Part III — Logic
Definitions

Based on lectures by T. E. Forster
Notes taken by Dexter Chua

Lent 2017

This course is the sequel to the Part II courses in Set Theory and Logic and in Automata and Formal Languages lectured in 2015-6. (It is already being referred to informally as "Son of ST&L and Automata & Formal Languages"). Because of the advent of that second course this Part III course no longer covers elementary computability in the way that its predecessor ("Computability and Logic") did, and this is reflected in the change in title. It will say less about Set Theory than one would expect from a course entitled 'Logic'; this is because in Lent term Benedikt Löwe will be lecturing a course entitled 'Topics in Set Theory' and I do not wish to tread on his toes. Material likely to be covered include: advanced topics in first-order logic (Natural Deduction, Sequent Calculus, Cut-elimination, Interpolation, Skolemisation, Completeness and Undecidability of First-Order Logic, Curry-Howard, Possible world semantics, Gödel's Negative Interpretation, Generalised quantifiers. . .); Advanced Computability (λ-representability of computable functions, Tennenbaum's theorem, Friedberg-Muchnik, Baker-Gill-Solovay. . .); Model theory background (ultraproducts, Los's theorem, elementary embeddings, omitting types, categoricity, saturation, Ehrenfeucht-Mostowski theorem. . .); Logical combinatorics (Paris-Harrington, WQO and BQO theory at least as far as Kruskal's theorem on wellquasiorderings of trees. . .). This is a new syllabus and may change in the coming months. It is entirely in order for students to contact the lecturer for updates.

Pre-requisites

The obvious prerequisites from last year's Part II are Professor Johnstone's Set Theory and Logic and Dr Chiodo's Automata and Formal Languages, and I would like to assume that everybody coming to my lectures is on top of all the material lectured in those courses. This aspiration is less unreasonable than it may sound, since in 2016-7 both these courses are being lectured the term before this one, in Michaelmas; indeed supervisions for Part III students attending them can be arranged if needed: contact me or your director of studies. I am lecturing Part II Set Theory and Logic and I am even going to be issuing a "Sheet 5" for Set Theory and Logic, of material likely to be of interest to people who are thinking of pursuing this material at Part III. Attending these two Part II courses in Michaelmas is a course of action that may appeal particularly to students from outside Cambridge.

Contents

1 Proof theory and constructive logic

1.1 Natural deduction

Definition (Harmony)**.** We say rules for a connective \$ are *harmonious* if $\phi\$\psi$ is the strongest assertion you can deduce from the assumptions in the rule of \$-introduction, and $\phi\$\psi$ is the weakest thing that implies the conclusion of the \$-elimination rule.

Definition (Maximal formula)**.** We say a formula in a derivation is *maximal* iff it is both the conclusion of an occurrence of an introduction rule, *and* the major premiss of an occurrence of the elimination rule for the same connective.

1.2 Curry–Howard correspondence

1.3 Possible world semantics

Definition (Possible world semantics)**.** Let P be a collection of propositions. A *world* is a subset $w \subseteq P$. A *model* is a collection W of worlds, and a partial order $\geq$ on W called *accessibility*, satisfying the *persistence* property:

– If $p \in P$ is such that $p \in w$ and $w' \geq w$, then $p \in w'$.

Given any proposition φ, we define the relation $w \vDash \varphi$ by

– $w \not\vDash \perp$
– If φ is *atomic* (and not $\perp$), then then $w \vDash \varphi$ iff $\varphi \in w$.
– $w \vDash \varphi \wedge \psi$ iff $w \vDash \varphi$ and $w \vDash \psi$.
– $w \vDash \varphi \vee \psi$ iff $w \vDash \varphi$ or $w \vDash \psi$.
– $w \vDash \varphi \to \psi$ iff (for all $w' \geq w$, if $w' \vDash \varphi$, then $w' \vDash \psi$).

We will say that w "believes" φ if $w \vDash \varphi$, and that w "sees" w' if $w' \geq w$.

We also require that there is a designated minimum element under $\leq$, known as the *root world*. We can think of a possible world model as a poset decorated with worlds, and such a poset is called a *frame*.

Notation. If the root world of our model is w, then we write

$$\vDash \varphi \iff w \vDash \varphi.$$

Definition (Heyting algebra)**.** A Heyting algebra is a poset with $\top$, $\perp$, $\wedge$ and $\vee$ and an operator $\Rightarrow$ such that $A \Rightarrow B$ is the largest C such that

$$C \wedge A \leq B.$$

1.4 Negative interpretation

Definition (Negative interpretation)**.** Given a proposition ϕ, the interpretation ϕ^* is defined recursively by

– $\perp^* = \perp$.

- If φ is atomic, then $\varphi^* = \neg\neg\varphi$.
- If φ is negatomic, then $\varphi^* = \varphi$.
- If $\varphi = \psi \wedge \theta$, then $\varphi^* = \psi^* \wedge \theta^*$.
- If $\varphi = \psi \vee \theta$, then $\varphi^* = \neg(\neg\psi^* \wedge \neg\theta^*)$.
- If $\varphi = \forall_x \psi(x)$, then $(\forall_x)(\psi^*(x))$.
- If $\varphi = \psi \to \theta$, then $\varphi^* = \neg(\psi^* \wedge \neg\theta^*)$.
- If $\varphi = \exists_x \psi(x)$, then $\varphi^* = \neg\forall_x \neg\psi^*(x)$.

Definition (Stable formula)**.** A formula is *stable* if

$$\vdash \varphi^* \to \varphi.$$

1.5 Constructive mathematics

Definition (Kuratowski finite)**.** We define "finite" recursively: $\emptyset$ is Kuratowski finite. If x is Kuratowski finite, then so is $x \cup \{y\}$.

Definition (N-finite)**.** $\emptyset$ is N-finite. If x is N-finite, and $y \notin x$, then $x \cup \{y\}$ is N-finite.

Definition (Non-empty set)**.** A set x is *non-empty* if $\neg\forall_y y \notin x$.

Definition (Inhabited set)**.** A set x is *inhabited* if $\exists_y y \in x$.

2 Model theory

2.1 Universal theories

Definition (Universal theory)**.** A universal theory is a theory that can be axiomatized in a way such that all axioms are of the form

$$\forall_{\ldots}(\text{stuff not involving quantifiers}) \tag{$*$}$$

Definition (Diagram)**.** Let $\mathcal{L}$ be a language and $\mathcal{M}$ a structure of this language. The *diagram* of $\mathcal{M}$ is the theory obtained by adding a constant symbol a_x for each $x \in \mathcal{M}$, and then taking the axioms to be all quantifier-free sentences that are true in M. We will write the diagram as $D(\mathcal{M})$.

2.2 Products

Definition (Product of structures)**.** Suppose $\{A_i\}_{i \in I}$ is a family of structures of the same signature. Then the product

$$\prod_{i \in I} A_i$$

has carrier set the set of all functions

$$\alpha : I \to \bigcup_{i \in I} A_i$$

such that $\alpha(i) \in A_i$.

Given an n-ary function f in the language, the interpretation in the product is given pointwise by

$$f(\alpha_1, \cdots, \alpha_n) = \lambda i. f(\alpha_1(i), \cdots, \alpha_n(i)).$$

Relations are defined by

$$\varphi(\alpha_1, \cdots, \alpha_n) = \bigwedge_{i \in I} \varphi(\alpha_1(i), \cdots, \alpha_n(i)).$$

Definition (Equational theory)**.** An equational theory is a theory all of whose axioms are of the form

$$\forall_{\mathbf{x}}(w_1(\mathbf{x}) = w_2(\mathbf{x})),$$

where w_i are some terms in $\mathbf{x}$.

Definition (Horn clause)**.** A *Horn clause* is a disjunction of atomics and negatomics of which at most one disjunct is atomic.

It is usually better to think of Horn clauses as formulae of the form

$$\left(\bigwedge \varphi_i\right) \to \chi$$

where φ_i and χ are atomic formulae. Note that $\perp$ is considered an atomic formula.

A *universal Horn clause* is a universal quantifier followed by a Horn clause.

Definition (Filter)**.** Let I be a set. A *filter* on I is a (non-empty) subset $F \subseteq P(I)$ such that F is closed under intersection and superset. A *proper filter* is a filter $F \neq P(I)$.

Definition (Principal filter)**.** A *principal filter* is a filter of the form

$$F = \{X \subseteq I : x \notin X\}$$

for some $x \in I$.

Definition (Complete filter)**.** A filter F is κ-complete if it is closed under intersection of $< \kappa$ many things.

Definition (Ultrafilter)**.** An *ultrafilter* is a maximal filter.

Definition (Reduced product)**.** let $\{A_i : i \in I\}$ be a family of structures, and F a filter on I. We define the *reduced product*

$$\prod_{i \in I} A_i / F$$

as follows: the underlying set is the usual product $\prod A_i$ quotiented by the equivalence relation

$$\alpha \sim_F \beta \Longleftrightarrow \{i : \alpha(i) = \beta(i)\} \in F$$

Given a function symbol f, the interpretation of f in the reduced product is induced by that on the product.

Given a relational symbol φ, we define

$$\varphi(\alpha_1, \cdots, \alpha_n) \Longleftrightarrow \{i : \varphi(\alpha_1(i), \cdots, \alpha_n(i))\} \in F.$$

If F is an ultrafilter, then we call it the *ultraproduct*. If all the factors in an ultraproduct are the same, then we call it an *ultrapower*.

2.3 Ehrenfeucht–Mostowski theorem

Definition (Skolem function)**.** *Skolem functions* for a structure are functions f_φ for each $\varphi \in \mathcal{L}$ such that if

$$\mathcal{M} \vDash \forall_{\mathbf{x}} \exists_{\mathbf{y}} \varphi(\mathbf{x}, \mathbf{y}),$$

then

$$\mathcal{M} \vDash \forall_{\mathbf{x}} \varphi(\mathbf{x}, f_\varphi(\mathbf{x})).$$

Definition (Skolem hull)**.** The *Skolem hull* of a structure is obtained from the constants term by closure under the Skolem functions.

Definition (Elementary embedding)**.** Let Γ be a set of formulae. A function $i : \mathcal{M}_1 \to \mathcal{M}_2$ is Γ-*elementary* iff for all $\varphi \in \Gamma$ we have $\mathcal{M}_1 \vDash \varphi(\mathbf{x})$ implies $\mathcal{M}_2 \vDash \varphi(i(\mathbf{x}))$.

If Γ is the set of all formulae in the language, then we just say it is *elementary*.

Definition (Monadic first order logic)**.** *Monadic first-order logic* is first order logic with only one-place predicates, no equality and no function symbols.

Definition (Set of indiscernibles). We say $\langle I, \leq_I \rangle$ is a *set of indiscernibles* for $\mathcal{L}$ and a structure $\mathcal{M}$ with $I \subseteq \mathcal{M}$ if for any $\varphi \in \mathcal{L}(M)$ of arity n, and for all increasing tuples $\mathbf{x}, \mathbf{y} \in I$,

$$\mathcal{M} \vDash \varphi(\mathbf{x}) \Longleftrightarrow \mathcal{M} \vDash \varphi(\mathbf{y})$$

Definition (Colimit). Let $\{A_i : i \in I\}$ be a family of structures index by a poset $\langle I, \leq \rangle$, with a family of (structure-preserving) maps $\{\sigma_{ij} : A_i \hookrightarrow A_j \mid i \leq j\}$ such that whenever $i \leq j \leq k$, we have

$$\sigma_{jk}\sigma_{ij} = \sigma_{ik}.$$

In particular σ_{ii} is the identity. A *colimit* or *direct limit* of this family of structures is a "minimal" structure A_∞ with maps $\sigma_i : A_i \hookrightarrow A_\infty$ such that whenever $i \leq j$, then the maps

$$\begin{array}{ccc} A_i & \xrightarrow{\sigma_i} & A_\infty \\ {\scriptstyle \sigma_{ij}}\downarrow & \nearrow_{\sigma_j} & \\ A_j & & \end{array}$$

commute.

By "minimal", we mean if A'_∞ is another structure with this property, then there is a unique inclusion map $A_\infty \hookrightarrow A'_\infty$ such that for any $i \in I$, the maps

$$\begin{array}{ccc} A_i & \xrightarrow{\sigma_i} & A_\infty \\ & \searrow^{\sigma'_i} & \downarrow \\ & & A'_\infty \end{array}$$

2.4 The omitting type theorem

Definition (Type). A *type* is a set of formulae all with the same number of free variables. An *n-type* is a set of formulae each with n free variables.

Definition (Realization of type). A model $\mathcal{M}$ *realizes an n-type* Σ if there exists $x_1, \cdots, x_n \in \mathcal{M}$ such that for all $\sigma \in \Sigma$, we have

$$\mathcal{M} \vDash \sigma(x_1, \cdots, x_n).$$

Definition (Omit a type). A model $\mathcal{M}$ *omits an n-type* Σ if for all $x_1, \cdots, x_n$, there exists $\sigma \in \Sigma$ such that

$$\mathcal{M} \nvDash \sigma(x_1, \cdots, x_n).$$

Definition (Locally realize). We say φ realizes Σ locally if

$$T \vdash \forall_x(\varphi(x) \to \sigma(x)).$$

3 Computability theory

3.1 Computability

Definition (Primitive recursive functions)**.** The class of *primitive recursive functions* $\mathbb{N}^n \to \mathbb{N}^m$ for all n, m is defined inductively as follows:

- The constantly zero function $f(\mathbf{x}) = 0$, $f : \mathbb{N}^n \to \mathbb{N}$ is primitive recursive.
- The successor function $\mathsf{succ} : \mathbb{N} \to \mathbb{N}$ sending a natural number to its successor (i.e. it "plus one") is primitive recursive.
- The identity function $\mathsf{id} : \mathbb{N} \to \mathbb{N}$ is primitive recursive.
- The projection functions

$$\begin{aligned} \mathsf{proj}^i_j(\mathbf{x}) : \mathbb{N}^j &\longrightarrow \mathbb{N} \\ (x_1, \cdots, x_j) &\longmapsto x_i \end{aligned}$$

  are primitive recursive.

Moreover,

- Let $f : \mathbb{N}^k \to \mathbb{N}^m$ and $g_1, \cdots, g_k : \mathbb{N}^n \to \mathbb{N}$ be primitive recursive. Then the function

$$(x_1, \cdots, x_n) \mapsto f(g_1(x_1, \cdots, x_n), \cdots, g_k(x_1, \cdots, x_n)) : \mathbb{N}^n \to \mathbb{N}^m$$

  is primitive recursive.

Finally, we have closure under *primitive recursion*

- If $g : \mathbb{N}^k \to \mathbb{N}^m$ and $f : \mathbb{N}^{m+k+1} \to \mathbb{N}^m$ are primitive recursive, then so is the function $h : \mathbb{N}^{k+1} \to \mathbb{N}^m$ defined by

$$\begin{aligned} h(0, \mathbf{x}) &= g(\mathbf{x}) \\ h(\mathsf{succ}\, n, \mathbf{x}) &= f(h(n, \mathbf{x}), n, \mathbf{x}). \end{aligned}$$

Definition (Ackermann function)**.** The *Ackermann function* is defined to be

$$\begin{aligned} A(0, n) &= n + 1 \\ A(m, 0) &= A(m - 1, 1) \\ A(m + 1, n + 1) &= A(m, A(m + 1, n)). \end{aligned}$$

Definition (Dominating function)**.** Let $f, g : \mathbb{N} \to \mathbb{N}$ be functions. Then we write $f < g$ if for all sufficiently large integer n, we have $f(n) < g(n)$. We say g *dominates* f.

Notation. We write $f(x) \uparrow$ if $f(x)$ is undefined, or alternatively, after we define what this actually means, if the computation of $f(x)$ doesn't halt. We write $f(x) \downarrow$ otherwise.

Definition (Partial recursive function)**.** The class of *partial recursive functions* is given inductively by

- Every primitive recursive function is partial recursive.
- The inverse of every primitive recursive function is partial recursive, where if $f : \mathbb{N}^{k+n} \to \mathbb{N}^m$, then the/an inverse of f is the function $f^{-1} : \mathbb{N}^{k+m} \to \mathbb{N}^n$ given by
$$f^{-1}(\mathbf{x}; \mathbf{y}) = \begin{cases} \min\{\mathbf{z} : f(\mathbf{x}, \mathbf{z}) = \mathbf{y}\} & \text{if exists} \\ \uparrow & \text{otherwise} \end{cases}.$$
- The set of partial recursive functions is closed under primitive recursion and composition.

3.2 Decidable and semi-decidable sets

Definition (Decidable set)**.** A subset $X \subseteq \mathbb{N}$ is *decidable* if there is a total computable function $\mathbb{N} \to \mathbb{N}$ such that
$$f(n) = \begin{cases} 1 & n \in X \\ 0 & n \notin X \end{cases}.$$

Definition (Semi-decidable set)**.** We say a subset $X \subseteq \mathbb{N}$ is *semi-decidable* if it satisfies one of the following equivalent definitions:

(i) X is the image of some partial computable function $f : \mathbb{N} \to \mathbb{N}$.

(ii) X is the image of some total computable function $f : \mathbb{N} \to \mathbb{N}$.

(iii) There is some partial computable function $f : \mathbb{N} \to \mathbb{N}$ such that
$$X = \{n \in \mathbb{N} : f(n) \downarrow\}$$

(iv) The function $\chi_X : \mathbb{N} \to \{0\}$ given by
$$\chi_X = \begin{cases} 0 & n \in X \\ \uparrow & n \notin X \end{cases}$$
is computable.

Definition (Halting set)**.** The *halting set* is
$$\{\langle p, i \rangle : \{p\}(i) \downarrow\} \subseteq \mathbb{N}^2.$$
Some people prefer to define it as
$$\{m : \{m\}(m) \downarrow\} \subseteq \mathbb{N}$$
instead.

3.3 Computability elsewhere

3.4 Logic

Definition (Productive set)**.** A set $X \subseteq \mathbb{N}$ is *productive* if there is a total computable function $f : \mathbb{N} \to \mathbb{N}$ such that for all $n \in \mathbb{N}$, we have $f(n) \in X \setminus (\mathrm{im}\{n\})$.

Definition (Sound theory)**.** A *sound* theory of arithmetic is one all of whose axioms are true in $\mathbb{N}$.

3.5 Computability by λ-calculus

Definition (λ terms)**.** The set Λ of λ terms is defined recursively as follows:

- If x is any variable, then $x \in \Lambda$.
- If x is a variable and $g \in \Lambda$, then $\lambda x.\ g \in \Lambda$.
- If $f, g \in \Lambda$, then $(f\ g) \in \Lambda$.

Definition (Free and bound variables)**.**

- In the λ term x, we say x is a free variable.
- In the λ term $\lambda x.\ g$, the free variables are all free variables of g except x.
- In the λ term $(f\ g)$, the free variables is the union of the free variables of f and those of g,

The variables that are not free are said to be bound.

Definition (α-equivalence)**.** We say two λ terms are α-equivalent if they are the same up to renaming of bound variables.

Definition (β-reduction)**.** If $f = (\lambda x.\ y(x))z$ and $g = y(z)$, then we say g is obtained from f via β-reduction.

Definition (η-reduction)**.** η-conversion is the conversion from $\lambda x.\ (f\ x)$ to f, whenever x is not free in f.

Definition (β-normal form)**.** We say a term is in β-normal form if it has no possible β-reduction.

Notation ($\rightsquigarrow$)**.** We write $f \rightsquigarrow g$ if f β-reduces to g.

Definition (Church numerals)**.** We define

$$\begin{aligned} \underline{0} &= K(\mathrm{id}) = \lambda f.\ \lambda x.\ x : \mathbf{N} \\ \mathsf{succ} &= \lambda n.\ \lambda f.\ \lambda x.\ f\ ((n\ f)\ x) : \mathbf{N} \to \mathbf{N} \end{aligned}$$

We write $\underline{n} = \mathsf{succ}^n(\underline{0})$.

We can define arithmetic as follows:

$$\begin{aligned} \mathsf{plus} &= \lambda n.\ \lambda m.\ \lambda f.\ \lambda x.\ (n\ f)\ ((m\ f)\ x) : \mathbf{N} \to \mathbf{N} \to \mathbf{N} \\ \mathsf{mult} &= \lambda n.\ \lambda m.\ \lambda f.\ \lambda x.\ (n\ (m\ f))\ x : \mathbf{N} \to \mathbf{N} \to \mathbf{N} \\ \mathsf{exp} &= \lambda n.\ \lambda m.\ \lambda f.\ \lambda x.\ ((n\ m)\ f))\ x : \mathbf{N} \to \mathbf{N} \to \mathbf{N} \end{aligned}$$

Here $\exp n\ m = n^m$.

Definition (Y-combinator)**.** The Y-*combinator* is

$$\mathsf{Y} = \lambda f.\Big[(\lambda x.f(x\ x))(\lambda x.f(x\ x))\Big].$$

3.6 Reducibility

Definition (Many-to-one reducibility)**.** Let $A, B \in \mathbb{N}$. We write $B \leq_m A$ if there exists a total computable functions $f : \mathbb{N} \to \mathbb{N}$ such that for all $n \in \mathbb{N}$, we have $n \in B \leftrightarrow f(n) \in A$.

Definition (Turing reducibility)**.** We say $B \leq_T A$, or simply $B \leq A$, if it is possible to determine membership of B whenever we have access to an oracle that computes χ_A.

4 Well-quasi-orderings

Definition (Quasi-order)**.** An order $\langle X, \leq\rangle$ is a *quasi-order* if it is *transitive* and *reflexive*, i.e. for all $a, b, c \in X$,

– If $a \leq b$ and $b \leq c$, then $a \leq c$.

– We always have $a \leq a$.

Definition (Well-founded relation)**.** A relation R on X is said to be *well-founded* if for all $A \subseteq X$ non-empty, there is some $x \in A$ such that for any $y \in A$, we have $\neg R(y, x)$.

Definition (Well-ordered quasi-order)**.** A quasi-order is *well-ordered* if there is no *strictly* decreasing infinite sequence.

Definition ($\mathcal{P}(X)$)**.** Let $\langle X, \leq_X\rangle$ be a quasi-order. We define a quasi-order $\leq_X^+$ on $\mathcal{P}(X)$ by

$$X_1 \leq_X^+ X_2 \text{ if } \forall_{x_1 \in X_1} \exists_{x_2 \in X_2} (x_1 \leq_X x_2).$$

Definition ($X^{<\omega}$)**.** Let $\langle X, \leq_X\rangle$ be a quasi-order. We let $X^{<\omega}$ be the set of all *finite* lists (i.e. sequences) in X. We define a quasi-order on $X^{<\omega}$ recursively by

– $\mathsf{nil} \leq \ell_1$, where nil is the empty list.

– $\mathsf{tail}(\ell_1) \leq \ell_1$

– If $\mathsf{tail}(\ell_1) \leq \mathsf{tail}(\ell_2)$ and $\mathsf{head}(\ell_1) \leq_X \mathsf{head}(\ell_2)$, then $\ell_1 \leq \ell_2$.

for all lists ℓ_1, ℓ_2.

Equivalently, sequences $\{x_i\}_{i=1}^n \leq_s \{y_i\}_{i=1}^\ell$ if there is a subsequence $\{y_{i_k}\}_{k=1}^n$ of $\{y_i\}$ such that $x_k \leq y_{i_k}$ for all k.

Definition (Tree)**.** Let $\langle X, \leq_X\rangle$ be a quasi-order. The set of all trees (in X) is defined inductively as follows:

– If $x \in X$ and L is a list of trees, then (x, L) is a tree. We call x a *node* of the tree. In particular (x, nil) is a tree. We write

$$\text{root}(x, L) = x, \quad \text{children}(x, L) = L.$$

Haskell programmers would define this by

```
data Tree a = Branch a [Tree a]
```

We write Trees(X) for the set of all trees on X.

We define an order relation $\leq_s$ on Trees(X) as follows — let $T_1, T_2 \in \text{Trees}(X)$. Then $T_1 \leq T_2$ if

(i) $T_1 \leq T'$ for some $T' \in \text{children}(T_2)$.

(ii) $\text{root}(T_1) \leq \text{root}(T_2)$ and $\text{children}(T_1) \leq \text{children}(T_2)$ as lists.

Definition (Well-quasi-order)**.** A *well-quasi-order* (WQO) is a well-founded quasi-order $\langle X, \leq_X\rangle$ such that $\langle \mathcal{P}(X), \leq_X^+\rangle$ is also well-founded.

Definition (Well-founded part of a relation)**.** Let $\langle X, R\rangle$ be a set with a relation. Then the *well-founded part* of a relation is the $\subseteq$-least subset $A \subseteq X$ satisfying the property

- If x is such that all predecessors of x are in A, then $x \in A$.

Definition (Bad sequence)**.** Let $\langle X, \leq\rangle$ be a well-founded quasi-order. A sequence $\{x_i\}$ is *bad* if for all $i < j$, we have $f(i) \not\leq f(j)$.

Definition (Minimal bad sequence)**.** Let $\langle X, \leq\rangle$ be a quasi-order. A *minimal bad sequence* is a bad sequence $\{x_i\}$ such that for each $k \in \mathbb{N}$, x_k is a $\leq$-minimal element in

$$\{x : \text{there exists a bad sequence starting with } x_1, x_2, \cdots, x_{k-1}, x\}.$$

Definition (ω^2-good)**.** A well-ordering $\langle X, \leq\rangle$ is ω^2-*good* if for any

$$f : \{\langle i, j\rangle : i < j \in \mathbb{N}\} \to X,$$

there is some $i < j < k$ such that $f(i, j) \leq f(j, k)$.