Journal of Universal Computer Science, vol. 11, no. 12 (2005), 2086-2095
submitted: 5/12/05, accepted: 10/12/05, appeared: 28/12/05 © J.UCS

A Direct Proof of the Equivalence between Brouwer's Fan Theorem and König's Lemma with a Uniqueness Hypothesis[1]

Helmut Schwichtenberg
(Mathematisches Institut, Universität München
Theresienstraße 39, 80333 München, Germany
Helmut.Schwichtenberg@mathematik.uni-muenchen.de)

Abstract: From results of Ishihara it is known that the weak (that is, binary) form of König's lemma (WKL) implies Brouwer's fan theorem (Fan). Moreover, Berger and Ishihara [MLQ 2005] have shown that a weakened form WKL! of WKL, where as an additional hypothesis it is required that in an effective sense infinite paths are unique, is equivalent to Fan. The proof that WKL! implies Fan is done explicitely. The other direction (Fan implies WKL!) is far less directly proved; the emphasis is rather to provide a fair number of equivalents to Fan, and to do the proofs economically by giving a circle of implications. Here we give a direct construction. Moreover, we go one step further and formalize the equivalence proof (in the Minlog proof assistant). Since the statements of both Fan and WKL! have computational content, we can automatically extract terms from the two proofs. It turns out that these terms express in a rather perspicuous way the informal constructions.

Key Words: Brouwer's fan theorem, König's lemma
Category: F.1

1 Introduction

In the framework of Bishop's constructive mathematics [4], Douglas Bridges and Fred Richman proved in [5] the equivalence of Brouwer's fan theorem with some other propositions, among them a positivity property, namely that every positively valued uniformly continuous function on [0,1] has a positive infimum. More recently, many other equivalents of the fan theorem have been found, most of them involving an effective uniqueness condition for potential solutions [1, 2].

From the results of [6] it is known that the weak (that is, binary) form of König's lemma (WKL for short) implies Brouwer's fan theorem (Fan for short); see below for precise formulations of these statements. Moreover, a direct proof of this implication has been given by Ishihara in 2002 (to appear in [7]). In [2], it is shown that a weakened form of WKL, where as an additional hypothesis it is required that in an effective sense infinite paths are unique, is equivalent to Fan.

Here we give a direct proof of the equivalence of Fan with WKL!. One direction (WKL! implies Fan) is essentially the proof in [7], enhanced by the addi-

[1] C. S. Calude, H. Ishihara (eds.). *Constructivity, Computability, and Logic. A Collection of Papers in Honour of the 60th Birthday of Douglas Bridges.*

tional requirement that the tree extension to be constructed satisfies the effective uniqueness condition (as in [2]). The other direction (Fan implies WKL!) is far less directly proved in [2], where the emphasis rather was to provide a fair number of equivalents to Fan, and to do the proof economically by giving a circle of implications. Hence it might be of interest to see a direct construction.

Moreover, in the appendix we go one step further and formalize the equivalence proof (in the Minlog proof assistant). Since the formulation of both Fan and WKL! has computational content (of types τ_F and τ_K, say), we can extract from the two proofs two terms t_{FK} and t_{KF} (of types $\tau_F \to \tau_K$ and $\tau_K \to \tau_F$, respectively). It turns out that both extracted terms express in a rather perspicuous way the informal constructions.

2 Basic definitions

Let $\mathbf{N}$ be the type of natural numbers, $\mathbf{B}$ the type of booleans $\mathtt{tt}$, $\mathtt{ff}$ and $\mathbf{L}(\mathbf{B})$ the type of lists of booleans. It is convenient to write lists in reverse order, that is, add elements at the end. We fix the types of some variables and state their intended meaning:

a, b, c	of type $\mathbf{L}(\mathbf{B})$	for nodes,
r, s, t	of type $\mathbf{L}(\mathbf{B}) \to \mathbf{B}$	for decidable sets of nodes,
as, bs, cs	of type $\mathbf{N} \to \mathbf{L}(\mathbf{B})$	for sequences of nodes,
f, g, h	of type $\mathbf{N} \to \mathbf{B}$	for paths,
n, m, k, i, j	of type $\mathbf{N}$	for natural numbers,
p, q	of type $\mathbf{B}$	for booleans,
ns, ms, ks	of type $\mathbf{N} \to \mathbf{N}$	for sequences of natural numbers.

Let $|a|$ be the *length* of a. Let $\bar{a}(n)$ denote the initial segment of a of length n, if $n \leq |a|$, and a otherwise. Similarly let $\bar{f}(n)$ denote the initial segment of f of length n, that is, the list $:f(0) :: f(1) \cdots :: f(n-1)$. Let $(a)_n$ denote the n-th element of a, if $n < |a|$, and $\mathtt{tt}$ otherwise. Clearly

$$|a| = n + 1 \to \bar{a}(n) :: (a)_n = a, \tag{1}$$

$$n < m < |a| \to \overline{\bar{a}(m)}(n) = \bar{a}(n). \tag{2}$$

We will also need to deal with lists of pairs of booleans, using variables

bc	of type $\mathbf{L}(\mathbf{B} \times \mathbf{B})$	for pair nodes,
ss	of type $\mathbf{L}(\mathbf{B} \times \mathbf{B}) \to \mathbf{B}$	for decidable sets of pair nodes,
gh	of type $\mathbf{N} \to \mathbf{B} \times \mathbf{B}$	for paths, w.r.t. pair nodes,
pq	of type $\mathbf{B} \times \mathbf{B}$	for pairs of booleans.

To switch from $\mathbf{L}(\mathbf{B}\times\mathbf{B})$ to $\mathbf{L}(\mathbf{B})$ and back we use Zip and Unzip:

$$\begin{aligned}
&\mathrm{Zip}(bc :: pq) := \mathrm{Zip}(bc) :: \mathrm{Lft}(pq) :: \mathrm{Rht}(pq), \quad \mathrm{Zip}(\mathrm{Nil}) := \mathrm{Nil},\\
&\mathrm{Unzip}(n+1, a :: p :: q) := \mathrm{Unzip}(n,a) :: (p,q),\\
&\mathrm{Unzip}(n+1, {:}p) := \mathrm{Unzip}(n+1,\mathrm{Nil}) := \mathrm{Unzip}(0,a) := \mathrm{Nil}.
\end{aligned}$$

Here $\mathrm{Lft}, \mathrm{Rht}\colon \mathbf{B}\times\mathbf{B}$ denote the two projections. When in a context containing gh we use g or h, we mean $\mathrm{Lft}\circ gh$ or $\mathrm{Rht}\circ gh$. Using this notation, for $\mathbf{N}\to\mathbf{B}\times\mathbf{B}$ and $\mathbf{N}\to\mathbf{B}$ we similarly define Fzip and Funzip, by

$$\begin{aligned}
&\mathrm{Fzip}(gh, 2i) := g(i), \quad \mathrm{Fzip}(gh, 2i+1) := h(i),\\
&\mathrm{Funzip}(f,i) := \big(f(2i), f(2i+1)\big).
\end{aligned}$$

Clearly $\mathrm{Fzip}(\mathrm{Funzip}(f)) = f$ and $\mathrm{Funzip}(\mathrm{Fzip}(gh)) = gh$. Moreover,

$$\mathrm{Unzip}(|bc|, \mathrm{Zip}(bc)) = bc, \tag{3}$$

$$|\mathrm{Zip}(bc)| = 2|bc|, \tag{4}$$

$$\mathrm{Zip}(\overline{\mathrm{Funzip}(f)}(m)) = \bar{f}(2m). \tag{5}$$

f is a *path in* t if all its initial segments $\bar{f}(n)$ are in t. Call t *infinite* if for every n there is a node of length n in t. Call t a *tree* if it is downwards closed, i.e., $\forall_a \forall_{n\le|a|}.\, a\in t \to \bar{a}(n)\in t$. Call s a *bar* if each path hits s, that is, $\forall_f \exists_m\, \bar{f}(m)\in s$. Call s a *uniform bar* if, for some k, each path hits s before k, that is, $\exists_k \forall_f \exists_{m\le k}\, \bar{f}(m)\in s$. We say that t has *at most one path* if it satifies the following *effective uniqueness condition*:

$\mathrm{EffUniq}_t$: for any g, h and n with $\bar{g}(n)\neq\bar{h}(n)$, there is an m such that it is impossible that both $\bar{g}(m)$ and $\bar{h}(m)$ are in t.

We can now formulate the two statements whose equivalence we prove:

Fan: Every bar is uniform.

WKL!: Every infinite tree with at most one path has a path.

3 WKL! implies Fan

Let s be given and assume $\mathrm{Bar}(s)$: $\forall_f \exists_m\, \bar{f}(m)\in s$. We need to construct a uniform bound, that is, some k such that each path hits s before k, that is, $\forall_f \exists_{m\le k}\, \bar{f}(m)\in s$. Let $r := \{\, a \mid \exists_{n\le|a|}\, \bar{a}(n)\in s \,\}$ be the upwards closure of s.

Call n *big* if every node of length n is in r. It suffices to construct a big k.

We extend the complement of r to an infinite tree t satisfying $\mathrm{EffUniq}_t$, so that WKL! can be applied. The idea is that for r finite we extend the leftmost of its longest nodes by $\mathtt{tt}$'s.

More precisely, we define the extension t as follows. A node b belongs to t if if it is not in r. If it is, check whether its length $|b|$ is big. If not, b is not in t. If it is big, let k be such that $k+1$ is big but k is not.

```
k+1:                     + + + + .... + +
k:        +    ....    +    a
```

Let a be the unique node such that on its length k to the left of a there are only nodes in r, but a itself is not in r. Then b is in t iff it is the extension of a by $\mathtt{tt}$'s to the length of b. So

$$t := \{\, b \mid b \in r \to \mathrm{Big}(|b|) \wedge b = a :: \mathtt{tt} \cdots :: \mathtt{tt} \,\}$$

We show that t is infinite. So let n be given. If n is big, let a be as above. Then $a :: \mathtt{tt} \cdots :: \mathtt{tt}$ of length n is in t. If n is not big, an arbitrary node b of length n that is not in r is in t.

We show that t is a tree. So let $b \in t$, $n \leq |b|$; we must show $\bar{b}(n) \in t$. So assume $\bar{b}(n) \in r$. Then also $b \in r$, because r is upwards closed. Hence $|b|$ is big and $b = a :: \mathtt{tt} \cdots :: \mathtt{tt}$. By definition of t, any initial segment of b is in t.

We show that t satisfies $\mathrm{EffUniq}_t$. Consider g, h, n such that $\bar{g}(n) \neq \bar{h}(n)$. Since every path hits s, there is an $m \geq n$ such that $\bar{g}(m)$, $\bar{h}(m)$ both are in its upwards closure r. Assume for contradiction that both are in t. Then by construction of t both are of the form $a :: \mathtt{tt} \cdots :: \mathtt{tt}$ and of the same length, hence equal, and therefore also $\bar{g}(n) = \bar{h}(n)$. This is the desired contradiction.

Now WKL! gives a path f in t. It must hit the bar s, hence r, and at this length we have the desired big k.

4 Fan implies WKL!

Given an infinite tree t satisfying $\mathrm{EffUniq}_t$, we construct a path in t.

We derive from Fan a related auxiliary proposition PFan, refering to pair nodes. Using PFan, from $\mathrm{EffUniq}_t$ we can prove

> $\mathrm{FanBound}_t$: For every n there is a $k \geq n$ such that for all b, c of length k and in t we have $\bar{b}(n) = \bar{c}(n)$,

From $\mathrm{FanBound}_t$ we then easily construct a path in t.

4.1 Fan implies PFan

From Fan we want to prove

$$\begin{aligned}\text{PFan: } \forall_n \forall_{ss}.\, &\big(\forall_{bc} \forall_{n \leq |bc|}.\, \overline{bc}(n) \in ss \to bc \in ss\big) \to \\ &\big(\forall_{gh}.\, \bar{g}(n) \neq \bar{h}(n) \to \exists_m\, \overline{gh}(m) \in ss\big) \to \\ &\exists_k \forall_{gh}.\, \bar{g}(n) \neq \bar{h}(n) \to \overline{gh}(k) \in ss.\end{aligned}$$

Given n, ss. Assume $\mathrm{Upclosed}_{ss}$: $\forall_{bc}\forall_{n\le|bc|}.\,\overline{bc}(n)\in ss\to bc\in ss$ and Bar_{ss}: $\forall_{gh}.\,\bar g(n)\ne\bar h(n)\to\exists_m\,\overline{gh}(m)\in ss$. To construct: k. We use Fan for

$$s_n := \{\, a \mid \forall_{i<n}.\,(a)_{2i}\ne(a)_{2i+1}\to \mathrm{Unzip}(\lfloor|a|/2\rfloor, a)\in ss\,\}.$$

We need to show that every path f hits s_n. Let f be given.

Case $f(2i)=f(2i+1)$ for all $i<n$. Then

$$(\bar f(2n))_{2i} = f(2i) = f(2i+1) = (\bar f(2n))_{2i+1}$$

for all $i<n$, hence $\bar f(2n)\in s_n$.

Case $f(2i)\ne f(2i+1)$ for some $i<n$. Then $\mathrm{Funzip}(f)$ is some gh with $\bar g(n)\ne\bar h(n)$, for

$$\begin{aligned}(\bar g(n))_i &= g(i) = (\mathrm{Lft}\circ\mathrm{Funzip}(f))(i) = f(2i),\\ (\bar h(n))_i &= h(i) = (\mathrm{Rht}\circ\mathrm{Funzip}(f))(i) = f(2i+1).\end{aligned}$$

By Bar_{ss} we can find an m such that $bc := \overline{\mathrm{Funzip}(f)}(m)\in ss$; because of $\mathrm{Upclosed}_{ss}$ we may assume $n\le m$. Now with

$$a := \mathrm{Zip}(bc) = \mathrm{Zip}(\overline{\mathrm{Funzip}(f)}(m)) = \bar f(2m)\quad\text{(by (5))}$$

we have $a\in s_n$, for by (4) and (3)

$$\mathrm{Unzip}(\lfloor|a|/2\rfloor, a) = \mathrm{Unzip}(\lfloor|\mathrm{Zip}(bc)|/2\rfloor, \mathrm{Zip}(bc)) = \mathrm{Unzip}(|bc|, \mathrm{Zip}(bc)) = bc.$$

Now by Fan we have k such that $\forall_f\exists_{m\le k}\,\bar f(m)\in s_n$. Since s_n clearly is upwards closed, we may assume that k is even and $2n\le k$. We show

$$\forall_{gh}.\,\bar g(n)\ne\bar h(n)\to\overline{gh}(\lfloor k/2\rfloor)\in ss.$$

So let gh with $\bar g(n)\ne\bar h(n)$ be given. Let $f := \mathrm{Fzip}(gh)$. By assumption $a := \bar f(k)\in s_n$. Now $a = {:}g(0)::h(0)::\cdots::g(\lfloor k/2\rfloor-1)::h(\lfloor k/2\rfloor-1)$. Because of $\bar g(n)\ne\bar h(n)$ and $n\le\lfloor k/2\rfloor$ we have $\exists_{i<n}(a)_{2i}\ne(a)_{2i+1}$. Hence

$$\overline{gh}(\lfloor k/2\rfloor) = \mathrm{Unzip}(\lfloor k/2\rfloor, a) = \mathrm{Unzip}(\lfloor|a|/2\rfloor, a)\in ss.$$

4.2 A bound for the fan

We shall prove

> $\mathrm{FanBound}_t$: For every n there is a $k\ge n$ such that for all b, c of length k and in t we have $\bar b(n)=\bar c(n)$.

The proof uses PFan – which we just proved from Fan – and also $\mathrm{EffUniq}_t$, to obtain one of its hypotheses. We fix n and apply PFan to

$$ss_t := \{\, bc \mid b \in t \to c \in t \to F \,\}.$$

$\forall_{bc}\forall_{n\leq|bc|}.\,\overline{bc}(n) \in ss_t \to bc \in ss_t$ holds because t is a tree.

$$\forall_{gh}.\,\bar{g}(n) \neq \bar{h}(n) \to \exists_m\, \overline{gh}(m) \in ss_t$$

follows from $\mathrm{EffUniq}_t$: it provides an m such that it is impossible that both $\bar{g}(m)$ and $\bar{h}(m)$ are in t. Hence $\overline{gh}(m) \in ss_t$.

Now PFan yields a k such that $\forall_{gh}.\,\bar{g}(n) \neq \bar{h}(n) \to \overline{gh}(k) \in ss_t$. We may assume $k \geq n$. To prove $\mathrm{FanBound}_t$, let b, c of length k and in t be given; we have to show $\bar{b}(n) = \bar{c}(n)$. Let gh be the extension of bc by (pairs of) $\mathtt{tt}$'s. Assume for contradiction $\bar{b}(n) \neq \bar{c}(n)$, hence $\bar{g}(n) \neq \bar{h}(n)$. Therefore $\overline{gh}(k) \in ss_t$, that is $\bar{b}(k)$ and $\bar{c}(k)$ cannot both be in t. This is the desired contradiction.

4.3 Construction of the path

Let ks_0 be the function provided by $\mathrm{FanBound}_t$, and let ks be the canonical monotone upper bound of ks_0. Write k_n for $ks(n)$, a_n for $as(n)$, b_n for $bs(n)$ etc.

Because t is infinite, we have a_n of length n in t. Define $b_n \in t$ by

$$b_n := \overline{a_{k_n}}(n).$$

We claim that the b_n's extend each other and hence make up a path, i.e., with $f(n) := (b_{n+1})_n$ we have $\bar{f}(n) = b_n \in t$. It suffices to prove

$$b_n = \overline{b_{n+1}}(n), \tag{6}$$

for then we obtain $\bar{f}(n) = b_n$ by induction, as follows. The base case is obvious, and in the step case we have by (1)

$$\bar{f}(n+1) = \bar{f}(n) :: f(n) = b_n :: (b_{n+1})_n = \overline{b_{n+1}}(n) :: (b_{n+1})_n = b_{n+1}.$$

For (6) we apply $\mathrm{FanBound}_t$ to the two nodes a_{k_n} and $\overline{a_{k_{n+1}}}(k_n)$, which are both of length k_n and in t. Hence by (2)

$$b_n = \overline{a_{k_n}}(n) = \overline{\overline{a_{k_{n+1}}}(k_n)}(n) = \overline{a_{k_{n+1}}}(n) = \overline{\overline{a_{k_{n+1}}}(n+1)}(n) = \overline{b_{n+1}}(n).$$

References

1. Josef Berger, Douglas Bridges, and Peter Schuster. The fan theorem and unique existence of maxima. Submitted, 2005.
2. Josef Berger and Hajime Ishihara. Brouwer's fan theorem and unique existence in constructive analysis. *Math. Log. Quart.*, 51(4):360–364, 2005.

3. Ulrich Berger, Stefan Berghofer, Pierre Letouzey, and Helmut Schwichtenberg. Program extraction from normalization proofs. *Studia Logica*, To appear, 2005.
4. Errett Bishop. *Foundations of Constructive Analysis.* McGraw-Hill, New York, 1967.
5. Douglas Bridges and Fred Richman. *Varieties of Constructive Mathematics.* Cambridge University Press, Cambridge, 1987.
6. Hajime Ishihara. An omniscience principle, the König lemma and the Hahn-Banach theorem. *Zeitschr. f. math. Logik und Grundlagen d. Math.*, 36:237–240, 1990.
7. Hajime Ishihara. Weak König lemma implies Brouwer's fan theorem: a direct proof. To appear: Notre Dame J. Formal Logic, 2005.
8. Erik Palmgren. Internalising modified realisability in constructive type theory. *Logical Methods in Computer Science*, 1(2):1–7, 2005. http://www.lmcs-online.org/.

A Appendix: Formalization and extraction

The proofs above have been done in sufficient detail to allow formalization. By the well known method of (modified) realizability (see [8, 3] for recent expositions and applications) one can extract from a (constructive) proof of a formula with computational content a term that "realizes" the formula.

Of course, the algorithm expressed by such a term is already – at least implicitely – present in the proof. This is particularly true for the proof that WKL! implies Fan, which essentially consists of a construction. However, it spite of this fact there are good reasons to be interested in formalization and term extraction:

- It can be important to know for sure (and to be able to machine check) that in a proof nothing has been overlooked.
- The same applies to the algorithm implicit in the proof: even if the latter is correct, errors may occur in the implemention of the algorithm.
- Finally, even if the algorithm is correctly implemented, for sensitive applications customers may (and do) require a formal proof that the code implementing algorithm is correct.

The realizability method takes care of all these points.

For space reasons, we neither give nor comment the formalizations[2], but only show the extracted terms (verbatim as produced by the Minlog system), and where necessary explain which algorithm they represent.

Notice that our trees and bars are decidable, and given by computable boolean functions. We use `impb` for boolean implication, `AllBNat` of type $\mathbf{N} \to (\mathbf{N} \to \mathbf{B}) \to \mathbf{B}$ for the bounded universal quantifier $\forall_{i<n} Q(i)$ and `AllBList` of type $\mathbf{N} \to (\mathbf{L}(\mathbf{B}) \to \mathbf{B}) \to \mathbf{B}$ for $\forall_a . |a| = n \to Q(a)$.

[2] The first formalization (in the Minlog proof assistant) of a related proof (that WKL implies Fan) has been done in a slightly different setup by Klaus Thiel, Freiric Barral, Josef Berger, Basil Karadice and Stefan Schimanski; they also extracted a term.

A.1 WKL! implies Fan

```
[wklu0,r1,inf2]
 inf2
 (wklu0(Ext(Up r1))
  ([n4][if (AllBList n4(Up r1))
           (LExt(UL n4(Up r1))n4)
           (BMu n4(Up r1))])
  ([f4,f5]NatMax(inf2 f4 max inf2 f5)))
```

We are given a functional `wklu0` realizing WKL!, a tree `r1` and a realizer `inf2` for its infinity. Apply `inf2` to the result of applying `wklu0` to (1) the extension of the complement of the upwards closure `Up r1` of `r1`, (2) a witness for its infinity and (3) a witness for the effective uniqueness of its paths. For (2), we are given `n4`. If `n4` is big, take the left extension (by tt's) of the uppermost leftmost node in the tree. If `n4` is not big, take the first node of length `n4` in the tree. For (3), we are given `f4` and `f5`. Take the function mapping `n` to the `max` of itself and what the witness for infinity gives at `f4` and `f5`.

A.2 Fan implies WKL!

A.2.1 Fan implies PFan

```
[fan0,n1,ss2,pbar3]
 fan0
 ([a4]
   AllBNat n1
   ([n5](a4__(n5+n5)=a4__Succ(n5+n5)impb False)impb
         ss2(Half Lh a4 unzip a4)))
 ([f4]
   [if (AllBNat n1([n5]f4(n5+n5)=f4(Succ(n5+n5))))
     (n1+n1)
     (pbar3([n5]f4(n5+n5)@f4(Succ(n5+n5)))+n1+
     pbar3([n5]f4(n5+n5)@f4(Succ(n5+n5)))+
     n1)])+n1
```

We are given a functional `fan0` realizing `Fan`, a number `n1`, a set `ss2` of pair nodes and a functional `pbar3` mapping a pair path differing at `n` to a bar. Apply `fan0` to (1) the set of all nodes `a4` such that for all `n5<n1`, if the elements of `a4` at `2*n5` and `2*n5+1` are distinct, then the result of unzipping `a4` at half of its length is in `ss2`, and (2) a witness that every path `f4` hits this set, and take the `max` of this number and `n1`. For (2), distinguish cases whether for some n5`<n1`, `f4` at `2*n5` equals `f4` at `2*n5+1`. If so, take `2*n1`. If not, let `n` be the result of applying `pbar3` to the unzipped form of `f4`, and take `2*n+2*n1`.

A.2.2 A bound for the fan

```
[pfan0,r1,uniq2,n3]
 n3 max
 pfan0 n3
 ([bc4]r1(([pq5]left pq5)map bc4)impb
       r1(([pq5]right pq5)map bc4)impb False)
 ([gh4]uniq2([n5]left(gh4 n5))([n5]right(gh4 n5))n3)
```

We are given a functional `pfan0` realizing `PFan`, a tree `r1`, a realizer `uniq2` of the effective uniqueness property and a number `n3`. Take the max of `n3` and the result of applying `pfan0` to this number, the set of all pair nodes whose left and right parts cannot both be in `r1`, and the functional mapping a pair path to the result of applying `uniq2` to it and this number.

A.2.3 Construction of the path

```
[r0,ns1,as2,n3]as2(ns1(Succ n3))__n3
```

We are given a tree `r0`, a sequence `ns1` of numbers provided by `FanBound`, a sequence `as2` of nodes witnessing the infinity of `r0`, and an argument `n3` for the path to be constructed. Take the `n3`-th element of the sequence `as2` applied to `ns1(n3+1)`.

A.2.4 Putting the parts together

```
[fan0,r1,as2,uniq3]
 cPath r1(Mon(cACNat(cFanBound(cFanImpPFan fan0)r1 uniq3)))
 (cACListBoole as2)
```

`cACNat` and `cACListBoole` can be ignored, for they are realisers of choice axioms and hence identities.

We can unfold the contents of the auxiliary propositions, by "animating" them. The result is

```
[fan0,r1,as2,uniq3,n4]
 cACListBoole as2
 (Mon
  (cACNat
   ([n6]n6 max
     (fan0
      ([a7]
        AllBNat n6
```

```
        ([n8]
          (a7__(n8+n8)=a7__Succ(n8+n8)impb False)impb
          r1(([pq9]left pq9)map Half Lh a7 unzip a7)impb
          r1(([pq9]right pq9)map Half Lh a7 unzip a7)impb
          False))
      ([f7]
        [if (AllBNat n6([n8]f7(n8+n8)=f7(Succ(n8+n8))))
          (n6+n6)
          (uniq3([n8]f7(n8+n8))([n8]f7(Succ(n8+n8)))n6+n6+
          uniq3([n8]f7(n8+n8))([n8]f7(Succ(n8+n8)))n6+
          n6)])+n6)))
  (Succ n4))__n4
```

So from the formalized proof that Fan implies WKL! we have automatically extracted a rather readable term expressing its computational content. When supplied with concrete arguments, this term can be evaluated (i.e., normalized) to the nodes of the path.