Verifying the Soundness of Resource Analysis for LogicGuard Monitors
Revised Version*

Temur Kutsia Wolfgang Schreiner
RISC, Johannes Kepler University Linz
{kutsia,schreine}@risc.jku.at
September 17, 2014

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 4
4 Soundness of Resource Analysis 9
5 Conclusion 14
A Proofs 16
  A.1 Theorem 1: Soundness Theorem . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
  A.2 Proposition 1: The Invariant Statement . . . . . . . . . . . . . . . . . . . . . . . . 26
  A.3 Lemma 1: Soundness Lemma for Formulas . . . . . . . . . . . . . . . . . . . . . . . 36
  A.4 Lemma 2: Equivalence of Left- and Right-Recursive Definitions of n-Step Reductions 46
  A.5 Lemma 3: History Cut-Off Lemma . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
  A.6 Lemma 4: n-Step Reductions to done Formulas for TN, TCS, TCP . . . . . . . . . . . 70
  A.7 Lemma 5: Soundness Lemma for Universal Formulas . . . . . . . . . . . . . . . . . . 95
  A.8 Lemma 6: Monotonicity of Reduction to done . . . . . . . . . . . . . . . . . . . . . 101
  A.9 Lemma 7: Shifting Lemma . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 106
  A.10 Lemma 8: Triangular Reduction Lemma . . . . . . . . . . . . . . . . . . . . . . . . 107
  A.11 Lemma 9: Soundness of Bound Analysis . . . . . . . . . . . . . . . . . . . . . . . . 126
  A.12 Lemma 10: Invariant Lemma for Universal Formulas . . . . . . . . . . . . . . . . . 130

*The project “LogicGuard II: The Optimized Checking of Time-Quantified Logic Formulas with Applications in Computer Security” is sponsored by the FFG BRIDGE program, project No. 846003.
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 [5]; 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 [6], we tried to fill this gap by presenting a static analysis that

- 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
- 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 [6] 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 [6]. 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 proofs can be found in the Appendix.

This paper is an extended and revised version of [3] and subsumes it: We fixed typos, added Lemma 10 and the proof of Lemma 5, and in some places modified the statements and proofs of the other lemmas.

2 The Core Language and Resource Analysis

The core language is depicted in Figure 1.

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 one of them 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 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 \( F_2 \);
- \( F_1 \lor\lor F_2 \) evaluates both \( F_1 \) and \( F_2 \) “in parallel” until the value of one subformula determines the value of the total formula;
- \( \forall X \in B_1..B_2 : F \) 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 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:

\[
\begin{align*}
M := & \text{monitor } X : F \\
F := & \ @X \mid \neg F \mid F_1 \&\& F_2 \mid F_1 \lor\lor F_2 \mid \forall X \in B_1..B_2 : F \\
B := & \ 0 \mid \text{infinity} \mid X \mid B + N \mid B - N \\
N := & \ 0 \mid 1 \mid 2 \mid \ldots \\
X := & \ x \mid y \mid z \mid \ldots
\end{align*}
\]

Figure 1: The Core Language
\[ \vdash M : \mathbb{N}^\infty \times \mathbb{N}^\infty \quad \text{Environment} \vdash F : \mathbb{N}^\infty \times \mathbb{N}^\infty \quad \text{Environment} \vdash B : \mathbb{Z}^\infty \times \mathbb{Z}^\infty \]

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

\[
\frac{e \vdash \forall X : (0,0)}{e \vdash \exists X : (h, d)}
\]

\[
\frac{e \vdash F_1 : (h_1, d_1), e \vdash F_2 : (h_2, d_2)}{e \vdash F_1 \land F_2 : (\max^\infty(h_1, h_2), \max^\infty(d_1, d_2))}
\]

\[
\frac{e \vdash B_1 : (l_1, u_1), e \vdash B_2 : (l_2, u_2)}{e \vdash \forall X \in [B_1, B_2] \forall X : (h, d)}
\]

\[
\frac{e \vdash 0 : (-\infty, 0)}{e \vdash \text{infinity} : (\infty, \infty)}
\]

\[
\frac{\frac{\frac{\frac{e \vdash X : (0,0)}{[[X] \notin \text{domain}(e)]}}{e \vdash \exists X : (0,0)}}{e \vdash \forall X : \epsilon([X])}}{\frac{e \vdash B : (l, u)}{e \vdash B \land N : (l + \infty \llbracket N \rrbracket, u + \infty \llbracket N \rrbracket)}}
\]

\[
\frac{e \vdash B : (l, u)}{e \vdash B \land N : (l - \infty \llbracket N \rrbracket, u - \infty \llbracket N \rrbracket)}
\]

Figure 2: The Analysis of the Core Language

- \[ \vdash M : (h, d) \] states that the evaluation of the monitor \( M \) 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 [6] 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 \( T : \text{Monitor} \rightarrow \text{TMonitor} \) from the abstract syntax domain \text{Monitor} to a domain \text{TMonitor} denoting the runtime representation of the monitor. First, we list the domains used in the formal-
\[
\begin{align*}
\text{Environment} & := \text{Variable} \rightarrow \mathbb{Z}^\infty \times \mathbb{Z}^\infty \\
N^\infty & := N \cup \{\infty\}, \mathbb{Z}^\infty := \mathbb{Z} \cup \{-\infty, \infty\} \\
<_\infty & \subseteq N \times N^\infty \\
n_1 <_\infty n_2 & :\iff n_2 = \infty \vee n_1 < n_2 \\
\leq_\infty & \subseteq N \times N^\infty \\
n_1 \leq_\infty n_2 & :\iff n_2 = \infty \vee n_1 \leq n_2 \\
>_\infty & \subseteq N \times N^\infty \\
n_1 >_\infty n_2 & :\iff n_2 \neq \infty \wedge n_1 > n_2 \\
\geq_\infty & \subseteq N \times N^\infty \\
n_1 \geq_\infty n_2 & :\iff n_2 \neq \infty \wedge n_1 \geq n_2 \\
max^\infty : & N \times N^\infty \rightarrow N^\infty \\
max^\infty(n_1, n_2) & := \text{if } n_2 = \infty \text{ then } \infty \text{ else } \text{max}(n_1, n_2) \\
+: & N^\infty \times N^\infty \rightarrow N^\infty \\
n_1 +^\infty n_2 & := \text{if } n_1 = \infty \vee n_2 = \infty \text{ then } \infty \text{ else } n_1 + n_2 \\
-^\infty : & N^\infty \times N \rightarrow N^\infty \\
n_1 -^\infty n_2 & := \text{if } n_1 = \infty \text{ then } \infty \text{ else } \text{max}(0, n_1 - n_2) \\
-: & \mathbb{Z}^\infty \rightarrow \mathbb{Z}^\infty \\
- i & := \text{if } i = \infty \text{ then } -\infty \text{ else if } i = -\infty \text{ then } \infty \text{ else } - i \\
N : & \mathbb{Z}^\infty \rightarrow N^\infty \\
N(i) & := \text{if } i = -\infty \vee i < 0 \text{ then } 0 \text{ else } i
\end{align*}
\]

Figure 3: The Semantic Algebras of the Analysis

ization, together with their definitions (\(\mathcal{P}\) stands for the powerset and \(\text{part.} \rightarrow\) for the partial function):

\[
\begin{align*}
\text{TMonitor} & := \text{TM of Variable} \times \text{TFormula} \times \mathcal{P}(TInstance) \\
\text{TInstance} & := N \times \text{TFormula} \times \text{Context} \\
\text{Context} & := (\text{Variable} \text{part.} \rightarrow N) \times (\text{Variable} \text{part.} \rightarrow \text{Message}) \\
\text{TFormula} & := \text{done of Bool} \mid \text{next of TFormulaCore} \\
\text{TFormulaCore} & := \\
& \text{TVof Variable} \mid \\
& \text{TN of TFormula} \mid \\
& \text{TCS of TFormula} \times \text{TFormula} \mid \\
& \text{TCP of TFormula} \times \text{TFormula} \mid \\
& \text{TA of Variable} \times \text{BoundValue} \times \text{BoundValue} \times \text{TFormula} \mid \\
& \text{TA0 of Variable} \times N \times N^\infty \times \text{TFormula} \mid
\end{align*}
\]
TA1 of Variable $\times \mathbb{N}^\infty \times TFormula \times \mathbb{P}(TInstance)$

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

**Translation.** The translation is defined for monitors, formulas, and bounds. Monitors are translated into $TMonitor$’s (translated monitors), formulas are translated into $TFormula$’s (translated formulas), and bounds are translated into $BoundValue$’s:

$$T : Monitor \rightarrow TMonitor$$

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

$$T : Formula \rightarrow TFormula$$

$$T(\forall X) := next(TV(X))$$

$$T(\forall F) := next(TN(T(F)))$$

$$T(F_1 \land F_2) := next(TCS(T(F_1),T(F_2)))$$

$$T(F_1 \lor F_2) := next(TCP(T(F_1),T(F_2)))$$

$$T(forall X in B_1..B_2 : F) := next(TA(X,T(B_1),T(B_2),T(F)))$$

$$T : Bound \rightarrow BoundValue$$

$$T(0)(c) := 0$$

$$T(\infty)(c) := \infty$$

$$T(X)(c) := c.l(X) \text{ if } X \in \text{dom}(c.1)$$

$$T(X)(c) := 0 \text{ if } X \notin \text{dom}(c.1)$$

$$T(B + N)(c) := T(B)(c) + [N]$$

$$T(B + N)(c) := T(B)(c) - [N]$$

**One-Step Operational Semantics.** Apart from the quantified position variable $X$ and the translation $f = T(F)$ of the body of the 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 $(p,m)$ 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:

$$TMonitor \xrightarrow{n,Message^w,Message,Context,P[N]} TMonitor$$

$$fs_0 = fs \cup \{(p,f,[X \mapsto (p,m)])\}$$

$$rs = \{ t \in \mathbb{N} \mid \exists g \in TFormula, c \in Context : (t,g,c) \in fs_0 \land$$

$$\vdash g \rightarrow_{p,ms,m,c,done}(false)\}$$

$$fs_1 = \{(t,next(fc),c) \in TInstance \mid \exists g \in TFormula : (t,g,c) \in fs_0 \land$$

$$\vdash g \rightarrow_{p,ms,m,c,next}(fc)\}$$

$$TM(X,f,fs) \xrightarrow{p,ms,m,rs} TM(X,f,fs_1)$$

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:

$$TFormula \xrightarrow{n,Message^w,Message,Context} TFormula$$
Atomic formula:
\[ X \in \text{dom}(c.2) \]
\[ \text{next}(TV(X)) \rightarrow (p, m, t, c.2) \]
\[ X \notin \text{dom}(c.2) \]
\[ \text{next}(TV(X)) \rightarrow (p, m, t) \]

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

Sequential Conjunction:
\[ f_1 \rightarrow (p, m, c) \]
\[ \text{next}(f'_1) \]
\[ \text{next}(TCS(f_1, f_2)) \rightarrow (p, m, c) \]
\[ \text{next}(TCS(f'_1, f_2)) \]
\[ f_1 \rightarrow (p, m, c) \]
\[ \text{done}(false) \]
\[ \text{next}(TCS(f_1, f_2)) \rightarrow (p, m, c) \]
\[ f_2 \rightarrow (p, m, c) \]
\[ f'_2 \]

Parallel Conjunction:
\[ f_1 \rightarrow (p, m, c) \]
\[ \text{next}(f'_1) \]
\[ f_2 \rightarrow (p, m, c) \]
\[ \text{next}(f'_2) \]
\[ \text{next}(TCP(f_1, f_2)) \rightarrow (p, m, c) \]
\[ \text{next}(TCP(f'_1, f'_2)) \]
\[ f_1 \rightarrow (p, m, c) \]
\[ \text{done}(true) \]
\[ \text{next}(TCP(f_1, f_2)) \rightarrow (p, m, c) \]
\[ f_2 \rightarrow (p, m, c) \]
\[ \text{done}(false) \]
\[ \text{next}(TCP(f_1, f_2)) \rightarrow (p, m, c) \]
\[ f_1 \rightarrow (p, m, c) \]
\[ \text{done}(false) \]
\[ \text{next}(TCP(f_1, f_2)) \rightarrow (p, m, c) \]
\[ f_2 \rightarrow (p, m, c) \]
\[ f'_2 \]

\[ f_1 \rightarrow (p, m, c) \]
\[ \text{done}(false) \]
\[ \text{next}(TCP(f_1, f_2)) \rightarrow (p, m, c) \]
\[ f_2 \rightarrow (p, m, c) \]
\[ f'_2 \]

\[ f_1 \rightarrow (p, m, c) \]
\[ \text{done}(true) \]
\[ \text{next}(TCP(f_1, f_2)) \rightarrow (p, m, c) \]
\[ f_2 \rightarrow (p, m, c) \]
\[ f'_2 \]
Universal Quantification:

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

\[ p_1 = b_1(c) \]
\[ p_2 = b_2(c) \]
\[ p_1 \neq \infty \land p_1 \leq \infty \]
\[ \text{next}(TA(X, p_1, p_2, f)) \rightarrow_{(p, ms, m, c)} \text{TA}0' \]
\[ \text{next}(TA(X, b_1, b_2, f)) \rightarrow_{(p, ms, m, c)} \text{TA}0' \]

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

\[ p \geq p_1 \]
\[ fs = \{(p_0, f, (c.1[X \mapsto p], c.2[X \mapsto ms(p_0 + p - |ms|)]) | p_1 \leq p_0 < \infty \land \min_{p_0 < \infty} (p, p_2 + \infty, 1) \}
\[ \text{next}(TA(X, p_2, f, fs)) \rightarrow_{(p, ms, m, c)} \text{TA}1' \]

\[ \exists t \in \mathbb{N}, g \in TFormula, c \in Context : (t, g, c) \in fs0 \land g \rightarrow_{(p, ms, m, c)} \text{done}(false) \]

\[ \text{next}(TA(X, p_2, f, fs)) \rightarrow_{(p, ms, m, c)} \text{done}(false) \]

\[ \exists t \in \mathbb{N}, g \in TFormula, c \in Context : (t, g, c) \in fs0 \land g \rightarrow_{(p, ms, m, c)} \text{done}(false) \]

\[ f s_0 = \{ (p_0, f, (c.1[X \mapsto p], c.2[X \mapsto m])) | \exists t \in \mathbb{N}, g \in TFormula, c \in Context : (t, g, c) \in fs0 \land g \rightarrow_{(p, ms, m, c)} \text{done}(false) \]

\[ f s_1 = \{(\text{next}(fc), c) \in TInstance | \exists t \in \mathbb{N}, g \in TFormula, c \in Context : (t, g, c) \in fs0 \land g \rightarrow_{(p, ms, m, c)} \text{next}(fc) \}
\[ f s_1 = \emptyset \land p \geq p_2 \]

\[ \text{next}(TA(X, p_2, f, fs)) \rightarrow_{(p, ms, m, c)} \text{done}(true) \]

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.** \[ TFormula \rightarrow^{*, (n, Stream, Environment)} TFormula \], where the first \( n \) is the number of steps and the second \( n \) is the current position.

\[ Ft \rightarrow_{0, p, s, c}^{*} Ft \]

\[ n > 0 \]
\[ c = (e, \{(X, s(e(X))) | X \in \text{dom}(e)\}) \]
\[ Ft \rightarrow_{(p, s, p, s, p, c)}^{*} Ft' \]
\[ Ft' \rightarrow_{(n-1, p+1, s, c)}^{*} Ft'' \]

**With history.** \[ TFormula \rightarrow^{*, (n, Stream, Environment, Message*)} TFormula \], where the first \( n \) is the
number of steps, the second $N$ is the current position, and $Message^*$ is the history.

$$Ft \rightarrow^{*}_{(0,p,s,c,h)} Ft$$

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

\[ Ft'' \rightarrow_{(n,p,s,c,h)} Ft'' \]

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

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

$$Ft \rightarrow^{I^*}_{(0,p,s,c)} Ft$$

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

\[ Ft'' \rightarrow^{I^*}_{(n,p,s,c)} Ft'' \]

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

$$Ft \rightarrow^{I^*}_{(0,p,s,c,h)} Ft$$

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

\[ Ft'' \rightarrow^{I^*}_{(n,p,s,c,h)} Ft'' \]

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:

\[ \forall X,Y \in \text{Variable}, F \in \text{Formula}, Ft \in TFormula, It \in \mathbb{P}(\text{Instance}), n \in N, s \in \text{Stream}, \]
\[ rs \in \mathbb{P}(N), h, d \in \mathbb{N}^\infty : \]
\[ \text{let } M = \text{monitor } X : F, Mt = TM(Y, Ft, It) : \]
\[ \vdash M : (h, d) \Rightarrow \]
\[ (d \in \mathbb{N} \Rightarrow (\vdash T(M) \rightarrow^{*}_{n,s,rs} Mt \Rightarrow |It| \leq d)) \land \]
\[ (h \in \mathbb{N} \Rightarrow (\vdash T(M) \rightarrow^{*}_{n,s,rs} Mt \leftarrow \vdash T(M) \rightarrow^{*}_{n,s,rs,h} Mt)). \]

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 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.

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:
• $X$ and $Y$ are the same and $F_t$ is the translation of $F$,

• all elements in the set of instances $I_t$ contain next formulas, which have been generated at different steps in the past, but not earlier than $d$ units before from the current step,

• the formulas in the elements of $I_t$ are obtained by reductions of $T(F)$, and they themselves will reduce to a done formula in at most $d$ steps from the moment of their creation.

More formally, the invariant definition looks as follows:

**Definition 3 (Invariant).**

\[
\forall X, Y \in \text{Variable}, F \in \text{Formula}, F_t \in T\text{Formula}, I_t \in \mathcal{P}(T\text{Instance}), n \in \mathbb{N}, s \in \text{Stream}, d \in \mathbb{N}^\infty : \\
\text{invariant}(X, Y, F, F_t, I_t, n, s, d) : \iff \\
X = Y \land F_t = T(F) \land \text{alldiff}(I_t) \land \text{allnext}(I_t) \land \\
\forall t \in \mathbb{N}, F_t' \in T\text{Formula}, c \in \text{Context} : \\
(t, F_t', c) \in I_t \land d \in \mathbb{N} \Rightarrow \\
c.1 = \{(X, t)\} \land c.2 = \{(X, s(t))\} \land \\
n - d \leq t \leq n - 1 \land \\
T(F) \xrightarrow{*_{n-t,s,c_1}} F_t' \land \\
\exists b \in \text{Bool}, d' \in \mathbb{N} : \\
d' \leq d \land \vdash F_t' \xrightarrow{*_{\text{max}(0,t+d'-n),n,s,c_1}} \text{done}(b),
\]

where alldiff($I_t$) means that $t_1 \neq t_2$ for all distinct elements $(t_1, F_{t_1}, c_1), (t_2, F_{t_2}, c_2)$ of $I_t$, and allnext($I_t$) denotes the fact that for all $(t, F_t, c) \in I_t$, $F_t$ is a next formula.

Then the Invariant Statement is formulated in the following way:
Proposition 1 (Invariant Statement).

∀X ∈ Variable, F ∈ Formula, h ∈ N∞, d ∈ N∞, n ∈ N, s ∈ Stream, rs ∈ P(ℕ), Y ∈ Variable, Ft ∈ TFormula, It ∈ P(TInstance) :

\[ \vdash \text{(monitor } X : F) : (h, d) \land T(\text{monitor } X : F) \rightarrow_{\text{invariant}(X, Y, F, Ft, It, n, s, d)} \]

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.

Lemma 1 (Soundness Lemma for Formulas).

∀F, F' ∈ Formula, re ∈ RangeEnv, e ∈ Environment, Ft ∈ TFormula, n, p ∈ ℕ, s ∈ Stream, d ∈ N∞, h ∈ N :

\[ \vdash (re \vdash F : (h, d)) \land \text{dom}(e) = \text{dom}(re) \land (d \in \mathbb{N} \Rightarrow \exists b \in \text{Bool}, d' \in \mathbb{N} : d' \leq d + 1 \land T(F) \rightarrow_{\text{done}} (b)) \land (\forall h' \in \mathbb{N} : h' \geq h \Rightarrow (T(F) \rightarrow n_p, p, s, e, h \Rightarrow (F_t \Rightarrow T(F) \rightarrow n_p, p, s, e, h' \Rightarrow (F_t))). \]

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 Lemma 4 and Lemma 10 below.

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

(a) ∀n, p ∈ ℕ, s ∈ Stream, e ∈ Environment, Ft1, Ft2 ∈ TFormula :

\[ Ft_1 \rightarrow_{n_p, p, s, e}^* Ft_2 \Leftrightarrow Ft_1 \rightarrow_{n_p, p, s, e}^* Ft_2. \]

(b) ∀n, p ∈ ℕ, s ∈ Stream, e ∈ Environment, Ft1, Ft2 ∈ TFormula, h ∈ ℕ :

\[ Ft_1 \rightarrow_{n_p, p, s, e, h}^* Ft_2 \Leftrightarrow Ft_1 \rightarrow_{n_p, p, s, e, h}^* Ft_2. \]

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).

∀F ∈ Formula, Ft ∈ TFormula, p ∈ ℕ, s ∈ Stream, h ∈ ℕ, d ∈ N∞, e ∈ Environment, re ∈ RangeEnv :

\[ \vdash (re \vdash F : (h, d)) \land \text{dom}(e) = \text{dom}(re) \land (\forall Y \in \text{dom}(e) : re(Y).1 + p \leq e(Y) \leq re(Y).2 + p \Rightarrow \text{let } c := (e, \{(X, s(e(X))) \mid X \in \text{dom}(e))\} : \forall h' \in \mathbb{N} : h' \geq h \Rightarrow T(F) \rightarrow_{p, s\uparrow p, s(p), e} Ft \Leftrightarrow T(F) \rightarrow_{p, s\uparrow (\text{max}(0, p-h'), \text{min}(p, h'))\upsilon, s(p), c} Ft. \]

The Soundness Lemma for Formulas 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 \((n\text{-Step Reductions to done Formulas for TN, TCS, TCP})\).

Statement 1. TN Formulas:

\[
\forall F \in \text{Formula}, n, p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, Ft \in T\text{Formula}:
\]
\[
T(F) \rightarrow_{n, p, s, e}^* \text{done(false)} \Rightarrow \text{next}(T(N(T(F)))) \rightarrow_{n, p, s, e}^* \text{done(true)} \land
\]
\[
T(F) \rightarrow_{n, p, s, e}^* \text{done(true)} \Rightarrow \text{next}(T(N(T(F)))) \rightarrow_{n, p, s, e}^* \text{done(false)}
\]

Statement 2. TCS Formulas:

\[
\forall p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}:
\]
\[
\forall Ft_1, Ft_2 \in T\text{Formula}, n \in \mathbb{N}:
\]
\[
n > 0 \land Ft_1 \rightarrow_{n, p, s, e}^* \text{done(false)} \Rightarrow
\]
\[
\text{next}(TCS(Ft_1, Ft_2)) \rightarrow_{n, p, s, e}^* \text{done(false)} \land
\]
\[
\forall Ft_1, Ft_2 \in T\text{Formula}, n_1, n_2 \in \mathbb{N}, b \in \text{Bool}:
\]
\[
n_1 > 0 \land n_2 > 0 \land Ft_1 \rightarrow_{n_1, p, s, e}^* \text{done(true)} \land Ft_2 \rightarrow_{n_2, p, s, e}^* \text{done(b)} \Rightarrow
\]
\[
\text{next}(TCS(Ft_1, Ft_2)) \rightarrow_{\text{max}(n_1, n_2), p, s, e}^* \text{done(b)}
\]

Statement 3. TCP Formulas:

\[
\forall p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, Ft_1, Ft_2 \in T\text{Formula}, n_1, n_2 \in \mathbb{N}:
\]
\[
n_1 > 0 \land Ft_1 \rightarrow_{n_1, p, s, e}^* \text{done(false)} \land Ft_2 \rightarrow_{n_2, p, s, e}^* \text{done(true)} \Rightarrow
\]
\[
\text{next}(TCP(Ft_1, Ft_2)) \rightarrow_{n_1, p, s, e}^* \text{done(false)} \land
\]
\[
n_1 > 0 \land n_2 > 0 \land Ft_1 \rightarrow_{n_1, p, s, e}^* \text{done(false)} \land Ft_2 \rightarrow_{n_2, p, s, e}^* \text{done(true)} \Rightarrow
\]
\[
\text{next}(TCP(Ft_1, Ft_2)) \rightarrow_{\text{min}(n_1, n_2), p, s, e}^* \text{done(true)} \land
\]
\[
n_1 > 0 \land n_2 > 0 \land Ft_1 \rightarrow_{n_1, p, s, e}^* \text{done(true)} \land Ft_2 \rightarrow_{n_2, p, s, e}^* \text{done(false)} \Rightarrow
\]
\[
\text{next}(TCP(Ft_1, Ft_2)) \rightarrow_{n_2, p, s, e}^* \text{done(false)}
\]

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.

Lemma 5 \((\text{Soundness Lemma for Universal Formulas})\).

\[
\forall F \in \text{Formula}, X \in \text{Variable}, B_1, B_2 \in \text{Bound}:
\]
\[
R(F) \Rightarrow R(\text{forall } X \in B_1 \ldots B_2 : F)
\]

where

\[
R(F): \leftrightarrow
\]
\[
\forall re \in \text{RangeEnv}, \ e \in \text{Environment}, s \in \text{Stream}, d \in \mathbb{N}^\infty, h \in \mathbb{N} p \in \mathbb{N}:
\]
\[
\vdash (re \vdash F : (h, d)) \land d \in \mathbb{N} \land \text{dom}(e) = \text{dom}(re) \land
\]
\[
\forall Y \in \text{dom}(e) : re(Y).1 + p \leq e(Y) \leq re(Y).2 + p \Rightarrow
\]
\[
(\exists b \in \text{Bool}, d' \in \mathbb{N} : d' \leq d + 1 \land \vdash T(F) \rightarrow_{d', p, s, e}^* \text{done(b)})
\]

Proving of Lemma 4 requires a couple of other statements. Besides Lemma 2 above, there are two other lemmas: for monotonicity (Lemma 6) and for shifting (Lemma 7). 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 \textit{done}).

\[ \forall Ft \in TFormula, p, k \in \mathbb{N}, s \in Stream, c \in Context, b \in \text{Bool} : \]
\[ k \geq p \Rightarrow Ft \rightarrow_{p,s+i(p),c} \text{done}(b) \Rightarrow Ft \rightarrow_{k,s+i(k),c} \text{done}(b). \]

The Shifting Lemma expresses a simple fact: If a \textit{next} formula reduced to a \textit{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).

\[ \forall f \in TFormulaCore, n, p \in \mathbb{N}, s \in Stream, e \in Environment, b \in \text{Bool} : \]
\[ n > 0 \Rightarrow \text{next}(f) \rightarrow_{n+1,p,s,e} \text{done}(b) \Rightarrow \text{next}(f) \rightarrow_{n,p+1,s,e} \text{done}(b). \]

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

Lemma 8 (Triangular Reduction Lemma).

\[ \forall f_1, f_2 \in TFormulaCore, Ft \in TFormula, p \in \mathbb{N}, s \in Stream, c \in Context : \]
\[ \text{next}(f_1) \rightarrow_{p,s+i(p),c} \text{next}(f_2) \wedge \text{next}(f_2) \rightarrow_{p+1,s+(p+1),c} \text{next}(f_1) \rightarrow_{p+1,s+(p+1),c} Ft. \]

Proving Lemma 5 is more involved. It relies on three statements: the already mentioned Shifting Lemma (Lemma 7), Soundness Lemma for Bound Analysis (Lemma 9), and the Invariant Lemma for Universal Formulas (Lemma 10). The proof of Lemma 3 also use Lemma 9.

Lemma 9 (Soundness Lemma for Bound Analysis).

\[ \forall re \in \text{RangeEnv}, e \in Environment, p \in \mathbb{N}, s \in Stream, B \in \text{Bound}, l, u \in \mathbb{Z}^\infty : \]
\[ re \vdash B : (l, u) \wedge \text{dom}(e) = \text{dom}(re) \wedge \]
\[ \forall Y \in \text{dom}(e) : re(Y),1 + p \leq e(Y) \leq re(Y),2 + p \Rightarrow \]
\[ \text{let } c := (e, \{(X, s(e(X))) | X \in \text{dom}(e)) : \]
\[ l + p \leq T(B)(c) \leq u + p. \]

Finally, the Invariant Lemma for Universal Formulas has the following form:

Lemma 10 (Invariant Lemma for Universal Formulas).

\[ \forall X \in \text{Variable}, b_1, b_2 \in \text{BoundValue}, f \in TFormulaCore : \]
\[ \forall n \in \mathbb{N} : n \geq 1 \Rightarrow \text{forall}(n, X, b_1, b_2, \text{next}(f)) \]

The predicate \text{forall} in this lemma is defined below:

\[ \text{forall} \subseteq \mathbb{N} \times \text{Variable} \times \text{BoundValue} \times \text{BoundValue} \times TFormula : \]
\[ \text{forall}(n, X, b_1, b_2, f) : \]
\[ \forall p \in \mathbb{N}, s \in Stream, e \in Environment, g \in TFormula : \]
\[ \vdash \text{next}(\text{TFormula}(X, b_1, b_2, f)) \rightarrow_{n,p,s,e} g \Rightarrow \]
\[ \text{let } c := (e, \{(Y, s(e(Y))) | Y \in \text{dom}(e)) : \]
\[ p_0 = p + n, p_1 = b_1(c), p_2 = b_2(c) : \]
\[ (n = 1 \wedge (p_1 = 1 \wedge p_2 > 1) \wedge g = \text{done}(\text{true})) \lor \]
\[ (n \geq 1 \wedge p_1 \neq 1 \wedge p_2 \leq p_0 \leq p_1 \wedge g = \text{next}(\text{TFormula}(X, p_1, p_2, f))) \lor \]
\[ (n \geq 1 \wedge p_1 \neq 1 \wedge p_2 \leq p_0 \neq p_1 \wedge \]

13
$$(\exists b \in \text{Bool} : g = \text{done}(b)) \lor$$

$$(\exists gs \in \mathbb{P}(\text{TInstance}) : (gs \neq \emptyset \lor p + n \leq \infty \land p_2) \land$$

forallInstances($$X, p, p_0, p_1, p_2, f, s, e, gs$$) \land $$g = \text{next}(TA1(X, p_2, f, gs)))\),

where the predicate forallInstances is defined as follows:

forallInstances $$\subseteq$$

$$\text{Variable} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{TFormula} \times \text{Stream} \times \text{Environment} \times \mathbb{P}(\text{TInstance}) :$$

forallInstances($$X, p, p_0, p_1, p_2, f, s, e, gs$$) $$\iff$$

$$\forall t \in \mathbb{N}, g \in \mathbb{TFormula}, c_0 \in \text{Context} : (t, g, c_0) \in gs \Rightarrow$$

$$((\forall t_1 \in \mathbb{N}, g_1 \in \mathbb{TFormula}, c_1 \in \text{Context} : (t_1, g_1, c_1) \in gs \land t = t_1 \Rightarrow$$

$$(t, g, c_0) = (t_1, g_1, c_1) \land$$

$$((\exists gc \in \mathbb{TFormulaCore} : g = \text{next}(gc)) \land$$

$$c_0.1 = e[X \mapsto t] \land$$

$$c_0.2 = \{(Y, s(c_0.1(Y))) \mid Y \in \text{dom}(e) \lor Y = X\} \land$$

$$p_1 \leq t \leq \infty \min\{p_0 - 1, p_2\} \lor f \rightarrow_{p_0 - \max(p,t)}^{\max(p,t),s,c_0.1} g$$

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

∀ 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, Ftf, Itf, 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, Ftf, Itf)

Prove

[4] |Itf| ≤ df

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

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

holds. That means, we know

(6) Xf = Yf
(7) Ftf = T(Ff)
(8) alldiffs(Itf)
(9) allnext(Itf)
(10) ∀ t∈N, Ft∈TFormula, c∈Context:
     (t,Ft,c) ∈ Itf ⇒
     c.1={(Xf,t)} ∧ c.2={(Xf,sf(t))} ∧
     T(Ff) →* (n-t,t,s,c.1) Ft1 ∧
     nf-df ≤ t ≤ nf-1 ∧
     ∃b∈Bool ∃d'∈N :
     d'≤df ∧ ⊢ Ft →* (max(0,t+df'-nf),nf,sf,c.1) done(b)

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

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

Parametrization:

Q(n) ⇐⇒ ∀ X∈Variable, F∈Formula, h∈N∞, d∈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))

We want to show
∀n∈N: Q(n).

For this is suffice to show
1. Q(0)
2. ∀n∈N: Q(n) ⇒ Q(n+1)

Proof of 1
--------

Q(0)

∀ X∈Variable, F∈Formula, h∈N∞, d∈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) →* (0,s,rs) Mt ⇔ ⊢ T(M) →* (0,s,rs,h) Mt))

We take Xf, Ff, Yf, Ft, cf, Itf, df, hf, sf, rsf arbitrary but fixed.

Assume

(1) ⊢ (monitor Xf : Ff): (hf,df)
(2) \( hf \in \mathbb{N} \)

Prove

\[ [3] \vdash T(\text{monitor } Xf : Ff) \Rightarrow (0, sf, rsf) \quad \text{TM}(Yf, Ft, Itf) \iff \vdash T(\text{monitor } Xf : Ff) \Rightarrow (0, sf, rsf, hf) \quad \text{TM}(Yf, Ft, Itf) \]

Direction (\( \Rightarrow \)). Assume

\[ (4) \vdash T(\text{monitor } Xf : Ff) \Rightarrow (0, sf, rsf) \quad \text{TM}(Yf, Ft, Itf) \]

Prove

\[ [5] \vdash T(\text{monitor } Xf : Ff) \Rightarrow (0, sf, rsf, hf) \quad \text{TM}(Yf, Ft, Itf) \]

From (4), by the def. of \( \Rightarrow (0, sf, rsf) \), we get

\[ (6) T(\text{monitor } Xf : Ff) = \text{TM}(Yf, Ft, Itf). \]

and

\[ (7) rsf = \emptyset. \]

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

Direction (\( \Leftarrow \)) can be proved analogously.

Hence, \( Q(0) \) holds.

========

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

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

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

\[ (1) \quad \forall X \in \text{Variable}, F \in \text{Formula}, h \in \mathbb{N}^\infty, d \in \mathbb{N}^\infty, s \in \text{Stream}, rs \in \mathbb{P}(\mathbb{N}), Y \in \text{Variable}, Ft \in \text{TFormula}, It \in \mathbb{P}(\text{Instance}) : \]

\[ \text{let } M = \text{monitor } X : F, Mt = \text{TM}(Y, Ft, It) : \]

\[ \vdash M : (h,d) \Rightarrow \]

\[ (h \in \mathbb{N} \Rightarrow (\vdash T(M) \Rightarrow *(n, s, rs) \quad Mt \iff \vdash T(M) \Rightarrow *(n, s, rs, h) \quad Mt)) \]

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

\[ [2] \quad \forall X \in \text{Variable}, F \in \text{Formula}, h \in \mathbb{N}^\infty, d \in \mathbb{N}^\infty, s \in \text{Stream}, rs \in \mathbb{P}(\mathbb{N}), Y \in \text{Variable}, Ft \in \text{TFormula}, It \in \mathbb{P}(\text{Instance}) : \]

\[ \text{let } M = \text{monitor } X : F, Mt = \text{TM}(Y, Ft, It) : \]

\[ \vdash M : (h,d) \Rightarrow \]

\[ (h \in \mathbb{N} \Rightarrow (\vdash T(M) \Rightarrow *(n+1, s, rs) \quad Mt \iff \vdash T(M) \Rightarrow *(n+1, s, rs, h) \quad Mt)) \]
We take Xf, Ff, hf, df, sf, rsf, Yf, Ftf, Itf arbitrary but fixed.

Assume

(3) ⊢ (monitor Xf : Ff): (hf, df)
(4) hf ∈ N

and prove

[5] ⊢ T(monitor Xf : Ff) → *(n+1, sf, rsf) TM(Yf, Ftf, Itf) ⇔
    ⊢ T(monitor Xf : Ff) → *(n+1, sf, rsf, hf) TM(Yf, Ftf, Itf)

To prove (5), we need to prove

[5.1]
    ⊢ T(monitor Xf : Ff) → *(n+1, sf, rsf) TM(Yf, Ftf, Itf) ⇒
    ⊢ T(monitor Xf : Ff) → *(n+1, sf, rsf, hf) TM(Yf, Ftf, Itf).

and

[5.2]
    ⊢ T(monitor Xf : Ff) → *(n+1, sf, rsf, hf) TM(Yf, Ftf, Itf) ⇒
    ⊢ T(monitor Xf : Ff) → *(n+1, sf, rsf) TM(Yf, Ftf, Itf).

Proof of [5.1]
---------

Since T(monitor Xf : Ff) = TM(Xf, T(Ff), ∅), we assume

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

and prove

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

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

(8) Yf = Xf, Ftf = T(Ff)

From (6) by the definition of → * we know that there exist Y', Ft', It', rs1' and rs2' such that

(9) rsf = rs1' ∪ rs2'
(10) ⊢ TM(Xf, T(Ff), ∅) → *(n, sf, rs1') TM(Y', Ft', It')
(11) ⊢ TM(Y', Ft', It') → (n, sf \downarrow (n), sf(n), rs2') TM(Xf, T(Ff), Itf)

From (10), by the definition of →, (and by the invariant) we have

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

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

(13) ⊢ TM(Xf, T(Ff), ∅) → *(n, sf, rs1', hf) TM(Xf, T(Ff), Itf)

From (11) by (12) we have
(14) \( \vdash TM(Xf,T(Ff),It') \rightarrow (n,sf\downarrow(n),sf(n),rs2') \) \( TM(Xf,T(Ff),Itf) \)

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

(15) \( rs2' = \{ t \in N \mid \exists g \in TFormula, c \in Context: (t,g,c) \in It0 \land \\
\vdash g \rightarrow (n,sf\downarrow(n),sf(n),c) \text{ done(false)} \} \)

(16) \( Itf = \{ (t,g1,c) \in TInstance \mid \exists g \in TFormula: (t,g,c) \in It0 \land \\
\vdash g \rightarrow (n,sf\downarrow(n),sf(n),c) \text{ next}(g1) \} \)

where

(17) \( It0 = It' \cup \{(n,T(Ff),\{(X,n)\},\{X,sf(n)\})\} \)

To prove (7), by the definition of \( \rightarrow^* \) with h-cutoff for TMonitors, and (12), we need to prove that there exist \( Y^*,Ft^*, It^*, rs1^* \) and \( rs2^* \) such that

(18) \( rs1^* \cup rs2^* = rsf \)

(19) \( \vdash TM(Xf,T(Ff),\emptyset) \rightarrow^*(n,sf,rs1^*,hf) \) \( TM(Y^*,Ft^*,It^*) \)

(20) \( TM(Y^*,Ft^*,It^*) \rightarrow (n,sf\uparrow(max(0,n-hf),min(n,hf)),s(n),rs2^*) \) \( TM(Xf,T(Ff),Itf) \).

We can take \( rs1^*=rs1' \), \( rs2^*=rs2' \), \( Y^*=Xf \), \( Ft^*=Ftf=T(Ff) \), \( It^*=It' \). 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) \( TM(Xf,T(Ff),It') \rightarrow (n,sf\uparrow(max(0,n-hf),min(n,hf)),sf(n),rs2') \) \( TM(Xf,T(Ff),Itf) \).

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

\[ \forall t \in N \mid \exists g \in TFormula, c \in Context: (t,g,c) \in It0 \land \\
\vdash g \rightarrow (n,sf\uparrow(max(0,n-hf),min(n,hf)),sf(n),c) \text{ done(false)} \} \]

and

\[ \forall (t,g1,c) \in TInstance \mid \\
\exists g \in TFormula: (t,g,c) \in It0 \land \\
\vdash g \rightarrow (n,sf\uparrow(max(0,n-hf),min(n,hf)),sf(n),c) \text{ next}(g1) \} \]

where \( Itf0 \) is defined as in (17).

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

\[ \forall t \in N \mid \exists g \in TFormula, c \in Context: (t,g,c) \in It0 \land \\
\vdash g \rightarrow (n,sf\downarrow(n),sf(n),c) \text{ done(false)} \} \]

\[ \forall t \in N \mid \\
\exists g \in TFormula, c \in Context: (t,g,c) \in It0 \land \\
\vdash g \rightarrow (n,sf\uparrow(max(0,n-hf),min(n,hf)),sf(n),c) \text{ done(false)} \} \]

By (16) and \[23\], we need to prove
To prove [24], we need to show

$$\forall t \in N : \exists g \in TFormula, c \in Context : (t,g,c) \in I_t \land \vdash g \rightarrow (n,sf(n),sf(n),c) \text{ done(false)}$$

$$\iff \exists g \in TFormula, c \in Context : (t,g,c) \in I_t \land \vdash g \rightarrow (n,sf(n),sf(n),c) \text{ done(false)}.$$
(10) ⊢ TM(Xf, T(Ff), 0) →∗(n, sf, rs1') TM(Y', Ft', It')

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) →∗(n−t0, t0, sf, c.1) g.

From (26.8), by Lemma 2 we get

(26.9) T(Ff) →l∗(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 →l*, we get

(26.11) T(Ff) →l∗(n−t0+1, t0, sf, c.1) done(false).

From (26.11), by Lemma 2, we get

(26.12) T(Ff) →∗(n−t0+1, t0, sf, c.1) done(false).

From (3) by the definition of ⊢, there exists re0 ∈ RangeEnv such

(26.13) re0 ⊢ Ff: (hf, df) and

(26.14) re0(Xf) = (0, 0)

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

(26.15) ∀Y∈dom(c.1): re0(Y).1+t0 ≤ c.1(Y) ≤ 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) →∗(n−t0+1, t0, sf, c.1, hf) done(false).

From (26.16), by Lemma 2 we get

(26.17) T(Ff) →l∗(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 →l* with
history, there exists Ft0 ∈ TFormula such that

(26.18) T(Ff) →l∗(n−t0, t0, sf, c.1, hf) Ft0,
(26.19) \( \text{Ft0} \rightarrow (n, s \uparrow (\max(0, n-hf), \min(n, hf)), s(n), c) \text{ done(false)}. \)

From (26.18), by Lemma 2, we get

(26.20) \( \text{T(Ff)} \rightarrow^* (n-t0, t0, sf, c.1, hf) \text{ Ft0} \).

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

(26.21) \( \text{T(Ff)} \rightarrow^* (n-t0, t0, sf, c.1) \text{ Ft0} \).

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

(26.22) \( \text{Ft0}=g \).

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

Now we consider the case (26.6):

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

(26.6) \( t0=n \), \( g=\text{T(Ff)} \), \( c=\{(Xf,n),\{Xf,sf(n)\}\} \).

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

(26.23) \( \vdash \text{T(Ff)} \rightarrow (n, sf \downarrow (n), sf(n), c) \text{ done(false)} \)

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

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

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

(26.25) \( re0 \vdash Ff: (hf, df) \) and
(26.26) \( re0 = \{Xf, (0,0)\} \)

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

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

From (26.26) and (26.28) we have

(26.28) \( \text{dom}(c.1) = \text{dom}(re0) \).

From (26.25), (26.27), (4), the definition of \( c \) in (26.6), (26.28), and Lemma 3 (instantiating \( F=Ff, Ft=\text{done(false)}, p=n, s=sf, h=h'=hf, d=df, e=c.1, re=re0 \)) we get [26.24].

Proof of [26, \( \Leftarrow \)].

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

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, ⇐].

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 \( \rightarrow * \), 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

\[
\begin{align*}
(28) & \quad \vdash TM(Xf, T(Ff), \emptyset) \rightarrow^* (n+1, sf, rsf, hf) TM(Yf, Ftf, Itf) \\
\text{and want to prove} & \quad [29] \quad \vdash TM(Xf, T(Ff), \emptyset) \rightarrow^* (n+1, sf, rsf) TM(Yf, Ftf, Itf). \\
\end{align*}
\]

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

\[
\begin{align*}
(30) & \quad rs1' \cup rs2' = rsf \\
(31) & \quad \vdash TM(Xf, T(Ff), \emptyset) \rightarrow^* (n, sf, rs1', hf) TM(Yf', Ftf', Itf') \quad \text{and} \\
(32) & \quad TM(Yf', Ftf', Itf') \rightarrow (n, sf^{\uparrow}(\max(0, n-hf), \min(n, hf)), sf(n), rs2') TM(Yf, Ftf, Itf)
\end{align*}
\]

From the definitions of \( \rightarrow^* \) and \( \rightarrow \) we can see that \( Yf' = Xf, Ftf' = T(Ff) \).

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

\[
\begin{align*}
(33) & \quad rs1^* \cup rs2^* = rsf \\
(34) & \quad \vdash TM(Xf, T(F), \emptyset) \rightarrow^* (n, sf, rs1^*) TM(Yf^*, Ftf^*, Itf^*) \quad \text{and} \\
(35) & \quad TM(Yf^*, Ftf^*, Itf^*) \rightarrow (n, sf \upharpoonright n, sf(n), rs2^*) TM(Xf, T(Ff), Itf)
\end{align*}
\]

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]:

\[
\begin{align*}
(36) & \quad TM(Xf, T(Ff), Itf') \rightarrow (n, sf \upharpoonright n, sf(n), rs2') TM(Xf, T(Ff), Itf)
\end{align*}
\]

To show it, by the definition of \( \rightarrow \) for TMonitors, we need to prove
[37] \( rs2' = \{ t \in \mathbb{N} | \exists g \in TFormula, c \in \text{Context: } (t,g,c) \in It0 \land \vdash g \rightarrow (n,sf\downarrow n,sf(n),c) \text{ done(false) } \} \)

and

[38] \( Itf = \{ (t,g1,c) \in TInstance | \exists g \in TFormula: (t,g,c) \in It0 \land \vdash g \rightarrow (n,sf\downarrow n,sf(n),c) \text{ next(g1) } \} \)

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

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

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

and

(40) \( Itf = \{ (t,g1,c) \in TInstance | \exists g \in TFormula: (t,g,c) \in It0' \land \vdash g \rightarrow (n,sf\uparrow (\max(0,n-hf),\min(n,hf)),sf(n),c) \text{ next(g1) } \} \)

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

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

[41] \( \{ t \in \mathbb{N} | \exists g \in TFormula, c \in \text{Context: } (t,g,c) \in It0 \land \vdash g \rightarrow (n,sf\downarrow n,sf(n),c) \text{ done(false) } \} = \{ t \in \mathbb{N} | \exists g \in TFormula, c \in \text{Context: } (t,g,c) \in It0 \land \vdash g \rightarrow (n,sf\uparrow (\max(0,n-hf),\min(n,hf)),sf(n),c) \text{ done(false) } \} \)

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

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

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

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

$$\forall X \in \text{Variable}, F \in \text{Formula}, h \in \mathbb{N}^\infty, d \in \mathbb{N}^\infty, n \in \mathbb{N}, s \in \text{Stream}, rs \in \mathcal{P}(\mathbb{N}),$$
$$Y \in \text{Variable}, Ft \in \text{TFormula}, It \in \mathcal{P}(\text{TInstance})$$:

\[\vdash (\text{monitor } X : F) : (h, d) \wedge\]

\[\vdash T(\text{monitor } X : F) \to^*(n, s, rs) \text{TM}(Y, Ft, It) \Rightarrow \]

\[\text{invariant}(X, Y, F, Ft, It, n, s, d)\]

PROOF

-----

Parameterization

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

\[P(n) : \leftrightarrow\]

\[\forall X \in \text{Variable}, F \in \text{Formula}, h \in \mathbb{N}^\infty, d \in \mathbb{N}^\infty, n \in \mathbb{N}, s \in \text{Stream}, rs \in \mathcal{P}(\mathbb{N}),
Y \in \text{Variable}, Ft \in \text{TFormula}, c \in \text{Context}, It \in \mathcal{P}(\text{Instance}) :\]

\[\vdash (\text{monitor } X : F) : (h, d) \wedge\]

\[\vdash T(\text{monitor } X : F) \to^*(n, s, rs) \text{TM}(Y, Ft, It) \Rightarrow \]

\[\text{invariant}(X, Y, F, Ft, It, n, s, d)\]

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

----------

\[P(0)\]

\[\forall X \in \text{Variable}, F \in \text{Formula}, h \in \mathbb{N}^\infty, d \in \mathbb{N}^\infty, n \in \mathbb{N}, s \in \text{Stream}, rs \in \mathcal{P}(\mathbb{N}),
Y \in \text{Variable}, Ft \in \text{TFormula}, c \in \text{Context}, It \in \mathcal{P}(\text{Instance}) :\]

\[\vdash (\text{monitor } X : F) : (h, d) \wedge\]

\[\vdash T(\text{monitor } X : F) \to^*(0, s, rs) \text{TM}(Y, Ft, It) \Rightarrow \]

\[\text{invariant}(X, Y, F, Ft, It, 0, s, d)\]

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

Assume

(1) \(\vdash (\text{monitor } X_f : F_f) : (h_f, d_f)\)

// (2) \(d_f \in \mathbb{N}\)

(3) \(T(\text{monitor } X_f : F_f) \to^*(0, s_f, r_s) \text{TM}(Y_f, F_{tf}, I_{tf})\)

and show

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

(4) \( \text{rsf} = \emptyset \)
(5) \( \text{T(monitor } X_f : F_f) = \text{TM}(Y_f,F_t,F,I) \)

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

(6) \( Y_f = X_f \)
(7) \( F_t = \text{T}(F) \)
(8) \( I_t = \emptyset \)

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.,

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

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

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

We take \( X_f,F_f,df,hf,sf,rsf,Y_f,F_t,I_t \) arbitrary but fixed.

Assume

(2) \( \vdash (\text{monitor } X_f : F_f) : (hf,df) \)
// (3) df \( \in \mathbb{N} \)
(4) \( \text{T(monitor } X_f : F_f) \rightarrow \ast(n+1,rsf) \text{TM}(Y,F_t,I_t) \)

and show

[b] \( \text{invariant}(X_f,Y,F,t,F_t,I_t,n+1,sf,df) \)

From (4) and def. \( \rightarrow \ast \) for TMonitors, we know for some \( rs_1,rs_2 \) and \( M_t=\text{TM}(X',F_t',I') \)

(5) \( \vdash \text{T(monitor } X_f : F_f) \rightarrow \ast(n,rs,rs_1) \text{TM}(X',F_t',I') \)
(6) \( \vdash \text{TM}(X',F_t',I') \rightarrow (n,sf \downarrow n,rs(n),rs_2) \text{TM}(Y,F_t,I_t) \)
(7) rsf = rs1 ∪ rs2

From (6) by the definition of → for TMonitors, we know

(8) Xf = Yf,
(9) Ft' = Ftf, and
(10) Itf = \{(t0, next(Fc1), c0) ∈ TInstance |
\exists Ft0 ∈ Tformula such that (t0, Ft0, c0) ∈ It0 and
\vdash Ft0 → (n, sf repayment(n), c0) next(Fc1)\}

where

(11) It0 = 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
\vdash T(\text{monitor } Xf : Ff) →* (n, sf, rs1) TM(Yf, Ftf, It') \implies
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 ∈ \mathbb{N}, Ft ∈ TFormula, c ∈ Context:
(t, Ft, c) ∈ It' \land d ∈ \mathbb{N} ⇒
c.1 = \{(Xf, t)\} \land c.2 = \{(Xf, sf(t))\} \land
n - df ≤ t ≤ n - 1 \land
T(Ff) →* (n - t, t, sf, c.1) Ft \land
\exists b ∈ \text{Bool} \exists d' ∈ \mathbb{N} :
d' ≤ df \land \vdash Ft →* (max(0, t + d' - n), n, sf, c.1) done(b)

Showing [b] means that we want to show

[b1] Xf = Yf
[b2] Ftf = T(Ff)
[b3] alldiff(Itf)
[b4] allnext(Itf)
[b5] \forall t ∈ \mathbb{N}, Ft ∈ TFormula, c ∈ Context:
(t, Ft, c) ∈ Itf \land d ∈ \mathbb{N} ⇒
c.1 = \{(Xf, t)\} \land c.2 = \{(Xf, sf(t))\} \land
n + 1 - df ≤ t ≤ n \land
T(Ff) →* (n + 1 - t, t, sf, c.1) Ft \land
\exists b ∈ \text{Bool} \exists d' ∈ \mathbb{N} :
d' ≤ df \land \vdash Ft →* (max(0, t + d' - n - 1), n + 1, sf, c.1) done(b)
Proof of \([b1]\)
------------
\([b1]\) is proved by (14).

Proof of \([b2]\)
------------
\([b2]\) is proved by (15).

Proof of \([b3]\)
------------
From (10) one can see that the elements \((t,F_t,c)\) in \(I_t\) inherit their tag \(t\) from \(I_t^0\), which is \(I_t' \cup \{(n,F_t,(c_p,c_m))\}\). 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 \(\{(n,F_t,c_f)\}\) to \(I_t'\), will guarantee all instances in \(I_t^0\) have different tags. Since these tags are transferred to \(I_t\), we conclude that \([b3]\) holds.

Proof of \([b4]\)
------------
\((b4)\) follows directly from (10), since every element in \(I_t\) has a form \((t,\text{next}(F_c),c)\).

Proof of \([b5]\)
------------
Recall that we have to prove
\[\forall t \in \mathbb{N}, F_t \in \text{TFormula}, c \in \text{Context}:
\( (t,F_t,c) \in I_t \land d \in \mathbb{N} \Rightarrow
\]
\[c.1 = \{(X_f,t)\} \land c.2 = \{(X_f,sf(t))\} \land
\]
\[n+1-df \leq t \leq n \land
\]
\[T(F_f) \rightarrow^* (n+1-t,t,sf,c.1) F_t \land
\]
\[\exists b \in \text{Bool} \exists d' \in \mathbb{N} :
\]
\[d' \leq df \land \vdash F_t \rightarrow^* (\max(0,t+b'-n-1),n+1,sf,c.1) \text{ done(b)}
\]

We take \(t_b, F_{t_b}, c_b\) arbitrary but fixed, assume
\[\text{(19)} \ (t_b,F_{t_b},c_b) \in I_t \land d \in \mathbb{N}
\]
and prove
\[\text{[b5.1]} \ c_b.1 = \{(X_f,t_b)\} \land c_b.2 = \{(X_f,sf(tb))\}
\]
\[\text{[b5.2]} \ n+1-df \leq t_b \leq n
\]
\[\text{[b5.3]} \ T(F_f) \rightarrow^* (n+1-t_b,t_b,sf,c.1) F_{t_b} \land
\]
\[\text{[b5.4]} \ \exists b \in \text{Bool} \ \exists d' \in \mathbb{N} :
\]
\[d' \leq df \land \vdash F_{t_b} \rightarrow^* (\max(0,t_b+d'-n-1),n+1,sf,c.1) \text{ done(b)}
\]

From (19) and (b4) we know that there exists \(F_{cb} \in \text{TFormulaCore}\) such that
\[\text{(20)} \ F_{t_b} = \text{next}(F_{cb})
\]
From (19), (20) and (10) of we know there exists \(F_{t_0} \in \text{TFormula}\) such that

29
Proof of [b5.1]
-----------------
We want to prove
[b5.1] cb.1=<(Xf,tb)> ∧ cb.2=<(Xf,sf(tb))>

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

(C1) (tb,Ft0,cb) = (n,Ftf,({(X',n)},{(X',sf(n))})) and
(C2) (tb,Ft0,cb) ∈ It'.

In case (C1) we have tb=n, Ft0 = 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 ≤ tb ≤ n.

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

(C1) (tb,Ft0,cb) = (n,Ftf,({(X',n)},{(X',sf(n))})) and
(C2) (tb,Ft0,cb) ∈ It'.

The case (C1)
-------------
In case (C1) we have tb=n, Ft0 = 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 re0∈RangeEnv such that re0(Xf) = (0,0) and

(24) re0 ⊢ Ff:(hf,0)

Now we apply Statement 1 of Lemma 1 with F=Ff, re=re0, e=<(Xf,n)>, s=sf, d=df=0, h=hf, s=sf, p=n, and since T(Ff)=Ftf by (17), we obtain
(25) \( \exists b \in \text{Bool} \ \exists d' \in \mathbb{N} : d' \leq 1 \land Tf \rightarrow *(d',n,sf,\{(Xf,n)\}) \) \( \text{done}(b) \)

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

(26) \( d' \leq 1 \) and
(27) \( Tf \rightarrow *(d',n,sf,\{(Xf,n)\}) \) \( \text{done}(b) \).

Note that since \( Tf = T(Ff) \), by the definition of the translation \( T \), \( Tf \) is a 'next' formula. Hence, \( d' \neq 0 \), because otherwise by (27) and the definition of \( \rightarrow * \) we would get \( Ff=\text{done}(b) \), which would contradict the fact that \( Tf \) is a 'next' formula. Therefore, from (26) we get

(28) \( d' = 1 \).

From (27) and (28) we get

(29) \( Tf \rightarrow *(1,n,sf,\{(Xf,n)\}) \) \( \text{done}(b) \).

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

(30) \( Tf \rightarrow (n,sf|n,sf(n),\{(Xf,n)\},\{(Xf,sf(n))\})) \) \( Ft' \)
(31) \( Ft' \rightarrow *(0,n+1,sf,\{(Xf,n)\}) \) \( \text{done}(b) \).

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

(32) \( Tf \rightarrow (n,sf|n,sf(n),\{(Xf,n)\},\{(Xf,sf(n))\})) \) \( \text{next}(Fcb) \)

From (30) and (32) and by the fact that the reduction \( \rightarrow \) is deterministic (one can not perform two different reductions from \( Tf \) with the same \( n,sf|n,sf(n),\) and \( \{(Xf,n)\},\{(Xf,sf(n))\}) \): 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) \( \text{next}(Fcb) \rightarrow *(0,n+1,sf,\{(Xf,n)\},\{(Xf,sf(n))\}) \) \( \text{done}(b) \).

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).

\-----------------------------
From \( (tb,Ft0,cb) \in I' \), by (18), we get

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

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

(36) \( n+1-df \leq tb \).

31
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

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

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

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

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

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

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

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

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 \( T(Ff) \rightarrow^{*} (n+1-tb, tb, sf, cb.1) \ Ftb \), which, by Lemma 2, is equivalent to proving

\[
(41) \ T(Ff) \rightarrow l^{*} (n+1-tb, tb, sf, cb.1) \ Ftb
\]

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

\[
[42] \ T(Ff) \rightarrow l^{*} (n-tb, tb, sf, cb.1) \ Ft' \ \text{and} \ [43] \ Ft' \rightarrow (n, sf\downarrow(n), s(n), c') \ Ftb
\]

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

\[
(44) \ c'=cb.
\]

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

From (21), we know that \( (tb, Ft0, cb) \in I0 \). By (11) and (14), we have

\[
(45) \ I0 = I' \cup \{(n, Ftf, \{(Xf, n), \{(Xf, sf(n))\})\})
\]

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

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

From (46), by Lemma 2, we get (42).

Now assume \( (tb, Ft0, cb) \in \{(n, Ftf, \{(Xf, n), \{(Xf, sf(n))\})\}) \). That means, taking
tb=n, Ft0=Ftf, and cb=((Xf,n),{(Xf, sf(n))}). Then, from (42), we need to prove

\[ T(Ff) \rightarrow l* (0,n,sf,\{(Xf,n)\}) \] \[ Ftf. \]

This follows from the definition of \( \rightarrow l* \) and [b2].

Hence, [b5.3] is proved.

Proof of [b5.4]
-----------------
Recall that we took tb, Ft0, cb arbitrary but fixed and assumed

(21) (tb,Ft0,cb) \( \in \) Itf.

We are looking for b* \( \in \) Bool and d'* \( \in \) N such that

\[ d'* \leq df \] and

\[ \vdash Ft0 \rightarrow *(\max(0,d'*,n-1),n+1,sf,cb.1) \] done(b*)

hold.

From (21) and (b4) we know that there exists Fcb \( \in \) TFormulaCore such that

(50) Ft0=next(Fcb)

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

(C1) (tb,Ft0,cb) = (n,Ftf,((X',n),{(X', sf(n))}))

(C2) (tb,Ft0,cb) \( \in \) It'

Case (C1):
----------
From (C1) we know

(51) tb = n

(52) Ft0 = Ftf

(53) cb = ((Xf,n),{(Xf, sf(n))})

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

[b5.3.a] \( \exists b \in \) Bool, d' \( \in \) N:

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

From (53), we know

(54) cb.1 = {(Xf,n)}

(55) cb.2 = {(Xf, sf(n))}

From (2) and the definition of \( \vdash \) we have some re \( \in \) RangeEnv such that

(56) re(Xf) = (0,0)

(57) re \( \vdash \) Ff: (hf,df)
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

\[
(58) \quad d_1' \leq df+1
\]

\[
(59) \quad \vdash Ftf \rightarrow \star(d_1',n,sf,\{(X_f,n)\}) \text{ done}(b_1)
\]

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

\[
(60) \quad d_1' > 0
\]

\[
(61) \quad \vdash Ftf \rightarrow (n,sf \downarrow n, sf(n), \{(X_f,n),\{(X_f,sf(n))\})) \text{ Ftb'}
\]

\[
(62) \quad \vdash Ftb' \rightarrow \ast(d_1'-1,n+1,sf,\{(X_f,n)\}) \text{ done}(b_1)
\]

From (22, 52, 53), we know

\[
(63) \quad \vdash Ftf \rightarrow (n,sf \downarrow n, sf(n), \{(X_f,n),\{(X_f,sf(n))\})) \text{ Ftb}
\]

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

\[
(64) \quad Ftb' = Ftb
\]

From (62, 64), we know

\[
(65) \quad \vdash Ftb \rightarrow \ast(d_1'-1,n+1,sf,\{(X_f,n)\}) \text{ done}(b_1)
\]

From (60), we know

\[
(66) \quad d_1'-1 = \max(0,d_1'-1)
\]

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 \( (tb,Ft0,cb) \in \text{It'} \).

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

\[
(67) \quad d_i' \leq df
\]

\[
(68) \quad \vdash Ft0 \rightarrow \ast(\max(0,\text{tb}+d_i'-n),n,sf,\text{cb}.1) \text{ done}(b_i)
\]

This implies that

\[
(69) \quad \text{tb}+d_i'-n > 0,
\]

otherwise we would have \( Ft0 = \text{done}(b_i) \), which contradicts the assumption \( (tb,Ft0,cb) \in \text{It'} \) and (20). Hence, we have

\[
(70) \quad \vdash Ft0 \rightarrow \ast(\text{tb}+d_i'-n,n,sf,\text{cb}.1) \text{ done}(b_i)
\]

Therefore, we can apply the definition \( \rightarrow \ast \) for TFormulas to (70) and (22), concluding \( \vdash \text{next(Fcb)} \rightarrow \ast(\text{tb}+d_i'-n-1,n+1,sf,\text{cb}.1) \text{ done}(b_i) \) and, hence

\[
(71) \quad \vdash Ft0 \rightarrow \ast(\text{tb}+d_i'-n-1,n+1,sf,\text{cb}.1) \text{ done}(b_i)
\]
Now we can take $d''=d'$ and $b''=b_i$. From (59) we get

\[(72) \quad tb+di''-n-1 = \max(0, tb+di''-n-1).\]

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, Ft ∈ TFormula, n ∈ N, p ∈ N, s ∈ Stream, d ∈ N∞, h ∈ N:

(↑ (re ⊨ F: (h, d)) ∧ dom(e) = dom(re) ∧
∀Y ∈ dom(e): re(Y).1 + i p ≤ e(Y) ≤ re(Y).2 + i p ∧

( h ∈ N ) ⇒
∃b ∈ Bool, ∃d’ ∈ N:

d’ ≤ d + 1 ∧ ⊢ T(F) →*(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)) ∧ dom(e) = dom(re) ∧
∀Y ∈ dom(e): re(Y).1 + i p ≤ e(Y) ≤ re(Y).2 + i p ∧

d ∈ N ) ⇒
∃b ∈ Bool, ∃d’ ∈ N:

d’ ≤ d + 1 ∧ ⊢ T(F) →*(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)) ∧ dom(e) = dom(re) ∧
∀Y ∈ dom(e): re(Y).1 + i p ≤ e(Y) ≤ re(Y).2 + i 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:
We want to prove
\[ \forall F \in \text{Formula}: R(F) \]

By structural induction over F:

\[ C1: F=\emptyset X. \text{ Then } T(F) = \text{next}(TV(X)) \]

---------

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

\[ (1.1) \vdash (\text{ref} \vdash \emptyset X: (hf,df)) \]
\[ (1.2) \text{df} \in \mathbb{N}, \]
\[ (1.3) \text{dom(ef)} = \text{dom(ref)} \land \forall Y \in \text{dom(ef)}: \text{ref}(Y).1 + i pf \leq i \text{ef}(Y) \leq i \text{ref}(Y).2 + i pf \]

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

\[ [1.4] d^*' \leq df+1 \text{ and } [1.5] \vdash \text{next}(TV(X)) \rightarrow *(d^*,pf,sf,ef) \text{ done}(b^*) \]

From (1.1) we get

\[ (1.6) hf=0 \text{ and } (1.7) df=0. \]

We define

\[ (1.8) c = (ef,\{(X,sf(ef(X))) \mid X \in \text{dom(ef)}\}) \]

and take

\[ (1.9) d^*'=1 \]

and

\[ (1.10) b^* = \]
\[ \text{ if } X \in \text{dom(c.2)} \text{ then } \]
\[ \text{c.2(X)} \]
\[ \text{ else } \]
\[ \text{false} \]

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]:

\[ [1.11] \vdash \text{next}(TV(X)) \rightarrow *(1,pf,sf,ef) \text{ done}(b^*). \]
where \( b^* \) is defined in (1.10). By the definition of \( \rightarrow^* \), to prove [1.11], we need to find \( F_t' \in T_{\text{Formula}} \) such that

\[
[1.12] \text{next}(TV(X)) \rightarrow (pf, sf, pf, sf(pf), c) F_t' \quad \text{and} \quad
[1.13] F_t' \rightarrow^* (0, pf+1, sf, ef) \text{done}(b^*)
\]

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 \( \text{next}(TV(X)) \), and [1.13] holds by the definition of \( \rightarrow^* \).

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

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

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

\begin{align*}
(2.1) &\vdash (\text{ref} \vdash \neg F_1: (hf, df)) \\
(2.2) &\text{df} \in \mathbb{N}, \\
(2.3) &\text{dom(ef)} = \text{dom(ref)} \land \forall Y \in \text{dom(ef)}: \text{ref}(Y).1 + i pf \leq i \text{ef}(Y) \leq i \text{ref}(Y).2 + i pf
\end{align*}

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

\begin{align*}
[2.4] &\text{d}^* \leq \text{df} + 1 \quad \text{and} \\
[2.5] &\vdash \text{next}(TN(T(F_1))) \rightarrow^* (\text{d}^*, pf, sf, ef) \text{done}(b^*)
\end{align*}

hold.

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

\begin{align*}
(2.6) &\vdash (\text{re} \vdash F_1): (hf, df).
\end{align*}

From (2.6), (2.3) and the induction hypothesis there exist \( b_i \in \mathbb{B} \) and \( \text{d}^i \in \mathbb{N} \) such that

\begin{align*}
(2.7) &\text{d}^i \leq \text{df} + 1 \quad \text{and} \\
(2.8) &\vdash T(F_1) \rightarrow^* (\text{d}^i, pf, sf, ef) \text{done}(b_i).
\end{align*}

We take

\begin{align*}
(2.9) &\text{d}^* = \text{d}^i \\
\end{align*}

and define

\begin{align*}
(2.10) & b^* := \\
&\quad \text{if } b_i = \text{true} \quad \text{then} \\
&\quad \text{false} \\
&\quad \text{else} \\
&\quad \text{true}
\end{align*}

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].

38
C3. \( F = F1 \& F2 \). Then \( T(F) = \text{next}(TCS(T(F1),T(F2))) \).

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

\[ (3.1) \vdash (ref \vdash F1 \& F2 : (hf,df)), \]
\[ (3.2) \text{df} \in \mathbb{N}, \]
\[ (3.3) \text{dom(ef)} = \text{dom(ref)} \land \forall Y \in \text{dom(ef)}: \text{ref}(Y).1 +i pf \leq i \text{ef}(Y) \leq i \text{ref}(Y).2 +i pf \]

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

\[ [3.4] \text{d*’} \leq \text{df}+1 \text{ and} \]
\[ [3.5] \vdash \text{next}(TCS(T(F1),T(F2))) \rightarrow *(d*’,pf,sf,ef) \text{ done}(b*) \]

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

\[ (3.6) \vdash (ref \vdash F1 : (h1,d1)) \]
\[ (3.7) \vdash (ref \vdash F2 : (h2,d2)) \]

such that \( h1,d1,h2,d2 \in \mathbb{N} \) and

\[ (3.8) \text{df} = \max\infty(d1,d2) = \max(d1,d2) \]

From (3.6), (3.3), and the induction hypothesis there exist \( b1i \in \text{Bool} \) and \( d1i’ \in \mathbb{N} \) such that

\[ (3.9) d1i’ \leq d1+1 \text{ and} \]
\[ (3.10) \vdash T(F1) \rightarrow *(d1i’,pf,sf,ef) \text{ done}(b1i). \]

From (3.7) and the induction hypothesis there exist \( b2i \in \text{Bool} \) and \( d2i’ \in \mathbb{N} \) such

\[ (3.11) d2i’ \leq d2+1 \text{ and} \]
\[ (3.12) \vdash T(F2) \rightarrow *(d2i’,pf,sf,ef) \text{ done}(b2i). \]

From (3.10) and (3.12) we have

\[ (3.13) d1i’ \geq 0 \text{ and} \]
\[ (3.14) d2i’ \geq 0 \]

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

We proceed by case distinction over \( b1i \).

\( b1i = \text{false} \)

We take

\[ (3.15) b* = b1i = \text{false} \text{ and} \]
\[ (3.16) d*’ = d1i’. \]

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

\[ \text{b}^* = b_{2i} \quad \text{and} \quad d^* = \max(d_{1i}', d_{2i}). \]

From (3.18, 3.9, 3.11) we get

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

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. \( F = F_1 \cup F_2 \). Then \( T(F) = \text{next}(TCP(T(F_1), T(F_2))). \)

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

\[ (4.1) \vdash (re \vdash F_1 \land F_2: (h_1, d_1)), \]
\[ (4.2) df \in \mathbb{N}, \]
\[ (4.3) \text{dom}(ef) = \text{dom}(ref) \land \forall Y \in \text{dom}(ef): ref(Y).1 + i pf \leq i ef(Y) \leq i ref(Y).2 + i pf \]

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

\[ (4.4) d^* \leq df + 1 \quad \text{and} \quad (4.5) \vdash \text{next}(TCP(T(F_1), T(F_2))) \rightarrow * (d^*, pf, sf, ef) \text{ done}(b^*) \]

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

\[ (4.6) \vdash (re \vdash F_1: (h_1, d_1)) \]
\[ (4.7) \vdash (re \vdash F_2: (h_2, d_2)) \]

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

\[ (4.8) df = \max \infty (d_1, d_2) = \max (d_1, d_2) \]

From (4.6), (4.3), and the induction hypothesis there exist \( b_{1i} \in \text{Bool} \) and \( d_{1i}' \in \mathbb{N} \) such that

\[ (4.9) d_{1i}' \leq d_1 + 1 \quad \text{and} \quad (4.10) \vdash T(F_1) \rightarrow * (d_{1i}', pf, sf, ef) \text{ done}(b_{1i}). \]

From (4.7), (4.3) and the induction hypothesis there exist \( b_{2i} \in \text{Bool} \) and \( d_{2i}' \in \mathbb{N} \) such that

\[ (4.11) d_{2i}' \leq d_2 + 1 \quad \text{and} \quad (4.12) \vdash T(F_2) \rightarrow * (d_{2i}', pf, sf, ef) \text{ done}(b_{2i}). \]

From (4.10) and (4.12) we have
(4.13) \(d_{1i'} > 0\) and 
(4.14) \(d_{2i'} > 0\)

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

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

\(b_{1i} = \text{false}, \ b_{2i} = \text{true}\)
-----------------------

We take

(4.15) \(b^* = \text{false},\)
(4.16) \(d^* = d_{1i'}\).

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

\(b_{1i} = \text{false}, b_{2i} = \text{false}\)
-----------------------

We take

(4.17) \(b^* = \text{false},\)
(4.18) \(d^* = \min(d_{1i'}, d_{2i'})\).

From (4.9, 4.11, 4.18) we get

(4.19) \(d^* = \min(d_{1i'}, d_{2i'}) \leq \min(d_{1} + 1, d_{2} + 1) = \min(d_{1}, d_{2}) + 1 \leq \max(d_{1}, d_{2}) + 1 = df + 1\).

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].

\(b_{1i} = \text{true}, b_{2i} = \text{true}\)
-----------------------

We take

(4.20) \(b^* = b_{2i'}\) and 
(4.21) \(d^* = \max(d_{1i'}, d_{2i'})\).

From (4.20, 4.9, 4.11) we get

(4.22) \(d^* = \max(d_{1i'}, d_{2i'}) \leq \max(d_{1} + 1, d_{2} + 1) = \max(d_{1}, d_{2}) + 1 = df + 1\)

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].

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

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

From (4.18, 4.9, 4.11) we get

(4.25) \( d^*'=d_{2i'} \leq d_{2i'}+1 \leq \max(d_{1i}+1, d_{2i}+1) = \max(d_{1i}, d_{2i})+1 = d_{f+1} \)

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

This case follows from the induction hypothesis for \( F_1 \) and Lemma 5.

It finishes the proof of Statement 1 of Lemma 1.

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

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

Proof
-----

Parametrization:

\( S(n) : \vdash \)
\( \forall F \in \text{Formula}, \ re \in \text{RangeEnv}, \ e \in \text{Environment}, \ Ft \in \text{TFormula}, \ p \in \mathbb{N}, \ s \in \text{Stream}, \ d \in \mathbb{N}_\infty, \ h \in \mathbb{N}, \ h' \in \mathbb{N}: \)
\( \vdash (re \vdash F: (h,d)) \land \forall Y \in \text{dom}(e): \ re(Y).1 + \lfloor i \lfloor p \leq e(Y) \leq i \lfloor re(Y).2 + i \lfloor p \land h'\geq h \Rightarrow \)
\( \text{ (T(F)} \rightarrow^* (n,p,s,e) \ Ft \iff \)
\( \text{ (T(F)} \rightarrow^* (n,p,s,e,h') \ 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 \( Ff \in \text{Formula}, \ re \in \text{RangeEnv}, \ ef \in \text{Environment}, \ Ftf \in \text{TFormulas}, \ pf \in \mathbb{N}, \)
sf ∈ Stream, df ∈ N∞, hf ∈ N, hf' ∈ N arbitrary but fixed, assume

(a.1) ⊢ (ref ⊢ Ff: (hf, df))
(a.2) ∀Y ∈ dom(ef): ref(Y).1 + i pf ≤ i ef(Y) ≤ i ref(Y).2 + i pf
(a.3) hf' ≥ hf

and prove

(a.4) T(Ff) →* (0, pf, sf, ef) Ftf ⇔
     T(Ff) →* (0, pf, sf, ef, hf') Ftf

(⇒)
Assume

(a.5) T(Ff) →* (0, pf, sf, ef) Ftf

and prove

(a.6) T(Ff) →* (0, pf, sf, ef, hf') Ftf.

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

(⇐). Analogous.

Proof of (b)
------------
We assume

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

and prove

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

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

(b.3) ⊢ (ref ⊢ Ff: (hf, df))
(b.4) ∀Y ∈ dom(ef): ref(Y).1 + i pf ≤ i ef(Y) ≤ i ref(Y).2 + i pf
(b.5) hf' ≥ hf

and prove
(b.6) \( T(Ff) \rightarrow^* (n+1, pf, sf, ef) \) \( Ftf \leftrightarrow \)
\( T(Ff) \rightarrow^* (n+1, pf, sf, ef, hf') \) \( Ftf \)

\( \implies \) 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 \( Ft' \in T\text{Formula} \)

(b.9) \( T(Ff) \rightarrow (pf, sf, pf, sf(pf), c) \) \( Ft' \)
\( (b.10) \) \( Ft' \rightarrow^* (n, pf+1, sf, ef) \) \( Ftf \)

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

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

(b.12) \( T(Ff) \rightarrow (pf, sf↑(\max(0, pf-hf'), \min(pf, hf')), sf(pf), c) \) \( Ft' \).

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].

\( \impliedby \) 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 \( Ft' \in T\text{Formula} \)

(b.17) \( T(Ff) \rightarrow (pf, sf↑(\max(0, pf-hf'), \min(pf, hf')), sf(pf), c) \) \( Ft' \)
\( (b.18) \) \( Ft' \rightarrow^* (n, pf+1, sf, ef, hf') \) \( Ftf \),

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

Then from (b.3), (b.19), (b.4), (b.5), (b.17) and Lemma 3 we get

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

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

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

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

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

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 \text{TFormula}, 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) \land 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) \( F_{t1} \rightarrow^* (n, p, s, e) F_{t2} \iff F_{t1} \rightarrow^{l*} (n, p, s, e) F_{t2}. \)

(\( \implies \)):  
-----
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 $F_t' \in T_{\text{Formula}}$ such that

(5) $F_t' \rightarrow (pf, sf \downarrow pf, sf(pf), c) F_t'$ and
(6) $F_t' \rightarrow * (n-1, pf+1, sf, ef) F_{t'2}$

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

From (6), by the induction hypothesis we get

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

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) $F_t' = F_{t'2}$.

From (8) and (5) we get

(9) $F_t' \rightarrow (pf, sf \downarrow pf, sf(pf), c) F_{t'2}$.

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

(10) $F_t' \rightarrow l*(0, pf, sf, ef) F_{t'1}$.

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

(11) $F_t' \rightarrow l*(1, pf, sf, ef) F_{t'2}$.

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


Case (ii)

From (7), by the definition of $\rightarrow l*$, there exists $F'_t' \in T_{\text{Formula}}$ such that

(12) $F_t' \rightarrow l*(n-2, pf+1, sf, ef) F'_t'$
(13) $F'_t' \rightarrow (pf+1, sf \downarrow pf+1, sf(pf+1), c) F_{t'2}$,

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

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

(14) $F_t' \rightarrow * (n-2, pf+1, sf, ef) F'_t'$.

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

(15) $F_t' \rightarrow * (n-1, pf, sf, ef) F'_t'$. 

47
From (15), by the induction hypothesis, we get

(16) \( F_{t_1} \rightarrow l^{*(n-1,pf, sf, ef)} F_{t''} \).

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

(4) \( F_{t_1} \rightarrow l^{*(n,pf, sf, ef)} F_{t_2} \).

(\( \iff \))

We assume

(17) \( F_{t_1} \rightarrow l^{*} (n,pf, sf, ef) F_{t_2} \)

and prove

(18) \( F_{t_1} \rightarrow ^{*} (n,pf, sf, ef) F_{t_2} \).

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

(19) \( F_{t_1} \rightarrow l^{*(n-1,pf, sf, ef)} F_{t'} \) and

(20) \( F_{t'} \rightarrow (pf+n-1, sf|_1(pf+n-1), sf(pf+n-1), c) F_{t_2} \),

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

From (19), by the induction hypothesis we get

(21) \( F_{t_1} \rightarrow ^{*} (n-1,pf, sf, ef) F_{t'} \)

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 \( F_{t'} = F_{t_1} \), which together with (20) and the fact \( n-1=0 \) implies

(22) \( F_{t_1} \rightarrow (pf, sf|_1 pf, sf(pf), c) F_{t_2} \).

On the other hand, by the definition of \( \rightarrow ^{*} \) we have

(23) \( F_{t_2} \rightarrow ^{*} (0,pf+1, sf, ef) F_{t_2} \).

From (22) and (23), by the definition of \( \rightarrow ^{*} \), w get

(24) \( F_{t_2} \rightarrow ^{*} (1,pf, sf, ef) F_{t_2} \).

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

Case (ii)

--------
From (21), by the definition of $\rightarrow^*$, there exists $Ft'' \in T\text{Formula}$ such that

(25) $Ftf_1 \rightarrow (pf, sf, pf, sf(c), c) Ft''$
(26) $Ft'' \rightarrow^*(n-2, pf+1, sf, ef) Ft'$

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

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

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

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

(28) $Ft'' \rightarrow l^*(n-1, pf+1, sf, ef) Ftf_2$.

From (28), by the induction hypothesis we get

(29) $Ft'' \rightarrow^*(n-1, pf+1, sf, ef) Ftf_2$.

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

$[18] Ftf_1 \rightarrow^*(n, pf, sf, ef) Ftf_2$.

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

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

$Q(n, Ft_1, Ft_2, p, s, e, h) : \iff$

$Ft_1 \rightarrow^* (n, p, s, e, h) Ft_2 \iff Ft_1 \rightarrow l^* (n, p, s, e, h) Ft_2$

We want to prove

(G) $\forall Ft_1, Ft_2 \in T\text{Formula}, p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, h \in \mathbb{N}, \forall n \in \mathbb{N} :$ $S(n, Ft_1, Ft_2, p, s, e, h)$.

We take $Ft_1, Ft_2, pf, sf, ef$, and $hf$ arbitrary but fixed.

We have to prove

(G1) $\forall k, n \in \mathbb{N} :$ $S(k, Ft_1, Ft_2, pf, sf, ef, hf) \land n > k \Rightarrow S(n, Ft_1, Ft_2, pf, sf, ef, hf)$.

Proof of (G1)
--------------

We take $n$ arbitrary but fixed, assume $n > k$ and

(1) $\forall k < n :$ $Ft_1 \rightarrow^* (k, pf, sf, ef, hf) Ft_2 \iff Ft_1 \rightarrow l^* (k, pf, sf, ef, hf) Ft_2$

and prove

(2) $Ft_1 \rightarrow^* (n, pf, sf, ef, hf) Ft_2 \iff Ft_1 \rightarrow l^* (n, pf, sf, ef, hf) Ft_2$.
We assume (3) \( F_{tf1} \rightarrow *_{(n, pf, sf, ef, hf)} F_{tf2} \) and prove
(4) \( F_{tf1} \rightarrow l*_{(n, pf, sf, ef, hf)} F_{tf2} \).

From (3) we know that there exists \( F_{t'} \in T_{\text{Formula}} \) such that
(5) \( F_{tf1} \rightarrow (pf, s \uparrow (\max(0, pf-hf), \min(pf, hf)), sf(pf), c) F_{t'} \) and
(6) \( F_{t'} \rightarrow *_{(n-1, pf+1, sf, ef, hf)} F_{tf2} \)
hold, where \( c = (ef, \{(X, sf(ef(X))) | X \in \text{dom}(ef)\}) \).

From (6), by the induction hypothesis we get
(7) \( F_{t'} \rightarrow l*_{(n-1, pf+1, sf, ef, hf)} F_{tf2} \).

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) \( F_{t'} = F_{tf2} \).

From (8) and (5) we get
(9) \( F_{tf1} \rightarrow (pf, s \uparrow (\max(0, pf-hf), \min(pf, hf)), sf(pf), c) F_{tf2} \).

On the other hand, by the definition of \( \rightarrow l* \) we have
(10) \( F_{tf1} \rightarrow l*_{(0, pf, sf, ef, hf)} F_{tf1} \).

From (10) and (9), by the definition of \( \rightarrow l* \), we get
(11) \( F_{tf1} \rightarrow l*_{(1, pf, sf, ef, hf)} F_{tf2} \).

Since \( n-1 = 0 \), we get that [4] holds:
[4] \( F_{tf1} \rightarrow l*_{(n, pf, sf, ef, hf)} F_{tf2} \).

Case (ii)
--------

From (7), by the definition of \( \rightarrow l* \) with history, there exists \( F_{t''} \in T_{\text{Formula}} \) such that
(12) \( F_{t'} \rightarrow l*_{(n-2, pf+1, sf, ef, hf)} F_{t''} \).
(13) $F_t'' \rightarrow (pf+n-2, sf^\uparrow (\max(0, pf+n-2-hf), \min(pf+n-2, hf)), sf(pf+n-2), c) F_t f_2$, 
where $c = (ef, \{ (X, sf(ef(X))) \mid X \in \text{dom}(ef) \})$.

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

(14) $F_t' \rightarrow^* (n-2, pf+1, sf, ef, hf) F_t''$.

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

(15) $F_t f_1 \rightarrow^* (n-1, pf, sf, ef, hf) F_t''$.

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

(16) $F_t f_1 \rightarrow l^* (n-1, pf, sf, ef, hf) F_t''$.

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

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

$(\Leftarrow)$

We assume

(17) $F_t f_1 \rightarrow l^* (n, pf, sf, ef, hf) F_t f_2$

and prove

[18] $F_t f_1 \rightarrow^* (n, pf, sf, ef, hf) F_t f_2$.

From (17), by the definition of $\rightarrow l^*$ with history, we know that there exists $F_t' \in \text{TFormula}$ such that

(19) $F_t f_1 \rightarrow l^* (n-1, pf, sf, ef) F_t'$ and
(20) $F_t' \rightarrow (pf+n-1, s^\uparrow (\max(0, pf+n-1-hf), \min(pf+n-1, hf)), sf(pf+n-1), c) F_t f_2$, 
hold, where $c = (ef, \{ (X, sf(ef(X))) \mid X \in \text{dom}(ef) \})$.

From (19), by the induction hypothesis we get

(21) $F_t f_1 \rightarrow^* (n-1, pf, sf, ef, hf) F_t'$

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

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

Case (i)

In this case, from (21) we get $F_t' = F_t f_1$, which together with (20) and the fact $n-1=0$ implies

(22) $F_t f_1 \rightarrow (pf, s^\uparrow (\max(0, pf-hf), \min(pf, hf)), sf(pf), c) F_t f_2$. 51
On the other hand, by the definition of $\rightarrow^*$ with history we have

$$(23) \text{Ftf}_2 \rightarrow^*(0, pf+1, sf, ef, hf) \text{Ftf}_2.$$  

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

$$(24) \text{Ftf}_2 \rightarrow^*(1, pf, sf, ef, hf) \text{Ftf}_2.$$  

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

Case (ii)

From (21), by the definition of $\rightarrow^*$ with history, there exists $\text{Ft}'' \in TFormula$ such that

$$(25) \text{Ftf}_1 \rightarrow (pf, s \uparrow (\max(0, pf-hf), \min(pf, hf)), sf(pf), c) \text{Ft}''$$  

$$(26) \text{Ft}'' \rightarrow^*(n-2, pf+1, sf, ef, hf) \text{Ft}',$$

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

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

$$(27) \text{Ft}'' \rightarrow l^*(n-2, pf+1, sf, ef, hf) \text{Ft}'.$$

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

$$(28) \text{Ft}'' \rightarrow l^*(n-1, pf+1, sf, ef, hf) \text{Ftf}_2.$$  

From (28), by the induction hypothesis we get

$$(29) \text{Ft}'' \rightarrow^*(n-1, pf+1, sf, ef, hf) \text{Ftf}_2.$$  

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

$$(18) \text{Ftf}_1 \rightarrow^*(n, pf, sf, ef, hf) \text{Ftf}_2.$$
A.5 Lemma 3: History Cut-Off Lemma

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

Proof

-----

Parametrization:

\[ S(F) : \Leftrightarrow \]
\[ \forall Ft \in T\text{formula}, p \in \mathbb{N}, s \in \text{Stream}, h \in \mathbb{N}, d \in \mathbb{N}^{\infty}, e \in \text{Environment}, re \in \text{RangeEnv}: \]
\[ \vdash (re \vdash F : (h,d)) \land \text{dom}(e) = \text{dom}(re) \land \]
\[ \forall Y \in \text{dom}(e): \text{re}(Y).1 + i p \leq i \text{e}(Y) \leq i \text{re}(Y).2 + i p \Rightarrow \]
\[ \text{let } c := (e, \{(X, s(e(X))) \mid X \in \text{dom}(e))\} \]
\[ \forall h' \in \mathbb{N}: h' \geq h \Rightarrow \]
\[ T(F) \rightarrow (p, s[p], s(p), c) Ft \]
\[ \Leftrightarrow \]
\[ T(F) \rightarrow (p, s[\max(0,p-h'), \min(p,h')], s(p), c) Ft \]

We prove \( \forall F \in \text{Formula} S(F) \) by structural induction over \( F \).

CASE 1. \( F = \emptyset X \). \( T(F) = \text{next}(TV(X)) \).

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

We take \( Ft, pf, sf, hf, df, ef, ref \) arbitrary but fixed and assume
\( Ft \in T\text{formula}, pf \in \mathbb{N}, sf \in \text{Stream}, hf \in \mathbb{N}, df \in \mathbb{N}^{\infty}, ef \in \text{Environment}, \)
\( ref \in \text{RangeEnv} \).

Assume

(1.1) \( \vdash (ref \vdash F : (hf, df)) \)

(1.2') \( \text{dom}(ef) = \text{dom}(ref) \)

(1.2) \( \forall Y \in \text{dom}(ef): \text{ref}(Y).1 + i pf \leq i \text{ef}(Y) \leq i \text{ref}(Y).2 + i pf \)

Define

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

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

(1.4) \( hf' \geq hf \)

And prove

[1.5] \( T(F) \rightarrow (pf, sf[pf], sf(pf), c) Ft \)
\[ \Leftrightarrow \]
\[ T(F) \rightarrow (pf, sf \uparrow (\max(0, pf-hf'), \min(pf, hf')), sf(pf), c) Ftf. \]

\[ T(F) = \text{next}(TV(X)). \] By the definition of \( \rightarrow \) for next(TV(X)), Ftf in [1.5] depends only whether \( X \in \text{dom}(c.1) \), which is the same in both sides if the equivalence. Hence, [1.5] holds.

CASE 2. \( F = \neg F_1. \) \( T(F) = \text{next}(TN(T(F_1))). \)

We take Ftf, pf, sf, hf, df, ef, ref arbitrary but fixed and assume Ftf \( \in \) Tformula, pf \( \in \) N, sf \( \in \) Stream, hf \( \in \) N, df \( \in \) N\(^\infty\), ef \( \in \) Environment, ref \( \in \) RangeEnv.

Assume

(2.1) \( \vdash (\text{ref} \vdash F : (hf, df)) \)
(2.2') dom(ef) = dom(ref)
(2.2) \( \forall Y \in \text{dom(ef)}: \text{ref}(Y).1 + i pf \leq i \text{ef}(Y) \leq i \text{ref}(Y).2 + i pf \)

Define

(2.3) \( c := (ef, \{(X, sf(ef(X))) | X \in \text{dom(ef)}\}) \)

Take hf' arbitrary but fixed. Assume

(2.4) hf' \geq hf

And prove

[2.5] \( T(F) \rightarrow (pf, sf\downarrow pf, sf(pf), c) Ftf \)
\[ \leftrightarrow \]
\( T(F) \rightarrow (pf, sf\uparrow (\max(0, pf-hf'), \min(pf, hf')), sf(pf), c) Ftf. \)

From (2.1), by the definition of \( \rightarrow \) for next(TN(T(F_1))), we get

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

We prove [2.5] in both directions.

(\( \Rightarrow \)) We assume

(2.7) \( T(\neg F_1) \rightarrow (pf, sf\downarrow pf, sf(pf), c) Ftf \)

and prove

[2.8] \( T(F) \rightarrow (pf, sf\uparrow (\max(0, pf-hf'), \min(pf, hf')), sf(pf), c) Ftf. \)

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

C1. Ftf = next(TN(next(f'))) for some f' \( \in \) TFormulaCore, such that

(2.8) \( T(F_1) \rightarrow (pf, sf\downarrow pf, sf(pf), c) \text{next}(f'). \)

We instantiate the induction hypothesis with
From (2.8), by (2.6), (2.2), (2.3), (2.4), and the induction hypothesis, we get

$$T(F_1) \to (pf, sf^{\uparrow}(\max(0,pf-hf'),\min(pf,hf')), sf(pf), c) \text{ next}(f').$$

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

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

$$T(F_1) \to (pf, sf\downarrow pf, sf(pf), c) \text{ done}(\text{true}).$$

From (2.10), by (2.6), (2.2), (2.3), (2.4), and the induction hypothesis, we get

$$T(F_1) \to (pf, sf^{\uparrow}(\max(0,pf-hf'),\min(pf,hf')), sf(pf), c) \text{ done}(\text{true}).$$

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

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

$(\Leftarrow)$ We assume

$$T(\neg F) \to (pf, sf^{\uparrow}(\max(0,pf-hf'),\min(pf,hf')), sf(pf), c) \text{ Ftf}$$

and prove

$$T(\neg F_1) \to (pf, sf\downarrow pf, sf(pf), c) \text{ Ftf}.\tag{2.13}$$

[2.13] can be proved by the same reasoning as the case $(\Rightarrow)$ 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 $F_{tf},pf,sf,hf,df,ef,ref$ arbitrary but fixed and assume $F_{tf} \in \text{Tformula}, pf \in \mathbb{N}, sf \in \text{Stream}, hf \in \mathbb{N}, df \in \mathbb{N}^{\infty}, ef \in \text{Environment}, ref \in \text{RangeEnv}.$

Assume

$(3.1) \vdash (ref \vdash F : (hf,df))$

$(3.2') \text{ dom}(ef) = \text{ dom}(ref)$

$(3.2) \forall Y \in \text{ dom}(ef): ref(Y).1 + i pf \leq i ef(Y) \leq i ref(Y).2 + i pf$

Define

$(3.3) cf := (ef, \{(X, sf(ef(X))) \mid X \in \text{ dom}(ef)\})$

Take $hf' \in \mathbb{N}$ arbitrary but fixed. Assume

$(3.4) hf' \geq hf$
And prove

\[ 3.5 \quad T(F) \rightarrow (pf, sf \downarrow pf, sf(pf), cf) Ff \]
\[ \quad \leftrightarrow \]
\[ T(F) \rightarrow (pf, sf \uparrow (\max(0,pf-hf'), \min(pf,hf')), sf(pf), cf) Ff. \]

From (3.1) and the assumption that \( hf \in \mathbb{N} \), \( df \in \mathbb{N}^{\infty} \), by the definition of \( \vdash \) for \( F1 \& F2 \), there exist \( h1, d1, h2 \in \mathbb{N}, d2 \in \mathbb{N}^{\infty} \) such that

(3.6) \( \vdash (\text{ref} \vdash F1 : (h1,d1)) \)

(3.7) \( \vdash (\text{ref} \vdash F2 : (h2,d2)) \)

(3.8) \( hf = \max(h1, h2+d1). \)

We prove [3.5] in both directions.

\((\Longrightarrow)\) We assume

(3.9) \( T(F1\&F2) \rightarrow (pf, sf \downarrow pf, sf(pf), cf) Ff \)

and prove

\[ 3.10 \quad T(F1\&F2) \rightarrow (pf, sf \uparrow (\max(0,pf-hf'), \min(pf,hf')), sf(pf), cf) Ff. \]

From (3.9), we prove [3.10] by case distinction over \( Ff \):

C1. \( Ff = \text{next}(TCS(\text{next}(f1), T(F2))) \) for some \( f1 \in TFormulaCore \) such that

(3.11) \( T(F1) \rightarrow (pf, sf \downarrow pf, sf(pf), cf) \text{ next}(f1). \)

We instantiate the induction hypothesis as \( F := F1, Ft := \text{next}(f1), \)
\( p := pf, s := sf, h := h1, d := d1 \) (since \( d1 \in \mathbb{N} \), we have \( d1 \in \mathbb{N}^{\infty} \)), \( e := ef, re := \text{ref}, \)
\( c := cf, h' := hf. \) Then from the IH by (3.2'), (3.2), (3.3), (3.4), (3.6),
(3.8), (3.11) we get

(3.12) \( T(F1) \rightarrow (pf, sf \uparrow (\max(0,pf-hf'), \min(pf,hf')), sf(pf), cf) \text{ next}(f'). \)

From (3.12), by the definition of \( \rightarrow \) for \( T(F1\&F2) \), we get [3.10].

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

(3.13) \( T(F1) \rightarrow (pf, sf \downarrow pf, sf(pf), cf) \text{ done}(false). \)

We instantiate the induction hypothesis as \( F := F1, Ft := \text{done}(false), \)
\( p := pf, s := sf, h := h1, d := d1 \) (since \( d1 \in \mathbb{N} \), we have \( d1 \in \mathbb{N}^{\infty} \)), \( e := ef, re := \text{ref}, \)
\( c := cf, h' := hf. \) Then from the IH by (3.2'), (3.2), (3.3), (3.4), (3.6),
(3.8), (3.13), we get

(3.14) \( T(F1) \rightarrow (pf, sf \uparrow (\max(0,pf-hf'), \min(pf,hf')), sf(pf), c) \text{ done}(false). \)

From (3.14), by the definition of \( \rightarrow \) for \( T(F1\&F2) \), we get [3.10].

C3. \( Ff = Ft2 \) for some \( Ft2 \in TFormula \). This happens when we have
(3.15) \( T(F1) \rightarrow (pf, sf \uparrow pf, sf(pf), cf) \) done(true) and
(3.16) \( T(F2) \rightarrow (pf, sf \uparrow pf, sf(pf), cf) \) \( F_t2 \).

From (3.4,3.8), we have

(3.17) \( hf' \geq hf \geq h1 \)
(3.18) \( hf' \geq hf \geq h2 \)

We instantiate the induction hypothesis as \( F := F1, F_t := done(true), \)
\( p := pf, s := sf, h := h1, d := d1 \) (since \( d \in \mathbb{N} \), we have \( d \in \mathbb{N}^\infty \)), \( e := ef, r := ref, \)
\( c := cf, h' := hf' \). Then from the IH by (3.2'),(3.2),(3.3),(3.6),(3.17),
(3.15) we get

(3.19) \( T(F1) \rightarrow (pf, sf \uparrow (\max(0, pf-hf'), \min(pf, hf')), sf(pf), cf) \) done(true).

Next, we instantiate the induction hypothesis as \( F := F2, F_t := F_t2, \)
\( p := pf, s := sf, h := h1, d := d2, e := ef, r := ref, c := cf, h' := hf \). Then from the
IH by (3.2'),(3.2),(3.3),(3.7),(3.16),(3.18) we get

(3.20) \( T(F2) \rightarrow (pf, sf \uparrow (\max(0, pf-hf'), \min(pf, hf')), sf(pf), cf) \) \( F_t2 \).

From (3.19) and (3.20), by the definition of \( \rightarrow \) for \( T(F1\&\&F2) \), we get [3.10].

(\( \Longleftrightarrow \)) We assume

(3.21) \( T(F1\&\&F2) \rightarrow (pf, sf \uparrow (\max(0, pf-hf'), \min(pf, hf')), sf(pf), cf) \) \( F_t \).

and prove

[3.22] \( T(F1\&\&F2) \rightarrow (pf, sf \downarrow pf, sf(pf), cf) \) \( F_t \)

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

CASE 4. \( F = F1 /\ F2 \). \( T(F) = next(TCP(T(F1),T(F2))) \).

We take \( F_t, pf, sf, hf, df, ef, ref \) arbitrary but fixed and assume
\( F_t \in \text{Formula}, pf \in \mathbb{N}, sf \in \text{Stream}, hf \in \mathbb{N}, df \in \mathbb{N}^\infty, ef \in \text{Environment}, \)
\( ref \in \text{RangeEnv} \).

Assume

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

Define

(4.3) \( cf := (ef, \{(X, sf(ef(X))) | X \in \text{dom}(ef)\}) \)

Take hf' arbitrary but fixed. Assume

57
(4.4) \(h_f' \geq h_f\)

And prove

\[4.5\] \(T(F) \rightarrow (pf, sf\downarrow pf, sf(pf), cf) F_t\)
\[\leftrightarrow\]
\(T(F) \rightarrow (pf, sf\uparrow (\max(0, pf-h_f'), \min(pf, h_f')), sf(pf), cf) F_t\)

From (4.1) and the assumption that \(h_f \in \mathbb{N}\), \(d_f \in \mathbb{N}_\infty\), by the definition of \(\vdash\) for \(F_1 \land F_2\), there exist \(h_1, h_2 \in \mathbb{N}, d_1, d_2 \in \mathbb{N}_\infty\) such that

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

From (4.4, 4.8), we have

\[4.9\] \(h_f' \geq h_f \geq h_1\)
\[4.10\] \(h_f' \geq h_f \geq h_2\)

We prove [4.5] in both directions.

(\(\Rightarrow\)) We assume

\[4.11\] \(T(F_1 \land F_2) \rightarrow (pf, sf\downarrow pf, sf(pf), cf) F_t\)

and prove

\[4.12\] \(T(F_1 \land F_2) \rightarrow (pf, sf\uparrow (\max(0, pf-h_f'), \min(pf, h_f')), sf(pf), cf) F_t\).

From (4.11), we prove [4.10] by case distinction over \(F_t\):

C1. \(F_t = \text{next}(TCS(\text{next}(f_1), \text{next}(f_2)))\) for some \(f_1, f_2 \in T\text{FormulaCore}\) such that

\[4.13\] \(T(F_1) \rightarrow (pf, sf\downarrow pf, sf(pf), cf) \text{ next}(f_1)\).
\[4.14\] \(T(F_2) \rightarrow (pf, sf\downarrow pf, sf(pf), cf) \text{ next}(f_2)\).

We instantiate the induction hypothesis as \(F := F_1, F_t := \text{next}(f_1), p := pf, s := sf, h := h_1, d := d_1, e := ef, re := \text{ref}, c := cf, h' := h_f'\).

Then from the IH, by (4.6), (4.2'), (4.2), (4.3), (4.9), (4.13) we get

\[4.15\] \(T(F_1) \rightarrow (pf, sf\uparrow (\max(0, pf-h_f'), \min(pf, h_f')), sf(pf), cf) \text{ next}(f_1)\).

Next, we instantiate the induction hypothesis as \(F := F_1, F_t := \text{next}(f_2), p := pf, s := sf, h := h_2, d := d_2, e := ef, re := \text{ref}, c := cf, h' := h_f'\).

Then from the IH, by (4.7), (4.2'), (4.2), (4.3), (4.10), (4.14) we get

\[4.16\] \(T(F_2) \rightarrow (pf, sf\uparrow (\max(0, pf-h_f'), \min(pf, h_f')), sf(pf), c) \text{ next}(f_2)\).

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

C2. \(F_t = \text{next}(f_1)\) for some \(f_1 \in T\text{FormulaCore}\) such that
(4.17) $T(F1) \rightarrow (pf, sf\downarrow pf, sf(pf), c) \text{ next}(f1)$.
(4.18) $T(F2) \rightarrow (pf, sf\downarrow pf, sf(pf), c) \text{ done}(true)$.

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

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

C3.1
(4.19) $T(F1) \rightarrow (pf, sf\downarrow pf, sf(pf), c) \text{ next}(f1)$.
(4.20) $T(F2) \rightarrow (pf, sf\downarrow pf, sf(pf), c) \text{ done}(false)$.

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

C3.2
(4.21) $T(F1) \rightarrow (pf, sf\downarrow pf, sf(pf), c) \text{ done}(false)$.

We instantiate the induction hypothesis as $F:=F1,Ft:=\text{done}(false)$,
$p:=pf,s:=sf,h:=h1,d:=d1,e:=ef,ref:=ref,c:=cf,h':=hf'$.
Then from the IH, by (4.6),(4.2'),(4.2),(4.3),(4.9),(4.21) we get

(4.22) $T(F1) \rightarrow (pf, sf\uparrow(\max(0,pf-hf'),\min(pf,hf')), sf(pf), c) \text{ done}(false)$.

From (4.22), by the definition of $\rightarrow$ for $T(F1 \land F2)$, we get [4.12].

C4. $Ftf=Ft2$ for some $Ft2 \in TFormula$. This happens when

(4.23) $T(F1) \rightarrow (pf, sf\downarrow pf, sf(pf), c) \text{ done}(true)$.
(4.24) $T(F2) \rightarrow (pf, sf\downarrow pf, sf(pf), c) \text{ Ft2}$.

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

($\iff$) We assume

(4.25) $T(F1 \land F2) \rightarrow (pf, sf\uparrow(\max(0,pf-hf'),\min(pf,hf')), sf(pf), c) Ftf$

and prove

[4.26] $T(F1 \land F2) \rightarrow (pf, sf\downarrow pf, sf(pf), c) Ftf$

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

It finishes the proof of CASE 4.

CASE 5. $F = \forall X \in B1..B2:F1$ $T(F) = \text{next}(TA(X, T(B1), T(B2), T(F1)))$.

We take $Ftf,pf,sf,hf,df,ef,ref$ arbitrary but fixed and assume
$Ftf \in TFormula, pf \in \mathbb{N}, sf \in Stream, hf \in \mathbb{N}, df \in \mathbb{N}\infty, ef \in \text{Environment},$
$ref \in \text{RangeEnv}$.

Assume
⊢ (ref ⊢ F : (hf, df))
(5.2') dom(ef) = dom(ref)
∀Y ∈ dom(ef): ref(Y).1 + i pf ≤ i ef(Y) ≤ i ref(Y).2 + i pf

Define

(5.3) cf := (ef, {(X, sf(ef(X))) | X ∈ dom(ef)})

Take hf' arbitrary but fixed. Assume

(5.4) hf' ≥ hf

And prove

[5.5] T(F) → (pf, sf ↑ pf, sf(pf), cf) F tf
⇔ T(F) → (pf, sf ↑ (max(0, pf - hf'), min(pf, hf')), sf(pf), cf) F tf

Let b1, b2 ∈ BoundValue and F t1 ∈ TFormula be such that

(5.6) b1 = T(B1)
(5.7) b2 = T(B2)
(5.7') F t1 = T(F1)

From (5.1), taking into account the assumptions hf ∈ N and df ∈ N∞, we know by the definition of ⊢ for "forall" for some l1 ∈ Z, u1, u2 ∈ Z∞, h1 ∈ N, d1 ∈ N∞:

(5.I.1) ⊢ (ref ⊢ B1 : (l1, u1))
(5.I.2) ⊢ (ref ⊢ B2 : (l2, u2))
(5.I.3) ⊢ (ref[X := (l1, u2)] ⊢ F1 : (h1, d1))
(5.I.4) hf = max∞(h1, N∞(- i(l1))) = (by h1 ∈ N, l1 ∈ Z) max(h1, |l1|).
(5.I.5) df = max∞(d1, N∞(u2))

We define

(5.I.6) p1 = b1(cf)
(5.I.7) p2 = b2(cf)

From (5.I.1) (5.I.2), (5.2'), (5.2), (5.3), (5.6), (5.7), (5.I.6), (5.I.7), we know by Lemma 9 (soundness of bound analysis)

(5.I.B.1) l1 + i pf ≤ i p1 ≤ i u1 + i pf
(5.I.B.2) l2 + i pf ≤ i p2 ≤ i u2 + i pf

(In fact, instead of l1 + i pf we can write l1 + pf in (5.I.B.1), because neither l1 nor pf can be ∞.)

Instantiating the induction hypothesis S(F1) with s := sf, h := h1, d := d1, re := ref[X := (l1, u2)], we know with (5.I.3), (5.2'), (5.3), (5.7')

(5.I.F)
∀F t ∈ TFormula, p ∈ N, e ∈ Environment:

60
We prove [5.5] in both directions.

(⇒) We assume

(5.8) \(\text{next}(\text{TA}(X,b_1,b_2,F_t)) \rightarrow (pf, sf\downarrow pf, sf(pf), cf) F_t\)

and prove

[5.9] \(\text{next}(\text{TA}(X,b_1,b_2,F_t)) \rightarrow (pf, sf\uparrow (\max(0,pf-h'),\min(pf,h')), sf(pf), cf) F_t\).

From (5.8), we have two cases:

CASE 1 (Rule 1 for TA)
----------------------

We know from the rule, (5.I.6) and (5.I.7) that

(5.10.1) \(p_1 = \infty \lor p_1 > \infty b_2(cf)\)
(5.10.2) \(F_t = \text{done(true)}\)

From (5.I.6), (5.I.7), (5.10.1), (5.10.2), we can derive with "Rule 1 for TA" [5.9].

CASE 2: (Rule 2 for TA)
-----------------------

We know from the rule, (5.I.6) and (5.I.7) that

(5.16) \(p_1 \neq \infty\)
(5.16') \(p_1 \leq \infty p_2\)
(5.17) next(\text{TA}_0(X,p_1,p_2,F_t)) \rightarrow (pf, sf\downarrow pf, sf(pf), cf) F_t

To prove [5.9], it suffices, by "Rule 2 for TA", together with (5.I.6), (5.I.7), (5.16), (5.16') to prove

[5.21] next(\text{TA}_0(X,p_1,p_2,F_t)) \rightarrow (pf, sf\uparrow (\max(0,pf-h'),\min(pf,h')), sf(pf), cf) F_t

Subcase 1.
(5.23) \(pf < p_1\).

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

In this case from (5.17) and "Rule 1 for TA_0" we have
\(F_t = \text{next(TA}_0(X,p_1,p_2,F_t))\). Then [5.21] follows from (5.17), (5.23) and "Rule 1 for TA_0".

61
Subcase 2.
(5.24) \( pf \geq p1. \)

We define

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

Before proving [5.21], we establish the following auxiliary fact:

[aux] \( \forall p0: p1 \leq p0 < \infty \min \infty (pf,p2+\infty) \Rightarrow 0 \leq p0-pf+|ms| < |ms| \)

Proof of [aux]: Take arbitrary \( p0 \) and assume

(aux1) \( p1 \leq p0 \)
(aux2) \( p0 < \infty \min \infty (pf,p2+\infty) \)

We have to show

[aux3] \( 0 \leq p0-pf+|ms| \)
[aux4] \( p0-pf+|ms| < |ms| \)

From (aux2) we have \( p0<pf \) and thus [aux4] holds.

To show [aux3], we show

[aux3.1] \( pf \leq p0+|ms| \)

From (5.25), we know

(aux3) \( |ms| = \min(pf,hf') \)

From (aux3), to show [aux3.1], it suffices to show

[aux3.2] \( pf \leq p0+\min(pf,hf') \)

We proceed by case distinction:

(aux4) Case \( pf \leq hf' \)
- ------------------

From (aux4), to show [aux3.2], it suffices to show

[aux3.2.1] \( pf \leq p0+pf \)

From \( p0 \) in Nat, we have

(aux5) \( p0 \geq 0 \)

and thus [aux3.2.1]

(aux6) Case \( pf > hf' \)
- ------------------

From (aux6), to show [aux3.2], it suffices to show
From (5.4) we know \( h' \geq h_f \). It thus suffices to show

\[ [\text{aux3.2.3}] \quad p_f \leq p_0 + h_f \]

From (aux5.1.4), it suffices to show

\[ [\text{aux3.2.4}] \quad p_f \leq p_0 + \max(h_1,|l_1|). \]

We know

\[ (\text{aux7}) \quad p_0 + \max(h_1,|l_1|) \geq (\text{by (aux1)}) \]
\[ p_1 + \max(h_1,|l_1|) \geq (\text{by } l_1 \in \mathbb{Z}) \]
\[ p_1 - l_1 \geq (\text{by } 5.1.B.1) \quad p_f \]

and thus have [aux3.2.4].

It proves [aux].

==========

From [aux] we can conclude

\[ (5.25') \quad \forall p_0: \quad p_1 \leq p_0 < \infty \min(\infty,(p_f,p_2+\infty)) \Rightarrow (s,f,p_0)=ms(p_0-p_f+|ms|). \]

Now, to prove [5.21], it suffices by "Rule 2 for TA0" to prove

\[ [5.26] \quad \text{next}(\text{TA1}(X,p_2,F_t_1,f_s)) \rightarrow (p_f,ms,sf(p_f),s_f) \quad F_t_f \]

where

\[ (5.27) \quad f_s = \{(p_0,F_t_1,(c_f.1[X \rightarrow p_0],c_f.2[X \rightarrow ms(p_0-p_f+|ms|)]) | p_1 \leq p_0 < \infty \min(\infty,(p_f,p_2+\infty))\}. \]

We prove [5.26] by case distinction over F_t_f.

(c1) F_t_f=\text{done(false)}

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

We prove

\[ [c1.1] \quad \text{next}(\text{TA1}(X,p_2,F_t_1,f_s)) \rightarrow (p_f,ms,sf(p_f),c_f) \quad \text{done(false)}. \]

To prove [c1.1], by Def.\(

\rightarrow\) we need to prove

\[ [c1.2] \quad \exists t \in \mathbb{N}, g \in T\text{Formula}, c \in \text{Context}: \quad (t,g,c) \in f_s_0 \land \vdash g \rightarrow (p_f,ms,sf(p_f),c) \quad \text{done(false)}, \]

where

\[ (c1.3) \quad f_s_0 = \]
\[ \text{if } p_f > \infty \text{ then } f_s \text{ else } f_s \cup \{(p_f,F_t_1,(c_f.1[X \rightarrow p_f],c_f.2[X \rightarrow sf(p_f)])\}) \]

From (5.17), by (c1) we know
(c1.4) \text{next}(\text{TA1}(X,p_2,Ft_1,fs')) \rightarrow (pf, sf \downarrow pf, sf(pf), cf) \text{ done(false)}

where (since p_0 - pf + |sf\downarrow pf| = p_0)

(c1.5) \begin{align*}
fs' &= \{(p_0,Ft_1,(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)\}\}
\end{align*}

From (c1.4) we know by the definition of \(\rightarrow\)

(c1.6) \exists t \in \mathbb{N}, g \in \text{TFormula}, c \in \text{Context}:
\begin{align*}
(t,g,c) \in fs_1 \land \vdash g \rightarrow (pf, sf \downarrow pf, sf(pf), c) \text{ done(false)},
\end{align*}

where

(c1.7) \begin{align*}
fs_1 = \begin{cases}
\text{if } pf > \infty p_2 \text{ then } fs' \text{ else } fs' \cup \{(pf,Ft_1,(cf.1[X \mapsto pf],cf.2[X \mapsto sf(pf)]))\}
\end{cases}
\end{align*}

From (c1.6), we have \((t_1,g_1,c_1)\) such that

(c1.8) \((t_1,g_1,c_1)\in fs_1\) 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) \(g_1 = Ft_1\)

and, hence, \(T(F_1) = g_1\).

From (c1.8), (c1.7), (c1.5), we have

Case 1: \(c_1 = (cf.1[X \mapsto t_1],cf.2[X \mapsto (sf\downarrow pf)(t_1)]) \land p_1 \leq t_1 < \infty \min_{\infty}(pf,p_2+\infty 1)\)
Case 2: \(c_1 = (cf.1[X \mapsto t_1],cf.2[X \mapsto sf(t_1)]) \land pf \leq \infty p_2 \land t_1 = pf\)

and with (5.24) consequently (in both cases)

(c1.12.1) \(p_1 \leq t_1 \leq \infty \min_{\infty}(pf, p_2)\)
(c1.12.2) \(c_1 = (cf.1[X \mapsto t_1],cf.2[X \mapsto (sf\downarrow (pf+1))(t_1)])\)

We have from (c1.12.2)

(c1.13.1) \(c_1.1(X) = t_1\)

We have from (5.2), (5.3) and (c1.12.2),

(c1.13.2) \(\forall Y \in \text{dom}(cf.1) \setminus \{X\}: \text{ref}(Y).1 + i pf \leq i c_1.1(Y) \leq i \text{ref}(Y).2 + i pf\)

From (c1.12.1), (5.I.B.1) and (5.I.B.2), we know

(c1.13.3) \(l_1 + i pf \leq i t_1 \leq i u_2 + i pf\)

We instantiate (5.I.F) with \(Ft:=\text{done(false)}, p:=pf, e:=c_1.1\).

With (5.2'), (5.3), (c1.12.2), (c1.13.2), (c1.13.3), we then have
Since (c1.14) is true for all $h_1' \geq h_1$, it is true, in particular, for $h_f'$, because by (5.4) we have $h_f' \geq h_f$, and in itself, $h_f \geq h_1$ by (5.1.4). Hence, from (c1.14) we get

\[(c1.15)\quad F_t \rightarrow (p_f, s_f \uparrow (\max(0, p_f - h_f'), \min(p_f, h_f')), s_f(p_f), c_1) \text{ done(false)}\]

From (c1.15) and (c1.9) we get

\[(c1.16)\quad F_t \rightarrow (p_f, s_f \uparrow (\max(0, p_f - h_f'), \min(p_f, h_f')), s_f(p_f), c_1) \text{ done(false)}\]

(c1.16), by (5.25'), proves the second conjunct of [c1.2].

Hence, it remains to prove the first conjunct of [c1.2]:

\[c1.3\quad (t_1, g_1, c_1) \in f_s 0.\]

By (c1.8), $(t_1, g_1, c_1) \in f_s 1$. By (c1.7) it means either

\[(c1.17)\quad (t_1, g_1, c_1) = (p_f, F_t, (c.f.1[X \mapsto \rightarrow p_f], c.f.2[X \mapsto \rightarrow s_f(p_f)]))\]

or

\[(c1.18)\quad (t_1, g_1, c_1) \in f_s'.\]

From (c1.17) we get [c1.3] due to the definition of $f_s 0$ in (c1.3).

From (c1.18) we have

\[(c1.19)\quad (t_1, g_1, c_1) = (p_0, F_t, (c.f.1[X \mapsto \rightarrow p_0], c.f.2[X \mapsto \rightarrow (s_f \downarrow p_f)(p_0)]))\]

for some $p_1 \leq p_0 < \infty$ min$(p_f, p_2 + \infty)$.

From (5.25'), (c1.19) and the definition of $f_s$ in (5.27) we get

\[(c1.21)\quad (t_1, g_1, c_1) \in f_s.\]

From (c1.2) we have $f_s \subseteq f_s 0$ and, hence, [c1.3] holds also in this case. It proves (c1).

(c2) $F_t = \text{done(true)}$

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

We prove
next(TA1(X,p2,Ft1,fs)) → (pf, ms, sf(pf), cf) done(true).

To prove [c2.1], by Def. of → ("Rule 2 for TA1") we need to prove

\[ \neg \exists t \in \mathbb{N}, g \in TFormula, c \in \text{Context}: \]
\[ (t,g,c) \in fs0 \land \vdash g \rightarrow (pf,ms,sf(pf),c) \text{ done(false)} \]
and

\[ fs1 = \emptyset \land pf \geq \infty p2 \]

where

\[ fs0 = \]
\[ \begin{cases}
  \text{if } pf > \infty p2 \text{ then } fs \text{ else } \text{fs' } \cup \{(pf,Ft1,(cf.1[X\mapsto pf],cf.2[X\mapsto sf(pf)]))\}
\end{cases} \]

\[ fs1 = \{ (t,next(fc),c) \in TInstance \mid \exists g \in TFormula: (t,g,c) \in fs0 \land \vdash g \rightarrow (pf,ms,sf(pf),c) next(fc) \} \]

From (5.17), by (c2) we know

\[ \text{(c2.5') next(TA0(X,p1,p2,Ft1)) → (pf, sf\downarrow pf, sf(pf), cf) done(true)}. \]

From (c2.5') and (5.24), by the definition of → ("Rule 2 for TA0") we know

\[ \text{(c2.6') next(TA1(X,p2,Ft1,fs')) → (pf, sf\downarrow pf, sf(pf), cf) done(true)}, \]

where

\[ fs' = \{ (p0,Ft1,(cf.1[X\mapsto p0],cf.2[X\mapsto (sf\downarrow pf)(p0)]))) \mid p1 \leq p0 < \infty \text{ min}_{\infty}(pf,p2+1) \} \]

Since p0-pf+|sf\downarrow pf|=p0, from (c2.6') we get

\[ \text{(c2.7) fs' = \{(p0,Ft1,(cf.1[X\mapsto p0],cf.2[X\mapsto (sf\downarrow pf)(p0)]))) \mid p1 \leq p0 < \infty \text{ min}_{\infty}(pf,p2+1) \} \]

From (c2.6), by Def. of → ("Rule 2 for TA1") we know

\[ \text{(c2.8) \neg \exists t \in \mathbb{N}, g \in TFormula, c \in \text{Context}:} \]
\[ (t,g,c) \in fs0' \land \vdash g \rightarrow (pf,ms,sf(pf),c) \text{ done(false)} \]
and

\[ fs1' = \emptyset \land pf \geq \infty p2 \]

where

\[ fs0' = \]
\[ \begin{cases}
  \text{if } pf > \infty p2 \text{ then } fs' \text{ else } fs' \cup \{(pf,Ft1,(cf.1[X\mapsto pf],cf.2[X\mapsto sf(pf)]))\}
\end{cases} \]

\[ fs1' = \{ (t,next(fc),c) \in TInstance \mid \exists g \in TFormula: (t,g,c) \in fs0' \land \vdash g \rightarrow (pf,sp\downarrow pf,sp(pf),c) next(fc) \}. \]

From (5.25'), (5.27) and (c2.7) we get

\[ \text{(c2.13) fs = fs'}. \]

which, by (c2.4) and (c2.10), implies

\[ \text{(c2.14) fs0=fs0'}. \]
To prove \([c2.2]\), we take
\((c2.15) \ (t_0,g_0,c_0) \in \mathcal{F}s_0\)
and prove that
\([c2.16]\) \(g_0 \rightarrow (pf,ms,sf(pf),c_0)\) done(false) does not hold.
From \((c2.15)\) and \((c2.14)\) we have
\((c2.17) \ (t_0,g_0,c_0) \in \mathcal{F}s_0'\).
From \((c2.17)\) and \((c2.8)\) we know
\((c2.18) \ g_0 \rightarrow (pf,sf \downarrow pf,sf(pf),c_0)\) done(false) does not hold.
From \((c2.4)\), \((5.27)\) and \((c2.15)\) we get
\((c2.19) \ g_0 = Ft_1\) and two cases:

Case 1: \(t_0=p_0 \land c_0=cf.1[X \mapsto \rightarrow t_0], cf.2[X \mapsto ms(p_0-pf+|ms|)] \land p_1 \leq p_0 < \infty \min \infty (pf,p_2+\infty 1) \land pf > \infty p_2\)
Case 2: \(t_0=pf \land c_0=cf.1[X \mapsto \rightarrow t_0], cf.2[X \mapsto sf(pf)] \land p_1 \leq \infty p_2\)
These cases can be rewritten and simplified (taking into account \((5.25')\) and \((5.24)\)) into

Case 1: \(c_0=cf.1[X \mapsto \rightarrow t_0], cf.2[X \mapsto sf(pf)](t_0) \land p_1 \leq t_0 < \infty \min \infty (pf,p_2)\)
Case 2: \(c_0=cf.1[X \mapsto \rightarrow t_0], cf.2[X \mapsto sf(t_0)] \land p_1 \leq pf \leq \infty p_2 \land pf = t_0\).
Consequently, in both cases we get
\((c2.20) \ p_1 \leq t_0 < \infty \min \infty (pf,p_2)\) and
\((c2.21) \ c_0=cf.1[X \mapsto \rightarrow t_0], cf.2[X \mapsto sf((pf+1))(t_0)]\)
From \((c2.21)\) we have
\((c2.22) \ c_0.1(X)=t_0\).
From \((5.2)\), \((5.3)\), and \((c2.21)\) we get
\((c2.23) \ \forall Y \in \text{dom}(cf.1)\{X\}: \text{ref}(Y).1 +i pf \leq i c_0.1(Y) \leq i \text{ref}(Y).2 +i pf\).
From \((c2.20)\), \((5.I.B.1)\) and \((5.I.B.2)\), we know
\((c2.24) \ l_1 +i pf \leq i t_0 \leq i u_2 +i pf\).
We instantiate \((5.I.F)\) with \(Ft:=\text{done(false)}\), \(p:=pf\), \(e:=c_0.1\).
With \((5.2')\), \((5.3)\), \((c2.21)\), \((c2.22)\), \((c2.23)\), \((c2.24)\), we then have
\((c2.25) \ \forall h_1' \in \mathbb{N} : h_1' \geq h_1 \Rightarrow \)
\(Ft_1 \rightarrow (pf, sf[pf, sf(pf), c_0)\) done(false)
\(\Leftrightarrow \)
\(Ft_1 \rightarrow (p, sf[\max(0,pf-h_1'), \min(pf,h_1')], sf(pf), c_0)\) done(false)
Since (c2.25) is true for all \( h_1' \geq h_1 \), it is true, in particular, for \( hf' \), because by (5.4) we have \( hf' \geq hf \), and in itself, \( hf \geq h_1 \) (5.1.4). Hence, from (c2.25) we get

\[
(c2.26) \quad F_{t1} \rightarrow (pf, sf, pf, sf(pf), c0) \text{ done(false)}
\]

\[
\Leftrightarrow
F_{t1} \rightarrow (p, sf\uparrow(\max(0, pf-hf'), \min(pf, hf')), sf(pf), c0) \text{ done(false)}
\]

From (c2.26), (c2.18), and (c2.19) we get

\[
(c2.27) \quad F_{t1} \rightarrow (p, sf\uparrow(\max(0, pf-hf'), \min(pf, hf')), sf(pf), c0) \text{ done(false)}
\]
does not hold.

From (c2.27), by (5.25), we get [c2.16].

To prove [c2.3], note that from (c2.14), (c2.5) and (c2.11) we get

\[
(c2.28) \quad fs_1 = fs_1'.
\]

Now [c2.3] follows from (c2.28) and (c2.9). It proves (c2).

\[
\text{(c3) } F_{tf} = \text{next}(TA_1(X, p_2, F_{t1}, fs'))
\]

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

We prove

\[
[c3.1] \quad \text{next}(TA_1(X, p_2, F_{t1}, fs)) \rightarrow (pf, ms, sf(pf), cf) \text{ next}(TA_1(X, p_2, F_{t1}, fs')).
\]

To prove [c3.1], by Def. of \( \rightarrow \) ("Rule 3 for TA1") we need to prove

\[
[c3.2] \quad \neg \exists t \in N, g \in \text{TFormula}, c \in \text{Context}:
\]

\[
(t, g, c) \in fs_0 \wedge \vdash g \rightarrow (pf, ms, sf(pf), c) \text{ done(false)} \text{ and }
\]

\[
[c3.3] \quad \neg (fs_1 = \emptyset \wedge pf \geq \infty p_2)
\]

where

\[
(c3.4) \quad fs_0 =
\]

\[
\begin{cases}
\text{if } pf > \infty p_2 \text{ then } fs \text{ else } fs \cup \{(pf, f, (cf.1[X \mapsto p], cf.2[X \mapsto sf(pf)]))\}
\end{cases}
\]

\[
(c3.5) \quad fs_1 = \{ (t, \text{next}(fc), c) \in T\text{Instance} | \exists g \in \text{TFormula}: (t, g, c) \in fs_0 \wedge \vdash g \rightarrow (pf, ms, sf(pf), c) \text{ next}(fc) \}
\]

From (5.17) by (c3) we know

\[
(c3.5') \quad \text{next}(TA_0(X, p_1, p_2, F_{t1})) \rightarrow (pf, sf\downarrow pf, sf(pf), cf) \text{ next}(TA_1(X, p_2, F_{t1}, fs')).
\]

From (c3.5') and by the definiton of \( \rightarrow \) ("Rule 2 for TA0") we know

\[
(c3.6) \quad \text{next}(TA_1(X, p_2, F_{t1}, fs')) \rightarrow (pf, sf\downarrow pf, sf(pf), cf) \text{ next}(TA_1(X, p_2, F_{t1}, fs'))
\]

where

\[
(c3.6') \quad fs' = \{(p_0, F_{t1}, (cf.1[X \mapsto p_0], cf.2[X \mapsto (sf\downarrow pf)(p_0-pf+|sf\downarrow pf|)]) |
\]

68
\[ p_1 \leq p_0 < \min_{\infty}(p_f, p_2 + \infty) \]

Since \( p_0 - p_f + |s_f| = p_0 \), from (c3.6') we get

\[(c3.7) \quad f_s' = \{ (p_0, F_t, 1, (c_f.1[X \mapsto p_0], c_f.2[X \mapsto s_f(p_f)(p_0)]) | p_1 \leq p_0 < \min_{\infty}(p_f, p_2 + \infty) \} \]

From (c3.6), by Def. of \( \rightarrow \) ("Rule 3 for TA") we know

\[(c3.8) \quad \neg \exists t \in N, g \in T_{Formula}, c \in Context: \quad (t, g, c) \in f_s \land \vdash g \rightarrow (p_f, s_f(p_f), c) \text{ done(false) and} \]

\[(c3.9) \quad \neg (f_s' = \emptyset \land p_f \geq \infty p_2) \]

where

\[(c3.10) \quad f_s = \begin{cases} f_s' & \text{if } p_f > \infty p_2 \text{ then } f_s' \text{ else } f_s' \cup \{ (p_f, F_t, 1, (c_f.1[X \mapsto p_0], c_f.2[X \mapsto s_f(p_f)])) \} \\ f_s' \end{cases} \]

\[(c3.11) \quad f_s' = \{ (t, next(f_c), c) \in T_{Instance} | \exists g \in T_{Formula}: (t, g, c) \in f_s \land \vdash g \rightarrow (p_f, s_f(p_f), c) \text{ next}(f_c) \} . \]

From (5.25'), (5.27) and (c3.7) we get

\[(c3.12) \quad f_s = f_s', \]

which, by (c3.4) and (c3.10), implies

\[(c3.13) \quad f_s = f_s'. \]

Now [c3.2] can be proved in the same as [c2.2] was proved above.

To prove [c3.3], note that from (c3.14), (c3.5) and (c3.11) we get

\[(c3.28) \quad f_s = f_s'. \]

Now [c3.3] follows from (c3.28) and (c3.9). It proves (c3).

Hence, the direction \( \Longrightarrow \) is proved.

\( \Longleftrightarrow \) This direction can be proved with the same reasoning as \( \Longrightarrow \).

It finishes the proof of CASE 5.

It finishes the proof of Lemma 3.
A.6 Lemma 4: $n$-Step Reductions to done Formulas for TN, TCS, TCP

Statement 1. TN Formulas.

∀F ∈ Formula, n ∈ N, p ∈ N, s ∈ Stream, e ∈ Environment, Ft ∈ TFormula :
T(F) → *(n, p, s, e) done(false) ⇒ next(TN(T(F))) → *(n, p, s, e) done(true) ∧
T(F) → *(n, p, s, e) done(true) ⇒ next(TN(T(F))) → *(n, p, s, e) done(false)

Proof
-----

We take Ff, sf, ef arbitrary but fixed and prove the formula
∀n ∈ N, p ∈ N :
T(Ff) → *(n, pf, sf, ef) done(false) ⇒
next(TN(T(Ff))) → *(n, pf, sf, ef) done(true)
∧
T(Ff) → *(n, pf, sf, ef) done(true) ⇒
next(TN(T(Ff))) → *(n, p, s, e) done(false)

by induction over n. Since T(Ff) is a next formula, for n=0 the antecedents of both conjuncts are false and the statement is trivially true.

Assume
(TN.1) ∀p ∈ N:
T(Ff) → *(n, p, sf, ef) done(false) ⇒
next(TN(T(Ff))) → *(n, p, sf, ef) done(true)
(TN.2) ∀p ∈ N:
T(Ff) → *(n, pf, sf, ef) done(true) ⇒
next(TN(T(Ff))) → *(n, p, s, e) done(false)

Prove

[TN.3] ∀p ∈ N:
T(Ff) → *(n+1, p, sf, ef) done(false) ⇒
next(TN(T(Ff))) → *(n+1, p, sf, ef) done(true)
and
[TN.4] ∀p ∈ N:
T(Ff) → *(n+1, p, sf, ef) done(true) ⇒
next(TN(T(Ff))) → *(n+1, p, s, e) done(false)

To prove [TN.3], we take pf arbitrary but fixed, assume
(TN.5) T(Ff) → *(n+1, pf, sf, ef) done(false)
and prove
[TN.6] next(TN(T(Ff))) → *(n+1, pf, sf, ef) done(true)

From (TN.5) by definition →* without history we know that there exists
Ft ∈ TFormula such that
(TN.7) T(Ff) → (pf, sf | pf, sf(pf), c) Ft
(TN.8) Ft → *(n, pf+1, sf, ef) done(false)
where $c = (ef, \{(X, sf(ef(X))) \mid X \in \text{dom}(ef)\})$.

We proceed by case distinction over $F_t$.

Case 'next': If $F_t$ is a next formula, then there exists $F_1 \in \text{Formula}$ such that

$\text{(TN.9)} \quad F_t = T(F_1)$

From (TN.9) and (TN.8) by (TN.1) we get

$\text{(TN.10)} \quad \text{next}(T(T(F_1))) \rightarrow^*(n, pf+1, sf, ef) \text{ done(true)}$

From (TN.7) by the definition of $\rightarrow$ we get

$\text{(TN.11)} \quad \text{next}(T(T(F_f))) \rightarrow^{pf, sf \uparrow pf, sf(pf), c} \text{ next}(T(T(F_1)))$

From (TN.11) and (TN.10) by the definition of $\rightarrow^*$ without history we get [TN.6].

Case 'done': If $F_t$ is a 'done' formula, then by (TN.8), we have

$\text{(TN.12)} \quad n = 0 \text{ and }$
$\text{(TN.13)} \quad F_t = \text{done(false)}$

From (TN.7) and (TN.13), by the definition of $\rightarrow$, we get

$\text{(TN.14)} \quad \text{next}(T(T(F_f))) \rightarrow^{pf, sf \uparrow pf, sf(pf), c} \text{ done(true)}$

On the other hand, from the definition of $\rightarrow^*$ we know

$\text{(TN.15)} \quad \text{done(true)} \rightarrow^*(0, pf+1, sf, ef) \text{ done(true)}$

From (TN.14), (TN.15), (TN.12), by the definition of $\rightarrow^*$ we get [TN.6].

Hence, we proved [TN.6] for both cases of $F_t$. This proves [TN.3]. [TN.4] can be proved analogously.

Statement 2. TCS Formulas.

$\forall p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}:$

$\forall F_{t1}, F_{t2} \in \text{Formula}, n \in \mathbb{N},$

$n > 0 \land F_{t1} \rightarrow^*(n, p, s, e) \text{ done(false)} \Rightarrow$

$\text{next}(TCS(F_{t1}, F_{t2})) \rightarrow^*(n, p, s, e) \text{ done(false)} \land$

$\forall 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(true)} \land F_{t2} \rightarrow^*(n_2, p, s, e) \text{ done(b)} \Rightarrow$

$\text{next}(TCS(F_{t1}, F_{t2})) \rightarrow^*(\max(n_1, n_2), p, s, e) \text{ done(b)}$

Proof

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

We split the statement in two:
∀p∈N, s∈Stream, e∈Environment, Ft1,Ft2∈TFormula, n∈N :
    n>0 ∧ Ft1 →*(n,p,s,e) done(false) ⇒
    next(TCS(Ft1,Ft2)) →*(n,p,s,e) done(false)

∀p∈N, s∈Stream, e∈Environment, Ft1,Ft2∈TFormula, n1,n2∈N, b∈Bool :
    n1>0 ∧ n2>0 ∧ Ft1 →*(n1,p,s,e) done(true) ∧ Ft2 →*(n2,p,s,e) done(b) ⇒
    next(TCS(Ft1,Ft2)) →*(max(n1,n2),p,s,e) done(b).

Proof of [TCS1]  
--------------
We take sf,ef arbitrary but fixed and define

Φ(n) :⇔
∀p∈N, Ft1,Ft2∈TFormula:
    n>0 ∧ Ft1 →*(n,p,sf,ef) done(false) ⇒
    next(TCS(Ft1,Ft2)) →*(n,p,sf,ef) done(false))

We prove ∀n∈N: Φ(n) by induction over n. For n=0 the formula is trivially true. We start the induction from 1. Prove:

[TCS1.a] Φ(1) and
[TCS1.b] ∀n∈N: Φ(n) ⇒ Φ(n+1)

Proof of [TCS1.a]
--------------
We take pf,Ft1f,Ft2f arbitrary but fixed and assume

(TCS1.1) 1>0
(TCS1.2) Ft1f →*(1,pf,sf,ef) done(false).

We want to prove

[TCS1.3] next(TCS(Ft1f,Ft2f)) →*(1,pf,sf,ef) done(false).

From (TCS1.2), by the definition of →* without history, there exists
Ft∈TFormula such that

(TCS1.4) Ft1f →(p,sf↓pf,sf(pf),c) Ft and
(TCS1.5) Ft →*(0,pf+1,sf,ef) done(false)

where

(TCS1.6) c={(ef, {(X,sf(ef(X)))| X∈dom(ef))}).

From (TCS1.5), by the definition of →* without history, we get

(TCS1.7) Ft=done(false).

From (TCS1.7) and (TCS1.4), by the definition of → for TCS, we get

(TCS1.8) next(TCS(Ft1f,Ft2f)) →(p,sf↓pf,sf(pf),c) done(false).
From (TCS1.8, TCS1.5, 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

\[(TCS1.8) \ \forall p \in \mathbb{N}, \text{TFormula:} \]
\[n > 0 \land Ft1 \rightarrow*(n,p,sf,ef) \text{ done(false)} \Rightarrow \]

\[\text{next(TCS(Ft1,Ft2))} \rightarrow*(n,p,sf,ef) \text{ done(false)} \]

and prove

\[(TCS1.9) \ \forall p \in \mathbb{N}, \text{TFormula:} \]
\[n+1 > 0 \land Ft1 \rightarrow*(n+1,p,sf,ef) \text{ done(false)} \Rightarrow \]

\[\text{next(TCS(Ft1,Ft2))} \rightarrow*(n+1,p,sf,ef) \text{ done(false)} \].

To prove [TCS1.9], we take \( pf,Ft1f,Ft2f \) arbitrary but fixed, assume

\[(TCS1.10) \ n+1 > 0 \]
\[(TCS1.11) \ Ft1f \rightarrow*(n+1,pf,sf,ef) \text{ done(false)} \]

and prove

\[(TCS1.12) \ \text{next(TCS(Ft1f,Ft2f))} \rightarrow*(n+1,p,sf,ef) \text{ done(false)} \].

From (TCS1.11), by the definition of \( \rightarrow* \) without history, there exists \( Ft \in \text{TFormula} \) such that

\[(TCS1.13) \ Ft1f \rightarrow (pf,sf\downarrow pf, sf(pf),c) \text{ Ft} \]
\[(TCS1.14) \ Ft \rightarrow *(n,pf+1,sf,ef) \text{ done(false)} \]

where

\[(TCS1.15) \ c=(ef, \{(X,sf(ef(X)))| X \in \text{dom(ef)}\}) \].

We proceed by case distinction over \( Ft \).

Case 1. \( Ft=\text{next(f)} \) for some \( f \in \text{TFormulaCore} \)

From (TCS1.13), by the definition of \( \rightarrow \) for TCS, we get

\[(TCS1.16) \ \text{next(TCS(Ft1f,Ft2f))} \rightarrow (pf,sf\downarrow pf, sf(pf),c) \text{ next(TCS(Ft,Ft2f))} \]

Since \( Ft \) is a 'next' formula, we have

\[(TCS1.17) \ n > 0 \].

From (TCS1.17) and (TCS1.14), by the induction hypothesis (TCS1.8) we get
(TCS1.18) \text{next(TCS}(Pt,Ft2f) ) \rightarrow (n,pf+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. \( Ft=done(b) \) 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) \( b=false \).

Then from (TCS1.13) and (TCS1.20), by the definition of \( \rightarrow \) for TCS we get

(TCS1.21) \text{next(TCS}(Pt1f,Ft2f) ) \rightarrow (pf,sf\uparrow pf, sf(pf),c) \text{ done(false)}.

From (TCS1.14), (TCS1.19), and (TCS1.20), we have

(TCS1.22) done(false) \rightarrow (0,pf+1,sf,ef) \text{ done(false)}.

From (TCS1.19), (TSC1.15), (TSC1.21), (TCS1.22), by the definition of \( \rightarrow^* \) without history, we get [TCS1.12].

This finishes the proof of [TCS1].

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

Proof of [TCS2]
--------------

Recall

[TCS2] \( \forall s \in \text{Stream, } e \in \text{Environment, } p \in \mathbb{N}, Ft1,Ft2 \in \text{TFormula, } n1,n2 \in \mathbb{N}, b \in \text{Bool: } \)

\( n1>0 \land n2>0 \land Ft1 \rightarrow *(n1,p,s,e) \text{ done(true)} \land Ft2 \rightarrow *(n2,p,s,e) \text{ done(b)} \Rightarrow \)

\text{next(TCS}(Ft1,Ft2) ) \rightarrow *(\text{max}(n1,n2),p,s,e) \text{ done(b)}.

We take \( sf,ef,bf \) arbitrary but fixed and define

\( \Phi(n1) : \)

\( \forall p \in \mathbb{N}, Ft1,Ft2 \in \text{TFormula, } n2 \in \mathbb{N} : \)

\( n1>0 \land n2>0 \land Ft1 \rightarrow *(n1,p,sf,ef) \text{ done(true)} \land Ft2 \rightarrow *(n2,p,sf,ef) \text{ done(bf)} \Rightarrow \)

\text{next(TCS}(Ft1,Ft2) ) \rightarrow *(\text{max}(n1,n2),p,sf,ef) \text{ done(bf)}.

We need to prove \( \forall n1 \in \mathbb{N} : \Phi(n1). \) We use induction. Prove:

[TCS2.a] : \( \Phi(1) \)

[TCS2.b] \( \forall n1 \in \mathbb{N} : \Phi(n1) \Rightarrow \Phi(n1+1). \)

Proof of [TCS2.a]
-----------------

We need to prove

\( \forall n2,p \in \mathbb{N}, Ft1,Ft2 \in \text{TFormula :} \)
\[1 > 0 \land n_2 > 0 \land F_{t1} \rightarrow^* (1, p, s_f, e_f) \text{ done(true)} \land F_{t2} \rightarrow^* (n_2, p, s_f, e_f) \text{ done(bf)} \Rightarrow\]
\[\text{next}(TCS(F_{t1}, F_{t2})) \rightarrow^* (\max(1, n_2), p, s_f, e_f) \text{ done(bf)}.\]

We take \(n_2, p_f, F_{t1f}, F_{t2f}\) arbitrary but fixed. Assume

(TCS1.a.1) \(n_2 > 0\)
(TCS1.a.2) \(F_{t1f} \rightarrow^*(1, p_f, s_f, e_f) \text{ done(true)}\)
(TCS1.a.3) \(F_{t2f} \rightarrow^*(n_2, p_f, s_f, e_f) \text{ done(bf)}\)

and prove

[TCS1.a.4] \(\text{next}(TCS(F_{t1f}, F_{t2f})) \rightarrow^* (\max(1, n_2), p_f, s_f, e_f) \text{ done(bf)}.\)

From (TCS1.a.2), by the definition of \(\rightarrow^*\), we have for some \(F_t'\)

(TCS1.a.5) \(F_{t1f} \rightarrow (p_f, s_f, p_f, s_f(p_f), c) F_t'\)
(TCS1.a.6) \(F_t' \rightarrow^* (0, p_f + 1, s_f, e_f) \text{ done(true)}\)

where

(TCS1.a.7) \(c = (e_f, \{(X, s_f(e_f(X))) | X \in \text{dom}(e_f)\}).\)

From (TCS1.a.6), by the definition \(p_f \rightarrow^*\), we know

(TCS1.a.8) \(F_t' = \text{done(true)}\).

From (TCS1.a.5) and (TCS1.a.8) we have

(TCS1.a.9) \(F_{t1f} \rightarrow (p_f, s_f, p_f, s_f(p_f), c) \text{ done(true)}\).

From (TCS1.a.3), by the definition of \(\rightarrow^*\), we have for some \(F_t''\)

(TCS1.a.10) \(F_{t2f} \rightarrow (p_f, s_f, p_f, s_f(p_f), c) F_t''\)
(TCS1.a.11) \(F_t'' \rightarrow^* (n_2 - 1, p_f + 1, s_f, e_f) \text{ done(bf)}\),

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) \(\text{next}(TCS(F_{t1f}, F_{t2f})) \rightarrow (p_f, s_f, p_f, s_f(p_f), c) F_t''\).

From (TCS1.a.13), (TCS1.a.7), and (TCS1.a.11), by the definition of \(\rightarrow^*\), we have

(TCS1.a.14) \(\text{next}(TCS(F_{t1f}, F_{t2f})) \rightarrow (n_2, p_f, s_f, e_f) \text{ done(bf)}.\)

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.,
∀n2, p ∈ dsN, Ft1, Ft2 ∈ TFormula:

\[ n1 > 0 \land n2 > 0 \land \text{Ft1} \rightarrow \ast(n1, p, sf, ef) \text{done(true)} \land \text{Ft2} \rightarrow \ast(n2, p, sf, ef) \text{done(bf)} \]
\[ \Rightarrow \text{next(TCS(Ft1, Ft2))} \rightarrow \ast(\max(n1, n2), p, sf, ef) \text{done(bf)}. \]

and prove

[TCS2.b.2] ∀n2, p ∈ dsN, Ft1, Ft2 ∈ TFormula:

\[ n1+1 > 0 \land n2 > 0 \land \text{Ft1} \rightarrow \ast(n1+1, p, sf, ef) \text{done(true)} \land \text{Ft2} \rightarrow \ast(n2, p, sf, ef) \text{done(bf)} \]
\[ \Rightarrow \text{next(TCS(Ft1, Ft2))} \rightarrow \ast(\max(n1+1, n2), p, sf, ef) \text{done(bf)}. \]

To prove [TCS2.b.2], we take n2, pf, Ft1f, Ft2f arbitrary but fixed. Assume

(TCS2.b.3) n1+1 > 0
(TCS2.b.4) n2 > 0
(TCS2.b.5) Ft1f \rightarrow \ast(n1+1, pf, sf, ef) \text{done(true)}
(TCS2.b.6) Ft2f \rightarrow \ast(n2, pf, sf, ef) \text{done(bf)}

and prove

[TCS2.b.7] next(TCS(Ft1f, Ft2f)) \rightarrow \ast(\max(n1+1, n2), pf, sf, ef) \text{done(bf)}.

From (TCS2.b.5), by the definition of \(\rightarrow\ast\), we have for some Ft'

(TCS2.b.8) Ft1f \rightarrow (pf, sf, pf, sf(pf), c) \text{Ft'}
(TCS2.b.9) Ft' \rightarrow \ast(n1, pf+1, sf, ef) \text{done(true)}

where

(TCS2.b.10) c = (ef, \{(X, sf(ef(X))) | X \in \text{dom}(ef)\}).

From (TCS2.b.6), by the definition of \(\rightarrow\ast\), we have for some Ft''

(TCS2.b.11) Ft2f \rightarrow (pf, sf, pf, sf(pf), c) \text{Ft''}
(TCS2.b.12) Ft'' \rightarrow \ast(n2-1, pf+1, sf, ef) \text{done(bf)},

where c is defined as in (TCS2.b.10).

Case n1=0
-----------
In this case we have Ft'=done(true) and from (TCS2.b.8) we get

(TCS2.b.13) Ft1f \rightarrow (pf, sf, pf, sf(pf), c) \text{done(true)}.

From (TCS2.b.13) and (TCS2.b.11), by the definition of \(\rightarrow\) for TCS, we have

(TCS2.b.14) next(TCS(Ft1f, Ft2f)) \rightarrow (pf, sf, pf, sf(pf), c) \text{Ft''}.

From (TCS2.b.4), (TCS2.b.10), (TCS2.b.14), (TCS2.b.12) by the definition of \(\rightarrow\ast\), we get
(TCS2.b.15) \( \text{next(TCS(Ft1f,Ft2f))} \rightarrow (n2, pf, sf, ef) \text{ done}(bf) \).

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 \( Ft'=\text{next}(f') \) for some \( f' \in \text{TFormulaCore} \).
Therefore, from (TCS3.b.8), by the definition of \( \rightarrow \) for TCS we have

(TCS2.b.16) \( \text{next(TCS(Ft1f,Ft2f))} \rightarrow (pf, sf, pf, c) \text{ next(TCS(Ft',Ft2f))} \).

Since \( n2-1>0 \) and, hence, \( n2>0 \), from (TCS2.b.6) by the Shifting Lemma 7 we get

(TCS2.b.17) \( Ft2f \rightarrow (n2-1, pf+1, sf, ef) \text{ done}(bf) \).

From \( n1>0, n2-1>0 \), (TCS2.b.9), (TCS2.b.17), by the induction hypothesis (TCS2.b.1) we get

(TCS2.b.18) \( \text{next(TCS(Ft',Ft2f))} \rightarrow (\max(n1,n2-1), pf+1, sf, ef) \text{ done}(bf) \).

From \( \max(n1,n2-1)+1>0 \), (TCS2.b.10), (TCS2.b.16), (TCS2.b.18) we get

(TCS2.b.18) \( \text{next(TCS(Ft1f,Ft2f))} \rightarrow (\max(n1,n2-1)+1, pf, sf, ef) \text{ done}(bf) \).

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 \( Ft''=\text{done}(bf) \), which from (TCS2.b.12) gives

(TCS2.b.19) \( Ft2f \rightarrow (pf, sf, pf, c) \text{ done}(bf) \).

From (TCS2.b.5), by Lemma 2, we have

(TCS2.b.23) \( Ft1f \rightarrow l^*(n1+1, pf, sf, ef) \text{ done}(true) \).

From (TCS2.b.23), by the definition of \( \rightarrow l^* \), we obtain for some \( Ft0 \)

(TCS2.b.24) \( Ft1f \rightarrow l^*(n1, pf, sf, ef) Ft0 \)
(TCS2.b.25) \( Ft0 \rightarrow (pf+n1, s\downarrow(pf+n1), s(pf+n1), c) \text{ done}(true) \),

where \( c \) is defined as in (TCS2.b.10).

From (TCS2.b.19), by the Lemma 6, we have

(TCS2.b.26) \( Ft2f \rightarrow (pf+n1, sf\downarrow(pf+n1), sf(pf+n1), c) \text{ done}(bf) \).

From (TCS2.b.25) and (TCS2.b.26), by the definition of \( \rightarrow \) for TCS, we get
From (TCS2.b.24), by Lemma 2 we have

(TCS2.b.28) \( F_{t1f} \rightarrow^* (n_1, p_f, s_f, e_f) F_{t0} \).

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) \( next(TCS(F_{t1f}, F_{t2f})) \rightarrow l^*(n_1+1, p_f, s_f, e_f) done(bf) \)

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:

\( \Theta(n) : \forall F_{t1}, F_{t2} \in T_{Formula}, f \in T_{FormulaCore}, p \in \mathbb{N}, s \in Stream, e \in Environment: n>0 \Rightarrow F_{t1} \rightarrow^* (n, p, s, e) next(f) \Rightarrow next(TCS(F_{t1}, F_{t2})) \rightarrow^* (n, p, s, e) next(TCS(next(f), F_{t2})) \)
We need to prove $\forall n \in \mathbb{N}: \Theta(n)$. Induction:

[Prop.a] $\Theta(1)$
[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] $\text{next}(TCS(Ft_1f, Ft_2f)) \rightarrow^*(1, pf, sf, ef) \text{ next}(TCS(\text{next}(f_0), Ft_2f))$.

From (p1), by the definition of $\rightarrow^*$ there exists $Ft' \in TFormula$ such that

(p3) $Ft_1f \rightarrow (pf, sf, pf, sf(pf), c) \text{ Ft'}$

(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 $TCS$, we have

(p7) $\text{next}(TCS(Ft_1f, Ft_2f)) \rightarrow (pf, sf, pf, sf(pf), c) \text{ next}(TCS(\text{next}(f_0), Ft_2f))$.

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

(p8) $\text{next}(TCS(\text{next}(f_0), Ft_2f)) \rightarrow^*(0, pf+1, sf, ef) \text{ next}(TCS(\text{next}(f_0), Ft_2f))$.

From (p7), (p5), (p8), by the definition of $\rightarrow^*$ we get [p2].

Proof of [Prop.b]
---------------
We take $n$ arbitrary but fixed, assume

(p9) $\forall Ft_1, Ft_2 \in TFormula, f \in TFormulaCore, p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}:$ $n > 0 \Rightarrow$

$Ft_1 \rightarrow^*(n, p, s, e) \text{ next}(f) \Rightarrow$

$\text{next}(TCS(Ft_1, Ft_2)) \rightarrow^*(n, p, s, e) \text{ next}(TCS(\text{next}(f), Ft_2))$

and prove

[p10] $\forall Ft_1, Ft_2 \in TFormula, f \in TFormulaCore, p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}:$ $n + 1 > 0 \Rightarrow$

$Ft_1 \rightarrow^*(n+1, p, s, e) \text{ next}(f) \Rightarrow$

$\text{next}(TCS(Ft_1, Ft_2)) \rightarrow^*(n+1, p, s, e) \text{ next}(TCS(\text{next}(f), Ft_2))$.  

79
To prove (p10), we take $F_{t_1}f,F_{t_2}f,f_0,p_f,s_f,e_f$ arbitrary but fixed, assume

\[(p11)\ F_{t_1}f \rightarrow^{\ast}(n+1,p_f,s_f,e_f)\ next(f_0)\]

and prove

\[[p12]\ next(TCS(F_{t_1}f,F_{t_2}f)) \rightarrow^{\ast}(n+1,p_f,s_f,e_f)\ next(TCS(next(f_0),F_{t_2}f)).\]

**Case n>0**

-----

From (p11), by the definition of $\rightarrow^{\ast}$, we obtain for some $F_{t'} \in TFormula$

\[(p13)\ F_{t_1}f \rightarrow(p,f,s_f,p_f,s_f(c),c)\ F_{t'}\]

\[(p14)\ F_{t'} \rightarrow^{\ast}(n,p_f+1,s_f,e_f)\ next(f_0)\]

where

\[(p15)\ c=(e_f, \{(X,s_f(e_f(X))| X \in \text{dom}(e_f)})\].

Since $n>0$, from (p14) and the induction hypothesis (p9) we obtain

\[(p16)\ next(TCS(F_{t'},F_{t_2}f)) \rightarrow^{\ast}(n,p_f+1,s_f,e_f)\ next(TCS(next(f_0),F_{t_2}f)).\]

Moreover, $F_{t'}$ is a 'next' formula. Therefore, from (p13), by the definition of $\rightarrow$ for TCS we have

\[(p17)\ next(TCS(F_{t_1}f,F_{t_2}f)) \rightarrow(p,f,s_f,p_f,s_f(c),c)\ next(TCS(F_{t'},F_{t_2}f)).\]

From (p16), (p15), (p17), since $n+1>0$, by the definition of $\rightarrow^{\ast}$ 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.**

\[
\forall p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, F_{t_1}, F_{t_2} \in TFormula, n_1, n_2 \in \mathbb{N}:
\]
\[
n_1>0 \land F_{t_1} \rightarrow^{\ast}(n_1,p,s,e)\ done(false) \land F_{t_2} \rightarrow^{\ast}(n_2,p,s,e)\ done(true) \Rightarrow
\]
\[
next(TCP(F_{t_1},F_{t_2})) \rightarrow^{\ast}(n_1,p,s,e)\ done(false)
\]
\[
\land
\]
\[
n_1>0 \land n_2>0 \land F_{t_1} \rightarrow^{\ast}(n_1,p,s,e)\ done(true) \land F_{t_2} \rightarrow^{\ast}(n_2,p,s,e)\ done(true) \Rightarrow
\]
\[
next(TCP(F_{t_1},F_{t_2})) \rightarrow^{\ast}\min(n_1,n_2),p,s,e\ done(true)
\]
\[
\land
\]
\[
n_1>0 \land n_2>0 \land F_{t_1} \rightarrow^{\ast}(n_1,p,s,e)\ done(false) \land F_{t_2} \rightarrow^{\ast}(n_2,p,s,e)\ done(true) \Rightarrow
\]
\[
next(TCP(F_{t_1},F_{t_2})) \rightarrow^{\ast}\min(n_1,n_2),p,s,e\ done(true)
\]
\[
\land
\]
\[
n_1>0 \land n_2>0 \land F_{t_1} \rightarrow^{\ast}(n_1,p,s,e)\ done(true) \land F_{t_2} \rightarrow^{\ast}(n_2,p,s,e)\ done(false) \Rightarrow
\]
\[
next(TCP(F_{t_1},F_{t_2})) \rightarrow^{\ast}(n_2,p,s,e)\ done(false)
\]
Proof
--------------------------------------

We split the statement in four:

[TCP1] ∀p ∈ N, s ∈ Stream, e ∈ Environment, Ft1,Ft2 ∈ TFormula, n1,n2 ∈ N :
  n1>0 ∧ n2>0 ∧ Ft1 → *(n1,p,s,e) done(false) ∧ Ft2 → *(n2,p,s,e) done(true) ⇒
  next(TCP(Ft1,Ft2)) → *(n1,p,s,e) done(false)

[TCP2] ∀p ∈ N, s ∈ Stream, e ∈ Environment, Ft1,Ft2 ∈ TFormula, n1,n2 ∈ N :
  n1>0 ∧ n2>0 ∧ Ft1 → *(n1,p,s,e) done(false) ∧
  Ft2 → *(n2,p,s,e) done(false) ⇒
  next(TCP(Ft1,Ft2)) → *(min(n1,n2),p,s,e) done(false)

[TCP3] ∀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(true) ⇒
  next(TCP(Ft1,Ft2)) → *(max(n1,n2),p,s,e) done(true).

[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) ⇒
  next(TCP(Ft1,Ft2)) → *(n2,p,s,e) done(false).

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

Proof of [TCP1]  
----------------

We take sf,ef arbitrary but fixed and define

Φ(n) :⇔
  ∀p ∈ N, s ∈ Stream, e ∈ Environment, Ft1,Ft2 ∈ TFormula, n1,n2 ∈ N :
  n1>0 ∧ n2>0 ∧ Ft1 → *(n1,p,s,e) done(false) ∧ Ft2 → *(n2,p,s,e) done(true) ⇒
  next(TCP(Ft1,Ft2)) → *(n1,p,s,e) done(false)

We prove ∀n1 ∈ N: Φ(n1) by induction over n1. For n1=0 the formula is trivially true.

We start the induction from 1. Prove:

[TCP1.a] Φ(1) and
[TCP1.b] ∀n1 ∈ N: Φ(n1) ⇒ Φ(n1+1)

Proof of [TCP1.a]  
----------------

We take pf,Ft1f,Ft2f,n2 arbitrary but fixed. 1>0 is satisfied. Assume

(TCP1.1) n2>0
(TCP1.2) Ft1f → *(1,pf,sf,ef) done(false).
(TCP1.3) Ft2f → *(n2,p,s,e) done(true).

We want to prove
[TCP1.4] $\text{next(TCP(Ft1f,Ft2f))} \rightarrow^*(1,pf,\text{s}f,ef) \text{ done(false)}$.

From (TCP1.2), by the definition of $\rightarrow^*$ without history, there exists $Ft \in \text{TFormula}$ such that

(TCP1.5) $\text{Ft1f} \rightarrow(p,\text{s}f,\text{pf},\text{sf}(\text{pf}),c) \text{ Ft}$ and
(TCP1.6) $\text{Ft} \rightarrow^*(0,pf+1,\text{s}f,\text{ef}) \text{ done(false)}$

where

(TCP1.7) $c=(ef, \{(X,\text{sf}(ef(X)))| X \in \text{dom(ef)}\})$.

From (TCP1.6), by the definition of $\rightarrow^*$ without history, we get

(TCP1.8') $\text{Ft}=\text{done(false)}$.

which from (TCP1.5) gives

(TCP1.9') $\text{Ft1f} \rightarrow(p,\text{s}f,\text{pf},\text{sf}(\text{pf}),c) \text{ done(false)}$ and

From (TCP1.9') and (TCP1.3), by the definition of $\rightarrow$ for TCP, we get

(TCP1.10') $\text{next(TCP(Ft1f,Ft2f))} \rightarrow(p,\text{s}f,\text{pf},\text{sf}(\text{pf}),c) \text{ done(false)}$.

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) $\forall p \in \mathbb{N}, \text{Ft1,Ft2} \in \text{TFormula}, n_2 \in \mathbb{N}$ : $n_1>0 \land n_2>0 \land 
\text{Ft1} \rightarrow^*(n_1,p,s,e) \text{ done(false)} \land \text{Ft2} \rightarrow^*(n_2,p,s,e) \text{ done(true)} \Rightarrow 
\text{next(TCP(Ft1,Ft2))} \rightarrow^*(n_1,p,s,e) \text{ done(false)}$

and prove

[TCP1.9] $\forall p \in \mathbb{N}, \text{Ft1,Ft2} \in \text{TFormula}, n_2 \in \mathbb{N}$ : $n_1>0 \land n_2>0 \land 
\text{Ft1} \rightarrow^*(n_1+1,p,s,e) \text{ done(false)} \land \text{Ft2} \rightarrow^*(n_2,p,s,e) \text{ done(true)} \Rightarrow 
\text{next(TCP(Ft1,Ft2))} \rightarrow^*(n_1+1,p,s,e) \text{ done(false)}$

To prove [TCP1.9], we take $p,Ft1f,Ft2f,n2$ arbitrary but fixed, assume

(TCP1.10) $n+1>0$
(TCP1.11) $n_2>0$
(TCP1.12) $\text{Ft1f} \rightarrow^*(n_1+1,pf,\text{s}f,ef) \text{ done(false)}$
(TCP1.13) $\text{Ft2f} \rightarrow^*(n_2,pf,\text{s}f,ef) \text{ done(true)}$

and prove

[TCP1.14] $\text{next(TCP(Ft1f,Ft2f))} \rightarrow^*(n_1+1,pf,\text{s}f,ef) \text{ done(false)}$. 

82
From (TCP1.12), by (TCP1.10) and the definition of →* without history, there exists $Ft' \in T_{\text{Formula}}$ such that

\[(TCP1.15) \quad Ft_1f \rightarrow (pf, sf \downarrow pf, sf(pf), c) \quad Ft'\]
\[(TCP1.16) \quad Ft' \rightarrow */(n_1, pf+1, sf, ef) \text{ done(false)}\]

where

\[(TCP1.17) \quad c= (ef, \{(X, sf(ef(X))) \mid X \in \text{dom}(ef)\}).\]

From (TCP1.13), by (TCP1.11) and the definition of →* without history, there exists $Ft'' \in T_{\text{Formula}}$ such that

\[(TCP1.18) \quad Ft_2f \rightarrow (pf, sf \downarrow pf, sf(pf), c) \quad Ft''\]
\[(TCP1.19) \quad Ft'' \rightarrow */(n_2-1, pf+1, sf, ef) \text{ done(true)}\]

where $c$ is defined as in (TCP1.17).

Case $n_1>0, n_2-1>0$

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

In this case $Ft'=\text{next}(f'), Ft''=\text{next}(f'')$ for some $f', f'' \in T_{\text{FormulaCore}}$. Therefore, from (TCP1.15,TCP1.18), by the definition of $\rightarrow$ for TCP we have

\[(TCP1.20) \quad \text{next(TCP}(Ft_1f, Ft_2f)) \rightarrow (pf, sf \downarrow pf, sf(pf), c) \text{ next(TCP}(Ft', Ft'')).\]

From $n_1>0, n_2-1>0$, (TCP1.16,TCP1.19), by the induction hypothesis (TCP1.8) we have

\[(TCP1.21) \quad \text{next(TCP}(Ft', Ft'')) \rightarrow */(n_1, pf+1, sf, ef) \text{ done(false)}.\]

From $n_1+1>0$, (TCP1.17), (TCP1.20), (TCP1.21), by the definition of $\rightarrow*$ we have

\[(TCP1.22) \quad \text{next(TCP}(Ft_1f, Ft_2f)) \rightarrow */(n_1+1, pf, sf, ef) \text{ done(false)}\]

which is [TCP1.14]

Case $n_1>0, n_2-1=0$

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

In this case $Ft'=\text{next}(f')$ for some $f' \in T_{\text{FormulaCore}}$ and, from (TCP1.18)

\[(TCP1.23) \quad Ft_2f \rightarrow (pf, sf \downarrow pf, sf(pf), c) \text{ done(true)}.$

Therefore, from (TCP1.15,TCP1.23), by the definition of $\rightarrow$ for TCP we have

\[(TCP1.24) \quad \text{next(TCP}(Ft_1f, Ft_2f)) \rightarrow (pf, sf \downarrow pf, sf(pf), c) \quad Ft'\]

From $n_1+1>0$, (TCP1.17), (TCP1.24), (TCP1.16), by the definition of $\rightarrow*$ we get [TCP1.14].

Case $n_1=0$

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

83
In this case \( F_{t''} = \text{next}(f'') \) for some \( f'' \in \text{TFormulaCore} \) and, from (TCP1.15) (TCP1.25) \( F_{t1f} \rightarrow (pf, sf \downarrow pf, sf(pf), c) \) done(false).

From (TCP1.25) by the definition of \( \rightarrow \) for TCP we have (TCP1.26) \( \text{next}(TCP(F_{t1f}, F_{t2f})) \rightarrow (pf, sf \downarrow pf, sf(pf), c) \) 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] \( \forall p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, F_{t1}, F_{t2} \in \text{TFormula}, n1, n2 \in \mathbb{N} : n1 > 0 \land n2 > 0 \land F_{t1} \rightarrow^* (n1, p, s, e) \) done(false) \land 
\( F_{t2} \rightarrow^* (n2, p, s, e) \) done(false) \implies
\( \text{next}(TCP(F_{t1}, F_{t2})) \rightarrow^* (\min(n1, n2), p, s, e) \) done(false)

Proof
-----

We take \( sf, ef \) arbitrary but fixed and define

\( \Phi(n) : \leftrightarrow \)

\( \forall p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, F_{t1}, F_{t2} \in \text{TFormula}, n1, n2 \in \mathbb{N} : n1 > 0 \land n2 > 0 \land F_{t1} \rightarrow^* (n1, p, s, e) \) done(false) \land 
\( F_{t2} \rightarrow^* (n2, p, s, e) \) done(false) \implies
\( \text{next}(TCP(F_{t1}, F_{t2})) \rightarrow^* (\min(n1, n2), p, s, e) \) done(false)

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] \( \forall n1 \in \mathbb{N} : \Phi(n1) \implies \Phi(n1+1) \)

Proof of [TCP2.a]
-----------------

We take \( pf, F_{t1f}, F_{t2f}, n2 \) arbitrary but fixed. 1 > 0 is satisfied. Assume

(TCP2.1) \( n2 > 0 \)
(TCP2.2) \( F_{t1f} \rightarrow^* (1, pf, sf, ef) \) done(false).
(TCP2.3) \( F_{t2f} \rightarrow^* (n2, p, s, e) \) done(false).

We want to prove
[TCP2.4] \( \text{next(TCP}(Ft1f,Ft2f)) \rightarrow *(\min(1,n2),pf,sf,ef) \text{ done(false)} \).

From (TCP2.2), by the definition of \( \rightarrow^* \) without history, there exists \( Ft \in TFormula \) such that

(TCP2.5) \( Ft1f \rightarrow (p,sf,\lceil pf \rfloor, sf(pf), c) \) \( Ft \) and
(TCP2.6) \( Ft \rightarrow *(0,pf+1,sf,ef) \text{ done(false)} \)

where

(TCP2.7) \( c = \{ (ef, \{(X, sf(ef(X))) \mid X \in \text{dom}(ef) \}) \} \).

From (TCP2.6), by the definition of \( \rightarrow^* \) without history, we get

(TCP2.8) \( Ft = \text{done(false)} \).

which from (TCP2.5) gives

(TCP2.9) \( Ft1f \rightarrow (p,sf,\lceil pf \rfloor, sf(pf), c) \text{ done(false)} \).

From (TCP2.9) and (TCP2.3), by the definition of \( \rightarrow \) for TCP, we get

(TCP2.10) \( \text{next(TCP}(Ft1f,Ft2f)) \rightarrow (p,sf,\lceil pf \rfloor, sf(pf), c) \text{ done(false)} \).

From (TCP2.10, TCP2.6, TCP2.8, TCP2.7), by the definition of \( \rightarrow^* \) without history, we get \( \text{next(TCP}(Ft1f,Ft2f)) \rightarrow *(1,pf,ef) \text{ done(false)} \), but since by (TCP2.1) we have \( 1 = \min(1,n2) \), we actually proved [TCP2.4].

Proof of [TCP2.b]
-----------------
We take \( n1 \) arbitrary but fixed, assume

(TCP2.8) \( \forall p \in \mathbb{N}, Ft1,Ft2 \in TFormula, n2 \in \mathbb{N} : \)
\( n1 > 0 \land n2 > 0 \land \)
\( Ft1 \rightarrow *(n1,p,s,e) \text{ done(false)} \land Ft2 \rightarrow *(n2,p,s,e) \text{ done(false)} \Rightarrow \)
\( \text{next(TCP}(Ft1,Ft2)) \rightarrow *(\min(n1,n2),p,s,e) \text{ done(false)} \)

and prove

[TCP2.9] \( \forall p \in \mathbb{N}, Ft1,Ft2 \in TFormula, n2 \in \mathbb{N} : \)
\( n1+1 > 0 \land n2 > 0 \land \)
\( Ft1 \rightarrow *(n1+1,p,s,e) \text{ done(false)} \land Ft2 \rightarrow *(n2,p,s,e) \text{ done(false)} \Rightarrow \)
\( \text{next(TCP}(Ft1,Ft2)) \rightarrow *(\min(n1+1,n2),p,s,e) \text{ done(false)} \).

To prove [TCP2.9], we take \( pf,Ft1f,Ft2f,n2 \) arbitrary but fixed, assume

(TCP2.10) \( n+1 > 0 \)
(TCP2.11) \( n2 > 0 \)
(TCP2.12) \( Ft1f \rightarrow *(n1+1,pf,sf,ef) \text{ done(false)} \)
(TCP2.13) \( Ft2f \rightarrow *(n2,pf,ef) \text{ done(false)} \)

and prove

[TCP2.14] \( \text{next(TCP}(Ft1f,Ft2f)) \rightarrow *(\min(n1+1,n2),pf,ef) \text{ done(false)} \).
From (TCP2.12), by (TCP2.10) and the definition of $\to^*$ without history, there exists $F_{t'} \in TFormula$ such that

\[(TCP2.15)\] $F_{t1f} \to (pf, sf \downarrow pf, sf(pf), c) F_{t'}$

\[(TCP2.16)\] $F_{t'} \to^* (n_1, pf+1, sf, ef) done(false)$

where

\[(TCP2.17)\] $c = (ef, \{(X, sf(ef(X))) \mid X \in \text{dom}(ef)\})$.

From (TCP2.13), by (TCP2.11) and the definition of $\to^*$ without history, there exists $F_{t''} \in TFormula$ such that

\[(TCP2.18)\] $F_{t2f} \to (pf, sf \downarrow pf, sf(pf), c) F_{t''}$

\[(TCP2.19)\] $F_{t''} \to^* (n_2-1, pf+1, sf, ef) done(false)$

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 $f', f'' \in TFormulaCore$. Therefore, from (TCP2.15, TCP2.18), by the definition of $\to$ for TCP we have

\[(TCP2.20)\] $\text{next}(TCP(F_{t1f}, F_{t2f})) \to (pf, sf \downarrow pf, sf(pf), c) \text{next}(TCP(F_{t'}, F_{t''}))$.

From $n_1 > 0$, $n_2 - 1 > 0$, (TCP2.16, TCP2.19), by the induction hypothesis (TCP2.8) we have

\[(TCP2.21)\] $\text{next}(TCP(F_{t'}, F_{t''})) \to^* (\min(n_1, n_2 - 1), pf + 1, sf, ef) done(false)$.

From $n_1 + 1 > 0$, (TCP2.17), (TCP2.20), (TCP2.21), by the definition of $\to^*$ we have

\[(TCP2.22)\] $\text{next}(TCP(F_{t1f}, F_{t2f})) \to^* (\min(n_1, n_2 - 1) + 1, pf, sf, ef) done(false)$

which is [TCP2.14]

Case $n_1 > 0$, $n_2 - 1 = 0$
-------------------------------------
In this case $F_{t'} = \text{next}(f')$ for some $f' \in TFormulaCore$ and, from (TCP2.18) we have

\[(TCP2.23)\] $F_{t2f} \to (pf, sf \downarrow pf, sf(pf), c) done(false)$.

Therefore, from (TCP2.15, TCP2.23), by the definition of $\to$ for TCP we have

\[(TCP2.24)\] $\text{next}(TCP(F_{t1f}, F_{t2f})) \to (pf, sf \downarrow pf, sf(pf), c) done(false)$

From $1 > 0$, (TCP2.17), (TCP2.24), (TCP2.19), by the definition of $\to^*$ we get

\[(TCP2.25)\] $\text{next}(TCP(F_{t1f}, F_{t2f})) \to^* (1, pf, sf, ef) done(false)$

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 \( \text{Ft''} = \text{next(f'')} \) for some \( f'\in\text{FormulaCore} \) and, from (TCP2.15) we have

(TCP2.26) \( \text{Ft1f} \rightarrow (pf, sf\downarrow pf, sf(pf),c) \) done(false).

From (TCP2.26) by the definition of \( \rightarrow \) for TCP we have

(TCP2.27) \( \text{next(TCP(Ft1,Ft2))} \rightarrow (pf, sf\downarrow pf, sf(pf),c) \) done(false).

From \( 1>0 \), (TCP2.17), (TCP2.27), (TCP2.16), by the definition of \( \rightarrow * \) we get

(TCP2.28) \( \text{next(TCP(Ft1,Ft2))} \rightarrow *(1,pf, sf, ef) \) done(false).

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] \( \forall p\in\mathbb{N}, s\in\text{Stream}, e\in\text{Environment}, Ft1,Ft2\in\text{Formula}, n1,n2\in\mathbb{N}, b\in\text{Bool} : \)
\( n1>0 \land n2>0 \land \)
\( Ft1 \rightarrow *(n1,p,s,e) \text{ done(true)} \land Ft2 \rightarrow *(n2,p,s,e) \text{ done(true)} \Rightarrow \)
\( \text{next(TCP(Ft1,Ft2))} \rightarrow *(\max(n1,n2),p,s,e) \text{ done(true)}. \)

Proof
-----

We take \( sf,ef \) arbitrary but fixed and define

\( \Phi(n1) : \)
\( \forall p\in ds\mathbb{N}, Ft1,Ft2\in\text{Formula}, n2\in\mathbb{N} : \)
\( n1>0 \land n2>0 \land \)
\( Ft1 \rightarrow *(n1,p,sf,ef) \text{ done(true)} \land Ft2 \rightarrow *(n2,p,sf,ef) \text{ done(true)} \Rightarrow \)
\( \text{next(TCP(Ft1,Ft2))} \rightarrow *(\max(n1,n2),p,sf,ef) \text{ done(true)}. \)

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

\( \forall n2,p\in ds\mathbb{N}, Ft1,Ft2\in\text{Formula} : \)
\( 1>0 \land n2>0 \land \)
\( Ft1 \rightarrow *(1,p,sf,ef) \text{ done(true)} \land Ft2 \rightarrow *(n2,p,sf,ef) \text{ done(true)} \Rightarrow \)
\( \text{next(TCP(Ft1,Ft2))} \rightarrow *(\max(1,n2),p,sf,ef) \text{ done(true)}. \)

We take \( n2,pf,Ft1f,Ft2f \) arbitrary but fixed. Assume
(TCP3.a.1) n2>0
(TCP3.a.2) Ft1f →*(1,pf,sf,ef) done(true)
(TCP3.a.3) Ft2f →*(n2,pf,sf,ef) done(true)

and prove

[TCP3.a.4] next(TCP(Ft1f,Ft2f)) →*(max(1,n2),pf,sf,ef) done(true).

From (TCP3.a.2), by the definition of →*, we have for some Ft'

(TCP3.a.5) Ft1f →(pf,sf↓pf,sf(pf),c) Ft'
(TCP3.a.6) Ft' →*(0,pf+1,sf,ef) done(true)

where

(TCP3.a.7) c=(ef, \{ (X, sf(ef(X))) | X \in \text{dom}(ef) \}).

From (TCP3.a.6), by the definition pf →*, we know

(TCP3.a.8) Ft'=done(true).

From (TCP3.a.5) and (TCP3.a.8) we have

(TCP3.a.9) Ft1f →(pf,sf↓pf,sf(pf),c) done(true).

From (TCP3.a.3), by the definition of →*, we have for some Ft''

(TCP3.a.10) Ft2f →(pf,sf↓pf,sf(pf),c) Ft''
(TCP3.a.11) Ft'' →*(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 → for TCP, we have

(TCP3.a.13) next(TCP(Ft1f,Ft2f)) → (pf,sf↓pf,sf(pf),c) Ft''.

From (TCP3.a.13), (TCP3.a.7), and (TCP3.a.11), by the definition of →*, we have

(TCP3.a.14) next(TCP(Ft1f,Ft2f)) →* (n2,pf,sf,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) \( \forall n2,p \in \text{dsN}, \text{Ft1,Ft2} \in \text{TFormula} : \)
\( n1>0 \land n2>0 \land \text{Ft1} \rightarrow *(n1,p,sf,ef) \text{ done(true)} \land \)
\( \text{Ft2} \rightarrow *(n2,p,sf,ef) \text{ done(true)} \)
⇒
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, p, Ft1f, Ft2f arbitrary but fixed. Assume

TCP3.b.3 n1 + 1 > 0
TCP3.b.4 n2 > 0
TCP3.b.5 Ft1f → *(n1 + 1, p, sf, ef) done(true)
TCP3.b.6 Ft2f → *(n2, p, sf, ef) done(true)

and prove

TCP3.b.7 next(TCP(Ft1f, Ft2f)) → *(max(n1 + 1, n2), p, 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, sf, 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 \(n_1=0\), we have \(n_2 = \max(1, n_2) = \max(n_1+1, n_2)\). Hence, (TCP3.b.15) proves \([TCP3.b.7]\).

Case \(n_1>0\), \(n_2-1>0\)

In this case \(F_t' = \text{next}(f')\), \(F_t'' = \text{next}(f'')\) for some \(f', f'' \in T\text{FormulaCore}\).

Therefore, from \((TCP3.b.8, TCP3.b.11)\), by the definition of \(\rightarrow\) for TCP we have

\[(TCP3.b.16) \quad \text{next}(TCP(F_t f_1, F_t f_2)) \rightarrow (pf, sf \downarrow pf, sf(pf), c) \text{ next}(TCP(F_t', F_t'')).\]

From \(n_1>0, n_2-1>0\), \((b9, b12)\), by the induction hypothesis \((TCP3.b.1)\) we have

\[(TCP3.b.17) \quad \text{next}(TCP(F_t', F_t'')) \rightarrow *\max(n_1, n_2-1), pf+1, sf, ef) \text{ done(true)}\]

From \(n_1+1>0\), \((TCP3.b.10)\), \((TCP3.b.16)\), \((TCP3.b.17)\), by the definition of \(\rightarrow *\)

we have

\[(TCP3.b.18) \quad \text{next}(TCP(F_t f_1, F_t f_2)) \rightarrow *\max(n_1, n_2-1)+1, pf, sf, ef) \text{ done(true)}\]

which is \([TCP3.b.7]\)

Case \(n_1>0\), \(n_2-1=0\)

In this case \(F_t' = \text{next}(f')\) for some \(f' \in T\text{FormulaCore}\). From \((TCP3.b.11)\) we have

\[(TCP3.b.19) \quad F_t f \rightarrow (pf, sf \downarrow pf, sf(pf), c) \text{ done(true)}\]

From \((TCP3.b.8, TCP3.b.19)\), by the definition of \(\rightarrow\) for TCP we have

\[(TCP3.b.20) \quad \text{next}(TCP(F_t f_1, F_t f_2)) \rightarrow (pf, sf \downarrow pf, sf(pf), c) \text{ F}_t'\]

From \(n_1+1>0\), \((TCP3.b.10)\), \((TCP3.b.20)\), \((TCP3.b.9)\), by the definition of \(\rightarrow *\)

we get

\[(TCP3.b.21) \quad \text{next}(TCP(F_t f_1, F_t f_2)) \rightarrow *\max(n_1+1, pf, sf, ef) \text{ done(true)}\]

But by \(n_1>0\) and \(n_2=1\) we have \(n_1+1 = \max(n_1+1, n_2)\). 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] \quad \forall p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, F_t_1, F_t_2 \in T\text{Formula}, n_1, n_2 \in \mathbb{N} :\]

\(n_1>0 \land n_2>0 \land F_t_1 \rightarrow *\max(n_1, p, s, e) \text{ done(true)} \land F_t_2 \rightarrow *\max(n_2, p, s, e) \text{ done(false)} \Rightarrow\)
\[ \text{next(TCP(Ft1,Ft2))} \rightarrow *\text{(n2,p,s,e) done(false)}. \]

Proof
-----

We take \(sf,ef,bf\) arbitrary but fixed and define

\[ \Phi(n1) :\]
\[ \forall p \in dsN, \text{Ft1,Ft2} \in T\text{Formula}, n2 \in \mathbb{N} : \]
\[ n1>0 \land n2>0 \land \]
\[ \text{Ft1} \rightarrow *(n1,p,ef,ef) \text{ done(true)} \land \text{Ft2} \rightarrow *(n2,p,ef,ef) \text{ done(false)} \Rightarrow \]
\[ \text{next(TCP(Ft1,Ft2))} \rightarrow *(n2,p,ef,ef) \text{ done(false)}. \]

We need to prove \( \forall n1 \in \mathbb{N}: \Phi(n1) \). We use induction. Prove:

[TCP4.a] \( \forall n2 \in \mathbb{N}: \Phi(1) \)

[TCP4.b] \( \forall n1 \in \mathbb{N}: \Phi(n1) \Rightarrow \Phi(n1+1) \).

Proof of [TCP4.a]
-----------------

We need to prove

\[ \forall n2,p \in dsN, \text{Ft1,Ft2} \in T\text{Formula} : \]
\[ 1>0 \land n2>0 \land \]
\[ \text{Ft1} \rightarrow *(1,p,ef,ef) \text{ done(true)} \land \text{Ft2} \rightarrow *(n2,p,ef,ef) \text{ done(false)} \Rightarrow \]
\[ \text{next(TCP(Ft1,Ft2))} \rightarrow *(n2,p,ef,ef) \text{ done(false)}. \]

We take \( n2,p,Ft1f,Ft2f \) arbitrary but fixed. Assume

(TCP4.a.1) \( n2>0 \)
(TCP4.a.2) \( \text{Ft1f} \rightarrow *(1,p,f,sf,ef) \text{ done(true)} \)
(TCP4.a.3) \( \text{Ft2f} \rightarrow *(n2,p,ef,ef) \text{ done(false)} \)

and prove

[TCP4.a.4] \( \text{next(TCP(Ft1f,Ft2f))} \rightarrow *(n2,p,ef,ef) \text{ done(false)}. \)

From (TCP4.a.2), by the definition of \( \rightarrow * \), we have for some \( \text{Ft'} \)

(TCP4.a.5) \( \text{Ft1f} \rightarrow *(pf,ef,pf,ef) \text{ done(true)} \)
(TCP4.a.6) \( \text{Ft'} \rightarrow *(0,pf+1,ef,ef) \text{ done(true)} \)

where

(TCP4.a.7) \( c=(ef, \{(X,ef(X))| X \in \text{dom}(ef)} \).)

From (TCP4.a.6), by the definition \( pf \rightarrow * \), we know

(TCP4.a.8) \( \text{Ft'}=\text{done(true)}. \)

From (TCP4.a.5) and (TCP4.a.8) we have

(TCP4.a.9) \( \text{Ft1f} \rightarrow *(pf,ef,pf,ef) \text{ done(true)}. \)
From (TCP4.a.3), by the definition of $\rightarrow^*$, we have for some $Ft''$

(TCP4.a.10) $Ft^f \rightarrow (pf, sf \downarrow pf, sf(pf), c) Ft''$
(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) $\text{next(TCP}(Ft^f_1, Ft^f_2)) \rightarrow (pf, sf \downarrow pf, sf(pf), c) Ft''.$

From (TCP4.a.13), (TCP4.a.7), and (TCP4.a.11), by the definition of $\rightarrow^*$, we have

(TCP4.a.14) $\text{next(TCP}(Ft^f_1, Ft^f_2)) \rightarrow^* (n2, pf, sf, ef) \text{ done(false)}.$

(TCP4.a.14) is [TCP4.a.4].

This finishes the proof of [TCP4.a].

Proof of [TCP4.b]
-----------------

We take $n_1$ arbitrary but fixed. Assume $\Phi(n_1)$, i.e.,

(TCP4.b.1) $\forall n_2, p \in \text{dsN}, Ft_1, Ft_2 \in \text{TFormula} :$

\[ n_1 > 0 \land n_2 > 0 \land Ft_1 \rightarrow^*(n_1, p, sf, ef) \text{ done(true)} \land \]
\[ Ft_2 \rightarrow^*(n_2, p, sf, ef) \text{ done(false)} \]
\[ \Rightarrow \]
\[ \text{next(TCP}(Ft_1, Ft_2)) \rightarrow^*(n_2, p, sf, ef) \text{ done(false)}. \]

and prove

[TCP4.b.2] $\forall n_2, p \in \text{dsN}, Ft_1, Ft_2 \in \text{TFormula} :$

\[ n_1 + 1 > 0 \land n_2 > 0 \land Ft_1 \rightarrow^*(n_1+1, p, sf, ef) \text{ done(true)} \land \]
\[ Ft_2 \rightarrow^*(n_2, p, sf, ef) \text{ done(bf)} \]
\[ \Rightarrow \]
\[ \text{next(TCP}(Ft_1, Ft_2)) \rightarrow^*(false, p, sf, ef) \text{ done(false)}. \]

To prove [TCP4.b.2], we take $n_2, pf, Ft^f_1, Ft^f_2$ arbitrary but fixed. Assume

(TCP4.b.3) $n_1 + 1 > 0$
(TCP4.b.4) $n_2 > 0$
(TCP4.b.5) $Ft^f_1 \rightarrow^*(n_1+1, pf, sf, ef) \text{ done(true)}$
(TCP4.b.6) $Ft^f_2 \rightarrow^*(n_2, pf, sf, ef) \text{ done(false)}$

and prove

[TCP4.b.7] $\text{next(TCP}(Ft^f_1, Ft^f_2)) \rightarrow^*(n_2, pf, sf, ef) \text{ done(false)}.$

From (TCP4.b.5), by the definition of $\rightarrow^*$, we have for some $Ft'$

(TCP4.b.8) $Ft^f \rightarrow (pf, sf \downarrow pf, sf(pf), c) Ft'$

92
(TCP4.b.9) \( F_{t'} \rightarrow^\ast(n_1,pf+1,\text{sf},\text{ef}) \text{ done(true)} \)

where

(TCP4.b.10) \( c=(\text{ef}, \{(X, \text{sf}(\text{ef}(X))) | X \in \text{dom(ef)}\}) \).

From (TCP4.b.6), by the definition of \( \rightarrow^\ast \), we have for some \( F_{t''} \)

(TCP4.b.11) \( F_{t2f} \rightarrow (pf, \text{sf} \downarrow pf, \text{sf}(pf), c) F_{t''} \)

(TCP4.b.12) \( F_{t''} \rightarrow^\ast(n_2-1,pf+1,\text{sf},\text{ef}) \text{ done(false)} \)

where \( c \) is defined as in (TCP4.b.10).

Case 1. \( n_1=0 \)
---------------
In this case we have \( F_{t'}=\text{done(true)} \) and from (TCP4.b.8) we get

(TCP4.b.13) \( F_{t1f} \rightarrow (pf, \text{sf} \downarrow pf, \text{sf}(pf), c) \text{ done(true)} \).

From (TCP4.b.13) and (TCP4.b.11), by the definition of \( \rightarrow \) for TCP, we have

(TCP4.b.14) \( \text{next(TCP}(F_{t1f},F_{t2f}) \rightarrow (pf, \text{sf} \downarrow pf, \text{sf}(pf), c) F_{t''} \).

From (TCP4.b.4), (TCP4.b.10), (TCP4.b.14), (TCP4.b.12) by the definition of \( \rightarrow^\ast \), we get

(TCP4.b.15) \( \text{next(TCP}(F_{t1f},F_{t2f}) \rightarrow^\ast(n_2,pf,\text{sf},\text{ef}) \text{ done(false)} \).

Hence, (TCP4.b.15) proves \([TCP4.b.7]\).

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 (TCP4.b.8,TCP4.b.11), by the definition of \( \rightarrow \) for TCP we have

(TCP4.b.16) \( \text{next(TCP}(F_{t1f},F_{t2f}) \rightarrow (pf, \text{sf} \downarrow pf, \text{sf}(pf), c) \text{ next(TCP}(F_{t'},F_{t''})) \).

From \( n_1>0, n_2-1>0, (b9,b12) \), by the induction hypothesis (TCP4.b.1) we have

(TCP4.b.17) \( \text{next(TCP}(F_{t'},F_{t''})) \rightarrow^\ast(n_2-1,pf+1,\text{sf},\text{ef}) \text{ done(false)} \).

From (TCP4.b.4), (TCP4.b.10), (TCP4.b.16), (TCP4.b.17), by the definition of \( \rightarrow^\ast \) we have

(TCP4.b.18) \( \text{next(TCP}(F_{t1f},F_{t2f}) \rightarrow^\ast(n_2,pf,\text{sf},\text{ef}) \text{ done(false)} \)

which is \([TCP4.b.7]\)

Case \( n_1>0, n_2-1=0 \)
---------------
In this case \( F_{t'}=\text{next}(f') \) for some \( f' \in \text{TFormulaCore} \). From (TCP4.b.11) we have

(TCP4.b.19) \( F_{t2f} \rightarrow (pf, \text{sf} \downarrow pf, \text{sf}(pf), c) \text{ done(false)} \).
From (TCP4.b.8,TCP4.b.19), by the definition of \( \rightarrow \) for TCP we have

\[(TCP4.b.23) \text{ next}(TCP(Ftf1,Ftf2)) \rightarrow (pf, sf \downarrow pf, sf(pf), c) \text{ done}(false).\]

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

\[ \forall F \in \text{Formula}, X \in \text{Variable}, B1, B2 \in \text{Bound}: \]
\[ R(F) \Rightarrow R(\text{forall } X \text{ in } B1..B2: \ F) \]

where

\[ R(F) :\]
\[ \forall \text{re} \in \text{RangeEnv}, e \in \text{Environment}, s \in \text{Stream}, d \in \mathbb{N}_\infty, h \in \mathbb{N}, p \in \mathbb{N}: \]
\[ \vdash (\text{re } \vdash F: (h,d)) \land d \in \mathbb{N} \land \text{dom}(e) = \text{dom}(\text{re}) \land \]
\[ (\forall Y \in \text{dom}(e): \text{re}(Y).1 +i p \leq i e(Y) \leq i \text{re}(Y).2 +i p) \Rightarrow \]
\[ (\exists b \in \text{Bool} \exists d' \in \mathbb{N}: \]
\[ d' \leq d+1 \land \vdash (F) \rightarrow^{*(d',p,s,e)} \text{done}(b) \]

PROOF:

-----

We take \( F, X, B1, B2 \) arbitrary but fixed, assume

(1) \( R(F) \)

and prove

[2] \( R(\text{forall } X \text{ in } B1..B2: \ F) \).

We denote \( b1 = T(B1), \ b2 = T(B2), \ f = T(F) \).

From the definition of \( T \) and \( f \), we know

(2) \( \exists fc \in \text{TFormulaCore}: f = \text{next}(fc) \)

We take \( \text{ref} \in \text{RangeEnv}, \ e \in \text{Environment}, \ s \in \text{Stream}, \ d \in \mathbb{N}_\infty, \ h \in \mathbb{N}, \ p \in \mathbb{N} \)

arbitrary but fixed. Assume

(3) \( \vdash (\text{ref } \vdash (\text{forall } X \text{ in } B1..B2: \ F): (h,d)) \)

(4) \( d \in \mathbb{N} \)

(4') \( \text{dom}(e) = \text{dom}(\text{ref}) \)

(5) \( \forall Y \in \text{dom}(e): \text{ref}(Y).1 +i p \leq i e(Y) \leq i \text{ref}(Y).2 +i p \)

and prove

[6] \( \exists b \in \text{Bool} \ \exists d' \in \mathbb{N}: \ d' \leq df+1 \land \vdash \text{next}(\text{TA}(X,b1,b2,f)) \rightarrow^{*(d',p,s,f,e)} \text{done}(b) \)

We prove [6] by contradiction. Assume

(7) \( \forall b \in \text{Bool} \ \forall d' \in \mathbb{N}: \ d' \leq df+1 \Rightarrow \neg (\vdash \text{next}(\text{TA}(X,b1,b2,f)) \rightarrow^{*(d',p,s,f,e)} \text{done}(b)) \).

Note that by the operational semantics,
\[ \neg (\vdash \text{next}(\text{TA}(X,b1,b2,f)) \rightarrow^{*(d',p,s,f,e)} \text{done}(b)) \]
is equivalent to
\[ \exists fc \in \text{TFormulaCore}: \vdash \text{next}(\text{TA}(X,b1,b2,f)) \rightarrow^{*(d',p,s,f,e)} \text{next}(fc) \].

Hence, (7) can be rewritten to
(8) \( \forall d' \in N: (d' \leq df + 1 \Rightarrow \exists fc \in TFormulaCore: \vdash \text{next}(TA(X,b1,b2,f)) \rightarrow *(d',pf,sf,ef) \text{ next}(fc)). \)

We thus know for some \( fc \in TFormulaCore \)

(9) \( \vdash \text{next}(TA(X,b1,b2,f)) \rightarrow *(df+1,pf,.sf,ef) \text{ next}(fc) \)

From the invariant, (2) and (9), there exist \( c \in \text{Context}, p0,p1,p2 \in N \) such that

(10) \( c = (ef,\{(X,sf(ef(X))) \mid X \in \text{dom}(ef))\}) \)
(11) \( p0 = pf+df+1 \)
(12) \( p1 = b1(c) \)
(13) \( p2 = b2(c) \)

and we have 2 cases:

CASE 1:

------

(20) \( df+1 \geq 1 \)
(21) \( p1 \neq \infty \)
(22) \( p0 \leq p1 \)
(23) \( p1 \leq \infty p2 \)
(24) \( fc = TA0(X,p1,p2,f) \)

From (3), by the analysis, we know for some \( l1,u1,l2,u2 \in \mathbb{Z}_{\infty} \) and \( h',d' \in \mathbb{N}_{\infty} \):

(24) \( \vdash B1 : (l1, u1) \)
(25) \( \vdash B2 : (l2, u2) \)
(26) \( \vdash [X \mapsto (l1, u2)] F : (h1, d1) \)
(27) \( hf = \max_{\infty}(h1, N_{\infty}(-i(l1))) \)
(28) \( df = \max_{\infty}(d1, N_{\infty}(u2)) \)

From (4), (28), and the definition of \( \max_{\infty} \), we know

(29) \( d1 \in N \)
(30) \( (u2 \in \mathbb{Z} \land u2 < 0 \land df = d1) \lor (u2 \in N \land df = \max(d1,u2)) \)

From (29) and (30), we can conclude

(31) \( u2 \in \mathbb{Z} \)
(32) \( df = \max(d1,u2) \)

Hence, from (32) we have

(33) \( df \geq u2 \).

From (33) we have

(34) \( pf + df + 1 \geq pf + u2 + 1 > pf + u2 \).

On the other hand, from (25), (4'), (5), and (10), by Lemma 9 we get

(35) \( l2 + i pf \leq i b2(c) \leq i u2 + i pf \).
From (11), (12), (22), and (35) we have

\[(36) \quad pf+df+1 = p_0 \leq p_1 = b_1(c) \leq b_2(c) \leq u_2 + 1 \cdot pf.\]

From (31), (34) and (36) we get a contradiction:

\[pf+df+1 > pf+u_2 \text{ and } pf+df+1 \leq pf+u_2.\]

This proves CASE 1.

CASE 2: 

---

There exist some \(f, g \in P(T\text{Instance})\) such that

\[(100) \quad df+1 \geq 1\]
\[(101) \quad p_1 \neq \infty\]
\[(102) \quad p_1 \leq \infty p_2\]
\[(103) \quad p_0 > p_1\]
\[(104) \quad g \neq \emptyset \lor pf+df+1 \leq \infty p_2\]
\[(105) \quad \forall \text{allInstances}(X,p,p_0,p_1,p_2,f,s,f,g)\]
\[(106) \quad f_c = TA_1(X,p_2,f,g)\]

From (3) and the definition of the analysis, we know for some \(l_1, u_1, l_2, u_2 \in \mathbb{Z}_\infty\) and \(h', d' \in \mathbb{N}_\infty\):

\[(111) \quad \text{ref} \vdash B_1 : (l_1, u_1)\]
\[(112) \quad \text{ref} \vdash B_2 : (l_2, u_2)\]
\[(113) \quad \text{ref}[X \mapsto (l_1, u_2)] \vdash F : (h', d')\]
\[(114) \quad h_f = \max\infty(h', \mathbb{N}_\infty(-\text{i}(l_1)))\]
\[(115) \quad df = \max\infty(d', \mathbb{N}_\infty(u_2))\]

From (4), (115), and the definition of \(\max\infty\), we know

\[(116) \quad d' \in \mathbb{N}\]
\[(117) \quad (u_2 \in \mathbb{Z} \land u_2 < 0 \land df = d') \lor \]
\[\quad (u_2 \in \mathbb{N} \land df = \max(d', u_2))\]

From (116) and (117), we can conclude

\[(118) \quad u_2 \in \mathbb{Z}\]
\[(119) \quad df = \max(d', u_2)\]

From (104), we have two subcases:

Subcase 2.1

---------

\[(200) \quad pf+df+1 \leq \infty p_2\]

From (119), we know

\[(201) \quad df \geq u_2\]

From Lemma 9 with (4’), (5), (10), (13), (112), (118) and the definition of \(b_2\), we know

97
From (200) and (202), we have

\[(203) \quad pf+df+1 \leq pf+u2\]

and thus

\[(204) \quad df+1 \leq u2\]

which contradicts (201).

Subcase 2.2

\[(300) \quad pf+df+1 > \infty p2\]
\[(301) \quad gs \neq \emptyset\]

From (301), (105) and the definition of "forallInstances", we know for some \(t \in \mathbb{N}\), \(g \in TFormula\), \(c0 \in Context\), \(gc \in TFormulaCore\):

\[(302) \quad (t,g,c0) \in gs\]
\[(303) \quad (\forall t1 \in \mathbb{N}, g1 \in TFormula, c1 \in Context:\ (t1,g1,c1) \in gs \land t = t1 \Rightarrow (t,g,c0) = (t1,g1,c1)\]
\[(304) \quad g = next(gc)\]
\[(305) \quad c0.1 = ef[X \mapsto t]\]
\[(306) \quad c0.2 = \{(Y, s(ef(Y))) \mid Y \in \text{dom}(ef) \lor Y = X\}\]
\[(307) \quad p1 \leq t\]
\[(308) \quad t \leq \min(\infty (p0-1, p2)\]
\[(309) \quad \vdash f \rightarrow \ast (p0-\max(pf,t), \max(pf,t), sf, c0.1) g\]

We define

\[(310) \quad \text{ref}' := \text{ref}[X \mapsto (11, u2)]\]
\[(311) \quad \text{ef}' := \text{ef}[X \mapsto t]\]

From (311), we know

\[(312) \quad \text{dom}(ef') = \text{dom}(ef) \cup \{X\}\]

and claim

\[(313) \quad \forall Y \in \text{dom}(ef'): \text{ref}'(Y).1 + pf \leq i \text{ef}'(Y) \leq i \text{ref}'(Y).2 + pf\]

----

Proof: take arbitrary \(Y \in \text{dom}(ef')\). From (312), we have two cases:

* case \(Y \neq X\): we have \(Y \in \text{dom}(ef)\) and by \((4')\) \(\text{ref}'(Y) = \text{ref}(Y)\) and \(\text{ef}'(Y) = \text{ef}(Y)\); it thus suffices to show \(\text{ref}(Y).1 + pf \leq i \text{ef}(Y) \leq i \text{ref}(Y).2 + pf\) which follows from (5).

* case \(Y = X\): we have \(\text{ref}'(Y) = (11, u2)\) and \(\text{ef}'(Y) = t\); it thus suffices to show \(11 + i pf \leq i t \leq i u2 + pf\). From (307) and (308) it suffices to show
From Lemma 9, (4'), (5), (10), (12), (11), and the definition of \( b_1 \), we have \( l_1 + i \cdot pf \leq i \cdot p_1 \) and thus [1].

From Lemma 9, (4'), (5), (10), (13), (112), and the definition of \( b_2 \), we have \( p_2 \leq i \cdot u_2 + i \cdot pf \) and thus [2].

---

From (1), (113), (116), (305), (310), (311), (313) and the definitions of \( R \) and \( f \), we know that there exists some \( b \in \text{Bool} \) and \( d_0 \in \mathbb{N} \) such that

\[
(314) \quad d_0 \leq d' + 1
\]

\[
(315) \quad \vdash f \rightarrow^*(d_0, pf, sf, c_0.1) \text{ done}(b)
\]

We proceed by case distinction.

Subcase 2.2.1

\[
(400) \quad t < pf
\]

From (304), (309) and (400), we know

\[
(401) \quad \vdash f \rightarrow^*(p_0 - pf, pf, sf, c_0.1) \text{ next}(gc)
\]

Because the rule system is deterministic and there is no transition starting with \( \text{done}(b) \), to derive a contradiction, it suffices with (315) and (401) to show

\[
(402) \quad d_0 \leq p_0 - pf
\]

which holds because

\[
d_0 \leq (314) d' + 1 \leq (119) df + 1 = (pf + df + 1) - pf = (11) p_0 - pf
\]

Subcase 2.2.2

\[
(500) \quad t \geq pf
\]

From (304), (309) and (500), we know

\[
(501) \quad \vdash f \rightarrow^*(p_0 - t, t, sf, c_0.1) \text{ next}(gc)
\]

By a generalization of Lemma 7, we know from (2), (315) and (500)

\[
(502) \quad \vdash f \rightarrow^*(\max(1, d_0 - (t - pf)), t, sf, c_0.1) \text{ done}(b)
\]

Because the rule system is deterministic and there is no transition starting with \( \text{done}(b) \), to derive a contradiction, it suffices with (501) and (502) to show

\[
(503) \quad \max(1, d_0 - (t - pf)) \leq p_0 - t
\]

From (308), we know
(504) \( t \leq p_0 - 1 \)

and thus

(505) \( 1 \leq p_0 - t \)

From (505), to show [503] it suffices to show

[506] \( d_0 - (t - pf) \leq p_0 - t \)

for which it suffices to show

[507] \( d_0 + pf \leq p_0 \)

which holds because

\[
d_0 + pf \leq (314) \ d' + 1 + pf \leq (119) \ df + 1 + pf = (11) \ p_0
\]

QED.
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) \( f \rightarrow (kf, sf \downarrow kf, s(kf), cf) \) done(bf).

From (2.5), by the definition of \( \rightarrow \) and (2.4) we get (2.2).

It proves C2.

C3. \( F_{t=next(TCS(f_1,f_2))} \) for some \( f_1,f_2 \in TFormula \)

We take \( cf \) arbitrary but fixed, assume

(3.1) \( next(TCS(f_1,f_2)) \rightarrow (pf, sf \downarrow pf, s(pf), cf) \) done(bf)

and prove

(3.2) \( next(TCS(f_1,f_2)) \rightarrow (kf, sf \downarrow kf, s(kf), cf) \) done(bf)

From (3.1) we have two alternatives:

(a) We have

(3.3) \( bf=false \) and

(3.4) \( f_1 \rightarrow (pf, sf \downarrow pf, s(pf), cf) \) done(false).

By the induction hypothesis, from (3.4) we get

(3.5) \( f_1 \rightarrow (kf, sf \downarrow kf, s(kf), cf) \) done(false).

From (3.5), by the definition of \( \rightarrow \) we get (3.2).

(b) We have

(3.6) \( f_1 \rightarrow (pf, sf \downarrow pf, s(pf), cf) \) done(true)

(3.7) \( f_2 \rightarrow (pf, sf \downarrow pf, s(pf), cf) \) done(bf).

By the induction hypothesis, we get from (3.6) and (3.7) respectively

(3.8) \( f_1 \rightarrow (kf, sf \downarrow kf, s(kf), cf) \) done(true)

(3.9) \( f_2 \rightarrow (kf, sf \downarrow pf, s(kf), cf) \) done(bf).

From (3.8) and (3.9), by the definition of \( \rightarrow \) we get (3.2).

It proves C3.

C4. \( F_{t=next(TCP(f_1,f_2))} \) for some \( f_1,f_2 \in TFormula \)

We take \( cf \) arbitrary but fixed, assume
(4.1) $\text{next(TCP}(f_1,f_2)) \rightarrow (pf, sf, s(pf), cf) \text{ done(bf)}$

and prove

(4.2) $\text{next(TCP}(f_1,f_2)) \rightarrow (kf, sf, s(kf), cf) \text{ done(bf)}$

From (4.1) we have three alternatives:

(a) We have

--------
(4.3) $bf=false$
(4.4) $f_1 \rightarrow (pf, sf, s(pf), cf) \text{ next}(f_1')$ for some $f_1' \in \text{TFormulaCore}$
(4.5) $f_2 \rightarrow (pf, sf, s(pf), cf) \text{ done(false)}$.

From (4.4) and (4.5) we obtain by the induction hypothesis, respectively,

(4.6) $f_1 \rightarrow (kf, sf, s(kf), cf) \text{ next}(f_1')$

(4.7) $f_2 \rightarrow (kf, sf, s(kf), cf) \text{ done(false)}$.

From (4.6) and (4.7), by the definition of $\rightarrow$ and (4.3) we get (4.2).

(b) We have

--------
(4.8) $bf=false \text{ and}$
(4.9) $f_1 \rightarrow (pf, sf, s(pf), cf) \text{ done(false)}$.

By the induction hypothesis, from (4.4) we get

(4.5) $f_1 \rightarrow (kf, sf, s(kf), cf) \text{ done(false)}$.

From (3.5), by the definition of $\rightarrow$ we get (4.2).

(c) We have

--------
(4.6) $f_1 \rightarrow (pf, sf, s(pf), cf) \text{ done(true)}$
(4.8) $f_2 \rightarrow (pf, sf, s(pf), cf) \text{ done(bf)}$.

By the induction hypothesis, we get from (3.6) and (3.7) respectively

(4.9) $f_1 \rightarrow (kf, sf, s(kf), cf) \text{ done(true)}$

(4.10) $f_2 \rightarrow (kf, sf, s(kf), cf) \text{ done(bf)}$.

From (4.9) and (4.10), by the definition of $\rightarrow$ we get (4.2).

It proves C4.

C5. $F_t=\text{next(TA}(X,b_1,b_2,f))$

--------------------------
We take \( cf \) arbitrary but fixed, assume

\[(5.1) \ \text{next}(TA(X,b_1,b_2,f)) \rightarrow (pf,sf, pf, s(pf), cf) \text{ done(bf)}\]

and prove

\[5.2 \ \text{next}(TA(X,b_1,b_2,f)) \rightarrow (kf,sf,kf,sf(kf), cf) \text{ done(bf)}\]

(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

\[5.3 \ p_1^* = b_1(cf)\]
\[5.4 \ p_2^* = b_2(cf)\]
\[5.5 \ p_1^* \neq \infty\]
\[5.6 \ \text{next}(TA_0(X,p_1^*,p_2^*,f)) \rightarrow (kf,sf,kf,sf(kf), cf) \text{ done(false)}\]

From (5.1) we know

\[5.7 \ p_1 = b_1(cf)\]
\[5.8 \ p_2 = b_2(cf)\]
\[5.9 \ p_1 \neq \infty\]
\[5.10 \ \text{next}(TA_0(X,p_1,p_2,f)) \rightarrow (pf,sf, pf, s(pf), cf) \text{ done(false)}\]

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

\[5.11 \ \text{next}(TA_0(X,p_1,p_2,f)) \rightarrow (kf,sf,kf,sf(kf), cf) \text{ done(false)}\]

By Def. \( \rightarrow \), to prove [5.11], we need to prove

\[5.12 \ kf \geq p_1\]
\[5.13 \ \text{next}(TA_1(X,p_2,f,fsk)) \rightarrow (kf,sf,kf,sf(kf), cf) \text{ done(false)}\]

where

\[5.14 \ fsk = \{(p_0,f,(cf.1[X \rightarrow p_0],cf.2[X \rightarrow (sf(kf)(p_0))]) \mid p_1 \leq p_0 < \infty \min \infty(kf,p_2+\infty1)\}\]

From (5.10), by the definition of \( \rightarrow \), we know

\[5.15 \ pf \geq p_1\]
\[5.16 \ \text{next}(TA_1(X,p_2,f,fsp)) \rightarrow (pf, sf, pf, s(pf), cf) \text{ done(false)}\]

where
(5.17) \( f_{sp} = \{(p_0,f,(cf.1[X\rightarrow p_0],cf.2[X\rightarrow sf(p_f)](p_0))) \mid p_1 \leq p_0 < \infty \min (p_f, p_2 + \infty 1)\} \)

Then \( [5.12] \) follows from (1) and (5.15).

To prove \( [5.13] \), by Def. \( \rightarrow \) we need to prove

\[ [5.18] \exists t \in \mathbb{N}, g \in TFormula, c \in Context: (t,g,c) \in f_{s0k} \land \vdash g \rightarrow (kf, sf(kf), c) \text{ done(false)} \]

where

\[ (5.19) f_{s0k} = \text{if } kf > \infty \text{ then } f_{sk} \text{ else } f_{sk} \cup \{(kf,f,(cf.1[X \rightarrow kf],cf.2[X \rightarrow sf(kf)])\}) \]

From (5.16) we know that there exist \( tp \in \mathbb{N}, gp \in TFormula, cp \in Context \) such that

\[ (5.20) (tp,gp,cp) \in f_{s0p} \]
\[ (5.21) gp \rightarrow (pf, sf(pf), sf(pf), cp) \text{ done(false)} \]

where

\[ (5.22) f_{s0p} = \text{if } pf > \infty \text{ then } f_{sp} \text{ else } f_{sp} \cup \{(pf,f,(cf.1[X \rightarrow pf],cf.2[X \rightarrow sf(pf)])\}) \]

Since by (1) \( kf \geq pf \), from (5.14) and (5.17) we have

\[ (5.23) f_{sp} \subseteq f_{sk} \]

Also, we have either

\[ (5.25) (pf,f,(cf.1[X \rightarrow pf],cf.2[X \rightarrow sf(pf)]) \in f_{sk} \text{ (when } kf > pf, \text{ since } sf(pf)(kf) = sf(pf)) \]

or

\[ (5.26) (pf,f,(cf.1[X \rightarrow pf],cf.2[X \rightarrow sf(pf)]) \in f_{s0k}, \text{ (kf=pf).} \]

From (5.25) and (5.26) we get

\[ (5.27) (pf,f,(cf.1[X \rightarrow pf],cf.2[X \rightarrow sf(pf)]) \in f_{s0k}, \text{ when } kf \geq pf. \]

From (1), (5.23), (5.27), (5.19), (5.22) we get

\[ (5.28) f_{sp} \subseteq f_{sk} \]

Then from (5.20) we get

\[ (5.29) (tp,gp,cp) \in f_{s0k} \]

From (5.21) and (2) we get

\[ (5.30) gp \rightarrow (kf, sf(kf), sf(kf), cp) \text{ done(false)} \]

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

Lemma 7 (Shifting Lemma).

∀ f∈TFormulaCore, n,p∈N: s∈Stream, e∈Environment, b∈Bool:
    n>0 ⇒ next(f) → *(n+1,p,s,e) done(b) ⇒ next(f) → *(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) → *(n+1,p,s,e) done(b)

and show

[3] next(f) → *(n,p+1,s,e) done(b).

From (2), by the definition of →*, there exists Ft′∈TFormula such that

(4) next(f) → (p,s↓p,s(p),c) Ft′
(5) Ft′ → *(n,p+1,s,e) done(b)

where

(6) c = (e,{(X,s(e(X))) | X ∈ dom(e)}).

Since n>0 by (1), we have that Ft′ is a 'next' formula, say next(f′).
Then from (5), by the definition of →*, we know that there exists
Ft′′∈TFormula such that

(7) next(f′) → (p+1,s↓(p+1),s(p+1),c) Ft′′
(8) Ft′′ → *(n-1,p+2,s,e) done(b).

In order to prove [3], by the definition of →*, we need to find such a
Ft0∈TFormula that

[9] next(f) → (p+1,s↓(p+1),s(p+1),c) Ft0
[10] Ft0 → *(n-1,p+2,s,e) done(b).

We take Ft0=Ft′′. Then [10] follows from (8). We only need to prove [9]:

Given

(4) next(f) → (p,s↓p,s(p),c) next(f′)
(7) next(f′) → (p+1,s↓(p+1),s(p+1),c) Ft′′

Prove:

[9] next(f) → (p+1,s↓(p+1),s(p+1),c) Ft′′.

It follows from Lemma 8.
A.10 Lemma 8: Triangular Reduction Lemma

Lemma 8 (Triangular Reduction G).

∀G1, G2 ∈ TFormulaCore, Ft ∈ TFormula, p ∈ N, s ∈ Stream, c ∈ Context :
next(G1) → (p, s ↓ p, s(p), c) next(G2) ∧ next(G2) → (p+1, s ↓ (p+1), s(p+1), c) Ft
⇒
next(G1) → (p+1, s ↓ (p+1), s(p+1), c) Ft.

Proof
-----

Φ ⊆ TFormulaCore
Φ(G1) :
∀G2 ∈ TFormulaCore, Ft ∈ TFormula, p ∈ N, s ∈ Stream, c ∈ Context :
next(G1) → (p, s ↓ p, s(p), c) next(G2) ∧ next(G2) → (p+1, s ↓ (p+1), s(p+1), c) Ft
⇒
next(G1) → (p+1, s ↓ (p+1), s(p+1), c) Ft.

We prove

(G) ∀G′ ∈ TFormulaCore : Φ(G′).

Case (C1) G′ = TN(Ft) for some Ft ∈ TFormula

We show

Φ(G′)

Take F2f, Ftf, pf, sf, cf arbitrary but fixed.

Assume

(C1.1) next(TN(Ft)) → (pf, sf ↓ pf, sf(pf), cf) next(G2f)
(C1.2) next(G2f) → (pf+1, sf ↓ (pf+1), sf(pf+1), cf) Ftf

Show

[C1.a] next(TN(Ft)) → (pf+1, sf ↓ (pf+1), sf(pf+1), cf) Ftf.

From (C1.1) and Def. →, we know for some G2′ ∈ TFormula

(C1.3) G2f = TN(next(G2′))
(C1.4) next(TN(Ft)) → (pf, sf ↓ pf, sf(pf), cf) next(TN(next(G2′)))
(C1.5) Ft → (pf, sf ↓ pf, sf(pf), cf) next(G2′)

From (C1.2, C1.3), we thus have

(C1.6) next(TN(next(G2′))) → (pf+1, sf ↓ (pf+1), sf(pf+1), cf) Ftf

From (C1.5) and Def. →, we know for some G ∈ TFormulaCore
(C1.7) \( F_t = \text{next}(G) \)
(C1.8) \( \text{next}(G) \rightarrow (p_f, s_f, (p_f), (s_f), (c_f)) \text{next}(G') \)

From (C1.7) and [C1.a], it suffices to show

[C1.b] \( \text{next}(\text{TN}(\text{next}(G))) \rightarrow (p_{f+1}, s_f, (p_{f+1}), (s_f), (p_{f+1}), (c_f)) \text{F}_t \).

From (C1,C1.8) and the induction assumption, we know \( \Phi(G) \) and thus

(C1.9)

\[
\forall G' \in \text{TFormulaCore}, F_t \in \text{TFormula}, p \in \mathbb{N}, s \in \text{Stream}, c \in \text{Context} : \\
\text{next}(G) \rightarrow (p, s \downarrow p, s(p), c) \text{next}(G) \land \text{next}(G) \rightarrow (p+1, s \downarrow (p+1), s(p+1), c) \text{ F}_t \\
\Rightarrow \text{next}(G) \rightarrow (p+1, s \downarrow (p+1), s(p+1), c) \text{ F}_t.
\]

From (C1.6) and Def. \( \rightarrow \), we have 3 cases.

Case C1.c1. there exists some \( F_c' \in \text{TFormulaCore} \) such that

(C1.c1.1) \( \text{next}(G') \rightarrow (p_{f+1}, s_f, (p_{f+1}), (s_f), (p_{f+1}), (c_f)) \text{next}(F_c') \)
(C1.c1.2) \( \text{F}_t f=\text{next}(\text{TN}(\text{next}(F_c'))) \)

From (C1.c1.2) and [C1.b], ut suffices thus to show

[C1.c1.b] \( \text{next}(\text{TN}(\text{next}(G))) \rightarrow (p_{f+1}, s_f, (p_{f+1}), (s_f), (p_{f+1}), (c_f)) \text{next}(\text{TN}(\text{next}(F_c'))) \)

From (C1.9), (C1.8), (C1.c1.1), we have

(C1.c1.3) \( \text{next}(G) \rightarrow (p_{f+1}, s_f, (p_{f+1}), (s_f), (p_{f+1}), (c_f)) \text{next}(F_c') \)

From (C1.c1.3) and Def. \( \rightarrow \), we know [C1.c1.b].

This proves the case C1.c1.

Case C1.c2. we have

(C1.c2.1) \( \text{next}(G') \rightarrow (p_{f+1}, s_f, (p_{f+1}), (s_f), (p_{f+1}), (c_f)) \text{ done(true)} \)
(C1.c2.2) \( \text{F}_t f=\text{done(false)} \)

From (C1.c2.2) and [C1.b], it suffices thus to show

[C1.c2.b] \( \text{next}(\text{TN}(\text{next}(G))) \rightarrow (p_{f+1}, s_f, (p_{f+1}), (s_f), (p_{f+1}), (c_f)) \text{ done(false)} \)

From (C1.9), (C1.8), (C1.c2.1), we have

(C1.c2.3) \( \text{next}(G) \rightarrow (p_{f+1}, s_f, (p_{f+1}), (s_f), (p_{f+1}), (c_f)) \text{ done(true)} \).

From (C1.c2.3) and Def. \( \rightarrow \), we know [C1.c2.b].

This proves the case C1.c2.
Case C1.c3. we have
-------------

(C1.c3.1) \text{next}(G'2') \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), cf) \text{ done}(false)
(C1.c3.2) \text{Ftf=done}(true)

It suffices thus to show

[C1.c3.b] \text{next}(\text{TN}(\text{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) \text{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) \text{next}(\text{TCS}(Ft1,Ft2)) \rightarrow (pf, sf \downarrow pf, sf(pf), cf) \text{ next}(G2f)
(C2.2) \text{next}(G2f) \rightarrow (pf+1, sf \downarrow (pf+1), sf(pf+1), cf) \text{ Ftf}

Show

[C2.a] \text{next}(\text{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(\text{next}(Fc1),Ft2)
(C2.c1.2) \text{next}(\text{TCS}(Fc1,Ft2)) \rightarrow (pf, sf \downarrow pf, sf(pf), cf) \text{ next}(\text{TCS}(\text{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) \text{next}(\text{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 = \text{next}(Fc0)
(C2.c1.6) \text{next}(Fc0) \rightarrow (pf, sf \downarrow pf, sf(pf), cf) \text{ next}(Fc1)
From (C2.c1.5) and [C2.a], we need to show

\[ \text{next(TCS(next(Fc0),Ft2))} \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \ F_{tf}. \]

From (C2), (C2.c1.5) and the induction hypothesis, we know \( \Phi(Fc0) \) and thus (C2.c1.7)

\[ \forall G2 \in TFormulaCore, Ft \in TFormula, p \in \mathbb{N}, s \in Stream, c \in Context : \]
\[ \text{next(Fc0)} \rightarrow (p, s\downarrow p, s(p), c) \ \text{next}(G2) \land \text{next}(G2) \rightarrow (p+1, s\downarrow (p+1), s(p+1), c) \ Ft \]
\[ \Rightarrow \text{next}(Fc0) \rightarrow (p+1, s\downarrow (p+1), s(p+1), c) \ F_{tf}. \]

From (C2.c1.4), we have the following cases.

**Case C2.c1.c1.** There exists \( Fc' \in TFormulaCore \) such that

\begin{align*}
(C2.c1.c1.1) & \quad F_{tf} = \text{next(TCS(next(Fc'),Ft2))} \\
(C2.c1.c1.2) & \quad \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) & \quad \text{next}(Fc1) \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \ \text{next}(Fc'). \\
\end{align*}

From (C2.c1.c1.1) and [C2.c1.b], we need to show

\[ \text{next(TCS(next(Fc0),Ft2))} \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \ \text{next(TCS(next(Fc'),Ft2))}. \]

In this case from (C2.c1.6), (C2.c1.c1.3), and (C2.c1.7) we have

\[ \text{next(Fc0)} \rightarrow (p+1, s\downarrow (p+1), s(p+1), c) \ \text{next}(Fc'). \]

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.**

\begin{align*}
(C2.c1.c2.1) & \quad F_{tf} = \text{done(false)} \\
(C2.c1.c2.2) & \quad \text{next(TCS(next(Fc1),Ft2))} \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \ \text{done(false)}. \\
(C2.c1.c2.3) & \quad \text{next}(Fc1) \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \ \text{done(false)}. \\
\end{align*}

From (C2.c1.c2.1) and [C2.c1.b], we need to show

\[ \text{next(TCS(next(Fc0),Ft2))} \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \ \text{done(false)}. \]

From (C2.c1.6), (C2.c1.c2.3) and (C2.c1.7) we have

\[ \text{next(Fc0)} \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \ \text{done(false)}. \]

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 $Ft2' \in \text{TFormula}$ such that

\begin{align*}
\text{(C2.c1.c3.1)} & \quad Ftf = Ft2' \\
\text{(C2.c1.c3.2)} & \quad \text{next(TCS(next(Fc1),Ft2))) } \rightarrow (pf+1, sf\downarrow pf+1), sf(pf+1), cf) \text{ } Ft2'. \\
\text{(C2.c1.c3.3)} & \quad \text{next(Fc1) } \rightarrow (pf+1, sf\downarrow pf+1), sf(pf+1), cf) \text{ done(true).} \\
\text{(C2.c1.c3.4)} & \quad Ft2 \rightarrow (pf+1, sf\downarrow pf+1), sf(pf+1), cf) \text{ } Ft2'.
\end{align*}

From (C2.c1.c3.1) and [C2.c1.b], we need to show

\[ \text{next(TCS(next(Fc0),Ft2))) } \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ } Ft2'. \]

From (C2.c1.c3.3), (C2.c1.c3.4), and (C2.c1.7) we have

\[ \text{next(Fc0) } \rightarrow (pf+1, sf\downarrow pf+1), sf(pf+1), cf) \text{ done(true).} \]

From (C2.c1.c3.5) and (C2.c1.c3.4), by Def. $\rightarrow$, 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

\[ \text{next(TCS(Ft1,Ft2)) } \rightarrow (pf, sf\downarrow pf), sf(pf), cf) \text{ next(G2f)} \]

Case C2.c1 considered the case when $G2f = \text{TCS(next(Fc1),Ft2)}$.

According to Def. $\rightarrow$, the other alternative for $G2f$ is the following:

There exists $G2' \in \text{TFormulaCore}$ such that

\begin{align*}
\text{(C2.c2.1)} & \quad G2f = G2' \\
\text{(C2.c2.2)} & \quad \text{next(TCS(Ft1,Ft2)) } \rightarrow (pf, sf\downarrow pf), sf(pf), cf) \text{ next(G2')} \\
\text{(C2.c2.3)} & \quad Ft1 \rightarrow (pf, sf\downarrow pf), sf(pf), cf) \text{ done(true)} \\
\text{(C2.c2.4)} & \quad Ft2 \rightarrow (pf, sf\downarrow pf), sf(pf), cf) \text{ next(G2')} \\
\end{align*}

From (C2.2) and (C2.c2.1) we have

\[ \text{next(G2')} \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ } Ftf. \]

From (C2.c2.6) and Def. $\rightarrow$, we know for some $Fc1 \in \text{TFormulaCore}$

\[ \text{Ft1 = next(Fc1)} \]

\[ \text{next(Fc1) } \rightarrow (pf, sf\downarrow pf), sf(pf), cf) \text{ done(true)} \]

From (C2.c2.4) and Def. $\rightarrow$, we know for some $Fc2 \in \text{TFormulaCore}$

\[ \text{Ft2 = next(Fc2)} \]

\[ \text{next(Fc2) } \rightarrow (pf, sf\downarrow pf), sf(pf), cf) \text{ next(G2')} \]

From (C2.c2.6), (C2.c2.8) and [C2.a], we need to show

\[ \text{next(TCS(next(Fc1),next(Fc2))) } \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ } Ftf. \]
From (C2.c2.7), by Lemma 6, we know

\((C2.c2.10) \text{ next}(F_{c1}) \rightarrow (p+1,sf\downarrow(p+1),sf(p+1),cf) \text{ done}(true)\).

From (C2), (C2.c2.8) and the induction hypothesis, we know \(\Phi(F_{c2})\) and thus

\((C2.c2.11) \forall G_{2} \in T_{\text{FormulaCore}}, F_{t} \in T_{\text{Formula}}, p \in \mathbb{N}, s \in \text{Stream}, c \in \text{Context} : \text{next}(F_{c2}) \rightarrow (p,s\downarrow p, s(p), c) \text{ next}(G_{2}) \land \text{next}(G_{2}) \rightarrow (p+1,s\downarrow(p+1), s(p+1), c) \text{ Ft} \Rightarrow \text{next}(F_{c2}) \rightarrow (p+1,s\downarrow(p+1), s(p+1), c) \text{ Ft.}\)

From (C2.c2.9), (C2.c2.5), and (C2.c2.11), we get

\((C2.c2.11) \text{ next}(F_{c2}) \rightarrow (p+1,sf\downarrow(p+1),sf(p+1),cf) \text{ Ftf.}\)

From (C2.c2.10) and (C2.c2.11), by Def.→, we get \([C2.c2.b]\).

This proves the case C2.

This finishes the proof of case C2.

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

Case (C3) \(G' = TCP(F_{t1},F_{t2})\) for some \(F_{t1},F_{t2} \in T_{\text{Formula}}\).

We show

\(\Phi(G')\)

Take \(F_{2f},F_{tf},p,f,sf,cf\) arbitrary but fixed.

Assume

\((C3.1) \text{ next}(TCP(F_{t1},F_{t2})) \rightarrow (p,f\downarrow p, sf(p), cf) \text{ next}(G_{2f})\)
\((C3.2) \text{ next}(G_{2f}) \rightarrow (p+1,f\downarrow(p+1), sf(p+1), cf) \text{ Ftf}\)

Show

\([C3.a] \text{ next}(TCP(F_{t1},F_{t2})) \rightarrow (p+1,f\downarrow(p+1), sf(p+1), cf) \text{ Ftf.}\)

From (C3.1), by Def.→, we have three cases.

Case C3.c1

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

There exists \(F_{c1},F_{c2} \in T_{\text{FormulaCore}}\) such that

\((C3.c1.1) G_{2f} = TCP(\text{next}(F_{c1}),\text{next}(F_{c2}))\)
\((C3.c1.2) F_{t1} \rightarrow (p,f\downarrow p, sf(p), cf) \text{ next}(F_{c1})\)
\((C3.c1.3) F_{t2} \rightarrow (p,f\downarrow p, sf(p), cf) \text{ next}(F_{c2})\)
\((C3.c1.4) \text{ next}(TCP(F_{t1},F_{t2})) \rightarrow (p,f\downarrow p, sf(p), cf) \text{ next}(TCP(\text{next}(F_{c1}),\text{next}(F_{c2})))\)

From (C3.2) and (C3.c1.1) we have
From (C3.c1.2) and Def.→, we know for some $F_{c1}' \in T_{\text{FormulCore}}$

\begin{align*}
(C3.c1.6) \quad & F_{t1} = \text{next}(F_{c1}') \\
(C3.c1.7) \quad & \text{next}(F_{c1}') \rightarrow (p_f, s_f \downarrow p_f, s_f(p_f), c_f) \ \text{next}(F_{c1})
\end{align*}

From (C3.c1.3) and Def.→, we know for some $F_{c2}' \in T_{\text{FormulCore}}$

\begin{align*}
(C3.c1.8) \quad & F_{t2} = \text{next}(F_{c2}') \\
(C3.c1.9) \quad & \text{next}(F_{c2}') \rightarrow (p_f, s_f \downarrow p_f, s_f(p_f), c_f) \ \text{next}(F_{c2})
\end{align*}

From (C3.c1.6), (C3.c1.8) and [C3.a], we need to show

\begin{align*}
[C3.c1.b] \quad & \text{next}(TCP(next(F_{c1}'), next(F_{c2}'))) \rightarrow (p_f+1, s_f \downarrow (p_f+1), s_f(p_f+1), c_f) \ \text{Ftf}.
\end{align*}

From (C3), (C3.c1.6) and the induction hypothesis, we know $\Phi(F_{c1}')$ and thus

\begin{align*}
(C3.c1.10) \quad & \forall G_{2} \in T_{\text{FormulaCore}}, F_t \in T_{\text{Formula}}, p \in \mathbb{N}, s \in \text{Stream}, c \in \text{Context} : \\
& \text{next}(F_{c1}') \rightarrow (p, s \downarrow p(s), c) \ \text{next}(G_{2}) \land \text{next}(G_{2}) \rightarrow (p+1, s \downarrow (p+1), s(p+1), c) \ \text{Ft} \\
& \Rightarrow \\
& \text{next}(F_{c1}') \rightarrow (p+1, s \downarrow (p+1), s(p+1), c) \ \text{Ft}.
\end{align*}

From (C3), (C3.c1.8) and the induction hypothesis, we know $\Phi(F_{c2}')$ and thus

\begin{align*}
(C3.c1.11) \quad & \forall G_{2} \in T_{\text{FormulaCore}}, F_t \in T_{\text{Formula}}, p \in \mathbb{N}, s \in \text{Stream}, c \in \text{Context} : \\
& \text{next}(F_{c2}') \rightarrow (p, s \downarrow p(s), c) \ \text{next}(G_{2}) \land \text{next}(G_{2}) \rightarrow (p+1, s \downarrow (p+1), s(p+1), c) \ \text{Ft} \\
& \Rightarrow \\
& \text{next}(F_{c2}') \rightarrow (p+1, s \downarrow (p+1), s(p+1), c) \ \text{Ft}.
\end{align*}

From (C3.c1.5), by Def.→, we have the following five cases.

Case C3.c1.1

-------------
There exist $F_{c1}'', F_{c2}'' \in T_{\text{FormulaCore}}$ such that

\begin{align*}
(C3.c1.1.1) \quad & F_{t} = \text{next}(TCP(next(F_{c1}''), next(F_{c2}''))) \\
(C3.c1.1.2) \quad & \text{next}(F_{c1}) \rightarrow (p_f+1, s_f \downarrow (p_f+1), s_f(p_f+1), c_f) \ \text{next}(F_{c1}'') \\
(C3.c1.1.3) \quad & \text{next}(F_{c2}) \rightarrow (p_f+1, s_f \downarrow (p_f+1), s_f(p_f+1), c_f) \ \text{next}(F_{c2}'') \\
(C3.c1.1.4) \quad & \text{next}(TCP(next(F_{c1}), next(F_{c2}))) \rightarrow (p_f+1, s_f \downarrow (p_f+1), s_f(p_f+1), c_f) \\
& \text{next}(TCP(next(F_{c1}''), next(F_{c2}''))) \\
\end{align*}

From (C3.c1.1.1) and [C3.c1.b] we need to prove

\begin{align*}
[C3.c1.b] \quad & \text{next}(TCP(next(F_{c1}''), next(F_{c2}''))) \rightarrow (p_f+1, s_f \downarrow (p_f+1), s_f(p_f+1), c_f) \\
& \text{next}(TCP(next(F_{c1}''), next(F_{c2}''))) \\
\end{align*}

From (C3.c1.7), (C3.c1.2), and (C3.c1.10) we have
\(\text{(C3.c1.c1.5)}\) \(\text{next}(\text{Fc1'}) \rightarrow (pf+1, \text{sf} \downarrow (pf+1), \text{sf}(pf+1), cf) \text{ next}(\text{Fc1''})\).

From \((\text{C3.c1.9}), (\text{C3.c1.c1.3}), \text{ and } (\text{C3.c1.11})\) we have

\(\text{(C3.c1.c1.6)}\) \(\text{next}(\text{Fc2'}) \rightarrow (pf+1, \text{sf} \downarrow (pf+1), \text{sf}(pf+1), cf) \text{ next}(\text{Fc2''})\)

From \((\text{C3.c1.c1.5})\) and \((\text{C3.c1.c1.6})\), by Def. \(\rightarrow\) we get \([\text{C3.c1.c1.b}].\)

This proves case the \(\text{C3.c1.c1}.\)

Case \(\text{C3.c1.c2}\)

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

There exist \(\text{Fc1''} \in \text{TFormulaCore}\) such that

\(\text{(C3.c1.c2.1)}\) \(\text{Ftf} = \text{next}(\text{Fc1''})\)

\(\text{(C3.c1.c2.2)}\) \(\text{next}(\text{Fc1}) \rightarrow (pf+1, \text{sf} \downarrow (pf+1), \text{sf}(pf+1), cf) \text{ next}(\text{Fc1''})\)

\(\text{(C3.c1.c2.3)}\) \(\text{next}(\text{Fc2}) \rightarrow (pf+1, \text{sf} \downarrow (pf+1), \text{sf}(pf+1), cf) \text{ done(true)}\)

\(\text{(C3.c1.c2.4)}\) \(\text{next}(\text{TCP(next}(\text{Fc1}), \text{next}(\text{Fc2}))) \rightarrow (pf+1, \text{sf} \downarrow (pf+1), \text{sf}(pf+1), cf) \text{ next}(\text{Fc1''})\)

From \((\text{C3.c1.c2.1})\) and \([\text{C3.c1.b}]\) we need to prove

\([\text{C3.c1.c2.b}]\) \(\text{next}(\text{TCP(next}(\text{Fc1'}), \text{next}(\text{Fc2'}))) \rightarrow (pf+1, \text{sf} \downarrow (pf+1), \text{sf}(pf+1), cf) \text{ next}(\text{Fc1''})\).  

From \((\text{C3.c1.7}), (\text{C3.c1.c2.2}), \text{ and } (\text{C3.c1.10})\) we have

\(\text{(C3.c1.c2.5)}\) \(\text{next}(\text{Fc1'}) \rightarrow (pf+1, \text{sf} \downarrow (pf+1), \text{sf}(pf+1), cf) \text{ next}(\text{Fc1''})\).

From \((\text{C3.c1.9}), (\text{C3.c1.c2.3}), \text{ and } (\text{C3.c1.11})\) we have

\(\text{(C3.c1.c2.6)}\) \(\text{next}(\text{Fc2'}) \rightarrow (pf+1, \text{sf} \downarrow (pf+1), \text{sf}(pf+1), cf) \text{ done(true)}.\)

From \((\text{C3.c1.c2.5})\) and \((\text{C3.c1.c2.6})\), by Def. \(\rightarrow\) we get \([\text{C3.c1.c2.b}].\)

This proves the case \(\text{C3.c1.c2}.\)

Case \(\text{C3.c1.c3}\)

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

There exist \(\text{Fc1''} \in \text{TFormulaCore}\) such that

\(\text{(C3.c1.c3.1)}\) \(\text{Ftf} = \text{done(false)}\)

\(\text{(C3.c1.c3.2)}\) \(\text{next}(\text{Fc1}) \rightarrow (pf+1, \text{sf} \downarrow (pf+1), \text{sf}(pf+1), cf) \text{ next}(\text{Fc1''})\)

\(\text{(C3.c1.c3.3)}\) \(\text{next}(\text{Fc2}) \rightarrow (pf+1, \text{sf} \downarrow (pf+1), \text{sf}(pf+1), cf) \text{ done(false)}\)

\(\text{(C3.c1.c3.4)}\) \(\text{next}(\text{TCP(next}(\text{Fc1}), \text{next}(\text{Fc2}))) \rightarrow (pf+1, \text{sf} \downarrow (pf+1), \text{sf}(pf+1), cf) \text{ done(false)}.\)

From \((\text{C3.c1.c3.1})\) and \([\text{C3.c1.b}]\) we need to prove

\([\text{C3.c1.c2.b}]\) \(\text{next}(\text{TCP(next}(\text{Fc1'}), \text{next}(\text{Fc2'}))) \rightarrow (pf+1, \text{sf} \downarrow (pf+1), \text{sf}(pf+1), cf) \text{ done(false)}.\)

From \((\text{C3.c1.7}), (\text{C3.c1.c3.2}), \text{ and } (\text{C3.c1.10})\) we have
(C3.c1.c3.5) \(\text{next}(Fc1') \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ next}(Fc1'')\).

From (C3.c1.9), (C3.c1.c3.3), and (C3.c1.11) we have

(C3.c1.c3.6) \(\text{next}(Fc2') \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ done(false)}\).

From (C3.c1.c3.5) and (C3.c1.c3.6), by Def. \(\rightarrow\) we get [C3.c1.c3.6].

This proves the case C3.c1.c3.

Case C3.c1.c4
-------------

(C3.c1.c4.1) \(Ftf = \text{done(false)}\)
(C3.c1.c4.2) \(\text{next}(Fc1) \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ done(false)}\)
(C3.c1.c4.3) \(\text{next}(TCP(\text{next}(Fc1), \text{next}(Fc2))) \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ done(false)}\)

From (C3.c1.c4.1) and [C3.c1.b] we need to prove

[C3.c1.c4.5] \(\text{next}(TCP(\text{next}(Fc1'), \text{next}(Fc2'))) \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ done(false)}\).

From (C3.c1.7), (C3.c1.c4.2), and (C3.c1.10) we have

(C3.c1.c4.5) \(\text{next}(Fc1') \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ done(false)}\).

From (C3.c1.c4.5) by Def. \(\rightarrow\) we get [C3.c1.c4.5].

This proves the case C3.c1.c4.

Case C3.c1.c5
-------------

There exist \(Fc2'' \in TFormulaCore\) such that

(C3.c1.c5.1) \(Ftf = \text{next}(Fc2'')\)
(C3.c1.c5.2) \(\text{next}(Fc1) \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ done(true)}\)
(C3.c1.c5.3) \(\text{next}(Fc2) \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ next}(Fc2'')\)
(C3.c1.c5.4) \(\text{next}(TCP(\text{next}(Fc1), \text{next}(Fc2))) \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ next}(Fc2'')\)

From (C3.c1.c5.1) and [C3.c1.b] we need to prove

[C3.c1.c5.5] \(\text{next}(TCP(\text{next}(Fc1'), \text{next}(Fc2'))) \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ next}(Fc2''\)).

From (C3.c1.7), (C3.c1.c5.2), and (C3.c1.10) we have

(C3.c1.c5.5) \(\text{next}(Fc1') \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ done(true)}\).
From (C3.c1.9), (C3.c1.c5.3), and (C3.c1.11) we have

\[(\text{C3.c1.c5.6}) \quad \text{next}(Fc_2') \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ next}(Fc_2'').\]

From (C3.c1.c5.5) and (C3.c1.c5.6), by Def. \(\rightarrow\) we get \([\text{C3.c1.c5.b}].\)

This proves the case C3.c1.c3.

This proves the case C3.c1.

Case C3.c2
----------

(C3.c2.1) \(Ft_1 \rightarrow (pf, sf\downarrow pf, sf(pf), cf) \text{ next}(G2f)\)
(C3.c2.2) \(Ft_2 \rightarrow (pf, sf\downarrow pf, sf(pf), cf) \text{ done(true)}\)

From (C3.c2.1) and Def. \(\rightarrow\), we know for some \(Fc_1' \in T\text{FormulCore}\)

(C3.c2.3) \(Ft_1 = \text{next}(Fc_1')\)
(C3.c2.4) \(\text{next}(Fc_1') \rightarrow (pf, sf\downarrow pf, sf(pf), cf) \text{ next}(G2f)\)

From (C3.c2.2) and Def. \(\rightarrow\), we know for some \(Fc_2' \in T\text{FormulCore}\)

(C3.c2.5) \(Ft_2 = \text{next}(Fc_2')\)
(C3.c2.6) \(\text{next}(Fc_2') \rightarrow (pf, sf\downarrow pf, sf(pf), cf) \text{ done(true)}\)

From (C3.c2.3), (C3.c2.5) and [C3.a], we need to show

([C3.c2.b] \(\text{next}(TCP(\text{next}(Fc_1'), \text{next}(Fc_2')))) \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ Ftf.}\)

From (C3), (C3.c2.2) and the induction hypothesis, we know \(\Phi(Fc_1')\) and thus

(C3.c2.7)

\[\forall G2 \in T\text{FormulaCore}, Ft \in T\text{Formula}, p \in \mathbb{N}, s \in \text{Stream}, c \in \text{Context} : \]
\[\text{next}(Fc_1') \rightarrow (p, s\downarrow p, s(p), c) \land \text{next}(G2) \rightarrow (p+1, s\downarrow (p+1), s(p+1), c) \] \(\Rightarrow\)
\[\text{next}(Fc_1') \rightarrow (p+1, s\downarrow (p+1), s(p+1), c) \text{ Ftf.}\)

From (C3.c2.4), (C3.2), and (C3.c2.7) we get

(C3.c2.8) \(\text{next}(Fc_1') \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ Ftf.}\)

From (C3.c2.6), by Lemma 6, we get

(C3.c2.9) \(\text{next}(Fc_2') \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), cf) \text{ done(true)}\).

From (C3.c2.8) and (C3.c2.9), by Def. \(\rightarrow\), we get \([\text{C3.c2.b}].\)

This proves the case C3.c2

116
Case C3.c3

(C3.c3.1) \( \text{Ft}_1 \rightarrow (pf, sf, pf, sf(pf), cf) \text{ done(true) } \)
(C3.c3.2) \( \text{Ft}_2 \rightarrow (pf, sf, pf, sf(pf), cf) \text{ next(G2f) } \)

This case can be proved similarly to case C3.c2.

This finishes the proof of C3.

Case (C4) \( G' = TA(X, b_1, b_2, \text{Ft}) \) for some \( X \in \text{Variable}, b_1, b_2 \in \text{BoundValue}, \text{Ft} \in \text{TFormula}. \)

We show \( \Phi(G') \)

Take F2f,Ftf,pf,sf,cf arbitrary but fixed.

Assume

(C4.1) \( \text{next(TA}(X,b_1,b_2,\text{Ft})) \rightarrow (pf, sf, pf, sf(pf), cf) \text{ next(G2f) } \)
(C4.2) \( \text{next(G2f)} \rightarrow (pf +1, sf\downarrow (pf +1), sf(pf +1), cf) \text{ Ftf } \)

Show

[C4.a] \( \text{next(TA}(X,b_1,b_2,\text{Ft})) \rightarrow (pf +1, sf\downarrow (pf +1), sf(pf +1), cf) \text{ Ftf. } \)

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) \( \text{next(TA}_0(X,p_1,p_2,\text{Ft})) \rightarrow (pf, sf\downarrow pf, sf(pf), cf) \text{ next(G2f) } \)

To prove [C4.a], by the definition of \( \rightarrow \), we would have two alternatives: Ftf=done(true) or Ftf\neq done(true). But the case Ftf=done(true) is impossible because of (C4.5). Hence, we assume Ftf\neq 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] \( \text{next(TA}_0(X,p_1,p_2,\text{Ft})) \rightarrow (pf +1, sf\downarrow (pf +1), sf(pf +1), cf) \text{ Ftf. } \)

[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

(C4.c1.1) \( pf < p_1 \)
(C4.c1.2) \text{next}(\text{TA0}(X,p_1,p_2,F_t)) \rightarrow (pf, sf|pf, sf(pf), cf) \text{next}(\text{TA0}(X,p_1,p_2,F_t))
(C4.c1.3) \text{next}(G2f) = \text{next}(\text{TA0}(X,p_1,p_2,F_t))

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) \text{pf} \geq p_1
(C4.c2.2) \text{fs} = \{(p_0, F_t, (cf.1[X \mapsto p_0], cf.2[X \mapsto sf(p_0)])) | p_1 \leq p_0 < \infty \min_{\infty}(pf, p_2 + \infty)\}
(C4.c2.3) \text{next}(\text{TA1}(X,p_2,F_t,fs)) \rightarrow (pf, sf|pf, sf(pf), cf) \text{next}(G2f)

From (C4.c2.3), by the definition of $\rightarrow$, we know

(C4.c2.4) G2f = \text{TA1}(X,p_2,F_t,fs_1), where

(C4.c2.5) \text{fs}_0 =
\begin{align*}
\text{if } \text{pf} > \infty \text{ p}_2 \text{ then } \\
\text{fs} \\
\text{else } \\
\text{fs} \cup \{(pf,F_t,(cf.1[X \mapsto pf],cf.2[X \mapsto sf(pf)]))\}
\end{align*}
(C4.c2.6) \neg \exists t \in \mathbb{N}, g \in \text{TFormula}, c \in \text{Context}:
\text{((t,g,c) \in fs}_0 \wedge \vdash g \rightarrow (pf, sf|pf, sf(pf), c) \text{ done(false)}
(C4.c2.7) \text{fs}_1 = \{(t, \text{next}(fc), c) \in \text{TInstance} | \\
\exists g \in \text{TFormula}:
\text{((t,g,c) \in fs}_0 \wedge \\
\vdash g \rightarrow (pf, sf|pf, sf(pf), c) \text{ next}(fc) \}
(C4.c2.8) \neg (\text{fs}_1 = \emptyset \wedge \text{pf} \geq \infty \text{ p}_2)

From (C4.2) and (C4.c2.4) we have

(C4.c2.9) \text{next}(\text{TA1}(X,p_2,F_t,fs_1)) \rightarrow (pf+1, sf|(pf+1), sf(pf+1), cf) \text{ Ftf}.

Recall that we need to prove

[C4.a.4] \text{next}(\text{TA0}(X, p_1, p_2, F_t)) \rightarrow (pf+1, sf|(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}(\text{TA1}(X,p_2,F_t,fs')) \rightarrow (pf+1, sf|(pf+1), sf(pf+1), cf) \text{ Ftf},

where

(C4.c2.10) \text{fs'} = \{(p_0, F_t, (cf.1[X \mapsto p_0], cf.2[X \mapsto sf(p_0)])) | \\
p_1 \leq p_0 < \infty \min_{\infty}(pf+1,p_2+\infty)\}.

Note that if $pf > \infty p_2$ then $\min_{\infty}(pf+1,p_2+\infty)=\min_{\infty}(pf,p_2+\infty)=\min_{\infty}(pf,p_2+\infty)$
else $\min_{\infty}(pf+1,p_2+\infty)=pf+1$. Therefore, from (C4.c2.2), (C4.c2.5), and (C4.c2.10) we have

(C4.c2.11) \text{fs'} = \text{fs}_0.
Hence, we need to prove

\[ C4.a.6 \] \quad \text{next}(\text{TA1}(X,p2,Ft,fs0)) \rightarrow (pf+1, sf (pf+1), sf (pf+1), cf) \text{ Ftf},

We prove \([C4.a.6]\) by case distinction over \( Ftf \).

\( Ftf = \text{done}(\text{false}) \)

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

In this case, from \((C4.c2.9)\) we get

\((C4.c2.12) \quad \text{next}(\text{TA1}(X,p2,Ft,fs1)) \rightarrow (pf+1, sf (pf+1), sf (pf+1), cf) \text{ done}(\text{false})\)

From \((C4.c2.12)\), by the definition of \( \rightarrow \) for forall we have

\((C4.c2.13) \quad \exists t \in \mathbb{N}, g \in \text{TFormula}, c \in \text{Context}: \quad (t,g,c) \in fs1' \land \vdash g \rightarrow (pf+1, sf (pf+1), sf (pf+1), c) \text{ done}(\text{false})\)

where

\((C4.c2.14) \quad fs1' = \)

if \( pf+1 > \infty \ p2 \) then

\( fs1 \)

else \( fs1 \cup \{ (pf+1, Ft, (cf.1[X \mapsto pf+1], cf.2[X \mapsto sf (pf+1)]) ) \} \).

Take \((t1,g1,c1)\) which is a witness for \((C4.c2.13)\). That means, we have

\((C4.c2.13') \quad (t1,g1,c1) \in fs1' \) and
\((C4.c2.13'') \quad g \rightarrow (pf+1, sf (pf+1), sf (pf+1), c1) \text{ done}(\text{false})\).

Assume first

\((C4.c2.15) \quad pf+1 > \infty \ p2, \) which from \((C4.c2.14)\) gives

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

\((C4.c2.16) \quad (t1,g1,c1) \in fs1.\)

To show \([C4.a.6]\), we need to prove

\([C4.a.7] \quad \exists t \in \mathbb{N}, g \in \text{TFormula}, c \in \text{Context}: \quad (t,g,c) \in fs0' \land \vdash g \rightarrow (pf+1, sf (pf+1), sf (pf+1), c) \text{ done}(\text{false})\)

where

\((C4.c2.17) \quad fs0' = \)

if \( pf+1 > \infty \ p2 \) then

\( fs0 \)

else \( fs0 \cup \{ (pf+1, Ft, (cf.1[X \mapsto pf+1], cf.2[X \mapsto sf (pf+1)]) ) \} \).

From \((C4.c2.15)\) and \((C4.c2.17)\), we have

\((C4.c2.18) \quad fs0' = fs0.\)

from \((C4.c2.16)\), by \((C4.c2.7)\), there exists \( g0 \in \text{TFormula} \) and \( fc1 \in \text{TFormulaCore} \) such that
(C4.c2.19) \( g_1 = \text{next}(fc_1) \)
(C4.c2.20) \( (t_1, g_0, c_1) \in fs_0 \)
(C4.c2.21) \( \vdash g_0 \rightarrow (pf, sf(pf), sf(pf), c_1) \) \( \text{next}(fc_1) \)

From (C4.c2.21), by the definition of \( \rightarrow \), there exists \( fc_0 \in T\text{FormulaCore} \) such that

(C4.c2.22) \( g_0 = \text{next}(fc_0) \).

From (C4.c2.13'), (C4.c2.19), and (C4.c2.13'') we know

(C4.c2.23) \( \vdash \text{next}(fc_1) \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), c_1) \) \( \text{done}(false) \).

From (C4.c2.21), (C4.c2.22), (C4.c2.23), by the induction hypothesis, we get

(C4.c2.24) \( \vdash g_0 \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), c_1) \) \( \text{done}(false) \).

From (C4.c2.18) and (C4.c2.20), we get

(C4.c2.25) \( (t_1, g_0, c_1) \in fs_0' \).

From (C4.c2.25) and (C4.c2.24), we get \([C4.a.7]\).

Now assume

(C4.c2.26) \( pf+1 \leq \infty \) \( p_2 \), which from (C4.c2.14) gives

(C4.c2.27) \( (t_1, g_1, c_1) \in fs_1 \cup \{(pf+1, Ft, (\text{cf.1}[X \rightarrow pf+1], \text{cf.2}[X \rightarrow sf(pf+1)])\}) \).

Recall:

To show \([C4.a.6]\), we need to prove

\([C4.a.7]\) \( \exists t \in \mathbb{N}, g \in T\text{Formula}, c \in \text{Context}: \)
\( (t, g, c) \in fs_0' \wedge \vdash g \rightarrow (pf+1, sf\downarrow (pf+1), sf(pf+1), c) \) \( \text{done}(false) \)

where

(C4.c2.17) \( fs_0' = \)
if \( pf+1 > \infty \) \( p_2 \) then
\( fs_0 \)
else
\( fs_0 \cup \{(pf+1, Ft, (\text{cf.1}[X \rightarrow pf+1], \text{cf.2}[X \rightarrow sf(pf+1)])}) \).

From (C4.c2.26) and (C4.c2.17), we have

(C4.c2.28) \( fc_0' = fs_0 \cup \{(pf+1, Ft, (\text{cf.1}[X \rightarrow pf+1], \text{cf.2}[X \rightarrow sf(pf+1)])}) \).

If \( (t_1, g_1, c_1) \in fs_1 \), the proof proceeds as for the case \( pf+1 > \infty \) \( p_2 \) above.

Consider

(C4.c2.29) \( (t_1, g_1, c_1) = (pf+1, Ft, (\text{cf.1}[X \rightarrow pf+1], \text{cf.2}[X \rightarrow sf(pf+1)])}) \).
From (C4.c2.28) and (C4.c2.29) we have

(C4.c2.30) \((t1,g1,c1) \in fc'\)

From (C4.c2.30) and (C4.c2.13'') we get \([C4.a.7]\).

This finishes the proof of the case \(Ftf=\text{done(false)}\).

Ftf = done(true). The case \(p1=\infty\) 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\(\downarrow\)(pf+1),sf(pf+1),cf) done(true),

which by Def. \(\rightarrow\) means, we need to prove

\([C4.a.true.2]\) \(\exists t \in N, g \in T\text{Formula}, c \in \text{Context:}
\quad (t,g,c) \in fs0 \land \vdash g \rightarrow (pf+1,sf\(\downarrow\)(pf+1),sf(pf+1),c)\) done(false)

\([C4.a.true.3]\) fs01 = \(\emptyset \land pf+1 \geq \infty\ p2\),

where

(C4.c2.true.1) fs00 =
\quad \text{if } pf+1 > \infty\ p2 \text{ then } fs0
\quad \text{else } fs0 \cup \{(pf+1,Ft,((cf.1[X\mapsto pf+1],c.2[X\mapsto sf(pf+1)]))\})

(C4.c2.true.2) fs01 =
\quad \{(t,\text{next}(fc),c) \in T\text{Instance} | \exists g \in T\text{Formula:} (t,g,c) \in fs0 \land \vdash g \rightarrow (pf+1,sf\(\downarrow\)(pf+1),sf(pf+1),c) \text{ 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\(\downarrow\)(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 T\text{Formula}, c \in \text{Context:}
\quad (t,g,c) \in fs1 \land \vdash g \rightarrow (pf+1,sf\(\downarrow\)(pf+1),sf(pf+1),c)\) done(false)

(C4.c2.true.5) fs11 = \(\emptyset \land pf+1 \geq \infty\ p2\)

where

(C4.c2.true.6) fs10 =
\quad \text{if } pf+1 > \infty\ p2 \text{ then } fs1
\quad \text{else } fs1 \cup \{(pf+1,Ft,((cf.1[X\mapsto pf+1],c.2[X\mapsto sf(pf+1)]))\})

(C4.c2.true.7) fs11 =
\quad \{(t,\text{next}(fc),c) \in T\text{Instance} | \exists g \in T\text{Formula:} (t,g,c) \in fs1 \land \vdash g \rightarrow (pf+1,sf\(\downarrow\)(pf+1),sf(pf+1),c) \text{ next}(fc) \}
Recall the relationship between \( fs_0 \) and \( fs_1 \):

(C4.c2.7) \( fs_1 = \{ (t,\text{next}(fc),c) \in T\text{Instance} \mid \exists g \in T\text{Formula}: (t,g,c) \in fs_0 \land \vdash g \rightarrow (pf,\text{sf} \downarrow pf,\text{sf}(pf),c) \text{ next}(fc) \} \)

From (C4.c2.true.6), (C4.c2.true.7), and (C4.c2.true.5) we know that

(C4.c2.true.8) \( \neg \exists fc \in T\text{FormulaCore}: \text{Ft} \rightarrow (pf+1,\text{sf} \downarrow (pf+1),\text{sf}(pf+1),c_0) \text{ next}(fc) \).

Now assume by contradiction that for some \((t_0,g_0,c_0) \in fs_0\) we have

(C4.c2.true.9) \( \exists fc \in T\text{FormulaCore}: g_0 \rightarrow (pf+1,\text{sf} \downarrow (pf+1),\text{sf}(pf+1),c_0) \text{ next}(fc) \).

From (C4.c2.true.9), by Lemma 6, there exist \( fc_0 \in T\text{FormulaCore} \) such that

(C4.c2.true.10) \( g_0 \rightarrow (pf,\text{sf} \downarrow (pf),\text{sf}(pf),c_0) \text{ next}(fc_0) \).

From (C4.c2.true.9) by (C4.c2.7) we have that there exists \( fc_0 \in T\text{FormulaCore} \) such that

(C4.c2.true.11) \( (t_0,\text{next}(fc_0),c_0) \in fs_1 \).

From (C4.c2.true.11) by (C4.c2.true.6) we get

(C4.c2.true.12) \( (t_0,\text{next}(fc_0),c_0) \in fs_{10} \).

From (C4.c2.true.12) by (C4.c2.true.7), (C4.c2.true.5), (C4.c2.true.4), we get

(C4.c2.true.13) \( \text{next}(fc_0) \rightarrow (pf+1,\text{sf} \downarrow (pf+1),\text{sf}(pf+1),c_0) \text{ done(true)} \).

From (C4.c2.true.10) and (C4.c2.true.13), by the induction hypothesis, we get

(C4.c2.true.14) \( g_0 \rightarrow (pf+1,\text{sf} \downarrow (pf+1),\text{sf}(pf+1),c_0) \text{ done(true)} \).

But (C4.c2.true.14) contradicts (C4.c2.true.9). Hence, we know that for all \((t,g,c) \in fs_0\)

(C4.c2.true.15) \( \neg \exists fc \in T\text{FormulaCore}: g \rightarrow (pf+1,\text{sf} \downarrow (pf+1),\text{sf}(pf+1),c) \text{ next}(fc) \).

From (C4.c2.true.8) and (C4.c2.true.15) we know that for all \((t,g,c) \in fs_{00}\)

(C4.c2.true.16) \( \neg \exists fc \in T\text{FormulaCore}: g \rightarrow (pf+1,\text{sf} \downarrow (pf+1),\text{sf}(pf+1),c) \text{ next}(fc) \).

From (C4.c2.true.16) we get

(C4.c2.true.17) \( fs_{01} = \emptyset \).

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
Recall that in (C4.c2.6) we have

(C4.c2.6) \( \neg \exists t \in \mathbb{N}, g \in TFormula, c \in Context:
\quad \langle t, g, c \rangle \in fs_00 \land \vdash g \rightarrow (pf, sf, pf, c) \text{ done(false)} \)

Hence, for no \( \langle t, g, c \rangle \in fs_00 \) we have \( g \rightarrow (pf, sf, pf, c) \text{ done(false)} \).
It proves \([C4.a.true.2]\).

Ftf is a 'next' formula.

Let \( Ftf = \text{next}(TA1(X, p2, Ft, fs_2)) \) for some \( fs_2 \).
Then from \([C4.a.6]\) and \((C4.c2.11)\),
we need to prove

\[ [C4.a.next.1] \quad \text{next}(TA1(X, p2, Ft, fs_0)) \rightarrow (pf+1, sf, pf, c)
\quad \text{next}(TA1(X, p2, Ft, fs_2)) \]

To prove \([C4.a.next.8]\), we define

(C4.c2.next.1) \( fs_00 := \)
if \( pf+1 >\infty p_2 \) then
\( fs_0 \)
else \( fs_0 \cup \{(pf+1, Ft, (cf.1[X\rightarrow pf+1], cf.2[X\rightarrow sf(pf+1)]))\} \)

(C4.c2.next.2) \( fs_01 := \)
\{ \( (t, \text{next}(fc), c) \in TInstance \mid \)
\exists g \in TFormula: \( (t, g, c) \in fs_00 \land
\quad \vdash g \rightarrow (pf+1, sf, pf, c) \text{ next(fc)} \) \}

and prove

\[ [C4.a.next.2] \quad \neg \exists t \in \mathbb{N}, g \in \text{FormulaStep}, c \in Context:
\quad (t, g, c) \in fs_00 \land \vdash g \rightarrow (pf+1, sf, pf, c) \text{ done(false)} \]
\[ [C4.a.next.3] \quad \neg (fs_01 = \emptyset \land pf+1 \geq\infty p_2) \]

On the other hand, from \((C4.c2.9)\) we know

(C4.c2.next.3) \( \text{next}(TA1(X, p2, Ft, fs_1)) \rightarrow (pf+1, sf, pf, c)
\quad \text{next}(TA1(X, p2, Ft, fs_2)). \)

From \((C4.c2.next.3)\), by Def. \( \rightarrow \), we know

(C4.c2.next.4) \( \neg \exists t \in \mathbb{N}, g \in \text{FormulaStep}, c \in Context:
\quad (t, g, c) \in fs_10 \land \vdash g \rightarrow (pf+1, sf, pf, c) \text{ done(false)} \)
\[ (C4.c2.next.5) \quad \neg (fs_11 = \emptyset \land pf+1 \geq\infty p_2) \]

where

(C4.c2.next.6) \( fs_{10} = \)
if $pf+1 > \infty p2$ then
\[
fs1
\]
else $fs1 \cup \{(pf+1,Ft,(cf.1[X\rightarrow pf+1],cf.2[X\rightarrow sf(pf+1)]))\}$

(C4.c2.next.7) $fs11 = \{(t,next(fc),c) \in TInstance | \exists g \in TFormula: (t,g,c) \in fs10 \land \vdash g \rightarrow ((pf+1,sf\downarrow(pf+1),sf(pf+1),c) next(fc))\}$

Recall the relation between $fs0$ and $fs1$:

(C4.c2.7) $fs1 = \{(t,next(fc),c) \in TInstance | \exists g \in TFormula: (t,g,c) \in fs0 \land \vdash g \rightarrow (pf,sf\downarrow pf,sf(pf),c) next(fc)\}$

By (C4.c2.6) and (C4.c2.next.1), to prove [C4.a.next.2], it suffices to prove that

[C4.a.next.4] $\vdash Ft \rightarrow (pf+1,sf\downarrow(pf+1),sf(pf+1),(cf.1[X\rightarrow pf+1],cf.2[X\rightarrow sf(pf+1)]))$

done(false) does not hold.

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) $pf+1 \geq \infty p2$

and prove

[C4.a.next.5] $fs01 \neq \emptyset$.

From (C4.c2.next.8) and (C4.c2.next.5) we know

(C4.c2.next.9) $fs11 \neq \emptyset$.

From (C4.c2.next.9), there exist $(t_1,g_1,c_1) \in fs10$ and $fc1 \in TFormulaCore$ such that

(C4.c2.next.9) $\vdash g_1 \rightarrow (pf+1,sf\downarrow(pf+1),sf(pf+1),c_1) next(fc1)$.

According to (C4.c2.next.6), $(t_1,g_1,c_1) \in fs10$ means either $(t_1,g_1,c_1) \in fs1$ or $(t_1,g_1,c_1) = (pf+1,Ft,(cf.1[X\rightarrow pf+1],cf.2[X\rightarrow sf(pf+1)]))$

First assume $(t_1,g_1,c_1) \in fs1$.

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

By (C4.c2.7), it means that there exist $(t_0,g_0,c_0) \in fs0$ and $fc0 \in TFormulaCore$ such that

(C4.c2.next.10) $\vdash g_0 \rightarrow (pf,sf\downarrow pf,sf(pf),c_0) next(fc0)$

(C4.c2.next.11) $g_1 = next(fc0)$

Moreover, $g_0$ is a 'next' formula.

(C4.c2.next.12) $g_0 = next(fc)$ for some $fc \in TFormulaCore$.

Besides, from (C4.c2.7) one can see that
(C4.c2.next.13) \( c_0 = c_1 \).

Hence, from (C4.c2.next.9--13) we have

(\text{(C4.c2.next.14)}) \( \text{next}(fc) \rightarrow (pf, sf, pf(pf), c_0) \) \( \text{next}(fc_0) \)
(\text{(C4.c2.next.15)}) \( \text{next}(fc_0) \rightarrow (pf+1, sf(pf+1), sf(pf+1), c_0) \) \( \text{next}(fc_1) \)

From (C4.c2.next.14) and (C4.c2.next.15), by the induction hypothesis, we obtain that

(\text{(C4.c2.next.16)}) \( \text{next}(fc) \rightarrow (pf+1, sf(pf+1), sf(pf+1), c_0) \) \( \text{next}(fc_1) \)

Hence, we got that for \( (t_0, g_0, c_0) \in fs_0 \) and \( fc_1 \in TFormulaCore \)

(\text{(C4.c2.next.17)}) \( g_0 \rightarrow (pf+1, sf(pf+1), sf(pf+1), c_0) \) \( \text{next}(fc_1) \).

By definition (C4.c2.next.1) of \( fs_0 \), we have \( (t_0, g_0, c_0) \in fs_0 \).

Now assume \( (t_1, g_1, c_1) = (pf+1, F_t, (cf.1[X \mapsto pf+1], cf.2[X \mapsto sf(pf+1)]) \)

Trivially, by definition (C4.c2.next.1) of \( fs_0 \), we have \( (t_1, g_1, c_1) \in fs_0 \).

Hence, in both cases we found a triple

(\text{(C4.c2.next.18)}) \( (t, g, c) \in fs_0 \)

such that

(\text{(C4.c2.next.19)}) \( g \rightarrow (pf+1, sf(pf+1), sf(pf+1), c) \) \( \text{next}(fc_1) \)

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 \( F_t \) is a 'next' formula.

This finishes the proof of C4.c2.

This finishes the proof of C4.

This finishes the proof of Lemma 8.
A.11 Lemma 9: Soundness of Bound Analysis

∀ re∈RangeEnv, e∈Environment, p∈N, s∈Stream, B∈Bound, l,u∈Z∞:
re ⊨ B : (l,u) ∧ dom(e) = dom(re) ∧
(∀Y∈dom(e): re(Y).1 +i p ≤i e(Y) ≤i re(Y).2 +i p) ⇒
let c := (e,{X, s(e(X)) | X∈dom(e)}):
l +i p ≤i T(B)(c) ≤i u +i p

Proof
-----

Denote
Φ(B) ⇔
∀ re∈RangeEnv, e∈Environment, p∈N, s∈Stream, l,u∈Z∞:
re ⊨ B : (l,u) ∧ dom(e) = dom(re) ∧
(∀Y∈dom(e): re(Y).1 +i p ≤i e(Y) ≤i re(Y).2 +i p) ⇒
let c := (e,{X, s(e(X)) | X∈dom(e)}):
l +i p ≤i T(B)(c) ≤i u +i p

Then we need to prove


We prove [1] by structural induction over B.

(a). B=0.
-------
We take ref, ef, pf, sf, Bf, lf, uf arbitrary but fixed, assume

(a1) ref ⊨ B : (lf,uf)
(a2) dom(ef) = dom(ref)
(a3) ∀Y∈dom(e): ref(Y).1 +i pf ≤i ef(Y) ≤i ref(Y).2 +i pf,
(a4) c = (ef,{X, sf(ef(X)) | X∈dom(ef)})

and prove

[a5] lf +i pf ≤i T(B)(c) ≤i uf +i pf

By the translation, we have

(a6) T(B)(c) = 0, when B=0.

Therefore, we need to prove

[a7] lf +i pf ≤i 0 ≤i uf +i pf

By the analysis rules, we have

(a8) ref ⊨ B : (-∞,0), when B=0.

That means, from (a8) and (a1) we need to consider the case, when
(a9) $lf = -\infty$
(a10) $uf = 0$.

From the definition of $+i$, we have $-\infty + n = -\infty$. Hence, from (a9,a10) we need to prove

[a11] $-\infty \leq i 0 \leq i 0$

which obviously holds. Hence, the case (a) is proved.

(b). $B = \infty$.
----------

We take $ref, ef, pf, sf, Bf, lf, uf$ arbitrary but fixed, assume

(b1) $ref \vdash B : (lf, uf)$
(b2) $\text{dom}(ef) = \text{dom}(ref)$
(b3) $\forall Y \in \text{dom}(e): ref(Y).1 +i pf \leq i ef(Y) \leq i ref(Y).2 +i pf,$
(b4) $c = (ef,\{X, sf(ef(X)) | X \in \text{dom}(ef)\})$

and prove

[b5] $lf +i pf \leq i T(B)(c) \leq i uf +i pf$

By the translation, we have

(b6) $T(B)(c) = \infty$, when $B=\infty$.

Therefore, we need to prove

[b7] $lf +i pf \leq i \infty \leq i uf +i pf$

By the analysis rules, we have

(b8) $ref \vdash B : (\infty, \infty)$, when $B=\infty$.

That means, from (b7) and (b1) we need to consider the case, when

(b9) $lf = \infty$
(b10) $uf = \infty$.

Hence, from (b9,b10) we need to prove

[b11] $\infty \leq i \infty \leq i \infty$

which obviously holds. Hence, the case (b) is proved.

(c). $B=X$.
----------

We take $ref, ef, pf, sf, Bf, lf, uf$ arbitrary but fixed, assume

(c1) $ref \vdash B : (lf, uf)$
(c2) $\text{dom}(ef) = \text{dom}(ref)$
\( \forall Y \in \text{dom}(e): \text{ref}(Y).1 + i \text{ pf} \leq i \text{ ef}(Y) \leq i \text{ ref}(Y).2 + i \text{ pf}, \)

\( \text{and prove} \)

\[ [c5] \text{l}f + i \text{ pf} \leq i \text{T}(B)(c) \leq i \text{u}f + i \text{ pf} \]

By the analysis rules, we have two subcases:

\[(c.\text{case1}) X \not\in \text{dom}(\text{ref})\]

\[ \text{In this case, by (c2) and (c3) we have } X \not\in \text{dom}(ef) = \text{dom}(c.1). \]

By the translation, we have

\[(c.\text{case1.1}) \text{T}(X)(c) = 0, \text{ when } B = X \text{ and } X \not\in \text{dom}(c.1). \]

Therefore, we need to prove

\[ [c.\text{case1.2}] \text{l}f + i \text{ pf} \leq i 0 \leq i \text{u}f + i \text{ pf}. \]

By the analysis rules, in this subcase we have

\[(c.\text{case1.3}) \text{ref} \vdash X : (-\infty,0), \text{ when } B = X \text{ and } X \not\in \text{dom}(\text{ref}). \]

From (c.\text{case1.3}) and (c1) we get

\[(c.\text{case1.4}) \text{l}f = -\infty \]
\[(c.\text{case1.5}) \text{u}f = 0. \]

Therefore, to prove \([c.\text{case1.2}], \text{we need to prove} \)

\[ [c.\text{case1.3}] -\infty + i \text{ pf} \leq i 0 \leq i 0 + i \text{ pf}, \]

which holds, because \(-\infty + i \text{ pf} = -\infty. \) It proves the subcase (c.\text{case1}).

\[(c.\text{case2}) X \in \text{dom}(\text{ref})\]

\[ \text{In this case, by (c2) and (c3) we have } X \in \text{dom}(ef) = \text{dom}(c.1). \]

By the translation, we have

\[(c.\text{case2.1}) \text{T}(X)(c) = c.1(X) = \text{ef}(X), \text{ when } B = X \text{ and } X \in \text{dom}(c.1). \]

Therefore, we need to prove

\[ [c.\text{case2.2}] \text{l}f + i \text{ pf} \leq i \text{ ef}(X) \leq i \text{ u}f + i \text{ pf} \]

By the analysis rules, in this subcase we have

\[(c.\text{case2.3}) \text{ref} \vdash X : \text{ref}(X), \text{ when } B = X \text{ and } X \in \text{dom}(\text{ref}). \]
From (c.case2.3) and (c1) we get

(c.case2.4) lf = ref(X).1
(c.case2.5) uf = ref(X).2

Therefore, to prove [c.case2.2], we need to prove

[c.case2.3] ref(X).1 +i pf ≤ ref(X) ≤ ref(X).2 +i pf,

which follows from (c3). The case (c.case2) is proved.

d. B=B0+N, for B0∈Bound and N∈N

We take ref, ef, pf, sf, Bf, lf, uf arbitrary but fixed, assume

(d1) ref ⊢ B : (lf, uf)
(d2) dom(ef) = dom(ref)
(d3) ∀Y∈dom(e): ref(Y).1 +i pf ≤ ref(Y) ≤ ref(Y).2 +i pf,
(d4) c = (ef, {X, sf(ef(X)) | X∈dom(ef)})

and prove

[d5] lf +i pf ≤ T(B)(c) ≤ uf +i pf.

By the translation, we have

(d6) T(B)(c) = T(B0)(c)+N, when B=B0+N.

Assume that

(d7) ref ⊢ B0 : (l0, u0).

Then, by the analysis rules, since ref ⊢ B+N : (l0+i[N], u0+i[N]), we have from (d7) and (d1):

(d8) lf = l0+i[N]
(d9) uf = u0+i[N]

and we need to prove

[d10] l0 +i [N] +i pf ≤ T(B0)(c)+[N] ≤ u0 +i [N] +i pf.

By the induction hypothesis for B0 we have

(d11) l0 +i pf ≤ T(B0)(c) ≤ u0 +i pf,

which implies [d10]. It proves the case (d).

(e) B=B0-N, for B0∈Bound and N∈N.

Similar to the case (d).
A.12 Lemma 10: Invariant Lemma for Universal Formulas

∀X ∈ Variable, b1 ∈ BoundValue, b2 ∈ BoundValue, f ∈ TFormulaCore:
∀n ∈ N: n ≥ 1 ⇒ forall(n, X, b1, b2, next(f))

Predicates
-------

forall ⊆ N × Variable × BoundValue × BoundValue × TFormula:
forall(n, X, b1, b2, f):
⇔
∀p ∈ N, s ∈ Stream, e ∈ Environment, g ∈ TFormula:
(⊢ next(TA(X, b1, b2, f)) → *(n, p, s, e) g) ⇒
let c = (e, {(Y, s(e(Y))) | Y ∈ dom(e)}) :
let p0 = p + n, p1 = b1(c), p2 = b2(c) :
( n = 1 ∧ (p1 = ∞ ∨ p1 >∞ p2) ∧ g = done(true) )
∨
( n ≥ 1 ∧ p1 ≠ ∞ ∧ p0 ≤ p1 ∧ p0 ≤ p1 ∧ g = next(TA0(X, p1, p2, f)) )
∨
( n ≥ 1 ∧ p1 ≠ ∞ ∧ p1 ≤ ∞ ∧ p0 > p1 ∧
( (∃b ∈ Bool: g = done(b)) ∨
(∃gs ∈ P(TInstance): (gs ≠ ∅ ∨ p + n ≤ ∞ p2) ∧
forallInstances(X, p, p0, p1, p2, s, e, gs) ∧
 g = next(TA1(X, p2, f, gs)))
) )
)

forallInstances ⊆ Variable × N × N × N × N∞ × TFormula × Stream × Environment × P(TInstance):
forallInstances(X, p, p0, p1, p2, s, e, gs) :
⇔
∀t ∈ N, g ∈ TFormula, c0 ∈ Context: (t, g, c0) ∈ gs ⇒
(∀t1 ∈ N, g1 ∈ TFormula, c1 ∈ Context:
 (t1, g1, c1) ∈ gs ∧ t = t1 ⇒ (t, g, c0) = (t1, g1, c1)) ∧
(∃gc ∈ TFormulaCore: g = next(gc)) ∧
c0.1 = e[X → t] ∧ c0.2 = {(Y, s(c0.1(Y))) | Y ∈ dom(e) ∨ Y = X} ∧
p1 ≤ t ≤ ∞ min∞(p0 − 1, p2) ∧
⊢ f → *(p0 − max(p, t), max(p, t), s, c0.1) g

Proof
------

Let X ∈ Variable, b1 ∈ BoundValue, b2 ∈ BoundValue, f ∈ TFormulaCore be arbitrary fixed.
We prove
∀n ∈ N: n ≥ 1 ⇒ forall(n, X, b1, b2, next(f))
by induction on n ≥ 1.
Let $n \in \mathbb{N}$ be arbitrary but fixed and assume

(0) $n \geq 1$.

Induction Base

We show

[1] $\forall (1, X, b1, b2, next(f))$

d. i.e. by the definition of "forall" for arbitrary but fixed $p \in \mathbb{N}$, $s \in \text{Stream}$, $e \in \text{Environment}$, $g \in T\text{Formula}$ under the assumptions

(1) $\vdash next(TA(X, b1, b2, f)) \rightarrow^*(1, p, s, e) g$
(2) $c := (e, \{(Y, s(e(Y))) \mid Y \in \text{dom}(e))\}
(3) $p0 := p+1$
(4) $p1 := b1(c)$
(5) $p2 := b2(c)$

the goal

[2] $(p1 = \infty \lor p1 > \infty p2) \land g=\text{done}(true)$

$\lor$

$(p1 \neq \infty \land p1 \leq \infty p2 \land p0 \leq p1 \land g=next(TA0(X, p1, p2, next(f))))$

$\lor$

$(p1 \neq \infty \land p1 \leq \infty p2 \land p0 > p1 \land$

$(\exists b \in \text{Bool}: g=\text{done}(b)) \lor$

$(\exists gs \in P(T\text{Instance}): (gs \neq \emptyset \lor p+n \leq \infty p2) \land$

$\forall \text{Instances}(X, p, p0, p1, p2, next(f), s, e, gs) \land$

$g = next(TA1(X, p2, next(f), gs)))$

$\lor$

From (1), (2) and the rules for $\rightarrow^*$, we know for some $Ft' \in T\text{Formula}$

(6) $\vdash next(TA(X, b1, b2, next(f))) \rightarrow(p, s\uparrow p(s(p), c) Ft'$
(7) $\vdash Ft' \rightarrow^*(0, p+1, s, e) g$

From (6), (7) and the rules for $\rightarrow^*$, we know

(8) $\vdash next(TA(X, b1, b2, next(f))) \rightarrow(p, s\uparrow p(s(p), c) g$

From (4), (5), (8) and the rules for $\rightarrow$, we have two cases.

Case 1

------
From (20) and (21), we have [2].

Case 2

(30) $p_1 \neq \infty \land p_1 \leq \infty p_2$

(31) $\vdash \text{next(TA0}(X,p_1,p_2,\text{next}(f))) \rightarrow (p,s,p,s(p),c)$

We proceed by case distinction.

Case 2.1

(40) $p_0 \leq \infty p_1$

From (30) and (40), to show [2] it suffices to show

[2.a] $g = \text{next(TA0}(X,p_1,p_2,\text{next}(f)))$

From (31) and the fact that the rule system for $\rightarrow$ is deterministic, to show [2.a], it suffices to show

[2.b] $\vdash \text{next(TA0}(X,p_1,p_2,\text{next}(f))) \rightarrow (p,s,p,s(p),c) \text{ next(TA0}(X,p_1,p_2,\text{next}(f)))$

which holds from (3), (40) and the rules for $\rightarrow$.

Case 2.2

(70) $p_0 > \infty p_1$

From (30) and (70), to show [2] it suffices to show

[2.a] $(\exists b \in \text{Bool}: g = \text{done}(b)) \lor$

$(\exists gs \in \text{P(TInstance)}: (gs \neq \emptyset \lor p+1 \leq \infty p_2) \land$

forallInstances$(X,p,p_0,p_1,p_2,\text{next}(f),s,e,gs) \land$

$g = \text{next(TA1}(X,p_2,\text{next}(f),gs)))$

We define

(72) $fs := \{(p_x,\text{next}(f),(c.1[X \mapsto -\rightarrow p_x],c.2[X \mapsto s[p(x-\rightarrow p_x)])) |$

$p_1 \leq p_x < \infty \min\infty(p,p_2+\infty 1)\}$

From (3), (31), (70), (72), and the rules for $\rightarrow$, we know

(73) $\vdash \text{next(TA1}(X,p_2,\text{next}(f),fs)) \rightarrow (p,s,p,s(p),c)$

From (72), we know with $|s[p]=p$

(74) $fs = \{(p_x,\text{next}(f),(c.1[X \mapsto p_x],c.2[X \mapsto s[p(x)])) | p_1 \leq p_x < \infty \min\infty(p,p_2+\infty 1)\}$

and thus

(74') $fs = \{(p_x,\text{next}(f),(c.1[X \mapsto p_x],c.2[X \mapsto s(p)])) | p_1 \leq p_x < \infty \min\infty(p,p_2+\infty 1)\}$
To show [2.a], we assume 

(75) \( \neg(\exists b \in \text{Bool}: g=\text{done}(b)) \)

and show

[2.b] \( \exists gs \in \text{P}(\text{TInstance}): (gs \neq \emptyset \lor p+1 \leq \infty \ p2) \land \forall \text{forallInstances}(X,p,p0,p1,p2,next(f),s,e,gs) \land \) 
\( g = \text{next}(\text{TA1}(X,p2,next(f),gs)) \)

From (73), (75), and the rules for \( \rightarrow \), we know

(76) \( \text{fs0} := \text{if } p \succ \infty \ p2 \text{ then } fs \text{ else } fs \cup \{(p, next(f), (c.1[X\mapsto p], c.2[X\mapsto s(p)]))\} \)

(77) \( \neg\exists t \in \mathbb{N}, g \in \text{TFormula}, c \in \text{Context}: (t, g, c) \in \text{fs0} \land \vdash g \rightarrow (p, s, p, s(p), c) \) done(false)

(78) \( \text{fs1} := \{(t, next(fc), c) \in \text{TInstance} | \exists g \in \text{TFormula}: (t, g, c) \in \text{fs0} \land \) 
\( \vdash g \rightarrow (p, s, p, s(p), c) \) next(fc) \} \)

(79) \( \neg(\text{fs1} = \emptyset \land p \geq \infty \ p2) \)

(80) \( g = \text{next}(\text{TA1}(X,p2,next(f),\text{fs1})) \)

From (80), to show [2.b], it suffices to show

[2.b.1] \( \text{fs1} \neq \emptyset \lor p+1 \leq \infty \ p2 \)

[2.b.2] \( \forall \text{forallInstances}(X,p,p0,p1,p2,next(f),s,e,\text{fs1}) \)

From \( p \in \mathbb{N} \) and (79), we have [2.b.1].

To show [2.b.2], we proceed by case distinction.

Case 2.2.1

----------

(100) \( p \succ \infty \ p2 \)

From (76) and (100), we know

(101) \( \text{fs0} = fs \)

From (74), (78), (101), we know

(102) \( \text{fs1} = \{(t, next(fc), (c.1[X\mapsto t], c.2[X\mapsto s(p(t)])) | p1 \leq t < \infty \min(p, p2+\infty 1) \land \) 
\( \vdash next(f) \rightarrow (p, s, p(t), (c.1[X\mapsto t], c.2[X\mapsto s(p(t)])) \) next(fc) \} \}

To show [2.b.2], from the definition of "forallInstances", we have to show for arbitrary but fixed \( t \in \mathbb{N}, g \in \text{TFormula}, c0 \in \text{Context} \) such that

(120) \( (t, g, c0) \in \text{fs1} \)

the following:

[2.b.2.1] \( \forall t \in \mathbb{N}, g1 \in \text{TFormula}, c1 \in \text{Context}: \) 
\( (t1, g1, c1) \in \text{fs1} \land t=t1 \Rightarrow (t, g, c0)=(t1, g1, c1) \)

[2.b.2.2] \( \exists gc \in \text{TFormulaCore}: g=\text{next}(gc) \)

[2.b.2.3] \( c0.1=e[X \mapsto t] \)
From (102) and the fact that the rule system for $\rightarrow$ is deterministic, we have [2.b.2.1].

From (102) and (120), we have for some $fc \in TFormulaCore$

(121) $g = \text{next}(fc)$

(122) $c0 = (c.1[X \rightarrow t], c.2[X \rightarrow s \downarrow p(t)])$

(123) $p1 \leq t < \infty \min \infty (p0, p2 + \infty 1)$

(124) $\vdash \text{next}(f) \rightarrow (p, s \downarrow p, s(p), c0)$ next(fc)

From (121), we have [2.b.2.2].

From (122) and (2), we have [2.b.2.3].

To show [2.b.2.5], from (3), it suffices to show

[2.b.2.5.1] $p1 \leq t$
[2.b.2.5.2] $t \leq p$
[2.b.2.5.3] $t \leq \infty p2$

which all three follow from (123).

We now show [2.b.2.4]. From (122), we know

(125) $c0.1 = c.1[X \rightarrow t]$
(126) $c0.2 = c.2[X \rightarrow s \downarrow p(t)]$

From (123), we know

(127) $t < p$

From (126) and (127), we have

(128) $c0.2 = c.2[X \rightarrow s(t)]$

From (125) and (128), to show [2.b.2.4], it suffices to show

[2.b.2.4.a] $c.2[X \rightarrow s(t)] = \{(Y, s(c.1[X \rightarrow t](Y))) \mid Y \in \text{dom}(e) \lor Y = X\}$

For this it suffices to show for arbitrary $Y$ with $Y \in \text{dom}(e) \lor Y = X$

[2.b.2.4.b] $c.2[X \rightarrow s(t)](Y) = s(c.1[X \rightarrow t](Y))$

Case $Y = X$:

We have

(130) $c.2[X \rightarrow s(t)](Y) = s(t)$
(131) $s(c.1[X \rightarrow t](Y)) = s(t)$
and thus [2.b.2.4.b].

Case $Y \neq X$:

We have

\begin{align}
(132) & \quad Y \in \text{dom}(e) \\
(133) & \quad c.2[X \mapsto s(t)](Y) = c.2(Y) \\
(134) & \quad s(c.1[X \mapsto t](Y)) = s(c.1(Y)) 
\end{align}

From (2) and (132), we have

\begin{align}
(135) & \quad c.1 = e \\
(136) & \quad c.2(Y) = s(e(Y)) 
\end{align}

From (133), (134), (135), (136), we have [2.b.2.4.b].

To show [2.b.2.6], by (3), it suffices to show

\[ [2.b.2.6.a] \vdash \text{next}(f) \rightarrow^{\ast}(p_0-\text{max}(p,t),\text{max}(p,t),s,c_0.1) \ g \]

From (123), we know

\[ (140) \quad \text{max}(p,t) = p \]

From (3) and (140), it suffices to show

\[ [2.b.2.6.b] \vdash \text{next}(f) \rightarrow^{\ast}(1,p,s,c_0.1) \ g \]

From (2), (125), (128), we know

\[ (141) \quad c_0 = (c_0.1, \{ (Y,s(c_0.1(Y)) \mid Y \in \text{dom}(c_0.1) \}) \]

From (141) and the definition of $\rightarrow^{\ast}$, it suffices to show

\[ [2.b.2.6.c] \vdash \text{next}(f) \rightarrow(p,\downarrow p,s(p),c_0) \ g \]

which follows from (121) and (124).

Case 2.2.2

----------

\[ (200) \quad p \leq \infty \ p_2 \]

To show [2.b.2], from the definition of "forallInstances", we have to show for arbitrary but fixed $t \in \mathbb{N}, g \in T\text{Formula}, c_0 \in \text{Context}$ such that

\[ (201) \quad (t,g,c_0) \in fs_1 \]

the following:

\[ [2.b.2.1] \forall t_1 \in \mathbb{N}, g_1 \in T\text{Formula}, c_1 \in \text{Context} : \]

\[ (t_1,g_1,c_1) \in fs_1 \land \ t = t_1 \Rightarrow (t,g,c_0) = (t_1,g_1,c_1) \]

\[ [2.b.2.2] \exists g_c \in T\text{FormulaCore} : g = \text{next}(g_c) \]

\[ [2.b.2.3] \quad c_0.1 = e[X \mapsto t] \]

135
We define

\[(202) \ c_1 := (c_{.1}[X\mapsto p], c_{.2}[X\mapsto s(p)])\]

From (76), (200), (202), we know

\[(203) \ fs_0 = fs \cup \{(p, \text{next}(f), c_1)\}\]

From (78) and (203), we know

\[(204) \ fs_1 = \{(t, \text{next}(fc), c) \in TInstance \mid \exists g \in TFormula: (t, g, c) \in fs \land \]
\[\vdash g \rightarrow (p, s|p, s(p), c) \text{ next}(fc) \lor \]
\[(t = p \land c = c_1 \land \vdash \text{next}(f) \rightarrow (p, s|p, s(p), c_1) \text{ next}(fc))\}\]

From (74'), (204), and the fact that the rule system is deterministic, we have [2.b.2.1].

From (201) and (204), we have [2.b.2.2].

It thus remains to show [2.b.2.3-6].

From (201), (202) and (204) we have two cases:

Case 2.2.2.1

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

There exists some fc \in TFormulaCore such that

\[(220) \ t = p\]
\[\ (221) \ g = \text{next}(fc)\]
\[\ (222) \ \vdash \text{next}(f) \rightarrow (p, s|p, s(p), c_0) \text{ next}(fc)\]
\[\ (223) \ c_{.1} = c_{.1}[X\mapsto p]\]
\[\ (224) \ c_{.2} = c_{.2}[X\mapsto s(p)]\]

From (2), (223), (224), we have [2.b.2.3].

From (2), (222), (223), (224), we have [2.b.2.4].

From (3) and (70) and (220), we have

\[(230) \ p_1 \leq \infty \ t\]

From (200) and (220), we have

\[(231) \ t \leq \infty \ p_2\]

From (3) and (220), we have

\[(232) \ t < \infty \ p_0\]

From (230), (231), (232), we have [2.b.2.5].
To show \([2.b.2.6]\), from (3) and (220), it suffices to show

\[ [2.b.2.6.a] \vdash \text{next}(f) \rightarrow \ast(1, p, s, c0.1) g \]

From the definition of \(\rightarrow\ast\), (2), (223), (224), it suffices to show

\[ [2.b.2.6.b] \vdash \text{next}(f) \rightarrow \langle p, s\downarrow, s(p), c0 \rangle g \]

which follows from (221) and (222).

Case 2.2.2.2
-------------
There exist some \(fc \in TFormulaCore\) and \(g0 \in TFormula\) such that

\[
\begin{align*}
(240) & \quad g = \text{next}(fc) \\
(241) & \quad (t, g0, c0) \in fs \\
(242) & \quad \vdash g0 \rightarrow \langle p, s\downarrow, s(p), c0 \rangle g 
\end{align*}
\]

From (74') and (241), we know

\[
\begin{align*}
(243) & \quad g0 = \text{next}(f) \\
(244) & \quad c0.1 = c.1[X \mapsto t] \\
(245) & \quad c0.2 = c.2[X \mapsto s(t)] \\
(246) & \quad p1 \leq t \\
(247) & \quad t < p \\
(248) & \quad t \leq x p2 
\end{align*}
\]

From (2) and (244), we know \([2.b.2.3]\).

From (2), (244) and (245), we know \([2.b.2.4]\).

From (3), (246), (247), and (248), we know \([2.b.2.5]\).

From (247), we know

\[ (249) \quad \text{max}(p, t) = p \]

From (3) and (249), to show \([2.b.2.6]\), we have to show

\[ [2.b.2.6.a] \vdash \text{next}(f) \rightarrow \ast(1, p, s, c0.1) g \]

From the definition of \(\rightarrow\ast\), (2), (244), (245), it suffices to show

\[ [2.b.2.6.b] \vdash \text{next}(f) \rightarrow \ast(p, s\downarrow, s(p), c0) g \]

which follows from (242) and (243).

Induction Step
---------------
We assume

\( (1) \quad \forall(n, X, b1, b2, \text{next}(f)) \)
and show

[1] \forall (n+1, X, b_1, b_2, \text{next}(f))

i.e. by the definition of "forall" for arbitrary but fixed
\(p \in \mathbb{N}, s \in \text{Stream}, e \in \text{Environment}, g \in \text{Formula}, c \in \text{Context}, p_1 \in \mathbb{N}_\infty, p_2 \in \mathbb{N}_\infty\)

under the assumptions

(2) \(\vdash \text{next}(\text{TA}(X, b_1, b_2, f)) \rightarrow^* (n+1, p, s, e) \ g\)

(3) \(c = (e, ((Y, s(e(Y))) \mid Y \in \text{dom}(e)))\)

(4) \(p_1 = b_1(c)\)

(5) \(p_2 = b_2(c)\)

the goal

[2] (n+1 = 1 \land (p_1 = \infty \lor p_1 > \infty p_2) \land g = \text{done}(true))

(\lor

(n+1 \geq 1 \land p_1 \neq \infty \land p_1 \leq \infty p_2 \land p+n+1 \leq p_1 \land
g = \text{next}(\text{TA}_0(X, p_1, p_2, \text{next}(f))))

(\lor

(n+1 \geq 1 \land p_1 \neq \infty \land p_1 \leq \infty p_2 \land p+n+1 > p_1 \land

(\exists b \in \text{Bool}: g = \text{done}(b)) \lor

(\exists gs \in \mathbb{P}(\text{TInstance}): (gs \neq \emptyset \lor p+n+1 \leq \infty p_2) \land

\text{forallInstances}(X, p, p+n+1, p_1, p_2, \text{next}(f), s, e, gs) \land

g = \text{next}(\text{TA}_1(X, p_2, \text{next}(f), gs)))
)

which with (0) can be simplified to

[3] (p_1 \neq \infty \land p_1 \leq \infty p_2 \land p+n+1 \leq p_1 \land g = \text{next}(\text{TA}_0(X, p_1, p_2, \text{next}(f))))

(\lor

(p_1 \neq \infty \land p_1 \leq \infty p_2 \land p+n+1 > p_1 \land

(\exists b \in \text{Bool}: g = \text{done}(b)) \lor

(\exists gs \in \mathbb{P}(\text{TInstance}): (gs \neq \emptyset \lor p+n+1 \leq \infty p_2) \land

\text{forallInstances}(X, p, p+n+1, p_1, p_2, \text{next}(f), s, e, gs) \land

g = \text{next}(\text{TA}_1(X, p_2, \text{next}(f), gs)))
)

From (2) and Lemma 2 "Equivalence of Left- and Right-Recursive Definitions of n-Step Reductions", we know

(6) \(\vdash \text{next}(\text{TA}(X, b_1, b_2, f)) \rightarrow^* (n+1, p, s, e) \ g\)
From (6) and the definition of →\textsuperscript{1*}, we know for some \( F_t' \in \text{TFormula} \)

\[
\therefore \quad (7) \vdash \text{next}(\text{TA}(X,b_1,b_2,\text{next}(f))) \rightarrow \text{1*}(n,p,s,e) \quad F_t'
\]

\[
\therefore \quad (8) \vdash F_t' \rightarrow (p+n,s \downarrow (p+n),s(p+n),c) \quad g
\]

From (7) and Lemma 2 "Equivalence of Left- and Right-Recursive Definitions of n-Step Reductions", we know

\[
\therefore \quad (9) \vdash \text{next}(\text{TA}(X,b_1,b_2,\text{next}(f))) \rightarrow \text{*}(n,p,s,e) \quad F_t'
\]

From (1),(3),(4),(5),(9), and the definition of "forall", we know

\[
(10) \quad \begin{cases} 
  n = 1 \land (p_1 = \infty \lor p_1 > \infty \ p_2) \land F_t' = \text{done}(true) \\
  \lor \\
  (n \geq 1 \land p_1 \neq \infty \land p_1 \leq \infty \ p_2 \land p+n \leq p_1 \land F_t' = \text{next}(\text{TA}_0(X,p_1,p_2,\text{next}(f))) \\
  \lor \\
  (n \geq 1 \land p_1 \neq \infty \land p_1 \leq \infty \ p_2 \land p+n > p_1 \land
  (\exists b \in \text{Bool}: F_t' = \text{done}(b)) \lor
  (\exists g_s \in \text{TInstance}: (g_s \neq \emptyset \lor p+n \leq \infty \ p_2) \land
  \text{forallInstances}(X,p,p+n,p_1,p_2,\text{next}(f),s,e,g_s) \land
  F_t' = \text{next}(\text{TA}_1(X,p_2,\text{next}(f),g_s)))
  \end{cases}
\]

From (10), we proceed by case distinction.

Case 1
------

(20) \( n = 1 \)
(21) \( p_1 = \infty \lor p_1 > \infty \ p_2 \)
(22) \( F_t' = \text{done}(true) \)

By the definition of \( \rightarrow \), (22) contradicts (8).

Case 2
------

(50) \( n \geq 1 \)
(51) \( p_1 \neq \infty \)
(52) \( p_1 \leq \infty \ p_2 \)
(53) \( p+n \leq p_1 \)
(54) \( F_t' = \text{next}(\text{TA}_0(X,p_1,p_2,\text{next}(f))) \)

By the definition of \( \rightarrow \), from (8) and (54), we have two subcases.

Subcase 2.1
----------

(60) \( p+n < p_1 \)
(61) \( g = F_t' \)

From (60), we know

(62) \( p + n + 1 \leq p_1 \)

From (51), (52), (54), (61), (62), we have [3] (first disjunct).

Subcase 2.2

There exists \( f_s \) such that

(70) \( p_n \geq p_1 \)

(71) \( f_s = \{(px, next(f), (c.1[X \mapsto px], c.2[X \mapsto s(p_n)])) | p_1 \leq px < \infty \min \infty(p_n, p_2 + \infty 1)\} \)

(72) \( \vdash next(TA_1(X, p_2, next(f), f_s)) \rightarrow (p_n, s(p_n), c) g \)

From (71), we know

(73) \( f_s = \{(px, next(f), (c.1[X \mapsto px], c.2[X \mapsto s(p_x)])) | p_1 \leq px < \infty \min \infty(p_n, p_2 + \infty 1)\} \)

From (51), (52), (70), to show [3], it suffices to show

\[ [4] \exists b \in \text{Bool}: g = \text{done}(b) \lor \exists gs \in P(T\text{Instance}) : (gs \neq \emptyset \lor p+n+1 \leq \infty p_2) \land \forall \text{Instances}(X, p, p+n+1, p_1, p_2, next(f), s, e, gs) \land g = \text{next}(TA_1(X, p_2, next(f), gs)) \]

To show [4], we assume

(74) \( \forall b \in \text{Bool}: g \neq \text{done}(b) \)

and show

\[ [5] \exists gs \in P(T\text{Instance}) : (gs \neq \emptyset \lor p+n+1 \leq \infty p_2) \land \forall \text{Instances}(X, p, p+n+1, p_1, p_2, next(f), s, e, gs) \land g = \text{next}(TA_1(X, p_2, next(f), gs)) \]

From (72) and (74), we know by the definition of \( \rightarrow \) for some \( f_s_0 \) and \( f_s_1 \)

(75) \( f_s_0 = \text{if } p+n > \infty p_2 \text{ then } f_s \text{ else } f_s \cup \{(p+n, next(f), (c.1[X \mapsto s(p+n)], c.2[X \mapsto s(p+n)]))\} \)

(76) \( \neg \exists t \in \mathbb{N}, g \in T\text{Formula}, c \in \text{Context} : (t, g, c) \in f_s_0 \land \vdash g \rightarrow (p+n, s(p_n), s(p+n), c) \text{ done}(false) \)

(77) \( f_s_1 = \{(t, next(fc), c) \in T\text{Instance} | \exists g \in T\text{Formula} : (t, g, c) \in f_s_0 \land \vdash g \rightarrow (p+n, s(p_n), s(p+n), c) next(fc) \} \)

(78) \( \neg (f_s_1 = \emptyset \land p+n \geq \infty p_2) \)

(79) \( g = \text{next}(TA_1(X, p_2, next(f), f_s_1)) \)

To show [5], it suffices to show \( (gs := f_s_1) \)

\[ [5.1] f_s_1 \neq \emptyset \lor p+n+1 \leq \infty p_2 \]

\[ [5.2] \forall \text{Instances}(X, p, p+n+1, p_1, p_2, next(f), s, e, f_s_1) \]

\[ [5.3] g = \text{next}(TA_1(X, p_2, next(f), f_s_1)) \]
To show [5.1], we assume

(80) \( fs_1 = \emptyset \)

and show

[5.1.a] \( p+n+1 \leq \infty \ p_2 \)

From (78) and (80), we know

(81) \( p+n < \infty \ p_2 \)

From (81), we know [5.1.a].

From (79), we know [5.3].

It remains to show [5.2], i.e., by the definition of "forallInstances", for arbitrary \( t \in \mathbb{N}, g_0 \in T\text{Formula}, c_0 \in \text{Context} \), that under the assumption

(82) \( (t,g_0,c_0) \in fs_1 \)

the following holds:

[5.2.1] (\( \forall t_1 \in \mathbb{N}, g_1 \in T\text{Formula}, c_1 \in \text{Context}: \)
\( (t_1,g_1,c_1) \in fs_1 \land t = t_1 \Rightarrow (t,g_0,c_0) = (t_1,g_1,c_1) \)

[5.2.2] \( \exists g_c \in T\text{FormulaCore}: g_0 = \text{next}(g_c) \)

[5.2.3] \( c_0.1 = e[X \mapsto t] \)

[5.2.4] \( c_0.2 = \{ (Y, s(c_0.1(Y)) \mid Y \in \text{dom}(e) \lor Y = X \} \)

[5.2.5] \( p_1 \leq t \)

[5.2.6] \( t \leq p+n \)

[5.2.7] \( t \leq \infty \ p_2 \)

[5.2.8] \( \vdash \text{next}(f) \rightarrow^{*} (p+n+1-\max(p,t),\max(p,t),s,c_0.1) \ g_0 \)

From (77) and (82), we know for some \( f_c \in T\text{FormulaCore}, g_1 \in T\text{Formula} \)

(83) \( g_0 = \text{next}(f_c) \)

(84) \( (t,g_1,c_0) \in fs_0 \)

(85) \( \vdash g_1 \rightarrow (p+n,s\mid(p+n),s(p+n),c_0) \ g_0 \)

From (53) and (70), we know

(86) \( p+n = p_1 \)

From (73) and (86), we know

(87) \( fs = \emptyset \)

From (84), we know

(88) \( fs_0 \neq \emptyset \)

From (75), (87), and (88), we know

(89) \( fs_0 = \{(p+n, \text{next}(f), (c.1[X \mapsto p+n], c.2[X \mapsto s(p+n)]))\} \)
From (84) and (89), we know

\[(100) \ t = p+n\]
\[(101) \ g_1 = \text{next}(f)\]
\[(102) \ c_{0.1} = c.1[X \mapsto p+n]\]
\[(103) \ c_{0.2} = c.2[X \mapsto s(p+n)]\]

From (77), (89), and the fact that the rule system is deterministic, we know [5.2.1].

From (83), we know [5.2.2].

From (3), (100), (102), and (103) we know [5.2.3] and [5.2.4].

From (86) and (100), we know [5.2.5] and [5.2.6].

From (52), (86), and (100), we know [5.2.7].

From (0) and (100), we know

\[(104) \ \max(p,t) = t\]

From (100), (101) and (104), to show [5.2.8], it suffices to show

\[\text{[5.2.8.a]} \vdash g_1 \rightarrow^{**}(1,p+n,s,c_{0.1}) \ g_0\]

From the definition of \(\rightarrow\), (85), (3), (102), and (103), we have [5.2.8.a].

Case 3
------
\[(200) \ n \geq 1\]
\[(201) \ p_1 \neq \infty\]
\[(202) \ p_1 \leq \infty p_2\]
\[(203) \ p+n > p_1\]
\[(204) \ (\exists b \in \text{Bool}: \text{Ft'}=\text{done}(b)) \lor\]
\[\ (\exists gs \in \mathcal{P}(T\text{Instance}): (gs \neq \emptyset \lor p+n \leq \infty p_2) \land\]
\[\ \text{forallInstances}(X,p,p+n,p_1,p_2,\text{next}(f),s,e,gs) \land\]
\[\ \text{Ft'} = \text{next(TA1}(X,p_2,\text{next}(f),gs))\]

From (204), we proceed by case distinction.

Subcase 3.1
-------------
We have some \(b \in \text{Bool}\) such that
\[(210) \ \text{Ft'}=\text{done}(b)\]

By the definition of \(\rightarrow\), (210) contradicts (8).

Subcase 3.2
-------------
We have some \(gs \in \mathcal{P}(T\text{Instance})\) such that
\[(301) \ gs \neq \emptyset \lor p+n \leq \infty p_2\]
\[(302) \ \text{forallInstances}(X,p,p+n,p_1,p_2,\text{next}(f),s,e,gs)\]
(303) \( F_t' = \text{next}(TA_1(X,p_2,\text{next}(f),gs)) \)

We define

\[
(304) \quad fs_0 = \text{if } p+n > \infty p_2 \text{ then } gs \text{ else } gs \cup \\
\{ (p+n, \text{next}(f), (c.1[X \mapsto p+n], c.2[X \mapsto s(p+n)])) \}
\]

From (8), (303), and (304), we have by the definition of \( \rightarrow \) three cases.

Subsubcase 3.2.1
----------------

We have some \( t_0 \in \mathbb{N}, g_0 \in T\text{Formula}, c_0 \in \text{Context} \) such that

\[
(310) \quad (t_0, g_0, c_0) \in fs_0 \\
(311) \quad \vdash g_0 \rightarrow (p+n, s\downarrow(p+n), s(p+n), c) \text{ done(false)} \\
(312) \quad g = \text{done(false)}
\]

From (201), (202), (203), and (312), we have [3] (second disjunct, first case).

Subsubcase 3.2.2
----------------

We have some \( fs_1 \) such that

\[
(320) \quad \neg \exists t \in \mathbb{N}, g \in T\text{Formula}, c \in \text{Context} : (t, g, c) \in gs \land \\
\quad \vdash g \rightarrow (p+n, s\downarrow(p+n), s(p+n), c) \text{ done(false)} \\
(321) \quad fs_1 = \{ (t, \text{next}(fc), c) \in T\text{Instance} | \exists g \in T\text{Formula} : (t, g, c) \in fs_0 \land \\
\quad \vdash g \rightarrow (p+n, s\downarrow(p+n), s(p+n), c) \text{ next}(fc) \} \\
(322) \quad fs_1 = \emptyset \\
(323) \quad p+n \geq \infty p_2 \\
(324) \quad g = \text{done(true)}
\]

From (201), (202), (203), and (324), we have [3] (second disjunct, first case).

Subsubcase 3.2.3
----------------

We have some \( fs_1 \) such that

\[
(330) \quad \neg \exists t \in \mathbb{N}, g \in T\text{Formula}, c \in \text{Context} : (t, g, c) \in gs \land \\
\quad \vdash g \rightarrow (p+n, s\downarrow(p+n), s(p+n), c) \text{ done(false)} \\
(331) \quad fs_1 = \{ (t, \text{next}(fc), c) \in T\text{Instance} | \exists g \in T\text{Formula} : (t, g, c) \in fs_0 \land \\
\quad \vdash g \rightarrow (p+n, s\downarrow(p+n), s(p+n), c) \text{ next}(fc) \} \\
(332) \quad \neg (fs_1 = \emptyset) \land p+n \geq \infty p_2 \\
(333) \quad g = \text{next}(TA_1(X,p_2,\text{next}(f),fs_1))
\]

From (201), (202), (203), and (333), to show [3], it suffices to show (second disjunct, second case, \( gs := fs_1 \)):

[3.1] \( fs_1 \neq \emptyset \lor p+n+1 \leq \infty p_2 \)
[3.2] \( \text{forallInstances}(X, p, p+n+1, p_1, p_2, \text{next}(f), s, e, fs_1) \)
[3.3] \( g = \text{next}(TA_1(X,p_2,\text{next}(f),fs_1)) \)

From (332), we have [3.1].
From (333), we have [3.3].

To show [3.2], by the definition of "forallInstances", we take arbitrary \( t, g_0, c_0 \) such that

143
(340) \((t,g_0,c_0)\in\text{fs}_1\)

and show

\[
\begin{align*}
[3.2.1] & \forall t_1\in\mathbb{N}, g_1\in\text{TFormula}, c_1\in\text{Context}: \\
& (t_1,g_1,c_1)\in\text{fs}_1 \land t=t_1 \Rightarrow (t,g_0,c_0)=(t_1,g_1,c_1) \\
[3.2.2] & \exists g_c\in\text{TFormulaCore}: g_0=\text{next}(g_c) \\
[3.2.3] & c_{0.1}=e[X\mapsto t] \\
[3.2.4] & c_{0.2}={(Y,s(c_{0.1}(Y))) \mid Y\in\text{dom}(e) \lor Y=X} \\
[3.2.5] & p_1 \leq t \\
[3.2.6] & t \leq p+n \\
[3.2.7] & t \leq \infty \\
[3.2.8] & \vdash \text{next}(f)\rightarrow^{\ast}(p+n-\max(p,t),\max(p,t),s,c_{0.1})g_0
\end{align*}
\]

From (331) and (340), we have some \(f_c_0\in\text{TFormulaCore}, g_1\in\text{TFormula}\) with

(341) \(g_0 = \text{next}(f_c_0)\)

(342) \((t,g_1,c_0)\in\text{fs}_0\)

(343) \(\vdash g_1\rightarrow(p+n,s\downarrow(p+n),s(p+n),c_0)\text{next}(f_c_0)\)

From (341), we have [3.2.2].

It remains to show [3.2.1] and [3.2.3-8].

From (302) and the definition of "forallInstances", we know

(344)

\[
\begin{align*}
\forall t\in\mathbb{N}, g\in\text{TFormula}, c_0\in\text{Context}: & (t,g,c_0)\in\text{gs} \Rightarrow \\
&(\forall t_1\in\mathbb{N}, g_1\in\text{TFormula}, c_1\in\text{Context}: \\
&(t_1,g_1,c_1)\in\text{gs} \land t=t_1 \Rightarrow (t,g,c_0)=(t_1,g_1,c_1)) \land \\
&(\forall g_c\in\text{TFormulaCore}: g=\text{next}(g_c)) \land \\
&c_{0.1}=e[X\mapsto t] \land c_{0.2}={(Y,s(c_{0.1}(Y))) \mid Y\in\text{dom}(e) \lor Y=X} \land \\
p_1 \leq t \leq \infty \min\infty(p+n-1,p_2) \land \\
&\vdash \text{next}(f)\rightarrow^{\ast}(p+n-\max(p,t),\max(p,t),s,c_{0.1})g
\end{align*}
\]

We proceed by case distinction.

Subsubsubcase 3.2.3.1
---------------------
(350) \(p+n > \infty \leq p_2\)

From (304) and (350), we have

(351) \(\text{fs}_0 = \text{gs}\)

From (342), (351), and (344), we know for some \(g_c_0\in\text{TFormulaCore}\)

(352) \(\forall t_2\in\mathbb{N}, g_2\in\text{TFormula}, c_2\in\text{Context}: \\
(t_2,g_2,c_2)\in\text{gs} \land t=t_2 \Rightarrow (t,g_1,c_0)=(t_2,g_2,c_2)\)

(353) \(g_1=\text{next}(g_c_0)\)

(354) \(c_{0.1}=e[X\mapsto t]\)

(355) \(c_{0.2}={(Y,s(c_{0.1}(Y))) \mid Y\in\text{dom}(e) \lor Y=X}\)

(356) \(p_1 \leq t\)

144
(357) \( t < p+n \)
(358) \( t \leq \infty \) \( p^2 \)
(359) \[ \vdash \text{next}(f) \rightarrow^* (p+n-\max(p,t),\max(p,t),s,c_0.1) \ g_1 \]

From (331), (351), (352), and the fact that the rule system for \( \rightarrow \) is deterministic, we know [3.2.1].

From (354), we know [3.2.3].
From (355), we know [3.2.4]
From (356), we know [3.2.5].
From (357), we know [3.2.6].
From (358), we know [3.2.7].

From (359) and Lemma 2 "Equivalence of Left- and Right-Recursive Definitions of n-Step Reductions", we know

(360) \[ \vdash \text{next}(f) \rightarrow^* (p+n-\max(p,t),\max(p,t),s,c_0.1) \ g_1 \]

From (343) and (360), we know by the definition of \( \rightarrow^* \)

(361) \[ \vdash \text{next}(f) \rightarrow^* (p+n+1-\max(p,t),\max(p,t),s,c_0.1) \text{next}(fc_0) \]

From (361) and Lemma 2 "Equivalence of Left- and Right-Recursive Definitions of n-Step Reductions", we know

(362) \[ \vdash \text{next}(f) \rightarrow^* (p+n+1-\max(p,t),\max(p,t),s,c_0.1) \text{next}(fc_0) \]

From (341) and (362), we know [3.2.8].

Subsubsubcase 3.2.3.2
---------------------
(400) \( p+n \leq \infty \) \( p^2 \)

From (304) and (400), we know

(401) \( f_s = g_s \cup \{(p+n,\text{next}(f),(c.1[X\rightarrow s(p+n)],c.2[X\rightarrow s(p+n)]))\} \)

From (342) and (401), we have two cases.

Subsubsubsubcase 3.2.3.2.1
--------------------------
(410) \( (t,g_1,c_0) \in g_s \)

From (344) and (410), we know for some \( gc_0 \in \text{TFormulaCore} \)

(412) \forall t_2 \in N, g_2 \in \text{TFormula}, c_2 \in \text{Context}: 
(413) \( g_1 = \text{next}(gc_0) \)
(414) \( c_0.1 = [X \mapsto t] \)
(415) \( c_0.2 = \{ (Y,s(c_0.1(Y)) \mid Y \in \text{dom}(s) \lor Y = X \} \)
(416) \( p_1 \leq t \)
(417) \( t < p+n \)
(418) \( t \leq \infty \) \( p^2 \)
(419) \[ \vdash \text{next}(f) \rightarrow^* (p+n-\max(p,t),\max(p,t),s,c_0.1) \ g_1 \]
To show [3.2.1], we take arbitrary \( t_2 \in \mathbb{N}, g_2 \in T\text{Formula}, c_2 \in \text{Context} \) for which we assume

\[
(420) \ (t_2, g_2, c_2) \in fs_1
\]
\[
(421) \ t = t_2
\]

and show

[3.2.1.a] \((t, g_0, c_0) = (t_2, g_2, c_2)\)

To show [3.2.1.a], from (421), it suffices to show

[3.2.1.a.1] \(g_0 = g_2\)
[3.2.1.a.2] \(c_0 = c_2\)

From (331) and (420), we have some \( g_3 \in T\text{Formula}, f c_3 \in T\text{FormulaCore} \) such that

\[
(422) \ g_2 = \text{next}(f c_3)
\]
\[
(423) \ (t, g_3, c_1) \in fs_0
\]
\[
(424) \vdash g_3 \to (p+n,s\uparrow(p+n),s(p+n),c_1) \ g_2
\]

From (401), (417), and (423), we know

\[
(425) \ (t, g_3, c_1) \in gs
\]

From (410), (412), and (425), we have

\[
(426) \ g_1 = g_3
\]
\[
(427) \ c_0 = c_1
\]

From (341), (343), (426), and (427), we have

\[
(428) \vdash g_3 \to (p+n,s\uparrow(p+n),s(p+n),c_1) \ g_0
\]

From (424), (428), and the fact that the rule system for \( \to \) is deterministic, we have [3.2.1.a.1].

From (427), we have [3.2.1.a.2].

From (414), we know [3.2.3].
From (415), we know [3.2.4]
From (416), we know [3.2.5].
From (417), we know [3.2.6].
From (418), we know [3.2.7].

From (419) and Lemma 2 "Equivalence of Left- and Right-Recursive Definitions of n-Step Reductions", we know

\[
(450) \vdash \text{next}(f) \to^* l^*(p+n-\text{max}(p,t),\text{max}(p,t),s,c_0.1) \ g_1
\]

From (343) and (450), we know by the definition of \( \to^* \)

\[
(451) \vdash \text{next}(f) \to^* l^*(p+n+1-\text{max}(p,t),\text{max}(p,t),s,c_0.1) \text{next}(fc_0)
\]
From (451) and Lemma 2 "Equivalence of Left- and Right-Recursive Definitions of n-Step Reductions", we know

\[(452) \vdash \text{next}(f) \rightarrow \ast(p+n+1-\max(p,t),\max(p,t),s,c_{0.1}) \text{next}(fc_{0})\]

From (341) and (452), we know [3.2.8].

Subsubsubsubcase 3.2.3.2.2

\[(500) t=p+n\]
\[(501) g_{1}=\text{next}(f)\]
\[(502) c_{0.1}=c_{1}[X\mapsto p+n]\]
\[(503) c_{0.2}=c_{2}[X\mapsto s(p+n)]\]

To show [3.2.1], we take arbitrary \(t_{2}\in \mathbb{N},g_{2}\in \text{TFormula},c_{2}\in \text{Context}\) for which we assume

\[(520) (t_{2},g_{2},c_{2})\in \text{fs}_{1}\]
\[(521) t=t_{2}\]

and show

[3.2.1.a] \((t,g_{0},c_{0})=(t_{2},g_{2},c_{2})\)

To show [3.2.1.a], from (521), it suffices to show

[3.2.1.a.1] \(g_{0} = g_{2}\)
[3.2.1.a.2] \(c_{0} = c_{2}\)

From (331) and (520), we have some \(g_{3}\in \text{TFormula}, fc_{3}\in \text{TFormulaCore}\) such that

\[(522) g_{2}=\text{next}(fc_{3})\]
\[(523) (t,g_{3},c_{2})\in \text{fs}_{0}\]
\[(524) \vdash g_{3} \rightarrow \ast(p+n,s\downarrow(p+n),s(p+n),c_{2}) g_{2}\]

From (344) and (500), we know

\[(525) (t,g_{3},c_{2})\not\in \text{gs}\]

From (401), (523), (525), we know

\[(526) g_{3} = \text{next}(f)\]
\[(527) c_{2.1} = c_{1}[X\mapsto p+n]\]
\[(528) c_{2.2} = c_{2}[X\mapsto s(p+n)]\]

From (341), (343), (501), (524), (527), (528), we know

\[(529) \vdash g_{3} \rightarrow \ast(p+n,s\downarrow(p+n),s(p+n),c_{2}) g_{0}\]

From (524), (529), and the fact that the rule system for \(\rightarrow\) is deterministic, we have [3.2.1.a.1].

From (502), (503), (527), (528), we know [3.2.1.a.2].
From (2), (500), (502), (503), we know [3.2.3] and [3.2.4].
From (203) and (500), we know [3.2.5].
From (500), we know [3.2.6].
From (400) and (500), we know [3.2.7].

From (500), to show [3.2.8], it suffices to show

\[ [3.2.8.a] \vdash \text{next}(f) \rightarrow (1, p+n, s, c0.1) g0 \]

From (526), (529), [3.2.1.a.2], and the definition of \( \rightarrow * \),
we know [3.2.8.a].

Q.E.D.