

A Proof Pearl with the Fan Theorem and Bar Induction

Walking through Infinite Trees with Mixed Induction and Coinduction

Keiko Nakata¹, Tarmo Uustalu¹, and Marc Bezem²

¹ Institute of Cybernetics, Tallinn University of Technology,
Akadeemia tee 21, 12618 Tallinn, Estonia

{keiko,tarmo}@cs.ioc.ee

² Institutt for Informatikk, Universitet i Bergen,
Postboks 7800, 5020 Bergen, Norway

bezem@ii.uib.no

Abstract. We study temporal properties over infinite binary red-blue trees in the setting of constructive type theory. We consider several familiar path-based properties, typical to linear-time and branching-time temporal logics like LTL and CTL*, and the corresponding tree-based properties, in the spirit of the modal μ -calculus. We conduct a systematic study of the relationships of the path-based and tree-based versions of “eventually always blueness” and mixed inductive-coinductive “almost always blueness” and arrive at a diagram relating these properties to each other in terms of implications that hold either unconditionally or under specific assumptions (Weak Continuity for Numbers, the Fan Theorem, Lesser Principle of Omniscience, Bar Induction).

We have fully formalized our development with the Coq proof assistant.

1 Introduction

In this paper, we study temporal properties over infinite binary red-blue trees in the setting of constructive type theory. We consider several familiar path-based properties, typical to linear-time and branching-time temporal logics like LTL and CTL*, and the corresponding tree-based properties, in the spirit of the modal μ -calculus. Classically, some of these properties coincide, but in our more discerning setting they come out generally inequivalent. We then look for weak assumptions under which they imply each other. It turns out that some implications are in fact equivalent to principles well known in constructive mathematics and others follow from such principles.

We are primarily interested in path-based and tree-based variations of the properties of “eventually always blueness” and “almost always blueness” of a tree where the latter is defined by mixing induction and coinduction. We conduct a systematic study of the relationships of these properties and arrive at a diagram

where we describe how these properties relate to each other in terms of implications that hold either unconditionally or under specific assumptions (Weak Continuity for Numbers, the Fan Theorem, Lesser Principle of Omniscience, Bar Induction). This way, we learn about the relative constructive strength of these properties in terms of the computational content of the assumptions used (cf., [11,12]) and conversely we get some intuition about the significance of these principles from a programmer’s viewpoint.

The paper proceeds as follows. After setting up the basic framework in Sect. 2, we first study universal properties over paths in a tree and compare the path-based and tree-based variations. In Sect. 3.1, we examine trees that are always blue and, in Sect. 3.2, we also look at trees that are eventually red. In Sect. 4.1, we study eventually always blue trees. In Sect. 4.2, we consider trees that are almost always blue in the sense of a mixed inductive-coinductive definition.

We then continue with always eventually and infinitely often red trees (Sect. 5). Our journey ends with a short discussion of existential properties over paths in a tree (Sect. 6). We discuss related work in Sect. 7 to conclude in Sect. 8.

In regards to the various non-constructive principles we employ we use the terminology of Troelstra and van Dalen [22]. We use Latin lower case letters to represent finite objects, and Greek lower case letters for infinite objects. We present the definitions of both inductive and coinductive types and predicates in terms of sets of rules. The rules of inductive definitions are denoted by a single line and the rules of coinductive definitions are marked by a double line.

We have fully formalized our development in Coq. The Coq development is available at <http://cs.ioc.ee/~keiko/cypress.tgz>.

2 Preliminaries

In this section, we set up a basis for our development in the paper.

We have two colors, red (R) and blue (B). We also have steps, represented by bits, 0 for left and 1 for right. Namely,

$$\overline{R : color} \quad \overline{B : color} \quad \overline{0 : step} \quad \overline{1 : step}$$

Streams $\alpha : A^\omega$ over a set A are infinite sequences over A defined coinductively by

$$\frac{a : A \quad \alpha : A^\omega}{a \alpha : A^\omega}$$

Bisimilarity on streams, $\alpha \sim \alpha'$, is also defined coinductively by

$$\frac{\alpha \sim \alpha'}{a \alpha \sim a \alpha'}$$

Our trees $\tau : tree$ are infinite binary trees with colored nodes, defined coinductively by

$$\frac{c : color \quad \tau_0 : tree \quad \tau_1 : tree}{\tau_0 c \tau_1 : tree}$$

with bisimilarity on them, $\tau \sim \tau'$, defined coinductively by

$$\frac{\tau_0 \sim \tau'_0 \quad \tau_1 \sim \tau'_1}{(\tau_0 \ c \ \tau_1) \sim (\tau'_0 \ c \ \tau'_1)}$$

Note that a tree has no leaves, hence all the paths are infinite.

The relations \sim (for both streams and trees) are straightforwardly seen to be equivalences. We take bisimilarity as the equality on streams (resp. trees), i.e., type-theoretically we treat streams (resp. trees) as a setoid with bisimilarity as the equivalence relation. Accordingly, we have to ensure that all functions and predicates we define on streams (resp. trees) are setoid functions and predicates (i.e., respect our notions of equality for them).

Lists $\ell : A^*$ over a set A are finite sequences over A defined inductively by

$$\frac{}{\langle \rangle : A^*} \quad \frac{a : A \quad \ell : A^*}{a \ \ell : A^*}$$

The notation $\langle a \rangle$ denotes singletons, i.e., $\langle a \rangle = a \ \langle \rangle$. For $\ell, \ell' : A^*$, we denote by $\ell * \ell'$ the concatenation ℓ and ℓ' .

Concatenation can be extended to concatenation of a finite sequence $\ell : A^*$ and an infinite one $\alpha : A^\omega$:

$$\langle \rangle * \alpha = \alpha \quad (a \ \ell) * \alpha = a \ (\ell * \alpha)$$

Non-empty lists $\ell : A^+$ over a set A are inductively defined by

$$\frac{a : A}{\langle a \rangle : A^+} \quad \frac{a : A \quad \ell : A^+}{a \ \ell : A^+}$$

When necessary, we tacitly coerce non-empty lists to lists.

The function $flatten : (A^+)^{\omega} \rightarrow A^{\omega}$ flattens the given stream of non-empty lists over A . Formally, we define it by mutual corecursion together with an auxiliary function $flattenseq$:

$$\begin{aligned} flatten(\langle a \rangle \ \alpha) &= a \ (flatten \ \alpha) & flatten((a \ \ell) \ \alpha) &= a \ (flattenseq \ \alpha \ \ell) \\ flattenseq \ \alpha \ \langle a \rangle &= a \ (flatten \ \alpha) & flattenseq \ \alpha \ (a \ \ell) &= a \ (flattenseq \ \alpha \ \ell) \end{aligned}$$

The initial segment of length n of a stream α is denoted by $\overline{\alpha}n$. Formally

$$\overline{\alpha}0 = \langle \rangle \quad \overline{(a \ \alpha)}(n + 1) = a \ (\overline{\alpha}n)$$

The suffix of a stream α at n , $\alpha@n$, is defined by

$$\alpha@0 = \alpha \quad (a \ \alpha)@(n + 1) = \alpha@n$$

The subtree of a tree $\tau : tree$ at a position $p : step^*$ is denoted by $\tau@p$. Formally,

$$\tau@\langle \rangle = \tau \quad (\tau_0 \ c \ \tau_1)@(0 \ p) = \tau_0@p \quad (\tau_0 \ c \ \tau_1)@(1 \ p) = \tau_1@p$$

For a tree $\tau : tree$ and a path $\pi : step^\omega$, $\llbracket \tau \rrbracket_\pi$ returns the stream of colors in τ along π . Formally,

$$\llbracket \tau_0 \ c \ \tau_1 \rrbracket_{0 \ \pi} = c \ \llbracket \tau_0 \rrbracket_\pi \quad \llbracket \tau_0 \ c \ \tau_1 \rrbracket_{1 \ \pi} = c \ \llbracket \tau_1 \rrbracket_\pi$$

Analogously, for a position $p : step^*$, $\llbracket \tau \rrbracket_p$ returns the list of colors in τ along p . Formally,

$$\llbracket \tau \rrbracket_{\langle \rangle} = \langle \rangle \quad \llbracket \tau_0 \ c \ \tau_1 \rrbracket_{(0 \ p)} = c \ \llbracket \tau_0 \rrbracket_p \quad \llbracket \tau_0 \ c \ \tau_1 \rrbracket_{(1 \ p)} = c \ \llbracket \tau_1 \rrbracket_p$$

It is convenient to introduce some predicates on streams of colors, typically written $\sigma : color^\omega$, and trees as primitives into our language for them. We define

$$\overline{red(R \ \sigma)} \quad \overline{blue(B \ \sigma)} \quad \overline{red(\tau_0 \ R \ \tau_1)} \quad \overline{blue(\tau_0 \ B \ \tau_1)}$$

For streams of colors, we also define

$$\frac{X \ \sigma}{\mathcal{F} X \ \sigma} \quad \frac{\mathcal{F} X \ \sigma}{\mathcal{F} X (c \ \sigma)} \quad \frac{X (c \ \sigma) \quad \mathcal{G} X \ \sigma}{\mathcal{G} X (c \ \sigma)}$$

Here, \mathcal{F} and \mathcal{G} are the “sometime in the future” (“finally”) and “always in the future” (“globally”) modalities of linear-time temporal logic. They are predicates on streams of colors parameterized over predicates X on streams of colors.¹ Analogously, we define “eventually” and “always” predicates on trees:

$$\frac{X \ \tau}{\mathcal{F} X \ \tau} \quad \frac{\mathcal{F} X \ \tau_0 \quad \mathcal{F} X \ \tau_1}{\mathcal{F} X (\tau_0 \ c \ \tau_1)} \quad \frac{X (\tau_0 \ c \ \tau_1) \quad \mathcal{G} X \ \tau_0 \quad \mathcal{G} X \ \tau_1}{\mathcal{G} X (\tau_1 \ c \ \tau_1)}$$

Again, \mathcal{F} and \mathcal{G} are predicates on trees parameterized over predicates X on trees. In section 6, we will consider variations of $\mathcal{G} X \ \tau$ and $\mathcal{F} X \ \tau$ which pick up one of the subtrees at every node as they go down through τ .

3 Always Blue and Eventually Red Trees

3.1 Always Blue Trees

A stream of colors $\sigma : color^\omega$ is *always blue*, if σ is “globally” blue, or $\mathcal{G} \ blue \ \sigma$. Similarly, a tree $\tau : tree$ is *always blue*, if every node of τ is blue, or $\mathcal{G} \ blue \ \tau$.

A tree is always blue if and only if every path of the tree is always blue:

Proposition 1. $\forall \tau : tree. \mathcal{G} \ blue \ \tau \Leftrightarrow (\forall \pi : step^\omega. \mathcal{G} \ blue \ \llbracket \tau \rrbracket_\pi)$.

¹ There is no need to see them as “first-class” predicate transformers, as there is no real impredicativity involved: the argument of \mathcal{F} is constantly X in the definition of \mathcal{F} , and the same is true of the definition of \mathcal{G} .

3.2 Eventually Red Trees

A stream of colors σ is *eventually red* if σ is red at some position, or, $\mathcal{F} \text{ red } \sigma$. An infinite tree τ is *eventually red* if a finite initial fragment of it has all leaves red, or $\mathcal{F} \text{ red } \tau$.

Constructively, we have neither that a stream of colors is either always blue or eventually red, $\forall \sigma. \mathcal{G} \text{ blue } \sigma \vee \mathcal{F} \text{ red } \sigma$, nor that a stream of colors not being always blue implies that it is eventually red, $\forall \sigma. \neg \mathcal{G} \text{ blue } \sigma \Rightarrow \mathcal{F} \text{ red } \sigma$. The former is equivalent to the Lesser Principle of Omniscience (LPO), saying that $(\forall n. P n \vee \neg P n) \Rightarrow \forall n. \neg P n \vee \exists n. P n$, the latter to Markov’s Principle (MP), saying that $(\forall n. P n \vee \neg P n) \Rightarrow \neg \forall n. \neg P n \Rightarrow \exists n. P n$ where P is a predicate on natural numbers. Both LPO and MP are important principles that are neither valid nor inconsistent constructively, but are valid classically. LPO is a special case of the Principle of the Excluded Middle (PEM). MP, which is a special case of the Double Negation Elimination, is even computationally meaningful, being realizable by search that we know cannot diverge.

If a tree is eventually red, then every path of the tree is eventually red:

Proposition 2. $\forall \tau : \text{tree}. \mathcal{F} \text{ red } \tau \Rightarrow \forall \pi : \text{step}^\omega. \mathcal{F} \text{ red } \llbracket \tau \rrbracket_\pi.$

To obtain the tree-based formulation from the path-based one, we invoke the Fan Theorem for a decidable bar (FAN_D). Let P and Q be predicates on positions. Then FAN_D can be expressed as

$$(\forall p. P p \vee \neg P p) \Rightarrow \text{FAN}$$

where FAN (the general Fan Theorem) is

$$(\forall \pi. \exists n. P(\bar{\pi}n)) \Rightarrow (\forall p. P p \Rightarrow Q p) \Rightarrow (\forall p. Q(p * \langle 0 \rangle) \Rightarrow Q(p * \langle 1 \rangle) \Rightarrow Q p) \Rightarrow Q \langle \rangle$$

FAN_D is not valid in basic constructive logic. It is the classical contrapositive of Weak König’s Lemma², which is valid classically. In fact, Weak König’s Lemma implies FAN_D even constructively [14].

FAN_D is both sufficient and necessary for path-based eventual redness to imply tree-based eventual redness.

Proposition 3. $\text{FAN}_D \Leftrightarrow (\forall \tau : \text{tree}. (\forall \pi : \text{step}^\omega. \mathcal{F} \text{ red } \llbracket \tau \rrbracket_\pi) \Rightarrow \mathcal{F} \text{ red } \tau).$

Proof. \Rightarrow : The claim is an instance of FAN_D by taking P and Q as follows. For any $p : \text{step}^*$, $P p$ holds if the subtree of τ at p is red, or $\text{red}(\tau @ p)$. For any $p : \text{step}^*$, $Q p$ holds if the subtree of τ at p is eventually red, or $\mathcal{F} \text{ red}(\tau @ p)$.

\Leftarrow : We define a tree τ_P by corecursion such that $\text{red}(\tau_P @ p)$ if and only if $P p$. Then the assumption $\forall \pi. \exists n. P(\bar{\pi}n)$ is equivalent to $\forall \pi. \mathcal{F} \text{ red } \llbracket \tau_P \rrbracket_\pi$. The assumption $\forall \tau. (\forall \pi. \mathcal{F} \text{ red } \llbracket \tau \rrbracket_\pi) \Rightarrow \mathcal{F} \text{ red } \tau$ therefore gives us $\mathcal{F} \text{ red } \tau_P$. Now $Q \langle \rangle$ follows from $\forall p. \mathcal{F} \text{ red}(\tau_P @ p) \Rightarrow Q p$ proved by induction on the proof of $\mathcal{F} \text{ red}(\tau_P @ p)$ using $\forall p. P p \Rightarrow Q p$ and $\forall p. Q(p * \langle 0 \rangle) \Rightarrow Q(p * \langle 1 \rangle) \Rightarrow Q p$.

² Weak König’s Lemma states that every infinite binary tree has an infinite path.

4 Eventually Always vs. Almost Always Blue Trees

In this section we look at path-based and tree-based concepts of eventually always and almost always blue trees.

4.1 Eventually Always Blue Trees

A stream of colors σ is *eventually always blue*, if, from some position on, σ is always blue, or $\mathcal{F}(\mathcal{G} \text{ blue}) \sigma$. A tree τ is *eventually always blue* if all nodes beyond some finite initial fragment of it are blue, or $\mathcal{F}(\mathcal{G} \text{ blue}) \tau$.

Again, the tree-based formulation is stronger than the path-based one:

Proposition 4. $\forall \tau : \text{tree}. \mathcal{F}(\mathcal{G} \text{ blue}) \tau \Rightarrow \forall \pi : \text{step}^\omega. \mathcal{F}(\mathcal{G} \text{ blue}) \llbracket \tau \rrbracket_\pi$.

To obtain the tree-based formulation from the path-based one, we invoke Weak Continuity for Numbers (WC-N) and the general Fan Theorem (FAN). Let P be a predicate on pairs of a path and natural number. Then WC-N can be expressed as

$$(\forall \pi. \exists n. P(\pi, n)) \Rightarrow \forall \pi. \exists m. \exists n. \forall \pi'. \overline{\pi}m = \overline{\pi'}n \Rightarrow P(\pi', n)$$

While FAN is valid classically, WC-N contradicts classical logic, but is nonetheless consistent with basic constructive logic.

We derive the tree-based formulation from the path-based one in two steps, to highlight the use of each of the two principles separately. We therefore introduce an intermediate step that is half path-based, half tree-based.

For any given path π , a tree τ is *eventually always blue along π* if the subtree of τ at some point along π is all blue, or $\exists n. \mathcal{G} \text{ blue} (\tau @ \overline{\pi}n)$.

If we accept WC-N, then we have that if every path of a tree is eventually always blue, then the tree is eventually always blue along every path:

Proposition 5. *Assuming WC-N,* $\forall \tau : \text{tree}. (\forall \pi : \text{step}^\omega. \mathcal{F}(\mathcal{G} \text{ blue}) \llbracket \tau \rrbracket_\pi) \Rightarrow \forall \pi : \text{step}^\omega. \exists n. \mathcal{G} \text{ blue} (\tau @ \overline{\pi}n)$.

Proof. For any given τ , we suppose that, $\forall \pi. \mathcal{F}(\mathcal{G} \text{ blue}) \llbracket \tau \rrbracket_\pi$. By WC-N, we have that, $\forall \pi. \exists m. \exists n. \forall \pi'. \overline{\pi}m = \overline{\pi'}n \Rightarrow \mathcal{G} \text{ blue} (\llbracket \tau \rrbracket_{\pi'} @ n)$, by taking $P(\pi, n)$ to mean $\mathcal{G} \text{ blue} (\llbracket \tau \rrbracket_\pi @ n)$. This gives us that, $\forall \pi. \exists n. \forall \pi'. \mathcal{G} \text{ blue} \llbracket \tau @ \overline{\pi}n \rrbracket_{\pi'}$. We conclude that $\forall \pi. \exists n. \mathcal{G} \text{ blue} (\tau @ \overline{\pi}n)$ by Prop. 1, as required.

If we accept FAN, then we have that if a tree is eventually always blue along every path, then the tree is eventually always blue:

Proposition 6. *Assuming FAN,* $\forall \tau : \text{tree}. (\forall \pi : \text{step}^\omega. \exists n. \mathcal{G} \text{ blue} (\tau @ \overline{\pi}n)) \Rightarrow \mathcal{F}(\mathcal{G} \text{ blue}) \tau$.

Proof. The claim is an instance of FAN by taking P and Q as follows. For any $p : \text{step}^*$, $P p$ holds if $\mathcal{G} \text{ blue} (\tau @ p)$. For any $p : \text{step}^*$, $Q p$ holds if $\mathcal{F}(\mathcal{G} \text{ blue}) (\tau @ p)$.

With the above two propositions, we derive the tree-based formulation from the path-based one:

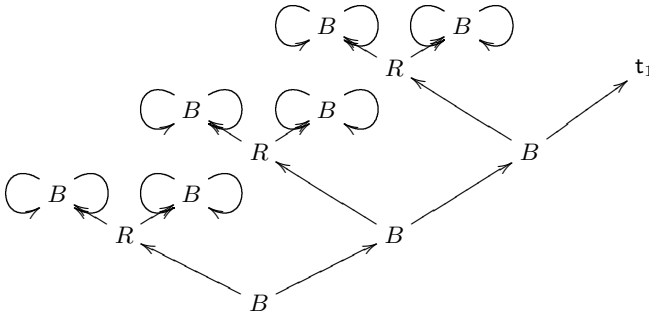


Fig. 1. The tree $t_1 = (t_0 R t_0) B t_1$

Corollary 1. *Assuming WC-N and FAN, $\forall \tau : tree. (\forall \pi : step^\omega. \mathcal{F}(\mathcal{G} blue) \llbracket \tau \rrbracket_\pi) \Rightarrow \mathcal{F}(\mathcal{G} blue) \tau$.*

The concepts introduced are well illustrated by the following example.

Let t_0 be an always blue tree, defined by corecursion by

$$t_0 = t_0 B t_0$$

(this is in fact the only always blue tree, up to bisimilarity).

Our example of interest, t_1 , is defined by corecursion by

$$t_1 = (t_0 R t_0) B t_1$$

so that t_1 is red exactly at positions of the form 1^*0 , i.e., it is red the first time a 0-step is taken. The tree is depicted in Fig. 1.

It is clear that $\mathcal{F}(\mathcal{G} blue) t_1$ is false, since it is impossible to carve out a finite initial fragment of t_1 such that the rest of the tree would be all blue. Similarly, $\forall \pi. \exists n. \mathcal{G} blue (t_1 @ \bar{\pi} n)$ is false: the path 1^ω refutes it: there are red nodes beyond all positions on it.

At the same time $\forall \pi. \mathcal{F}(\mathcal{G} blue) \llbracket t_1 \rrbracket_\pi$ is neither true nor false in basic constructive logic. Its truth is equivalent to every path either containing a turn to the left or always going to the right, which is LPO. With WC-N, however, one can conclude that the formula is false: this follows, e.g., from Prop. 5 and falsity of $\forall \pi. \exists n. \mathcal{G} blue (t_1 @ \bar{\pi} n)$.

4.2 Almost Always Blue Trees

We proceed to two concepts of almost always blue trees. We obtain them by mixing induction and coinduction, more precisely, by nesting coinduction into induction in the style of [18].

We start with streams of colors that are almost always blue. They are defined as the least fixed point of a weak until operator in linear-time temporal logic. An equivalent definition is also found in the thesis of C. Raffalli [19]. The weak

until operator, $\mathcal{W} X$, is parameterized over any predicate X on streams of colors and is defined coinductively by

$$\frac{\mathcal{W} X \sigma}{\mathcal{W} X (B \sigma)} \qquad \frac{X \sigma}{\mathcal{W} X (R \sigma)}$$

so that $\mathcal{W} X \sigma$ holds if, whenever the first occurrence of red in σ is encountered, X holds on the suffix after the occurrence. Classically it is equivalent to that σ is either always blue or it is eventually red and X holds on the suffix after the first occurrence of red (which is guaranteed to exist as σ is eventually red). Our definition of $\mathcal{W} X$ avoids upfront decisions of LPO, i.e., whether the stream of colors is always blue or eventually red.

We then take the least fixed point of $\mathcal{W} X$. Define $\mu\mathcal{W}$ inductively in terms of $\mathcal{W} X$ by the (Park-style) rule:

$$\frac{\mathcal{W} \mu\mathcal{W} \sigma}{\mu\mathcal{W} \sigma}$$

As $\mathcal{W} X$ is monotone in X , the above definition makes sense. For the purpose of proof, in particular to avoid explicitly invoking monotonicity of the underlying predicate transformer \mathcal{W} , it is however convenient to use the Mendler-style rule:

$$\frac{\forall \sigma. X \sigma \Rightarrow \mu\mathcal{W} \sigma \quad \mathcal{W} X \sigma}{\mu\mathcal{W} \sigma}$$

The Park-style rule is derivable from the Mendler-style rule. We can also recover the inversion principle for $\mu\mathcal{W}$, thanks to the monotonicity of $\mathcal{W} X$ in X . We use the Mendler-style rule in our Coq formalization, as Coq’s guardedness condition for coinduction nested into induction (as well as induction nested into coinduction) is often too weak to work with the Park style. The Mendler-style rule however requires impredicativity.

The statement $\mu\mathcal{W} \sigma$ does not give a clue as to where to find the red positions in σ or how many they are. Nonetheless it refutes that the stream of colors is infinitely often red (to be formulated below). We have previously scrutinized the definition of $\mu\mathcal{W} \sigma$, placed in a hierarchy of alternative definitions of streams of colors being finitely red, from the viewpoint of constructive mathematics [4]. In the remainder of the paper we refer to $\mu\mathcal{W}$ as mixed inductive-coinductive almost always blueness.

If a stream of colors is eventually always blue, then it is almost always blue:

Proposition 7. $\forall \sigma : color^\omega. \mathcal{F}(\mathcal{G} \text{ blue}) \sigma \Rightarrow \mu\mathcal{W} \sigma.$

Analogously, we define trees that are almost always blue, $\mu\mathcal{W} \tau$, by taking the least fixed point of a weak-until operator for trees. This time, we only give the Park-style rule:

$$\frac{X \tau_0 \quad X \tau_1}{\mathcal{W} X (\tau_0 R \tau_1)} \qquad \frac{\mathcal{W} X \tau_0 \quad \mathcal{W} X \tau_1}{\mathcal{W} X (\tau_0 B \tau_1)} \qquad \frac{\mathcal{W} \mu\mathcal{W} \tau}{\mu\mathcal{W} \tau}$$

If a tree is eventually all blue, then it is almost always blue:

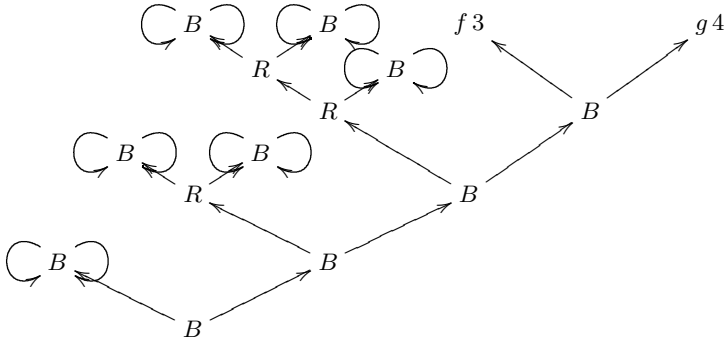


Fig. 2. The tree $t'_1 = g 0$, with the subtrees $f 0, f 1, f 2$ fully expanded

Proposition 8. $\forall \tau : tree. \mathcal{F}(\mathcal{G} \text{ blue}) \tau \Rightarrow \mu \mathcal{W} \tau.$

Our example tree, t_1 , is almost always blue.

Lemma 1. $\mu \mathcal{W} t_1.$

Proof. We have $\mathcal{W} \mu \mathcal{W} t_0$, proved by coinduction, therefore $\mu \mathcal{W} t_0$, which yields $\mathcal{W} \mu \mathcal{W} (t_0 R t_0)$.

We then prove $\mathcal{W} \mu \mathcal{W} t_1$ by coinduction: we already know that $\mathcal{W} \mu \mathcal{W} (t_0 R t_0)$ and by the coinduction hypothesis $\mathcal{W} \mu \mathcal{W} t_1$, hence $\mu \mathcal{W} ((t_0 R t_0) B t_1)$ as required.

To show another proof of almost always blueness, let us also consider a more reddish tree, t'_1 , where the number of red nodes increases in proportion to the depth at which a 0-step is taken for the first time. The tree t'_1 , depicted in figure 2, is defined as $g 0$, where the functions $f, g : nat \rightarrow tree$ are defined by corecursion by

$$f 0 = t_0 \quad f (n + 1) = (f n) R t_0 \quad g n = (f n) B (g (n + 1))$$

The tree t'_1 is almost always blue.

Lemma 2. $\mu \mathcal{W} t'_1.$

Proof. We prove $\forall n. \mathcal{W} \mu \mathcal{W} (f n)$ by induction on n . We then prove $\forall n. \mathcal{W} \mu \mathcal{W} (g n)$ by coinduction, which yields $\mu \mathcal{W} (g 0)$, namely $\mu \mathcal{W} t'_1$, as required.

As usual, the tree-based formulation is stronger than the path-based one. We give the proof here to demonstrate the use of the Mendler-style induction.

Proposition 9. $\forall \tau : tree. \mu \mathcal{W} \tau \Rightarrow \forall \pi : step^\omega. \mu \mathcal{W} \llbracket \tau \rrbracket_\pi.$

Proof. By induction on the proof of $\mu \mathcal{W} \tau$. We are given as the induction hypothesis that, $\forall \tau. X \tau \Rightarrow \forall \pi. \mu \mathcal{W} \llbracket \tau \rrbracket_\pi$ for some predicate X on trees. We

also have $\mathcal{W}X\tau$. We have to prove that, $\forall\pi. \mu\mathcal{W} \llbracket \tau \rrbracket_\pi$. We do so by proving that, $\forall\tau. \mathcal{W}X\tau \Rightarrow \forall\pi. \mathcal{W}\mu\mathcal{W} \llbracket \tau \rrbracket_\pi$ by coinduction, using the main induction hypothesis.

In contrast to the earlier considered case of eventually always blue streams of colours, a proof that a stream of colors is almost always blue does not give us a position at which the suffix of the stream is all blue. Indeed, knowing it, i.e., $\forall\sigma. \mu\mathcal{W}\sigma \Rightarrow \mathcal{F}(\mathcal{G} \textit{blue})\sigma$, is equivalent to LPO [4].

Lemma 3. $(\forall\sigma. \mathcal{F} \textit{red}\sigma \vee \mathcal{G} \textit{blue}\sigma) \Leftrightarrow (\forall\sigma. \mu\mathcal{W}\sigma \Rightarrow \mathcal{F}(\mathcal{G} \textit{blue})\sigma)$.

Our proof to obtain the tree-based formulation from the path-based formulation is sketched as follows. We build infinitely branching trees from binary trees (the function $t2T$ defined below). We then find a decidable bar condition for these infinitely branching trees (Lemma 5). We know that, if every path of a binary tree τ is almost always blue, then a bar exists in the infinitely branching tree θ corresponding to τ (Cor. 2), therefore we can apply Bar Induction on θ (Lemma 7). This in turn proves that the original tree τ is almost always blue (Lemma 8). Below we will make this argument formal.

Our infinitely branching trees, $\theta : Tree$, have nodes labeled by binary trees and edges labeled by non-empty lists of steps. They are defined coinductively by

$$\frac{\tau : tree \quad f : step^+ \rightarrow Tree}{(\tau, f) : Tree}$$

A path in $\theta : Tree$ is characterized by a stream of non-empty lists of steps. For a tree $\theta : Tree$ and a position $q : (step^+)^*$, $\theta@q : Tree$ and $\llbracket \theta \rrbracket_q : (color^+)^*$ are defined naturally by

$$\begin{aligned} (\tau, f)@ \langle \rangle &= (\tau, f) & (\tau, f)@(p q) &= (f p)@q \\ \llbracket (\tau, f) \rrbracket_{\langle \rangle} &= \langle \rangle & \llbracket (\tau, f) \rrbracket_{(p q)} &= (\llbracket \tau \rrbracket_p) (\llbracket (f p) \rrbracket_q) \end{aligned}$$

For $\rho : (step^+)^\omega$, $\llbracket \theta \rrbracket_\rho : (color^+)^\omega$ is defined analogously.

We define a function, $t2T : tree \rightarrow Tree$, from binary trees to infinitely branching trees by corecursion by

$$t2T \tau = (\tau, \lambda p : step^+. t2T \tau @ p)$$

so that, for any position $q : (step^+)^*$, the label of $t2T \tau$ at q is the subtree of τ at $flatten q$ (assuming $flatten$ is extended to finite sequences of non-empty lists in an obvious way). In particular, the streams of colors in $t2T \tau$ and τ along a path $\rho : (step^+)^\omega$ agree up to flattening. This is what the next lemma proves.

Lemma 4. $\forall\tau : tree, \rho : (step^+)^\omega. flatten \llbracket t2T \tau \rrbracket_\rho \sim \llbracket \tau \rrbracket_{(flatten \rho)}$.

A non-empty list of colors $s : color^+$ is *good*, *good s*, if s is of the form B^*R . Formally,

$$\frac{\textit{good } s}{\textit{good } \langle R \rangle} \quad \frac{\textit{good } s}{\textit{good } (B s)}$$

The predicate *good* is decidable:

Lemma 5. $\forall s : color^+. good\ s \vee \neg good\ s.$

We will use $\neg good$ as the bar condition.

A stream over non-empty lists of colors $\alpha : (color^+)^{\omega}$, is wellfounded, $wf\ \alpha$, if α contains a color list that is not good. Formally,

$$\frac{\neg good\ s}{wf\ (s\ \alpha)} \quad \frac{good\ s\ wf\ \alpha}{wf\ (s\ \alpha)}$$

Then we have that, for any $\alpha : (color^+)^{\omega}$, if $flatten\ \alpha$ is almost always blue, then α is wellfounded:

Lemma 6. $\forall \alpha : (color^+)^{\omega}. \mu\mathcal{W}(flatten\ \alpha) \Rightarrow wf\ \alpha.$

As a corollary to Lemmata 4 and 6, we obtain that, if every path of a tree τ is almost always blue, then every path of $t2T\ \tau$ is wellfounded:

Corollary 2. $\forall \tau : tree. (\forall \pi : step^{\omega}. \mu\mathcal{W} \llbracket \tau \rrbracket_{\pi}) \Rightarrow \forall \rho : (step^+)^{\omega}. wf \llbracket (t2T\ \tau) \rrbracket_{\rho}.$

We lift wellfoundedness on streams of nonempty lists of colors to trees:

$$\frac{\forall p : step^+. good \llbracket \tau \rrbracket_p \Rightarrow wf\ (f\ p)}{wf\ (\tau, f)}$$

Now we are to apply Bar Induction (BI) (the generalization of $FAN_{\mathbb{D}}$ from binary trees to infinitely branching trees) to obtain wellfounded trees from trees whose paths are wellfounded. Let P and Q be predicates on lists of nonempty lists of steps. Noticing the isomorphism between natural numbers and nonempty lists of steps, Bar Induction can be expressed as

$$\begin{aligned} (\forall q : (step^+)^*. P\ q \vee \neg P\ q) &\Rightarrow (\forall \rho : (step^+)^{\omega}. \exists n. P\ (\bar{\rho}n)) \Rightarrow \\ (\forall q : (step^+)^*. P\ q \Rightarrow Q\ q) &\Rightarrow (\forall q : (step^+)^*. (\forall p : step^+. Q\ (q * \langle p \rangle)) \Rightarrow Q\ q) \Rightarrow \\ &Q\ \langle \rangle \end{aligned}$$

If we accept BI, we have that, if every path of a tree $\theta : Tree$ is wellfounded, then θ is wellfounded:

Lemma 7. *Assuming BI,* $\forall \theta : Tree. (\forall \rho : (step^+)^{\omega}. wf \llbracket \theta \rrbracket_{\rho}) \Rightarrow wf\ \theta.$

Proof. The claim is an instance of BI by taking P and Q as follows. For any $q : (step^+)^*$, $P\ q$ holds if $\llbracket \theta \rrbracket_q = u * \langle s \rangle$ and $\neg good\ s$. For any $q : (step^+)^*$, $Q\ q$ holds if $P\ q$ or $wf\ (\theta @ q)$.

The following lemma says that, for any tree $\tau : tree$, if $t2T\ \tau$ is wellfounded then τ is almost always blue:

Lemma 8. $\forall \tau : tree. wf\ (t2T\ \tau) \Rightarrow \mu\mathcal{W}\ \tau.$

Finally, putting the above lemmata together, we have that if every path of a tree $\tau : tree$ is almost always blue, then τ is almost always blue:

Proposition 10. *Assuming BI,* $\forall \tau : tree. (\forall \pi : step^{\omega}. \mu\mathcal{W} \llbracket \tau \rrbracket_{\pi}) \Rightarrow \mu\mathcal{W}\ \tau.$

Analogously, we define tree-based always eventually redness of τ as $\mathcal{G}(\mathcal{F} \text{ blue}) \tau$ and tree-based infinitely often redness as $\nu\mathcal{U} \tau$ defined by

$$\frac{\mathcal{U} X \tau_0 \quad \mathcal{U} X \tau_1}{\mathcal{U} X (\tau_0 B \tau_1)} \quad \frac{X \tau_0 \quad X \tau_1}{\mathcal{U} X (\tau_0 R \tau_1)} \quad \frac{\mathcal{U} \nu\mathcal{U} \tau}{\nu\mathcal{U} \tau}$$

Again, the two properties are equivalent:

Proposition 13. $\forall \tau : \text{tree. } \nu\mathcal{U} \tau \Leftrightarrow \mathcal{G}(\mathcal{F} \text{ red}) \tau.$

The tree-based property implies the path-based property:

Proposition 14. $\forall \tau : \text{tree. } \nu\mathcal{U} \tau \Rightarrow (\forall \pi : \text{step}^\omega. \nu\mathcal{U} \llbracket \tau \rrbracket_\pi).$

For the converse implication, we assume FAN_D :

Proposition 15. *Assuming FAN_D , $\forall \tau : \text{tree. } (\forall \pi : \text{step}^\omega. \nu\mathcal{U} \llbracket \tau \rrbracket_\pi) \Rightarrow \nu\mathcal{U} \tau.$*

6 Existential Properties

So far, we have been looking at universal properties over all paths of a tree. In this section, we turn them into existential properties. It turns out that the path-based and tree-based formulations are then necessarily equivalent.

We introduce two new primitives, $\mathcal{F}^\exists X$ and $\mathcal{G}^\exists X$, parameterized over tree predicates X , into our language for trees:

$$\frac{X \tau}{\mathcal{F}^\exists X \tau} \quad \frac{\mathcal{F}^\exists X \tau_0}{\mathcal{F}^\exists X (\tau_0 c \tau_1)} \quad \frac{\mathcal{F}^\exists X \tau_1}{\mathcal{F}^\exists X (\tau_0 c \tau_1)}$$

$$\frac{X (\tau_0 c \tau_1) \quad \mathcal{G}^\exists X \tau_0}{\mathcal{G}^\exists X (\tau_0 c \tau_1)} \quad \frac{X (\tau_0 c \tau_1) \quad \mathcal{G}^\exists X \tau_1}{\mathcal{G}^\exists X (\tau_0 c \tau_1)}$$

In contrast to $\mathcal{F} X \tau$ and $\mathcal{G} X \tau$, the new primitives $\mathcal{F}^\exists X \tau$ and $\mathcal{G}^\exists X \tau$ step down through the tree, picking up one of the two subtrees at every node.

The path-based and tree-based properties that we have considered coincide, with the exception of “always eventually red”, for which the path-based property is stronger. That the converse implication does not hold is witnessed by our example tree t_1 . The reason for the failure is that $\mathcal{G}^\exists (\mathcal{F}^\exists \text{ red}) \tau$ does not require the red nodes to be on the same path.

Proposition 16. 1. $\forall \tau : \text{tree. } (\exists \pi : \text{step}^\omega. \mathcal{G} \text{ blue} \llbracket \tau \rrbracket_\pi) \Leftrightarrow \mathcal{G}^\exists \text{ blue} \tau.$

2. $\forall \tau : \text{tree. } (\exists \pi : \text{step}^\omega. \mathcal{F} \text{ red} \llbracket \tau \rrbracket_\pi) \Leftrightarrow \mathcal{F}^\exists \text{ red} \tau.$

3. $\forall \tau : \text{tree. } (\exists \pi : \text{step}^\omega. \mathcal{F} (\mathcal{G} \text{ blue}) \llbracket \tau \rrbracket_\pi) \Leftrightarrow \mathcal{F}^\exists (\mathcal{G}^\exists \text{ blue}) \tau.$

4. $\forall \tau : \text{tree. } (\exists \pi : \text{step}^\omega. \mathcal{G} (\mathcal{F} \text{ red}) \llbracket \tau \rrbracket_\pi) \Rightarrow \mathcal{G}^\exists (\mathcal{F}^\exists \text{ red}) \tau.$

$\exists \tau : \text{tree. } \mathcal{G}^\exists (\mathcal{F}^\exists \text{ red}) \tau \wedge \neg (\exists \pi : \text{step}^\omega. \mathcal{G} (\mathcal{F} \text{ red}) \llbracket \tau \rrbracket_\pi).$

For a tree having a path that is almost always blue or infinitely often red, we introduce corresponding weak until and strong until operators:

$$\begin{array}{ccccc}
\frac{X \tau_0}{\mathcal{W}^\exists X (\tau_0 R \tau_1)} & \frac{X \tau_1}{\mathcal{W}^\exists X (\tau_0 R \tau_1)} & \frac{\mathcal{W}^\exists X \tau_0}{\mathcal{W}^\exists X (\tau_0 B \tau_1)} & \frac{\mathcal{W}^\exists X \tau_1}{\mathcal{W}^\exists X (\tau_0 B \tau_1)} & \frac{\mathcal{W}^\exists \mu \mathcal{W}^\exists \tau}{\mu \mathcal{W}^\exists \tau} \\
\frac{X \tau_0}{\mathcal{U}^\exists X (\tau_0 R \tau_1)} & \frac{X \tau_1}{\mathcal{U}^\exists X (\tau_0 R \tau_1)} & \frac{\mathcal{U}^\exists X \tau_0}{\mathcal{U}^\exists X (\tau_0 B \tau_1)} & \frac{\mathcal{U}^\exists X \tau_1}{\mathcal{U}^\exists X (\tau_0 B \tau_1)} & \frac{\mathcal{U}^\exists \nu \mathcal{U}^\exists \tau}{\nu \mathcal{U}^\exists \tau}
\end{array}$$

The path-based and tree-based properties are equivalent for both almost always blueness as well as infinitely often redness.

Proposition 17. $\forall \tau : \text{tree}. (\exists \pi : \text{step}^\omega. \mu \mathcal{W} \llbracket \tau \rrbracket_\pi) \Leftrightarrow \mu \mathcal{W}^\exists \tau.$

Proposition 18. $\forall \tau : \text{tree}. (\exists \pi : \text{step}^\omega. \nu \mathcal{U} \llbracket \tau \rrbracket_\pi) \Leftrightarrow \nu \mathcal{U}^\exists \tau.$

Lemma 10. *Assuming PEM, $\forall \tau : \text{tree}. \mu \mathcal{W} \tau \vee \nu \mathcal{U}^\exists \tau$ and $\forall \tau : \text{tree}. \nu \mathcal{U} \tau \vee \mu \mathcal{W}^\exists \tau.$*

7 Related Work

Dam [8] gave a direct translation from CTL* into the modal μ -calculus in a classical setting. Classically, the problem reduces to translation of formulae of the form $E\phi$ where ϕ is a linear-time formula, i.e., ϕ does not contain path quantifiers. Then the translation is given by carefully analyzing the tableau representing $E\phi$ and thereby characterizing infinite paths in the tableau by least or greatest fixpoints.

Formalizations of LTL, CTL* and the modal μ -calculus in Coq have been given by several authors (cf. [17,21,20,7,3]). These works study either LTL (or CTL*, which subsumes LTL) or the modal μ -calculus, and focus on different issues from ours, e.g. issues in encoding modal μ -calculus formulae in higher-order abstract syntax [17] or machine verification of a model checker for the modal μ -calculus [20]. Moreover, our use of mixed induction and coinduction for formalizing almost always blueness and infinitely often redness appears new.

It is known that the Weak König's lemma, WKL, constructively implies FAN_D [13,14]. Moreover, a weakened form of WKL, which additionally requires that the tree under consideration has at most one infinite path, is equivalent to FAN_D [1]. A recent account of the computational content of the principles we use can be found in, e.g., [11,12] in that FAN is realized by the fan functional and bar induction is realized (in some sense) by bar recursion.

In our recent work [4] we studied alternative definitions of streams of colors being finitely red, including $\mathcal{F}(\mathcal{G} \text{ blue}) \sigma$ and $\mu \mathcal{W} \sigma$, and characterized their differences in strength in a precise way by weak instances of PEM. Coquand and Spiwack [6] introduced four notions of finiteness of sets in Bishop's set theory [5]. The two works exhibit a pleasant correspondence [4].

Mixed inductive-coinductive definitions seem to be quite fundamental in applications (e.g., infinitely often red, subtyping [9], the stream processors of

Hancock et al. [15], uniformly continuous functions on a compact real interval [2], weak bisimilarity and delay-free operational semantics of interactive programs [18]). Mendler-style (co)recursion [16] uses that a monotone (co)inductive definition is equivalent to a positive one, via a syntactic left (right) Kan extension along identity (e.g., instead of $\mu X. F X$ one works with $\mu X. \exists Y. (Y \rightarrow X) \times F Y$). We exploited this fact to enable Coq’s structural recursion for an inductive definition with a nested coinductive definition and vice versa, at the price of impredicativity.

8 Conclusion

We analyzed several temporal operators from the point of view of constructive logic. We observed that, with operators like “eventually always” and “almost always”, various classically equivalent definitions become inequivalent. Which one is more adequate in any actual application depends on the purpose at hand. It is also plausible that some of them have a smoother metatheory—more likely the tree-based ones, especially the tree-based “almost always”.

We chose to treat streams and infinite trees as coinductive data, defined the temporal properties of interest in terms of inductive and coinductive predicates, and reasoned about them with induction and coinduction. We are pleased with the concision and elegance this approach offered, compared with more “low-level” arithmetized concepts as is more common in works on constructive mathematics.

We witnessed that the differences between the variations correspond to well-known principles from constructive mathematics, e.g., the implication from the path-based “eventually” operator to tree-based “eventually” is exactly the decidable Fan Theorem etc.

This demonstrates, to our mind, that the studies into constructive mathematics, which were initiated by Brouwer and elaborated by Bishop and others, and are not particularly well-known in the programming languages community, are not without significance for modern formalized programming theory or dependently typed programming.

In future work, we wish to reach a deeper understanding of the computational aspects in our results and their implications for programming and reasoning about interactive and concurrent systems.

Acknowledgments. We are indebted to Christine Paulin-Mohring, Hugo Herbelin, Thorsten Altenkirch for fruitful discussions.

K. Nakata and T. Uustalu’s research was supported by ERDF through the Estonian Centre of Excellence in Computer Science (EXCS). M. Bezem’s visit to Estonia in Feb. 2011 was supported by the same project.

References

1. Berger, J., Ishihara, H.: Brouwer’s fan theorem and unique existence in constructive analysis. *Math. Log. Quart.* 51(4), 360–364 (2005)
2. Berger, U.: From coinductive proofs to exact real arithmetic: theory and applications. *Logical Methods in Comput. Sci.* 7(1) (2011)

3. Bertot, Y., Castéran, P.: *Interactive Theorem Proving and Program Development: Coq'Art: The Calculus of Inductive Constructions*. Springer, Heidelberg (2004)
4. Bezem, M., Nakata, K., Uustalu, T.: On streams that are finitely red (submitted for publication 2011) (manuscript)
5. Bishop, E.: *Foundations of Constructive Analysis*. McGraw-Hill, New York (1967)
6. Coquand, T., Spiwack, A.: Constructively finite? In: Laureano Lambán, L., Romero, A., Rubio, J. (eds.) *Scientific Contributions in Honor of Mirian Andrés Gómez*. Universidad de La Rioja (2010)
7. Coupet-Grimal, S.: An axiomatization of Linear Temporal Logic in the Calculus of Inductive Constructions. *J. of Logic and Comput.* 13(6), 801–813 (2003)
8. Dam, M.: CTL* and ECTL* as fragments of the modal μ -calculus. *Theor. Comput. Sci.* 126(1), 77–96 (1994)
9. Danielsson, N.A., Altenkirch, T.: Subtyping, declaratively: an exercise in mixed induction and coinduction. In: Bolduc, C., Desharnais, J., Ktari, B. (eds.) *MPC 2010*. LNCS, vol. 6120, pp. 100–118. Springer, Heidelberg (2010)
10. Emerson, E.A.: Temporal and modal logic. In: van Leeuwen, J. (ed.) *Handbook of Theoretical Computer Science*, vol. B, pp. 905–1072. MIT Press (1990)
11. Escardó, M.H., Oliva, P.: Selection functions, bar recursion and backward induction. *Math. Struct. in Comput. Sci.* 20(2), 127–168 (2010)
12. Escardó, M.H., Oliva, P.: What sequential games, the Tychonoff Theorem and the double-negation shift have in common. In: *Proc. of 3rd ACM SIGPLAN Wksh. on Mathematically Structured Functional Programming, MSFP 2010*, pp. 21–32. ACM Press (2010)
13. Ishihara, H.: An omniscience principle, the König Lemma and the Hahn-Banach theorem. *Math. Log. Quart.* 36(3), 237–240 (1990)
14. Ishihara, H.: Weak König's lemma implies Brouwer's fan theorem: a direct proof. *Notre Dame J. of Formal Logic* 47(2), 249–252 (2006)
15. Hancock, P., Pattinson, D., Ghani, N.: Representations of stream processors using nested fixed points. *Logical Methods in Comput. Sci.* 5(3) (2009)
16. Mendler, N.P.: Inductive types and type constraints in the second-order lambda calculus. *Ann. of Pure and Appl. Logic* 51(1-2), 159–172 (1991)
17. Miculan, M.: On the formalization of the modal μ -Calculus in the Calculus of Inductive Constructions. *Inform. and Comput.* 164(1), 199–231 (2001)
18. Nakata, K., Uustalu, T.: Resumptions, weak bisimilarity and big-step semantics for While with interactive I/O: an exercise in mixed induction-coinduction. In: Aceto, L., Sobocinski, P. (eds.) *Proc. of 7th Wksh. on Structural Operational Semantics, SOS 2010*, *Electron. Proc. in Theor. Comput. Sci.*, vol. 32, pp. 57–75 (2010)
19. Raffalli, C.: *L' Arithmétiques Fonctionnelle du Second Ordre avec Points Fixes*. PhD thesis, Université Paris VII (1994)
20. Sprenger, C.: A Verified Model Checker for the Modal μ -calculus in Coq. In: Steffen, B. (ed.) *TACAS 1998*. LNCS, vol. 1384, pp. 167–183. Springer, Heidelberg (1998)
21. Tsai, M.-H., Wang, B.-Y.: Formalization of CTL* in Calculus of Inductive Constructions. In: Okada, M., Satoh, I. (eds.) *ASIAN 2006*. LNCS, vol. 4435, pp. 316–330. Springer, Heidelberg (2008)
22. Troelstra, A.S., van Dalen, D.: *Constructivism in Mathematics*, vol. I, II. North-Holland (1988)