ruitenburg's theorem mechanized and contextualized

or

certified computation of periodic sequences in ipc resistant to meaningful generalizations

Tadeusz Litak

FiCS, 19–20 February 2024

Informatik 8, FAU Erlangen-Nürnberg

- in JSL 1984, a surprising result about periodic sequences in IPC by Wim Ruitenburg
- heavily syntactic proof
- for years, not too well-known a result

  the late Sergey Mardaev one of the few researchers using it
- more recent references quoting Ruitenburg: Ghilardi et al. 2016 (FoSSaCS 2016, ACM ToCL 2020) or Humberstone's monograph
- most commonly quoted in the context of definability (eliminability) of fixpoints, where it is just one of possible lines of attack: more later
- finally, Ghilardi and Santocanale (AiML 2018, MSCS 2020) provided a semantic proof via duality

  . . . which does not provide tight bounds, unlike Ruitenburg's approach
- the property deserves still more attention: a surprising generalization of local finiteness

what I did, mostly 2015–17

- formalized the proof in the Coq proof assistant ⇒ extracting a verified program computing cycles in IPC
- carried out a comparative study with other natural classes of implicative logics ⇒

  obtaining a confirmation that Ruitenburg did something remarkable
- referees and Alexis Saurin kindly asked me to focus on the first point, but some discussion of the second one is unavoidable
  - the formalization should be upgraded anyway
  - we need to explain why this paper belongs in FiCS at all

A wild property in a zoo of logics

- Consider a formula A

- Consider a formula A
- Fix a propositional variable p

  It can be thought of as representing a context hole . . . or an argument variable, while A itself is a polynomial in that variable (other propositional variables being additional constants).

- Consider a formula A
- Fix a propositional variable p

  It can be thought of as representing a context hole . . . or an argument variable, while A itself is a polynomial in that variable (other propositional variables being additional constants).
- Given any other formula B, write $A(B)$ for the result of substituting B for p.

- Consider a formula A
- Fix a propositional variable p

  It can be thought of as representing a context hole ... or an argument variable, while A itself is a polynomial in that variable (other propositional variables being additional constants).

- Given any other formula B, write $A(B)$ for the result of substituting B for p.
- Define the obvious iterated substitution operation:

$$A^0(p) := p, \quad A^{n+1}(p) := A(A^n(p)).$$

In future, we can just write A^n instead of $A^n(p)$, where no confusion arises

- Consider a formula A
- Fix a propositional variable p

  It can be thought of as representing a context hole . . . or an argument variable, while A itself is a polynomial in that variable (other propositional variables being additional constants).

- Given any other formula B, write $A(B)$ for the result of substituting B for p.
- Define the obvious iterated substitution operation:

$$A^0(p) := p, \quad A^{n+1}(p) := A(A^n(p)).$$

  In future, we can just write A^n instead of $A^n(p)$, where no confusion arises

- Question: modulo provable equivalence in CPC, when are you going to enter a cycle?

- Consider a formula A
- Fix a propositional variable p

  It can be thought of as representing a context hole ... or an argument variable, while A itself is a polynomial in that variable (other propositional variables being additional constants).
- Given any other formula B, write $A(B)$ for the result of substituting B for p.
- Define the obvious iterated substitution operation:

$$A^0(p) := p, \quad A^{n+1}(p) := A(A^n(p)).$$

  In future, we can just write A^n instead of $A^n(p)$, where no confusion arises
- Question: modulo provable equivalence in CPC, when are you going to enter a cycle?
- Actually, how do you know you must enter a cycle at all?

- That is right: for any A,

$$\vdash_{\mathsf{CPC}} A \leftrightarrow A^3.$$

- That is right: for any A,

$$\vdash_{\mathsf{CPC}} A \leftrightarrow A^3.$$

- How about other logics?

- That is right: for any A,

$$\vdash_{\mathsf{CPC}} A \leftrightarrow A^3.$$

- How about other logics?
- Say a logic L has "periodic sequences" property if we can always find b s.t. $\vdash_{\mathsf{L}} A^b \leftrightarrow A^{b+c}$ holds

  ($A^b \dashv\vdash_{\mathsf{L}} A^{b+c}$ when "well-behaved" implication missing)

  Generally, $A^b \equiv_{\mathsf{L}} A^c$ is a good notation

- That is right: for any A,

$$\vdash_{\mathsf{CPC}} A \leftrightarrow A^3.$$

- How about other logics?
- Say a logic L has "periodic sequences" property if we can always find b s.t. $\vdash_{\mathsf{L}} A^b \leftrightarrow A^{b+c}$ holds

  ($A^b \dashv\vdash_{\mathsf{L}} A^{b+c}$ when "well-behaved" implication missing)

  Generally, $A^b \equiv_{\mathsf{L}} A^c$ is a good notation
- One universal quantifier (over A) and two existential ones (over b and c), so several orderings possible:

	globally	locally
uniform	$\exists b, c.\forall A$	$\exists c.\forall A.\exists b$
parametric	$\exists b.\forall A.\exists c$	$\forall A.\exists b, c$

$\vdash_{\mathsf{L}} A^b \leftrightarrow A^{b+c}$

Substructural logics without lppsp

- Just consider $A(p) := p \otimes p$. This breaks down:
- Sublogics of $\mathsf{Ł}_\infty$

  (In−)$\mathsf{FL}_{(\mathsf{ew})}$, multiplicative-additive fragment of linear logic MALL (and its intuitionistic fragment IMALL), minimal fuzzy logics like BL, MTL . . .
- the product logic Π
- the bunched implication logic BI—also its boolean variant BBI, its classical linear variant CBI . . .

  In fact, one can find a refuting valuation in the famous heap model of BBI
- The sequence does stabilize for the relevance logic RM ("R with Mingle") . . . which, however, is locally finite

Modal logic: the lppsp appears to coincide with local finitess too

- $A(p) := \Box p$ kills lppsp for GL.3 (the modal logic of linear Noetherian strict orders) and its sublogics

  K, K4, GL . . .

- $A(p) := q \vee \Box(q \to \Box p)$ kills lppsp for Grz.3 (the modal logic of linear Noetherian posets) and its sublogics

  D, T, S4 or Grz . . .

  in fact, the same sequence would do for sublogics of GL.3

- We can also break down most intutionistic modal logics using one of these two sequences: see the abstract.

- S5, the modal logic of equivalence relations, has the psp . . .
- . . . thanks again to local finiteness

  sometimes also known as local tabularity
- Obviously, existence of a sequence not entering a cycle would directly contradict local finiteness
- A more detailed discussion of connections between local finiteness and various forms of the psp in a spare slide (if time)

We can transfer the above counterexamples into proofs that in many natural lattices of logic, the (lp)psp is equivalent to local finiteness. Example:

We can transfer the above counterexamples into proofs that in many natural lattices of logic, the (lp)psp is equivalent to local finiteness. Example:

Theorem

An extension of $\mathsf{K4}$ *has the (local parametric) psp iff it is locally finite (iff it is of finite depth)*

Proof.

Use the sequence discussed above and proof of Theorem 12.21 in *Modal Logic* by Chagrov & Zakharyaschev □

We can transfer the above counterexamples into proofs that in many natural lattices of logic, the (lp)psp is equivalent to local finiteness. Example:

Theorem

An extension of K4 *has the (local parametric) psp iff it is locally finite (iff it is of finite depth)*

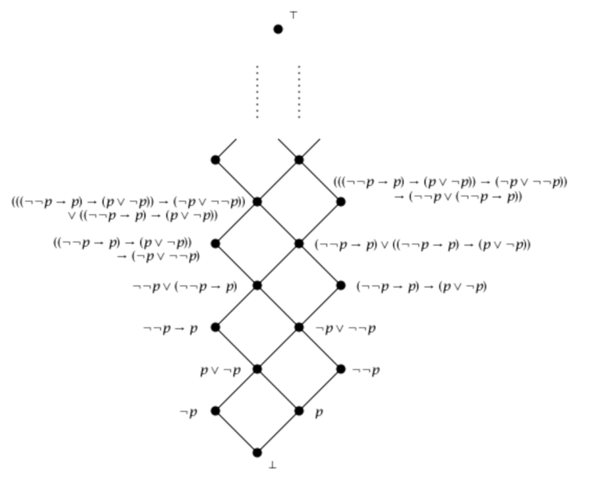
Proof.

Use the sequence discussed above and proof of Theorem 12.21 in *Modal Logic* by Chagrov & Zakharyaschev □

And yet . . .

The case of IPC

The Rieger-Nishimura lattice (wikipedia screenshot)

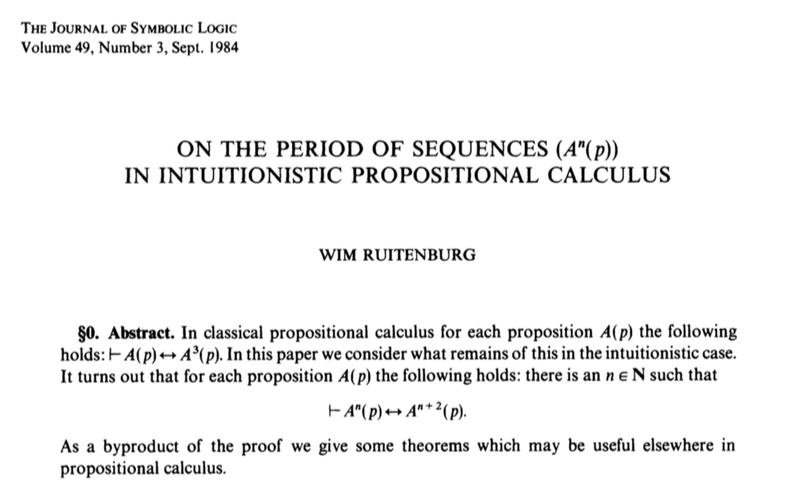

$\top$
$(((\neg\neg p \to p) \to (p \vee \neg p)) \to (\neg p \vee \neg\neg p))$
$\to (\neg\neg p \vee (\neg\neg p \to p))$
$(((\neg\neg p \to p) \to (p \vee \neg p)) \to (\neg p \vee \neg\neg p))$
$\vee ((\neg\neg p \to p) \to (p \vee \neg p))$
$((\neg\neg p \to p) \to (p \vee \neg p))$
$\to (\neg p \vee \neg\neg p)$
$(\neg\neg p \to p) \vee ((\neg\neg p \to p) \to (p \vee \neg p))$
$\neg\neg p \vee (\neg\neg p \to p)$
$(\neg\neg p \to p) \to (p \vee \neg p)$
$\neg\neg p \to p$
$\neg p \vee \neg\neg p$
$p \vee \neg p$
$\neg\neg p$
$\neg p$
p
$\bot$

Not locally finite even in one variable

- Note it's all about the interaction of $\to$ and $\vee$
- $\wedge$ is not even needed to define elements of this lattice
- the Diego-McKay theorem:

  the reduct of IPC without $\vee$ is locally finite

  (although few references seem to state clearly the bounds involved; seem doubly exponential to me) Btw the same property holds for $\vee$-free PLL (propositional lax logic, the logic of nuclei, strong monads ...
- But with $\vee$ in the full signature, can IPC have the psp??



THE JOURNAL OF SYMBOLIC LOGIC
Volume 49, Number 3, Sept. 1984

ON THE PERIOD OF SEQUENCES $(A^n(p))$ IN INTUITIONISTIC PROPOSITIONAL CALCULUS

WIM RUITENBURG

§0. Abstract. In classical propositional calculus for each proposition $A(p)$ the following holds: $\vdash A(p) \leftrightarrow A^3(p)$. In this paper we consider what remains of this in the intuitionistic case. It turns out that for each proposition $A(p)$ the following holds: there is an $n \in \mathbf{N}$ such that

$$\vdash A^n(p) \leftrightarrow A^{n+2}(p).$$

As a byproduct of the proof we give some theorems which may be useful elsewhere in propositional calculus.

- Ruitenburg proves the local uniform psp for IPC
- *Very* uniform, in fact: $\forall A. \exists b.\ \vdash_{\mathsf{IPC}} A^b \leftrightarrow A^{b+2}$

  (so the same c as for CPC does the job)
- Furthermore, b in question is linear in the size of A

  In many cases still smaller: constant when p is the only propositional variable
- This is as good as it gets
- Ruitenburg shows that quantifiers cannot be shifted

$$\forall b. \exists A. \quad \nvdash_{\mathsf{IPC}} A^b \quad \& \quad \vdash_{\mathsf{IPC}} A^{b+1} \quad \& \quad \vdash_{\mathsf{IPC}} A(\top)$$

  hence, $\forall b. \exists A$ s.t. $\nvdash_{\mathsf{IPC}} A^b \leftrightarrow A^{b+2}$

  his counterexample works even for the logic of linear orders LC

  a.k.a. the *Gödel-Dummet logic* or—in Johnstone's Elephant—the logic of *strong de Morgan law*—and this logic is locally finite

  (unlike its modal counterpart)

  So, even local finiteness does not guarantee the globally uniform psp!

- How is b obtained from A?

- How is b obtained from A?
- It is sufficient to take begin with the sequence of formulas $B_1, \ldots, B_n$ containing

- How is b obtained from A?
- It is sufficient to take begin with the sequence of formulas $B_1, \ldots, B_n$ containing
    - all atoms occurring in A

- How is b obtained from A?
- It is sufficient to take begin with the sequence of formulas $B_1, \ldots, B_n$ containing
    - all atoms occurring in A
    - all implicational subformulas of A

- How is b obtained from A?
- It is sufficient to take begin with the sequence of formulas $B_1, \dots, B_n$ containing
  - all atoms occurring in A
  - all implicational subformulas of A
- and then replace it with any sequence of formulas $C_1, \dots, C_m$ s.t.

$$\forall i \leq n . \exists j \leq m \quad \text{s.t.} \quad \vdash_{\mathsf{IPC}} B_i(\top) \leftrightarrow C_j$$

- How is b obtained from A?
- It is sufficient to take begin with the sequence of formulas $B_1, \ldots, B_n$ containing
    - all atoms occurring in A
    - all implicational subformulas of A
- and then replace it with any sequence of formulas $C_1, \ldots, C_m$ s.t.

$$\forall i \leq n . \exists j \leq m \quad \text{s.t.} \quad \vdash_{\mathsf{IPC}} B_i(\top) \leftrightarrow C_j$$

- Now set $b := 2 * m + 2$

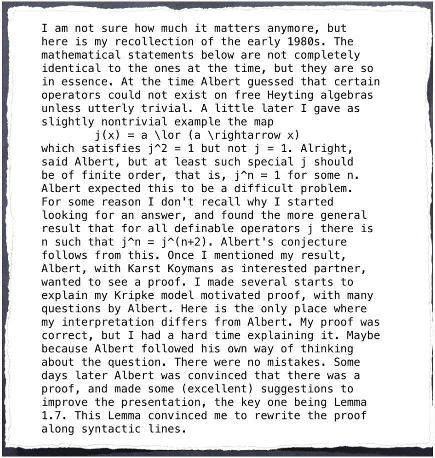
- You may still be curious whether there were additional reasons to be interested in it
- …or, to put it differently, why Albert Visser told me about this particular result
- See spare slides for the connection with definability of fixpoints
- Just one remark in connection with Anupam's talk: we're interested in provable equivalence (logic of type inhabitation)
- From this perspective, fixpoints are definable (via Pitts' uniform interpolation or Ruitenburg's result)
- If you're interested in the Curry-Howard perspective (proof terms/programs), fixpoints become a very non-trivial addition to IPC, as you've learned from Anupam

Back to the theorem itself

- The published proof is heavily syntactic
- Some light on how it become the way it is and where it stands cast by old email remarks by Albert Visser, from whom I learned about it:

> *Wim found the result as a PhD student. When he first presented it Carst Koymans and I did not believe it. We quickly found a mistake / gap in the proof. After a few days Wim came back with a repair. Again we shot at it and found a mistake / gap. This repeated itself a number of times until the proof seemed airtight. The whole thing is a singleton result. Nobody ever analysed the methods or connected it e.g. to Ghilardi's work or anything. I still think Wim did something remarkable —not only finding the proof but also asking the question— and should have gotten far more credit for it.*

- Quite different light cast by email remarks by Wim Ruitenburg, who actually proved this theorem:



I am not sure how much it matters anymore, but here is my recollection of the early 1980s. The mathematical statements below are not completely identical to the ones at the time, but they are so in essence. At the time Albert guessed that certain operators could not exist on free Heyting algebras unless utterly trivial. A little later I gave as slightly nontrivial example the map

        j(x) = a \lor (a \rightarrow x)

which satisfies j^2 = 1 but not j = 1. Alright, said Albert, but at least such special j should be of finite order, that is, j^n = 1 for some n. Albert expected this to be a difficult problem. For some reason I don't recall why I started looking for an answer, and found the more general result that for all definable operators j there is n such that j^n = j^(n+2). Albert's conjecture follows from this. Once I mentioned my result, Albert, with Karst Koymans as interested partner, wanted to see a proof. I made several starts to explain my Kripke model motivated proof, with many questions by Albert. Here is the only place where my interpretation differs from Albert. My proof was correct, but I had a hard time explaining it. Maybe because Albert followed his own way of thinking about the question. There were no mistakes. Some days later Albert was convinced that there was a proof, and made some (excellent) suggestions to improve the presentation, the key one being Lemma 1.7. This Lemma convinced me to rewrite the proof along syntactic lines.

- A good candidate for a Coq formalization
- A curious property, which does not seem to fit the pattern seen elsewhere in the large zoo of non-classical logics
- A heavily syntactic proof, which arose via several rounds of rewriting until all traces of semantics were lost
- Until quite recently:
  - Apparently understood just by a very small bunch of logicians
  - Hardly analyzed from either semantic/dual or Curry-Howard point of view

    Recall that Ghilardi and Santocanale published their work after I developed my formalization . . .

Formalization

- 3000∼4000 lines of code

  A better hacker would do better, I guess

  Developed mostly in 2015, when I was still a relative novice in Coq

  Uses features available in versions as old as 8.4pl6. To Coq's credit, it was surprisingly easy to make it work in recent versions like 8.18

- Formalizes the first part of Ruitenburg's paper (ca. 5 dense pages) which contains the actual syntactic proof of the main theorem

  The second part contains (counter-)examples, e.g, such as the one I showed you before, discussions of improvements possible in concrete cases and arguments why these cannot be generalized. It also involves Kripke frames, not just syntax

- Did not uncover any worthwhile errors
- Available at `https://git8.cs.fau.de/software/ruitenburg1984`

- Verified implementation of Ruitenburg's proof

  either directly in Coq, or via extracted (and thus certified) Haskell or Ocaml programs

- Verified implementation of Ruitenburg's proof

  either directly in Coq, or via extracted (and thus certified) Haskell or Ocaml programs
- Exponential blowup can be painful

- Verified implementation of Ruitenburg's proof

  either directly in Coq, or via extracted (and thus certified) Haskell or Ocaml programs
- Exponential blowup can be painful
- The fact that b itself is linear in the size of A does not mean that the size of A^b is linear in the size of A!

- Verified implementation of Ruitenburg's proof

  either directly in Coq, or via extracted (and thus certified) Haskell or Ocaml programs

- Exponential blowup can be painful
- The fact that b itself is linear in the size of A does not mean that the size of A^b is linear in the size of A!
- Consider $A(p) := (q \to (p \to r)) \wedge ((p \to r) \to (p \vee r))$.

- Verified implementation of Ruitenburg's proof

  either directly in Coq, or via extracted (and thus certified) Haskell or Ocaml programs

- Exponential blowup can be painful
- The fact that b itself is linear in the size of A does not mean that the size of A^b is linear in the size of A!
- Consider $A(p) := (q \to (p \to r)) \wedge ((p \to r) \to (p \vee r))$.
- Bound $b = 10$ would work. Now think what A^{10} is ...

  To the credit of Coq's core functional language (Gallina), after several minutes it actually computed the output as a mere 3.2 MB text file. Extracted Haskell hit stack overflow, at least back in 2015–17

Still more about the structure

- The first file `HilbertIPCsetup`, somewhat over 1000 lines, contains the basic metatheory of (somewhat idiosyncratic somewhat Gentzen-style formulation) of a Hilbert-style system for IPC.

  This was entirely implicit in Ruitenburg's paper
- Just the very last part is a proof of an actual lemma from the paper (Lemma 1.2)
- The second one `Ruitenburg1984Aux` contains Lemmas 1.6–1.8
- All these lemmas would work in any modal logic over IPC…where local and global consequence coincide
- Modulo the spadework done in opening file, they were actually nice and easy. Coq proofs follow closely the structure

- It is the proof of the key Theorem 1.4 that require most work: two files `BoundsSubformulas` and `Ruitenburg1984Key`
- `BoundsLists` contains the apparatus necessary for actual computation and program extraction and `Ruitenburg1984Main` just puts it all together
- Some Coq details: the actual proof is done in terms of `Ensembles`, meaning in fact working with `Prop`
- The extractable code formulated in terms of `Lists`

  The connection between the two setups, while made precise in the formalization, does not even mention reflection ...

How to overhaul it

- Today, Férée and van Gool (2023) or Shillito and coauthors (2021–24) have closely related formalizations of G4ip-style terminating sequent calculi for various extensions of IPC

  Recently, coming together in the mechanized proof of syntactic strong interpolation for strong Löb: Iris can tell you more

- Ultimately, this whole formalization should be recast in such a setting

  Also improving on Ruitenburg's framework, in fact

  Tbqh, at some point I needed decidability/completeness of IPC and I didn't even bother to formalize it

  The nice thing about Coq is that it could be bypassed easily: simply by sneakily importing meta-level excluded middle in the corresponding file/submodule

Future work

- Attack the Lax logic PLL, look for good examples among (idempotent?) relevance logics

  Cf. *Diego's Theorem for nuclear implicative semilattices* by Bezhanishvilis et al, 2021. Note it's not the PLL you've heard about in Gianluca's talk this morning!

Future work

- Attack the Lax logic PLL, look for good examples among (idempotent?) relevance logics

  Cf. *Diego's Theorem for nuclear implicative semilattices* by Bezhanishvilis et al, 2021. Note it's not the PLL you've heard about in Gianluca's talk this morning!

- Are all four forms of the psp encountered independently in nature? Finite $\Rightarrow$ global psp? Locally finite $\Rightarrow$ **u**psp?

Future work

- Attack the Lax logic PLL, look for good examples among (idempotent?) relevance logics

  Cf. *Diego's Theorem for nuclear implicative semilattices* by Bezhanishvilis et al, 2021. Note it's not the PLL you've heard about in Gianluca's talk this morning!

- Are all four forms of the psp encountered independently in nature? Finite ⇒ global psp? Locally finite ⇒ **u**psp?
- Replace Hilbert-style with Gentzen-style, extend to syntactic cut elimination, compare with the Pitts-uniform-interpolation route

  And do all the other things I mentioned on the other slide

Future work

- Attack the Lax logic PLL, look for good examples among (idempotent?) relevance logics

  Cf. *Diego's Theorem for nuclear implicative semilattices* by Bezhanishvilis et al, 2021. Note it's not the PLL you've heard about in Gianluca's talk this morning!
- Are all four forms of the psp encountered independently in nature? Finite ⇒ global psp? Locally finite ⇒ **u**psp?
- Replace Hilbert-style with Gentzen-style, extend to syntactic cut elimination, compare with the Pitts-uniform-interpolation route

  And do all the other things I mentioned on the other slide
- A TACL'17 question: is psp related to "good" properties of 1-generated free algebras?

Spare slides

- Local finiteness guarantees at least locally parametric psp
- As the example of CPC shows, for concrete finite logics we may get the globally uniform psp: and with *very* tight bounds at that

  (in such cases, the size of resulting A^b is also pleasant:

  something we will say much more about later on)
- I don't have an argument that the globally uniform psp would obtain in general for finite implicative logics

  We'll see further it does not obtain in general for locally finite ones
- In fact, I don't even know if globally parametric psp obtains in general for locally finite implicative ones, or even for finite implicative ones

Connection with definability of fixpoints

- It is easy to define what it means for A to have a definable L-fixpoint B:

$$p\#B \text{ and } \vdash_{\mathsf{L}} A(B) \leftrightarrow B$$

- It is also easy to define L-least fixpoints: the satisfy in addition

$$\text{for some/any fresh } q,\ A(q) \to q \vdash_{\mathsf{L}} B \to q$$

- Dually, L-greatest fixpoints satisfy in addition

$$\text{for some/any fresh } q,\ q \to A(q) \vdash_{\mathsf{L}} q \to B$$

- If a fixpoint is both least and greatest, it is called unique

- It is easy to realize that the psp + presence of $\bot$ implies definable least L-fixpoints of monotone formulas
- It is easy to realize that the psp + presence of $\top$ implies definable greatest L-fixpoints of monotone formulas
- In other words, extending such logics with fixpoint operators does not improve expressivity!

  Well, one has to be a bit careful that B in question *preserves monotonicity/positivity*: Mardaev and other references discuss this
- But as it turns out, even when the logic itself does not have the psp, it may be useful to have a propositional reduct which has this property to establish fixpoint results

- Consider intuitionistic or classical logics with the Löb axiom $\Box(\Box p \to p) \to \Box p$

  in stronger, but classically useless variant $(\Box p \to p) \to p$

- In the classical setup, these logics arise as modal logics of well-founded transitive structures

  think of finite trees, for example

- In the intuitionistic setup, these logics arise as the Curry-Howard counterparts of calculi with guarded (co-)recursion

- We’ve seen that these logics do not have the psp
- But they do have unique fixpoints of guarded or modalized formulas

  i.e., those where every occurrence of p is within the scope of $\Box$

  curiously enough, in numerous calculi for guarded (co-)recursion—from Nakano LiCS 2000 to Clouston, Birkedal et al. FoSSaCS 2015—people found it useful to add explicit fixpoint operators for such formulas
- van Benthem around 2005: a semantic proof that this result can be used to prove definability of ordinary fixpoints in GL

  Seems that it was independently and even earlier found by Mardaev
- Visser around the same time: a syntactic proof relying on the psp of CPC...
- In fact, we can even define not-quite-least fixpoints of mixed formulas which contain both positive and guarded occurrences of p ...

and finally, one more puzzling connection

- There is a well-known trick of defining fixpoints (of monotone formulas) using propositional quantifiers in, say, system F:
$$\mu p.A = \forall p.(A \to p) \to p.$$
See, e.g., Wadler's *Recursive types for free!* manuscript for a discussion of this in connection with parametricity
- We know the name for definability of (a certain kind of) propositional quantifiers in L ...
- ... it's uniform interpolation, of course!
- How does it relate to definability of fixpoints? And, for that matter, to the psp?

- For an arbitrary L, uniform interpolation does not imply definability of fixpoints of monotone formulas
- Consider K and $\mu p.\Box p$ ...
- What one needs in addition is a form of global deduction theorem ...
- ... which in the modal context, boils down to definability of master modality

  ... and for AAL, it boils down to the EDPC—equationally definable principal congruences
- A trick used by d'Agostino and Lenzi in TCS 2005 when showing that PDL with bisimulation quantifiers is equivalent to μ-calculus

- IPC, though, has trivially such a global deduction theorem, as long as we do not add modalities etc. to it
- So here lies another route for fixpoints
- See Ghilardi et al. for a more detailed comparison
- Funnily enough, even though Ruitenburg's paper was written many years before Pitts, it does hint at a connection with uniform interpolation at the very end!