Practical for Advanced Topics in Category Theory

https://beta.homotopy.io

1 Introduction

Here is a long list of proofs that are suitable for formalizing in *homotopy.io*. You are free to choose from the ideas given here, or to come up with your own ideas of what to formalize (it would make sense to discuss your ideas here with the lecturer first.) Stars give a rough guide to difficulty:

$*$ Easy $**$ Medium $***$ Hard $****$ Insane

You will work on these formalizations during term, building up a portfolio of work that will be submitted electronically at the end of term. At most 5 files should be submitted, corresponding to 5 distinct problems. There will be practical classes after the some of the lectures, which will give ideal opportunities to learn about the tool and work on your portfolio. However, you are also expected to work on your portfolio in your own time.

The ideas are drawn from the lecture notes, and also more broadly from knot theory, control theory, pure mathematics, and computer science. The more difficult ideas should be seen more as open-ended projects than as specific, achievable goals for formalization; good credit could certainly be attained for partial solutions. So don't be scared to have a go! Some of the more difficult ideas are a bit under-specified, please ask the teaching assistants or lecturer if you want some help to understand them.

To achieve a pass mark, there is no requirement to treat problems of any particular difficulty, and you should do what you feel comfortable with. For the highest marks, portfolios should show evidence of serious engagement with some of the harder or more open-ended problems. Additional credit will also be given if you develop your own problem ideas.

2 Results from the lecture notes

1. $(*)$ Example 1.9. A simple example of planar isotopy.

2. $(*)$ Exercise 1.4.3. Comparing diagrams in monoidal, braided monoidal and symmetric monoidal categories.

3. $(*)$ Exercise 1.4.4 (a), (b). Simple computations in a braided monoidal category. (Part (c) does not make sense in *homotopy.io*, since the proof assistant does not allow you to represent the monoidal unit explicitly.)

4. $(*)$ Lemma 3.4. In a monoidal category with $L \dashv R$, if $L \dashv R$, then $L \dashv R'$ if and only if $R \simeq R'$. Similarly, if $L \dashv R$, then $L' \dashv R$ if and only if $L \simeq L'$.

5. $(*)$ Lemma 3.5. In a monoidal category, if $(L, R, \eta, \varepsilon)$ and $(L, R, \eta, \varepsilon')$ both exhibit a duality, then $\varepsilon = \varepsilon'$. Similarly, if $(L, R, \eta, \varepsilon)$ and $(L, R, \eta', \varepsilon)$ both exhibit a duality, then $\eta = \eta'$.

6. $(*)$ Lemma 3.7. In a monoidal category, $L \dashv R$ and $L' \dashv R'$ implies $L \otimes L' \dashv R' \otimes R$; also, $I \dashv I$.

7. $(*)$ Lemma 3.8. In a braided monoidal category, $L \dashv R \;\Rightarrow\; R \dashv L$.

8. $(*)$ Lemma 3.12. In a monoidal category with chosen dualities $A \dashv A^*$ and $B \dashv B^*$, some equations hold for all morphisms $A \xrightarrow{f} B$.

9. $(*)$ Proposition 3.11. The duality operation $(-)^*$ satisfies $(f \circ g)^* = g^* \circ f^*$.

10. $(**)$ Lemma 4.8. Product comonoid.

11. $(*)$ Lemma 4.11. Pants monoid.

12. $(**)$ Lemma 4.25. Nested cup lemma.

13. $(**)$ Theorem 4.27. Uniform copying implies every endomorphism is a scalar multiple of the identity.

14. $(**)$ Lemma 5.4. Identities for Frobenius structures.

15. $(**)$ Proposition 5.16. Frobenius structures via nondegenerate form.

16. $(*)$ Lemma 5.19. Homomorphisms of Frobenius structures are invertible.

17. $(**)$ Lemma 6.6. The pants and twisted pants are complementary.

18. $(**)$ Lemma 6.6. Complementarity via a unitary.

19. $(**)$ Proposition 6.14. Oracles via complementarity.

20. $(*)$ Theorem 6.23. Frobenius bialgebras are trivial.

21. $(**)$ Theorem 6.35. Swap via three CNOTs.

22. (∗∗) Theorem 8.12. In a 2-category, every equivalence gives rise to a dual equivalence.

23. (∗∗∗) Theorem 8.21. In a monoidal 2-category, if objects L, R, R' satisfy coherent dualities $L \dashv R$ and $L \dashv R'$, then R and R' are canonically equipped with a dual equivalence.

24. (∗∗∗) Theorem 8.22. In a monoidal 2-category, every dual pair of objects gives rise to a coherent dual pair.

25. (∗∗) Theorem 8.25. In a monoidal 2-category, the Frobenius structure induced by an oriented duality is commutative.

    See existing formalization http://globular.science/1601.005

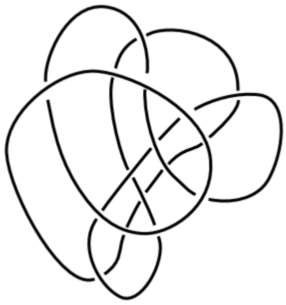
26. (∗∗) Theorem 8.63. Dense coding and teleportation are equivalent in a monoidal bicategory.

27. (∗∗) Theorem 8.65. Complementarity in a monoidal bicategory gives rise to complementarity in a monoidal category.

3 Knot theory

28. (∗∗) Show that this knot is isotopic to the unknot.

    http://homepages.math.uic.edu/ kauffman/IntellUnKnot.pdf, Figure 4

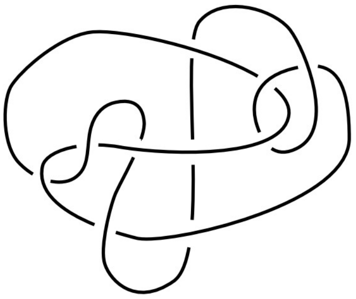


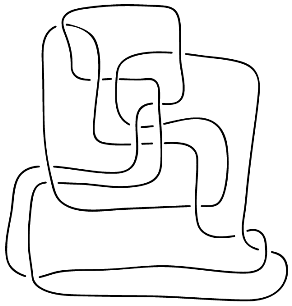
29. (∗∗∗, ∗∗∗∗) Show that these knots and links are unknotted.

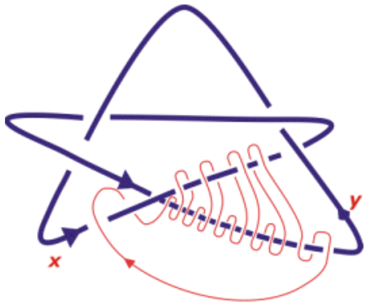
    First image: http://homepages.math.uic.edu/~kauffman/IntellUnKnot.pdf

    Second image: Thistlethwaite's Unknot (Wikipedia image)

    Third image: http://groupoids.org.uk/outofline/motion.html



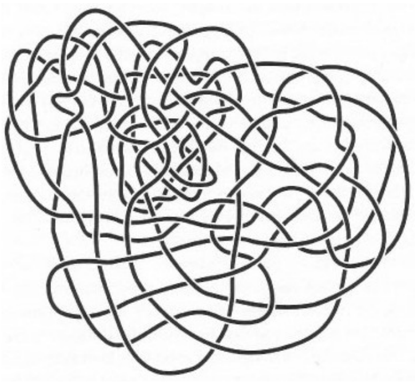


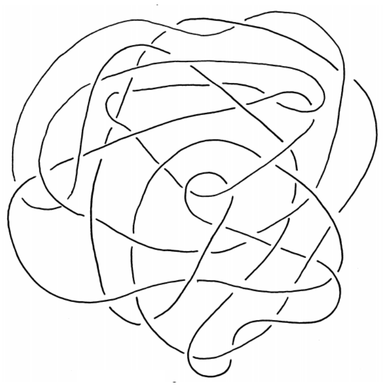


30. (∗∗∗∗) Show that these knots are isotopic to the unknot.

First image: Haken's Gordian knot
https://mickburton.co.uk/2015/06/05/how-do-you-construct-hakens-gordian-knot/

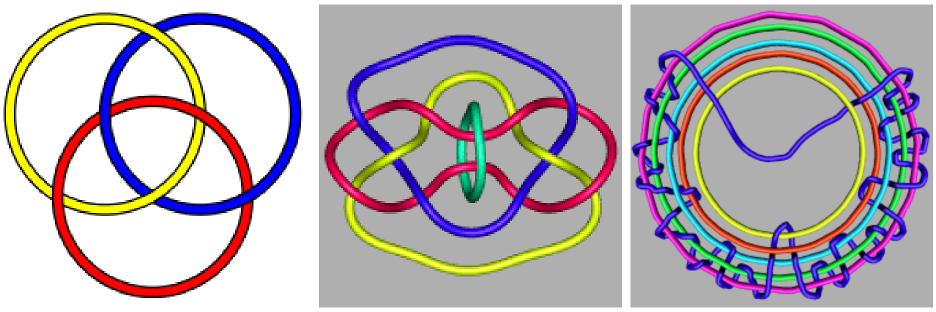
Second image: Ochiai's Second Unknot
https://www.youtube.com/watch?v=HJuPMMJHOlg
http://arxiv.org/pdf/1110.2871v1.pdf
http://repository.kulib.kyoto-u.ac.jp/dspace/bitstream/2433/99940/1/0624-1.pdf





31. (∗∗, ∗∗∗, ∗∗∗∗) Investigate *Brunnian links*, nontrivial links which become trivial when any one strand is removed. (Add a 'fire' scalar, with higher cells that allows this to 'burn' through strands.)

(Pictures from Google Images; there are lots more that are easy to find.)



32. (∗∗∗∗) Turn the torus inside out (www.youtube.com/watch?v=kQcy5DvpvlM). See Chapter 8 of the notes for the graphical calculus for oriented surfaces.

33. (∗∗∗) Verify the relations of the *loop braid group*, governing the interactions of topological loops in $\mathbb{R}^4$.

(See pages 19-21 of http://www1.maths.leeds.ac.uk/ matzk/Modelling%20TPM%20Resources_files/Leedshandout.pdf)

4 Engineering

34. (∗∗∗∗) String diagrams give a sound and complete semantics for *control systems* (Fong, Rapisarda and Sobocinski, http://arxiv.org/abs/1510.05076). Implement the rules in *Globular*, build some interesting control systems, and prove their correctness.

5 Pure mathematics

35. (∗) Define a monoid, and then add the pentagon equation as a 4-cell.

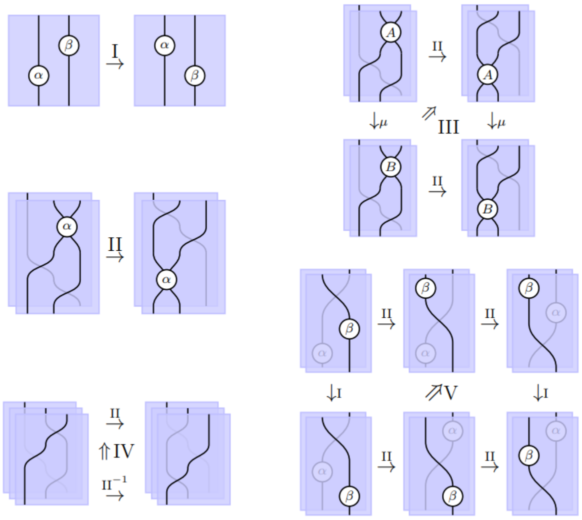
36. (∗∗∗) Define a monoid equipped with the pentagon as a 4-cell, then add the *associahedron* as a 5-cell. See pages 9 and 10 of http://arxiv.org/abs/1301.1053.

37. (∗∗∗) Formalize some of the results from Simon Willerton's paper "A diagrammatic approach to Hopf monads" (http://arxiv.org/abs/0807.0658).

38. (∗∗∗∗) Formalize some results from "Distributive laws for pseudomonads" by Marmolejo (www.tac.mta.ca/tac/volumes/1999/n5/5-05abs.html).

39. (∗∗, ∗∗∗) Formalize some results from "Braided Hopf algebras" by Shouchuan Zhang (http://arxiv.org/abs/math/0511251).

40. (∗, ∗∗, ∗∗∗). Formalize these homotopies I, II, III, IV and V (from "Data structures for quasistrict 4-categories", http://arxiv.org/abs/1610.06908):



6 Computer science

41. (∗∗) A *Petri net* gives the generating data for a symmetric monoidal category, with reachability of one configuration from another corresponding to the existence of a morphism with a particular type. Find examples of Petri nets in the literature, or build some of your own, and analyze them with *homotopy.io* by building traces that exhibit specific behaviours.

    There is an enormous amount of material online that is easy to find. This web page is a good place to start:

    https://www.informatik.uni-hamburg.de/TGI/PetriNets/introductions/aalst/

42. (∗∗∗) Build a universal Turing machine, with read/write head and transition rules in the signature, and the tape and program in the diagram.