Coherence via Well-Foundedness

– Taming Set-Quotients in Homotopy Type Theory

HoTTEST Fall 2020

Jakob von Raumer[1], j.w.w. Nicolai Kraus[1,2] | September 25, 2020

[1]University of Nottingham, [2]University of Birmingham

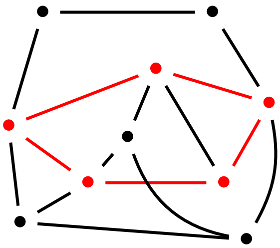
Outline of the Talk

A Graph Theoretic Problem

Noetherian Cycle Induction

Application to Coherence

General Problem

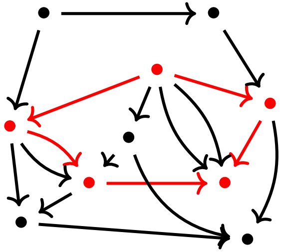


Consider paths in a graph.

If we want to prove a property. . .

- *for all paths:* **Induction!**
- *for all closed paths:* **how???**

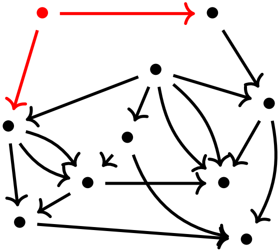
Graph Theoretic Formulation



Problem: Prove a property for every *closed zig-zag* (from now on *cycle*) in a graph.

Assumptions: The graph is

Graph Theoretic Formulation

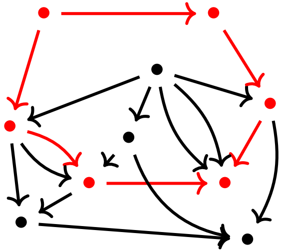


Problem: Prove a property for every *closed zig-zag* (from now on *cycle*) in a graph.

Assumptions: The graph is

- locally confluent, and

Graph Theoretic Formulation

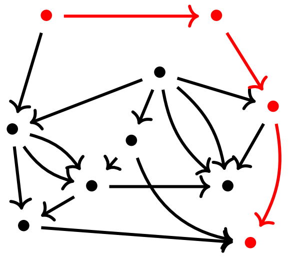


Problem: Prove a property for every *closed zig-zag* (from now on *cycle*) in a graph.

Assumptions: The graph is

- locally confluent, and

Graph Theoretic Formulation



Problem: Prove a property for every *closed zig-zag* (from now on *cycle*) in a graph.

Assumptions: The graph is

- locally confluent, and
- Noetherian (co-wellfounded).

Example: Reductions in Free Groups

Reduction steps on words in a free group on a set M form such a graph on $\mathrm{List}(M + M)$.

$$[m_0^{-1}, m_1, m_1^{-1}, m_2^{-1}, m_3^{-1}, m_3]$$

$$[m_0^{-1}, m_2^{-1}, m_3^{-1}, m_3] \qquad [m_0^{-1}, m_1, m_1^{-1}, m_2^{-1}]$$

$$[m_0^{-1}, m_2^{-1}]$$

Graph Theoretic Formulation

Our proposed solution consists of the following four steps:

1. Given a relation $\rightsquigarrow$ on a set A, we define a new relation $\rightsquigarrow^\circ$ on cycles on A.
2. If $\rightsquigarrow$ is Noetherian, then so is $\rightsquigarrow^\circ$.
3. If $\rightsquigarrow$ further is locally confluent, then any cycle can be split into a $\rightsquigarrow^\circ$-smaller cycle and a confluence cycle
4. Consequence: We can show a property *for all cycles* inductively by showing it *for empty cycles, confluence cycles, and merged cycles*.

Step 1: List Extension

Definition

The *list extension* of a relation $\rightsquigarrow$ on A is a relation $\rightsquigarrow^L$ on $\mathrm{List}(A)$ generated by

$$[\vec{a_1}, a, \vec{a_2}] \rightsquigarrow^L [\vec{a_1}, x_0, x_1, \ldots, x_k, \vec{a_2}]$$

where all x_i are such that $a \rightsquigarrow x_i$.

Step 1: List Extension

Definition

The *list extension* of a relation $\rightsquigarrow$ on A is a relation $\rightsquigarrow^L$ on List(A) generated by

$$[\vec{a}_1, a, \vec{a}_2] \rightsquigarrow^L [\vec{a}_1, x_0, x_1, \ldots, x_k, \vec{a}_2]$$

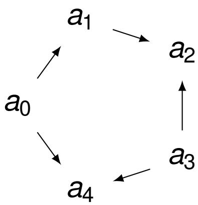
where all x_i are such that $a \rightsquigarrow x_i$.

Lemma

If $\rightsquigarrow$ is Noetherian, so is $\rightsquigarrow^L$.

This is similar to the well-founded *multiset extension* by Tobias Nipkow.

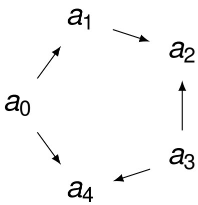
Step 2: A Relation on Cycles



Definition

For γ a cycle, write $\varphi(\gamma)$ for the *vertex sequence* of γ.
Write $\gamma \rightsquigarrow^{\circ} \delta$ if there is a rotation δ' of δ such that $\varphi(\gamma) \rightsquigarrow^{L} \varphi(\delta')$.

Step 2: A Relation on Cycles



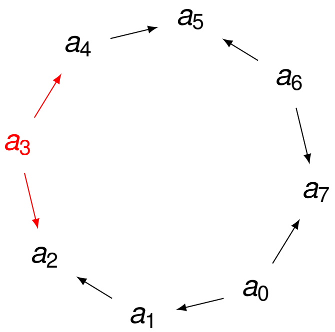
Definition

For γ a cycle, write $\varphi(\gamma)$ for the *vertex sequence* of γ.
Write $\gamma \rightsquigarrow^{\circ} \delta$ if there is a rotation δ' of δ such that $\varphi(\gamma) \rightsquigarrow^{L} \varphi(\delta')$.

Lemma

If $\rightsquigarrow$ is Noetherian, so is $\rightsquigarrow^{\circ}$ (and thus also $\rightsquigarrow^{+\circ+}$).

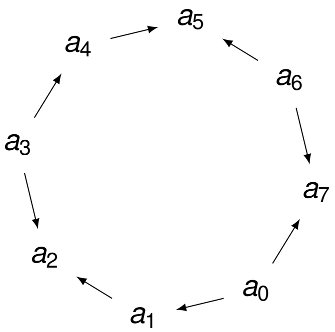
Step 3: Dissecting Cycles



Lemma

If a relation is Noetherian, then any of its cycles is empty or contains a span.

Step 3: Dissecting Cycles



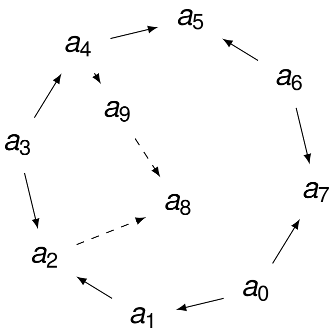
Lemma

If a relation is Noetherian, then any of its cycles is empty or contains a span.

Theorem

If $\rightsquigarrow$ is Noetherian and locally confluent, then any cycle can be written as the "merge" of a $\rightsquigarrow^{+\circ+}$-smaller cycle and a confluence diamond.

Step 3: Dissecting Cycles



Lemma

If a relation is Noetherian, then any of its cycles is empty or contains a span.

Theorem

If $\rightsquigarrow$ is Noetherian and locally confluent, then any cycle can be written as the "merge" of a $\rightsquigarrow^{+\circ+}$-smaller cycle and a confluence diamond.

Step 4: The Induction Principle

Theorem (Noetherian Cycle Induction)

Given a Noetherian and locally confluent relation $\rightsquigarrow$ on a set A and a property P on its cycles, such that

Step 4: The Induction Principle

Theorem (Noetherian Cycle Induction)

Given a Noetherian and locally confluent relation $\rightsquigarrow$ on a set A and a property P on its cycles, such that

- *P is stable under rotating of cycles: P$(\alpha\gamma) \to$ P$(\gamma\alpha)$,*

Step 4: The Induction Principle

Theorem (Noetherian Cycle Induction)

Given a Noetherian and locally confluent relation $\rightsquigarrow$ on a set A and a property P on its cycles, such that

- *P is stable under rotating of cycles: $P(\alpha\gamma) \to P(\gamma\alpha)$,*
- *P is stable under "merging" of cycles: $P(\alpha\gamma) \to P(\gamma^{-1}\beta) \to P(\alpha\beta)$,*

Step 4: The Induction Principle

Theorem (Noetherian Cycle Induction)

Given a Noetherian and locally confluent relation $\rightsquigarrow$ on a set A and a property P on its cycles, such that

- *P is stable under rotating of cycles:* $P(\alpha\gamma) \to P(\gamma\alpha)$,
- *P is stable under "merging" of cycles:* $P(\alpha\gamma) \to P(\gamma^{-1}\beta) \to P(\alpha\beta)$,
- *P holds for the empty cycle, and*

Step 4: The Induction Principle

Theorem (Noetherian Cycle Induction)

Given a Noetherian and locally confluent relation $\rightsquigarrow$ on a set A and a property P on its cycles, such that

- *P is stable under rotating of cycles: $P(\alpha\gamma) \to P(\gamma\alpha)$,*
- *P is stable under "merging" of cycles: $P(\alpha\gamma) \to P(\gamma^{-1}\beta) \to P(\alpha\beta)$,*
- *P holds for the empty cycle, and*
- *P holds for confluence diamonds.*

Step 4: The Induction Principle

Theorem (Noetherian Cycle Induction)

Given a Noetherian and locally confluent relation $\rightsquigarrow$ on a set A and a property P on its cycles, such that

- *P is stable under rotating of cycles:* $P(\alpha\gamma) \to P(\gamma\alpha)$,
- *P is stable under "merging" of cycles:* $P(\alpha\gamma) \to P(\gamma^{-1}\beta) \to P(\alpha\beta)$,
- *P holds for the empty cycle, and*
- *P holds for confluence diamonds.*

Then, $P(\gamma)$ ***holds for any cycle*** γ.

Step 4 with Type Theory Flavour

Theorem (Noetherian Cycle Induction)

Given a type A : Type and a Noetherian and locally confluent relation $\rightsquigarrow : A \to A \to$ Type.

Let P : (cycles of $\rightsquigarrow$) $\to$ Type be such that

- *P is stable under rotating of cycles: $P(\alpha\gamma) \to P(\gamma\alpha)$,*
- *P is stable under "merging" of cycles: $P(\alpha\gamma) \to P(\gamma^{-1}\beta) \to P(\alpha\beta)$,*
- *P holds for the empty cycle, and*
- *P holds for confluence diamonds.*

***Then,** $P(\gamma)$ **holds for any cycle** γ.*

Maps into a 1-Type

Theorem

Let A be a type, $\rightsquigarrow : A \to A \to \mathrm{Type}$ be Noetherian and locally confluent (with confluence "diamonds" $\mathfrak{L}$), and X be a 1-type.
Then, the type $A/\rightsquigarrow\; \to X$ is equivalent to the type of tuples (f, h, d_1, d_2), where

$$
\begin{aligned}
f &: A \to X, \\
h &: \Pi\{a, b : A\}.(a \rightsquigarrow b) \to f(a) = f(b), \\
d_1 &: \Pi\{a : A\}.\Pi(p : a = a).\mathrm{ap}_f(p) = \mathrm{refl}, \\
d_2 &: \Pi(\kappa : \cdot \leftsquigarrow \cdot \rightsquigarrow \cdot).h(\mathfrak{L}(\kappa)) = \mathrm{refl}.
\end{aligned}
$$

Maps into a 1-Type

Proof of the Theorem:

1. Show that the statement is true if instead of d_2, we index over *all* cycles.
2. Apply Noetherian Cycle Induction with $P(\gamma) :\equiv (h(\gamma) = \mathsf{refl})$.

Two Ways to Define Free Groups

How to define the carrier of the free group on a set M?

1. As a (set-)quotient of words $\mathrm{List}(M+M)/\rightsquigarrow$ where the $\rightsquigarrow$ is generated by

$$[\ldots, k, m, m^{-1}, l, \ldots] \rightsquigarrow [\ldots, k, l, \ldots].$$

Two Ways to Define Free Groups

How to define the carrier of the free group on a set M?

1. As a (set-)quotient of words $\mathrm{List}(M + M)/\rightsquigarrow$ where the $\rightsquigarrow$ is generated by

$$[\ldots, k, m, m^{-1}, l, \ldots] \rightsquigarrow [\ldots, k, l, \ldots].$$

2. As the loop space $F_M :\equiv \Omega(H_M, \star)$ of the higher inductive type

$$\begin{aligned} &\text{data } H_M : \text{Type where} \\ &\quad \star : H_M \\ &\quad \text{loops} : M \to (\star = \star) \end{aligned}$$

Two Ways to Define Free Groups

How to define the carrier of the free group on a set M?

1. As a (set-)quotient of words $\mathrm{List}(M + M)/\!\rightsquigarrow$ where the $\rightsquigarrow$ is generated by

$$[\ldots, k, m, m^{-1}, l, \ldots] \rightsquigarrow [\ldots, k, l, \ldots].$$

2. As the loop space $F_M :\equiv \Omega(H_M, \star)$ of the higher inductive type

$$\begin{aligned} &\text{data } H_M : \text{Type where} \\ &\quad \star : H_M \\ &\quad \text{loops} : M \to (\star = \star) \end{aligned}$$

Open question: Do these coincide?

Two Ways to Define Free Groups

How to define the carrier of the free group on a set M?

1. As a (set-)quotient of words $\mathrm{List}(M + M)/\rightsquigarrow$ where the $\rightsquigarrow$ is generated by

$$[\dots, k, m, m^{-1}, l, \dots] \rightsquigarrow [\dots, k, l, \dots].$$

2. As the loop space $F_M :\equiv \Omega(H_M, \star)$ of the higher inductive type

$$\begin{aligned} &\text{data } H_M : \text{Type where} \\ &\quad \star : H_M \\ &\quad \text{loops} : M \to (\star = \star) \end{aligned}$$

Approximation: Do their 1-truncations coincide?
Or: Is the fundamental group of the free group trivial?

Map Between the Definitions

- There is a canonical map $F_M \to \text{List}(M+M)/\rightsquigarrow$ factoring through $\|F_M\|_1$.
- Need to construct an inverse map $\text{List}(M+M)/\rightsquigarrow \to \|F_M\|_1$.
- By the previous theorem, we need to give $f : \text{List}(M+M) \to \|F_M\|_1$, $h : \Pi\{k, l : \text{List}(M+M)\}.(k \rightsquigarrow l) \to f(k) = f(l)$, and show that h is refl on confluence diamonds.

Map Between the Definitions

How to show that h is refl on confluence diamonds:

$$f[m_0^{-1}, m_1, m_1^{-1}, m_2^{-1}, m_3^{-1}, m_3]$$

$$f[m_0^{-1}, m_2^{-1}, m_3^{-1}, m_3] \qquad\qquad f[m_0^{-1}, m_1, m_1^{-1}, m_2^{-1}]$$

$$f[m_0^{-1}, m_2^{-1}]$$

Map Between the Definitions

How to show that *h* is refl on confluence diamonds:

$$f[m_0^{-1}] \cdot f[m_1] \cdot f[m_1^{-1}] \cdot f[m_2^{-1}] \cdot f[m_3^{-1}] \cdot f[m_3]$$

$$f[m_0^{-1}] \cdot f[m_2^{-1}] \cdot f[m_3^{-1}] \cdot f[m_3] \qquad\qquad f[m_0^{-1}] \cdot f[m_1] \cdot f[m_1^{-1}] \cdot f[m_2^{-1}]$$

$$f[m_0^{-1}] \cdot f[m_2^{-1}]$$

Map Between the Definitions

How to show that h is refl on confluence diamonds:

$$f[m_1] \cdot f[m_1^{-1}] \cdot f[m_3^{-1}] \cdot f[m_3]$$

$$f[m_3^{-1}] \cdot f[m_3] \qquad\qquad f[m_1] \cdot f[m_1^{-1}]$$

$$\mathsf{refl}$$

A List of Open Problems

1. Is the free group HIT on a set again a set?
2. Is the suspension of a set a 1-type?
3. Does adding a loop to a type preserve it being 1-truncated?
4. Does adding set many loops to a type preserve it being 1-truncated?

Common generalisation:

- Is the pushout $B +_A C$ a 1-type, if A is a set and B, C are 1-types?

A List of Open Problems

Approximate the generalisation by showing that $\|B +_A C\|_2$ is a 1-type:

- Consider the encoding of equalities in pushouts (Seifert-van Kampen) à la Favonia and Shulman.
- The lists of equalities generalise the type List$(M + M)$ in the free group example.
- Likewise apply Noetherian Cycle Induction.

Potential Application: Type Theory in Type Theory

- Want to internalise the syntax of type theory inside HoTT (à la Altenkirch, Kaposi).
- For many purposes treat convertability relations as equalities
- Take a quotient by a reduction relation!
- Standard model: Construct function from contexts to the universe of sets
- Open question: How to generalise the theorem such that it can deal with QIITs?

Conclusions

- We found a way to tackle proofs about cycles
- We used it to solve approximations to open problems
- The contents formalised in the Lean theorem prover ($\sim$ 1600 LoC)
- We are exploring applicability
  - to other open problems in HoTT
  - to the field of higher-dimensional rewriting (Thanks to Vincent van Oostrom for his remarks!)