1 Contact

Office hours: by appointment
Email: tgraf@ucla.edu
Homepage: http://tgraf.bol.ucla.edu/
Please prefix the subject with MathLing1. Replies usually take 24h-48h, depending on current workload and the length of my answer.

2 How to Approach Math

- **Don't be afraid of funny symbols**. Mathematical formulas sometimes seem dazzlingly complicated, but in fact they are pretty easy to decipher. Just don't let all the Greek letters, subscripts, superscripts and indices intimidate you.

- **Definitions are important**. Don't assume that an intuitive understanding of the concepts will be enough. It might work in syntax or phonology, but if you're doing math you need to keep an eye on the details. That isn't to say you should learn the definitions by rote. Rather, you should go over them carefully and consider special cases that might arise. A short example: When mathematicians speak of some subset A of some set B, keep in mind that A might be empty (sometimes yielding weird results).

- **When reading a definition, always think of things it doesn't cover.** That way you will get a better grasp of what kind of object the definition actually talks about.

- **Avoid common mistakes.** In mathematics, many inferences that are mandatory in ordinary communication are actually invalid. Keep the following things in mind:

  - There are no pragmatic implicatures in math. Consider the phrase "a is less than b". It is true when $a = b$, while "a is strictly less than b" is false in this case. Similarly, the condition "there is a node labeled a with all its children labeled b" is true if the node has no children (but not if there is no a-labeled node at all).
  - Don't confuse implication (*if*, $\rightarrow$) and equivalence (*iff*, $\longleftrightarrow$). If it rains tomorrow, the streets will be wet, but the converse doesn't hold in the general case. On the other hand, some animal is a groundhog if it is a woodchuck, *et vice versa*. Thus some animal is a groundhog iff it is a woodchuck.

- **Don't skip proofs.** Even though proofs play only a minor role in this course (compared to standard math classes, that is), you should read them carefully and try to figure out how they work. Most of the time, efficient proofing is just efficient bookkeeping. Always keep the following questions in mind when reading a proof.

- What do I need to show? If, for instance, you need to show that some relation R over a set A is a weak partial order, just check the definition for weak partial orders and you will see that you need to show that R is reflexive, transitive and antisymmetric.
- What do I already know? If the proof concerns only some restricted structure, recall all those restrictions. Also keep in mind that most mathematical proofs make us of results obtained earlier, so check those for their usability in the current proof.
- Are there any special cases to consider? Unfortunately, minor details have a tendency to turn a straight-forward proof into a longwinded *tour de force* (but not in this course, so don't worry).

3 Proof Techniques

- **Direct proof**

  The simplest proof technique, though often the most difficult one. Simply look at the theorem, determine what needs to be shown, and then find a way of showing it.

  Theorem 3.1. *If integer n is even, than n^2 is even.*

  Proof. Implicational statements are vacuously true if their antecedent is wrong (in our case, if n is odd), so we only need to focus on the case where n is even. If n is even, then there is some integer k such that $n = 2k$. Hence $n^2 = (2k)^2 = 4k^2 = 2(2k^2)$. Clearly, $2k^2 = l$ for some integer l. But since any integer multiplied by 2 is even, so is $2l = 2(2k^2) = n^2$. □

- **Indirect proof/Proof by contradiction**

  This is the most common proof technique. We try to show that if all premises are fulfilled, a contradiction arises if the consequent does not hold. Note that in contrast to a direct proof, an indirect proof is not *constructive*. For example, the indirect proof below does not give us a procedure for computing n.

  Theorem 3.2. *If n^2 is even, n is even.*

  Proof. Our premise is that n^2 is even and the consequent is that n is even, too. So assume that n^2 is even, whereas n is odd. Therefore $n = 2k + 1$ for some integer k. Then

$n^2 = (2k+1)^2 = 4k^2 + 4k + 1$. As $4k^2 + 4k$ is even for every integer k, it follows that $n^2 = 4k^2 + 4k + 1$ is odd, contradicting our initial assumption that n^2 is even. Hence n must be even. □

- **Proof by contrapositive**
  Instead of showing that a implies b, we show that 'not b' implies 'not a' (the two are equivalent; why?).

  Proof. We proof the contrapositive of theorem 3.2, i.e. if n is odd, then n^2 is odd. Since n is odd, $n = 2k+1$ for some integer k. So $n^2 = (2k+1)^2 = 4k^2 + 4k + 1$, whence n^2 is odd, too. □

- **Proving iff-statements**
  To prove that a iff b, we first prove that a implies b and then that b implies a. For each case we may pick any suitable proof technique.

- **Proof by induction (or how to get a proof to say "and so on")**
  Out of all the techniques presented here, this is the most complex one, but next to indirect proofs it is also the most common one, so by the end of the course you should have it down if you want to do further work in mathematical linguistics. This technique is used to proof a result for an infinite sequence of objects. The general proof idea is to first establish the validity of the *base case* and then the validity of the *induction step*.

  Base case: The result is true for the "simplest case".
  Induction step: If the result holds for n, then it also holds for $n+1$.

Theorem 3.3. *Given a lexicon Lex, every tree that can be obtained from Lex by finitely many applications of the rule Merge is finite.*

Proof. Base case: The simplest case is a tree involving no applications of Merge at all. Such a tree consists of a single node labeled with a lexical item. Hence it is finite.

Induction step: Let T_1 and T_2 two arbitrary trees obtained from *Lex* by finitely many applications of the rule Merge. By our *induction hypothesis*, both T_1 and T_2 are finite. We need to show that $T := \text{Merge}(T_1, T_2)$, the tree resulting from the merger of T_1 and T_2, is also finite. It is easy to see from the definition of Merge that if T_1 and T_2 have a many and b many nodes, respectively, then T has $a + b + 1$ many. Hence T is finite. □

4 Literature

4.1 Supplementary readings

I second Ed's recommendation of Partee et al. (1990) and Gamut (1990a). Another beginner friendly math introduction is Schaum's textbook on Boolean Algebra (Mendelson 1970). This book is also considerably shorter than the other two. Mathematical proofs have a tendency to confuse and intimidate the uninitiated. If you find yourself struggling in that area, Cupillari (2005) will teach you how to read proofs, in less than 100 pages.

Those not familiar with linguistics may find Bruce Hayes' course material for Ling20 useful (`http://www.linguistics.ucla.edu/people/hayes/20/index.htm`). If you have problems understanding all that linguistic jargon, consult the online linguistics glossary (`http://www.sil.org/linguistics/GlossaryOfLinguisticTerms/`).

4.2 Specialized Introductions — If You Want to Dig Deeper

- *Linguistics*: Currently, Kracht (2003) is the only advanced introduction to mathematical linguistics. Fortunately, it is a masterpiece that covers more material than one could ever hope to find in a single textbook. It assumes a mathematical background roughly equivalent to what is covered in Partee et al. (1990). A useful (albeit sketchy) supplement to Kracht's book would be James Rogers' ESSLLI07 lecture notes, which I uploaded to the course website.

  If you want to learn more about how to use logic in the study of language, Gamut (1990b) will get you up to speed in numerous areas ranging from Montague semantics to Categorial grammar. In the area of generalized quantifiers, Peters and Westerståhl (2006) is widely considered the new standard reference. Finally, overview articles on many essential topics in mathematical linguistics can be found in van Benthem and ter Meulen (1997) (a new edition is currently in the making).

- *Logic*: Enderton (2001) is the standard undergrad introduction to mathematical logic. There are no specific prerequisites, but familiarity with mathematical reasoning and basic concepts like functions and relations is of great help. If you want to learn more about first-order logic after taking this class, you should definitely give Enderton a try. Some of you might prefer the less leisurely Ebbinghaus et al. (1996) (the fifth edition features solutions to selected exercises, but apparently it hasn't been translated to English yet).

  Two specialized areas that are of relevance to mathematical linguistics are (finite) model theory and modal logic. The standard textbook treatments are Ebbinghaus and Flum (1995), Hodges (1997) and Blackburn et al. (2002).

- *Formal language theory*: Formal language theory won't be tackled in this course. Nevertheless it forms an integral part of mathematical linguistics, so if you are a curious person, you might want to take a look at Sipser (2005), the darling of computer science undergrads. Personally, I think Kozen (1997) is a better book for beginners, mainly due to the plethora of exercises and its short chapters, which are ideal for a quick read during lunch break or before going to bed. Unless you are already familiar with the basic notions of the field or talented in math, avoid Hopcroft and Ullman (1979). Its an awesome book, but too austere for beginners.

4.3 Research Articles — Get Some Hands-On Experience

I prefer real research problems over textbook problem sets. If the same applies to you, here are some not-so-difficult articles you might want to check out:

- *Generative focus*: Barker and Pullum (1990), Kornai and Pullum (1990), Potts and Pullum (2002)
  These articles have a strong linguistic focus; they investigate the properties of command relations, the content of X'-theory, and the expressivity of various OT constraint classes.

- *Model-theoretic syntax*: Blackburn et al. (1993), Blackburn (1994), Rogers (1996), Rogers (2003)
  These articles study linguistic tools like trees and feature structures from a logical perspective. Rogers (2003) is the foundation for my own research and thus mandatory reading for you ;-)

References

Barker, Chris, and Geoffrey K. Pullum. 1990. A theory of command relations. *Linguistics and Philosophy* 13:1–34.

van Benthem, Johan, and Alice ter Meulen, ed. 1997. *Handbook of logic and language*. MIT Press.

Blackburn, Patrick. 1994. Structure, languages and translations: the structural approach to feature logic. In *Constraints, language and computation*, ed. C. Rupp, M. Rosner, and R. Johnson, 1–27. San Diego, CA: Academic Press.

Blackburn, Patrick, Claire Gardent, and Wilfried Meyer-Viol. 1993. Talking about trees. In *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics*, 30–36.

Blackburn, Patrick, Maarten de Rijke, and Yde Venema. 2002. *Modal logic*. Cambridge University Press.

Cupillari, Antonella. 2005. *The nuts and bolts of proofs*. Academic Press, third edition.

Ebbinghaus, Heinz-Dieter, and Jörg Flum. 1995. *Finite model theory*. Berlin: Springer.

Ebbinghaus, Heinz-Dieter, Jörg Flum, and Wolfgang Thomas. 1996. *Mathematical logic*. Springer.

Enderton, Herbert. 2001. *A mathematical introduction to logic*. San Diego, CA: Academic Press, 2^{nd} edition.

Gamut, L. T. F. 1990a. *Logic, language, and meaning, volume 1: Introduction to logic*. Chicago: University of Chicago Press.

Gamut, L. T. F. 1990b. *Logic, language, and meaning, volume 2: Intensional logic and logical grammar*. Chicago: University of Chicago Press.

Hodges, Wilfried. 1997. *A shorter model theory*. Cambridge University Press.

Hopcroft, John E., and Jeffrey D. Ullman. 1979. *Introduction to automata theory, languages, and computation*. Reading, Mass.: Addison Wesley.

Kornai, Andras, and Geoffrey K. Pullum. 1990. The X-bar theory of phrase structure. *Language* 66:24–50.

Kozen, Dexter C. 1997. *Automata and computability*. Springer.

Kracht, Marcus. 2003. Features in phonological theory. In *Foundations of the formal sciences II, applications of mathematical logic in philosophy and linguistics*, ed. Benedikt Löwe, Wolfgang Malzkorn, and Thomas Räsch, number 17 in Trends in Logic, 123–149. Dordrecht: Kluwer. Papers of a conference held in Bonn, November 11–13, 2000.

Mendelson, Elliott. 1970. *Schaum's outline of boolean algebra and switching circuits*. McGraw-Hill.

Partee, Barbara, Alice ter Meulen, and Robert Wall. 1990. *Mathematical methods in linguistics*. Dordrecht: Kluwer.

Peters, Stanley, and Dag Westerståhl. 2006. *Quantifiers in language and logic*. Oxford University Press.

Potts, Christopher, and Geoffrey K. Pullum. 2002. Model theory and the content of OT constraints. *Phonology* 19:361–393.

Rogers, James. 1996. A model-theoretic framework for theories of syntax. In *Proceedings of the 34th Annual Meeting of the ACL*, 10–16. Santa Cruz, USA.

Rogers, James. 2003. Syntactic structures as multi-dimensional trees. *Research on Language and Computation* 1:265–305.

Sipser, Michael. 2005. *Introduction to the theory of computation*. Course Technology, second edition.