

FOUNDATIONS OF MATHEMATICS, LECTURE 7

András Kornai

BMETE91AM35 Fall 2020-21

ORGANIZATION OF THIS CLASS

- First half (12:15 – 13:00) midterm discussion
- Second half (13:00 – 13:45) Lecture, continuing with discussion of logic, the language of formulas
- Only half of the people submitted \LaTeX !
- In traditional terms, 2 As; 5 Bs; 7 Cs; 1 D; 3Fs
- Solution key at `sol1.pdf`

LOGIC

- To define a logic we will need four things:
- A language to write formulas
- A notion of truth
- A notion of what the formulas mean ‘model theory’
- A deduction procedure ‘proof theory’
- We will discuss three main varieties: propositional, first order, and higher order logic
- We begin at the middle, even though both propositional and higher-order systems are substantially simpler

FIRST ORDER LANGUAGE

- It is convenient to use a very large (transfinite) list of *constants*. These are the things we want to talk about (points on the plane, sets, etc.)
- We also permit an infinite, but denumerable list of *variables* x, y, z, \dots to help us talk about many things at the same time
- Relation symbols, each with a fixed **arity** (the number of factors in the direct product) – again we can permit a more than denumerably infinite supply
- Connectives $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
- Quantifiers \forall, \exists and brackets $[,]$
- **context free languages** are given by a set of **nonterminals** V , a set of **terminals** Σ , and a **grammar** with **productions** in the form $A \rightarrow \alpha$ where $\alpha \in (\Sigma \cup V)^*$
- The **yield** of a CFG is the set of strings that can be obtained from a distinguished **start symbol** $S \in V$ by application of productions until no nonterminal is left

FOL (ALMOST) BY CFG

- Ignoring cardinality issues, the nonterminals include WFF, AF, Const, Var, and Rel_n . We will also have some technical symbols $,$ (comma) and $_$ (placeholder). Brackets, parentheses, connectives and quantifiers are considered terminal symbols, as are the individual constants, variables, and relation symbols
- The rules for **Atomic Formulas**: $AF \rightarrow R_n((-,)^{n-1} _)$ 'Each n-ary relation symbol must be followed by $(,$ a string of n empty slots separated by n-1 commas, and terminated by $)$ '
- $_ \rightarrow c, _ \rightarrow v$ 'Slots of n-ary relational symbols must be filled by constants or variables. So $R(a, x, b)$ is an Atomic Formula, but $S(x, _)$ is an incomplete atomic formula (doesn't count in the yield, because it still has a nonterminal $_$)
- To check if a string is an Atomic Formula, you need to check if it starts with a relational symbol, what is the arity of that symbol, and whether the slots are filled by variables/constants

MOVING FROM ATOMIC TO MORE COMPLEX FORMULAS

- 1 WFF \rightarrow [AF] 'bracketing an atomic formula gives a well-formed formula'
- 2 WFF \rightarrow [\neg WFF]||[WFF \vee WFF]||[WFF \wedge WFF]||[WFF \Rightarrow WFF]||[WFF \Leftrightarrow WFF] 'logical operations on WFFs lead to WFFs'
- 3 WFF \rightarrow [(\forall Var) WFF]||[(\exists Var) WFF] 'quantification'
- 4 We need to make sure that e.g. [($\forall x$)[($\exists x$) $R(a, x)$]] is *not* a WFF 'capturing variables'. This can't be done with a CFG (Type 2) , but very easy with a linear bounded TM (Type 1)
- 5 **HW7.1-3 Write ZFC_{1,2,5} in FOL** Remember = and \in are binary relations