash g \rightarrow (n,sf \downarrow (n),sf(n),c) \ next(g1) \} = \\
\{ (t,g1,c) \in TInstance \mid \\
\exists g \in TFormula: (t,g,c) \in I_0 \land \\
\vdash g \rightarrow (n,sf \uparrow (\max(0,n-hf),\min(n,hf)),sf(n),c) \ next(g1) \}

To prove [24], we need to show

[26] \forall t \in \mathbb{N} : \\
\exists g \in TFormula, c \in Context: \\
(t,g,c) \in I_0 \land \vdash g \rightarrow (n,sf \downarrow (n),sf(n),c) \ done(false) \\
\iff \\
\exists g \in TFormula, c \in Context: \\
(t,g,c) \in I_0 \land \vdash g \rightarrow (n,sf \uparrow (\max(0,n-hf),\min(n,hf)),sf(n),c) \ done(false).

To prove (25), we need to show

[27] \forall t \in \mathbb{N}, g1 \in TFormula, c \in Context \\
\exists g \in TFormula: \\
(t,g,c) \in I_0 \land \vdash g \rightarrow (n,sf \downarrow (n),sf(n),c) \ next(g1) \\
\iff \\
\exists g \in TFormula: \\
(t,g,c) \in I_0 \land \vdash g \rightarrow (n,sf \uparrow (\max(0,n-hf),\min(n,hf)),sf(n),c) \ next(g1).

Proof of [26, \Rightarrow].

-------------------

We take t0 arbitrary but fixed. Let g \in TFormula and c \in Context be such that

(26.1) (t0,g,c) \in I_0 and \\
(26.2) \vdash g \rightarrow (n,sf \downarrow (n),sf(n),c) \ done(false)

hold. We need to find g* \in TFormula and c* \in Context such that

[26.3] (t0,g*,c*) \in I_0 and \\
[26.4] \vdash g* \rightarrow (n,sf \uparrow (\max(0,n-hf),\min(n,hf)),sf(n),c*) \ done(false)

hold. We take g*=g and c*=c. Then (26.3) holds because of (26.1). Hence, we only need to prove

[26.4] \vdash g \rightarrow (n,sf \uparrow (\max(0,n-hf),\min(n,hf)),sf(n),c) \ done(false)

Since (t0,g,c) \in I_0, we have either

(26.5) (t0,g,c) \in I'_t, or \\
(26.6) t0=n, g=T(Ff), c=\{(Xf,n)\},\{Xf,sf(n)\}).

Let first consider the case (26.5).

-------------------

We had
From (3) and (10), by the invariant statement, we have

(26.7) invariant(Xf,Y',Ff,Ft',It',n,sf,df)

The invariant (26.7) implies

(12) Y'=Xf, Ft'=T(Ff)

and by (26.5) the following:

(26.8) T(Ff) \to^* (n-t0,t0,sf,c.1) g.

From (26.8), by Lemma 2 we get

(26.9) T(Ff) \to^* (n-t0,t0,sf,c.1) g.

From (26.5) and (26.7) we get

(26.10) c.1=\{(Xf,t0)\}, c.2=\{(X,sf(t0))\}=\{(X,sf(c.1(Xf)))\}

Since by the invariant n-t0+1>0, from (26.9), (26.2), (26.10), by the definition of \to^*l^*, we get

(26.11) T(Ff) \to^* (n-t0+1,t0,sf,c.1) done(false).

From (26.11), by Lemma 2, we get

(26.12) T(Ff) \to^* (n-t0+1,t0,sf,c.1) done(false).

From (3) by the definition of \vdash, there exists re0\in RangeEnv such

(26.13) re0 \vdash Ff: (hf, df) and
(26.14) re0(Xf) = (0,0)

From (26.10) and (26.14) the following is satisfied

(26.15) \forall Y\in \text{dom}(c.1): \text{re0}(Y).1+t0 \leq c.1(Y) \leq \text{re0}(Y).2+t0.

Hence, from (26.13), (26.15), (26.12) and the Statement 2 of Lemma 1 (taking F=Ff, re=re0, e=c.1, Ft=g, n=n-t0, p=t0, s=sf, d=df, h=h'=hf) we get

(26.16) T(Ff) \to^* (n-t0+1,t0,sf,c.1,hf) done(false).

From (26.16), by Lemma 2 we get

(26.17) T(Ff) \to^* (n-t0+1,t0,sf,c.1,hf) done(false).

Since by the invariant n-t0+1>0, from (26.17), by the definition of \to^*l^* with history, there exists Ft0\in TFormula such that
From (26.18), by Lemma 2, we get

(26.20) $T(F_f) \rightarrow^* (n-t_0,t_0,sf,c.1,hf) F_t_0$.

From (26.20), by (26.13), (26.15), and Statement 2 of Lemma 1 we get

(26.21) $T(F_f) \rightarrow^* (n-t_0,t_0,sf,c.1) F_t_0$.

From (26.21) and (26.8), since the rules for $\rightarrow$ are deterministic and $\rightarrow^*$ is defined based on $\rightarrow$, we conclude

(26.22) $F_t_0 = g$.

From (26.22) and (26.19), we get [26.4]

Now we consider the case (26.6):

-----------------------------

(26.6) $t_0 = n$, $g = T(F_f)$, $c = \{ (X_f,n) \}, \{ X_f,sf(n) \}$.

Under (26.6), the formula (26.2) now looks as

(26.23) $\vdash T(F_f) \rightarrow (n,sf\downarrow(n),sf(n),c) done(false)$

We need to prove [26.4], which, by (26.6) has the form

[26.24] $\vdash T(F_f) \rightarrow (n,sf\uparrow(max(0,n-hf),min(n,hf)),sf(n),\{(X,n)\},\{X,sf(n)\})) done(false)$

From (3) by the definition of $\vdash$, there exists $r_0 \in rangeEnv$ such

(26.25) $r_0 \vdash F_f: (hf, df)$ and
(26.26) $r_0(X_f) = (0,0)$

From (26.25) and (26.26) the following is satisfied

(26.27) $\forall Y \in dom(c.1): r_0(Y).1+n \leq c.1(Y) \leq r_0(Y).2+n$.

From (26.25), (26.27), the definition of $c$ in (26.6), and Lemma 5 (instantiating $F=F_f$, $F_t=done(false)$, $p=n$, $s=sf$, $h=h'=hf$, $d=df$, $e=c.1$) we get [26.24].

Proof of [26, $\Leftarrow$].

-----------------------------

The direction (\$\Leftarrow$) can proved analogously to the direction (\$\Rightarrow$). This is easy to see, because the proof of (\$\Leftarrow$) relies on Statement 2 of Lemma 1 and on Lemma 3. Both of these propositions assert equivalence between a formula expressed in the version of $\rightarrow^*$ (resp. $\rightarrow$) without history and a formula expressed in the version of $\rightarrow^*$ (resp. $\rightarrow$) with history. Hence, for proving [26, $\Rightarrow$] we can use Statement 2 of Lemma 1 and Lemma 3 in the direction opposite to the one used in the proof of [26, $\Leftarrow$].

21
Proof of [27]
-------------

Proof of [27] is analogous to the proof of [26]. This is easy to see, because [27] and [26] differ only with a TFormula in the right hand side of →*, and the proof of [26] does not depend on what stands in that side. Hence, we can replace done(false) in the proof of [26] with next(g1) and we obtain the proof of [27].

-------------

Proof of [5.2].
-------------

We assume

(28) ⊢ TM(Xf, T(Ff), ∅) →*(n+1, sf, rsf, hf) TM(Yf, Ftf, Itf)

and want to prove

[29] ⊢ TM(Xf, T(Ff), ∅) →*(n+1, sf, rsf) TM(Yf, Ftf, Itf).

From (28), by the definition of →* with cut-off for TMonitors, we know that there exist Yf', Ftf', Itf', rs1', rs2', such that

(30) rs1' ∪ rs2' = rsf
(31) ⊢ TM(Xf, T(Ff), ∅) →*(n, sf, rs1', hf) TM(Yf', Ftf', Itf') and
(32) TM(Yf', Ftf', Itf') →(n, sf↑(max(0, n-hf), min(n, hf)), sf(n), rs2') TM(Yf, Ftf, Itf)

From the definitions of →* and → we can see that Yf' = Xf, Ftf' = T(F).

To prove [29], by the definition of →* for TMonitors, we need to find such Yf*, Ftf*, Itf*, rs1*, and rs2* that

[33] rs1* ∪ rs2* = rsf
[34] ⊢ TM(Xf, T(F), ∅) →*(n, sf, rs1*) TM(Yf*, Ftf*, Itf*) and
[35] TM(Yf*, Ftf*, Itf*) →(n, sf↓n, sf(n), rs2*) TM(Xf, T(Ff), Itf)

We take Yf* = Xf, Ftf* = T(F), Itf* = Itf', rs1* = rs1', rs2* = rs2'. Then:

- [33] follows from (30).
- [34] follows from (31) by (3,4) and the induction hypothesis (1).

Hence, it is only left to prove the following instance of [35]:

[36] TM(Xf, T(Ff), Itf') →(n, sf↓n, sf(n), rs2') TM(Xf, T(Ff), Itf)

To show it, by the definition of → for TMonitors, we need to prove

[37] rs2' = { t ∈ N | 
            ∃ g ∈ TFormula, c ∈ Context: (t, g, c) ∈ It0 ∧ 
            g → (n, sf↓n, sf(n), c) done(false) }

22
and

\[ \text{Itf} = \{ (t,g1,c) \in T\text{Instance} \mid \exists g \in T\text{Formula}: (t,g,c) \in \text{It0} \land \vdash g \rightarrow (n,\text{sf} \downarrow n,\text{sf}(n),c) \text{ next}(g1) \} \]

where \( \text{It0} = \text{Itf}' \cup \{(n,\text{T(Ff)},\{(X,n)\},\{X,\text{sf}(n)\})\} \)

On the other hand, from (32) we know that

\[ \text{rs2'} = \{ t \in \mathbb{N} \mid \exists g \in T\text{Formula}, c \in \text{Context}: (t,g,c) \in \text{It0}' \land \vdash g \rightarrow (n,\text{sf} \uparrow (\max(0,n-hf),\min(n,hf)),\text{sf}(n),c) \text{ done(false) } \} \]

and

\[ \text{Itf} = \{ (t,g1,c) \in T\text{Instance} \mid \exists g \in T\text{Formula}: (t,g,c) \in \text{It0}' \land \vdash g \rightarrow (n,\text{sf} \uparrow (\max(0,n-hf),\min(n,hf)),\text{sf}(n),c) \text{ next}(g1) \} \]

where \( \text{It0}' \) is defined exactly as \( \text{It0} \): \( \text{It0}' = \text{It0} \).

Hence, by [37] and (39), we need to prove

\[ \{ t \in \mathbb{N} \mid \exists g \in T\text{Formula}, c \in \text{Context}: (t,g,c) \in \text{It0} \land \vdash g \rightarrow (n,\text{sf} \downarrow n,\text{sf}(n),c) \text{ done(false) } \} = \{ t \in \mathbb{N} \mid \exists g \in T\text{Formula}, c \in \text{Context}: (t,g,c) \in \text{It0}' \land \vdash g \rightarrow (n,\text{sf} \uparrow (\max(0,n-hf),\min(n,hf)),\text{sf}(n),c) \text{ done(false) } \} \]

But this is exactly [24] which we have already proved. Hence, [41] holds.

By (40) and [38], we need to prove

\[ \{ (t,g1,c) \in T\text{Instance} \mid \exists g \in T\text{Formula}: (t,g,c) \in \text{It0}' \land \vdash g \rightarrow (n,\text{sf} \downarrow n,\text{sf}(n),c) \text{ next}(g1) \} = \{ (t,g1,c) \in T\text{Instance} \mid \exists g \in T\text{Formula}: (t,g,c) \in \text{It0}' \land \vdash g \rightarrow (n,\text{sf} \uparrow (\max(0,n-hf),\min(n,hf)),\text{sf}(n),c) \text{ next}(g1) \} \]

But this is exactly [25] which we have already proved. Hence, [42] holds.

It means, we proved also [35]. It finished the proof of [5.2] and, hence, of the soundness theorem.
A.2 Proposition 1: The Invariant Statement

Invariant Statement
-------------------
\( \forall X \in \text{Variable}, F \in \text{Formula}, h \in \mathbb{N}^\infty, d \in \mathbb{N}^\infty, n \in \mathbb{N}, s \in \text{Stream}, rs \in \mathbb{P}(\mathbb{N}), Y \in \text{Variable}, Ft \in \text{TFormula}, It \in \mathbb{P}(\text{TInstance}) : \)
\( \vdash \text{monitor } X : F : (h,d) \land \)
\( \vdash \text{T(monitor } X : F ) \rightarrow *(n,s,rs) \text{ TM}(Y,Ft,It) \Rightarrow \)
\( \text{invariant}(X,Y,F,Ft,It,n,s,d) \)

PROOF
-----

Parameterization
-----------------
\( P(n) : \iff \)
\( \forall X \in \text{Variable}, F \in \text{Formula}, h \in \mathbb{N}^\infty, d \in \mathbb{N}^\infty, s \in \text{Stream}, rs \in \mathbb{P}(\mathbb{N}), Y \in \text{Variable}, Ft \in \text{TFormula}, c \in \text{Context}, It \in \mathbb{P}(\text{Instance}) : \)
\( \vdash \text{monitor } X : F : (h,d) \land \)
\( \vdash \text{T(monitor } X : F ) \rightarrow *(n,s,rs) \text{ TM}(Y,Ft,It) \Rightarrow \)
\( \text{invariant}(X,Y,F,Ft,It,n,s,d) \)

We want to show
\( \forall n \in \mathbb{N} : P(n) \)

For this it suffices to show
1. \( P(0) \)
2. \( \forall n \in \mathbb{N} : P(n) \Rightarrow P(n+1) \)

Proof of 1
--------

\( P(0) \)
\( \forall X \in \text{Variable}, F \in \text{Formula}, h \in \mathbb{N}^\infty, d \in \mathbb{N}^\infty, s \in \text{Stream}, rs \in \mathbb{P}(\mathbb{N}), Y \in \text{Variable}, Ft \in \text{TFormula}, c \in \text{Context}, It \in \mathbb{P}(\text{Instance}) : \)
\( \vdash \text{monitor } X : F : (h,d) \land \)
\( \vdash \text{T(monitor } X : F ) \rightarrow *(0,s,rs) \text{ TM}(Y,Ft,It) \Rightarrow \)
\( \text{invariant}(X,Y,F,Ft,It,0,s,d) \)

We take \( X_f,F_f,d_f,h_f,s_f,rs_f,Y_f,F_{tf},I_{tf} \) arbitrary but fixed.

Assume
\( (1) \vdash \text{monitor } X_f : F_f : (h_f,d_f) \)
\( // (2) d_f \in \mathbb{N} \)
\( (3) \text{T(monitor } X_f : F_f ) \rightarrow *(0,s_f,rs_f) \text{ TM}(Y_f,F_{tf},I_{tf}) \)

and show
\[ a \text{ invariant}(X_f,Y_f,F_f,F_{tf},I_{tf},0,s_f,d_f) \]
From (3) and def. \(\rightarrow^*\), we know

(4) \(r_{sf} = \emptyset\)
(5) \(T(\text{monitor } X_f : F_f) = TM(X_f,F_f,I_f)\)

From (5) and Def. of \(T(M)\), we know

(6) \(Y_f = X_f\)
(7) \(F_{tf} = T(F_f)\)
(8) \(I_{tf} = \emptyset\)

From (6,7,8) and the definitions of alldiff, allnext, and the invariant, we get [a].

==============================

Proof of 2
----------
\(\forall n \in \mathbb{N}: P(n) \Rightarrow P(n+1)\)

Take arbitrary \(n \in \mathbb{N}\).

Assume \(P(n)\), i.e.,

(1) \(\forall X \in \text{Variable}, F \in \text{Formula}, h \in \mathbb{N}^\infty, d \in \mathbb{N}^\infty, s \in \text{Stream}, rs \in \mathbb{P}(\mathbb{N})\),
\(Y \in \text{Variable}, Ft \in \text{TFormula}, It \in \mathbb{P}(\text{Instance})\):
\(\vdash (\text{monitor } X : F) : (h,d) \land \vdash T(\text{monitor } X : F) \rightarrow^* (n,s,rs) TM(Y,Ft,It) \Rightarrow \text{invariant}(X,Y,F,Ft,It,n,s,d)\)

Show \(P(n+1)\), i.e.,

(a) \(\forall X \in \text{Variable}, F \in \text{Formula}, h \in \mathbb{N}^\infty, d \in \mathbb{N}^\infty, s \in \text{Stream}, rs \in \mathbb{P}(\mathbb{N})\),
\(Y \in \text{Variable}, Ft \in \text{TFormula}, It \in \mathbb{P}(\text{Instance})\):
\(\vdash (\text{monitor } X : F) : (h,d) \land \vdash T(\text{monitor } X : F) \rightarrow^* (n+1,s,rs) TM(Y,Ft,It) \Rightarrow \text{invariant}(X,Y,F,Ft,It,n+1,s,d)\)

We take \(X_f,F_f,df,hf,rsf,Y_f,Ftf,Itf\) arbitrary but fixed.

Assume

(2) \(\vdash (\text{monitor } X_f : F_f) : (hf,df)\)
\(// (3) df \in \mathbb{N}\)
(4) \(T(\text{monitor } X_f : F_f) \rightarrow^* (n+1,rsf) TM(Y_f,Ftf,Itf)\)

and show

[b] \(\text{invariant}(X_f,Y_f,F_f,Ftf,Itf,n+1,rsf,df)\)

From (4) and def. \(\rightarrow^*\) for TMonitors, we know for some \(rs_1,rs_2\) and \(M_t = TM(X',F_t',I_t')\)

(5) \(\vdash T(\text{monitor } X_f : F_f) \rightarrow^* (n,rsf,rsf) TM(X',F_t',I_t')\)
(6) ⊢ TM(X',Ft',It') \rightarrow (n,sf \downarrow n,sf(n),rs2) TM(Yf,Ftf,Itf)
(7) rsf = rs1 \cup rs2

From (6) by the definition of $\rightarrow$ for TMonitors, we know

(8) $X' = Yf$,
(9) $Ft' = Ftf$, and
(10) $Itf = \{((t_0,\text{next}(Fc1),c_0) \in TInstance |$
\hspace{1cm} $\exists Ft_0 \in TFormula$ such that $(t_0,Ft_0,c_0) \in It_0$ and
\hspace{1cm} $\vdash Ft_0 \rightarrow (n,sf \downarrow n,sf(n),c_0) \text{next}(Fc1))\}"

where

(11) $It_0 = It' \cup \{(n,Ft,\{((Yf,n),\{(Yf,sf(n))\})\})\}$

From (1), for $X=Xf$, $F=Ff$, $h=hf$, $d=df$, $s=sf$, $rs=rs1$, $Y=Yf$, $Ft=Ftf$, and $It=It'$, we obtain

(12) $\vdash (\text{monitor } Xf : Ff) : (hf,df)$ $\land$
\hspace{1cm} $\vdash T(\text{monitor } Xf : Ff) \rightarrow (n,sf,rs1) TM(Yf,Ftf,It') \Rightarrow$
\hspace{1cm} invariant(Xf,Yf,Ff,Ftf,It',n,sf,df)

From (14,2,3,5,8,9) we obtain

(13) invariant(Xf,Yf,Ff,Ftf,It',n,sf,df)

It means, we know

(14) $Xf = Yf$
(15) $Ftf = T(Ff)$
(16) alldiffs(It')
(17) allnext(It')
(18) $\forall t \in \mathbb{N}, Ft \in TFormula, c \in \text{Context}$:
$\ (t,Ft,c) \in It' \land d \in \mathbb{N} \Rightarrow$
\hspace{1cm} $c.1=\{(Xf,t)\} \land c.2=\{(Xf,sf(t))\} \land$
\hspace{1cm} $n \land n \leq d \leq t \leq n-1 \land$
\hspace{1cm} $T(Ff) \rightarrow \star (n-t,t,sf,c.1) Ft \land$
\hspace{1cm} $\exists b \in \text{Bool} \exists d' \in \mathbb{N} :$
\hspace{1cm} $d' \leq d \land \vdash Ft \rightarrow \star (\max(0,t+d'-n-1),n+1,sf,c.1) \text{done}(b)$

Showing [b] means that we want to show

[b1] $Xf = Yf$
[b2] $Ftf = T(Ff)$
[b3] alldiffs(Itf)
[b4] allnext(Itf)
[b5] $\forall t \in \mathbb{N}, Ft \in TFormula, c \in \text{Context}$:
$\ (t,Ft,c) \in Itf \land d \in \mathbb{N} \Rightarrow$
\hspace{1cm} $c.1=\{(Xf,t)\} \land c.2=\{(Xf,sf(t))\} \land$
\hspace{1cm} $n \land n+1 \leq df \leq t \leq n \land$
\hspace{1cm} $T(Ff) \rightarrow \star (n+1-t,t,sf,c.1) Ft \land$
\hspace{1cm} $\exists b \in \text{Bool} \exists d' \in \mathbb{N} :$
\hspace{1cm} $d' \leq df \land \vdash Ft \rightarrow \star (\max(0,t+d'-n-1),n+1,sf,c.1) \text{done}(b)$

26
Proof of \([b_1]\)  
-------------  
\([b_1]\) is proved by (14).

Proof of \([b_2]\)  
-------------  
\([b_2]\) is proved by (15).

Proof of \([b_3]\)  
-------------  
From (10) one can see that the elements \((t,F_t,c)\) in \(I_{tf}\) inherit their tag \(t\) from \(I_{t0}\), which is \(I_t' \cup \{(n,F_{tf},(cp,cm))\}\). From (18) we know \(\text{alldiff}(I_t')\). From (18) we have \(t \leq n-1\) for all \((t,F_t,c) \in I_t'\). Adding \(\{(n,F_{tf},cf)\}\) to \(I_t'\), will guarantee all instances in \(I_{t0}\) have different tags. Since these tags are transferred to \(I_{tf}\), we conclude that \([b_3]\) holds.

Proof of \([b_4]\)  
-------------  
\((b_4)\) follows directly from (10), since every element in \(I_{tf}\) has a form \((t,\text{next}(F_c),c)\).

Proof of \([b_5]\)  
-------------  
Recall that we have to prove

\[
\forall t \in \mathbb{N}, F_t \in T_{\text{Formula}}, c \in \text{Context}: \\
(t,F_t,c) \in I_{tf} \land d \in \mathbb{N} \Rightarrow \\
c.1 = \{(X_f,t)\} \land c.2 = \{(X_f, sf(t))\} \land \\
n+1-d_f \leq t \leq n \land \\
T(F_f) \rightarrow^* (n+1-t,t,sf,c.1) F_t \land \\
\exists b \in \text{Bool} \exists d' \in \mathbb{N}: \\
d' \leq d_f \land \vdash F_t \rightarrow^* (\max(0,t+d'-n-1),n+1,sf,c.1) \text{ done}(b)
\]

We take \(t_b, F_{tb}, c_b\) arbitrary but fixed, assume

\((19) \ (t_b,F_{tb},c_b) \in I_{tf} \land d \in \mathbb{N}\)

and prove

\([b_{5.1}] \ c_b.1 = \{(X_f,t_b)\} \land c_b.2 = \{(X_f, sf(t_b))\}
\]

\([b_{5.2}] \ n+1-d_f \leq t_b \leq n
\]

\([b_{5.3}] \ T(F_f) \rightarrow^* (n+1-t_b,t_b,sf,c_b.1) F_{tb} \land
\]

\([b_{5.4}] \ \exists b \in \text{Bool} \ \exists d' \in \mathbb{N}: \\
d' \leq d_f \land \vdash F_{tb} \rightarrow^* (\max(0,t_b+d'-n-1),n+1,sf,c_b.1) \text{ done}(b)
\]

From (19) and (b4) we know that there exists \(F_{cb} \in T_{\text{FormulaCore}}\) such that

\((20) \ F_{tb} = \text{next}(F_{cb})\)
From (19), (20) and (10) of we know there exists $Ft_0 \in T_{\text{Formula}}$ such that

(21) $(tb,Ft_0,cb) \in I_{t_0}$ and
(22) $\vdash Ft_0 \rightarrow (n,sf|n,sf(n),cb) \text{ next}(Fcb)$.

Proof of [b5.1]
-----------------

We want to prove

[b5.1] $cb.1=\{(Xf,tb)\} \land cb.2=\{(Xf,sf(tb))\}$

From (21) and (11), we have two cases:

(C1) $(tb,Ft_0,cb) = (n,Ftf,\{(X',n)\},\{(X',sf(n))\})$ and
(C2) $(tb,Ft_0,cb) \in I_{t'}$.

In case (C1) we have $tb=n$, $Ft_0 = Ftf$, and $cb = \{(X',n)\},\{(X',sf(n))\}$. From the latter, by (8) and (14), we have $cb = \{(Xf,n)\},\{(Xf,sf(n))\}$ and, hence, since $tb=n$, we get $cb.1=\{(Xf,tb)\}$ and $cb.2=\{(Xf,sf(tb))\}$, which proves (b5.1) for the case (C1).

In case (C2), [b5.1] follows from (18).

Hence, [b5.1] is proved.

Proof of [b5.2]
-----------------

We want to prove

[b5.2] $n+1-df \leq tb \leq n$.

Again, from (21) and (11), we have two cases:

(C1) $(tb,Ft_0,cb) = (n,Ftf,\{(X',n)\},\{(X',sf(n))\})$ and
(C2) $(tb,Ft_0,cb) \in I_{t'}$.

The case (C1)
-------------

In case (C1) we have $tb=n$, $Ft_0 = Ftf$, and $cb = \{(X',n)\},\{(X',sf(n))\})$. From the latter, by (8) and (14), we have $cb = \{(Xf,n)\},\{(Xf,sf(n))\})$. To show [b5.2], it just remains to prove

[23] $df > 0$.

Assume by contradiction that $df=0$. Then from (2) we get that there exists $re_0 \in \text{RangeEnv}$ such that $re_0(Xf) = (0,0)$ and

(24) $re_0 \vdash Ff:(hf,0)$

Now we apply Statement 1 of Lemma 1 with $F=Ff$, $re=re_0$, $e=\{(Xf,n)\}, s=sf,$
d=df=0, h=hf, s=sf, p=n, and since T(Ff)=Ftf by (17), we obtain

\[(25) \exists b \in \text{Bool} \exists d' \in \mathbb{N}: d' \leq 1 \land \vdash Ftf \rightarrow^*(d',n,\{Xf,n\}) \text{ done}(b)\]

From (25), there exist bl \in \text{Bool} and dl' \in \mathbb{N} such that

\[(26) \exists dl' \leq 1 \text{ and } \exists Ftf \rightarrow^*(dl',n,\{Xf,n\}) \text{ done}(bl).\]

Note that since Ftf = T(Ff), by the definition of the translation T, Ftf is a 'next' formula. Hence, dl' \neq 0, because otherwise by (27) and the definition of \(\rightarrow^*\) we would get Fft=done(bl), which would contradict the fact that Ftf is a 'next' formula. Therefore, from (26) we get

\[(28) dl'=1.\]

From (27) and (28) we get

\[(29) Ftf \rightarrow^*(1,n,\{Xf,n\}) \text{ done}(bl).\]

From (29), by the definition of \(\rightarrow^*\) for TFormulas, we get that there exists Ft' such that

\[(30) Ftf \rightarrow(n,\{Xf,n\},\{Xf,\text{sf}(n)\}) Ft'\]

\[(31) Ft' \rightarrow^*(0,n+1,\{Xf,\text{sf}(n)\}) \text{ done}(bl).\]

On the other hand, from (22), by Ft0=Ftf and (b5.1) we get

\[(32) Ftf \rightarrow(n,\{Xf,n\},\{Xf,\text{sf}(n)\}) \text{ next}(Fcb)\]

From (30) and (32) and by the fact that the reduction \(\rightarrow\) is deterministic (one can not perform two different reductions from Ftf with the same n,\{Xf,n\},\{Xf,\text{sf}(n)\}), this can be seen by inspecting the rules for \(\rightarrow\), we obtain

\[(33) Ft'=\text{next}(Fcb).\]

Then from (31) and (33) we get

\[(34) \text{next}(Fcb) \rightarrow^*(0,n+1,\{Xf,n\},\{Xf,\text{sf}(n)\}) \text{ done}(bl).\]

But this contradicts the definition of \(\rightarrow^*\): A 'next' formula can not be reduced to a 'done' formula in 0 steps. Hence, the obtained contradiction proves [23] and, therefore, [b5.2] for the case (C1).

Now we consider the case (C2).

\[\text{-----------------------------}\]

From (tb,Ft0,cb) \in I't', by (18), we get

\[(35) n-df \leq tb \leq n-1.\]

In order to prove [b5.2], we need to show

\[(36) n+1-df \leq tb.\]
Assume by contradiction that \(n+1-df > tb\). By (35) it means \(n-df = tb\).

From (18) with \(t=tb\), \(Ft=Ft0\), \(c=cb\) we get

\[
\exists b \in \text{Bool} \exists d' \in \mathbb{N} : \\
d' \leq df \land \vdash Ft0 \rightarrow *(\max(0,tb+d'-n),sf,cb.1) \text{ done}(b)
\]

Since \(tb+d'-n = n-df+d'-n = d'-df\), from (37), we obtain that there exist \(b\) and \(d'\) such that

\[
d' \leq df \land \vdash Ft0 \rightarrow *(\max(0,d'-df),sf,cb.1) \text{ done}(b)
\]

holds. But then \(\max(0,d'-df)=0\) and we get

\[
Ft0 \rightarrow *(0,sf,cb.1) \text{ done}(b)
\]

which, by definition of \(\rightarrow *\) for TFormulas, implies

\[
Ft0 = \text{done}(b).
\]

However, this contradicts (22) and the definition of \(\rightarrow\) for TFormulas, because no 'done' formula can be reduced. Hence, (36) holds, which implies [b5.2] also in this case.

Proof of [b5.3]

We have to prove \(T(Ff) \rightarrow * (n+1-tb,tb,sf,cb.1) Ft0\), which, by Lemma 2, is equivalent to proving

\[
T(Ff) \rightarrow l* (n+1-tb,tb,sf,cb.1) Ft0
\]

Since \(n+1-tb>0\) (by b5.2), by the definition of \(\rightarrow l^*\), proving (41) reduces to proving that there exists such a \(Ft'\) that

\[
T(Ff) \rightarrow l* (n-tb,tb,sf,cb.1) Ft' \quad \text{and} \\
Ft' \rightarrow (n,sf\downarrow(n),s(n),c') Ft0
\]

where \(c'=(cb.1,\{(X,sf(cb.1(X))) | X \in \text{dom}(cb.1)\})\). But since \(\text{dom}(cb.1)={Xf}\), we actually get

\[
c' = cb.
\]

Let us take \(Ft'=Ft0\). Then (43) follows from (22). To prove (41), we reason as follows:

From (21), we know that \((tb,Ft0,cb) \in It0\). By (11) and (14), we have

\[
It0 = It' \cup \{(n,Ftf,\{(Xf,n),\{(Xf,sf(n))\})\})
\]

Let us first consider the case when \((tb,Ft0,cb) \in It'\). From (18) we have

\[
T(Ff) \rightarrow * (n-tb,tb,sf,cb.1) Ft0
\]

From (46), by Lemma 2, we get (42).
Now assume \((t_b,F_{t_0},c_b)\) = \((n,F_{t_f},\{(X_f,n),(X_f,\text{sf}(n))\})\). That means, taking \(t_b=n\), \(F_{t_0}=F_{t_f}\), and \(c_b=\{(X_f,n),(X_f,\text{sf}(n))\}\). Then, from (42), we need to prove

\[ T(F_f) \to (0,n,\text{sf},\{(X_f,n)\}) F_{t_f}. \]

This follows from the definition of \(\to\) and \([b2]\).

Hence, \([b5.3]\) is proved.

Proof of \([b5.4]\)

Recall that we took \(t_b\), \(F_{t_b}\), and \(c_b\) arbitrary but fixed and assumed

\[ (21) \, (t_b,F_{t_b},c_b) \in \text{It}_f. \]

We are looking for \(b^*\in\text{Bool}\) and \(d'^*\in\mathbb{N}\) such that

\[ d'^* \leq d_f \text{ and } \vdash F_{t_b} \to^* (\max(0,t_b+d'^*-n-1),n+1,\text{sf},c_b.1) \text{ done}(b^*) \]

hold.

From (21) and (b4) we know that there exists \(F_{c_b}\in\text{TFormulaCore}\) such that

\[ (50) \, F_{t_b}=\text{next}(F_{c_b}) \]

From (21), by (11) there are two cases:

(C1) \((t_b,F_{t_0},c_b) = (n,F_{t_f},\{(X',n),(X',\text{sf}(n))\})\)

(C2) \((t_b,F_{t_0},c_b) \in \text{It}'\)

Case (C1):

From (C1) we know

\[ (51) \, t_b = n \]

\[ (52) \, F_{t_0} = F_{t_f} \]

\[ (53) \, c_b = \{(X_f,n),(X_f,\text{sf}(n))\} \]

From (51), to show \([b5.3]\), it suffices to show

\[ [b5.3.a] \, \exists b\in\text{Bool}, \, d'\in\mathbb{N}: \]

\[ d' \leq d_f \land \vdash F_{t_b} \to^* (\max(0,d'-1),n+1,\text{sf},c_b.1) \text{ done}(b) \]

From (53), we know

\[ (54) \, c_b.1 = \{(X_f,n)\} \]

\[ (55) \, c_b.2 = \{(X_f,\text{sf}(n))\} \]

From (2) and the definition of \(\vdash\) we have some \(r\in\text{RangeEnv}\) such that

\[ (56) \, r(X_f) = (0,0) \]

\[ (57) \, r \vdash F_f : (h_f,d_f) \]

31
From (Statement 1 of Lemma 1,57,19,15), we have some $b_1 \in \text{Bool}$ and $d_1' \in \mathbb{N}$ such that

(58) $d_1' \leq d_f + 1$
(59) $\vdash F_t \rightarrow \ast(d_1',n,s_f,\{(X_f,n)\}) \text{ done}(b_1)$

From (20,59) and the definition of $\rightarrow \ast$, we know for some $F_{t_b} \in \text{TInstance}$

(60) $d_1' > 0$
(61) $\vdash F_t \rightarrow (n,s_f \downarrow n,s_f(n),\{(X_f,n),\{(X_f,s_f(n))\})) F_{t_b}$
(62) $\vdash F_{t_b} \rightarrow \ast(d_1'-1,n+1,s_f,\{(X_f,n)\}) \text{ done}(b_1)$

From (22,52,53), we know

(63) $\vdash F_t \rightarrow (n,s_f \downarrow n,s_f(n),\{(X_f,n),\{(X_f,s_f(n))\})) F_{t_b}$

From (61,63) and the fact that the rules for $\rightarrow$ are deterministic
(i.e., $\forall F_t,F_{t_b},F_{t_b}' : (\vdash F_t \rightarrow F_{t_b}) \land (\vdash F_t \rightarrow F_{t_b}') \Rightarrow F_{t_b} = F_{t_b}'$, a lemma easy to prove), we know

(64) $F_{t_b} = F_{t_b}$

From (62,64), we know

(65) $\vdash F_{t_b} \rightarrow \ast(d_1'-1,n+1,s_f,\{(X_f,n)\}) \text{ done}(b_1)$

From (60), we know

(66) $d_1'-1 = \max(0,d_1'-1)$

From (58,65,66,54), we know $[b5.3.a]$ with $b:=b_1$ and $d:=d_1'-1$.

Case (C2).
-------------
Recall that in this case $(t_b,F_{t_0},c_b) \in I_t'$.

By the induction hypothesis (18) there exist $b_i \in \text{Bool}$ and $d_i' \in \mathbb{N}$ such that

(67) $d_i' \leq d_f$ and
(68) $\vdash F_{t_0} \rightarrow \ast(\max(0,t_b+d_i'-n),n,s_f,c_b.1) \text{ done}(b_i)$

This implies that

(69) $t_b+d_i'-n > 0$,

otherwise we would have $F_{t_0} = \text{done}(b_i)$, which contradicts the assumption $(t_b,F_{t_0},c_b) \in I_t'$ and (20). Hence, we have

(70) $\vdash F_{t_0} \rightarrow \ast(t_b+d_i'-n,n,s_f,c_b.1) \text{ done}(b_i)$

Therefore, we can apply the definition $\rightarrow \ast$ for TFormulas to (70) and (22), concluding $\vdash \text{next}(F_{c_b}) \rightarrow \ast(t_b+d_i'-n-1,n+1,s_f,c_b.1) \text{ done}(b_i)$ and, hence
(71) \[ \text{Ft} \vdash \text{tb} \rightarrow_{*}(\text{tb+di}'-n-1,n+1,\text{sf},\text{cb}.1) \text{ done(bi)} \]

Now we can take \(d'=d'\) and \(b'=bi\). From (59) we get

(72) \[ \text{tb+di*'-n-1} = \max(0,\text{tb+di*'-n-1}). \]

From (71) and (72) we get [49]. From (67) and the assumption \(d*'=d'\) we get [48].

Hence, [b5.3] is true also in case (b6.2 C2).

This finishes the invariant proof.
A.3 Lemma 1: Soundness Lemma for Formulas

∀F,F'∈Formula, re∈RangeEnv, e∈Environment, s∈Stream, d∈N∞, h∈N:
∥ (re ⊢ F: (h,d)) ∧ ∀Y∈dom(e): re(Y).1+p ≤ e(Y) ≤ re(Y).2+p ∴
( d∈N ⇒
  ∃b∈Bool, ∃d'∈N:
  d'≤d+1 ∧ ⊢ (d',p,s,e) done(b) ) ∧
( ∀h'∈N: h'≥h ⇒
  ( T(F) →* (n,p,s,e) Ft ⇔
  T(F) →* (n,p,s,e,h') Ft ) )

We split the lemma in two parts:

Statement 1.
∀F∈Formula, re∈RangeEnv, e∈Environment, s∈Stream, d∈N∞, h∈N:
∥ (re ⊢ F: (h,d)) ∧ ∀Y∈dom(e): re(Y).1+p ≤ e(Y) ≤ re(Y).2+p ∴
( d∈N ⇒
  ∃b∈Bool, ∃d'∈N:
  d'≤d+1 ∧ ⊢ (d',p,s,e) done(b) )

Statement 2.
∀F∈Formula, re∈RangeEnv, e∈Environment, Ft∈TFormula, n∈N, p∈N,
s∈Stream, d∈N∞, h∈N, h'∈N:
∥ (re ⊢ F: (h,d)) ∧ ∀Y∈dom(e): re(Y).1+p ≤ e(Y) ≤ re(Y).2+p ∧ h'≥h ∴
( T(F) →* (n,p,s,e) Ft ⇔
  T(F) →* (n,p,s,e,h') Ft )

Parametrization
---------------
R(F) :
∀re∈RangeEnv, e∈Environment, s∈Stream, d∈N∞, h∈N:
∥ (re ⊢ F: (h,d)) ∧ ∀Y∈dom(e): re(Y).1+p ≤ e(Y) ≤ re(Y).2+p ∧ d∈N⇒
( ∀p∈N ∃b∈Bool, ∃d'∈N:
  d'≤d+1 ∧ ⊢ (d',p,s,e) done(b) )

We want to prove
\( \forall F \in \text{Formula} : R(F) \)

By structural induction over \( F \):

**C1:** \( F = \Theta X \). Then \( T(F) = \text{next}(TV(X)) \).

---

We take \( \text{ref}, \text{ef}, \text{sf}, \text{df}, \text{hf}, \text{pf} \) arbitrary but fixed. Assume

1. \( \vdash (\text{ref} \vdash \Theta X : (\text{hf}, \text{df})) \)
2. \( \text{df} \in \mathbb{N} \),
3. \( \forall Y \in \text{dom(ef)} : \text{ref}(Y) \cdot 1 + \text{pf} \leq \text{ef}(Y) \leq \text{ref}(Y) \cdot 2 + \text{pf} \)

and look for \( b^* \in \text{Bool} \) and \( d^*' \in \mathbb{N} \) such that

1. \( d^*' \leq \text{df} + 1 \)
2. \( \vdash \text{next}(TV(X)) \rightarrow (d^*, \text{pf}, \text{sf}, \text{ef}) \) done\( (b^*) \)

hold.

From (1.1) we get

1. \( \text{hf} = 0 \)
2. \( \text{df} = 0 \).

We define

1. \( c = (\text{ef}, \{(X, \text{sf}(\text{ef}(X))) \mid X \in \text{dom(ef)}\}) \),

and take

1. \( d^* = 1 \)

and

1. \( b^* = \\
    \begin{cases} \\
    \text{c.2}(X) & \text{if } X \in \text{dom(c.2)} \\
    \text{false} & \text{else} \\
    \end{cases} \\
    \\
\)

From (1.7, 1.9), we see that \( d^* \) satisfies [1.4]. Hence, we only need to prove the following formula obtained from [1.5]:

1. \( \vdash \text{next}(TV(X)) \rightarrow (1, \text{pf}, \text{sf}, \text{ef}) \) done\( (b^*) \).

where \( b^* \) is defined in (1.10). By the definition of \( \rightarrow^* \), to prove [1.11], we need to find \( F_t' \in \text{TFormula} \) such that

1. \( \text{next}(TV(X)) \rightarrow (\text{pf}, \text{sf}, \text{pf}, \text{sf}(\text{pf}), c) F_t' \) and
2. \( F_t' \rightarrow (0, \text{pf} + 1, \text{sf}, \text{ef}) \) done\( (b^*) \)

hold, where \( c \) is defined as in (1.8).
We take \( F_t' = \text{done}(b*) \). Then [1.12] holds by (1.10) and the definition of \( \rightarrow \) for \( \text{next}(\text{TV}(X)) \), and [1.13] holds by the definition of \( \rightarrow^* \).

C2. \( F = \neg F_1 \). Then \( T(F) = \text{next}(T(N(T(F_1)))) \).

We take \( \text{ref}, \text{ef}, \text{sf}, \text{df}, \text{hf}, \text{pf} \) arbitrary but fixed. Assume

(2.1) \( \vdash (\text{ref} \vdash \neg F_1: (\text{hf}, \text{df})) \)
(2.2) \( \text{df} \in \mathbb{N} \),
(2.2) \( \forall Y \in \text{dom}(\text{ef}): \text{ref}(Y).1+\text{pf} \leq \text{ef}(Y) \leq \text{ref}(Y).2+\text{pf} \)

and look for such \( b* \in \text{Bool} \) and \( d*' \in \mathbb{N} \) such that

[2.4] \( d*' \leq \text{df}+1 \) and
[2.5] \( \vdash \text{next}(T(N(T(F_1)))) \rightarrow^*(d*', \text{pf}, \text{sf}, \text{ef}) \text{ done}(b*) \)

hold.

From (2.1) by the definition of the \( \vdash \) relation we get

(2.6) \( \vdash (\text{ref} \vdash F_1): (\text{hf}, \text{df}) \).

From (2.6) and the induction hypothesis there exist \( b_i \in \text{Bool} \) and \( d_i' \in \mathbb{N} \) such that

(2.7) \( d_i' \leq \text{df}+1 \) and
(2.8) \( \vdash T(F_1) \rightarrow^*(d_i', \text{pf}, \text{sf}, \text{ef}) \text{ done}(b_i) \).

We take

(2.9) \( d*' = d_i' \)

and define

(2.10) \( b* := \)
if \( b_i = \text{true} \) then
false
else
true

By (2.7,2.9), the inequality [2.4] holds. From (2.8), (2.9), (2.10), by the Statement 1 of the Lemma 4 we get [2.5].

C3. \( F = F_1 \& F_2 \). Then \( T(F) = \text{next}(T(CS(T(F_1),T(F_2)))) \).

We take \( \text{ref}, \text{ef}, \text{sf}, \text{df}, \text{hf}, \text{pf} \) arbitrary but fixed. Assume

(3.1) \( \vdash (\text{ref} \vdash F_1 \& F_2: (\text{hf}, \text{df})) \),
(3.2) \( \text{df} \in \mathbb{N} \),
(3.3) \( \forall Y \in \text{dom}(ef): \text{ref}(Y).1 + pf \leq ef(Y) \leq \text{ref}(Y).2 + pf \)

and look for such \( b* \in \text{Bool} \) and \( d* \in \mathbb{N} \) such that

[3.4] \( d* \leq df + 1 \) and

[3.5] \( \vdash \text{next}(\text{TCS}(T(F_1), T(F_2))) \rightarrow *(d*, pf, sf, ef) \text{ done}(b*) \)

From (3.1), by the definition of the \( \vdash \) relation we get

(3.6) \( \vdash (\text{ref} \vdash F_1: (h_1, d_1)) \)

(3.7) \( \vdash (\text{ref} \vdash F_2: (h_2, d_2)) \)

such that \( h_1, d_1, h_2, d_2 \in \mathbb{N} \) and

(3.8) \( df = \max(\infty(d_1, d_2)) = \max(d_1, d_2) \)

From (3.6) and the induction hypothesis there exist \( b_{1i} \in \text{Bool} \) and \( d_{1i} \in \mathbb{N} \) such that

(3.9) \( d_{1i} \leq d_1 + 1 \) and

(3.10) \( \vdash T(F_1) \rightarrow *(d_{1i} , pf, sf, ef) \text{ done}(b_{1i}) \).

From (3.7) and the induction hypothesis there exist \( b_{2i} \in \text{Bool} \) and \( d_{2i} \in \mathbb{N} \) such

(3.11) \( d_{2i} \leq d_2 + 1 \) and

(3.12) \( \vdash T(F_2) \rightarrow *(d_{2i} , pf, sf, ef) \text{ done}(b_{2i}) \).

From (3.10) and (3.12) we have

(3.13) \( d_{1i} > 0 \) and

(3.14) \( d_{2i} > 0 \)

Otherwise we would have a 'next' formula reducing to a 'done' formula in 0 steps, which is impossible.

We proceed by case distinction over \( b_{1i} \).

\( b_{1i} = \text{false} \)

-------------

We take

(3.15) \( b* = b_{1i} = \text{false} \) and

(3.16) \( d* = d_{1i} \).

From (3.8, 3.9, 3.16) we get [3.4]. From (3.10, 3.13, 3.15, 3.16) and Statement 2 of Lemma 4 we get [3.5].

\( b_{1i} = \text{true}. \)

-------------

We take

(3.17) \( b* = b_{2i}' \) and

(3.18) \( d* = \max(d_{1i}', d_{2i}') \).
From (3.18, 3.9, 3.11) we get

\[(3.19) \quad \text{d}^* = \max(d_{1i}', d_{2i}') \leq \max(d_1+1, d_2+1) = \max(d_1, d_2) + 1 = \text{df} + 1\]

Hence, (3.19) gives [3.4].

From (3.10, 3.12, 3.13, 3.14, 3.18) and Statement 2 of Lemma 4 we get [3.5].

C4. F = F1/\F2. Then T(F) = next(TCP(T(F1),T(F2))).

We take ref, ef, sf, df, hf, pf arbitrary but fixed. Assume

\[(4.1) \quad \vdash (\text{re} \vdash F1 \land F2: (hf, df)),\]
\[(4.2) \quad \text{df} \in \mathbb{N},\]
\[(4.3) \quad \forall Y \in \text{dom(ef)}: \text{ref}(Y).1 + pf \leq \text{ef}(Y) \leq \text{ref}(Y).2 + pf\]

and look for such b* \in \text{Bool} and d*' \in \mathbb{N} such that

\[\text{[4.4]} \quad \text{d}^*' \leq \text{df} + 1\]
\[\text{[4.5]} \quad \vdash \text{next}(TCP(T(F1),T(F2))) \rightarrow\ast(d^*, pf, sf, ef) \text{ done}(b^*)\]

From (4.1), by the definition of the \vdash relation we get

\[(4.6) \quad \vdash (\text{re} \vdash F1: (h_1, d_1))\]
\[(4.7) \quad \vdash (\text{re} \vdash F2: (h_2, d_2))\]

such that h_1, d_1, h_2, d_2 \in \mathbb{N} and

\[(4.8) \quad \text{df} = \max\infty(d_1, d_2) = \max(d_1, d_2)\]

From (4.6) and the induction hypothesis there exist b_{1i} \in \text{Bool} and d_{1i}' \in \mathbb{N} such that

\[(4.9) \quad d_{1i}' \leq d_1+1\]
\[(4.10) \quad \vdash T(F1) \rightarrow\ast(d_{1i}', pf, sf, ef) \text{ done}(b_{1i}).\]

From (4.7) and the induction hypothesis there exist b_{2i} \in \text{Bool} and d_{2i}' \in \mathbb{N} such

\[(4.11) \quad d_{2i}' \leq d_2+1\]
\[(4.12) \quad \vdash T(F2) \rightarrow\ast(d_{2i}', pf, sf, ef) \text{ done}(b_{2i}).\]

From (4.10) and (4.12) we have

\[(4.13) \quad d_{1i}' > 0\]
\[(4.14) \quad d_{2i}' > 0\]

(Otherwise we would have a 'next' formula reducing to a 'done' formula in 0 steps, which is impossible.)

We proceed by case distinction over b_{1i} and b_{2i}.

b_{1i} = \text{false}, b_{2i} = \text{true}
We take

(4.15) \( b^* = \text{false}, \)
(4.16) \( d^* = d_{i}' \).

From (4.8, 4.9, 4.16) we get \( d^* = d_{i}' \leq d_1 + 1 \leq \max(d_1, d_2) + 1 = d_f + 1 \) and, hence [4.4].

From (4.10, 4.12, 4.13, 4.14, 4.15, 4.16) and the case [TCP1] of the Statement 3 of Lemma 4 we get [4.5].

\[ \text{b}_{i1} = \text{false}, \text{b}_{2i} = \text{false} \]

We take

(4.17) \( b^* = \text{false}, \)
(4.18) \( d^* = \min(d_{i1}', d_{i2}') \).

From (4.9, 4.11, 4.18) we get

(4.19) \( d^* = \min(d_{i1}', d_{i2}') \leq \min(d_1 + 1, d_2 + 1) = \min(d_1, d_2) + 1 \leq \max(d_1, d_2) + 1 = d_f + 1. \)

Hence, (4.19) proves [4.4].

From (4.10, 4.12, 4.13, 4.14, 4.17, 4.18) and the case [TCP2] of the Statement 3 of Lemma 4 we get [4.5].

\[ \text{b}_{i1} = \text{true}, \text{b}_{2i} = \text{true} \]

We take

(4.20) \( b^* = b_{2i}' \) and
(4.21) \( d^* = \max(d_{i1}', d_{i2}') \).

From (4.20, 4.9, 4.11) we get

(4.22) \( d^* = \max(d_{i1}', d_{i2}') \leq \max(d_1 + 1, d_2 + 1) = \max(d_1, d_2) + 1 = d_f + 1 \)

Hence, (4.22) gives [4.4].

From (4.10, 4.12, 4.13, 4.14, 4.20, 4.22) and the case [TCP3] of the Statement 3 of Lemma 4 we get [4.5].

\[ \text{b}_{i1} = \text{true}, \text{b}_{2i} = \text{false} \]

We take

(4.23) \( b^* = b_{2i}' \) and
(4.24) \( d^* = d_{2i}' \).

From (4.18, 4.9, 4.11) we get

(4.25) \( d^* = d_2 + 1 \leq \max(d_1 + 1, d_2 + 1) = \max(d_1, d_2) + 1 = d_f + 1 \)

Hence, (4.25) gives [4.4].
From (4.10, 4.12, 4.13, 4.14, 4.23, 4.24) and the case TCP4 of the Statement 3 of Lemma 4 we get [4.5].

C5. \( F = \forall X \in B1..B2:F1 \). Then \( T(F) = \text{next}(TA(X,T(B1),T(B2),T(F1))) \)

This case follows from the induction hypothesis and Lemma 5.

It finishes the proof of Statement 1 of Lemma 1.

================================================================================

Statement 2.
\( \forall F \in \text{Formula}, re \in \text{RangeEnv}, e \in \text{Environment}, Ft \in \text{TFormula}, n \in \mathbb{N}, p \in \mathbb{N}, \
 s \in \text{Stream}, d \in \mathbb{N}^\infty, h \in \mathbb{N}, h' \in \mathbb{N}: \)
\( \vdash (re \vdash F: (h,d)) \land \forall Y \in \text{dom}(e): \text{re}(Y).1+p \leq e(Y) \leq \text{re}(Y).2+p \land h' \geq h \Rightarrow \
( T(F) \rightarrow* (n,p,s,e) \quad Ft \iff \\
T(F) \rightarrow* (n,p,s,e,h') \quad Ft ) \)

Proof
-----

Parametrization:
\( S(n) : \iff \\
\forall F \in \text{Formula}, re \in \text{RangeEnv}, e \in \text{Environment}, Ft \in \text{TFormula}, p \in \mathbb{N}, \
 s \in \text{Stream}, d \in \mathbb{N}^\infty, h \in \mathbb{N}, h' \in \mathbb{N}: \)
\( \vdash (re \vdash F: (h,d)) \land \forall Y \in \text{dom}(e): \text{re}(Y).1+p \leq e(Y) \leq \text{re}(Y).2+p \land h' \geq h \Rightarrow \
( T(F) \rightarrow* (n,p,s,e) \quad Ft \iff \\
T(F) \rightarrow* (n,p,s,e,h') \quad Ft ) \)

We need to prove

(a) \( S(0) \)
(b) \( \forall n \in \mathbb{N}: S(n) \Rightarrow S(n+1) \)

Proof of (a)
----------

We take \( Ff \in \text{Formula}, ref \in \text{RangeEnv}, ef \in \text{Environment}, Ft \in \text{TFormulas}, p \in \mathbb{N}, \
sf \in \text{Stream}, df \in \mathbb{N}^\infty, hf \in \mathbb{N}, hf' \in \mathbb{N} \) arbitrary but fixed, assume

(a.1) \( \vdash (ref \vdash Ff: (hf,df)) \)
(a.2) \( \forall Y \in \text{dom}(ef): \text{ref}(Y).1+pf \leq \text{ef}(Y) \leq \text{ref}(Y).2+pf \)
(a.3) \( hf' \geq hf \)

and prove

(a.4) \( T(Ff) \rightarrow* (0,pf,sf,ef) \quad Ft \iff \\
T(Ff) \rightarrow* (0,pf,sf,ef,hf') \quad Ft \)
Assume (a.5) \[ T(Ff) \rightarrow^* (0,pf,sf,ef) Ftf \]
and prove (a.6) \[ T(Ff) \rightarrow^* (0,pf,sf,ef,hf') Ftf. \]

From (a.5), by the definition of \( \rightarrow^* \) without history, we have \( Ftf = T(Ff) \).
Then (a.6) follows from the definition of \( \rightarrow^* \) with history.

\( \Leftarrow \). Analogous.

Proof of (b) -----------

We assume

(b.1) \[ \forall F \in \text{Formula}, \, \text{ref} \in \text{RangeEnv}, \, e \in \text{Environment}, \, Ft \in \text{TFormula}, \, p \in \mathbb{N}, \, s \in \text{Stream}, \, d \in \mathbb{N}, \, h \in \mathbb{N}, \, h' \in \mathbb{N}
\] \[ \vdash (\text{re} \vdash F : (h,d)) \land \forall Y \in \text{dom}(e): \text{re}(Y).1 + p \leq e(Y) \leq \text{re}(Y).2 + p \land h' \geq h \Rightarrow
\]
\[ (T(F) \rightarrow^* (n,p,s,e) Ft \iff
\]
\[ (T(F) \rightarrow^* (n,p,s,e,h') Ft )
\]

and prove

(b.2) \[ \forall F \in \text{Formula}, \, \text{ref} \in \text{RangeEnv}, \, e \in \text{Environment}, \, Ft \in \text{TFormula}, \, p \in \mathbb{N}, \, s \in \text{Stream}, \, d \in \mathbb{N}, \, h \in \mathbb{N}, \, h' \in \mathbb{N}
\] \[ \vdash (\text{re} \vdash F : (h,d)) \land \forall Y \in \text{dom}(e): \text{re}(Y).1 + p \leq e(Y) \leq \text{re}(Y).2 + p \land h' \geq h \Rightarrow
\]
\[ (T(F) \rightarrow^* (n+1,p,s,e) Ft \iff
\]
\[ (T(F) \rightarrow^* (n+1,p,s,e,h') Ft )
\]

We take \( Ff, \, \text{ref}, \, ef, \, Ft, \, pf, \, sf, \, df, \, hf, \, hf' \) arbitrary but fixed. Assume

(b.3) \[ \vdash (\text{ref} \vdash Ff : (hf,df))
\]
(b.4) \[ \forall Y \in \text{dom}(ef): \text{ref}(Y).1 + pf \leq ef(Y) \leq \text{ref}(Y).2 + pf
\]
(b.5) \[ hf' \geq hf
\]

and prove

(b.6) \[ T(Ff) \rightarrow^* (n+1,pf,sf,ef) Ftf \iff
\]
\[ T(Ff) \rightarrow^* (n+1,pf,sf,ef,hf') Ftf
\]

\( \Rightarrow \) Assume

(b.7) \[ T(Ff) \rightarrow^* (n+1,pf,sf,ef) Ftf
\]

and prove

[b.8] \[ T(Ff) \rightarrow^* (n+1,pf,sf,ef,hf') Ftf
\]
From (b.7), by the definition of $\rightarrow^*$ without history, we know for some $Ft' \in T_{Formula}$

(b.9) $T(Ff) \rightarrow (pf, sf, pf, sf(pf), c) Ft'$

(b.10) $Ft' \rightarrow^* (n, pf+1, sf, ef) Ftf,$

(b.11) $c := (ef, \{(X, sf(ef(X))) | X \in \text{dom}(ef)\}).$

Then from (b.3), (b.4), (b.11), (b.5), (b.9) and Lemma 3 we get

(b.12) $T(Ff) \rightarrow (pf, sf^{\uparrow}(\max(0, pf-hf'), \min(pf, hf')), sf(pf), c) Ft'.$

Assume $Ft'$ is a 'next' formula, i.e., there exists $F' \in \text{Formula}$ such that

(b.13) $Ft' = T(F')$.

From (b.3), (b.4), (b.5), (b.10), by the induction hypothesis (b.1) we get

(b.14) $Ft' \rightarrow^* (n, pf+1, sf, ef, hf') Ftf.$

If $Ft'$ is a 'done' formula, then from (b.10) by the definition of $\rightarrow^*$ without history we get $n=0$. Then, (b.14) again holds by the definition of $\rightarrow^*$ with history.

From (b.11), (b.12) and (b.14), by the definition of $\rightarrow^*$ with history we get [b.8].

($\Leftarrow$) Assume

(b.15) $T(Ff) \rightarrow^* (n+1, pf, sf, ef, hf') Ftf$

and prove

[b.16] $T(Ff) \rightarrow^* (n+1, pf, sf, ef) Ftf$

From (b.15), by the definition of $\rightarrow^*$ without history, we know for some $Ft' \in T_{Formula}$

(b.17) $T(Ff) \rightarrow (pf, sf^{\uparrow}(\max(0, pf-hf'), \min(pf, hf')), sf(pf), c) Ft'$

(b.18) $Ft' \rightarrow^* (n, pf+1, sf, ef, hf') Ftf,$

where

(b.19) $c := (ef, \{(X, sf(ef(X))) | X \in \text{dom}(ef)\}).$

Then from (b.3), (b.19), (b.4), (b.5), (b.18) and Lemma 3 we get

(b.20) $T(Ff) \rightarrow (pf, sf, pf, sf(pf), c) Ft'.$

Assume $Ft'$ is a 'next' formula, i.e., there exists $F' \in \text{Formula}$ such that

(b.21) $Ft' = T(F')$.

42
From (b.3), (b.4), (b.5), (b.18) by the induction hypothesis (b.1) we get

(b.22) $Ft' \rightarrow^* (n, pf+1, sf, ef) Ftf$.

If $Ft'$ is a 'done' formula, then from (b.18) by the definition of $\rightarrow^*$ without history we get $n=0$. Then, (b.22) again holds by the definition of $\rightarrow^*$ with history.

From (b.19), (b.20) and (b.22), by the definition of $\rightarrow^*$ with history we get [b.16].

It finishes the proof of Statement 2 of Lemma 1.
A.4 Lemma 2: Equivalence of Left- and Right-Recursive Definitions of n-Step Reductions

Lemma 2 (Equivalence of Left- and Right-Recursive Definitions of n-Step Reductions):

(a) \( \forall n,p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, F_{t1}, F_{t2} \in T_{\text{Formula}} \)
\[ F_{t1} \rightarrow^* (n,p,s,e) F_{t2} \iff F_{t1} \rightarrow^{l*} (n,p,s,e) F_{t2} \]

(b) \( \forall n,p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, F_{t1}, F_{t2} \in T_{\text{Formula}}, h \in \mathbb{N} \)
\[ F_{t1} \rightarrow^* (n,p,s,e,h) F_{t2} \iff F_{t1} \rightarrow^{l*} (n,p,s,e,h) F_{t2} \]

Proof of (a)
---------------

Parametrization:
---------------

S(n,F_{t1},F_{t2},p,s,e) :
\[ \iff F_{t1} \rightarrow^* (n,p,s,e) F_{t2} \iff F_{t1} \rightarrow^{l*} (n,p,s,e) F_{t2} \]

We want to prove

\[ [G] \forall F_{t1}, F_{t2} \in T_{\text{Formula}}, p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, \forall n \in \mathbb{N}: S(n,F_{t1},F_{t2},p,s,e). \]

We take F_{t1}, F_{t2}, p, s, and e arbitrary but fixed.

We have to prove

\[ [G1] \forall k,n \in \mathbb{N}: S(k,F_{t1},F_{t2},p,s,e) \wedge n > k \Rightarrow S(n,F_{t1},F_{t2},p,s,e). \]

Proof of \([G1]\)
---------------

We take n arbitrary but fixed, assume

(1) \( \forall k < n : F_{t1} \rightarrow^* (k,p,s,e) F_{t2} \iff F_{t1} \rightarrow^{l*} (k,p,s,e) F_{t2} \)

and prove

\[ [2] F_{t1} \rightarrow^* (n,p,s,e) F_{t2} \iff F_{t1} \rightarrow^{l*} (n,p,s,e) F_{t2}. \]

(\( \iff \):)
----
We assume

(3) \( F_{t1} \rightarrow^*(n,p,s,e) F_{t2} \)

and prove

\[ [4] F_{t1} \rightarrow^{l*} (n,p,s,e) F_{t2}. \]
From (3) we know that there exists $Ft' \in T\text{Formula}$ such that

(5) $Ft_1 \rightarrow (pf, sf, pf, sf(pf), c) \,Ft'$ and
(6) $Ft' \rightarrow \star(n-1, pf+1, sf, ef)\,Ft_2$

hold, where $c = (ef, \{(X, sf(ef(X))) \mid X \in \text{dom}(ef)\})$.

From (6), by the induction hypothesis we get

(7) $Ft' \rightarrow \star(n-1, pf+1, sf, ef)\,Ft_2$.

From (7), by the definition of $\rightarrow \star$, there are two alternatives:

(i) $n-1 = 0$
(ii) $n-1 > 0$.

In case (i), we get

(8) $Ft' = Ft_2$.

From (8) and (5) we get

(9) $Ft_1 \rightarrow (pf, sf, pf, sf(pf), c)\,Ft_2$.

On the other hand, by the definition of $\rightarrow \star$ we have

(10) $Ft_1 \rightarrow \star(0, pf, sf, ef)\,Ft_1$.

From (10) and (9), by the definition of $\rightarrow \star$, we get

(11) $Ft_1 \rightarrow \star(1, pf, sf, ef)\,Ft_2$.

Since $n-1=0$, we get that [4] holds:

[4] $Ft_1 \rightarrow \star(n, pf, sf, ef)\,Ft_2$.

Case (ii)

From (7), by the definition of $\rightarrow \star$, there exists $Ft'' \in T\text{Formula}$ such that

(12) $Ft' \rightarrow \star(n-2, pf+1, sf, ef)\,Ft''$
(13) $Ft'' \rightarrow (pf+n-1, sf, pf+n-1, sf(ef+n-1), c)\,Ft_2$,

where $c = (ef, \{(X, sf(ef(X))) \mid X \in \text{dom}(ef)\})$.

From (12), by the induction hypothesis, we get

(14) $Ft' \rightarrow \star(n-2, pf+1, sf, ef)\,Ft''$.

From (5) and (14), by the definition of $\rightarrow \star$ we get

(15) $Ft_1 \rightarrow \star(n-1, pf, sf, ef)\,Ft''$.

From (15), by the induction hypothesis, we get
(16) $F_{tf1} \rightarrow l^*(n-1,pf,sf,ef) F_{t''}$.

From (16) and (13), by the definition of $\rightarrow l^*$, we get

[4] $F_{tf1} \rightarrow l^*(n,pf,sf,ef) F_{tf2}$.

(\iff)

We assume

(17) $F_{tf1} \rightarrow l^* (n,pf,sf,ef) F_{tf2}$

and prove

[18] $F_{tf1} \rightarrow* (n,pf,sf,ef) F_{tf2}$.

From (17), by the definition of $\rightarrow l^*$, we know that there exists $F_{t'} \in T_{Formula}$ such that

(19) $F_{tf1} \rightarrow l^* (n-1,pf,sf,ef) F_{t'}$ and
(20) $F_{t'} \rightarrow (pf+n-1, sf \downarrow (pf+n-1), sf(pf+n-1), c) F_{tf2}$,

hold, where $c = (ef, \{(X, sf(ef(X))) \mid X \in \text{dom}(ef)\})$.

From (19), by the induction hypothesis we get

(21) $F_{tf1} \rightarrow* (n-1,pf,sf,ef) F_{t'}$

from (20), by the definition of $\rightarrow l^*$, there are two alternatives:

(i) $n-1 = 0$
(ii) $n-1 > 0$.

Case (i)

---------

In this case, from (21) we get $F_{t'} = F_{tf1}$, which together with (20) and the fact $n-1=0$ implies

(22) $F_{tf1} \rightarrow (pf,sf, pf, c) F_{tf2}$.

On the other hand, by the definition of $\rightarrow*$ we have

(23) $F_{tf2} \rightarrow* (0,pf+1, sf, ef) F_{tf2}$.

From (22) and (23), by the definition of $\rightarrow*$, we get

(24) $F_{tf2} \rightarrow* (1,pf, sf, ef) F_{tf2}$.

Since $n-1=0$, from (24) we get [18].

Case (ii)

--------

From (21), by the definition of $\rightarrow*$, there exists $F_{t''} \in T_{Formula}$ such that
From (26), by the induction hypothesis, we get

(27) \( F'' \rightarrow l^* (n-2, pf+1, sf, ef) F' \).

From (27) and (20), by the definition of \( \rightarrow l^* \) we get

(28) \( F'' \rightarrow l^* (n-1, pf+1, sf, ef) Ftf2 \).

From (28), by the induction hypothesis we get

(29) \( F'' \rightarrow \ast (n-1, pf+1, sf, ef) Ftf2 \).

From (25) and (29), by the definition of \( \rightarrow \ast \), we get

[18] \( Ftf1 \rightarrow \ast (n, pf, sf, ef) Ftf2 \).
We assume (3) $F_{f_1} \rightarrow^{*}(n, pf, sf, ef, hf) F_{f_2}$ and prove (4) $F_{f_1} \rightarrow^{l*}(n, pf, sf, ef, hf) F_{f_2}$.

From (3) we know that there exists $F_{t'} \in T_{Formula}$ such that (5) $F_{f_1} \rightarrow (pf, s^\uparrow(\max(0, pf-hf), \min(pf, hf)), sf(pf), c) F_{t'}$ and (6) $F_{t'} \rightarrow^{*}(n-1, pf+1, sf, ef, hf) F_{f_2}$ hold, where $c = (ef, \{(X, sf(ef(X))) \mid X \in \text{dom}(ef)\})$.

From (6), by the induction hypothesis we get (7) $F_{t'} \rightarrow^{l*}(n-1, pf+1, sf, ef, hf) F_{f_2}$.

From (7), by the definition of $\rightarrow l*$, there are two alternatives:

(i) $n-1 = 0$
(ii) $n-1 > 0$.

In case (i), we get

(8) $F_{t'} = F_{f_2}$.

From (8) and (5) we get

(9) $F_{f_1} \rightarrow (pf, s^\uparrow(\max(0, pf-hf), \min(pf, hf)), sf(pf), c) F_{f_2}$.

On the other hand, by the definition of $\rightarrow l*$ we have

(10) $F_{f_1} \rightarrow^{l*}(0, pf, sf, ef, hf) F_{f_1}$.

From (10) and (9), by the definition of $\rightarrow l*$, we get

(11) $F_{f_1} \rightarrow^{l*}(1, pf, sf, ef, hf) F_{f_2}$.

Since $n-1=0$, we get that [4] holds:

[4] $F_{f_1} \rightarrow^{l*}(n, pf, sf, ef, hf) F_{f_2}$.

Case (ii)

From (7), by the definition of $\rightarrow l*$ with history, there exists $F_{t''} \in T_{Formula}$ such that

(12) $F_{t'} \rightarrow^{l*}(n-2, pf+1, sf, ef, hf) F_{t''}$

(13) $F_{t''} \rightarrow (pf+n-2, s^\uparrow(\max(0, pf+n-2-hf), \min(pf+n-2, hf)), sf(pf+n-2), c) F_{f_2}$,
where $c = (\text{ef}, \{(X, \text{sf}(\text{ef}(X))) \mid X \in \text{dom(ef)}\})$.

From (12), by the induction hypothesis, we get

(14) $Ft' \rightarrow_*(n-2, pf+1, sf, ef, hf) Ft''$.

From (5) and (14), by the definition of $\rightarrow_*$ with history we get

(15) $Ftf1 \rightarrow_*(n-1, pf, sf, ef, hf) Ft''$.

From (15), by the induction hypothesis, we get

(16) $Ftf1 \rightarrow_1(n-1, pf, sf, ef, hf) Ft''$.

From (16) and (13), by the definition of $\rightarrow_*$ with history, we get


(\Leftarrow)

We assume

(17) $Ftf1 \rightarrow_1(n, pf, sf, ef, hf) Ftf2$

and prove

[18] $Ftf1 \rightarrow_*(n, pf, sf, ef, hf) Ftf2$.

From (17), by the definition of $\rightarrow_1$ with history, we know that there exists $Ft' \in \text{TFormula}$ such that

(19) $Ftf1 \rightarrow_1(n-1, pf, sf, ef) Ft'$ and
(20) $Ft' \rightarrow_1(pf+n-1, s^{\uparrow}((\text{max}(0, pf+n-1-hf), \text{min}(pf+n-1, hf)), sf(pf+n-1), c) Ftf2$,

hold, where $c = (\text{ef}, \{(X, \text{sf}(\text{ef}(X))) \mid X \in \text{dom(ef)}\})$.

From (19), by the induction hypothesis we get

(21) $Ftf1 \rightarrow_1(n-1, pf, sf, ef, hf) Ft'$

from (20), by the definition of $\rightarrow_1$ with history, there are two alternatives:

(i) $n-1 = 0$
(ii) $n-1 > 0$.

Case (i)

In this case, from (21) we get $Ft' = Ftf1$, which together with (20) and the fact $n-1 = 0$ implies

(22) $Ftf1 \rightarrow_1(pf, s^{\uparrow}((\text{max}(0, pf-hf), \text{min}(pf, hf)), sf(pf), c) Ftf2$.

On the other hand, by the definition of $\rightarrow_*$ with history we have
(23) $F_{tf2} \rightarrow^*(0,pf+1,sf,ef,hf) F_{tf2}$.

From (22) and (23), by the definition of $\rightarrow^*$ with history, we get

(24) $F_{tf2} \rightarrow^*(1,pf,sf,ef,hf) F_{tf2}$.

Since $n-1=0$, from (24) we get [18].

Case (ii)

From (21), by the definition of $\rightarrow^*$ with history, there exists $F_{t''} \in T_{Formula}$ such that

(25) $F_{tf1} \rightarrow (pf, s\uparrow(max(0,pf-hf),min(pf,hf)),sf(pf),c) F_{t''}$

(26) $F_{t''} \rightarrow^*(n-2,pf+1,sf,ef,hf) F_{t'}$,

where $c = (ef,\{(X,sf(ef(X))) \mid X \in \text{dom}(ef)\})$.

From (26), by the induction hypothesis, we get

(27) $F_{t''} \rightarrow l^*(n-2,pf+1,sf,ef,hf) F_{t'}$.

From (27) and (20), by the definition of $\rightarrow l^*$ with history we get

(28) $F_{t''} \rightarrow l^*(n-1,pf+1,sf,ef,hf) F_{tf2}$.

From (28), by the induction hypothesis we get

(29) $F_{t''} \rightarrow^*(n-1,pf+1,sf,ef,hf) F_{tf2}$.

From (25) and (29), by the definition of $\rightarrow^*$, we get

[18] $F_{tf1} \rightarrow^*(n,pf, sf, ef, hf) F_{tf2}$. 
A.5 Lemma 3: History Cut-Off Lemma

Lemma 3 (History Cut-Off Lemma):

∀F ∈ Formula, Ft ∈ TFormula, p,q ∈ N, s ∈ Stream, h ∈ N, d ∈ N, e ∈ Environment, re ∈ RangeEnv:
⊢ (re ⊩ F : (h,d)) ∧ q ≤ p ∧ ∀Y ∈ dom(e): re(Y).1+q ≤ e(Y) ≤ re(Y).2+q ⇒
let c := (e, {(X, s(e(X))) | X ∈ dom(e)})
∀h' ∈ N : h' ≥ h ⇒
T(F) → (p, s↓p, s(p), c) Ft
⇔
T(F) → (p, s↑(max(0,p−h'),min(p,h')), s(p), c) Ft

Proof
-----

Parametrization:

S(F) :
∀Ft ∈ TFormula, p,q ∈ N, s ∈ Stream, h ∈ N, d ∈ N, e ∈ Environment, re ∈ RangeEnv:
⊢ (re ⊩ F : (h,d)) ∧ q ≤ p ∧ ∀Y ∈ dom(e): re(Y).1+q ≤ e(Y) ≤ re(Y).2+q ⇒
let c := (e, {(X, s(e(X))) | X ∈ dom(e)})
∀h' ∈ N : h' ≥ h ⇒
T(F) → (p, s↓p, s(p), c) Ft
⇔
T(F) → (p, s↑(max(0,p−h'),min(p,h')), s(p), c) Ft

We prove ∀F ∈ Formula S(F) by structural induction over F.

CASE 1. F = @X. T(F) = next(TV(X)).
-----------------------------------

We take ref,Ftf,pf,qf,sf,df,ef arbitrary but fixed. Assume

(1.1) ⊢ (ref ⊩ F : (hf,df))
(1.1') qf ≤ pf
(1.2) ∀Y ∈ dom(ef): ref(Y).1+qf ≤ ef(Y) ≤ ref(Y).2+qf

Define

(1.3) c := (ef, {(X, s(ef(X))) | X ∈ dom(ef)})

Take hf' arbitrary but fixed. Assume

(1.4) hf' ≥ hf

And prove

[1.5] T(F) → (pf, sf↓pf, sf(pf), c) Ftf
⇔
T(F) → (pf, sf↑(max(0,pf−hf'),min(pf,hf')), sf(pf), c) Ftf.

T(F)=next(TV(X)). By the definition of → for next(TV(X)), Ftf in [1.5] depends only whether X ∈ dom(c.1), which is the same in both sides if the equivalence. Hence, [1.5] holds.
CASE 2. $F = \sim F_1$. $T(F) = \text{next}(TN(T(F1)))$.

We take $\text{ref}, F_t, p_f, q_f, s_f, h_f, d_f, e_f$ arbitrary but fixed. Assume

(2.1) $\vdash (\text{ref} \vdash F : (h_f, d_f))$
(2.1') $q_f \leq p_f$
(2.2) $\forall Y \in \text{dom}(e_f): \text{ref}(Y).1 + q_f \leq e_f(Y) \leq \text{ref}(Y).2 + q_f$

Define

(2.3) $c := (e_f, \{(X, s_f(e_f(X))) \mid X \in \text{dom}(e_f))\}$

Take $h_f'$ arbitrary but fixed. Assume

(2.4) $h_f' \geq h_f$

And prove

[2.5] $T(F) \rightarrow (p_f, s_f \downarrow p_f, s_f(p_f), c) F_t$ $\iff$

$T(F) \rightarrow (p_f, s_f \uparrow \max(0, p_f - h_f'), \min(p_f, h_f'))$, $s_f(p_f), c) F_t$.

From (2.1), by the definition of $\rightarrow$ for $\text{next}(TN(T(F1)))$, we get

(2.6) $\vdash (\text{ref} \vdash \sim F_1 : (h_f, d_f))$.

We prove [2.5] in both directions.

$(\Longrightarrow)$ We assume

(2.7) $T(\sim F_1) \rightarrow (p_f, s_f \downarrow p_f, s_f(p_f), c) F_t$

and prove

[2.8] $T(F) \rightarrow (p_f, s_f \uparrow \max(0, p_f - h_f'), \min(p_f, h_f'))$, $s_f(p_f), c) F_t$.

From (2.7), we prove [2.8] by case distinction over $F_t$:

C1. $F_t = \text{next}(TN(\text{next}(f'))) for some f' \in TFormulaCore, such that$

(2.8) $T(F_1) \rightarrow (p_f, s_f \downarrow p_f, s_f(p_f), c) \text{next}(f')$.

From (2.8), by (2.6), (2.1'), (2.2), (2.3), (2.4), and the induction hypothesis, we get

(2.9) $T(F_1) \rightarrow (p_f, s_f \uparrow \max(0, p_f - h_f'), \min(p_f, h_f'))$, $s_f(p_f), c) \text{next}(f')$.

From (2.9), by the definition of $\rightarrow$ for $T(\sim F)$, we get [2.8].

C2. $F_t = \text{done}(false)$. This happens when

(2.10) $T(F_1) \rightarrow (p_f, s_f \downarrow p_f, s_f(p_f), c) \text{done}(true)$.
From (2.10), by (2.6), (2.1'), (2.2), (2.3), (2.4), and the induction hypothesis, we get

\[(2.11) \ T(F1) \rightarrow (p_f, sf \uparrow (\max(0, p_f-h_f'), \min(p_f, h_f')), sf(p_f), c) \ \text{done(true)}.\]

From (2.11), by the definition of \(\rightarrow\) for \(T(F)\), we get [2.8].

C3. \(F_{tf}=\text{done(false)}\). Similar to the case C2.

\(\iff\) We assume

\[(2.12) \ T(\neg F) \rightarrow (p_f, sf \uparrow (\max(0, p_f-h_f'), \min(p_f, h_f')), sf(p_f), c) \ F_{tf}\]

and prove

\[2.13 \ T(\neg F1) \rightarrow (p_f, sf \downarrow p_f, sf(p_f), c) \ F_{tf}.\]

[2.13] can be proved by the same reasoning as the case \(\iff\) above. It finishes the proof of CASE2.

CASE 3. \(F = F_1 \& F_2\). \(T(F) = \text{next}(TCS(T(F_1), T(F_2))).\)

We take \(\text{ref}, F_{tf}, p_f, q_f, s_f, h_f, d_f, e_f\) arbitrary but fixed. Assume

\[(3.1) \vdash (\text{ref} \vdash F : (h_f, d_f))\]

\[(3.1') q_f \leq p_f\]

\[(3.2) \forall Y \in \text{dom}(e_f): \text{ref}(Y).1+q_f \leq e_f(Y) \leq \text{ref}(Y).2+q_f\]

Define

\[(3.3) c := (e_f, \{(X, s_f(e_f(X))) | X \in \text{dom}(e_f))\}\]

Take \(h_f' \in \mathbb{N}\) arbitrary but fixed. Assume

\[(3.4) h_f' \geq h_f\]

And prove

\[3.5 \ T(F) \rightarrow (p_f, s_f \downarrow p_f, s_f(p_f), c) \ F_{tf}\]

\(\iff\)

\[T(\neg F) \rightarrow (p_f, sf \uparrow (\max(0, p_f-h_f'), \min(p_f, h_f')), sf(p_f), c) \ F_{tf}.\]

From (3.1) and the assumption that \(h_f \in \mathbb{N}, d_f \in \mathbb{N}\), by the definition of \(\vdash\) for \(F_1 \& F_2\), there exist \(h_1, d_1, h_2, d_2 \in \mathbb{N}\) such that

\[(3.6) \vdash (\text{ref} \vdash F_1 : (h_1, d_1))\]

\[(3.7) \vdash (\text{ref} \vdash F_2 : (h_2, d_2))\]

\[(3.8) h_f = \max(h_1, h_2+d_1).\]

We prove [3.5] in both directions.

\(\implies\) We assume
(3.9) $T(F_1 \& F_2) \rightarrow (pf, sf|pf, sf(pf), c) F_{tf}$

and prove

[3.10] $T(F_1 \& F_2) \rightarrow (pf, sf|\max(0,pf-hf'), \min(pf,hf')), sf(pf), c) F_{tf}$.

From (3.9), we prove [3.10] by case distinction over $F_{tf}$:

C1. $F_{tf}=\text{next}(TCS(\text{next}(f_1), T(F_2)))$ for some $f_1 \in \text{TFormulaCore}$ such that

(3.11) $T(F_1) \rightarrow (pf, sf|pf, sf(pf), c) \text{next}(f_1)$.

From (3.11), by (3.6),(3.1'),(3.3),(3.6),(3.8), and the induction hypothesis, we get

(3.12) $T(F_1) \rightarrow (pf, sf|\max(0,pf-hf'), \min(pf,hf')), sf(pf), c) \text{next}(f')$.

From (3.12), by the definition of $\rightarrow$ for $T(F_1 \& F_2)$, we get [3.10].

C2. $F_{tf}=\text{done}(false)$. This happens when

(3.13) $T(F_1) \rightarrow (pf, sf|pf, sf(pf), c) \text{done}(false)$.

From (3.13), by (3.6),(3.1'),(3.2),(3.3),(3.4),(3.8), and the induction hypothesis, we get

(3.14) $T(F_1) \rightarrow (pf, sf|\max(0,pf-hf'), \min(pf,hf')), sf(pf), c) \text{done}(false)$.

From (3.14), by the definition of $\rightarrow$ for $T(F_1 \& F_2)$, we get [3.10].

C3. $F_{tf}=F_{t2}$ for some $F_{t2} \in \text{TFormula}$. This happens when we have

(3.15) $T(F_1) \rightarrow (pf, sf|pf, sf(pf), c) \text{done}(true)$ and
(3.16) $T(F_2) \rightarrow (pf, sf|pf, sf(pf), c) F_{t2}$.

From (3.4,3.8), we have

(3.17) $hf' \geq hf \geq h_1$
(3.18) $hf' \geq hf \geq h_2$

From (3.15), by (3.6),(3.1'),(3.2),(3.3),(3.17), we get

(3.19) $T(F_1) \rightarrow (pf, sf|\max(0,pf-hf'), \min(pf,hf')), sf(pf), c) \text{done}(true)$

From (3.16), (3.7),(3.1'),(3.2),(3.3),(3.18), we get

(3.20) $T(F_2) \rightarrow (pf, sf|\max(0,pf-hf'), \min(pf,hf')), sf(pf), c) F_{t2}$.

From (3.19) and (3.20), by the definition of $\rightarrow$ for $T(F_1 \& F_2)$, we get [3.10].

($\Leftarrow$) We assume

(3.21) $T(F_1 \& F_2) \rightarrow (pf, sf|\max(0,pf-hf'), \min(pf,hf')), sf(pf), c) F_{tf}$.
and prove

[3.22] $T(F_1 \& F_2) \rightarrow (pf, sf[pf], sf(pf), c) Ftf$

[3.22] can be proved by the same reasoning as the case ($\implies$) above. It finishes the proof of CASE3.

CASE 4. $F = F_1 \setminus F_2$. $T(F) = \text{next}(TCP(T(F_1), T(F_2)))$.

We take $\text{ref}, Ftf, pf, qf, sf, hf, df, ef$ arbitrary but fixed. Assume

(4.1) $\vdash (\text{ref} \vdash F : (hf, df))$
(4.1') $qf \leq pf$
(4.2) $\forall Y \in \text{dom}(ef): \text{ref}(Y).1 + qf \leq \text{ef}(Y) \leq \text{ref}(Y).2 + qf$

Define

(4.3) $c : = (\text{ef}, \{(X, sf(ef(X))) \mid X \in \text{dom}(ef)\})$

Take $hf'$ arbitrary but fixed. Assume

(4.4) $hf' \geq hf$

And prove

[4.5] $T(F) \rightarrow (pf, sf[pf], sf(pf), c) Ftf$
$\iff$
$T(F) \rightarrow (pf, sf[\max(0, pf-hf'), \min(pf, hf')], sf(pf), c) Ftf$

From (4.1) and the assumption that $hf, df \in \mathbb{N}$, by the definition of $\vdash$ for $F_1 \& F_2$, there exist $h_1, d_1, h_2, d_2 \in \mathbb{N}$ such that

(4.6) $\vdash (\text{ref} \vdash F_1 : (h_1, d_1))$
(4.7) $\vdash (\text{ref} \vdash F_2 : (h_2, d_2))$
(4.8) $hf = \max(h_1, h_2)$.

From (4.4, 4.8), we have

(4.9) $hf' \geq hf \geq h_1$
(4.10) $hf' \geq hf \geq h_2$

We prove [4.5] in both directions.

($\implies$) We assume

(4.11) $T(F_1 \setminus F_2) \rightarrow (pf, sf[pf], sf(pf), c) Ftf$

and prove

[4.12] $T(F_1 \setminus F_2) \rightarrow (pf, sf[\max(0, pf-hf'), \min(pf, hf')], sf(pf), c) Ftf$.

From (4.11), we prove [4.10] by case distinction over $Ftf$: 55
C1. \( F_{tf} = \text{next}(TCS(\text{next}(f_1),\text{next}(f_2))) \) for some \( f_1, f_2 \in T\text{FormulaCore} \) such that

\begin{align*}
(4.13) \ T(F_1) &\rightarrow (pf, sf_{\downarrow} pf, sf(pf), c) \text{ next}(f_1). \\
(4.14) \ T(F_2) &\rightarrow (pf, sf_{\downarrow} pf, sf(pf), c) \text{ next}(f_2).
\end{align*}

From (4.13), by (4.6), (4.1'), (4.3), (4.9), and the induction hypothesis, we get

\begin{align*}
(4.15) \ T(F_1) &\rightarrow (pf, sf_{\uparrow} (\max(0, pf-hf'), \min(pf, hf'))), sf(pf), c) \text{ next}(f_1).
\end{align*}

From (4.14), by (4.7), (4.1'), (4.3), (4.10), and the induction hypothesis, we get

\begin{align*}
(4.16) \ T(F_2) &\rightarrow (pf, sf_{\uparrow} (\max(0, pf-hf'), \min(pf, hf'))), sf(pf), c) \text{ next}(f_2).
\end{align*}

From (4.15, 4.16), by the definition of \( \rightarrow \) for \( T(F_1 \land F_2) \), we get \([4.12]\).

C2. \( F_{tf} = \text{next}(f_1) \) for some \( f_1 \in T\text{FormulaCore} \) such that

\begin{align*}
(4.17) \ T(F_1) &\rightarrow (pf, sf_{\downarrow} pf, sf(pf), c) \text{ next}(f_1).
(4.18) \ T(F_2) &\rightarrow (pf, sf_{\downarrow} pf, sf(pf), c) \text{ done(true)}.
\end{align*}

By the same reasoning as in C1 above we get that \([4.12]\) holds.

C3. \( F_{tf} = \text{done(false)} \). This happens in one of the following possible cases:

C3.1

\begin{align*}
(4.19) \ T(F_1) &\rightarrow (pf, sf_{\downarrow} pf, sf(pf), c) \text{ next}(f_1).
(4.20) \ T(F_2) &\rightarrow (pf, sf_{\downarrow} pf, sf(pf), c) \text{ done(false)}.
\end{align*}

By the same reasoning as in C1 above we get that \([4.12]\) holds.

C3.2

\begin{align*}
(4.21) \ T(F_1) &\rightarrow (pf, sf_{\downarrow} pf, sf(pf), c) \text{ done(false)}.
\end{align*}

From (4.21), by (4.6), (4.1'), (4.3), (4.9), and the induction hypothesis, we get

\begin{align*}
(4.22) \ T(F_1) &\rightarrow (pf, sf_{\uparrow} (\max(0, pf-hf'), \min(pf, hf'))), sf(pf), c) \text{ done(false)}.
\end{align*}

From (4.22), by the definition of \( \rightarrow \) for \( T(F_1 \land F_2) \), we get \([4.12]\).

C4. \( F_{tf} = F_{t2} \) for some \( F_{t2} \in T\text{Formula} \). This happens when

\begin{align*}
(4.23) \ T(F_1) &\rightarrow (pf, sf_{\downarrow} pf, sf(pf), c) \text{ done(true)}.
(4.24) \ T(F_2) &\rightarrow (pf, sf_{\downarrow} pf, sf(pf), c) \text{ Ft2}.
\end{align*}

By the same reasoning as in C1 above we get that \([4.12]\) holds.

\( \iff \) We assume

\begin{align*}
(4.25) \ T(F_1 \land F_2) &\rightarrow (pf, sf_{\uparrow} (\max(0, pf-hf'), \min(pf, hf'))), sf(pf), c) \text{ Ftf}.
\end{align*}
and prove

\[ T(F_1 \land F_2) \rightarrow (pf, sf|pf, sf(pf), c) Ftf \]

[4.26] can be proved by the same reasoning as the case \( \Longrightarrow \) above.

It finishes the proof of CASE 4.

CASE 5. \( F = \forall X \in B_1..B_2: F_1 \) \( T(F) = \text{next}(TA(X, T(B_1), T(B_2), T(F_1))) \).

We take \( F_{tf}, pf, qf, hf, df, ef \) arbitrary but fixed. Assume

(5.1) \( \vdash (\text{ref} \vdash F : (hf, df)) \)

(5.1') \( qf \leq pf \)

(5.2) \( \forall Y \in \text{dom}(ef): \text{ref}(Y)\cdot 1 + pf \leq ef(Y) \leq \text{ref}(Y)\cdot 2 + pf \)

Define

(5.3) \( cf := (ef, \{(X, sf(ef(X))) \mid X \in \text{dom}(ef)\}) \)

Take \( hf' \) arbitrary but fixed. Assume

(5.4) \( hf' \geq hf \)

And prove

\[ T(F) \rightarrow (pf, sf|pf, sf(pf), cf) Ftf \]

\[ \iff \]

\[ T(F) \rightarrow (pf, sf^{\uparrow}(\max(0, pf-hf'), \min(pf, hf')), sf(pf), cf) Ftf \]

Let \( b_1, b_2 \in \text{BoundValue} \) and \( Ft_1 \in \text{TFormula} \) be such that

(5.6) \( b_1 = T(B_1) \)

(5.7) \( b_2 = T(B_2) \)

(5.7') \( Ft_1 = T(F_1) \)

We prove [5.5] in both directions.

\( \Longrightarrow \) We assume

(5.8) \( \text{next}(TA(X, b_1, b_2, Ft_1)) \rightarrow (pf, sf|pf, sf(pf), cf) Ftf \)

and prove

\[ \text{next}(TA(X, b_1, b_2, Ft_1)) \]

\[ \rightarrow (pf, sf^{\uparrow}(\max(0, pf-hf'), \min(pf, hf')), sf(pf), cf) Ftf. \]

CASE 1:

(5.10) \( Ftf = \text{done(true)} \) with \( b_1(cf) = \infty \).

------------------------------------

\( b_1(cf) = \infty \) imply [5.9].
CASE 2:
(5.13) Ftf is arbitrary
---------------------

From (5.8), by the definition of $\rightarrow$ for forall, there exists $p_1, p_2, TA_0'$ such that

(5.14) $p_1 = b_1(cf)$
(5.15) $p_2 = b_2(cf)$
(5.16) $p_1 \neq \infty$
(5.17) $next(TA_0(X, p_1, p_2, Ft_1)) \rightarrow (pf, sf \downarrow pf, sf(pf), cf) TA_0'$.

To prove [5.9], we should find such $p_1^*, p_2^*, TA_0'^*$ that

[5.18] $p_1^* = b_1(cf)$
[5.19] $p_2^* = b_2(cf)$
[5.20] $p_1^* \neq \infty$
[5.21] $next(TA_0(X, p_1^*, p_2^*, Ft_1)) \rightarrow (pf, sf \uparrow (max(0, pf-hf'), min(pf, hf')), sf(pf), cf) TA_0'^*$.

We take $p_1^* = p_1$, $p_2^* = p_2$, $TA_0'^* = TA_0'$. Then [5.18-5.20] follow from (5.14-5.16) and we need to prove only

[5.22] $next(TA_0(X, p_1, p_2, Ft_1)) \rightarrow (pf, sf \uparrow (max(0, pf-hf'), min(pf, hf')), sf(pf), cf) TA_0'$.

Subcase 1.
(5.23) $pf < p_1$.
---------------------

In this case from (5.17) we have $TA_0' = next(TA_0(X, p_1, p_2, Ft_1))$. Then [5.22] follows from the definition of $\rightarrow$ for forall.

Subcase 2.
(5.24) $pf \geq p_1$.
---------------------

We introduce the notation:

(5.25) $ms := sf \uparrow (max(0, pf-hf'), min(pf, hf'))$

By definition of $\rightarrow$, to prove [5.22], we need to prove

[5.26] $next(TA_1(X, p_2, Ft_1, fs)) \rightarrow (pf, ms, sf(pf), cf) TA_0'$,

where

(5.27) $fs = \{(p_0, Ft_1, (cf.1[X \rightarrow p_0], c.2[X \rightarrow ms(p_0+pf-\|ms\|)])) | p_1 \leq p_0 < \infty \min\\{pf, p_2+\infty\1\}$.

We prove [5.26] by case distinction over $TA_0'$ from (5.17).

(c1) $TA_0' = done(false)$
---------------------

We prove
To prove [c1.1], by Def. \( \rightarrow \) we need to prove

\[
[c1.2] \exists t \in \mathbb{N}, g \in TFormula, c \in Context:
    (t, g, c) \in fs_0 \wedge \vdash g \rightarrow (pf, ms, sf(pf), c) \text{ done(false)},
\]

where

\[
[c1.3] fs_0 = \begin{cases} \text{if } pf > \infty \text{ then } fs \text{ else } \cup \{(pf, F_t, (cf.1[X \mapsto pf], cf.2[X \mapsto sf(pf)]))} \end{cases}
\]

On the other hand, from (5.17) by (c1) we know

\[
[c1.4] \text{next(TA1}(X, p_2, F_t, fs') \rightarrow (pf, sf_{\downarrow} pf, sf(pf), c) \text{ done(false)}
\]

where (since \( p_0 + pf - |sf_{\downarrow} pf| = p_0 \))

\[
[c1.5] fs' = \{(p_0, F_t, (cf.1[X \mapsto p_0], cf.2[X \mapsto (sf_{\downarrow} pf)(p_0)])) | p_1 \leq p_0 < \infty \min\infty(pf, p_2 + \infty 1)\}.
\]

From (c1.4) we know

\[
[c1.6] \exists t \in \mathbb{N}, g \in TFormula, c \in Context:
    (t, g, c) \in fs_1 \wedge \vdash g \rightarrow (pf, sf_{\downarrow} pf, sf(pf), c) \text{ done(false)},
\]

where

\[
[c1.7] fs_1 = \begin{cases} \text{if } pf > \infty \text{ then } fs' \text{ else } fs' \cup \{(pf, F_t, (cf.1[X \mapsto pf], cf.2[X \mapsto sf(pf)]))} \end{cases}
\]

From (c1.6), take \((t_1, g_1, c_1)\) arbitrary but fixed such that

\[
[c1.8] (t_1, g_1, c_1) \in fs_1 \text{ and }
[c1.9] \vdash g_1 \rightarrow (pf, sf_{\downarrow} pf, sf(pf), c_1) \text{ done(false)}.
\]

From (c1.8), (c1.7), (c1.5) we see that

\[
[c1.10] g_1 = F_t
\]

and, hence, \( T(F_t) = g_1 \).

Let \( h_1, d_1 \in \mathbb{N} \) be such that

\[
[c1.11] (\text{ref } \vdash F_t : (h_1, d_1))
\]

From (c1.8), (c1.7), (c1.5), we have

\[
[c1.12] c_1 = (cf.1[X \mapsto t_1], cf.2[X \mapsto (sf_{\downarrow} pf)(t_1)]).
\]

Note that

\[
[c1.12] t_1 \leq pf
\]
and for all \(Y \in \text{dom}(\text{cf.}1[X \rightarrow t1])\), we have

\[(c1.13) \text{ ref}(Y).1 + t1 \leq \text{cf.}1[X \rightarrow t1](Y) \leq \text{ref}(Y).2 + t1\]

We apply the induction hypothesis with \(G=Ft1\), \(Ft=\text{done}(false)\), \(re=\text{ref}\), \(p=pf\), \(q=t1\), \(s=\text{sf}\), \(h=hg\), \(d=dg\), \(e=\text{cf.}1[X \rightarrow t1]\). From (c1.11) and (c1.13), and by defining \(c\) as \(c1\) in (c1.12), we obtain

\[(c1.14) \forall h1' \in \mathbb{N} : h1' \geq h1 \Rightarrow \\
Ft1 \rightarrow (pf, \text{sf}\downarrow(pf), \text{sf}(pf), c1) \text{ done}(false) \Rightarrow \\
Ft1 \rightarrow (pf, \text{sf}\uparrow(\max(0,pf-h1'),\min(pf,h1')), s, s(pf), c1) \text{ done}(false)\]

Since (c1.14) is true for all \(h1' \geq h1\), it is true, in particular, for \(hf'\), because by (5.4) we have \(hf' \geq hf\), and in itself, \(hf \geq h1\) by the analysis rules for forall formulas. Hence, from (c1.14) we get

\[(c1.15) \\
Ft1 \rightarrow (pf, \text{sf}\downarrow(pf), \text{sf}(pf), c1) \text{ done}(false) \Rightarrow \\
Ft1 \rightarrow (pf, \text{sf}\uparrow(\max(0,pf-hf'),\min(pf,hf')), s, s(pf), c1) \text{ done}(false)\]

From (c1.15) and (c1.9) we get

\[(c1.16) Ft1 \rightarrow (pf, \text{sf}\uparrow(\max(0,pf-hf'),\min(pf,hf')), s, s(pf), c1) \text{ done}(false)\]

(c1.16), by (5.25), proves the second conjunct of [c1.2].

Hence, it remains to prove the first conjunct of [c1.2]:

\[[c1.3] (t1,g1,c1) \in fs0.\]

By (c1.8), \((t1,g1,c1) \in fs1\). By (c1.7) it means either

\[(c1.17) (t1,g1,c1) \in (pf,Ft1,(\text{cf.}1[X \rightarrow pf],\text{cf.}2[X \rightarrow sf(pf)])\]

or

\[(c1.18) (t1,g1,c1) \in fs'.\]

From (c1.17) we get [c1.3] due to the definition of fs0 in (c1.2).

From (c1.18) we have

\[(c1.19) (t1,g1,c1) \in (p0,Ft1,(\text{cf.}1[X \rightarrow p0],\text{cf.}2[X \rightarrow sf(pf)](p0)))\]

for some \(p0 \leq p0 < \infty \min(\infty pf,p2+\infty 1)\). Note that

\[(c1.20) (\text{sf}\downarrow pf)(p0) = \text{ms}(p0+pf-|ms|).\]

From (c1.20), (c1.19) and the definition of fs in (5.24) we get

\[(c1.21) (t1,g1,c1) \in fs.\]
From (c1.2) we have \( f_s \subseteq f_{s0} \) and, hence, [c1.3] holds also in this case.

(c2) TA0' = done(true)
---------------------
We prove

\[ [c2.1] \text{next}(TA1(X,p2,Ft1,fs)) \rightarrow (pf, ms, sf(pf), cf) \text{ done(true)}. \]

To prove [c2.1], by Def. \( \rightarrow \) we need to prove

\[ [c2.2] \lnot \exists t \in N, g \in TFormula, c \in \text{Context} : (t,g,c) \in f_{s0} \land \vdash g \rightarrow (pf,ms,sf(pf),c) \text{ done(false)} \]
\[ [c2.3] f_{s1} = \emptyset \land pf \geq \infty \]

where

(c2.4) \( f_{s0} = \)
\[ \text{if } pf > \infty \text{ then } f_{s} \text{ else } f_{s} \cup \{(pf,Ft1,(cf.1[X \rightarrow pf],cf.2[X \rightarrow sf(pf)]))\} \]
(c2.5) \( f_{s1} = \{ (t,\text{next}(fc),c) \in TInstance | \exists g \in TFormula : (t,g,c) \in f_{s0} \land \vdash g \rightarrow (pf,ms,sf(pf),c) \text{ next(fc) } \} \)

On the other hand, from (5.17) by (c2) we know

(c2.6) \( \text{next}(TA1(X,p2,Ft1,fs')) \rightarrow (pf,sf \downarrow pf,sf(pf),cf) \text{ done(true)} \)

where (since \( p_0 + pf - |sf \downarrow pf| = p_0 \))

(c2.7) \( f_{s'} = \{(p0,Ft1,(c.1[X \rightarrow p0],c.2[X \rightarrow sf(pf)](p0))) | p_1 \leq p_0 < \infty \min_{\infty}(pf,p_2+\infty 1)\} \)

From (c2.6), by Def. \( \rightarrow \) we know

(c2.8) \( \lnot \exists t \in N, g \in TFormula, c \in \text{Context} : (t,g,c) \in f_{s0'} \land \vdash g \rightarrow (pf,sf \downarrow pf,sf(pf),c) \text{ done(false)} \) and

(c2.9) \( f_{s1'} = \emptyset \land pf \geq \infty \)

where

(c2.10) \( f_{s0'} = \)
\[ \text{if } pf > \infty \text{ then } fs' \text{ else } fs' \cup \{(pf,f,(cf.1[X \rightarrow pf],cf.2[X \rightarrow sf(pf)]))\} \]
(c2.11) \( f_{s1'} = \{ (t,\text{next}(fc),c) \in TInstance | \exists g \in TFormula : (t,g,c) \in f_{s0'} \land \vdash g \rightarrow (pf,sf \downarrow pf,sf(pf),c) \text{ next(fc) } \} \)

Note that for all

(c2.12) \( \forall p_0: p_1 \leq p_0 < \infty \min_{\infty}(pf,p_2+\infty 1) \Rightarrow (sf \downarrow pf)(p_0) = ms(p_0 + pf - |ms|). \)

Therefore, from (5.24) and (c2.7) we get

(c2.13) \( f_s = f_{s'} \),

which, by (c2.4) and (c2.10), implies

61
To prove [c2.2], we take
(c2.15) \((t_0, g_0, c_0) \in \text{fs}_0\)

and prove that
[\text{c2.16}] \(g_0 \rightarrow (\text{pf}, \text{ms}, \text{sf}(\text{pf}), c_0) \text{ done(false)}\) does not hold.

From (c2.15) and (c2.14) we have
(c2.17) \((t_0, g_0, c_0) \in \text{fs}_0'\).

From (c2.17) and (c2.8) we know
(c2.18) \(g_0 \rightarrow (\text{pf}, \text{sf}_\downarrow \text{pf}, \text{sf}(\text{pf}), c_0) \text{ done(false)}\) does not hold.

From (c2.18), by the induction hypothesis, we get [c2.16]

To prove [c2.3], note that from (c2.11) and (c2.9) we have that
for all \((t,g,c) \in \text{fs}_0\)' we have that
\(\vdash g \rightarrow (\text{pf}, \text{sf}_\downarrow \text{pf}, \text{sf}(\text{pf}), c) \text{ next(fc)}\) does not hold, which, by (c2.14), is claimed for all \((t,g,c) \in \text{fs}_0\). It means, for each \((t,g,c) \in \text{fs}_0\) there exists \(b \in \text{Bool}\) such that
[\text{c2.19}] \(\vdash g \rightarrow (\text{pf}, \text{sf}_\downarrow \text{pf}, \text{sf}(\text{pf}), c) \text{ done(b)}\).

From (c2.19), by the induction hypothesis, we get that for each \((t,g,c) \in \text{fs}_0\) there exists \(b \in \text{Bool}\) such that
(c2.20) \(\vdash g \rightarrow (\text{pf}, \text{ms}, \text{sf}(\text{pf}), c) \text{ done(b)}\).

From (c2.20) we get
(c2.21) \(\text{fs}_1 = \emptyset\).

From (c2.21) and the second conjunct of (c2.9) we get [c2.3]

(c3) \(\text{TA}_0' = \text{next(\text{TA}_1(X,p_2,F_{t_1},\text{fs}'))}\)

--------------

We prove
[c3.1] \(\text{next(\text{TA}_1(X,p_2,F_{t_1},\text{fs}'))} \rightarrow (\text{pf, ms, sf}(\text{pf}), \text{cf}) \text{ next(\text{TA}_1(X,p_2,F_{t_1},\text{fs}'))}\).

To prove [c3.1], by Def. \(\rightarrow\) we need to prove
[c3.2] \(\neg \exists t \in \mathbb{N}, g \in \text{TFormula}, c \in \text{Context}: (t,g,c) \in \text{fs}_0 \land \vdash g \rightarrow (\text{pf}, \text{ms}, \text{sf}(\text{pf}), c) \text{ done(false)}\) and
[c3.3] \(\neg (\text{fs}_1 = \emptyset \land \text{pf} \geq \infty \ p_2)\)

where
(c3.4) \( fs_0 = \text{if } p_f > \infty \text{ then } fs \cup \{(p_f, f, (c.f.1[X \mapsto p], c.f.2[X \mapsto sf(p_f)]))\} \)

(c3.5) \( fs_1 = \{(t, \text{next}(f_c), c) \in T\text{Instance} \mid \\
\exists g \in T\text{Formula}: (t, g, c) \in fs_0 \land \vdash g \rightarrow (p, m_s, sf(p_f), c) \text{ next}(f_c)\} \)

On the other hand, from (5.17) by (c3) we know

(c3.6) \( \text{next}(TA_1(X, p_2, F_t_1, fs''')) \rightarrow (p_f, sf\downarrow p_f, sf(p_f), c) \text{ next}(TA_1(X, p_2, F_t_1, fs''')) \)

where (since \( p_0 + p_f - |sf\downarrow p_f| = p_0 \))

(c3.7) \( fs''' = \{(p_0, F_t_1, (c.1[X \mapsto p_0], c.2[X \mapsto sf(p_f)(p_0)])) \mid \\
p_1 \leq p_0 < \min(\infty, p_2 + 1)\} \)

From (c3.6), by Def. \( \rightarrow \) we know

(c3.8) \( \neg \exists t \in \mathbb{N}, g \in T\text{Formula}, c \in \text{Context}: \\
(t, g, c) \in fs_0''' \land \vdash g \rightarrow (p_f, sf\downarrow p_f, sf(p_f), c) \text{ done(false) and} \)
(c3.9) \( \neg (fs_1''' = \emptyset \land p_f \geq \infty p_2) \)

where

(c3.10) \( fs_0''' = \\
\text{if } p_f > \infty \text{ then } fs''' \cup \{(p_f, f, (c.f.1[X \mapsto p_f], c.f.2[X \mapsto sf(p_f)]))\} \)
(c3.11) \( fs_1''' = \{(t, \text{next}(f_c), c) \in T\text{Instance} \mid \\
\exists g \in T\text{Formula}: (t, g, c) \in fs_0''' \land \vdash g \rightarrow (p_f, sf\downarrow p_f, sf(p_f), c) \text{ next}(f_c)\} \).

[c3.2] can be proved analogously to [c2.2] above. The proof relies to the fact

(c3.12) \( fs_0''' = fs_0 \).

To prove [c3.3], we assume \( p_f < \infty p_2 \) and prove

[c3.13] \( fs_1 \neq \emptyset \).

From the assumption \( p_f < \infty p_2 \) and (c3.9) we obtain

(c3.14) \( fs_1''' \neq \emptyset \).

Then (c3.14) means that for some \( (t_1, g_1, c_1) \in fs_0''' \),

(c3.15) \( \vdash g_1 \rightarrow (p_f, sf\downarrow p_f, sf(p_f), c_1) \text{ next}(f_c) \).

By the induction hypothesis, from (c3.15) we get

(c3.16) \( \vdash g_1 \rightarrow (p_f, m_s, sf(p_f), c_1) \text{ next}(f_c) \).

Then (c3.16) proves [c3.13]

(\( \iff \)) This direction can be proved with the same reasoning as (\( \implies \)).

It finishes the proof of CASE 5.

It finishes the proof of Lemma 3.
A.6 Lemma 4: $n$-Step Reductions to $\text{done}$ Formulas for TN, TCS, TCP

Statement 1. TN Formulas.

$\forall F \in \text{Formula}, n \in \mathbb{N}, p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, F_t \in T\text{Formula} :$

$T(F) \rightarrow *(n,p,s,e) \text{ done(false)} \Rightarrow \text{next(TN(T(F)))} \rightarrow *(n,p,s,e) \text{ done(true)} \wedge$

$T(F) \rightarrow *(n,p,s,e) \text{ done(true)} \Rightarrow \text{next(TN(T(F)))} \rightarrow *(n,p,s,e) \text{ done(false)}$

Proof

We take $F_f$, $s_f$, $e_f$ arbitrary but fixed and prove the formula

$\forall n \in \mathbb{N}, p \in \mathbb{N} :$

$T(F_f) \rightarrow *(n,p,f,s_f,e_f) \text{ done(false)} \Rightarrow$

$\text{next(TN(T(F_f)))} \rightarrow *(n,p,f,s_f,e_f) \text{ done(true)}$\wedge

$T(F_f) \rightarrow *(n,p,f,s_f,e_f) \text{ done(true)} \Rightarrow$

$\text{next(TN(T(F_f)))} \rightarrow *(n,p,s,e) \text{ done(false)}$

by induction over $n$. Since $T(F_f)$ is a next formula, for $n=0$ the antecedents of both conjuncts are false and the statement is trivially true.

Assume

(TN.1) $\forall p \in \mathbb{N}:$

$T(F_f) \rightarrow *(n,p,s_f,e_f) \text{ done(false)} \Rightarrow$

$\text{next(TN(T(F_f)))} \rightarrow *(n,p,s_f,e_f) \text{ done(true)}$

(TN.2) $\forall p \in \mathbb{N} :$

$T(F_f) \rightarrow *(n,p,s_f,e_f) \text{ done(true)} \Rightarrow$

$\text{next(TN(T(F_f)))} \rightarrow *(n,p,s,e) \text{ done(false)}$

Prove

[TN.3] $\forall p \in \mathbb{N} :$

$T(F_f) \rightarrow *(n+1,p,s_f,e_f) \text{ done(false)} \Rightarrow$

$\text{next(TN(T(F_f)))} \rightarrow *(n+1,p,s_f,e_f) \text{ done(true)}$

and

[TN.4] $\forall p \in \mathbb{N} :$

$T(F_f) \rightarrow *(n+1,p,s_f,e_f) \text{ done(true)} \Rightarrow$

$\text{xnext(TN(T(F_f)))} \rightarrow *(n+1,p,s,e) \text{ done(false)}$

To prove [TN.3], we take $p_f$ arbitrary but fixed, assume

(TN.5) $T(F_f) \rightarrow *(n+1,p_f,s_f,e_f) \text{ done(false)}$

and prove

[TN.6] $\text{next(TN(T(F_f)))} \rightarrow *(n+1,p_f,s_f,e_f) \text{ done(true)}$

From (TN.5) by definition $\rightarrow*$ without history we know that there exists

$F_t \in T\text{Formula}$ such that

(TN.7) $T(F_f) \rightarrow (p_f,s_f,p_f,s_f(p_f),c) F_t$

(TN.8) $F_t \rightarrow *(n,p_f+1,s_f,e_f) \text{ done(false)}$
where \( c = (ef, \{ (X, sf(ef(X))) \mid X \in \text{dom}(ef) \}) \).

We proceed by case distinction over \( Ft \).

Case 'next': If \( Ft \) is a next formula, then there exists \( F1 \in \text{Formula} \) such that

\[
\text{(TN.9) } Ft = T(F1)
\]

From (TN.9) and (TN.8) by (TN.1) we get

\[
\text{(TN.10) } \quad \text{next}(T(T(F1))) \rightarrow \ast(n, pf + 1, sf, ef) \text{ done(true)}
\]

From (TN.7) by the definition of \( \rightarrow \) we get

\[
\text{(TN.11) } \quad \text{next}(T(T(Ff))) \rightarrow (pf, sf \downarrow pf, sf(pf), c) \text{ next}(T(T(F1)))
\]

From (TN.11) and (TN.10) by the definition of \( \rightarrow \ast \) without history we get \([\text{TN.6}]\).

Case 'done': If \( Ft \) is a 'done' formula, then by (TN.8), we have

\[
\text{(TN.12) } \quad n = 0 \text{ and } \quad \text{(TN.13) } Ft = \text{done(false)}.
\]

From (TN.7) and (TN.13), by the definition of \( \rightarrow \), we get

\[
\text{(TN.14) } \quad \text{next}(T(T(Ff))) \rightarrow (pf, sf \downarrow pf, sf(pf), c) \text{ done(true)}.
\]

On the other hand, from the definition of \( \rightarrow \ast \) we know

\[
\text{(TN.15) } \quad \text{done(true)} \rightarrow \ast(0, pf + 1, sf, ef) \text{ done(true)}.
\]

From (TN.14), (TN.15), (TN.12), by the definition of \( \rightarrow \ast \) we get \([\text{TN.6}]\).

Hence, we proved \([\text{TN.6}]\) for both cases of \( Ft \). This proves \([\text{TN.3}]\). 

\([\text{TN.4}]\) can be proved analogously.

**Statement 2. TCS Formulas.**

\( \forall n \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment} : \)

\( \forall Ft_1, Ft_2 \in \text{Formula}, n \in \mathbb{N}, \)

\[ n > 0 \land Ft_1 \rightarrow \ast(n, p, s, e) \text{ done(false)} \Rightarrow \]

\( \text{next}(\text{TCS}(Ft_1, Ft_2)) \rightarrow \ast(n, p, s, e) \text{ done(false)} \land \)

\( \forall Ft_1, Ft_2 \in \text{Formula}, n_1, n_2 \in \mathbb{N}, b \in \text{Bool} : \)

\[ n_1 > 0 \land n_2 > 0 \land Ft_1 \rightarrow \ast(n_1, p, s, e) \text{ done(true)} \land Ft_2 \rightarrow \ast(n_2, p, s, e) \text{ done(b)} \Rightarrow \]

\( \text{next}(\text{TCS}(Ft_1, Ft_2)) \rightarrow \ast(\max(n_1, n_2), p, s, e) \text{ done(b)} \)

**Proof**

\[ \quad \]

We split the statement in two:

\([\text{TCS1}] \) \( \forall n \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, Ft_1, Ft_2 \in \text{Formula}, n \in \mathbb{N} : \)

\[ n > 0 \land Ft_1 \rightarrow \ast(n, p, s, e) \text{ done(false)} \Rightarrow \]
next(TCS(Ft1,Ft2)) \rightarrow *(n,p,s,e) \text{ done(false)}

[TCS2] \forall p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, Ft1,Ft2 \in \text{TFormula}, n1,n2 \in \mathbb{N}, b \in \text{Bool}:
\begin{align*}
n1 > 0 & \land n2 > 0 \land Ft1 \rightarrow *(n1,p,s,e) \text{ done(true)} \land Ft2 \rightarrow *(n2,p,s,e) \text{ done(b)} \Rightarrow \\
\text{next(TCS(Ft1,Ft2))} & \rightarrow *(\text{max}(n1,n2),p,s,e) \text{ done(b)}.
\end{align*}

Proof of [TCS1]
-------------

We take sf,ef arbitrary but fixed and define

\Phi(n) :\iff \\
\forall p \in \mathbb{N}, Ft1,Ft2 \in \text{TFormula}:
\begin{align*}
n > 0 & \land Ft1 \rightarrow *(n,p,sf,ef) \text{ done(false)} \Rightarrow \\
\text{next(TCS(Ft1,Ft2))} & \rightarrow *(n,p,sf,ef) \text{ done(false)}
\end{align*}

We prove \forall n \in \mathbb{N}: \Phi(n) by induction over n. For n=0 the formula is trivially true. We start the induction from 1. Prove:

[TCS1.a] \Phi(1) and 
[TCS1.b] \forall n \in \mathbb{N}: \Phi(n) \Rightarrow \Phi(n+1)

Proof of [TCS1.a]
-------------

We take pf,Ft1f,Ft2f arbitrary but fixed and assume

(TCS1.1) 1>0
(TCS1.2) Ft1f \rightarrow *(1,pf,sf,ef) \text{ done(false)}.

We want to prove

[TCS1.3] next(TCS(Ft1f,Ft2f)) \rightarrow *(1,pf,sf,ef) \text{ done(false)}.

From (TCS1.2), by the definition of \rightarrow* without history, there exists Ft \in \text{TFormula} such that

(TCS1.4) Ft1f \rightarrow (p,sf\uparrow pf, sf(pf), c) Ft and 
(TCS1.5) Ft \rightarrow *(0,pf+1, sf, ef) \text{ done(false)}

where

(TCS1.6) c=\{ef, \{(X,sf(ef(X))) | X \in \text{dom}(ef)}\}.

From (TCS1.5), by the definition of \rightarrow* without history, we get

(TCS1.7) Ft=\text{done(false)}.

From (TCS1.7) and (TCS1.4), by the definition of \rightarrow for TCS, we get

(TCS1.8) next(TCS(Ft1f,Ft2f)) \rightarrow (p,sf\uparrow pf, sf(pf), c) done(false).

From (TCS1.8, TCS1.7, TCS1.6), by the definition of \rightarrow* without history, we get [TCS1.2].
This finishes the proof of [TCS1.a]

Proof of [TCS1.b]
---------------
We take $n$ arbitrary but fixed, assume

\[(TCS1.8) \quad \forall p \in \mathbb{N}, \ F_t1, F_t2 \in T_{Formula}:
\begin{align*}
&\quad n>0 \land F_t1 \rightarrow (n,p,sf,ef) \text{ done(false)} \Rightarrow \\
&\quad \text{next(TCS}(F_t1,F_t2)) \rightarrow (n,p,sf,ef) \text{ done(false)}
\end{align*}
\]

and prove

\[(TCS1.9) \quad \forall p \in \mathbb{N}, \ F_t1, F_t2 \in T_{Formula}:
\begin{align*}
&\quad n+1>0 \land F_t1 \rightarrow (n+1,p,sf,ef) \text{ done(false)} \Rightarrow \\
&\quad \text{next(TCS}(F_t1,F_t2)) \rightarrow (n+1,p,sf,ef) \text{ done(false)}.
\end{align*}
\]

To prove [TCS1.9], we take $p_f, F_t1_f, F_t2_f$ arbitrary but fixed, assume

\[(TCS1.10) \quad n+1>0
\]
\[(TCS1.11) \quad F_t1_f \rightarrow (n+1,p_f,sf,ef) \text{ done(false)}
\]

and prove

\[(TCS1.12) \quad \text{next(TCS}(F_t1_f,F_t2_f)) \rightarrow (n+1,p,sf,ef) \text{ done(false)}.
\]

From (TCS1.11), by the definition of $\rightarrow$ without history, there exists $F_t \in T_{Formula}$ such that

\[(TCS1.13) \quad F_t1_f \rightarrow (p_f,sf\downarrow p_f, sf(p_f),c) \ F_t
\]
\[(TCS1.14) \quad F_t \rightarrow (n,p_f+1,sf,ef) \text{ done(false)}
\]

where

\[(TCS1.15) \quad c=(ef, \{(X,sf(ef(X)))| X \in \text{dom}(ef)}).
\]

We proceed by case distinction over $F_t$.

Case 1. $F_t=\text{next}(f)$ for some $f \in T_{FormulaCore}$
----------------------------------------------------------
From (TCS1.13), by the definition of $\rightarrow$ for TCS, we get

\[(TCS1.16) \quad \text{next(TCS}(F_t1_f,F_t2_f)) \rightarrow (p_f,sf\downarrow pf, sf(p_f),c) \ \text{next(TCS}(F_t,F_t2_f))
\]

Since $F_t$ is a 'next' formula, we have

\[(TCS1.17) \quad n>0.
\]

From (TCS1.17) and (TCS1.14), by the induction hypothesis (TCS1.8) we get

\[(TCS1.18) \quad \text{next(TCS}(F_t,F_t2_f)) \rightarrow (n,p_f+1,sf,ef) \text{ done(false)}
\]

From (TCS1.10), (TCS1.15), (TCS1.16), and (TCS1.18), by the definition of $\rightarrow$
without history, we get [TCS1.12]

Case 2. \( F_t = \text{done}(b) \) for some \( b \in \text{Bool} \)

In this case we have

(TCS1.19) \( n = 0 \) (a 'done' formula can be reduced only in 0 steps)
(TCS1.20) \( b = \text{false} \).

Then from (TCS1.13) and (TCS1.20), by the definition of \( \rightarrow \) for TCS we get

(TCS1.21) \( \text{next}(\text{TCS}(F_{t1}, F_{t2})) \rightarrow (p_f, sf, pf, sf, pf, c) \text{ done}(\text{false}) \).

From (TCS1.14), (TCS1.19), and (TCS1.20), we have

(TCS1.22) \( \text{done}(\text{false}) \rightarrow *(0, pf+1, sf, ef) \text{ done}(\text{false}) \).

From (TCS1.19), (TCS1.15), (TCS1.21), (TCS1.22), by the definition of \( \rightarrow * \) without history, we get [TCS1.12].

This finishes the proof of [TCS1].

Proof of [TCS2]

Recall

[TCS2] \( \forall s \in \text{Stream}, e \in \text{Environment}, p \in \mathbb{N}, F_{t1}, F_{t2} \in \text{Formula}, n_1, n_2 \in \mathbb{N}, b \in \text{Bool}: \)
\[ n_1 > 0 \land n_2 > 0 \land F_{t1} \rightarrow *(n_1, p, s, e) \text{ done}(\text{true}) \land F_{t2} \rightarrow *(n_2, p, s, e) \text{ done}(b) \Rightarrow \]
\[ \text{next}(\text{TCS}(F_{t1}, F_{t2})) \rightarrow *(\text{max}(n_1, n_2), p, s, e) \text{ done}(b). \]

We take \( sf, ef, bf \) arbitrary but fixed and define

\( \Phi(n_1) : \)
\[ \forall p \in dsN, F_{t1}, F_{t2} \in \text{Formula}, n_2 \in \mathbb{N} : \]
\[ n_1 > 0 \land n_2 > 0 \land F_{t1} \rightarrow *(n_1, p, sf, ef) \text{ done}(\text{true}) \land F_{t2} \rightarrow *(n_2, p, sf, ef) \text{ done}(bf) \Rightarrow \]
\[ \text{next}(\text{TCS}(F_{t1}, F_{t2})) \rightarrow *(\text{max}(n_1, n_2), p, sf, ef) \text{ done}(bf). \]

We need to prove \( \forall n_1 \in \mathbb{N}: \Phi(n_1) \). We use induction. Prove:

[TCS2.a] : \( \Phi(1) \)
[TCS2.b] \( \forall n_1 \in \mathbb{N}: \Phi(n_1) \Rightarrow \Phi(n_1+1) \).

Proof of [TCS2.a]

We need to prove

\( \forall n_2, p \in dsN, F_{t1}, F_{t2} \in \text{Formula} : \)
\[ 1 > 0 \land n_2 > 0 \land F_{t1} \rightarrow *(1, p, sf, ef) \text{ done}(\text{true}) \land F_{t2} \rightarrow *(n_2, p, sf, ef) \text{ done}(bf) \Rightarrow \]
\[ \text{next}(\text{TCS}(F_{t1}, F_{t2})) \rightarrow *(\text{max}(1, n_2), p, sf, ef) \text{ done}(bf). \]
We take \( n_2, p, F_{t1}, F_{t2} \) arbitrary but fixed. Assume

(TCS1.a.1) \( n_2 > 0 \)
(TCS1.a.2) \( F_{t1} \rightarrow^*(1, p, s, e) \) done(true)
(TCS1.a.3) \( F_{t2} \rightarrow^*(n_2, p, s, e) \) done(bf)

and prove

[TCS1.a.4] \( \text{next}(TCS(F_{t1}, F_{t2})) \rightarrow^*(\max(1, n_2), p, s, e) \) done(bf).

From (TCS1.a.2), by the definition of \( \rightarrow^* \), we have for some \( F_{t'} \)

(TCS1.a.5) \( F_{t1} \rightarrow(p, s, p, s, p(f), c) \) \( F_{t'} \)
(TCS1.a.6) \( F_{t'} \rightarrow^*(0, p+1, s, e) \) done(true)

where

(TCS1.a.7) \( c = (e, \{ (X, s, e(X)) \mid X \in \text{dom}(e) \}) \).

From (TCS1.a.6), by the definition \( p \rightarrow^* \), we know

(TCS1.a.8) \( F_{t'} \) = done(true).

From (TCS1.a.5) and (TCS1.a.8) we have

(TCS1.a.9) \( F_{t1} \rightarrow(p, s, p, s, p(f), c) \) done(true).

From (TCS1.a.3), by the definition of \( \rightarrow^* \), we have for some \( F_{t''} \)

(TCS1.a.10) \( F_{t2} \rightarrow(p, s, p, s, p(f), c) \) \( F_{t''} \)
(TCS1.a.11) \( F_{t''} \rightarrow^*(n_2-1, p+1, s, e) \) done(bf),

where \( c \) is defined as in (TCS1.a.7).

From (TCS1.a.9) and (TCS1.a.10), by the definition of \( \rightarrow^* \) for TCS, we have

(TCS1.a.13) \( \text{next}(TCS(F_{t1}, F_{t2})) \rightarrow (p, s, p, s, p(f), c) \) \( F_{t''} \).

From (TCS1.a.13), (TCS1.a.7), and (TCS1.a.11), by the definition of \( \rightarrow^* \), we have

(TCS1.a.14) \( \text{next}(TCS(F_{t1}, F_{t2})) \rightarrow (n_2, p, s, e) \) done(bf).

From (TCS1.a.1), we have \( n_2 = \max(1, n_2) \). Therefore, (TCS1.a.14) proves [TCS1.a.4]

This finishes the proof of [TCS2.a].

Proof of [TCS2.b]

We take \( n_1 \) arbitrary but fixed. Assume \( \Phi(n_1) \), i.e.,

(TCS2.b.1) \( \forall n_2, p \in \mathbb{N}, F_{t1}, F_{t2} \in TFormula : \)
\( n_1 > 0 \land n_2 > 0 \land F_{t1} \rightarrow^*(n_1, p, s, e) \) done(true) \( \land \)

---
\[ \text{Ft2} \rightarrow (n2, p, sf, ef) \text{ done(bf)} \]
\[ \Rightarrow \text{next(TCS(Ft1, Ft2))} \rightarrow (\max(n1, n2), p, sf, ef) \text{ done(bf)}. \]

and prove

\[ [\text{TCS2.b.2}] \forall n2,p \in \text{dsN}, \text{Ft1, Ft2} \in \text{TFormula} : \]
\[ n1+1 > 0 \land n2 > 0 \land \text{Ft1} \rightarrow (n1+1, p, sf, ef) \text{ done(true)} \land \]
\[ \text{Ft2} \rightarrow (n2, p, sf, ef) \text{ done(bf)} \]
\[ \Rightarrow \text{next(TCS(Ft1, Ft2))} \rightarrow (\max(n1+1, n2), p, sf, ef) \text{ done(bf)}. \]

To prove \[TCS2.b.2\], we take \( n2, p, \text{Ft1f, Ft2f} \) arbitrary but fixed. Assume

\[ (\text{TCS2.b.3}) \ n1+1 > 0 \]
\[ (\text{TCS2.b.4}) \ n2 > 0 \]
\[ (\text{TCS2.b.5}) \ \text{Ft1f} \rightarrow (n1+1, p, sf, ef) \text{ done(true)} \]
\[ (\text{TCS2.b.6}) \ \text{Ft2f} \rightarrow (n2, p, sf, ef) \text{ done(bf)} \]

and prove

\[ [\text{TCS2.b.7}] \text{next(TCS(Ft1f, Ft2f))} \rightarrow (\max(n1+1, n2), p, sf, ef) \text{ done(bf)}. \]

From \( \text{TCS2.b.5} \), by the definition of \( \rightarrow * \), we have for some \( \text{Ft'} \)

\[ (\text{TCS2.b.8}) \ \text{Ft1f} \rightarrow (p, sf, p, sf(p), c) \text{ Ft'} \]
\[ (\text{TCS2.b.9}) \ \text{Ft'} \rightarrow (n1, p+1, sf, ef) \text{ done(true)} \]

where

\[ (\text{TCS2.b.10}) \ c = (ef, \{(X, sf(ef(X))) : X \in \text{dom}(ef)})]. \]

From \( \text{TCS2.b.6} \), by the definition of \( \rightarrow * \), we have for some \( \text{Ft''} \)

\[ (\text{TCS2.b.11}) \ \text{Ft2f} \rightarrow (p, sf, p, sf(p), c) \text{ Ft''} \]
\[ (\text{TCS2.b.12}) \ \text{Ft''} \rightarrow (n2-1, p+1, sf, ef) \text{ done(bf)}, \]

where \( c \) is defined as in \( \text{TCS2.b.10} \).

Case \( n1 = 0 \)
---------
In this case we have \( \text{Ft'} = \text{done(true)} \) and from \( \text{TCS2.b.8} \) we get

\[ (\text{TCS2.b.13}) \ \text{Ft1f} \rightarrow (p, sf, p, sf(p), c) \text{ done(true)}. \]

From \( \text{TCS2.b.13} \) and \( \text{TCS2.b.11} \), by the definition of \( \rightarrow \) for TCS, we have

\[ (\text{TCS2.b.14}) \ \text{next(TCS(Ft1f, Ft2f))} \rightarrow (p, sf, p, sf(p), c) \text{ Ft''}. \]

From \( \text{TCS2.b.4}, \ \text{TCS2.b.10}, \ \text{TCS2.b.14}, \ \text{TCS2.b.12} \) by the definition of \( \rightarrow * \), we get

\[ (\text{TCS2.b.15}) \ \text{next(TCS(Ft1f, Ft2f))} \rightarrow (n2, p, sf, ef) \text{ done(bf)}. \]
By (TCS2.b.4) and n1=0, we have n2=max(1,n2)=max(n1+1,n2).
Hence, (TCS2.b.16) proves [TCS2.b.7].

Case n1>0, n2-1>0
-----------------
In this case F_t'='next(f') for some f'∈TFormulaCore.
Therefore, from (TCS3.b.8), by the definition of → for TCS we have
(TCS2.b.16) next(TCS(F_t1,F_t2')) (pf, sf↓ pf, sf(pf), c) next(TCS(F_t',F_t2f)).
Since n2-1>0 and, hence, n2>0, from (TCS2.b.6) by the Shifting Lemma 7 we get
(TCS2.b.17) F_t2f → *(n2-1, pf+1, sf, ef) done(bf)
From n1>0, n2-1>0, (TCS2.b.9), (TCS2.b.17), by the induction hypothesis
(TCS2.b.1) we get
(TCS2.b.18) next(TCS(F_t',F_t2f)) → *(max(n1,n2-1), pf+1, sf, ef) done(bf)
From max(n1,n2-1)+1>0, (TCS2.b.10), (TCS2.b.16), (TCS2.b.18) we get
(TCS2.b.18) next(TCS(F_t1f,F_t2f)) → *(max(n1,n2-1)+1, pf, sf, ef) done(bf)
Since max(n1,n2-1)+1=max(n1+1,n2), (TCS2.b.18) proves [TCS2.b.7]

Case 2. n1>0, n2-1=0
----------------------
In this case from (TCS2.b.12) we have F_t''=done(bf), which from (TCS2.b.12) gives
(TCS2.b.19) F_t2f → (pf, sf↓ pf, sf(pf), c) done(bf).
From (TCS2.b.5), by Lemma 2, we have
(TCS2.b.23) F_t1f → l*(n1+1, pf, sf, ef) done(true).
From (TCS2.b.23), by the definition of → l*, we obtain for some F_t0
(TCS2.b.24) F_t1f → l*(n1, pf, sf, ef) F_t0
(TCS2.b.25) F_t0 → (pf+n1, sf↓ (pf+n1), s(pf+n1), c) done(true),
where c is defined as in (TCS2.b.10).
From (TCS2.b.19), by the Lemma 6, we have
(TCS2.b.26) F_t2f → (pf+n1, sf↓ (pf+n1), sf(pf+n1), c) done(bf).
From (TCS2.b.25) and (TCS2.b.26), by the definition of → for TCS, we get
(TCS2.b.27) next(TCS(F_t0,F_t2f)) → (pf+n1, sf↓ (pf+n1), sf(pf+n1), c) done(bf).
From (TCS2.b.24), by Lemma 2 we have
Moreover, (TCS2.b.23) implies that $F_{t1f}$ is not a 'done' formula. Also, from (TCS2.b.25) since $p_f + n_1 > 0$ due to $n_1 > 0$, we have that $F_{t0}$ is a 'next' formula. Hence, there exists $f_0 \in T_{FormulaCore}$ such that

(TCS2.b.29) $F_{t0} = next(f_0)$

and from (TCS2.b.28) we have

(TCS2.b.30) $F_{t1f} \rightarrow (n_1, p_f, s_f, e_f) next(f_0)$.

Now we would like to use the following proposition, which will be proved below:

(Prop) $\forall F_{t1}, F_{t2} \in T_{Formula}, n \in \mathbb{N}, f \in T_{FormulaCore}, p \in \mathbb{N}, s \in Stream, e \in Environment: n > 0 \Rightarrow F_{t1} \rightarrow (n, p, s, e) next(f) \Rightarrow next(TCS(F_{t1}, F_{t2})) \rightarrow (n, p, s, e) next(TCS(next(f), F_{t2}))$

Using (Prop) under the assumptions $n_1 > 0$ and (TCS2.b.30), we obtain

(TCS2.b.31) $next(TCS(F_{t1f}, F_{t2f})) \rightarrow (n_1, p_f, s_f, e_f) next(TCS(next(f_0), F_{t2f}))$

which, by (TCS2.b.29) and Lemma 2 is

(TCS2.b.32) $next(TCS(F_{t1f}, F_{t2f})) \rightarrow l*(n_1, p_f, s_f, e_f) next(TCS(F_{t0}, F_{t2f}))$

From $n_1 + 1 > 0$, (TCS2.b.10), (TCS2.b.32), (TCS2.b.27), by the definition of $\rightarrow l*$ we get

(TCS2.b.33) $next(TCS(F_{t1f}, F_{t2f})) \rightarrow l*(n_1 + 1, p_f, s_f, e_f) done(b_f)$

Since $n_2 = 1$, we have $n_1 + 1 = \max(n_1 + 1, 1) = \max(n_1 + 1, n_2)$. Therefore, from (TCS2.b.33) by Lemma 2 we obtain [TCS2.b.7]

This finishes the proof of [TCS2.b].

This finishes the proof of [TCS2].

This finishes the proof of the Statement 2 of Lemma 4.

-------------
Proof of (Prop)
-------------
Parametrization:

$\Theta(n) : \iff$

$\forall F_{t1}, F_{t2} \in T_{Formula}, f \in T_{FormulaCore}, p \in \mathbb{N}, s \in Stream, e \in Environment: n > 0 \Rightarrow$

$F_{t1} \rightarrow (n, p, s, e) next(f) \Rightarrow$

$next(TCS(F_{t1}, F_{t2})) \rightarrow (n, p, s, e) next(TCS(next(f), F_{t2}))$

We need to prove $\forall n \in \mathbb{N}: \Theta(n)$. Induction:
\[\text{[Prop.a]} \ \Theta(1)\]
\[\text{[Prop.b] } \forall n \in \mathbb{N}: \ \Theta(n) \Rightarrow \Theta(n+1)\]

Proof of [Prop.a]
-----------------
We take \(Ft_1f, Ft_2f, f_0, pf, sf, ef\) arbitrary but fixed. Assume

(p1) \(Ft_1f \rightarrow^*(1, pf, sf, ef) \text{ next}(f_0)\)

and prove

[p2] \(\text{next}(\text{TCS}(Ft_1f, Ft_2f)) \rightarrow^*(1, pf, sf, ef) \text{ next}(\text{TCS}(\text{next}(f_0), Ft_2f))\).

From (p1), by the definition of \(\rightarrow^*\) there exists \(Ft' \in \text{TFormula}\) such that

(p3) \(Ft_1f \rightarrow (pf, sf, sf(pf), c) \quad Ft'\)

(p4) \(Ft' \rightarrow^*(0, pf+1, sf, ef) \text{ next}(f_0)\)

where

(p5) \(c = (ef, \{(X, sf(ef(X)))| X \in \text{dom(ef)}\}).\)

From (p4), we have \(Ft' = \text{next}(f_0)\) and, hence, from (p3) we get

(p6) \(Ft_1f \rightarrow (pf, sf, pf, sf(pf), c) \text{ next}(f_0)\).

From (p6), by the definition of \(\rightarrow^*\) for \text{TCS}, we have

(p7) \(\text{next}(\text{TCS}(Ft_1f, Ft_2f)) \rightarrow (pf, sf, sf(pf), c) \text{ next}(\text{TCS}(\text{next}(f_0), Ft_2f))\).

On the other hand, we have by the definition of \(\rightarrow^*\):

(p8) \(\text{next}(\text{TCS}(\text{next}(f_0), Ft_2f)) \rightarrow^*(0, pf+1, sf, ef) \text{ next}(\text{TCS}(\text{next}(f_0), Ft_2f))\).

From (p7), (p5), (p8), by the definition of \(\rightarrow^*\) we get [p2].

Proof of [Prop.b]
-----------------
We take \(n\) arbitrary but fixed, assume

(p9) \(\forall Ft_1, Ft_2 \in \text{TFormula}, f \in \text{TFormulaCore}, p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}: n > 0 \Rightarrow\)

\[Ft_1 \rightarrow^*(n, p, s, e) \text{ next}(f) \Rightarrow \]

\[\text{next}(\text{TCS}(Ft_1, Ft_2)) \rightarrow^*(n, p, s, e) \text{ next}(\text{TCS}(\text{next}(f), Ft_2))\]

and prove

[p10] \(\forall Ft_1, Ft_2 \in \text{TFormula}, f \in \text{TFormulaCore}, p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}: n+1 > 0 \Rightarrow\)

\[Ft_1 \rightarrow^*(n+1, p, s, e) \text{ next}(f) \Rightarrow \]

\[\text{next}(\text{TCS}(Ft_1, Ft_2)) \rightarrow^*(n+1, p, s, e) \text{ next}(\text{TCS}(\text{next}(f), Ft_2))\).

To prove (p10), we take \(Ft_1f, Ft_2f, f_0, pf, sf, ef\) arbitrary but fixed, assume
(p11) \( F_{t1f} \rightarrow^* (n+1,pf,sf,ef) \) next(f0)

and prove

[p12] \( \text{next}(\text{TCS}(F_{t1f},F_{t2f})) \rightarrow^* (n+1,pf,ef,sf) \) next(\( \text{TCS} \) \( \text{next} \) (f0),\( F_{t2f} \)).

Case \( n>0 \)
-------
From (p11), by the definition of \( \rightarrow^* \), we obtain for some \( F_t' \in \text{TFormula} \)

(p13) \( F_{t1f} \rightarrow (pf,sf,pf,sf(pf),c) \) \( F_t' \)
(p14) \( F_t' \rightarrow^* (n,pf+1,sf,ef) \) next(f0)

where

(p15) \( c=\{(ef,(X,\text{sf}(ef(X)))| X\in\text{dom}(ef))\} \).

Since \( n>0 \), from (p14) and the induction hypothesis (p9) we obtain

(p16) \( \text{next}(\text{TCS}(F_t',F_{t2f})) \rightarrow^* (n,pf+1,ef,sf) \) next(\( \text{TCS} \) \( \text{next} \) (f0),\( F_{t2f} \)).

Moreover, \( F_t' \) is a 'next' formula. Therefore, from (p13), by the definition of \( \rightarrow^* \) for \( \text{TCS} \) we have

(p17) \( \text{next}(\text{TCS}(F_{t1f},F_{t2f})) \rightarrow (pf,sf,pf,sf(pf),c) \) next(\( \text{TCS} \) \( F_t' \),\( F_{t2f} \)).

From (p16), (p15), (p17), since \( n+1>9 \), by the definition of \( \rightarrow^* \) we get [p12].

Case \( n=0 \)
-------
In this [p12] can be proved as it has been done in the base case [Prop.a]

This finishes the proof of [Prop.b] and, hence of (Prop).

Statement 3. TCP Formulas.

Lemma 4 (n-Step Reductions to Done Formulas).

Statement 3. (TCP formulas)

\[ \forall p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, F_{t1}, F_{t2} \in \text{TFormula}, n_1, n_2 \in \mathbb{N}: \]
\[ n_1 > 0 \land \text{next}(\text{TCS}(F_{t1},F_{t2})) \rightarrow^* (n_1,p,s,e) \text{ done}(false) \land \text{next}(\text{TCS}(F_{t1},F_{t2})) \rightarrow^* (n_1,p,s,e) \text{ done}(false) \]
\[ \land \]
\[ n_1 > 0 \land n_2 > 0 \land \text{next}(\text{TCS}(F_{t1},F_{t2})) \rightarrow^* (n_1,n_2,p,s,e) \text{ done}(false) \land \text{next}(\text{TCS}(F_{t1},F_{t2})) \rightarrow^* (n_1,n_2,p,s,e) \text{ done}(false) \]
\[ \land \]
\[ n_1 > 0 \land n_2 > 0 \land \text{next}(\text{TCS}(F_{t1},F_{t2})) \rightarrow^* (n_1,n_2,p,s,e) \text{ done}(false) \land \text{next}(\text{TCS}(F_{t1},F_{t2})) \rightarrow^* (n_1,n_2,p,s,e) \text{ done}(false) \]
\[ \land \]
\[ n_1 > 0 \land n_2 > 0 \land \text{next}(\text{TCS}(F_{t1},F_{t2})) \rightarrow^* (n_1,n_2,p,s,e) \text{ done}(false) \land \text{next}(\text{TCS}(F_{t1},F_{t2})) \rightarrow^* (n_1,n_2,p,s,e) \text{ done}(false) \]

Proof
We split the statement in four:

\[ \text{TCP1} \] \forall p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, F_t, F_{t'} \in \text{Formula}, n_1, n_2 \in \mathbb{N} : \]
\begin{align*}
n_1 > 0 & \land n_2 > 0 \land F_t \to \ast(n_1, p, s, e) \text{ done(false)} \land F_{t'} \to \ast(n_2, p, s, e) \text{ done(true)} \Rightarrow \\
\text{next(TCP}(F_t, F_{t'}))) & \to \ast(n_1, p, s, e) \text{ done(false)}
\end{align*}

\[ \text{TCP2} \] \forall p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, F_t, F_{t'} \in \text{Formula}, n_1, n_2 \in \mathbb{N} : \]
\begin{align*}
n_1 > 0 & \land n_2 > 0 \land F_t \to \ast(n_1, p, s, e) \text{ done(false)} \land \\
F_{t'} & \to \ast(n_2, p, s, e) \text{ done(false)} \\
\Rightarrow \\
\text{next(TCP}(F_t, F_{t'})) & \to \ast(n_1, p, s, e) \text{ done(false)}
\end{align*}

\[ \text{TCP3} \] \forall p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, F_t, F_{t'} \in \text{Formula}, n_1, n_2 \in \mathbb{N} : \]
\begin{align*}
n_1 > 0 & \land n_2 > 0 \land F_t \to \ast(n_1, p, s, e) \text{ done(true)} \land F_{t'} \to \ast(n_2, p, s, e) \text{ done(true)} \Rightarrow \\
\text{next(TCP}(F_t, F_{t'})) & \to \ast(n_1, n_2, p, s, e) \text{ done(true)}
\end{align*}

\[ \text{TCP4} \] \forall p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, F_t, F_{t'} \in \text{Formula}, n_1, n_2 \in \mathbb{N} : \]
\begin{align*}
n_1 > 0 & \land n_2 > 0 \land F_t \to \ast(n_1, p, s, e) \text{ done(true)} \land F_{t'} \to \ast(n_2, p, s, e) \text{ done(false)} \Rightarrow \\
\text{next(TCP}(F_t, F_{t'})) & \to \ast(n_2, p, s, e) \text{ done(false)}
\end{align*}

Proof of [TCP1]
---------

We take \(s_f, e_f\) arbitrary but fixed and define

\[ \Phi(n) : \iff \]
\[ \forall p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, F_t, F_{t'} \in \text{Formula}, n_1, n_2 \in \mathbb{N} : \]
\begin{align*}
n_1 > 0 & \land n_2 > 0 \land F_t \to \ast(n_1, p, s, e) \text{ done(false)} \land F_{t'} \to \ast(n_2, p, s, e) \text{ done(true)} \Rightarrow \\
\text{next(TCP}(F_t, F_{t'})) & \to \ast(n_1, p, s, e) \text{ done(false)}
\end{align*}

We prove \(\forall n_1 \in \mathbb{N} : \Phi(n_1)\) by induction over \(n_1\). For \(n_1 = 0\) the formula is trivially true.

We start the induction from 1. Prove:

[TCP1.a] \(\Phi(1)\) and
[TCP1.b] \(\forall n_1 \in \mathbb{N} : \Phi(n_1) \Rightarrow \Phi(n_1+1)\)

Proof of [TCP1.a]
----------

We take \(p_f, F_{t1f}, F_{t2f}, n_2\) arbitrary but fixed. \(1 > 0\) is satisfied. Assume

\begin{align*}
(TCP1.1) n_2 & > 0 \\
(TCP1.2) F_{t1f} & \to \ast(1, p_f, s_f, e_f) \text{ done(false)}. \\
(TCP1.3) F_{t2f} & \to \ast(n_2, p, s, e) \text{ done(true)}.
\end{align*}

We want to prove

\[ T[CP1.4] \text{next(TCP}(F_{t1f}, F_{t2f})) \to \ast(1, p_f, s_f, e_f) \text{ done(false)}. \]
From (TCP1.2), by the definition of \( \rightarrow^* \) without history, there exists \( F_t \in T_{\text{Formula}} \) such that

\[(TCP1.5) \; F_t^1 \rightarrow (p, sf|pf, sf(pf), c) \; F_t \quad \text{and} \quad (TCP1.6) \; F_t \rightarrow^* (0, pf+1, sf, ef) \; \text{done(false)}\]

where

\[(TCP1.7) \; c = (ef, \{(X, sf(ef(X))) | X \in \text{dom}(ef)\}).\]

From (TCP1.6), by the definition of \( \rightarrow^* \) without history, we get

\[(TCP1.8') \; F_t = \text{done(false)}\]

which from (TCP1.5) gives

\[(TCP1.9') \; F_t^1 \rightarrow (p, sf|pf, sf(pf), c) \; \text{done(false)} \; \text{and} \quad (TCP1.10') \; \text{next(TCP}(F_t^1, F_t^2)) \rightarrow (p, sf|pf, sf(pf), c) \; \text{done(false)}\]

From (TCP1.10', TCP1.6, TCP1.8', TCP1.7), by the definition of \( \rightarrow^* \) without history, we get [TCP1.4].

Proof of [TCP1.b]
--------------
We take \( n_1 \) arbitrary but fixed, assume

\[(TCP1.8) \; \forall p \in \mathbb{N}, F_{t1}, F_{t2} \in T_{\text{Formula}}, n_2 \in \mathbb{N} : \]
\[n_1 > 0 \land n_2 > 0 \land \]
\[F_{t1} \rightarrow^*(n_1, p, s, e) \; \text{done(false)} \land F_{t2} \rightarrow^*(n_2, p, s, e) \; \text{done(true)} \Rightarrow \]
\[\text{next(TCP}(F_{t1}, F_{t2})) \rightarrow^*(n_1, p, s, e) \; \text{done(false)}\]

and prove

\[\text{[TCP1.9]} \; \forall p \in \mathbb{N}, F_{t1}, F_{t2} \in T_{\text{Formula}}, n_2 \in \mathbb{N} : \]
\[n_1 + 1 > 0 \land n_2 > 0 \land \]
\[F_{t1} \rightarrow^*(n_1+1, p, s, e) \; \text{done(false)} \land F_{t2} \rightarrow^*(n_2, p, s, e) \; \text{done(true)} \Rightarrow \]
\[\text{next(TCP}(F_{t1}, F_{t2})) \rightarrow^*(n_1+1, p, s, e) \; \text{done(false)}\]

To prove [TCP1.9], we take \( p, F_{t1}, F_{t2}, n_2 \) arbitrary but fixed, assume

\[(TCP1.10) \; n_1 + 1 > 0 \]
\[(TCP1.11) \; n_2 > 0 \]
\[(TCP1.12) \; F_{t1} \rightarrow^*(n_1+1, pf, sf, ef) \; \text{done(false)} \]
\[(TCP1.13) \; F_{t2} \rightarrow^*(n_2, pf, sf, ef) \; \text{done(true)}\]

and prove

\[\text{[TCP1.14]} \; \text{next(TCP}(F_{t1}, F_{t2})) \rightarrow^*(n_1+1, pf, sf, ef) \; \text{done(false)}\]

From (TCP1.12), by (TCP1.10) and the definition of \( \rightarrow^* \) without history,
there exists \(F_t' \in \text{TFormula}\) such that

\[
\begin{align*}
\text{(TCP1.15)} & \quad F_t f \rightarrow (pf, sf \downarrow pf, sf(pf), c) F_t' \\
\text{(TCP1.16)} & \quad F_t' \rightarrow (n_1, pf + 1, sf, ef) \text{ done(false)}
\end{align*}
\]

where

\[
\begin{align*}
\text{(TCP1.17)} & \quad c = (ef, \{(X, sf(ef(X))) \mid X \in \text{dom}(ef)\}).
\end{align*}
\]

From (TCP1.13), by (TCP1.11) and the definition of \(\rightarrow^*\) without history, there exists \(F_t'' \in \text{TFormula}\) such that

\[
\begin{align*}
\text{(TCP1.18)} & \quad F_t f \rightarrow (pf, sf \downarrow pf, sf(pf), c) F_t'' \\
\text{(TCP1.19)} & \quad F_t'' \rightarrow (n_2 - 1, pf + 1, sf, ef) \text{ done(true)}
\end{align*}
\]

where \(c\) is defined as in (TCP1.17).

Case \(n_1 > 0, n_2 - 1 > 0\)

-------------

In this case \(F_t' = \text{next}(f')\), \(F_t'' = \text{next}(f'')\) for some \(f', f'' \in \text{TFormulaCore}\). Therefore, from (TCP1.15, TCP1.18), by the definition of \(\rightarrow\) for TCP we have

\[
\begin{align*}
\text{(TCP1.20)} & \quad \text{next(TCP}(F_t f, F_t f)) \rightarrow (pf, sf \downarrow pf, sf(pf), c) \text{ next(TCP}(F_t', F_t'')).
\end{align*}
\]

From \(n_1 > 0, n_2 - 1 > 0\), (TCP1.16, TCP1.19), by the induction hypothesis (TCP1.8) we have

\[
\begin{align*}
\text{(TCP1.21)} & \quad \text{next(TCP}(F_t', F_t'')) \rightarrow (n_1, pf + 1, sf, ef) \text{ done(false)}.
\end{align*}
\]

From \(n_1 + 1 > 0\), (TCP1.17), (TCP1.20), (TCP1.21), by the definition of \(\rightarrow^*\) we have

\[
\begin{align*}
\text{(TCP1.22)} & \quad \text{next(TCP}(F_t f, F_t f)) \rightarrow (n_1 + 1, pf, sf, ef) \text{ done(false)}
\end{align*}
\]

which is [TCP1.14]

Case \(n_1 > 0, n_2 - 1 = 0\)

-------------

In this case \(F_t' = \text{next}(f')\) for some \(f' \in \text{TFormulaCore}\) and, from (TCP1.18)

\[
\begin{align*}
\text{(TCP1.23)} & \quad F_t f \rightarrow (pf, sf \downarrow pf, sf(pf), c) \text{ done(true)}.
\end{align*}
\]

Therefore, from (TCP1.15, TCP1.23), by the definition of \(\rightarrow\) for TCP we have

\[
\begin{align*}
\text{(TCP1.24)} & \quad \text{next(TCP}(F_t f, F_t f)) \rightarrow (pf, sf \downarrow pf, sf(pf), c) F_t'
\end{align*}
\]

From \(n_1 + 1 > 0\), (TCP1.17), (TCP1.24), (TCP1.16), by the definition of \(\rightarrow^*\) we get [TCP1.14].

Case \(n_1 = 0\)

-------------

In this case \(F_t'' = \text{next}(f'')\) for some \(f'' \in \text{TFormulaCore}\) and, from (TCP1.15)
(TCP1.25) \( \text{Ft1f} \rightarrow (\text{pf}, \text{sf}\downarrow \text{pf}, \text{sf}(\text{pf}), \text{c}) \) done(false).

From (TCP1.25) by the definition of \( \rightarrow \) for TCP we have

(TCP1.26) \( \text{next}(\text{TCP}(\text{Ft1f}, \text{Ft2f})) \rightarrow (\text{pf}, \text{sf}\downarrow \text{pf}, \text{sf}(\text{pf}), \text{c}) \) done(false).

From \( n1+1>0 \), (TCP1.17), (TCP1.26), (TCP1.16), by the definition of \( \rightarrow * \) we get [TCP1.14].

This finishes the proof of (b) and, therefore, the proof of [TCP1].

==============================================================

Proof of [TCP2]
---------------

Recall

[TCP2] \( \forall p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, \text{Ft1}, \text{Ft2} \in \text{Formula}, n1,n2 \in \mathbb{N} : 
\begin{align*}
n1>0 & \land n2>0 \land \text{Ft1} \rightarrow *(n1,p,s,e) \text{ done(false)} \land 
\text{Ft2} \rightarrow *(n2,p,s,e) \text{ done(false)} \\
\Rightarrow 
\text{next}(\text{TCP(Ft1,Ft2)}) & \rightarrow *(\min(n1,n2),p,s,e) \text{ done(false)}
\end{align*} \)

Proof
-----

We take \( sf, ef \) arbitrary but fixed and define

\( \Phi(n) : \leftrightarrow \\
\forall p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, \text{Ft1}, \text{Ft2} \in \text{Formula}, n1,n2 \in \mathbb{N} : 
\begin{align*}
n1>0 & \land n2>0 \land 
\text{Ft1} \rightarrow *(n1,p,s,e) \text{ done(false)} \land \text{Ft2} \rightarrow *(n2,p,s,e) \text{ done(false)} \\
\Rightarrow 
\text{next}(\text{TCP(Ft1,Ft2)}) & \rightarrow *(\min(n1,n2),p,s,e) \text{ done(false)}
\end{align*} \)

We prove \( \forall n1 \in \mathbb{N} : \Phi(n1) \) by induction over \( n1 \). For \( n1=0 \) the formula is trivially true.

We start the induction from 1. Prove:

[TCP2.a] \( \Phi(1) \) and
[TCP2.b] \( \forall n1 \in \mathbb{N} : \Phi(n1) \Rightarrow \Phi(n1+1) \)

Proof of [TCP2.a]
-----------------

We take \( pf, \text{Ft1f}, \text{Ft2f}, n2 \) arbitrary but fixed. \( 1>0 \) is satisfied. Assume

(TCP2.1) \( n2>0 \)
(TCP2.2) \( \text{Ft1f} \rightarrow *(1,pf,\text{sf},\text{ef}) \text{ done(false)} \).
(TCP2.3) \( \text{Ft2f} \rightarrow *(n2,p,s,e) \text{ done(false)} \).

We want to prove

[TCP2.4] \( \text{next}(\text{TCP(\text{Ft1f},\text{Ft2f})}) \rightarrow *(\min(1,n2),pf,\text{sf},\text{ef}) \text{ done(false)} \).
From (TCP2.2), by the definition of \( \rightarrow^* \) without history, there exists 
\( F_t \in T_{\text{Formula}} \) such that 

\[
\text{(TCP2.5)} \quad F_{t1} \rightarrow (p,s_f,p_f,s_f(p),c) F_t \\
\text{(TCP2.6)} \quad F_t \rightarrow^* (0,p_f+1,s_f,ef) \text{ done(false)}
\]

where 

\[
\text{(TCP2.7)} \quad c=(ef, \{(X,s_f(ef(X))) \mid X \in \text{dom(ef)}\}).
\]

From (TCP2.6), by the definition of \( \rightarrow^* \) without history, we get 

\[
\text{(TCP2.8)} \quad F_t=\text{done(false)}.
\]

which from (TCP2.5) gives 

\[
\text{(TCP2.9)} \quad F_{t1} \rightarrow (p,s_f,p_f,s_f(p),c) \text{ done(false)}.
\]

From (TCP2.9) and (TCP2.3), by the definition of \( \rightarrow \) for TCP, we get 

\[
\text{(TCP2.10)} \quad \text{next}(TCP(F_{t1},F_{t2})) \rightarrow (p,s_f,p_f,s_f(p),c) \text{ done(false)}.
\]

From (TCP2.10, TCP2.6, TCP2.8, TCP2.7), by the definition of \( \rightarrow^* \) without history, we get \( \text{next}(TCP(F_{t1},F_{t2})) \rightarrow^*(1,p_f,s_f,ef) \text{ done(false)} \), but since by (TCP2.1) we have \( 1=\min(1,n_2) \), we actually proved [TCP2.4].

Proof of [TCP2.b]

-----------------

We take \( n_1 \) arbitrary but fixed, assume 

\[
\text{(TCP2.8)} \quad \forall p \in \mathbb{N}, \; F_{t1},F_{t2} \in T_{\text{Formula}}, \; n_2 \in \mathbb{N} : \\
n_1>0 \land n_2>0 \land \\
F_{t1} \rightarrow^*(n_1,p,s,e) \text{ done(false)} \land F_{t2} \rightarrow^*(n_2,p,s,e) \text{ done(false)} \Rightarrow \\
\text{next}(TCP(F_{t1},F_{t2})) \rightarrow^*(\min(n_1,n_2),p,s,e) \text{ done(false)}
\]

and prove 

\[
\text{[TCP2.9]} \quad \forall p \in \mathbb{N}, \; F_{t1},F_{t2} \in T_{\text{Formula}}, \; n_2 \in \mathbb{N} : \\
n_1+1>0 \land n_2>0 \land \\
F_{t1} \rightarrow^*(n_1+1,p,s,e) \text{ done(false)} \land F_{t2} \rightarrow^*(n_2,p,s,e) \text{ done(false)} \Rightarrow \\
\text{next}(TCP(F_{t1},F_{t2})) \rightarrow^*(\min(n_1+1,n_2),p,s,e) \text{ done(false)}.
\]

To prove [TCP2.9], we take \( p,F_{t1},F_{t2},n_2 \) arbitrary but fixed, assume 

\[
\text{(TCP2.10)} \quad n_1>0 \\
\text{(TCP2.11)} \quad n_2>0 \\
\text{(TCP2.12)} \quad F_{t1} \rightarrow^*(n_1+1,p_f,s_f,ef) \text{ done(false)} \\
\text{(TCP2.13)} \quad F_{t2} \rightarrow^*(n_2,p_f,s_f,ef) \text{ done(false)}
\]

and prove 

\[
\text{[TCP2.14]} \quad \text{next}(TCP(F_{t1},F_{t2})) \rightarrow^*(\min(n_1+1,n_2),p_f,s_f,ef) \text{ done(false)}.
\]

From (TCP2.12), by (TCP2.10) and the definition of \( \rightarrow^* \) without history,
there exists $F_t' \in \text{TFormula}$ such that

(TCP2.15) $F_t \rightarrow (pf, sf \downarrow pf, sf(pf), c) F_t'$
(TCP2.16) $F_t' \rightarrow* (n1, pf+1, sf, ef) \text{done}(false)$

where

(TCP2.17) $c = (ef, \{(X, sf(ef(X)))| X \in \text{dom}(ef)\})$.

From (TCP2.13), by (TCP2.11) and the definition of $\rightarrow*$ without history, there exists $F_t'' \in \text{TFormula}$ such that

(TCP2.18) $F_t \rightarrow (pf, sf \downarrow pf, sf(pf), c) F_t''$
(TCP2.19) $F_t'' \rightarrow* (n2-1, pf+1, sf, ef) \text{done}(false)$

where $c$ is defined as in (TCP2.17).

Case $n_1 > 0, n_2 - 1 > 0$

In this case $F_t' = \text{next}(f')$, $F_t'' = \text{next}(f'')$ for some $f', f'' \in \text{TFormulaCore}$. Therefore, from (TCP2.15, TCP2.18), by the definition of $\rightarrow$ for TCP we have

(TCP2.20) $\text{next}(TCP(F_t f_1, F_t f_2)) \rightarrow (pf, sf \downarrow pf, sf(pf), c) \text{next}(TCP(F_t', F_t''))$.

From $n_1 > 0, n_2 - 1 > 0$, (TCP2.16, TCP2.19), by the induction hypothesis (TCP2.8) we have

(TCP2.21) $\text{next}(TCP(F_t', F_t'')) \rightarrow* (\min(n_1, n_2 - 1), pf+1, sf, ef) \text{done}(false)$.

From $n_1 + 1 > 0$, (TCP2.17), (TCP2.20), (TCP2.21), by the definition of $\rightarrow*$ we have

(TCP2.22) $\text{next}(TCP(F_t f_1, F_t f_2)) \rightarrow* (\min(n_1, n_2 - 1) + 1, pf, sf, ef) \text{done}(false)$

which is [TCP2.14]

Case $n_1 > 0, n_2 - 1 = 0$

In this case $F_t' = \text{next}(f')$ for some $f' \in \text{TFormulaCore}$ and, from (TCP2.18) we have

(TCP2.23) $F_t f_2 \rightarrow (pf, sf \downarrow pf, sf(pf), c) \text{done}(false)$.

Therefore, from (TCP2.15, TCP2.23), by the definition of $\rightarrow$ for TCP we have

(TCP2.24) $\text{next}(TCP(F_t f_1, F_t f_2)) \rightarrow (pf, sf \downarrow pf, sf(pf), c) \text{done}(false)$

From $1 > 0$, (TCP2.17), (TCP2.24), (TCP2.19), by the definition of $\rightarrow*$ we get

(TCP2.25) $\text{next}(TCP(F_t f_1, F_t f_2)) \rightarrow* (1, pf, sf, ef) \text{done}(false)$

But by $n_1 > 0$ and $n_2 = 1$ we have $1 = \min(n_1 + 1, n_2)$. Hence, (TCP2.25) proves [TCP2.14].

Case $n_1 = 0$

In this case $F_t'' = \text{next}(f'')$ for some $f'' \in \text{TFormulaCore}$ and, from (TCP2.15)
we have

(TCP2.26) \( F_{tf} \rightarrow (pf, sf \downarrow pf, sf(pf), c) \) done(false).

From (TCP2.26) by the definition of \( \rightarrow \) for TCP we have

(TCP2.27) next(TCP(F_{tf1}, F_{tf2})) \( \rightarrow (pf, sf \downarrow pf, sf(pf), c) \) done(false).

From 1>0, (TCP2.17), (TCP2.27), (TCP2.16), by the definition of \( \rightarrow^* \) we get

(TCP2.28) next(TCP(F_{tf1}, F_{tf2})) \( \rightarrow^*(1, pf, sf, ef) \) done(true).

But by \( n1=0 \) and \( n2>0 \) we have \( 1=\min(n1+1,n2) \). Hence, (TCP2.28) proves [TCP2.14].

This finishes the proof of (b) and, therefore, the proof of [TCP2].

=====================================================  

Proof of [TCP3]  
-----------------------

[TCP3] \( \forall \ p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, F_{t1}, F_{t2} \in \text{TFormula}, n1, n2 \in \mathbb{N}, b \in \text{Bool} \) :  
\( n1>0 \ \land \ n2>0 \ \land \  
F_{t1} \rightarrow^*(n1, p, s, e) \) done(true) \land \( F_{t2} \rightarrow^*(n2, p, s, e) \) done(true) \Rightarrow  
next(TCP(F_{t1}, F_{t2})) \rightarrow^*(\max(n1, n2), p, s, e) \) done(true).

Proof
-----

We take \( sf, ef \) arbitrary but fixed and define

\( \Phi(n1) : \leftarrow  
\forall p \in dsN, F_{t1}, F_{t2} \in \text{TFormula}, n2 \in \mathbb{N} :  
n1>0 \ \land \ n2>0 \ \land 
F_{t1} \rightarrow^*(n1, p, sf, ef) \) done(true) \land \( F_{t2} \rightarrow^*(n2, p, sf, ef) \) done(true) \Rightarrow  
next(TCP(F_{t1}, F_{t2})) \rightarrow^*(\max(n1, n2), p, sf, ef) \) done(true).

We need to prove \( \forall n1 \in \mathbb{N} : \Phi(n1) \). We use induction. Prove:

[TCP3.a] \( \forall n2 \in \mathbb{N} : \Phi(1) \)  
[TCP3.b] \( \forall n1 \in \mathbb{N} : \Phi(n1) \Rightarrow \Phi(n1+1) \).

Proof of [TCP3.a]
------------

We need to prove

\( \forall n2, p \in dsN, F_{t1}, F_{t2} \in \text{TFormula} :  
n1>0 \ \land \ n2>0 \ \land 
F_{t1} \rightarrow^*(1, p, sf, ef) \) done(true) \land \( F_{t2} \rightarrow^*(n2, p, sf, ef) \) done(true) \Rightarrow  
next(TCP(F_{t1}, F_{t2})) \rightarrow^*(\max(1, n2), p, sf, ef) \) done(true).

We take \( n2, pf, F_{t1f}, F_{t2f} \) arbitrary but fixed. Assume
\[(TCP3.a.1)\] n2>0
\[(TCP3.a.2)\] Ft1f \(\rightarrow\) *(1,pf,sf,ef) done(true)
\[(TCP3.a.3)\] Ft2f \(\rightarrow\) *(n2,pf,sf,ef) done(true)

and prove

\[(TCP3.a.4)\] next(TCP(Ft1f,Ft2f)) \(\rightarrow\) *(max(1,n2),pf,ef) done(true).

From (TCP3.a.2), by the definition of \(\rightarrow\), we have for some \(Ft'\)

\[(TCP3.a.5)\] Ft1f \(\rightarrow\) (pf,sf,\{pf\}) Ft'
\[(TCP3.a.6)\] Ft' \(\rightarrow\) *(0,pf+1,sf,ef) done(true)

where

\[(TCP3.a.7)\] \(c=\{(X,sf(ef(X)))\mid X \in \text{dom}(ef)\}\).

From (TCP3.a.6), by the definition \(pf \rightarrow\), we know

\[(TCP3.a.8)\] Ft' = done(true).

From (TCP3.a.5) and (TCP3.a.8) we have

\[(TCP3.a.9)\] Ft1f \(\rightarrow\) (pf,sf,\{pf\}) done(true).

From (TCP3.a.3), by the definition of \(\rightarrow\), we have for some \(Ft''\)

\[(TCP3.a.10)\] Ft2f \(\rightarrow\) (pf,sf,\{pf\}) Ft''
\[(TCP3.a.11)\] Ft'' \(\rightarrow\) *(n2-1,pf+1,sf,ef) done(true),

where \(c\) is defined as in (TCP3.a.7).

From (TCP3.a.9) and (TCP3.a.10), by the definition of \(\rightarrow\) for TCP, we have

\[(TCP3.a.13)\] next(TCP(Ft1f,Ft2f)) \(\rightarrow\) (pf,sf,\{pf\}) Ft''.

From (TCP3.a.13), (TCP3.a.7), and (TCP3.a.11), by the definition of \(\rightarrow\), we have

\[(TCP3.a.14)\] next(TCP(Ft1f,Ft2f)) \(\rightarrow\) *(n2,pf,ef) done(true).

From (TCP3.a.1), we have n2=\(\max(1,n2)\). Therefore, (TCP3.a.14) proves [TCP3.a.4]

This finishes the proof of [TCP3.a].

Proof of [TCP3.b]
-----------------

We take \(n1\) arbitrary but fixed. Assume \(\Phi(n1)\), i.e.,

\[(TCP3.b.1)\] \(\forall n2, p \in \text{dsN}, Ft1,Ft2 \in TFormula :\)
\(n1>0 \land n2>0 \land Ft1 \rightarrow\) *(n1,p,ef) done(true) \land
\(Ft2 \rightarrow\) *(n2,p,ef) done(true)
⇒
next(TCP(Ft1,Ft2)) →*(max(n1,n2),p,sf,ef) done(true).

and prove

[TCP3.b.2] ∀n2,p∈dsN, Ft1,Ft2∈TFormula :
   n1+1>0 ∧ n2>0 ∧ Ft1 →*(n1+1,p,sf,ef) done(true) ∧ Ft2 →*(n2,p,sf,ef) done(true)
   ⇒
next(TCP(Ft1,Ft2)) →*(max(n1+1,n2),p,sf,ef) done(true).

To prove [TCP3.b.2], we take n2, pf, Ft1f, Ft2f arbitrary but fixed. Assume

(TCP3.b.3) n1+1>0
(TCP3.b.4) n2>0
(TCP3.b.5) Ft1f →*(n1+1,pf,sf,ef) done(true)
(TCP3.b.6) Ft2f →*(n2,pf,sf,ef) done(true)

and prove

[TCP3.b.7] next(TCP(Ft1f,Ft2f)) →*(max(n1+1,n2),pf,sf,ef) done(true).

From (TCP3.b.5), by the definition of →*, we have for some Ft'

(TCP3.b.8) Ft1f →(pf,sf↓pf,sf(pf),c) Ft'
(TCP3.b.9) Ft' →*(n1,pf+1,sf,ef) done(true)

where

(TCP3.b.10) c=(ef, {(X,sf(ef(X)))| X∈dom(ef)}).

From (TCP3.b.6), by the definition of →*, we have for some Ft''

(TCP3.b.11) Ft2f →(pf,sf↓pf,sf(pf),c) Ft''
(TCP3.b.12) Ft'' →*(n2-1,pf+1,ef) done(true)

where c is defined as in (TCP3.b.10).

Case 1. n1=0
----------
In this case we have Ft'='done(true) and from (TCP3.b.8) we get

(TCP3.b.13) Ft1f →(pf, sf↓pf,sf(pf),c) done(true).

From (TCP3.b.13) and (TCP3.b.11), by the definition of → for TCP, we have

(TCP3.b.14) next(TCP(Ft1f,Ft2f)) →(pf,sf↓pf,sf(pf),c) Ft''.

From (TCP3.b.4), (TCP3.b.10), (TCP3.b.14), (TCP3.b.12) by the definition of →*
we get

(TCP3.b.15) next(TCP(Ft1f,Ft2f)) →*(n2,pf,sf,ef) done(true).
By (TCP3.b.4) and n1=0, we have n2=max(1,n2)=max(n1+1,n2). Hence, (TCP3.b.15) proves [TCP3.b.7].

Case n1>0, n2-1>0
-----------------
In this case Ft'=next(f'), Ft''=next(f'') for some f',f''∈TFormulaCore. Therefore, from (TCP3.b.8,TCP3.b.11), by the definition of → for TCP we have

(TCP3.b.16) next(TCP(Ft1,Ft2)) →(pf,sf↓pf, sf(pf),c) next(TCP(Ft',Ft'')).

From n1>0, n2-1>0, (b9,b12), by the induction hypothesis (TCP3.b.1) we have

(TCP3.b.17) next(TCP(Ft',Ft'')) →*(max(n1,n2-1),pf+1,sf,ef) done(true).

From n1+1>0, (TCP3.b.10), (TCP3.b.16), (TCP3.b.17), by the definition of →* we have

(TCP3.b.18) next(TCP(Ft1,Ft2)) →*(max(n1,n2-1)+1,pf,sf,ef) done(true)
which is [TCP3.b.7]

Case n1>0, n2-1=0
-----------------
In this case Ft'=next(f') for some f'∈TFormulaCore. From (TCP3.b.11) we have

(TCP3.b.19) Ft2f →(pf,sf↓pf, sf(pf),c) done(true).

From (TCP3.b.8,TCP3.b.19), by the definition of → for TCP we have

(TCP3.b.20) next(TCP(Ft1,Ft2)) →(pf,sf↓pf, sf(pf),c) Ft'

From n1+1>0, (TCP3.b.10), (TCP3.b.20), (TCP3.b.9), by the definition of →* we get

(TCP3.b.21) next(TCP(Ft1,Ft2)) →*(n1+1,pf,sf,ef) done(true)

But by n1>0 and n2=1 we have n1+1=max(n1+1,n2). Hence, from (TCP3.b.21) we get [TCP3.b.7].

This finishes the proof of [TCP3.b].

This finishes the proof of [TCP3].

=========================================================================
Proof of [TCP4]
-----------------------

[TCP4] ∀p∈N, s∈Stream, e∈Environment, Ft1,Ft2∈TFormula, n1,n2∈N :

n1>0 ∧ n2>0 ∧
Ft1 →*(n1,p,s,e) done(true) ∧ Ft2 →*(n2,p,s,e) done(false) ⇒
\[ \text{next}(TCP(Ft1,Ft2)) \rightarrow^*(n2,p,s,e) \text{ done(false)}. \]

Proof
-----

We take \( sf, ef, bf \) arbitrary but fixed and define

\[ \Phi(n1) : \iff \forall p \in dsN, Ft1, Ft2 \in TFormula, n2 \in \mathbb{N} : n1 > 0 \land n2 > 0 \land Ft1 \rightarrow^*(n1,p,sf,ef) \text{ done(true)} \land Ft2 \rightarrow^*(n2,p,ef) \text{ done(false)} \Rightarrow \text{next}(TCP(Ft1,Ft2)) \rightarrow^*(n2,p,ef) \text{ done(false)}. \]

We need to prove \( \forall n1 \in \mathbb{N} : \Phi(n1) \). We use induction. Prove:

\[ \text{TCP4.a] } \forall n2 \in \mathbb{N} : \Phi(1) \]
\[ \text{TCP4.b] } \forall n1 \in \mathbb{N} : \Phi(n1) \Rightarrow \Phi(n1+1). \]

Proof of [TCP4.a]
-----------------

We need to prove

\[ \forall n2, p \in dsN, Ft1, Ft2 \in TFormula : n1 > 0 \land n2 > 0 \land Ft1 \rightarrow^*(1,p,sf,ef) \text{ done(true)} \land Ft2 \rightarrow^*(n2,p,ef) \text{ done(false)} \Rightarrow \text{next}(TCP(Ft1,Ft2)) \rightarrow^*(n2,p,ef) \text{ done(false)}. \]

We take \( n2, p, Fti, Ftjf \) arbitrary but fixed. Assume

\[ \text{TCP4.a.1] } n2 > 0 \]
\[ \text{TCP4.a.2] } Ft1f \rightarrow^*(1,pf,sf,ef) \text{ done(true)} \]
\[ \text{TCP4.a.3] } Ft2f \rightarrow^*(n2,pf,ef) \text{ done(false)} \]

and prove

\[ \text{TCP4.a.4] } \text{next}(TCP(Ft1f,Ft2f)) \rightarrow^*(n2,pf,ef) \text{ done(false)}. \]

From (TCP4.a.2), by the definition of \( \rightarrow^* \), we have for some \( Ft' \)

\[ \text{TCP4.a.5] } Ft1f \rightarrow (pf, sf, pf, ef) \text{ done(true)} \]
\[ \text{TCP4.a.6] } Ft' \rightarrow^*(0,pf+1,ef) \text{ done(true)} \]

where

\[ \text{TCP4.a.7] } c=(ef, \{(X, sf(ef(X)))| X \in \text{dom}(ef)}) \].

From (TCP4.a.6), by the definition \( pf \rightarrow^* \), we know

\[ \text{TCP4.a.8] } Ft'=\text{done(true)}. \]

From (TCP4.a.5) and (TCP4.a.8) we have

\[ \text{TCP4.a.9] } Ft1f \rightarrow (pf, sf, pf, ef) \text{ done(true)}. \]
From (TCP4.a.3), by the definition of $\rightarrow^*$, we have for some $Ft''$

(TCP4.a.10) $Ft2f \rightarrow (pf, sf, pf, sf(pf), c) Ft''$
(TCP4.a.11) $Ft'' \rightarrow^* (n2-1, pf+1, sf, ef) \text{ done(false)},$

where $c$ is defined as in (TCP4.a.7).

From (TCP4.a.9) and (TCP4.a.10), by the definition of $\rightarrow$ for TCP, we have

(TCP4.a.13) $\text{next}(TCP(Ft1f, Ft2f)) \rightarrow (pf, sf, pf, sf(pf), c) Ft''$.

From (TCP4.a.13), (TCP4.a.7), and (TCP4.a.11), by the definition of $\rightarrow^*$, we have

(TCP4.a.14) $\text{next}(TCP(Ft1f, Ft2f)) \rightarrow^* (n2, pf, sf, ef) \text{ done(false)}$.

(TCP4.a.14) is [TCP4.a.4].

This finishes the proof of [TCP4.a].

Proof of [TCP4.b]
-----------------

We take $n1$ arbitrary but fixed. Assume $\Phi(n1)$, i.e.,

(TCP4.b.1) $\forall n2, p \in dsN, Ft1, Ft2 \in TFormula:
\begin{align*}
& n1 > 0 \land n2 > 0 \land Ft1 \rightarrow^* (n1, p, sf, ef) \text{ done(true)} \land
& Ft2 \rightarrow^* (n2, p, sf, ef) \text{ done(false)} \\
& \Rightarrow \\
& \text{next}(TCP(Ft1, Ft2)) \rightarrow^* (n2, p, sf, ef) \text{ done(false)}.
\end{align*}$

and prove

[TCP4.b.2] $\forall n2, p \in dsN, Ft1, Ft2 \in TFormula:
\begin{align*}
& n1+1 > 0 \land n2 > 0 \land Ft1 \rightarrow^* (n1+1, p, sf, ef) \text{ done(true)} \land
& Ft2 \rightarrow^* (n2, p, sf, ef) \text{ done(bf)} \\
& \Rightarrow \\
& \text{next}(TCP(Ft1, Ft2)) \rightarrow^* (false, p, sf, ef) \text{ done(false)}.
\end{align*}$

To prove [TCP4.b.2], we take $n2$, $pf$, $Ft1f$, $Ft2f$ arbitrary but fixed. Assume

(TCP4.b.3) $n1+1 > 0$
(TCP4.b.4) $n2 > 0$
(TCP4.b.5) $Ft1f \rightarrow^* (n1+1, pf, sf, ef) \text{ done(true)}$
(TCP4.b.6) $Ft2f \rightarrow^* (n2, pf, sf, ef) \text{ done(false)}$

and prove

[TCP4.b.7] $\text{next}(TCP(Ft1f, Ft2f)) \rightarrow^* (n2, pf, sf, ef) \text{ done(false)}$.

From (TCP4.b.5), by the definition of $\rightarrow^*$, we have for some $Ft'$

(TCP4.b.8) $Ft1f \rightarrow (pf, sf, pf, sf(pf), c) Ft'$
\[(TCP4.b.9) \quad F' {}\rightarrow^* (n1,pf+1,sf,ef) \text{ done(true)}\]

where

\[(TCP4.b.10) \quad c = (ef, \{ (X,sf(ef(X))) | X \in \text{dom}(ef) \})\].

From (TCP4.b.6), by the definition of \(\rightarrow^*\), we have for some \(F''\)

\[(TCP4.b.11) \quad F'' {}\rightarrow (pf,sf \downarrow pf,sf(pf),c) \quad F''\]
\[(TCP4.b.12) \quad F'' {}\rightarrow^* (n2-1,pf+1,sf,ef) \text{ done(false)}\]

where \(c\) is defined as in (TCP4.b.10).

**Case 1. \(n1=0\)**

In this case we have \(F'=\text{done(true)}\) and from (TCP4.b.8) we get

\[(TCP4.b.13) \quad F_{1f} {}\rightarrow (pf, sf \downarrow pf,sf(pf),c) \quad \text{done(true)}\].

From (TCP4.b.13) and (TCP4.b.11), by the definition of \(\rightarrow\) for TCP, we have

\[(TCP4.b.14) \quad \text{next(TCP} (F_{1f,2f}) \rightarrow (pf, sf \downarrow pf,sf(pf),c) \quad \text{F''}\].

From (TCP4.b.4), (TCP4.b.10), (TCP4.b.14), (TCP4.b.12) by the definition of \(\rightarrow^*\), we get

\[(TCP4.b.15) \quad \text{next(TCP} (F_{1f,2f}) \rightarrow^* (n2,pf,sf,ef) \quad \text{done(false)}\].

Hence, (TCP4.b.15) proves [TCP4.b.7].

**Case \(n1>0, n2-1>0\)**

In this case \(F'=\text{next(f')}\), \(F''=\text{next(f'')}\) for some \(f',f''\in TFormulaCore\). Therefore, from (TCP4.b.8,TCP4.b.11), by the definition of \(\rightarrow\) for TCP we have

\[(TCP4.b.16) \quad \text{next(TCP} (F_{1f,2f}) \rightarrow (pf, sf \downarrow pf, sf(pf),c) \quad \text{next(TCP} (F',F''))\].

From \(n1>0, n2-1>0, (b9,b12)\), by the induction hypothesis (TCP4.b.1) we have

\[(TCP4.b.17) \quad \text{next(TCP} (F',F'')) \rightarrow^* (n2-1,pf+1,sf,ef) \quad \text{done(false)}\].

From (TCP4.b.4), (TCP4.b.10), (TCP4.b.16), (TCP4.b.17), by the definition of \(\rightarrow^*\) we have

\[(TCP4.b.18) \quad \text{next(TCP} (F_{1f,2f}) \rightarrow^* (n2,pf,sf,ef) \quad \text{done(false)}\]

which is [TCP4.b.7]

**Case \(n1>0, n2-1=0\)**

In this case \(F'=\text{next(f')}\) for some \(f'\in TFormulaCore\). From (TCP4.b.11) we have
(TCP4.b.19) \( \text{Ft2f} \rightarrow (pf, sf|pf, sf(pf), c) \text{ done(false)}. \)

From (TCP4.b.8, TCP4.b.19), by the definition of \( \rightarrow \) for TCP we have

(TCP4.b.23) \( \text{next(TCP(Ftf1,Ftf2))} \rightarrow (pf, sf|pf, sf(pf), c) \text{ done(false)}. \)

From (TCP4.b.12), by \( n2-1=0 \) and \( bf=false \) we have

(TCP4.b.24) \( \text{done(false)} \rightarrow *(n2-1, pf+1, sf, ef) \text{ done(false)} \)

From (TCP4.b.4), (TCP4.b.10), (TCP4.b.23), (TCP4.b.24) by the definition of \( \rightarrow * \) we get

(TCP4.b.20) \( \text{next(TCP(Ftf1,Ftf2))} \rightarrow *(n2, pf, sf, ef) \text{ done(false)} \)

which is [TCP4.b.7]

This finishes the proof of [TCP4.b].

This finishes the proof of [TCP4].

This finishes the proof of the Statement 3 of Lemma 4.
A.7 Lemma 5: Soundness Lemma for Universal Formulas

Lemma 5. (Soundness Lemma for Universal Formulas)

∀F∈Formula, X∈Variable, B1,B2∈Bound:
R(F) ⇒ R(forall X in B1..B2: F)

where

R(F) :
∀re∈RangeEnv, e∈Environment, s∈Stream, d∈N∞, h∈N:
⊢ (re ⊨ F: (h,d)) ∧ d∈N ∧ ∀Y∈dom(e): re(Y).1+p ≤ e(Y) ≤ re(Y).2+p ⇒
( ∀p∈N ∃b∈Bool ∃d′∈N:
   d′≤d+1 ∧ ⊢ T(F) →*(d′,p,s,e) done(b) )
A.8 Lemma 6: Monotonicity of Reduction to done

∀ Ft∈TFormula, p∈N, s∈Stream, c∈Context, b∈Bool :
    ∀ k ≥ p:
        Ft → (p,s↓p,s(p),c) done(b) ⇒
        Ft → (k,s↓k,s(k),c) done(b)

PROOF
----

We take pf,sf,bf,kf arbitrary but fixed, assume

(1) kf ≥ pf

and prove

(2) ∀ Ft∈TFormula ∀ c∈Context:
    Ft → (pf,sf↓pf,s(pf),c) done(bf) ⇒
    Ft → (kf,sf↓kf,s(kf),c) done(bf)

We prove (2) by structural induction over Ft:

C1. Ft=next(TV(X))
----------

We take cf arbitrary but fixed, assume

(1.1) next(TV(X)) → (pf,sf↓pf,s(pf),cf) done(bf)

and prove

(1.2) next(TV(X)) → (kf,sf↓kf,s(kf),cf) done(bf)

By definition of →, the value of bf depends only on cf, which is the same in
(1.1) and (1.2). Hence, (1.1) implies (1.2)

It proves C1.

C2. Ft=next(TN(f)) for some f∈TFormula
----------

We take cf arbitrary but fixed, assume

(2.1) next(TN(f)) → (pf,sf↓pf,s(pf),cf) done(bf)

and prove

(2.2) next(TN(f)) → (kf,sf↓kf,s(kf),cf) done(bf)

From (2.1), by the definition of →, we have

(2.3) f → (pf,sf↓pf,s(pf),cf) done(b₁)

where

(2.4) b₁ = if bf = false true else false.
By the induction hypothesis, from (2.3) we get

\[(2.5) \ f \rightarrow (kf, sf \downarrow kf, s(kf), cf) \text{ done}(b1).\]

From (2.5), by the definition of \(\rightarrow\) and (2.4) we get (2.2).

It proves C2.

C3. \(F_t = \text{next}(TCS(f_1, f_2))\) for some \(f_1, f_2 \in TFormula\)

We take \(cf\) arbitrary but fixed, assume

\[(3.1) \ \text{next}(TCS(f_1, f_2)) \rightarrow (pf, sf \downarrow pf, s(pf), cf) \text{ done}(bf)\]

and prove

\[(3.2) \ \text{next}(TCS(f_1, f_2)) \rightarrow (kf, sf \downarrow kf, s(kf), cf) \text{ done}(bf)\]

From (3.1) we have two alternatives:

(a) We have

\[(3.3) \ bf = \text{false and}\]

\[(3.4) \ f_1 \rightarrow (pf, sf \downarrow pf, s(pf), cf) \text{ done}(false).\]

By the induction hypothesis, from (3.4) we get

\[(3.5) \ f_1 \rightarrow (kf, sf \downarrow kf, s(kf), cf) \text{ done}(false).\]

From (3.5), by the definition of \(\rightarrow\) we get (3.2).

(b) We have

\[(3.6) \ f_1 \rightarrow (pf, sf \downarrow pf, s(pf), cf) \text{ done}(true)\]

\[(3.7) \ f_2 \rightarrow (pf, sf \downarrow pf, s(pf), cf) \text{ done}(bf).\]

By the induction hypothesis, we get from (3.6) and (3.7) respectively

\[(3.8) \ f_1 \rightarrow (kf, sf \downarrow kf, s(kf), cf) \text{ done}(true)\]

\[(3.9) \ f_2 \rightarrow (kf, sf \downarrow pf, s(kf), cf) \text{ done}(bf).\]

From (3.8) and (3.9), by the definition of \(\rightarrow\) we get (3.2).

It proves C3.

C4. \(F_t = \text{next}(TCP(f_1, f_2))\) for some \(f_1, f_2 \in TFormula\)

We take \(cf\) arbitrary but fixed, assume

\[(4.1) \ \text{next}(TCP(f_1, f_2)) \rightarrow (pf, sf \downarrow pf, s(pf), cf) \text{ done}(bf)\]
and prove

\[(4.2) \quad \text{next}(TCP(f_1,f_2)) \rightarrow (kf,sf,kf,sf(kf),cf) \text{ done(bf)}\]

From (4.1) we have three alternatives:

(a) We have

\[\begin{align*}
(4.3) & \quad bf=false \\
(4.4) & \quad f_1 \rightarrow (pf,sf,pf,s(pf),cf) \text{ next(f}_1') \text{ for some } f_1' \in TFormulaCore \\
(4.5) & \quad f_2 \rightarrow (pf,sf,pf,s(pf),cf) \text{ done(false).}
\end{align*}\]

From (4.4) and (4.5) we obtain by the induction hypothesis, respectively,

\[\begin{align*}
(4.6) & \quad f_1 \rightarrow (kf,sf,kf,s(kf),cf) \text{ next(f}_1') \\
(4.7) & \quad f_2 \rightarrow (kf,sf,kf,s(kf),cf) \text{ done(false).}
\end{align*}\]

From (4.6) and (4.7), by the definition of \(\rightarrow\) and (4.3) we get (4.2).

(b) We have

\[\begin{align*}
(4.8) & \quad bf=false \text{ and} \\
(4.9) & \quad f_1 \rightarrow (pf,sf,pf,s(pf),cf) \text{ done(false).}
\end{align*}\]

By the induction hypothesis, from (4.4) we get

\[\begin{align*}
(4.5) & \quad f_1 \rightarrow (kf,sf,kf,s(kf),cf) \text{ done(false).}
\end{align*}\]

From (3.5), by the definition of \(\rightarrow\) we get (4.2).

(c) We have

\[\begin{align*}
(4.6) & \quad f_1 \rightarrow (pf,sf,pf,s(pf),cf) \text{ done(true)} \\
(4.8) & \quad f_2 \rightarrow (pf,sf,pf,s(pf),cf) \text{ done(bf).}
\end{align*}\]

By the induction hypothesis, we get from (3.6) and (3.7) respectively

\[\begin{align*}
(4.9) & \quad f_1 \rightarrow (kf,sf,kf,s(kf),cf) \text{ done(true)} \\
(4.10) & \quad f_2 \rightarrow (kf,sf,pf,s(kf),cf) \text{ done(bf).}
\end{align*}\]

From (4.9) and (4.10), by the definition of \(\rightarrow\) we get (4.2).

It proves C4.

C5. \(F_t=\text{next}(TA(X,b_1,b_2,f))\)

\[\begin{align*}
\text{--------------------------}
\end{align*}\]

We take \(cf\) arbitrary but fixed, assume
(5.1) \(\text{next}(\text{TA}(X,b_1,b_2,f)) \rightarrow(pf, sf\downarrow pf, s(pf), cf) \text{ done}(bf)\)

and prove

[5.2] \(\text{next}(\text{TA}(X,b_1,b_2,f)) \rightarrow(kf, sf\downarrow kf, sf(kf), cf) \text{ done}(bf)\)

(a) \(bf=\text{true}\).

--------------

From (5.1) we have

\(p_1 = b_1(cf)\)
\(p_1 = \infty\)

which immediately imply [5.2].

(b) \(bf=\text{false}\)

--------------

To prove [5.2], we need to find \(p_1^*, p_2^*\) such that

[5.3] \(p_1^* = b_1(cf)\)
[5.4] \(p_2^* = b_2(cf)\)
[5.5] \(p_1^* \neq \infty\)
[5.6] \(\text{next}(\text{TA}(X,p_1^*, p_2^*, f)) \rightarrow(kf, sf\downarrow kf, sf(kf), cf) \text{ done}(false)\)

From (5.1) we know

(5.7) \(p_1 = b_1(cf)\)
(5.8) \(p_2 = b_2(cf)\)
(5.9) \(p_1 \neq \infty\)
(5.10) \(\text{next}(\text{TA}(X,p_1, p_2, f)) \rightarrow(pf, sf\downarrow pf, s(pf), cf) \text{ done}(false)\)

We take \(p_1^* = p_1, p_2^* = p_2\). Then [5.3-5.5] follow from (5.7-5.9) and we need to prove

[5.11] \(\text{next}(\text{TA}(X,p_1, p_2, f)) \rightarrow(kf, sf\downarrow kf, sf(kf), cf) \text{ done}(false)\).

By Def. \(\rightarrow\), to prove [5.11], we need to prove

[5.12] \(kf \geq p_1\)
[5.13] \(\text{next}(\text{TA}(X,p_2, f, fsk)) \rightarrow(kf, sf\downarrow kf, sf(kf), cf) \text{ done}(false)\)

where

(5.14) \(fsk = \{(p_0, f, (cf.1[X\rightarrow p_0], cf.2[X\rightarrow sf\downarrow kf](p_0)))\mid\)
\(p_1 \leq p_0 < \infty \text{ min}\infty(kf, p_2+\infty 1)\})

From (5.10), by the definition of \(\rightarrow\), we know

(5.15) \(pf \geq p_1\)
(5.16) \(\text{next}(\text{TA}(X,p_2, f, fsp)) \rightarrow(pf, sf\downarrow pf, s(pf), cf) \text{ done}(false)\)

where
Then [5.12] follows from (1) and (5.15).

To prove [5.13], by Def. \( \rightarrow \) we need to prove

\[
\exists t \in \mathbb{N}, g \in T_{\text{Formula}}, c \in \text{Context} : \\
(t, g, c) \in f_{s0k} \land \vdash g \rightarrow (k_f, s_f, k_f, c) \text{ done(false)}
\]

where

\[
f_{s0k} = \begin{cases} 
\text{if } k_f > \infty p2 \text{ then } f_{sk} \text{ else } f_{sk} \cup \{(k_f, f, (c[1[X \rightarrow k_f], c[2[X \rightarrow s_f(k_f)]])\} & \\
\text{if } p_f > \infty p2 \text{ then } f_{sp} \text{ else } f_{sp} \cup \{(p_f, f, (c[1[X \rightarrow p_f], c[2[X \rightarrow s_f(p_f)]])\}
\end{cases}
\]

From (5.16) we know that there exist \( t_p \in \mathbb{N}, g_p \in T_{\text{Formula}}, c_p \in \text{Context} \) such that

\[
(5.20) \ (t_p, g_p, c_p) \in f_{s0p} \\
(5.21) \ g_p \rightarrow (p_f, s_f, s_f(p_f), c_p) \text{ done(false)}
\]

where

\[
f_{s0p} = \begin{cases} 
\text{if } p_f > \infty p2 \text{ then } f_{sp} \text{ else } f_{sp} \cup \{(p_f, f, (c[1[X \rightarrow p_f], c[2[X \rightarrow s_f(p_f)]])\}
\end{cases}
\]

Since by (1) \( k_f \geq p_f \), from (5.14) and (5.17) we have

\[
(5.23) \ f_s \subseteq f_{sk}.
\]

Also, we have either

\[
(5.25) \ (p_f, f, (c[1[X \rightarrow p_f], c[2[X \rightarrow s_f(p_f)]]) \in f_{sk} \text{ (when } k_f > p_f, \text{ since } (s_f[p_f](k_f) = s_f(p_f))
\]

or

\[
(5.26) \ (p_f, f, (c[1[X \rightarrow p_f], c[2[X \rightarrow s_f(p_f)]]) \in f_{sk} \text{ (when } k_f = p_f, k_f \leq p_f).
\]

From (5.25) and (5.26) we get

\[
(5.27) \ (p_f, f, (c[1[X \rightarrow p_f], c[2[X \rightarrow s_f(p_f)]]) \in f_{sk}, \text{ when } k_f \geq p_f.
\]

From (1), (5.23), (5.27), (5.28), (5.22) we get

\[
(5.28) \ f_{s0p} \subseteq f_{sk}.
\]

Then from (5.20) we get

\[
(5.29) \ (t_p, g_p, c_p) \in f_{sk}.
\]

From (5.21) and (2) we get

\[
(5.30) \ g_p \rightarrow (k_f, s_f, k_f, s_f(k_f), c_p) \text{ done(false)}
\]
From (5.29) and (5.30) we obtain [5.18].

It proves C5.

It finishes the proof of Lemma 6.
**A.9 Lemma 7: Shifting Lemma**

\( \forall f \in T_{\text{FormulaCore}}, n,p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, b \in \text{Bool}: \)

\[ n > 0 \Rightarrow \]

next\( (f) \rightarrow ^* (n+1,p,s,e) \) done\( (b) \Rightarrow \) next\( (f) \rightarrow ^* (n,p+1,s,e) \) done\( (b) \)

**Proof**

-----

We take \( f,n,p,s,e,b \) arbitrary but fixed, assume

1. \( n > 0 \)
2. next\( (f) \rightarrow ^* (n+1,p,s,e) \) done\( (b) \)

and show

3. next\( (f) \rightarrow ^* (n,p+1,s,e) \) done\( (b) \).

From (2), by the definition of \( \rightarrow ^* \), there exists \( F_t' \in T_{\text{Formula}} \) such that

4. next\( (f) \rightarrow (p,s,p,s(p),c) F_t' \)
5. \( F_t' \rightarrow ^* (n,p+1,s,e) \) done\( (b) \)

where

6. \( c = (e,((X,s(e(X)))) \mid X \in \text{dom}(e))) \).

Since \( n > 0 \) by (1), we have that \( F_t' \) is a 'next' formula, say next\( (f') \).

Then from (5), by the definition of \( \rightarrow ^* \), we know that there exists \( F_t'' \in T_{\text{Formula}} \) such that

7. next\( (f') \rightarrow (p+1,s\downarrow (p+1),s(p+1),c) F_t'' \)
8. \( F_t'' \rightarrow ^* (n-1,p+2,s,e) \) done\( (b) \).

In order to prove [3], by the definition of \( \rightarrow ^* \), we need to find such a \( F_t0 \in T_{\text{Formula}} \) that

9. next\( (f) \rightarrow (p+1,s\downarrow (p+1),s(p+1),c) F_t0 \)
10. \( F_t0 \rightarrow ^* (n-1,p+2,s,e) \) done\( (b) \).

We take \( F_t0 = F_t'' \). Then [10] follows from (8). We only need to prove [9]:

Given

4. next\( (f) \rightarrow (p,s\downarrow p,s(p),c) \) next\( (f') \)
7. next\( (f') \rightarrow (p+1,s\downarrow (p+1),s(p+1),c) F_t'' \)

Prove:

9. next\( (f) \rightarrow (p+1,s\downarrow (p+1),s(p+1),c) F_t'' \).

It follows from Lemma 8.
Lemma 8 (Triangular Reduction Lemma)

\[ \forall G_1, G_2 \in TFormulaCore, F_t \in TFormula, p \in \mathbb{N}, s \in \text{Stream}, c \in \text{Context} : \]
\[ \text{next}(G_1) \rightarrow (p, s \downarrow p, s(p), c) \text{next}(G_2) \land \text{next}(G_2) \rightarrow (p+1, s \downarrow (p+1), s(p+1), c) F_t \]
\[ \Rightarrow \text{next}(G_1) \rightarrow (p+1, s \downarrow (p+1), s(p+1), c) F_t. \]

Proof

\[ \Phi \subseteq TFormulaCore \]
\[ \Phi(G_1) :\]
\[ \forall G_2 \in TFormulaCore, F_t \in TFormula, p \in \mathbb{N}, s \in \text{Stream}, c \in \text{Context} : \]
\[ \text{next}(G_1) \rightarrow (p, s \downarrow p, s(p), c) \text{next}(G_2) \land \text{next}(G_2) \rightarrow (p+1, s \downarrow (p+1), s(p+1), c) F_t \]
\[ \Rightarrow \text{next}(G_1) \rightarrow (p+1, s \downarrow (p+1), s(p+1), c) F_t. \]

We prove

\[ (G) \forall G' \in TFormulaCore : \Phi(G'). \]

Case (C1) \( G' = TN(F_t) \) for some \( F_t \in TFormula \)

We show

\[ \Phi(G') \]

Take \( F_2f, F_{tf}, pf, sf, cf \) arbitrary but fixed.

Assume

\[ (C1.1) \text{next}(TN(F_t)) \rightarrow (pf, sf \downarrow pf, sf(pf), cf) \text{next}(G_{2f}) \]
\[ (C1.2) \text{next}(G_{2f}) \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), cf) F_{tf} \]

Show

\[ [C1.a] \text{next}(TN(F_t)) \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), cf) F_{tf}. \]

From (C1.1) and Def. \( \rightarrow \), we know for some \( G_2' \in TFormula \)

\[ (C1.3) G_{2f} = TN(\text{next}(G_2')) \]
\[ (C1.4) \text{next}(TN(F_t)) \rightarrow (pf, sf \downarrow pf, sf(pf), cf) \text{next}(TN(\text{next}(G_2'))) \]
\[ (C1.5) F_t \rightarrow (pf, sf \downarrow pf, sf(pf), cf) \text{next}(G_2') \]

From (C1.2,C1.3), we thus have

\[ (C1.6) \text{next}(TN(\text{next}(G_2'))) \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), cf) F_{tf} \]

From (C1.5) and Def. \( \rightarrow \), we know for some \( G \in TFormulaCore \)

\[ (C1.7) F_t = \text{next}(G) \]
(C.8) \( \text{next}(G) \rightarrow (pf, sf \downarrow pf, sf(pf), cf) \text{ next}(G'_2) \)

From (C.7) and [C.1.a], it suffices to show

[C.1.b] \( \text{next}(\text{TN}(\text{next}(G))) \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), cf) \text{ Ftf.} \)

From (C.1,C.8) and the induction assumption, we know \( \Phi(G) \) and thus

(C.9)

\[
\forall G \in \text{TFormulaCore}, Ft \in \text{TFormula}, p \in \mathbb{N}, s \in \text{Stream}, c \in \text{Context} : \\
\text{next}(G) \rightarrow (p, s \downarrow p, s(p), c) \text{ next}(G2) \land \text{next}(G2) \rightarrow (p+1, s \downarrow (p+1), s(p+1), c) \text{ Ft} \\
\Rightarrow \\
\text{next}(G) \rightarrow (p+1, s \downarrow (p+1), s(p+1), c) \text{ Ft.}
\]

From (C.6) and Def.\( \rightarrow \), we have 3 cases.

Case C.1.c1. there exists some \( Fc' \in \text{TFormulaCore} \) such that

(C.1.c1.1) \( \text{next}(G'_2) \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), cf) \text{ next}(Fc') \)
(C.1.c1.2) \( \text{Ftf=next}(\text{TN}(\text{next}(Fc'))) \)

From (C.1.c1.2) and [C.1.b], it suffices thus to show

[C.1.c1.b] \( \text{next}(\text{TN}(\text{next}(G))) \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), cf) \text{ next}(\text{TN}(\text{next}(Fc'))) \)

From (C.9), (C.8), (C.1.c1.1), we have

(C.1.c1.3) \( \text{next}(G) \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), cf) \text{ next}(Fc') \)

From (C.1.c1.3) and Def.\( \rightarrow \), we know [C.1.c1.b].

This proves the case C.1.c1.

Case C.1.c2. we have

(C.1.c2.1) \( \text{next}(G'_2) \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), cf) \text{ done(true)} \)
(C.1.c2.2) \( \text{Ftf=done(false)} \)

From (C.1.c2.2) and [C.1.b], it suffices thus to show

[C.1.c2.b] \( \text{next}(\text{TN}(\text{next}(G))) \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), cf) \text{ done(false)} \)

From (C.9), (C.8), (C.1.c2.1), we have

(C.1.c2.3) \( \text{next}(G) \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), cf) \text{ done(true).} \)

From (C.1.c2.3) and Def.\( \rightarrow \), we know [C.1.c2.b].

This proves the case C.1.c2.

Case C.1.c3. we have
(C1.c3.1) next(G') \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), cf) \text{ done}(false)
(C1.c3.2) Ftf=done(true)

It suffices thus to show

[C1.c3.b] next(TN(next(G))) \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), cf) \text{ done}(true)

From (C1.9), (C1.8) (C1.c3.1), we have

(C1.c3.3) next(G) \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), cf) \text{ done}(false).

From (C1.c3.3) and Def. \rightarrow, we know [C1.c3.b].

This proves the case C1.c3.

This finishes the proof of case C1.

------------------------

Case (C2) G' = TCS(Ft1,Ft2) for some Ft1,Ft2 \in TFormula.

We show

\Phi(G')

Take F2f,Ftf,pf,sf,cf arbitrary but fixed.

Assume

(C2.1) next(TCS(Ft1,Ft2)) \rightarrow (pf, sf \downarrow pf, sf(pf), cf) \text{ next}(G2f)
(C2.2) next(G2f) \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), cf) \text{ Ftf}

Show

[C2.a] next(TCS(Ft1,Ft2)) \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), cf) \text{ Ftf}.

From (C2.1), by Def. \rightarrow, we have two cases:

Case C2.c1. There exists Fc1 \in TFormulaCore such that

--------
(C2.c1.1) G2f = TCS(next(Fc1),Ft2)
(C2.c1.2) next(TCS(Ft1,Ft2)) \rightarrow (pf, sf \downarrow pf, sf(pf), cf) \text{ next(TCS(next(Fc1),Ft2))}
(C2.c1.3) Ft1 \rightarrow (pf, sf \downarrow pf, sf(pf), cf) \text{ next}(Fc1)

From (C2.2) and (C2.c1.1) we have

(C2.c1.4) next(TCS(next(Fc1),Ft2)) \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), cf) \text{ Ftf}.

From (C2.c1.3) and Def. \rightarrow, we know for some Fc0 \in TFormulaCore

--------
(C2.c1.5) Ft1 = next(Fc0)
(C2.c1.6) next(Fc0) \rightarrow (pf, sf \downarrow pf, sf(pf), cf) \text{ next}(Fc1)
From (C2.c1.5) and [C2.a], we need to show

\[ [C2.c1.b] \text{ next(TCS(next(Fc0),Ft2))) } \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) \text{ Ftf.} \]

From (C2), (C2.c1.5) and the induction hypothesis, we know \( \Phi(Fc0) \) and thus

\[ (C2.c1.7) \]

\[
\forall G2 \in TFormulaCore, Ft \in TFormula, p \in \mathbb{N}, s \in Stream, c \in Context : \\
\text{next(Fc0) } \rightarrow (p, s\downarrow p, s(p), c) \text{ next(G2) } \land \text{ next(G2) } \rightarrow (p+1, s\downarrow(p+1), s(p+1), c) \text{ Ft} \\
\Rightarrow \text{next(Fc0) } \rightarrow (p+1, s\downarrow(p+1), s(p+1), c) \text{ Ft.} \\
\]

From (C2.c1.4), we have the following cases.

Case C2.c1.c1. There exists Fc' \in TFormulaCore such that

\[ (C2.c1.c1.1) \text{ Ftf = next(TCS(next(Fc'),Ft2))) } \]
\[ (C2.c1.c1.2) \text{ next(TCS(next(Fc1),Ft2))) } \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) \text{ next(TCS(next(Fc'),Ft2)).} \]
\[ (C2.c1.c1.3) \text{ next(Fc1) } \rightarrow (pf+1, s\downarrow(p+1), s(p+1), c) \text{ next(Fc').} \]

From (C2.c1.c1.1) and [C2.c1.b], we need to show

\[ [C2.c1.c1.b] \text{ next(TCS(next(Fc0),Ft2)) } \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) \text{ next(TCS(next(Fc'),Ft2)).} \]

In this case from (C2.c1.6), (C2.c1.c1.3), and (C2.c1.7) we have

\[ (C2.c1.c1.4) \text{ next(Fc0) } \rightarrow (p+1, s\downarrow(p+1), s(p+1), c) \text{ next(Fc').} \]

From (C2.c1.c1.4), by the definition of \( \rightarrow \), we get [C2.c1.c1.b].

This proves the case C2.c1.c1.

Case C2.c1.c2.

\[ (C2.c1.c2.1) \text{ Ftf = done(false)} \]
\[ (C2.c1.c2.2) \text{ next(TCS(next(Fc1),Ft2))) } \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) \text{ done(false).} \]
\[ (C2.c1.c2.3) \text{ next(Fc1) } \rightarrow (pf+1, s\downarrow(p+1), s(p+1), c) \text{ done(false).} \]

From (C2.c1.c2.1) and [C2.c1.b], we need to show

\[ [C2.c1.c2.b] \text{ next(TCS(next(Fc0),Ft2)) } \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) \text{ done(false).} \]

From (C2.c1.6), (C2.c1.c2.3) and (C2.c1.7) we have

\[ (C2.c1.c2.4) \text{ next(Fc0) } \rightarrow (pf+1, s\downarrow(pf+1), sf(pf+1), cf) \text{ done(false).} \]

From (C2.c1.c2.4), by the definition of \( \rightarrow \), we get [C2.c1.c2.b].

This proves the case C2.c1.c2.
Case C2.c1.c3. There exists Ft2' ∈ TFormula such that

(C2.c1.c3.1) Ftf = Ft2'
(C2.c1.c3.2) next(TCS(next(Fc1),Ft2)) → (pf+1,sf↓(pf+1),sf(pf+1),cf) Ft2'.
(C2.c1.c3.3) next(Fc1) → (pf+1,sf↓(pf+1),sf(pf+1),cf) done(true).
(C2.c1.c3.4) Ft2 → (pf+1,sf↓(pf+1),sf(pf+1),cf) Ft2'.

From (C2.c1.c3.1) and [C2.c1.b], we need to show

[C2.c1.c3.b] next(TCS(next(Fc0),Ft2)) → (pf+1,sf↓(pf+1),sf(pf+1),cf) Ft2'.

From (C2.c1.6), (C2.c1.c3.3), and (C2.c1.7) we have

(C2.c1.c3.5) next(Fc0) → (pf+1,sf↓(pf+1),sf(pf+1),cf) done(true).

From (C2.c1.c3.5) and (C2.c1.c3.4), by Def.→, we get [C2.c1.c3.b].

This proves the case C2.c1.c2.

This proves the case C2.c1.

Case C2.c2.

Recall that we consider alternatives of G2f in

(C2.1) next(TCS(Ft1,Ft2)) → (pf,sf↓pf,sf(pf),cf) next(G2f)

Case C2.c1 considered the case when G2f = TCS(next(Fc1),Ft2).

According to Def.→, the other alternative for G2f is the following:

There exists G2' ∈ TFormulaCore such that

(C2.c2.1) G2f = G2'
(C2.c2.2) next(TCS(Ft1,Ft2)) → (pf,sf↓pf,sf(pf),cf) next(G2')
(C2.c2.3) Ft1 → (pf,sf↓pf,sf(pf),cf) done(true)
(C2.c2.4) Ft2 → (pf,sf↓pf,sf(pf),cf) next(G2')

From (C2.2) and (C2.c2.1) we have

(C2.c2.5) next(G2') → (pf+1,sf↓(pf+1),sf(pf+1),cf) Ftf.

From (C2.c2.3) and Def.→, we know for some Fc1 ∈ TFormulaCore

(C2.c2.6) Ft1 = next(Fc1)
(C2.c2.7) next(Fc1) → (pf,sf↓pf,sf(pf),cf) done(true)

From (C2.c2.4) and Def.→, we know for some Fc2 ∈ TFormulaCore

(C2.c2.8) Ft2 = next(Fc2)
(C2.c2.9) next(Fc2) → (pf,sf↓pf,sf(pf),cf) next(G2')

From (C2.c2.6), (C2.c2.8) and [C2.a], we need to show
From (C2.c2.7), by Lemma 6, we know

\((C2.c2.10) \quad \text{next}(Fc1) \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ done(true)}\).

From (C2), (C2.c2.8) and the induction hypothesis, we know \(\Phi(Fc2)\) and thus

\((C2.c2.11)\)

\(\forall G2\in TFormulaCore, Ft\in TFormula, p\in \mathbb{N}, s\in Stream, c\in Context : \quad \text{next}(Fc2) \rightarrow (p, s, p, c) \text{ next}(G2) \land \text{next}(G2) \rightarrow (p+1, s\downarrow (p+1), s(p+1), c) \text{ Ft} \Rightarrow \quad \text{next}(Fc2) \rightarrow (p+1, s\downarrow (p+1), s(p+1), c) \text{ Ft.}\)

From (C2.c2.9), (C2.c2.5), and (C2.c2.11), we get

\((C2.c2.11) \quad \text{next}(Fc2) \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ Ftf.}\)

From (C2.c2.10) and (C2.c2.11), by Def. \(\rightarrow\), we get \([C2.c2.b]\).

This proves the case C2.c2.

This finishes the proof of case C2.

-----------------------------------

Case (C3) \(G' = TCP(Ft1, Ft2)\) for some \(Ft1, Ft2 \in TFormula\).

We show

\(\Phi(G')\)

Take \(F2f, Ftf, pf, sf, cf\) arbitrary but fixed.

Assume

\((C3.1) \quad \text{next}(TCP(Ft1, Ft2)) \rightarrow (pf, sf\downarrow pf, sf(pf), cf) \text{ next}(G2f)\)

\((C3.2) \quad \text{next}(G2f) \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ Ftf}\)

Show

\([C3.a] \quad \text{next}(TCP(Ft1, Ft2)) \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ Ftf.}\)

From (C3.1), by Def. \(\rightarrow\), we have three cases.

Case C3.c1

-------------

There exists \(Fc1, Fc2 \in TFormulaCore\) such that

\((C3.c1.1) \quad G2f = TCP(next(Fc1), next(Fc2))\)

\((C3.c1.2) \quad Ft1 \rightarrow (pf, sf\downarrow pf, sf(pf), cf) \text{ next}(Fc1)\)

\((C3.c1.3) \quad Ft2 \rightarrow (pf, sf\downarrow pf, sf(pf), cf) \text{ next}(Fc2)\)

\((C3.c1.4) \quad \text{next}(TCP(Ft1, Ft2)) \rightarrow (pf, sf\downarrow pf, sf(pf), cf) \text{ next}(TCP(next(Fc1), next(Fc2)))\)
From (C3.2) and (C3.c1.1) we have

\[(C3.c1.5) \quad \text{next}(\text{TCP}(\text{next}(Fc1),\text{next}(Fc2))) \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) \ Ftf\]

From (C3.c1.2) and Def.→, we know for some $Fc1' \in \text{TFormulaCore}$

\[(C3.c1.6) \quad \text{Ft1} = \text{next}(Fc1') \]
\[(C3.c1.7) \quad \text{next}(Fc1') \rightarrow (pf, sf\downarrow pf, sf(pf), cf) \ \text{next}(Fc1)\]

From (C3.c1.3) and Def.→, we know for some $Fc2' \in \text{TFormulaCore}$

\[(C3.c1.8) \quad \text{Ft2} = \text{next}(Fc2') \]
\[(C3.c1.9) \quad \text{next}(Fc2') \rightarrow (pf, sf\downarrow pf, sf(pf), cf) \ \text{next}(Fc2)\]

From (C3.c1.6), (C3.c1.8) and [C3.a], we need to show

\[(C3.c1.10) \quad \forall G2 \in \text{TFormulaCore}, Ft \in \text{TFormula}, p \in \mathbb{N}, s \in \text{Stream}, c \in \text{Context} : \]
\[\text{next}(Fc1') \rightarrow (p, s\downarrow p, s(p), c) \ \text{next}(G2) \land \text{next}(G2) \rightarrow (p+1, s\downarrow(p+1), s(p+1), c) \ Ft \Rightarrow \]
\[\text{next}(Fc1') \rightarrow (p+1, s\downarrow(p+1), s(p+1), c) \ Ft.\]

From (C3), (C3.c1.6) and the induction hypothesis, we know $\Phi(Fc1')$ and thus

\[(C3.c1.11) \quad \forall G2 \in \text{TFormulaCore}, Ft \in \text{TFormula}, p \in \mathbb{N}, s \in \text{Stream}, c \in \text{Context} : \]
\[\text{next}(Fc2') \rightarrow (p, s\downarrow p, s(p), c) \ \text{next}(G2) \land \text{next}(G2) \rightarrow (p+1, s\downarrow(p+1), s(p+1), c) \ Ft \Rightarrow \]
\[\text{next}(Fc2') \rightarrow (p+1, s\downarrow(p+1), s(p+1), c) \ Ft.\]

From (C3), (C3.c1.8) and the induction hypothesis, we know $\Phi(Fc2')$ and thus

\[(C3.c1.12) \quad \text{next}(\text{TCP}(\text{next}(Fc1'),\text{next}(Fc2'))) \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) \ \text{Ftf}.\]

From (C3), (C3.c1.6) and the induction hypothesis, we know $\Phi(Fc1')$ and thus

\[(C3.c1.13) \quad \text{next}(\text{TCP}(\text{next}(Fc1'),\text{next}(Fc2'))) \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) \ \text{Ftf}.\]

From (C3), (C3.c1.6) and the induction hypothesis, we know $\Phi(Fc2')$ and thus

\[(C3.c1.14) \quad \text{next}(\text{TCP}(\text{next}(Fc1'),\text{next}(Fc2'))) \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) \ \text{Ftf}.\]

From (C3.c1.1) and [C3.c1.b] we need to prove

\[(C3.c1.15) \quad \text{next}(\text{TCP}(\text{next}(Fc1'),\text{next}(Fc2'))) \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) \ \text{next}(\text{TCP}(\text{next}(Fc1'),\text{next}(Fc2')))\]

From (C3.c1.7), (C3.c1.12), and (C3.c1.10) we have
(C3.c1.c1.5) next(Fc1') \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) next(Fc1'').

From (C3.c1.9), (C3.c1.c1.3), and (C3.c1.11) we have

(C3.c1.c1.6) next(Fc2') \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) next(Fc2'')

From (C3.c1.c1.5) and (C3.c1.c1.6), by Def. \rightarrow we get [C3.c1.c1.b].

This proves case the C3.c1.c1.

Case C3.c1.c2
-------------
There exist Fc1'' \in TFormulaCore such that

(C3.c1.c2.1) Ftf = next(Fc1'')
(C3.c1.c2.2) next(Fc1) \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) next(Fc1'')
(C3.c1.c2.3) next(Fc2) \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) done(true)
(C3.c1.c2.4) next(TCP(next(Fc1),next(Fc2)))
\rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) next(Fc1'')

From (C3.c1.c2.1) and [C3.c1.b] we need to prove

[C3.c1.c2.b] next(TCP(next(Fc1'),next(Fc2')))
\rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) next(Fc1'').

From (C3.c1.7), (C3.c1.c2.2), and (C3.c1.10) we have

(C3.c1.c2.5) next(Fc1') \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) next(Fc1'').

From (C3.c1.9), (C3.c1.c2.3), and (C3.c1.11) we have

(C3.c1.c2.6) next(Fc2') \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) done(true).

From (C3.c1.c2.5) and (C3.c1.c2.6), by Def. \rightarrow we get [C3.c1.c2.b].

This proves the case C3.c1.c2.

Case C3.c1.c3
-------------
There exist Fc1''' \in TFormulaCore such that

(C3.c1.c3.1) Ftf = done(false)
(C3.c1.c3.2) next(Fc1) \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) next(Fc1''')
(C3.c1.c3.3) next(Fc2) \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) done(false)
(C3.c1.c3.4) next(TCP(next(Fc1),next(Fc2)))
\rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) done(false)

From (C3.c1.c3.1) and [C3.c1.b] we need to prove

[C3.c1.c3.b] next(TCP(next(Fc1'),next(Fc2')))
\rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) done(false).
From (C3.c1.7), (C3.c1.c3.2), and (C3.c1.10) we have

\[(C3.c1.c3.5) \text{next}(F_{c1'}) \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) \text{next}(F_{c1''}).\]

From (C3.c1.9), (C3.c1.c3.3), and (C3.c1.11) we have

\[(C3.c1.c3.6) \text{next}(F_{c2'}) \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) \text{done}(false).\]

From (C3.c1.c3.5) and (C3.c1.c3.6), by Def. we get \([C3.c1.c3].\)

This proves the case C3.c1.c3.

Case C3.c1.c4
-------------

\[(C3.c1.c4.1) F_{tf} = \text{done}(false)\]
\[(C3.c1.c4.2) \text{next}(F_{c1}) \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) \text{done}(false)\]
\[(C3.c1.c4.3) \text{next}(TCP(\text{next}(F_{c1}), \text{next}(F_{c2}))) \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) \text{done}(false)\]

From (C3.c1.c4.1) and \([C3.c1.b]\) we need to prove

\([C3.c1.c4.b] \text{next}(TCP(\text{next}(F_{c1'}), \text{next}(F_{c2'}))) \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) \text{done}(false).\]

From (C3.c1.7), (C3.c1.c4.2), and (C3.c1.10) we have

\[(C3.c1.c4.5) \text{next}(F_{c1'}) \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) \text{done}(false).\]

From (C3.c1.c4.5) by Def. we get \([C3.c1.c4.b]\).

This proves the case C3.c1.c4.

Case C3.c1.c5
-------------

There exist \(F_{c2''}\in T\text{FormulaCore}\) such that

\[(C3.c1.c5.1) F_{tf} = \text{next}(F_{c2''})\]
\[(C3.c1.c5.2) \text{next}(F_{c1}) \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) \text{done}(true)\]
\[(C3.c1.c5.3) \text{next}(F_{c2}) \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) \text{next}(F_{c2''})\]
\[(C3.c1.c5.4) \text{next}(TCP(\text{next}(F_{c1}), \text{next}(F_{c2}))) \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) \text{next}(F_{c2''})\]

From (C3.c1.c5.1) and \([C3.c1.b]\) we need to prove

\([C3.c1.c5.b] \text{next}(TCP(\text{next}(F_{c1'}), \text{next}(F_{c2'}))) \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) \text{next}(F_{c2''}).\]

From (C3.c1.7), (C3.c1.c5.2), and (C3.c1.10) we have

\[(C3.c1.c5.5) \text{next}(F_{c1'}) \rightarrow (pf+1, sf\downarrow(pf+1), sf(pf+1), cf) \text{done}(true).\]
From (C3.c1.9), (C3.c1.c5.3), and (C3.c1.11) we have

\[(C3.c1.c5.6) \text{ next}(Fc2') \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), cf) \text{ next}(Fc2'')\].

From (C3.c1.c5.5) and (C3.c1.c5.6), by Def. \(\rightarrow\) we get [C3.c1.c5.b].

This proves the case C3.c1.c3.

This proves the case C3.c1.

Case C3.c2
----------

\[(C3.c2.1) \text{ Ft1} \rightarrow (pf, sf \downarrow pf, sf(pf), cf) \text{ next}(G2f)\]
\[(C3.c2.2) \text{ Ft2} \rightarrow (pf, sf \downarrow pf, sf(pf), cf) \text{ done(true)}\]

From (C3.c2.1) and Def. \(\rightarrow\), we know for some Fc1' \(\in\) TFormulCore

\[(C3.c2.3) \text{ Ft1} = \text{ next}(Fc1')\]
\[(C3.c2.4) \text{ next}(Fc1') \rightarrow (pf, sf \downarrow pf, sf(pf), cf) \text{ next}(G2f)\]

From (C3.c2.2) and Def. \(\rightarrow\), we know for some Fc2' \(\in\) TFormulCore

\[(C3.c2.5) \text{ Ft2} = \text{ next}(Fc2')\]
\[(C3.c2.6) \text{ next}(Fc2') \rightarrow (pf, sf \downarrow pf, sf(pf), cf) \text{ done(true)}\]

From (C3.c2.3), (C3.c2.5) and [C3.a], we need to show

[C3.c2.b] next(TCP(next(Fc1'),next(Fc2')))) \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), cf) \text{ Ftf.}

From (C3), (C3.c2.2) and the induction hypothesis, we know \(\Phi(Fc1')\) and thus

(C3.c2.7)
\[
\forall G2 \in TFormulCore, Ft \in TFormula, p \in \mathbb{N}, s \in \text{Stream}, c \in \text{Context} : \\
\text{next}(Fc1') \rightarrow (p, s \uparrow p, s(p), c) \text{ next}(G2) \land \text{next}(G2) \rightarrow (p+1, s \uparrow (p+1), s(p+1), c) \text{ Ft} \\
\Rightarrow \\
\text{next}(Fc1') \rightarrow (p+1, s \uparrow (p+1), s(p+1), c) \text{ Ft.}
\]

From (C3.c2.4), (C3.2), and (C3.c2.7) we get

\[(C3.c2.8) \text{ next}(Fc1') \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), cf) \text{ Ftf.}\]

From (C3.c2.6), by Lemma 6, we get

\[(C3.c3.9) \text{ next}(Fc2') \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), cf) \text{ done(true)}.\]

From (C3.c2.8) and (C3.c2.9), by Def. \(\rightarrow\), we get [C3.c2.b].

This proves the case C3.c2
Case C3.c3

(C3.c3.1) $Ft_1 \rightarrow (pf,sf,pf(pf),cf)$ done(true)

(C3.c3.2) $Ft_2 \rightarrow (pf,sf,pf(pf),cf)$ next(G2f)

This case can be proved similarly to case C3.c2.

This finishes the proof of C3.

Case (C4) $G' = TA(X,b_1,b_2,Ft)$ for some $X \in \text{Variable}$, $b_1,b_2 \in \text{BoundValue}$, $Ft \in \text{TFormula}$.

We show

$\Phi(G')$

Take $F2f,Ftf,pf,sf,cf$ arbitrary but fixed.

Assume

(C4.1) $\text{next}(TA(X,b_1,b_2,Ft)) \rightarrow (pf,sf,pf(pf),cf)$ next(G2f)

(C4.2) $\text{next}(G2f) \rightarrow (pf+1,sf(pf+1),sf(pf+1),cf)$ Ftf

Show

[C4.a] $\text{next}(TA(X,b_1,b_2,Ft)) \rightarrow (pf+1,sf(pf+1),sf(pf+1),cf)$ Ftf.

From (C4.1), by Def. $\rightarrow$, we have that there exist $p_1,p_2 \in \mathbb{N}$ such that

(C4.3) $p_1 = b_1(cf)$

(C4.4) $p_2 = b_2(cf)$

(C4.5) $p_1 \neq \infty$

(C4.6) $\text{next}(TA_0(X,p_1,p_2,Ft)) \rightarrow (pf,sf,pf(pf),cf)$ next(G2f)

To prove [C4.a], by the definition of $\rightarrow$, we would have two alternatives: $Ftf=\text{done(true)}$ or $Ftf \neq \text{done(true)}$. But the case $Ftf=\text{done(true)}$ is impossible because of (C4.5). Hence, we assume $Ftf \neq \text{done(true)}$ and prove

[C4.a.1] $p_1 = b_1(cf)$

[C4.a.2] $p_2 = b_2(cf)$

[C4.a.3] $p_1 \neq \infty$

[C4.a.4] $\text{next}(TA_0(X,p_1,p_2,Ft)) \rightarrow (pf+1,sf(pf+1),sf(pf+1),cf)$ Ftf.

[C4.a.1-3] are immediately proved due to (C4.3-5).

To prove [C4.a.4], from (C4.6), by Def. $\rightarrow$, we consider two cases.

Case C4.c1.

In this case from (C4.6) we have
From (C4.2) and (C4.c1.3) we get [C4.a.4]

This finishes the proof of C4.c1.

Case C4.c2.

------------
In this case from (C4.6) we have

(C4.c2.1) \( pf \geq p1 \)
(C4.c2.2) \( fs = \{ (p0,Ft,(cf.1[X\mapsto p0], cf.2[X\mapsto sf(p0)]) ) \mid p1 \leq p0 < \min \infty (pf,p2+\infty 1) \} \)
(C4.c2.3) \( \text{next(TA1}(X,p2,Ft,fs)) \rightarrow (pf,sf\downarrow pf, sf(pf), cf) \) \( \text{next(G2f)} \)

From (C4.c2.3), by the definition of \( \rightarrow \), we know

(C4.c2.4) \( G2f = TA1}(X,p2,Ft,fs1) \), where

(C4.c2.5) \( fs0 = \)
- if \( pf > \infty \) \( p2 \) then \( fs \)
- else \( fs \cup \{ (pf,Ft,(cf.1[X\mapsto pf], cf.2[X\mapsto sf(pf)]) ) \} \)
(C4.c2.6) \( \neg \exists t \in \mathbb{N}, g \in TFormula, c \in Context: (t,g,c) \in fs0 \land \vdash g \rightarrow (pf,sf\downarrow pf, sf(pf), c) \) \( \text{done(false)} \)
(C4.c2.7) \( fs1 = \{ (t,\text{next}(fc),c) \in TInstance \mid \exists g \in TFormula: (t,g,c) \in fs0 \land \vdash g \rightarrow (pf,sf\downarrow pf, sf(pf), c) \) \( \text{next}(fc) \} \)
(C4.c2.8) \( \neg (fs1 = \emptyset \land pf \geq \infty p2) \)

From (C4.2) and (C4.c2.4) we have

(C4.c2.9) \( \text{next(TA1}(X,p2,Ft,fs1)) \rightarrow (pf+1,sf\downarrow (pf+1), sf(pf+1), cf) \) \( \text{Ftf.} \)

Recall that we need to prove

[C4.a.4] \( \text{next(TA0}(X,p1,p2,Ft)) \rightarrow (pf+1,sf\downarrow (pf+1), sf(pf+1), cf) \) \( \text{Ftf.} \)

By definition of \( \rightarrow \) and (C4.c2.1), in order to prove [C4.a.4], we need to prove

[C4.a.5] \( \text{next(TA1}(X,p2,Ft,fs')) \rightarrow (pf+1,sf\downarrow (pf+1), sf(pf+1), cf) \) \( \text{Ftf} \),

where

(C4.c2.10) \( fs' = \{ (p0,Ft,(cf.1[X\mapsto p0], cf.2[X\mapsto sf(p0)]) ) \mid p1 \leq p0 < \min \infty (pf+1,p2+\infty 1) \} \).

Note that
- if \( pf > \infty \) \( p2 \) then \( \min \infty (pf+1,p2+\infty 1) = \min \infty (pf,p2+\infty 1) \)
- else \( \min \infty (pf+1,p2+\infty 1) = pf+1 \).

Therefore, from (C4.c2.2), (C4.c2.5), and (C4.c2.10) we have

(C4.c2.11) \( fs'=fs0 \).
Hence, we need to prove

\[ \text{next}(\text{TA1}(X,p2,Ft,fs0)) \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ Ftf}, \]

We prove \[\text{C4.a.6}\] by case distinction over \text{Ftf}.

\text{Ftf} = \text{done}(false)

------------------

In this case, from (C4.c2.9) we get

\( \text{(C4.c2.12) next}(\text{TA1}(X,p2,Ft,fs1)) \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ done}(false) \)

From (C4.c2.12), by the definition of \(\rightarrow\) for forall we have

\( \exists t \in \mathbb{N}, g \in \mathbf{TFormula}, c \in \mathbf{Context}: \)

\( (t,g,c) \in \mathbf{fs1'} \wedge \vdash g \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), c) \text{ done}(false) \)

where

\( \mathbf{fs1'} = \)

\( \begin{cases} 
\text{if pf+1} >_{\infty} p2 \text{ then } \mathbf{fs1} \\
\text{else } \mathbf{fs1} \cup \{(pf+1,Ft,(cf.1[X\mapsto pf+1],cf.2[X\mapsto sf(pf+1)]))\}. 
\end{cases} \)

Take \((t1,g1,c1)\) which is a witness for (C4.c2.13). That means, we have

\( \mathbf{(C4.c2.13')} (t1,g1,c1) \in \mathbf{fs1'} \) and

\( \mathbf{(C4.c2.13'')} g1 \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), c1) \text{ done}(false). \)

Assume first

\( \mathbf{(C4.c2.15) pf+1} >_{\infty} p2, \) which from (C4.c2.14) gives

\( \mathbf{(C4.c2.16) (t1,g1,c1) \in \mathbf{fs1}}. \)

To show \[\text{C4.a.6}\], we need to prove

\[ \mathbf{(C4.a.7)} \exists t \in \mathbb{N}, g \in \mathbf{TFormula}, c \in \mathbf{Context}: \]

\( (t,g,c) \in \mathbf{fs0'} \wedge \vdash g \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), c) \text{ done}(false) \)

where

\( \mathbf{(C4.c2.17) fs0'} = \)

\( \begin{cases} 
\text{if pf+1} >_{\infty} p2 \text{ then } \mathbf{fs0} \\
\text{else } \mathbf{fs0} \cup \{(pf+1,Ft,(cf.1[X\mapsto pf+1],cf.2[X\mapsto sf(pf+1)]))\}. 
\end{cases} \)

From (C4.c2.15) and (C4.c2.17), we have

\( \mathbf{(C4.c2.18) fs0'=fs0}. \)

from (C4.c2.16), by (C4.c2.7), there exists \(g0 \in \mathbf{TFormula}\) and \(fc1 \in \mathbf{TFormulaCore}\) such that

\( \mathbf{(C4.c2.19) g1=next(fc1)} \)

109
(C4.c2.20) \((t_1,g_0,c_1) \in s_0\)
(C4.c2.21) \(\vdash g_0 \rightarrow (p_f,s_f,p_f,c_1) \text{ next}(c_1)\)

From (C4.c2.21), by the definition of \(\rightarrow\), there exists \(f_0 \in T\text{FormulaCore}\) such that

(C4.c2.22) \(g_0 = \text{next}(f_0)\).

From (C4.c2.13'), (C4.c2.19), and (C4.c2.13'') we know

(C4.c2.23) \(\vdash \text{next}(f_1) \rightarrow (p_{f+1},s_f(p_{f+1}),s_f(p_{f+1}),c_1) \text{ done(false)}\).

From (C4.c2.21), (C4.c2.22), (C4.c2.23), by the induction hypothesis, we get

(C4.c2.24) \(\vdash g_0 \rightarrow (p_{f+1},s_f(p_{f+1}),s_f(p_{f+1}),c_1) \text{ done(false)}\).

From (C4.c2.18) and (C4.c2.20), we get

(C4.c2.25) \((t_1,g_0,c_1) \in s_0'\).

From (C4.c2.25) and (C4.c2.24), we get [C4.a.7].

Now assume

(C4.c2.26) \(p_{f+1} \leq \infty \ p_2\), which from (C4.c2.14) gives

(C4.c2.27) \((t_1,g_1,c_1) \in s_1 \cup \{(p_{f+1},F_t,(c_f.1[X \mapsto \rightarrow p_{f+1}],c_f.2[X \mapsto s_f(p_{f+1})]))\} \).

Recall:

To show [C4.a.6], we need to prove

[C4.a.7] \(\exists t \in \mathbb{N}, g \in T\text{Formula}, c \in Context: \)
\((t,g,c) \in s_0' \wedge \vdash g \rightarrow (p_{f+1},s_f(p_{f+1}),s_f(p_{f+1}),c) \text{ done(false)}\)

where

(C4.c2.17) \(s_0' = \)
if \(p_{f+1} > \infty \ p_2\) then \(s_0\)
else \(s_0 \cup \{(p_{f+1},F_t,(c_f.1[X \rightarrow p_{f+1}],c_f.2[X \rightarrow s_f(p_{f+1})]))\} \).

From (C4.c2.26) and (C4.c2.17), we have

(C4.c2.28) \(fc_0' = s_0 \cup \{(p_{f+1},F_t,(c_f.1[X \rightarrow p_{f+1}],c_f.2[X \rightarrow s_f(p_{f+1})]))\} \).

If \((t_1,g_1,c_1) \in s_1\), the proof proceeds as for the case \(p_{f+1} \leq \infty \ p_2\) above.

Consider

(C4.c2.29) \((t_1,g_1,c_1) = (p_{f+1},F_t,(c_f.1[X \rightarrow p_{f+1}],c_f.2[X \rightarrow s_f(p_{f+1})]))\).

From (C4.c2.28) and (C4.c2.29) we have

(C4.c2.30) \((t_1,g_1,c_1) \in fc_0'\)
From (C4.c2.30) and (C4.c2.13''') we get [C4.a.7].

This finishes the proof of the case Ftf=done(false).

-------------------------------
Ftf = done(true). The case p1=∞ is excluded due to (C4.5), and Def. of $\rightarrow$.

Hence, we need to prove

[C4.a.true.1] next(TA1(X,p2,Ft,fs0)) $\rightarrow$ (pf+1,sf↓(pf+1),sf(pf+1),cf) done(true),

which by Def. $\rightarrow$ means, we need to prove

[C4.a.true.2] $\neg\exists t \in N, g \in TFormula, c \in Context:
(t,g,c)\in fs00 \land \vdash g \rightarrow (pf+1,sf↓(pf+1),sf(pf+1),c) done(false)

[C4.a.true.3] fs01 = 0 \land pf+1 \geq \infty p2,

where

(C4.c2.true.1) fs00 =
if pf+1 >\infty p2 then fs0
else fs0 \cup \{(pf+1,Ft,(cf.1[X\mapsto\rightarrow pf+1],c.2[X\mapsto sf(pf+1)]))\}

(C4.c2.true.2) fs01 =
\{ (t,next(fc),c) \in TInstance |
\exists g \in TFormula:
(t,g,c)\in fs00 \land \vdash g \rightarrow (pf+1,sf↓(pf+1),sf(pf+1),c) next(fc) \}

On the other hand, from (C4.c2.9) we know

(C4.c2.true.3) next(TA1(X,p2,Ft,fs1)) $\rightarrow$ (pf+1,sf↓(pf+1),sf(pf+1),cf) done(true).

From (C4.c2.true.3), by Def. $\rightarrow$, we know

(C4.c2.true.4) $\exists t \in N, g \in TFormula, c \in Context:
(t,g,c)\in fs10 \land \vdash g \rightarrow (pf+1,sf↓(pf+1),sf(pf+1),c) done(false)

(C4.c2.true.5) fs11 = 0 \land pf+1 \geq \infty p2

where

(C4.c2.true.6) fs10 =
if pf+1 >\infty p2 then fs1
else fs1 \cup \{(pf+1,Ft,(cf.1[X\mapsto\rightarrow pf+1],c.2[X\mapsto sf(pf+1)]))\}

(C4.c2.true.7) fs11 =
\{ (t,next(fc),c) \in TInstance |
\exists g \in TFormula:
(t,g,c)\in fs10 \land \vdash g \rightarrow (pf+1,sf↓(pf+1),sf(pf+1),c) next(fc) \}

Recall the relationship between fs0 and fs1:

(C4.c2.7) fs1 = \{ (t,next(fc),c) \in TInstance |
\exists g \in TFormula:
(t,g,c)\in fs0 \land \vdash g \rightarrow (pf,sf↓pf,sf(pf),c) next(fc)\}

111
From (C4.c2.true.6), (C4.c2.true.7), and (C4.c2.true.5) we know that

(C4.c2.true.8) \( \neg \exists f_c \in T_{\text{FormulaCore}}: \)
\[ F_t \rightarrow (pf+1,sf, sf(pf+1), cf.1 [X \rightarrow pf+1]) \text{next}(f_c). \]

Now assume by contradiction that for some \((t_0,g_0,c_0)\) \(\in f_s_0\) we have

(C4.c2.true.9) \( \exists f_c \in T_{\text{FormulaCore}}: g_0 \rightarrow (pf+1,sf, sf(pf+1),c_0) \text{next}(f_c) \)

From (C4.c2.true.9), by Lemma 6, there exist \(f_c_0 \in T_{\text{FormulaCore}}\) such that

(C4.c2.true.10) \( g_0 \rightarrow (pf,sf, sf(pf),c_0) \text{next}(f_c_0) \)

From (C4.c2.true.9) by (C4.c2.7) we have that there exists \(f_c_0 \in T_{\text{FormulaCore}}\) such that

(C4.c2.true.11) \( (t_0, \text{next}(f_c_0),c_0) \in f_s_1. \)

From (C4.c2.true.11) by (C4.c2.true.6) we get

(C4.c2.true.12) \( (t_0, \text{next}(f_c_0),c_0) \in f_s_10. \)

From (C4.c2.true.12) by (C4.c2.true.7), (C4.c2.true.5), (C4.c2.true.4), we get

(C4.c2.true.13) \( \text{next}(f_c_0) \rightarrow (pf+1,sf, sf(pf+1),c_0) \text{done(true)} \)

From (C4.c2.true.10) and (C4.c2.true.13), by the induction hypothesis, we get

(C4.c2.true.14) \( g_0 \rightarrow (pf+1,sf, sf(pf+1),c_0) \text{done(true)} \)

But (C4.c2.true.14) contradicts (C4.c2.true.9). Hence, we know that for all \((t,g,c)\) \(\in f_s_0\)

(C4.c2.true.15) \( \neg \exists f_c \in T_{\text{FormulaCore}}: g \rightarrow (pf+1,sf, sf(pf+1),c) \text{next}(f_c) \)

From (C4.c2.true.8) and (C4.c2.true.15) we know that for all \((t,g,c)\) \(\in f_s_00\)

(C4.c2.true.16) \( \neg \exists f_c \in T_{\text{FormulaCore}}: g \rightarrow (pf+1,sf, sf(pf+1),c) \text{next}(f_c). \)

From (C4.c2.true.16) we get

(C4.c2.true.17) \( f_s_10 = 0 \)

From (C4.c2.true.17) and the second conjunct of (C4.c2.true.5) we get

[C4.a.true.3].

To prove [C4.a.true.2] note that from (C4.c2.true.4) and (C4.c2.true.6) we have

(C4.c2.true.18) \( F_t \rightarrow (pf+1,sf, sf(pf+1), cf.1 [X \rightarrow pf+1]) \text{done(false)} \)

does not hold.

Recall that in (C4.c2.6) we have
Hence, for no \((t,g,c)\in fs00\) we have \(g\rightarrow(pf,sf\downarrow pf, sf(pf), c)\) done(false).
It proves \([C4.a.true.2]\).

Ftf is a 'next' formula.

Let Ftf = next(TA1(X,p2,Ft,fs2)) for some fs2. Then from \([C4.a.6]\) and \([C4.c2.11]\),
we need to prove

\([C4.a.next.1]\) \(\text{next}(TA1(X,p2,Ft,fs0))\)
\(\rightarrow(pf+1,sf\downarrow(pf+1), sf(pf+1), cf)\) \(\text{next}(TA1(X,p2,Ft,fs2))\)

To prove \([C4.a.next.8]\), we define

\([C4.c2.next.1]\) \(fs00 := \)
if \(pf+1 \gg p2\) then \(fs0\)
else \(fs0 \cup \{(pf+1,Ft,(cf.1[X↦pf+1], cf.2[X↦sf(pf+1)])\}\}\)

\([C4.c2.next.2]\) \(fs01 := \)
\{ \((t, next(fc), c)\in T\text{Instance} | \)
\(\exists g\in T\text{Formula}: \)
\((t,g,c)\in fs00 \land \vdash g\rightarrow(pf+1,sf\downarrow(pf+1), sf(pf+1), c)\) \(\text{next}(fc)\} \}

and prove

\([C4.a.next.2]\) \(\exists t\in N, g\in T\text{FormulaStep}, c\in T\text{Context}: \)
\((t,g,c)\in fs00 \land \vdash g\rightarrow(pf+1,sf\downarrow(pf+1), sf(pf+1), c)\) done(false)

\([C4.a.next.3]\) \(\neg(fs01 = \emptyset \land pf+1 \geq \infty p2)\)

On the other hand, from \([C4.c2.9]\) we know

\([C4.c2.next.3]\) \(\text{next}(TA1(X,p2,Ft,fs1))\)
\(\rightarrow(pf+1,sf\downarrow(pf+1), sf(pf+1), cf)\) \(\text{next}(TA1(X,p2,Ft,fs2))\).

From \([C4.c2.next.3]\), by Def.\(→\), we know

\([C4.c2.next.4]\) \(\exists t\in N, g\in T\text{FormulaStep}, c\in T\text{Context}: \)
\((t,g,c)\in fs10 \land \vdash g\rightarrow(pf+1,sf\downarrow(pf+1), sf(pf+1), c)\) done(false)

\([C4.c2.next.5]\) \(\neg(fs11 = \emptyset \land pf+1 \geq \infty p2)\)

where

\([C4.c2.next.6]\) \(fs10 = \)
if \(pf+1 \gg p2\) then \(fs1\)
else \(fs1 \cup \{(pf+1,Ft,(cf.1[X↦pf+1], cf.2[X↦sf(pf+1)])\}\}\)

\([C4.c2.next.7]\) \(fs11 = \)
Recall the relation between \( fs_0 \) and \( fs_1 \):

\[ (C4.c2.7) \quad \text{fs}_1 = \{ (t,\text{next}(fc),c) \in T\text{Instance} \mid \\
\exists g \in T\text{Formula}:
\quad (t,g,c) \in \text{fs}_0 \land \vdash g \rightarrow ((\text{pf+1},sf \downarrow (pf+1),sf(pf+1),c) \text{next}(fc)) \} \]

By \((C4.c2.6)\) and \((C4.c2.next.1)\), to prove \([C4.a.next.2]\), it suffices to prove that

\[ [C4.a.next.4] \vdash Ft \rightarrow (pf+1,sf \downarrow (pf+1),sf(pf+1),(\text{cf.1}[X \mapsto pf+1],\text{cf.2}[X \mapsto sf(pf+1)])) \]

done(false)

But this directly follows from \((C4.c2.next.6)\) and \((C4.c2.next.4)\). Hence, \([C4.a.next.4]\) is proved.

To prove \([C4.a.next.3]\), we assume

\[ (C4.c2.next.8) \quad \text{pf+1} \geq \infty \]

and prove

\[ [C4.a.next.5] \quad \text{fs}_1 \neq \emptyset. \]

From \((C4.c2.next.8)\) and \((C4.c2.next.5)\) we know

\[ (C4.c2.next.9) \quad \text{fs}_1 \neq \emptyset. \]

From \((C4.c2.next.9)\), there exist \((t_1,g_1,c_1) \in \text{fs}_1\) and \(fc_1 \in T\text{FormulaCore}\) such that

\[ (C4.c2.next.9) \quad \vdash g_1 \rightarrow (pf+1,sf \downarrow (pf+1),sf(pf+1),c_1) \text{next}(fc_1). \]

According to \((C4.c2.next.6)\), \((t_1,g_1,c_1) \in \text{fs}_1\) means either \((t_1,g_1,c_1) \in \text{fs}_1\) or

\[ (t_1,g_1,c_1) = (pf+1,Ft,(\text{cf.1}[X \mapsto pf+1],\text{cf.2}[X \mapsto sf(pf+1)])) \]

First assume \((t_1,g_1,c_1) \in \text{fs}_1. \)

---------------

By \((C4.c2.7)\), it means that there exist \((t_0,g_0,c_0) \in \text{fs}_0\) and \(fc_0 \in T\text{FormulaCore}\) such that

\[ (C4.c2.next.10) \quad \vdash g_0 \rightarrow (pf,sf \downarrow pf,sf(pf),c_0) \text{next}(fc_0) \]

\[ (C4.c2.next.11) \quad g_1 = \text{next}(fc_0) \]

Moreover, \(g_0\) is a 'next' formula.

\[ (C4.c2.next.12) \quad g_0 = \text{next}(fc) \text{ for some } fc \in T\text{FormulaCore}. \]

Besides, from \((C4.c2.7)\) one can see that
Hence, from (C4.c2.next.9--13) we have

\[(\text{C4.c2.next.14}) \quad \text{next}(fc) \rightarrow (pf, sf \downarrow (pf), c0) \text{ next}(fc0)\]
\[(\text{C4.c2.next.15}) \quad \text{next}(fc0) \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), c0) \text{ next}(fc1)\]

From (C4.c2.next.14) and (C4.c2.next.15), by the induction hypothesis, we obtain that

\[(\text{C4.c2.next.16}) \quad \text{next}(fc) \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), c0) \text{ next}(fc1)\]

Hence, we got that for \((t0, g0, c0) \in fs0\) and \(fc1 \in T\text{FormulaCore}\)

\[(\text{C4.c2.next.17}) \quad g0 \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), c0) \text{ next}(fc1).\]

By definition (C4.c2.next.1) of \(fs00\), we have \((t0, g0, c0) \in fs00\).

Now assume \((t1, g1, c1) = (pf+1, Ft, (cf.1[X \mapsto pf+1], cf.2[X \mapsto sf(pf+1)]))\)

Trivially, by definition (C4.c2.next.1) of \(fs00\), we have \((t1, g1, c1) \in fs00\).

Hence, in both cases we found a triple

\[(\text{C4.c2.next.18}) \quad (t, g, c) \in fs00\]

such that

\[(\text{C4.c2.next.19}) \quad g \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), c) \text{ next}(fc1)\]

holds. (C4.c2.next.18), (C4.c2.next.19), and (C4.c2.next.2) imply [C4.a.next.5].

This finishes the proof of the case \(Ftf\) is a 'next' formula.

This finishes the proof of C4.c2.

This finishes the proof of C4.

This finishes the proof of Lemma 8.