Extensions of First Order Logic

María Manzano y Antonia Huertas

August 12-23 1996

Contents

0.1 INTRODUCTION

This course considers various extensions of first order logic, giving detailed and elaborate treatment to many useful logical systems: second order logic (SOL), type theory (RTT, ETT and FTT), modal logic (PML and FOML), dynamic logic (PDL) and many-sorted logic (MSL). A substantial dose of logical perspective is also provided.

The second objective of this course is to pursue the thesis that most reasonable logical systems can be naturally translated into many-sorted first order logic. The thesis is maintained throughout the course, but only appears openly and explicitly in the last part. There, all the logic systems treated are put in direct correspondence with many-sorted logic because this logic offers a unifying framework in which to place other logics. In itself, many-sorted logic is a natural logic for formalizing statements in a great variety of disciplines and it also has an efficient proof theory with a complete deductive calculus.

Currently, the proliferation of logics used in philosophy, computer science, artificial intelligence, mathematics and linguistics makes a working reduction of this variety an urgent issue. The aim is two-fold:

- To be able to use only one deductive calculus and a unique theorem prover for all logics -i.e., an MSL theorem prover;
- To avoid the proofs of the metaproperties of the different existing logics by borrowing them from many-sorted logic.

The appeal of this approach is that it is so intuitive and easy that only common sense is needed to understand the construction. Besides, as the basic ingredients change, the recipe can be adapted and used to prepare different dishes. So with very little effort the results obtained are considerable. It is difficult to trace the development of this approach because almost every non-classical logic has found its standard counterpart at birth. Nevertheless, I like to credit most of the ideas involved in our current presentation to Henkin's paper "Completeness in the theory of types" (1950). However I do not want to be misleading you are not going to find in this paper of 1950 anything like the translation of formulas into another formal language, or the open appearance of many-sorted logic. In connection with SOL, many-sorted logic appeared later, in Henkin's "Banishing the rule of substitution for functional variables" (1953), where a new second order calculus with the comprehension rule was presented. As noted in that paper, from this calculus it is possible to isolate the many-sorted calculus by leaving out the comprehension rule. Another remarkable discovery included in the 1953 paper is that we can weaken comprehension so that it applies only to a restricted class of formulas of our choice. This is of great help when treating the modal and dynamic logics where we restrict comprehension to the sets and relations defined by translations of formulas of PML or of PDL.

In Henkin 1950 paper the completeness of type theory is proved and the general models are presented. How did Henkin prove the completeness theorem

for type theory? A very rapid answer to this question is: by changing the semantics and hence the logic. Roughly presented, the idea is very simple: The set of validities is so wide because our class of standard structures is too small. We have been very restrictive when requiring the relational universes of any model to contain all possible relations (where "possible" means in the background set theory used as metalanguage) and we have paid a high price for it. If we also allow nonstandard structures, and if we now interpret validity as being true in all general models, redefining all the semantic notions referring to this larger class of general structures, completeness (in both weak and strong senses), Löwenheim-Skolem, and all these theorems can be proven as in first order logic.

In addition to its usefulness, the general model's construction is far from being an ad hoc solution lacking naturalness and common sense. Throughout the pages of this reader you will find good reasons for wondering whether the philosophy of standard structures is the only possible choice. The reasons are directly related to the following questions:

1. Are we satisfied with the limitation on the class of models they require? Would it not be highly instructive to discover new and sensible models for a known existing theory?

2. Don't we feel uneasy about crossing the border with set theory? Don't second order validities refer to the given set-theoretical environment instead of the logic in itself?

3. Do we need all the expressive power they provide?

4. Are we willing to pay the price that standard semantics demands?

Further motivation for using general models may be found in van Benthem's essay "The sources of complexity".

In fact, when considering some of the arguments that we will use in the second chapter of this book, one can argue that the standard semantics is not logically adequate in the sense that it does not allow all logically possible interpretations of second order formulas as models because of the argument posed by Németi in the following form:

> We have to be placed in a set-theoretical universe, even assuming that there could be more than one such. Nevertheless, in the set theoretical universe you choose to be in, the GCH is either true or false. Assume it is true. Then, in every standard model for SOL the SOL formula φ expressing this hypothesis is true and so φ is valid. But since GCH is not derivable from ZFC, the result suggest that an interpretation $\mathcal{I}$ such that $\mathcal{I} \not\models \varphi$ can not be excluded as "logically impossible". So, at least one $\mathcal{I}$ with $\mathcal{I} \models \neg\varphi$ is a logical possibility (by Paul Cohen's classical result). But such a model is not allowed in the standard semantics. So, we feel that the standard

> semantics does not include all logically possible worlds as models (we have to think about formulas, like GCH, which are both expressible in second order logic and independent from Zermelo-Fraenkel set theory). This argument is reinforced by the fact that there is an inexhaustible supply of independent formulas like GCH. In Henkin's general semantics many possibilities are restored as possible models; for instance, models with or without the GCH.

As you will see, the general model strategy is also used in modal logic and dynamic logic. Both logics are faithfully represented by many-sorted theories with a comprehension schema restricted to a definable subclass of many-sorted formulas.

This course is inspired in Manzano's book [1]. A brief description of each chapter follows.

Chapter I. Standard Second Order Logic

The first chapter is an introduction to standard second order logic with emphasis placed on the expressive power this logic provides. It consists of extending first order logic to second order logic by allowing quantification over sets and relations. Also mentioned are the model-theoretic counterparts of expressiveness, without overlooking the incompleteness result.

A section is devoted to standard semantics where meaning is given to formulas of the formal language by introducing the related notions of satisfiability, validity and consequence. The question of what sets and relations are definable is then considered and a number of notions of definability are proposed. The chapter closes by proving a series of semantic theorems for standard second order logic, including the coincidence lemma, the substitution lemma and the isomorphism theorem.

Chapter II. Deductive Calculi

A number of deductive calculus for second order logic are defined and soundness and incompleteness results are presented.

The chapter begins with an informal introduction defining what a calculus is, explaining the usual motivations for wanting a calculus, and stating the desired metaproperties of a calculus; i.e., that it would never drive us to erroneous reasoning and that it would also help us to derive all the consequences of a given set of hypotheses. Of the three calculi introduced, the first is a very simple extension of a first order calculus of sequents where the quantifiers' rules also cover the set and relation variables. It is defined for a second order language where equality among individuals is treated as in first order logic; that is, it is a primitive or logical symbol rather than a defined one. The calculus also contains the rules dealing with equality. Since we also have equality for predicates as primitive, we adjoin some equality rules for them and we will have the extensionality axiom as a rule without hypothesis. This calculus will not count as a second order calculus for many people. In fact, it is plain many-sorted, and in an imprecise way we can name it MSL. Adding comprehension or lambda

rules we obtain proper second order calculus. Many deductions are done in full as exercises.

Another section is devoted to proving the soundness theorems for the three calculi. There is a very easy proof of incompleteness of MSL with standard semantics showing that $\exists x X\ Xx$ is valid in the standard semantics but is not derivable in MSL.

The chapter closes by proving the incompleteness result for second order logic with standard semantics, in both weak and strong senses, for any calculus.

Chapter III. Categoricity of Second Order Peano Arithmetic

The definition of standard structure is given and the common relationships between standard structures are defined. Specifically, notions such as substructure, homomorphism, isomorphism and embedding are dealt with.

This chapter introduces Peano arithmetic in SOL and proves that with the standard semantics this theory is categorical; that is, any two second order models of Peano arithmetic are isomorphic. The question of non-standard models of first order Peano arithmetic is raised and put in direct correspondence with non-standard models of second order logic; quite different meanings for the word "non-standard" which are, nevertheless, related.

Later it is proved that in Peano models it is possible to introduce recursive operations of any kind, but induction models where not all recursive operations can be introduced are also defined. The induction models which are not Peano models come in only two shapes: cycles and "spoons".

Chapter IV. Frames and General Structures

This chapter consists of six sections. In the first, frames and general structures are informally introduced and the dichotomy between standard and non-standard views is discussed, arriving at the conclusion that it is intimately related with how the concept of subset is considered: that is, as a "logical" or undefined concept or as something to be defined in the logic.

In two other sections frames and general structures with the related semantics are introduced. The question of whether the general structures can be defined algebraically is dealt with in another section. The logics obtained by weakening comprehension, whose relevance is pointed out in this book, are also dealt with. The chapter closes by considering weak second order logic; that is, a logic where the concept of finiteness is taken from the metatheory and imposed as a "logical" concept.

.Chapter V Type Theory

This chapter basically consists of the presentation of three different languages for type theory and a brief discussion of paradoxes and their solution in type theory. A deductive calculus for type theory is presented, which is a simple extension for all types of one for our second order calculus. The semantics of frames and general structures is also defined, and there is a very detailed proof of the equivalence of the usual definition of general structures with a proposed

algebraic definition of general structures. The original functional presentation of Church is also treated, and the relationship with the previous relational presentation is given in full. Another section is devoted to equational type theory, a very illuminating logic where the only primitives are equality and abstraction and where the remaining logical concepts including connectors and quantifiers are definable. The chapter closes with our obtaining a calculus for this language.

Chapter VI. Many-sorted Logic

This chapter consists of eight sections devoted to many-sorted logic. There is a long introduction where reasons for using many-sorted logic are provided. The fact that in many branches of mathematics and computer science the natural structures are many-sorted is highlighted. The standard treatment of reduction to single-sorted structures and logic is raised and the weak points of this reduction are considered. It is seen that many-sorted logic has been used with success in computer science and also in a wide range of logics, as a unifier logic. Many-sorted language, semantics and deductive calculus are presented in detail and the completeness of this calculus is proved in full by extending the usual method. The chapter closes with the reduction of many-sorted logic to first order unsorted logic in the classical way; namely, unification of domains for the conversion of structures and relativization of quantifiers in the translation of formulas.

Chapter VII. Applying Many-Sorted Logic

In the final chapter, in line with the philosophy presented throughout the book, all the logics thus far discussed, including modal and dynamic logic, are translated into many-sorted logic and their usual structures are converted into many-sorted structures thus giving us many-sorted theories. These theories are the standard counterparts of the original logics and are capable of representing them faithfully. In addition, some of the metaproperties of many-sorted semantics and calculus are transferred to these logics. In the first section of this chapter the general plan of the translation, its aims and usual steps are discussed. The second section is devoted to the translation of higher order logic into many-sorted logic. Using this translation technique we obtain completeness and soundness results for higher order logic. Modal logic is afterwards introduced, and a many-sorted theory, MODO, is proposed which proves the usual semantic theorems of compactness, enumerability and Löwenheim-Skolem and also tests the calculus indirectly by proving soundness and completeness. MODO(S4) does the same for the modal logic S4. For PDL the many-sorted theory proposed is SOLO^2.

This reader is mainly a collection of definitions and statements of theorems. During the course we will explain the concepts we are introducing and we will also prove (at least in sketch form) the theorems. As we noted before, a substantial dose of logical perspective will also be provided. The whole proofs are in the above mentioned book [1].

Chapter 1

STANDARD SECOND ORDER LOGIC

1.1 Second Order Grammar

A specific second order language is defined by giving its alphabet and the rules for its calculus of formulas. The alphabet contains enough logical symbols, quantifiable variables of several types and a possibly empty set of operation constants. Between two second order languages the differences always lay in the alphabet and they may affect any of these sets, but while some of these differences can be considered just minor ones (for example, when they only affect the operation constants), others can have greater relevance (for instance, when they affect the quantifiable relation variables by restricting them to unary relation variables). The set of logical symbols of a second order language always contains enough connectives and quantifiers, but it might or might not include equality. As in first order logic, we say that the set of connectives is complete when all the truth functions can be defined in terms of connectives in the set and similarly for the quantifiers. The language is second order in so far as it contains quantifiable individual and relation variables.

Definitions (signature and alphabet)

To specify a particular second order language we will give its signature. From the signature we learn the kinds of quantifiable variables and we also learn the relation and function constants we are having and their types. The only thing we specify separately is whether we are having λ-abstraction or not.

A signature Σ is a pair $\langle$VAR, FUNC$\rangle$ where VAR is the set containing all the kinds of quantifiable variables (in higher order logic the kinds are types, in many-sorted logic they are sorts) and FUNC is a function whose domain is the set OPER.CONS of operation constants of the language and it gives types as values; i.e., finite sequences of members of VAR.

In 2.1.1 we will present the signature of a second order language in a very general case; i.e., it might have function variables as well as relation variables and all of them are quantified. Moreover, if we have functions of degree n, then we have functions of degree 1,..., $n-1$ as well; so we can continually go from unary variables to n-ary variables.

Signature of any second-order language

(Please, skip this definition in the first reading).

By a second order signature Σ we mean an ordered pair $\Sigma = \langle \text{VAR}, \text{FUNC} \rangle$ where:

1. VAR is a set such that: (1) $1 \in$ VAR, $\langle 0, 1 \rangle \in$ VAR and (2) whenever $\alpha \in$ VAR then $\alpha = \langle 0, 1, \overset{n}{\ldots}, 1 \rangle$ with $n \geq 1$ or $\alpha = \langle 1, \overset{n+1}{\ldots}, 1 \rangle$.
   Besides that, whenever $\alpha \in$ VAR and $\alpha = \langle 0, 1, \overset{n}{\ldots}, 1 \rangle$ with $n > 1$ then also $\langle 0, 1, \overset{n-1}{\ldots}, 1 \rangle \in$VAR. And whenever $\beta \in$ VAR and $\beta = \langle 1, \overset{n}{\ldots}, 1 \rangle$ with $n > 1$, then also$\langle 1, \overset{n-1}{\ldots}, 1 \rangle \in$ VAR**.**

2. FUNC $=$ FUNC(Σ) is a function whose values are of these forms: $\langle 0, 1, \overset{n}{\ldots}, 1 \rangle$ with $n \geq 1$ or $\langle 1, \overset{n+1}{\ldots}, 1 \rangle$ with $n \geq 0$. We are using OPER.CONS as domain of FUNC and call its elements operation constants.

Explanation The set VAR contains the kinds of quantifiable variables, while in FUNC we obtain the type of each operation constant. 1 is the type of individuals, $\langle 0, 1, \overset{n}{\ldots}, 1 \rangle$ is the type of n-ary relations and $\langle 1, \overset{n+1}{\ldots}, 1 \rangle$ is the type of n-ary functions. (So zero-ary functions are safely identified with individuals.) Since in second order logic our variables are for sets and relations, the kinds of variables are also types. In the classical presentation of second order logic the function and relation constants are always among individuals and so the types of relation constants are types of certain variables as well.

Signature of the classes of languages SOL and $\lambda-$SOL

1. VAR $= \{1, \langle 0, 1 \rangle, \langle 0, 1, 1 \rangle, \langle 0, 1, 1, 1 \rangle, ...\}$

2. FUNC $=$ FUNC(Σ) is a function defined as in 2.1.1

Remark 1 *The set VAR of any language in SOL is formed by two disjoint sets corresponding to individual and relation types. Since we want to keep the formulas easy to read, we are not using the types as superscripts, instead we will use the more conventional treatment of first order logic: we will use just a number indicating the arity of the variable or relation constant.*

Alphabet of the $\lambda-$SOL language $\lambda - L_2$

The alphabet of a second order language of signature Σ contains all the operation constants in OPER.CONS, logical symbols and an infinite number of variables for each type $\alpha \in$ VAR. Besides that, it may contain the symbol λ.

In particular, our λ-language ($\lambda - L_2$) will have:

1. Connectives: $\neg$, $\vee, \wedge, \rightarrow, \leftrightarrow$
2. Quantifiers: $\forall$, $\exists$.
3. Abstractor: λ.
4. Parentheses: *),(*.
5. Equality symbols: E, E_1, $E_2, ...$ (for individuals and relations).
   They have types: $\langle 0,1,1\rangle$, $\langle 0,\langle 0,1\rangle,\langle 0,1\rangle\rangle, ..., \langle 0,\langle 0,1,\overset{n}{...},1\rangle,\langle 0,1,\overset{n}{...},1\rangle\rangle$, etc.
6. Falsity: $\perp$ (its type is 0).
7. A set $\mathcal{V}$ of individual variables: x, y, z, x_1, x_2, $x_n, ...$ (their type is 1)
   A set $\mathcal{V}_1$ of unary relation variables: X^1, Y^1, Z^1, X^1_1, X^1_2, $X^1_3, ...$ (their type is $\langle 0,1\rangle$.)
   A set $\mathcal{V}_2$ of binary relation variables: X^2, Y^2, Z^2, X^2_1, X^2_2, $X^2_3, ...$ (their type is $\langle 0,1,1\rangle$)
   and so on.
8. We will consider a countable infinite set, OPER.CONS, including:
   (a) Zero-ary function constants: a, b, c, c_1, c_2, $c_3, ...$ (their type is 1, as the type of individual variables)
   (b) Unary function constants: $f^{\ 1}$, g^1, h^1, f^1_1, f^1_2, $f^1_3, ...$ (they have a type $\langle 1,1\rangle$, which is not a member of VAR because in there we just put the types of quantifiable variables)
   (c) Binary function constants: $f^{\ 2}$, g^2, h^2, f^2_1, f^2_2, $f^2_3, ...$ (whose type is $\langle 1,1,1\rangle$)
   and so on.
   (d) Unary relation constants: R^1, S^1, T^1, R^1_1, R^1_2, $R^1_3, ...$ (of type $\langle 0,1\rangle$)
   (e) Binary relation constants: R^2, S^2, T^2, R^2_1, R^2_2, $R^2_3, ...$ (of type $\langle 0,1,1\rangle$)
   and so on.

Alphabet of the SOL language L_2

The alphabet L_2 is obtained from the alphabet of $\lambda - L_2$ by leaving out the λ-abstractor.

Expressions: terms, predicates and formulas

Now, from the set of finite strings of elements of the alphabet we are going to select the expressions of $\lambda - L_2$ and of L_2; that is, the predicates, formulas and terms of $\lambda - L_2$ and of L_2. The sets $\text{TERM}(L_2)$, $\text{TERM}(\lambda - L_2)$, $\text{PRED}(L_2)$, $\text{PRED}(\lambda - L_2)$, $\text{FORM}(L_2)$ and $\text{FORM}(\lambda - L_2)$ are defined in the expected way, giving rise to individual and relational quantification. They are the smallest sets obtained by the following inductive rules:

Terms

- (T1) Any individual variable x is a term.
- (T2) Any individual constant b is a term.
- (T3) If f is an n-ary function constant with arity $n \geq 1$ and $\tau_1, ..., \tau_n$ are terms, then $f\tau_1...\tau_n$ is a term.

$\text{TERM}(\lambda - L_2)$ is the smallest set obtained by these rules.
$\text{TERM}(L_2) = \text{TERM}(\lambda - L_2)$.

Predicates

- (P1) Any n-ary relation variable X^n is a predicate of degree n.
- (P2) Any n-ary relation constant P^n is a predicate of degree n.
- (P3) Also, E, E_1, $E_2, ...$ are predicates of degree 2.
- (P4) If ψ is any formula of $\lambda - L_2$ and $x_1, ..., x_n$ are pairwise distinct individual variables, then $\lambda x_1...x_n\psi$ is an n-ary predicate (where the inductive definition of $\text{FORM}(\lambda - L_2)$ is presented below).

$\text{PRED}(\lambda - L_2)$ is the smallest set obtained by these rules.
$\text{PRED}(L_2)$ is the smallest set obtained by rules (P1), (P2) and (P3).

Formulas

- (F1) If Π is an n-ary predicate and $\tau_1, ..., \tau_n$ are terms, then $\Pi\tau_1...\tau_n$ is a formula.
  In particular, $E\tau_1\tau_2$(written $\tau_1 = \tau_2$) is a formula when τ_1 and τ_2 are terms.
  $\perp$ *is* a formula.
- (F2) If Π^n and Ψ^n are n-ary predicates, then $E_n\Pi^n\Psi^n$ (written $\Pi^n = \Psi^n$) is a formula.
- (F3) If φ and ψ are formulas, then $\neg\varphi$, $(\varphi \vee \psi)$, $(\varphi \wedge \psi)$, $(\varphi \rightarrow \psi)$ and $(\varphi \leftrightarrow \psi)$ are formulas.

- (F4) If φ is a formula and x is any individual variable, then $\forall x\varphi$ and $\exists x\varphi$ are both formulas.
- (F5) If φ is a formula and X^n is any n-ary relation variable, then $\forall X^n\varphi$ and $\exists X^n\varphi$ are both formulas.

$\text{FORM}(\lambda - L_2)$ is the smallest set obtained by these rules.
$\text{FORM}(L_2)$ is the smallest set obtained by these rules.

Expressions

The set union of the sets $\text{TERM}(\lambda - L_2)$, $\text{PRED}(\lambda - L_2)$ and $\text{FORM}(\lambda - L_2)$ is the set of expressions of $\lambda - L_2$, denoted by $\text{EXPR}(\lambda - L_2)$.

The set union of the sets $\text{TERM}(L_2)$, $\text{PRED}(L_2)$ and $\text{FORM}(L_2)$ is the set of expressions of L_2, denoted by $\text{EXPR}(L_2)$. A member of any of them is an expression, denoted by ϵ.

Remark 2 *Please recall that for a language without λ we suppress the rule (P4) of the definition of Predicates and so the rules (F1) throught (F5) in the definition of Formulas are weakened since we then lack λ-predicates.*

Example 3 *We have mentioned the comprehension schema. Here are some easy instances of it:*

1. $\exists Z\forall x(Zx \leftrightarrow \neg Rx)$. *This says that there exists the complement of the set denoted by R.*
2. $\exists Z\forall x(Zx \leftrightarrow (Rx \wedge \neg Sx))$. *This says that there exists the set difference of sets denoted by R and S.*
3. $\exists Z\forall x(Zx \leftrightarrow \exists y\ Rxy)$. *The domain of the relation denoted by R exists.*
4. $\exists Z\forall x(Zx \leftrightarrow x = x)$. *The universe exists. In the* Equality-free *SOL language the equality sign should be replaced by its definition, write $\forall X(Xx \leftrightarrow Xx)$ instead of $x = x$.*
5. $\exists Z^2\forall x\forall y(Z^2xy \leftrightarrow R^2yx)$. *The reciprocal of a given relation exists.*
6. *The restriction on the variable Z^n cannot be harmlessly dropped. Otherwise we would have monsters like this formula $\exists Z\forall x(Zx \leftrightarrow \neg Zx)$.*
7. *In $\lambda - SOL$ the comprehension axiom can be simply expressed as*

$$\forall x_1...x_n(\lambda x_1...x_n\varphi\ x_1...x_n \leftrightarrow \varphi)$$

Cardinality of a language As usual, the cardinality of a language is the cardinality of the set of its formulas. Therefore, $\text{CARD}(L_2) = \text{CARD}(\lambda - L_2) = |\text{FORM}(L_2)| = |\text{FORM}(\lambda - L_2)| = \aleph_0$.

1.2 Standard Semantics

Definition of standard structures

The second order languages we have introduced are both designed to talk about structures of signature $\Sigma = \langle \text{VAR, FUNC} \rangle$.

Standard structures of signature Σ are ordered tuples

$$\mathcal{A} = \langle \mathbf{A}, \langle \mathbf{A}_n \rangle_{n \geq 1}, \langle C^{\mathcal{A}} \rangle_{C \in \text{OPER.CONS}} \rangle$$

such that:

1. $\mathbf{A} \neq \emptyset$, the universe of individuals is a non-empty set.

2. For each $n \geq 1$, $\mathbf{A}_n = \wp \mathbf{A}^n$; that is, the universe of n-ary relations is the power set of the n-ary cartesian product of $\mathbf{A}$. Therefore, the n-ary relational universe contains all possible n-ary relations on $\mathbf{A}$.

3. For each n-ary relation constant, $R \in$ OPER.CONS and $\text{FUNC}(R) = \langle 0, 1, \overset{n}{\ldots}, 1 \rangle$, $R^{\mathcal{A}}$ is an n-ary relation on individuals. That is,

$$R^{\mathcal{A}} \subseteq \mathbf{A} \times \overset{n}{\ldots} \times \mathbf{A}$$

   (Thus, $R^{\mathcal{A}}$ is a subset of the n-ary cartesian product of $\mathbf{A}$, an element of $\wp \mathbf{A}^n$.)

4. For each n-ary function constant, $f \in$OPER.CONS and $\text{FUNC}(f) = \langle 1, \overset{n+1}{\ldots}, 1 \rangle$, $f^{\mathcal{A}}$ is an n-ary function on individuals. That is,

$$f^{\mathcal{A}} : \mathbf{A} \times \overset{n}{\ldots} \times \mathbf{A} \longrightarrow \mathbf{A}$$

   For individual constants the condition reduces to belonging to $\mathbf{A}$.

Assignment

In second order logic an *assignment* is a mapping from the set of variables into the universes

$$M : \mathcal{V} \cup (\bigcup_{n \geq 1} \mathcal{V}_n) \longrightarrow \mathbf{A} \cup (\bigcup_{n \geq 1} \mathbf{A}_n)$$

in such a way that $M[\mathcal{V}] \subseteq \mathbf{A}$ and $M[\mathcal{V}_n] \subseteq \mathbf{A}_n$ for each n; that is to say that n-ary relation variables always get their value in $\mathbf{A}_n$.

We define $M^{\mathbf{v}_1 \ldots \mathbf{v}_n}_{v_1 \ldots v_n}$ for the pairwise distinct variables $v_1, \ldots, v_n$ of any type and the individuals and relations $\mathbf{v}_1, \ldots, \mathbf{v}_n$ of $\mathcal{A}$ of the same type as the variable of equal subscript.

$$M^{\mathbf{v}_1 \ldots \mathbf{v}_n}_{v_1 \ldots v_n} = (M - \{\langle v_1, M(v_1) \rangle, \ldots, \langle v_n, M(v_n) \rangle\}) \cup \{\langle v_1, \mathbf{v}_1 \rangle, \ldots, \langle v_n, \mathbf{v}_n \rangle\}.$$

Interpretation

An interpretation $\mathcal{I}$ over a standard structure $\mathcal{A}$ is a pair $\langle \mathcal{A}, M\rangle$ where M is an assignment on $\mathcal{A}$. If $\mathcal{I} = \langle \mathcal{A}, M\rangle$ then we will write $\mathcal{I}^{\mathbf{v}_1..\mathbf{v}_n}_{v_1..v_n}$ instead of $\langle \mathcal{A}, M^{\mathbf{v}_1...\mathbf{v}_n}_{v_1...v_n}\rangle$.

Definition of denotation of terms and predicates and of satisfaction of formulas of $\lambda - L_2$

Let $\mathcal{I} = \langle \mathcal{A}, M\rangle$.

- (T1) $\mathcal{I}(x) = M(x)$
- (T2) $\mathcal{I}(a) = a^{\mathcal{A}}$
- (T3) $\mathcal{I}(f\tau_1...\tau_n) = f^{\mathcal{A}}(\mathcal{I}(\tau_1)...\mathcal{I}(\tau_n))$
- (P1) $\mathcal{I}(X^n) = M(X^n)$
- (P2) $\mathcal{I}(R) = R^{\mathcal{A}}$
- (P3) $\mathcal{I}(=) = \{\langle x, y\rangle \in A^2 | x = y\} \cup (\bigcup_{n\geq 1}\{\langle X^n, Y^n\rangle \in A^2_n | X^n = Y^n\})$ (equality is identity in every universe)
- (P4) $\mathcal{I}(\lambda x_1...x_n\varphi) = \{\langle \mathbf{x}_1, ..., \mathbf{x}_n\rangle \in \mathbf{A}\times\overset{n}{...}\times\mathbf{A} \mid \mathcal{I}^{\mathbf{x}_1...\mathbf{x}_n}_{x_1...x_n} \text{ sat } \varphi\}$
- (F1) $\mathcal{I}$ sat $\Pi\ \tau_1...\tau_n$ iff $\langle \mathcal{I}(\tau_1), ..., \mathcal{I}(\tau_n)\rangle \in \mathcal{I}(\Pi)$
  In particular, $\mathcal{I}$ sat $\tau_1 = \tau_2$ iff $\mathcal{I}(\tau_1) = \mathcal{I}(\tau_2)$.
  Also, it is not true that $\mathcal{I}$ *sat* $\perp$
- (F2) $\mathcal{I}$ sat $\Pi^n = \Psi^n$ iff $\mathcal{I}(\Pi^n) = \mathcal{I}(\Psi^n)$
- (F3) $\mathcal{I}$ sat $\neg\varphi$ iff it is not true that $\mathcal{I}$ sat φ
  - $\mathcal{I}$ sat $\varphi \vee \psi$ iff $\mathcal{I}$ sat φ or $\mathcal{I}$ sat ψ
  - $\mathcal{I}$ sat $\varphi \wedge \psi$ iff $\mathcal{I}$ sat φ and $\mathcal{I}$ sat ψ
  - $\mathcal{I}$ sat $\varphi \rightarrow \psi$ iff if $\mathcal{I}$ sat φ then $\mathcal{I}$ sat ψ
  - $\mathcal{I}$ sat $\varphi \leftrightarrow \psi$ iff $\mathcal{I}$ sat φ if and only if $\mathcal{I}$ sat ψ
- (F4) $\mathcal{I}$ sat $\forall x\varphi$ iff for all $x \in \mathbf{A} : \mathcal{I}^{\mathbf{x}}_x$ sat φ
  - $\mathcal{I}$ sat $\exists x\varphi$ iff there is $x \in \mathbf{A}$ such that: $\mathcal{I}^{\mathbf{x}}_x$ sat φ
- (F5) $\mathcal{I}$ sat $\forall X^n\varphi$ iff for all $\mathbf{X}^n \in \mathbf{A}_n : \mathcal{I}^{\mathbf{X}^n}_{X^n}$ sat φ
  - $\mathcal{I}$ sat $\exists X^n\varphi$ iff there is $\mathbf{X}^n \in \mathbf{A}_n$ such that: $\mathcal{I}^{\mathbf{X}^n}_{X^n}$ *sat* φ

Consequence and validity

- A formula φ is a *consequence* of a set of formulas Γ if and only if each interpretation $\mathcal{I}$ which happens to be a model of Γ is also a model of φ. Therefore, φ is a consequence of Γ, if and only if, for every structure $\mathcal{A}$ of signature Σ and every assignment M on its universes, if $\langle \mathcal{A}, M\rangle$ is a model of every $\psi \in \Gamma$, then $\langle \mathcal{A}, M\rangle$ is also a model of φ.

- We will write $\Gamma \models \varphi$ to indicate that φ is a consequence of Γ.

- We will write $\Gamma \not\models \varphi$ to indicate that φ is not a consequence of Γ. In this case we also say that φ is *independent* of Γ.

Definable sets and relations in a given structure

Definition of first order relations of the structure $\mathcal{A}$ If $\mathcal{A}$ is a second order structure, any subset of the n-ary cartesian product of the universe of individuals—i.e., any $\mathbf{X} \subseteq \mathbf{A}^n$— is an n-ary *first order relation on* $\mathcal{A}$.
Let $\text{REL}^{1st}(\mathcal{A})$ be the class of all first order relations on $\mathcal{A}$.

- We say that an n-ary first order relation X is *into* $\mathcal{A}$ when $\mathbf{X} \in \mathbf{A}_n$ or $\mathbf{X} = R^{\mathcal{A}}$ for $R \in$ OPER.CONS.
  Let $\text{REL}^{1st}(\in \mathcal{A})$ be the class of all first order relations into $\mathcal{A}$.

Definition of second order relations of $\mathcal{A}$

- If $\mathcal{A}$ is a second order structure, $\mathbf{A}_{i_l}, ..., \mathbf{A}_{i_n}$ are universes of $\mathcal{A}$ (either individual or relation universes; $\mathbf{A}_{i_j} = \mathbf{A}$ or $\mathbf{A}_{i_j} = \mathbf{A}_m$, with $m \geq 1$), any subset of the cartesian product of these universes $-i.e.$, any $\mathbf{X} \subseteq \mathbf{A}_{i_l} \times ... \times \mathbf{A}_{i_n} -$ *is* an n-ary *second order relation of* $\mathcal{A}$.
  Let $\text{REL}(\mathcal{A})$ be the class of all second order relations of $\mathcal{A}$.

- We say that an n-ary second order relation X is *into* $\mathcal{A}$ when $\mathbf{X} \in \mathbf{A}_n$ or $\mathbf{X} = R^{\mathcal{A}}$ for $R \in$OPER.CONS.
  Let $\text{REL}(\in \mathcal{A})$ be the class of all second order relations into $\mathcal{A}$.

- We can define the set of *proper second order relations* as

$$\text{REL}^{2^{nd}}(\mathcal{A}) = \text{REL}(\mathcal{A}) - \text{REL}^{1^{st}}(\mathcal{A}).$$

Definition of $\mathcal{A}$-definable first order relations using a given language

- Let $\mathcal{A}$ be a second order structure and $\mathbf{X} \subseteq \mathbf{A}^n$ an n-ary relation on individuals. We say that X is an *$\mathcal{A}$-definable first order relation using* $\lambda - L_2$ iff there exist a formula φ of $\lambda - L_2$, pairwise distinct individual variables $x_1, ..., x_n$ such that $\text{FREE}(\lambda x_1 ... x_n \varphi) = \emptyset$ and an interpretation $\mathcal{I}$ over $\mathcal{A}$ such that

$$\mathbf{X} = \mathcal{I}(\lambda x_1 ... x_n \varphi)$$

Let $\text{DEF}^{1^{st}}(\mathcal{A}, \lambda - L_2)$ be the smallest class containing all $\mathcal{A}$-definable first order relations using $\lambda - L_2$.

- For the SOL language without λ-abstraction, L_2, we say that the formula φ along with the sequence of individual variables $\langle x_1, ..., x_n \rangle$ *defines* X *in the structure* $\mathcal{A}$ *using* L_2 when

$$\mathbf{X} = \{\langle \mathbf{x}_1, ..., \mathbf{x}_n \rangle | \langle \mathcal{A}, M^{\mathbf{x}_1 ... \mathbf{x}_n}_{x_1 ... \, x_n} \rangle \text{ sat } \varphi\}$$

where M is an assignment and $\text{FREE}(\varphi) \subseteq \{x_1, ..., x_n\}$.
Let $\text{DEF}^{1^{st}}(\mathcal{A}, L_2)$ be the smallest class containing all $\mathcal{A}$-definable first order relations using L_2.

Definition of $\mathcal{A}$-definable second order relations using a given language
$\mathbf{X}$ is an $\mathcal{A}$-*definable second order relation using a language* $\lambda - L_2$ (or L_2) when $\mathbf{X}$ is a second order relation of $\mathcal{A}$ and there is a formula φ of $\lambda - L_2$ (or of L_2) such that

$$\mathbf{X} = \{\langle \mathbf{v}_1, ..., \mathbf{v}_n \rangle \in \mathbf{A}_{i_l} \times ... \times \mathbf{A}_{i_n} | \ \mathcal{A}[^{\mathbf{v}_1 ... \mathbf{v}_n}_{v_1 ... v_n}] \text{ sat } \varphi\}$$

where $\text{FREE}(\varphi) \subseteq \{v_1, ..., v_n\}$ and for each j, $1 \leq j \leq n$, the variable v_j is of type $i_j \in VAR$.

Let $\text{DEF}(\mathcal{A}, L_2)$ and $\text{DEF}(\mathcal{A}, \lambda - L_2)$ be the corresponding sets.

Definition of first and second order parametrically $\mathcal{A}$-definable relations using a given language

- $\mathbf{X}$ is a *parametrically* ***$\mathcal{A}$-definable first order relation using*** $\lambda - L_2$ (or L_2) iff there are a formula φ, individual variables $x_1, ..., x_n$ and variables $v_1, ..., v_m$ of any types $i_1, ..., i_m \in VAR$ such that

$$X = \{\langle \mathbf{x}_1, ..., \mathbf{x}_n \rangle \in \mathbf{A}^n | \ \mathcal{A}[^{\mathbf{x}_1 ... \mathbf{x}_n \mathbf{v}_1 ... \mathbf{v}_m}_{x_1 ... x_n v_1 ... v_m}] \text{ sat } \varphi\}$$

where $\text{FREE}(\varphi) \subseteq \{x_1, ..., x_n, v_1, ..., v_m\}$ and the parameters $v_1, ..., v_m$ are in $\mathbf{A}_{i_l}, ..., \mathbf{A}_{i_m}$ of types $i_1, ..., i_m$.

- $\mathbf{X}$ is a *parametrically* ***$\mathcal{A}$-definable second order relation using*** $\lambda - L_2$ (or L_2) iff there are a formula φ, variables $u_1, ..., u_n$ and variables $v_1, ..., v_m$ of any types $k_1, ..., k_n, i_1, ..., i_m \in VAR$ such that

$$\mathbf{X} = \{\langle \mathbf{u}_1, ..., \mathbf{u}_n \rangle \in \mathbf{A}_{k_l} \times ... \times \mathbf{A}_{k_n} | \ \mathcal{A}[^{\mathbf{u}_1 ... \mathbf{u}_n \mathbf{v}_1 ... \mathbf{v}_m}_{u_1 ... u_n v_1 ... v_m}] \text{ sat } \varphi\}$$

where $\text{FREE}(\varphi) \subseteq \{u_1, ..., u_n, v_1, ..., v_m\}$ and the parameters $\mathbf{v}_1, ..., \mathbf{v}_m$, are in $\mathbf{A}_{i_l}, ..., \mathbf{A}_{i_m}$ of types $i_1, ..., i_m$.
Let $\text{PARAM.DEF}^{1^{st}}(\mathcal{A}, L_2)$, $\text{PARAM.DEF}^{1^{st}}(\mathcal{A}, \lambda - L_2)$, $\text{PARAM.DEF}(\mathcal{A}, L_2)$ and $\text{PARAM.DEF}(\mathcal{A}, \lambda - L_2)$ be the corresponding sets.

More about the expressive power of standard SOL

Axiomatizable property of the domain

- A property $\mathcal{P}$ of the domain of individuals is *axiomatizable* if and only if there is a recursive set of sentences Γ of *Pure* SOL such that $\mathcal{A}$ is a model of Γ iff $\mathbf{A}$ has the property $\mathcal{P}$, for every standard structure $\mathcal{A}$.

- A property $\mathcal{P}$ of the domain of individuals is *finitely axiomatizable* if and only if there is a sentence φ of *Pure* SOL such that $\mathcal{A}$ is a model of φ iff $\mathbf{A}$ has the property $\mathcal{P}$, for every standard structure $\mathcal{A}$.

Global relation

- A global n-ary relation $\mathbf{R}$ on "the" mathematical universe is *first order and globally definable* if and only if there is a SOL formula φ with $\text{FREE}(\varphi) \subseteq \{x_1, ..., x_n\}$ such that for every structure $\mathcal{A}$,

$$\{\langle \mathbf{x}_1, ..., \mathbf{x}_n \rangle | \ \mathcal{A}[^{\mathbf{x}_1 ... \mathbf{x}_n}_{x_1 ... x_n}] \text{ sat } \varphi\} = \mathbf{R} \cap \mathbf{A}^n$$

  In $\lambda - SOL$ we express the condition by saying: $\mathcal{A}(\lambda x_1 ... x_n \varphi) = \mathbf{R} \cap \mathbf{A}^n$.

- A global n-ary relation $\mathbf{R}$ on "the" mathematical universe is *second order and globally definable* if and only if there is a SOL formula φ with $\text{FREE}(\varphi) \subseteq \{v_1, ..., v_n\}$ of types $i_1, ..., i_n$ such that for every structure $\mathcal{A}$,

$$\{\langle \mathbf{v}_1, ..., \mathbf{v}_n \rangle \in \mathbf{A}_{i_l} \times ... \times \mathbf{A}_{i_n} | \ \mathcal{A}[^{\mathbf{v}_1 ... \mathbf{v}_n}_{v_1 ... v_n}] \text{ sat } \ \varphi\} = \mathbf{R} \cap (\mathbf{A}_{i_l} \times ... \times \mathbf{A}_{i_n})$$

Categorically axiomatizable structure

- A given structure $\mathcal{B}$ is *categorically axiomatizable* if and only if there is a recursive set of sentences Γ such that $\mathcal{A}$ is a model of Γ iff $\mathcal{B} \cong \mathcal{A}$.

- A given structure $\mathcal{B}$ is *categorically finitely axiomatizable* if and only if there is a sentence φ such that $\mathcal{A}$ is a model of φ iff $\mathcal{B} \cong \mathcal{A}$.

Mathematical universe

A property $\mathcal{P}$ of "the" mathematical universe is *axiomatizable* if and only if there is a sentence φ of *Pure* SOL such that, φ is valid iff $\mathcal{P}$ holds in "the" mathematical universe.

Negative results

Due to the great expressive power of SOL it is easy to demonstrate that some very classical first order theorems do not hold in SOL. We are going to prove this here for compactness and Löwenheim-Skolem theorems. The weak and strong incompleteness theorems are proved in the next chapter.

- *The compactness theorem does not hold for* SOL**.**

- *The strong completeness theorem does not hold for* SOL *with a sound calculus.*
- *The Löwenheim-Skolem theorem does not hold for* SOL**.**

1.3 Semantic Theorems

Coincidence lemma

Let $\mathcal{A}$ be a second order structure, M_1 and M_2 two assignments on $\mathcal{A}$ and let $\mathcal{I}_1 = \langle \mathcal{A}, M_1 \rangle$ and $\mathcal{I}_2 = \langle \mathcal{A}, M_2 \rangle$ be the corresponding interpretations. Then:

1. For any term τ such that $M_1 \restriction \mathrm{FREE}(\tau) = M_2 \restriction \mathrm{FREE}(\tau)$, we have $\mathcal{I}_1(\tau) = \mathcal{I}_2(\tau)$.
2. For any predicate Π^n such that $M_1 \restriction \mathrm{FREE}(\Pi^n) = M_2 \restriction \mathrm{FREE}(\Pi^n)$, we have
$$\mathcal{I}_1(\Pi^n) = \mathcal{I}_2(\Pi^n)$$
3. For any formula φ such that $M_1 \restriction \mathrm{FREE}(\varphi) = M_2 \restriction \mathrm{FREE}(\varphi)$, we have
$$\mathcal{I}_1 \text{ sat } \varphi \text{ iff } \quad \mathcal{I}_2 \text{ sat } \varphi$$

Substitution lemma

Substitution lemma for individuals Let x be an individual variable and τ a term. Given a second order structure $\mathcal{A}$ and assignment M on $\mathcal{A}$,

1. For every term $t : \langle \mathcal{A}, M_x^{\langle \mathcal{A}, M \rangle(\tau)} \rangle(t) = \langle \mathcal{A}, M \rangle(t\frac{\tau}{x})$
2. For every n-ary predicate $\Pi : \langle \mathcal{A}, M_x^{\langle \mathcal{A}, M \rangle(\tau)} \rangle(\Pi) = \langle \mathcal{A}, M \rangle(\Pi\frac{\tau}{x})$
3. For every formula $\varphi : \langle \mathcal{A}, M_x^{\langle \mathcal{A}, M \rangle(\tau)} \rangle$ sat φ iff $\langle \mathcal{A}, M \rangle$ sat $\varphi\frac{\tau}{x}$

Isomorphism theorem

Let $\mathcal{A} \overset{\mathbf{h}}{\cong} \mathcal{B}$ and extend the function $\mathbf{h}$ so as to cover all the universes.
(That is, define:

$$\begin{array}{rrcl} \bar{\mathbf{h}}: & \mathbf{A} \cup (\bigcup_{n>0} \mathbf{A}_n) & \longrightarrow & \mathbf{B} \cup (\bigcup_{n>0} \mathbf{B}_n) \\ & \mathbf{x} & \longmapsto & \mathbf{h}(\mathbf{x}) \\ & \mathbf{X}^n & \longmapsto & \{\langle \mathbf{h}(\mathbf{x}_1), ..., h(\mathbf{x}_n) \rangle \mid \langle \mathbf{x}_1, ..., \mathbf{x}_n \rangle \in \mathbf{X}^n\} \end{array}$$

Then

1. For every term $t : \bar{\mathbf{h}}(\langle \mathcal{A}, M \rangle(t)) = \langle \mathcal{B}, \bar{\mathbf{h}} \circ M \rangle(t)$
2. For every predicate $\Pi^n : \bar{\mathbf{h}}(\langle \mathcal{A}, M \rangle(\Pi^n)) = \langle \mathcal{B}, \bar{\mathbf{h}} \circ M \rangle(\Pi^n)$
3. For every formula $\varphi : \langle \mathcal{A}, M \rangle$ sat φ iff $\langle \mathcal{B}, \bar{\mathbf{h}} \circ M \rangle$ sat (φ)

Chapter 2

DEDUCTIVE CALCULI

2.1 Sequent calculi

Very briefly we will present three sequent calculi for second order logic, two of them are equivalent in deductive power.

All three are based on the sequent calculus of Ebbinghaus, Flum & Thomas's **Mathematical Logic** [1984]; it has been enlarged with the extensionality axiom, with quantifier rules for the new quantification and with either λ-rules, or comprehension rule.

Deductions

A deduction is a finite non-empty sequence of lines, each of which is a finite non-empty sequence of formulas $\varphi_1 ... \varphi_n \psi$ which is called a *sequent* whose *antecedent* is the sequence $\varphi_1 ... \varphi_n$ and whose *consequent* is ψ. In the following we will use the letters Ω and Θ for sequences of formulas (possible empty). We can also write sequences in the form $\Theta\ \varphi$. For a sequent we will use the notation $\Theta \hookrightarrow \psi$, where Θ is $\varphi_1 ... \varphi_n$. Θ is the antecedent and ψ the consequent.

Sequent rules allow us to pass from one line to another and they are formulated in such a way that in each line the consequent is a logical consequence of the antecedent; *i.e.*, if $\Theta \hookrightarrow \psi$ is a line in a deduction, then $\Theta \models \psi$. To be more precise,

if $\Theta \hookrightarrow \psi$ is a line in a deduction, then $\{\varphi \mid \varphi$ is a member of $\Theta\} \models \psi$

The rules are the following:

Sequent rules common to all calculi

We introduce below the rules for calculus $\lambda - C_2$, C_2 and C_2^-. We will use the $\lambda - C_2$ calculus with the language $\lambda - L_2$, and the calculus C_2 and its subcalculus C_2^- with the λ-free language L_2.

(HI) Hypothesis introduction

$\dfrac{}{\Omega \hookrightarrow \varphi}$, when $\varphi \in \Omega$

(M) Monotony

$\dfrac{\Omega \hookrightarrow \varphi}{\Theta \hookrightarrow \varphi}$, when all the formulas in Ω are also in Θ.

(i.e., when $\Omega \subseteq \Theta$, if we identify a sequent, Ω or Θ, with the set of its members.)

(PC) Proof by cases

$$\dfrac{\begin{array}{ll}\Omega & \psi \hookrightarrow \varphi \\ \Omega & \neg\psi \hookrightarrow \varphi\end{array}}{\Omega \quad \hookrightarrow \varphi}$$

(NC) Non contradiction

$$\dfrac{\begin{array}{ll}\Omega & \neg\varphi \hookrightarrow \psi \\ \Omega & \neg\varphi \hookrightarrow \neg\psi\end{array}}{\Omega \quad \hookrightarrow \varphi}$$

(IDA) Introducing disjunction in the antecedent

$$\dfrac{\begin{array}{lll}\Omega & \varphi & \hookrightarrow \gamma \\ \Omega & \psi & \hookrightarrow \gamma\end{array}}{\Omega\ \varphi \vee \psi \hookrightarrow \gamma}$$

(IDC) Introducing disjunction in the consequent

$\dfrac{\Omega \hookrightarrow \varphi}{\Omega \hookrightarrow \varphi \vee \psi}$ and $\dfrac{\Omega \hookrightarrow \varphi}{\Omega \hookrightarrow \psi \vee \varphi}$

(IPA) Introducing individual particularization in the antecedent

$\dfrac{\Omega \ \ \varphi\frac{y}{x} \hookrightarrow \psi}{\Omega \ \ \exists x\varphi \hookrightarrow \psi}$, $y \notin \mathrm{FREE}(\Omega \cup \{\exists x\varphi,\, \psi\})$, x and y are individual variables.

(IPC) Introducing individual particularization in the consequent

$\dfrac{\Omega \hookrightarrow \varphi\frac{\tau}{x}}{\Omega \hookrightarrow \exists x\varphi}$ where x and τ are a variable and a term

(RE) Reflexivity of equality for individuals

$\dfrac{}{\hookrightarrow \tau = \tau}$

(ES) Equals substitution for individuals

$$\frac{\Omega \quad \hookrightarrow \varphi\frac{\tau}{x}}{\Omega\ \tau = t \hookrightarrow \varphi\frac{t}{x}}$$

All the previous rules are also first order rules. Now we extend the calculus to cover relational quantification.

(IPA)^n Introducing relation quantification in the antecedent

$$\frac{\Omega\ \ \varphi\frac{Y^n}{X^n} \hookrightarrow \psi}{\Omega\ \ \exists X^n\varphi \hookrightarrow \psi}$$

(where X^n and Y^n are n-ary relation variables $Y^n \notin \text{FREE}(\Omega \cup \{\exists X^n\varphi, \psi\})$

(I.P.C)^n Introducing Relation Quantification in the Consequence

$$\frac{\Omega \hookrightarrow \varphi\frac{\Pi^n}{X^n}}{\Omega \hookrightarrow \exists X^n\varphi}$$

(where Π^n is an n-ary predicate and X^n an n-ary relation variable)

(Ext.) Extensionality

$$\frac{}{\hookrightarrow \forall X^n Y^n (X^n = Y^n \longleftrightarrow \forall x_1 ... x_n (X^n x_1 ... x_n \longleftrightarrow Y^n x_1 ... x_n))}$$

Let's call C_2^- the calculus having only these rules:
(HI), (M), (PC), (NC), (IDA), (IDC), (IPA), (IPC), (IPA)^n, (IPC)^n, (RE), (ES), (Ext.).

Sequent rules for calculus C_2

Nevertheless, one of the capabilities of second order logic we have mentioned many times is the definition of relations. We can do two things, we add either a comprehension schema or abstraction rules. The simpler one is to add to C_2^- the comprehension schema as a rule without hypothesis,

(CS) Comprehension schema

$$\frac{}{\hookrightarrow \exists X^n \forall x_1 ... x_n (X^n x_1 ... x_n \longleftrightarrow \varphi)} \text{ where } X^n \notin \text{FREE}(\varphi)$$

and call the resulting calculus C_2.

Sequent rules for a calculus including lambda

An equivalent formulation of C_2, but for a language with λ-abstraction, is the result of adding to the calculus C_2^- the rules for introducing the abstractor λ in the antecedent and in the consequent. As noted above, while using the language $\lambda - L_2$, the rules (IPC)^n and (IPA)^n for relation quantification are also more powerful, even though their formulation remains unchanged.

(IAC) Introducing abstraction in the consequent

$$\frac{\Omega \hookrightarrow \varphi\dfrac{\tau_1...\tau_n}{x_1...x_n}}{\Omega \hookrightarrow \lambda x_1...x_n\varphi\tau_1...\tau_n} \quad (x_1, ..., x_n \text{ are pairwise distinct individual variables})$$

(IAA) Introducing abstraction in the antecedent

$$\frac{\Omega\ \ \varphi\dfrac{\tau_1...\tau_n}{x_1...x_n} \qquad \hookrightarrow \psi}{\Omega\ \ \lambda x_1...x_n\varphi\tau_1...\tau_n \hookrightarrow \psi} \quad (x_1, ..., x_n \text{ are pairwise distinct individual variables})$$

Let us call this calculus $\lambda - C_2$.

Deductions

1. If, according to the C_2^- calculus rules, we obtain as a line the sequent $\Omega \hookrightarrow \varphi$ of formulas of L_2, we say that it *is derivable in* C_2^- and we write $\vdash_{C_2^-} \Omega\ \varphi$.

2. If, according to the C_2 calculus rules, we obtain as a line the sequent $\Omega \hookrightarrow \varphi$ of formulas of L_2, we say that it *is derivable in* C_2 and we write $\vdash_{C_2} \Omega\ \varphi$.

3. If, according to the $\lambda - C_2$ calculus rules, we obtain as a line the sequent $\Omega \hookrightarrow \varphi$ of formulas of $\lambda - L_2$, we say that it *is derivable in* $\lambda - C_2$ and we write $\vdash_{\lambda - C_2} \Omega\ \varphi$.

4. More generally, a formula φ *is derivable from a set* Δ of formulas in our C_2^- calculus (or in the C_2 or in the $\lambda - C_2$ calculus) if and only if there is a finite subset of Δ, $\{\varphi_1, \varphi_2, ..., \varphi_n\}$, such that $\vdash_{C_2^-} \varphi_1\varphi_2...\varphi_n\varphi$ (or, such that $\vdash_{C_2} \varphi_1\varphi_2...\varphi_n\varphi$; or, such that $\vdash_{\lambda - C_2} \varphi_1\varphi_2...\varphi_n\varphi$). We will write $\Delta \vdash_{C_2^-} \varphi$. (Or, we will write $\Delta \vdash_{C_2} \varphi$; or, we will write $\Delta \vdash_{\lambda - C_2} \varphi$.)

2.2 Soundness Theorem in Standard Semantics

The soundness of rules common to all three calculi

I will prove the correctness for the language $\lambda - L_2$, since from it follows the correctness of the language without λ. I will use the shorter notation $\models$ (instead of$\models_{\mathcal{S}.\mathcal{S}}$) for *standard semantic* notions. Besides that, to simplify the notation, we identify the sequence Ω with the set of its members, $\{\gamma \mid \gamma \text{ is in } \Omega\}$.

2.3 Incompletenss in Standard Structures

All second order calculi are incomplete while using standard semantics. The incompleteness is in both senses: weak and strong. We will prove here the

incompleteness of C_2^- in the weak sense and also the incompleteness of any second order calculus in both senses.

Incompleteness of C_2^- in standard structures

Theorem There is a formula γ of L_2 such that $\models_{\mathcal{S}.\mathcal{S}} \gamma$ but not $\vdash_{\mathbf{C_2^-}} \gamma$.

Strong incompleteness of second order logic in standard structures

Theorem In L_2 there are a set of sentences Γ and a sentence γ such that $\Gamma \models_{\mathcal{S}.\mathcal{S}} \gamma$ but it is not the case that $\Gamma \vdash \gamma$ in any of our second order calculi.

Weak incompleteness of second order logic in standard structures

There are several well known incompleteness proofs in the literature. In what follows I will sketch another incompleteness proof based on Sain [1979]. This proof also uses the expressive power of standard second order logic.

General idea of the proof The original idea of this proof is that we can provide a set-theoretic development of second order logic; that is, in the mathematical universe of set theory we can find substitutes for the SOL formulas and so — since our formulas are treated as sets, we can talk about our formulas in the language of set theory, $L^{\in}$, exactly as we do with any other sets. In fact, we can use that the set of formulas of second order logic is definable in the math universe using the language of set theory. Also the concept of standard structure and the concept of validity can be defined set-theoretically. One can give a natural set-theoretical description of the notions of sequent and derivation, developing the whole syntax set-theoretically. We can easily agree that the mathematical resources needed for this task aren't very deep and the set-theoretical formulas used to define provability are of a very simple character, since they are what are usually termed persistent formulas. Finally, even the completeness theorem of second order logic can be formalized as a sentence of $L^{\in}$. If it were complete, the mathematical proof of completeness would have to be established with limited resources, i.e., in the axiomatic basis of set theory (ZFC). In this proof these properties are taken as facts, namely, as initial assumptions from where we will derive incompleteness.

Then, using the known theorems of Cohen and Gödel, we can obtain the incompleteness of SOL from the fact that the generalized continuum hypothesis is expressible in SOL. That means that there is a formula φ_{GCH} such that this formula is valid in the standard semantics if and only if GCH holds in the mathematical universe we are assuming we are in when defining the semantic for SOL. Thus, when Cohen's model is considered as our enviroment, φ_{GCH} cannot be valid, but φ_{GCH} is valid when Gödel's model is taken as our enviroment. Of course, that the second order formula expressing the generalized continuum

hypothesis is valid in one environment and not valid in another is not a contradiction, but nevertheless it is telling you already that something is getting wrong with the standard semantics.

It is important to see that the real contradiction we will obtain is that φ_{GCH} must be valid and must not be valid in the "same" mathematical enviroment. To obtain this contradiction the completeness of second order logic is assumed.

Let us term CAL the second order calculus of your choice; it could be C_2, $\lambda - C_2$ or any other of your preference.

The substance of the proof rest on the unbalanced relation between the set-theoretical formulas expressing $\vdash_{CAL}\ \varphi_{GCH}$ and $\models_{\mathcal{S}.\mathcal{S}}\ \varphi_{GCH}$. While the first one is a persistent formula whose truth value does not change when we go up from a transitive class of sets to a wider one, the second one does change since the meaning of φ_{GCH} is involved.

Thus, the $L^{\in}$ formulas expressing provability and validity of this formula φ_{GCH} cannot be equivalent, as required by the completeness theorem.

Summarizing, $\vdash_{CAL}\ \varphi_{GCH}$ is persistent while $\models_{\mathcal{S}.\mathcal{S}}\ \varphi_{GCH}$ is not, hence they can not coincide.

Incompleteness theorem No sound calculus of SOL can be proved to be complete for the standard models of second order logic (assuming that ZFC is consistent).

Proof

Assume that ZFC is consistent and that we have a sound calculus CAL for SOL. Accept also the initial assumptions mentioned in the previous section. By the theorem of Cohen, there is a model $\mathcal{U} = \langle \mathcal{W}, \in^{\mathcal{U}} \rangle$ of $ZFC + \neg GCH$. That is, a model of set theory where the generalized continuum hypothesis does not hold.

Let $\mathcal{L} \subseteq \mathcal{W}$ be the constructible part of it. Then, by the theorem Gödel,

$$\mathcal{L} = \langle \mathcal{L}, \in^{\mathcal{L}} \rangle \text{ is a model of GCH}$$

(where $\mathcal{L} \sqsubseteq \mathcal{U}$; that is, $\in^{\mathcal{L}} = \in^{\mathcal{U}} \cap (\mathcal{L} \times \mathcal{L})$).
As noted in chapter I, in SOL we can express GCH by the formula φ_{GCH}. So we have

$$\models_{\mathcal{S}.\mathcal{S}}\ \varphi_{GCH} \text{ iff GCH holds in "the" set-theoretical universe.}$$

Therefore, while confining ourselves to $\mathcal{L}$ as set-theoretical universe,

$$\models_{\mathcal{S}.\mathcal{S}}\ \varphi_{GCH}$$

Since validity can be introduced set-theoretically by a formula of $L^{\in}$, we have that the set-theoretic formula expressing it is true in the constructible universe. Thus,

$(*)\mathcal{L}$ is a model of "$[\models_{\mathcal{S}.\mathcal{S}}\ \varphi_{GCH}]$" (the quoted formula is written in language $L^{\in}$)

Now assume that we have a calculus CAL for SOL which is sound and complete. Not only can we express these properties in the language of set theory, but also the proof can be proved in set theory using only limited resources; that is, resources contained in ZF (the axiom of choice is needed only when you want completeness for an uncountable language). Then, by our initial assumption there is a formula of set theory expressing completeness which can be derived from ZFC,

$$\text{ZFC} \vdash \text{“[CAL is complete for } \mathcal{S}.\mathcal{S}]_{\text{J}}$$

where, again, “[CAL is complete for $\mathcal{S}.\mathcal{S}$]” is a sentence of $L^{\in}$.
Therefore, eliminating the quantification we obtain

$$\text{ZFC} \vdash \text{“}[\models_{\mathcal{S}.\mathcal{S}} \varphi_{GCH}]\text{”} \rightarrow \text{“}[\vdash_{CAL} \varphi_{GCH}]_{\text{J}}$$

Since $\mathcal{L}$ is a model of ZFC (by Gödel theorem) and we are assuming soundness in the background set theory, also

$$\mathcal{L} \text{ is a model of } \text{“}[\models_{\mathcal{S}.\mathcal{S}} \varphi_{GCH}]\text{”} \rightarrow \text{“}[\vdash_{CAL} \varphi_{GCH}]_{\text{J}}$$

Therefore, using (*)

$$\mathcal{L} \text{ is a model of } \text{“}[\vdash_{CAL} \varphi_{GCH}]_{\text{J}}$$

Now we take the crucial step: As was noted in proposition above, the formula expressing provability in the calculus is of a very easy and peculiar kind since it is a persistent formula whose truth value does not change when we go from the constructible universe $\mathcal{L}$ to the whole of $\mathcal{U}$ (recall that $\mathcal{L}$ is a transitive subclass of $\mathcal{W}$).

$$\mathcal{U} \text{ is a model of } \text{“}[\vdash_{CAL} \varphi_{GCH}]\text{”}$$

We will see also that the formula of $L^{\in}$ expressing the validity of φ_{GCH} is true in $\mathcal{U}$. To get it we will use several facts: (1) that the soundness theorem of second order logic is provable in set theory from the axiomatic basis of ZFC, (2) that $\mathcal{U}$ is a model of ZFC and (3) soundness in the background set theory.
By our initial assumption, the soundness theorem of second order calculus can be expressed set-theoretically and its proof is provable in set theory.

$$\text{ZFC} \vdash \text{“[CAL is sound for } \mathcal{S}.\mathcal{S}]_{\text{J}}$$

Therefore,

$$\text{ZFC} \vdash \text{“}[\vdash_{CAL} \varphi_{GCH}]_{\text{J}} \rightarrow \text{“}[\models_{\mathcal{S}.\mathcal{S}} \varphi_{GCH}]_{\text{J}}$$

Since $\mathcal{U}$ is a model of ZFC and we are assuming soundness in the background set theory, then

$$\mathcal{U} \text{ is a model of } \text{“}[\models_{\mathcal{S}.\mathcal{S}} \varphi_{GCH}]_{\text{J}}$$

Therefore, using again our set-theoretical presentation, we have that when our set-theoretic universe is $\mathcal{U}$,

$$\models_{\mathcal{S}.\mathcal{S}} \varphi_{GCH}$$

But $\mathcal{U}$ is not a model of GCH, according to our starting point; namely, $\mathcal{U}$ is Cohen's class model. Therefore, taking $\mathcal{U}$ as "the" set-theoretical universe,

$$\not\models_{\mathcal{S}.\mathcal{S}} \varphi_{GCH}$$

This is a contradiction.

Chapter 3

CATEGORICITY OF PEANO ARITHMETIC

3.1 Relations between standard structures

Substructures

Let $\mathcal{A}$ and $\mathcal{B}$ be two standard structures of the same signature. We say that $\mathcal{A}$ is a *substructure* of $\mathcal{B}$ iff

1. $A \subseteq B$
2. For every n-ary function constant $f \in$ OPER.CONS, then $f^{\mathcal{A}} = f^{\mathcal{B}} \restriction A^n$. (As a special case, zero-ary functions are the same in both structures: $a^{\mathcal{A}} = b^{\mathcal{B}}$)
3. For every n-ary relation constant $R \in$ OPER.CONS, then $R^{\mathcal{A}} = R^{\mathcal{B}} \cap A^n$.

$\mathcal{A} \sqsubseteq \mathcal{B}$ stands for "$\mathcal{A}$ is a substructure of $\mathcal{B}$".
(The notation $f^{\mathcal{B}} \restriction A^n$ corresponds to the restriction of the function $f^{\mathcal{B}}$ to the n-ary cartesian product of $\mathbf{A}$.)

Homomorphism

Let $\mathcal{A}$ and $\mathcal{B}$ be two standard structures of the same signature. $\mathbf{h}$ is called a *homomorphism* from $\mathcal{A}$ into $\mathcal{B}$ iff:

1. $\mathbf{h} : A \longrightarrow B$ (h is a function whose domain is $\mathbf{A}$ and whose values are in B)
2. For every n-ary function constant $f \in$ OPER.CONS and for every $x_1, ..., x_n \in A$:
$$\mathbf{h}(f^{\mathcal{A}}(x_1, ..., x_n)) = f^{\mathcal{B}}(\mathbf{h}(x_1), ..., \mathbf{h}(x_n))$$
(as a special case, $\mathbf{h}(a^{\mathcal{A}}) = a^{\mathcal{B}}$)

3. For every n-ary relation constant $R \in$OPER.CONS and for every $x_1, ..., x_n \in A$:

$$\text{if } \langle x_1, ..., x_n \rangle \in R^{\mathcal{A}} \text{ then } \langle \mathbf{h}(x_1), ..., \mathbf{h}(x_n) \rangle \in R^{\mathcal{B}}$$

$\mathcal{A} \xrightarrow{\mathbf{h}} \mathcal{B}$ stands for "**h** is a homomorphism from $\mathcal{A}$ into $\mathcal{B}$".

$\mathcal{A} \longrightarrow \mathcal{B}$ stands for "there is **h** such that $\mathcal{A} \xrightarrow{\mathbf{h}} \mathcal{B}$".

•When the function **h** satisfying all the previous conditions is *onto*, we say that **h** is a homomorphism of $\mathcal{A}$ *onto* $\mathcal{B}$; i.e.surjective.

•• When condition (3) is changed so that the "if...then" condition is replaced by an "if and only if" condition, we say that the homomorphism is *strong.*

••• When $\mathcal{A} \xrightarrow{\mathbf{h}} \mathcal{B}$ and **h** is onto and the homomorphism is strong, we also say that $\mathcal{B}$ is a *homomorphic image* of $\mathcal{A}$.

Isomorphism

Let $\mathcal{A}$ and $\mathcal{B}$ be two standard structures of the same signature. A function $\mathbf{h} : A \longrightarrow B$ is called an *isomorphism* from $\mathcal{A}$ onto $\mathcal{B}$ iff:

1. The function **h** is a bijection; that is, $\forall \mathbf{xy}(\mathbf{h}(\mathbf{x}) = \mathbf{h}(\mathbf{y}) \Rightarrow \mathbf{x} = \mathbf{y})$ and $\mathbf{h}[A] = B$, where $\mathbf{h}[A]$ is the image of **A** under **h**; i.e., is surjective and injective.

2. This condition is the same as in homomorphisms.

3. For every n-ary relation constant $R \in$ OPER.CONS and for every $x_1, ..., x_n \in A$:

$$\langle x_1, ..., x_n \rangle \in R^{\mathcal{A}} \text{ iff } \langle \mathbf{h}(x_1), ..., \mathbf{h}(x_n) \rangle \in R^{\mathcal{B}}$$

$\mathcal{A} \overset{\mathbf{h}}{\cong} \mathcal{B}$ stands for "**h** is an isomorphism from $\mathcal{A}$ onto $\mathcal{B}$J.

$\mathcal{A} \cong \mathcal{B}$ stands for "there is an **h** such that $\mathcal{A} \overset{\mathbf{h}}{\cong} \mathcal{B}$J.

Embedding

Let $\mathcal{A}$ and $\mathcal{B}$ be two standard structures of the same signature. A function $\mathbf{h} : A \longrightarrow B$ is called an *embedding* from $\mathcal{A}$ into $\mathcal{B}$ iff:

1. The function **h** is one-to-one; that is, $\forall xy(\mathbf{h}(x) = \mathbf{h}(y) \rightarrow x = y)$.

Conditions (2) and (3) are the same as in isomorphisms.

$\mathcal{A} \overset{\mathbf{h}}{\tilde{\sqsubseteq}} \mathcal{B}$ stands for "**h** is an embedding from $\mathcal{A}$ into $\mathcal{B}$".

$\mathcal{A} \tilde{\sqsubseteq} \mathcal{B}$ stands for "there is an **h** such that $\mathcal{A} \overset{\mathbf{h}}{\tilde{\sqsubseteq}} \mathcal{B}$".

Proposition 1: Let **h** be a strong homomorphism from $\mathcal{A}$ into $\mathcal{B}$. **h** is an embedding if and only if **h** is an isomorphism from $\mathcal{A}$ into $\mathbf{h}[\mathcal{A}]$

(where $\mathbf{h}[\mathcal{A}] = \langle \mathbf{h}[A], \langle \mathbf{h}[C^{\mathcal{A}}] \rangle_{C \in \text{OPER.CONS}} \rangle$)

Proposition 2: Let $\mathbf{h}$ be a homomorphism from $\mathcal{A}$ into $\mathcal{B}$. Then $\mathbf{h}$ is an embedding if and only if $\mathbf{h}$ is an isomorphism from $\mathcal{A}$ into $\mathcal{B} \restriction \mathbf{h}[A]$ (where $\mathcal{B} \restriction \mathbf{h}[A]$ is a structure with domain $\mathbf{h}[A]$ having as n-ary functions the ones in $\mathcal{B}$ restricted to $\mathbf{h}[A]^n$ and as relations the intersection of the relations in $\mathcal{B}$ with $\mathbf{h}[A]^n$).

Proposition 3: h is an embedding from $\mathcal{A}$ into $\mathcal{B}$ if and only if there is a structure $\mathcal{C}$ such that $\mathcal{C} \sqsubseteq \mathcal{B}$ and $\mathbf{h}$ is an isomorphism from $\mathcal{A}$ onto $\mathcal{C}$.

Proposition 4: The concepts presented above are related among themselves as shown below:

$$\mathcal{A} \cong \mathcal{B} \implies \mathcal{A} \tilde{\sqsubseteq} \mathcal{B} \implies \mathcal{A} \longrightarrow \mathcal{B}$$
$$\Uparrow$$
$$\mathcal{A} \sqsubseteq \mathcal{B}$$

Congruence relations and homomorphisms

Given a structure $\mathcal{A}$ and a binary relation $\mathbf{R} \subseteq \mathbf{A} \times \mathbf{A}$, $\mathbf{R}$ is a congruence relation on $\mathcal{A}$ iff:

1. $\mathbf{R}$ is an equivalence relation on $\mathbf{A}$; that is, it is reflexive, symmetric and transitive and its domain and range is $\mathbf{A}$.

2. For every n-ary function constant $f \in$OPER.CONS and $\mathbf{x}_1, ..., \mathbf{x}_n, \mathbf{y}_1, ..., \mathbf{y}_n \in \mathbf{A}$:

   if $\langle \mathbf{x}_1, \mathbf{y}_1 \rangle \in \mathbf{R}, ..., \langle \mathbf{x}_n, \mathbf{y}_n \rangle \in \mathbf{R}$ then $\langle f^{\mathcal{A}}(\mathbf{x}_1, ..., \mathbf{x}_n), f^{\mathcal{A}}(\mathbf{y}_1, ..., \mathbf{y}_n) \in \mathbf{R}$

3. For every n-ary relation constant $T \in$ OPER.CONS and $\mathbf{x}_1, ..., \mathbf{x}_n, \mathbf{y}_1, ..., \mathbf{y}_n \in \mathbf{A}$:

   if $\langle \mathbf{x}_1, \mathbf{y}_1 \rangle \in \mathbf{R}, ..., \langle \mathbf{x}_n, \mathbf{y}_n \rangle \in \mathbf{R}$ then $(\mathbf{x}_1, ..., \mathbf{x}_n) \in T^{\mathcal{A}}$ iff $(\mathbf{y}_1, ..., \mathbf{y}_n) \in T^{\mathcal{A}}$.

Proposition 1: If $\mathbf{h}$ is a strong homomorphism from $\mathcal{A}$ into any other structure $\mathcal{B}$, then
$\mathbf{h} = \{\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbf{A} \times \mathbf{A} \mid \mathbf{h}(\mathbf{x}) = \mathbf{h}(\mathbf{y})\}$ is a congruence relation on $\mathcal{A}$.

Proposition 2: For every congruence relation $\approx$ *on* $\mathcal{A}$, there is a strong homomorphism $\mathbf{h}$ from $\mathcal{A}$ onto some other structure, $\mathcal{A} \approx = \langle \mathbf{A} \approx, \langle C^{\mathcal{A}\approx} \rangle_{C \in \text{OPER.CONS}} \rangle$ such that $\approx = \mathbf{h}$.

Proposition 3: If $\mathbf{h}$ is a strong homomorphisms from $\mathcal{A}$ onto $\mathcal{B}$, then $\mathcal{A}^{\mathbf{h}} \cong \mathcal{B}$.

Proposition 4: If $\mathbf{h}_1$ and $\mathbf{h}_2$ are two strong homomorphisms from $\mathcal{A}$ onto $\mathcal{B}_1$ and $\mathcal{B}_2$ respectively, and if $\mathbf{h}_1 = \mathbf{h}_2$, then $\mathcal{B}_1 \cong \mathcal{B}_2$.

3.2 Second Order Peano Axioms

Note. Throughout the following sections our basic SOL language, L_2, will only contain c and σ as zero-ary and unary functions.

Definitions: Peano models and induction models

Let Π be the most famous set of axioms in the world; namely,
$1P := \forall x \neg c = \sigma x$
$2P := \forall xy(\sigma x = \sigma y \rightarrow x = y)$
$3P := \forall X(Xc \wedge \forall z(Xz \rightarrow X\sigma z) \rightarrow \forall x\ Xx)$

We will say that a model of these axioms is a *Peano model* (or a Peano structure).

We will say that a model of 3P is an *induction model.*

Let $PA^2 = \text{CON}(\Pi) = \{\varphi \in \text{SENT}(L_2) |\ \Pi \models \varphi\}$.

3.3 Categoricity of Peano Axioms

To prove the categoricity of the Peano axioms we are going to follow the path which follows. We begin by proving two lemmas from which our theorem 1 is obtained.

$$\textbf{Theorem 1} \iff \left[\begin{array}{l} \textbf{Theorem } \boldsymbol{\gamma} \Longleftarrow \left[\begin{array}{l} \textbf{Lemma } \boldsymbol{\alpha} \\ \textbf{Lemma } \boldsymbol{\beta} \end{array}\right. \\ \textbf{Theorem } \boldsymbol{\delta} \end{array}\right.$$

Theorem γ There is a unique homomorphism between any Peano model $\mathcal{N} = \langle \mathbb{N}, c^{\mathcal{N}}, \sigma^{\mathcal{N}} \rangle$ and any structure $\mathcal{A} = \langle \mathbf{A}, c^{\mathcal{A}}, \sigma^{\mathcal{A}} \rangle$.

Lemma α Every element of $\mathbb{N}$ is in the domain of a partial function.

Lemma β Partial functions agree for joint elements.

Theorem δ Let $\mathcal{N}$ be a Peano model and $\mathcal{A}$ any structure. A necessary and sufficient condition that $\mathcal{A}$ be a homomorphic image of $\mathcal{N}$ is that $\mathcal{A}$ be an induction model.

Theorem 1 Any two Peano models are isomorphic.

Weak incompleteness of SOL with standard semantics

Gödel's incompleteness theorem for PA^2 There is a sentence γ such that $\gamma \in \text{THEO}(\mathcal{N})$ but it is not the case that $PA^2 \vdash_{CAL} \gamma$ in any second order calculus.

Note The formula γ is the well known Gödel formula asserting of itself that it is not a theorem of Peano arithmetic; as it is in fact not a theorem of PA^2, it holds in the standard model of natural numbers, $\mathcal{N}$.

Incompleteness theorem of SOL There is a sentence φ such that $\models_{\mathcal{S}.\mathcal{S}} \varphi$ but it is not the case that $\vdash_{CAL} \varphi$.

Proof We will see that there is a sentence $\varphi \in L_2$ such that $\models_{\mathcal{S}.\mathcal{S}} \varphi$, but it is not the case that $\vdash_{CAL} \varphi$. By Gödel's theorem stated above, $\gamma \in \text{THEO}(\mathcal{N})$. But, since $\text{THEO}(\mathcal{N}) = \text{PA}^2$, $\gamma \in PA^2$; equivalently, $\gamma \in \text{CON}(\Pi)$. Take $\varphi \equiv 1P \wedge 2P \wedge 3P \rightarrow \gamma$. Thus, $\models_{\mathcal{S}.\mathcal{S}} \varphi$ but not $\vdash_{CAL} \varphi$ (otherwise we will have $PA^2 \vdash_{CAL} \gamma$ contradicting Gödel's Theorem stated above).

Chapter 4

FRAMES AND GENERAL STRUCTURES

4.1 Second Order Frames

Definition of frames

A *second order frame* $\mathcal{A}$ (some authors call them pre-structures) is defined by:

$$\mathcal{A} = \langle \mathbf{A}, \langle \mathbf{A}_n \rangle_{n \geq 1}, \langle C^{\mathcal{A}} \rangle_{C \in \text{OPER.CONS}} \rangle$$

where

1. $\mathbf{A} \neq \emptyset$ is the universe of individuals which should be a non-empty set.

2. $\emptyset \neq \mathbf{A}_n \subseteq \wp \mathbf{A}^n$, for each $n \geq 1$.

3. For every n-ary relation constant $R \in$ OPER.CONS with $\text{FUNC}(R) = \langle 0, 1, \underset{n}{\ldots}, 1 \rangle$, $R^{\mathcal{A}}$ is, as in standard structures, an n-ary relation on individuals.
   Now we add the condition that $R^{\mathcal{A}} \in \mathbf{A}_n$; every distinguished relation must be a member of the corresponding universe.

4. For every n-ary function constant $f \in$ OPER.CONS with $\text{FUNC}(f) = \langle 1, 1, \underset{n}{\ldots}, 1 \rangle$, $f^{\mathcal{A}}$ is an n-ary function on individuals. Furthermore, $f^{\mathcal{A}} \in \mathbf{A}_{n+1}$.
   Specifically, for individual constants we agree that the condition becomes $\{a^{\mathcal{A}}\} \in \mathbf{A}_1$.

Let's call the class of frames $\mathcal{F}$.

Definable sets and relations in a given frame

Given a frame

$$\mathcal{A} = \langle \mathbf{A}, \langle \mathbf{A}_n \rangle_{n \geq 1}, \langle C^{\mathcal{A}} \rangle_{C \in \text{OPER.CONS}} \rangle$$

and a second order language L_2, there are several kinds of relations which are related, in one way or another, with $\mathcal{A}$ and $\text{FORM}(L_2)$. In particular, we want to distinguish: (1) first and second order relations *on* the universes of the frame, (2) relations *into* the universes of the frame, (3) *definable* relations of the frame using a given language and (4) *parametrically definable* relations of the frame using a given language.

Definition of first order relations of the frame $\mathcal{A}$

- Please recall the definition of an n-ary *first order relation on* $\mathcal{A}$.
  Let $\text{REL}^{1st}(\mathcal{A})$ be the class of all first order relations of $\mathcal{A}$.

- As above, we say that an n-ary first order relation $\mathbf{X}$ is *into* $\mathcal{A}$ when $\mathbf{X} = R^{\mathcal{A}}$ for $R \in$ OPER.CONS or $\mathbf{X} \in \mathbf{A}_n$.
  Let $\text{REL}^{1st}(\in \mathcal{A})$ be the class of all first order relations into $\mathcal{A}$.

Definition of second order relations of $\mathcal{A}$

- Please recall the definition of an n-ary *second order relation of* $\mathcal{A}$ in chapter I.
  Let $\text{REL}(\mathcal{A})$ be the class of all second order relations of $\mathcal{A}$.

- We say that an n-ary second order relation $\mathbf{X}$ is *into* $\mathcal{A}$ as in chapter I
  Let $\text{REL}(\in \mathcal{A})$ be the class of all second order relations into $\mathcal{A}$.

Definition of $\mathcal{A}$-definable first order relations using L_2

As in chapter I, we say that the formula φ along with the sequence of individual variables $\langle x_1, ..., x_n \rangle$ *defines* $\mathbf{X}$ *in the structure* $\mathcal{A}$ *using* L_2 when

$$\mathbf{X} = \{ \langle \mathbf{x}_1, ..., \mathbf{x}_n \rangle |\ \mathcal{A}^{[\mathbf{x}_1 ... \mathbf{x}_n]}_{[x_1 ... x_n]} sat \varphi \}$$

where $\text{FREE}(\varphi) \subseteq \{x_1, ..., x_n\}$.

Let $\text{DEF}^{1st}(\mathcal{A}, L_2)$ be the smallest class containing all $\mathcal{A}$-definable first order relations using L_2.

Definition of $\mathcal{A}$-definable second order relations using L_2

Recall definition of a second order relation $\mathcal{A}$-*definable using a language* L_2 in chapter I.

Let $\text{DEF}(\mathcal{A}, L_2)$ be the corresponding set.

Definition of first and second order parametrically $\mathcal{A}$-definable relations using L_2

1. A first order relation $\mathbf{X}$ is parametrically $\mathcal{A}$-definable using L_2 iff there are a formula φ, individual variables $x_1, ..., x_n$ and variables $v_1, ..., v_m$ of any types $i_1, ..., i_m \in$ VAR such that

$$\mathbf{X} = \{\langle \mathbf{x}_1,...,\mathbf{x}_n\rangle \in \mathbf{A}^n |\ \mathcal{A}^{[\mathbf{u}_1...\mathbf{u}_n\mathbf{v}_1...\mathbf{v}_m}_{u_1...u_n v_1...v_m}] \text{ sat } \varphi\}$$

   where FREE(φ) $\subseteq \{x_1, ..., x_n, v_1, ..., v_m\}$ and the parameters $\mathbf{v}_1$**,...,**$\mathbf{v}_m$ are in $\mathbf{A}_{i_1}$**,...,**$\mathbf{A}_{i_m}$ of types i_1,...,i_m.

2. A second order relation $\mathbf{X}$ is parametrically $\mathcal{A}$-definable using L_2 iff there are a formula φ, variables $u_1, ..., u_n$ and variables $v_1, ..., v_m$ of any types k_1,...,k_n,i_1,...,i_m $\in$VAR such that

$$\mathbf{X}=\{\langle \mathbf{u}_1,...,\mathbf{u}_n\rangle \in \mathbf{A}_{k_1}\times...\times\mathbf{A}_{k_n} |\ \mathcal{A}^{[\mathbf{u}_1...\mathbf{u}_n\mathbf{v}_1...\mathbf{v}_m}_{u_1...u_n v_1...v_m}] \text{ sat } \varphi\}$$

   where FREE(φ) $\subseteq \{u_1, ..., u_n, v_1, ..., v_m\}$ and the parameters $\mathbf{v}_1, ..., \mathbf{v}_m$ are in $\mathbf{A}_{i_1}$,...,$\mathbf{A}_{i_m}$ of types $i_1, ..., i_m$.

Let PARAM.DEF$^{1st}(\mathcal{A}, L_2)$ and PARAM.DEF$(\mathcal{A}, L_2)$ be the corresponding sets.

4.2 General Structures

Definition of general structures

Definition of general structures by soundness condition Let $\mathcal{A}$ be a frame. We say that $\mathcal{A}$ is a general structure if and only if it is a frame-model of the set Δ of all the comprehension sentences.

Definition of general structures by definable closure Let $\mathcal{A}$ be a frame. We say that $\mathcal{A}$ is a general structure if and only if all parametrically $\mathcal{A}$-definable relations on individuals using the language L_2 are in the corresponding relation universes of the structure; i.e., $\mathbf{A}_n = \wp\mathbf{A}^n \cap$ PARAM.DEF$(\mathcal{A}, L_2)$

Soundness and completeness in general structures

Soundness of C_2 in general structures $\Gamma \vdash_{C_2} \varphi$ implies $\Gamma \models_{\mathcal{G}.\mathcal{S}} \varphi$ for all Γ and φ, where $\Gamma \cup \{\varphi\} \subseteq$FORM$(L_2)$.

The soundness of C_2 in general structures easily follows from the soundness of C_2^- in frames. Therefore, when we have the later we only have to apply this schema,

$$\begin{array}{ccc} \Gamma \vdash_{C_2} \varphi & & \Gamma \models_{\mathcal{G}.\mathcal{S}} \varphi \\ \Updownarrow & & \Updownarrow \\ \Gamma \cup \Delta \vdash_{C_2^-} \varphi & \Longrightarrow & \Gamma \cup \Delta \models_{\mathcal{F}} \varphi \end{array}$$

The soundness of C_2^- in frames is easy to prove, but it has to be done directly, since it does not follow from the soundness in standard structures; that is, standard consequence does not imply frame-consequence.

Soundness of $\lambda - C_2$ in general structures $\Gamma \vdash_{\lambda - C_2} \varphi$ implies $\Gamma \models_{\mathcal{G}.\mathcal{S}} \varphi$ for all Γ and φ, where $\Gamma \cup \{\varphi\} \subseteq \mathrm{FORM}(\lambda - L_2)$.

4.3 Algebraic Definition of General Structures

We have given so far two equivalent definitions of general structures, but both of them depend on the formal language. Here, we are going to see another definition of the same concept using algebraic closure conditions on the domains.

Fundamental relations of a structure

Let $\mathrm{REL}(\mathcal{A})$ be the class of all relations on the universes of $\mathcal{A}$. This class includes all the relations of any finite arity n, $n \geq 1$. We are going to name certain relations in $\mathrm{REL}(\mathcal{A})$ and to define certain operations in it which will allow us to define by an algebraic procedure the class of all parametrically $\mathcal{A}$-definable first and second order relations. From this class we obtain the class of relations between individuals.

The fundamentals relations and operations are the following:

1. **Membership.** For every $n \geq 1$, let $\in_n = \{\langle \mathbf{X}, \mathbf{x}_1, ..., \mathbf{x}_n \rangle \in \mathbf{A}_n \times \mathbf{A}^n \mid \langle \mathbf{x}_1, ..., \mathbf{x}_n \rangle \in \mathbf{X}\}$.

2. **Difference.** For every $\mathbf{R}, \mathbf{S} \in \mathrm{REL}(\mathcal{A})$ of the same type —that is, both $\mathbf{R}$ and $\mathbf{S}$ are subsets of the same cartesian product of certain universes of $\mathcal{A}$— then $\mathbf{R} - \mathbf{S}$ will denote the usual set difference of sets.

3. **Cartesian product.** For every $\mathbf{R} \in \mathrm{REL}(\mathcal{A})$ we denote the cartesian product $\mathbf{A}_n \times \mathbf{R}$ in the standard way, for every $n \geq 1$ and also, $\mathbf{A} \times \mathbf{R}$.

4. **Permutation 1.** ("The last will be first...") Let $\mathbf{R} \in \mathrm{REL}(\mathcal{A})$ and $\mathbf{R} \subseteq \mathbf{A}_{i_1} \times ... \times \mathbf{A}_{i_n}$ with $n \geq 2$, where each $\mathbf{A}_{i_j}$ is either $\mathbf{A}$ or a certain $\mathbf{A}_k$.
$$\mathrm{PER}_1(\mathbf{R}) = \{\langle \mathbf{v}_n, \mathbf{v}_1, ..., \mathbf{v}_{n-1} \rangle \mid \langle \mathbf{v}_1, ..., \mathbf{v}_n \rangle \in \mathbf{R}\}$$

5. **Permutation 2.** ("...and the first, second") Let $\mathbf{R} \in \mathrm{REL}(\mathcal{A})$ as above,
$$\mathrm{PER}_2(\mathbf{R}) = \{\langle \mathbf{v}_2, \mathbf{v}_1, \mathbf{v}_3 ..., \mathbf{v}_{n-1} \rangle \mid \langle \mathbf{v}_1, ..., \mathbf{v}_n \rangle \in \mathbf{R}\}$$

6. **Projection.** Let $\mathbf{R} \in \mathrm{REL}(\mathcal{A})$ as above,
$$\mathrm{PROJ}_2(\mathbf{R}) = \{\langle \mathbf{v}_1, ..., \mathbf{v}_{n-1} \rangle \mid \exists \mathbf{u} \langle \mathbf{v}_1, ..., \mathbf{v}_{n-1}, \mathbf{u} \rangle \in \mathbf{R}\}$$

7. **Singletons.** For every $\mathbf{R} \in \mathbf{A}$ or $\mathbf{R} \in \mathbf{A}_n$, $\{\mathbf{R}\}$ is the singleton of $\mathbf{R}$.

Closure under seven Let $\mathcal{A}$ be a second order frame and $\text{REL}(\mathcal{A})$ the class of all relations on the universes of $\mathcal{A}$. For every $\mathbf{D} \subseteq \text{REL}(\mathcal{A})$, $\mathbf{D}$ is *closed under seven* if and only if all the fundamentals relations are in $\mathbf{D}$ and it is closed under all fundamental operations.

Algebraically defined relations Given a second order frame $\mathcal{A}$, $\text{ALG.DEF}(\mathcal{A})$ is defined as the least subset of $\text{REL}(\mathcal{A})$ closed under seven. That is,

$\text{ALG.DEF}(\mathcal{A}) = \cap\{\mathbf{D}|\ \text{REL}(\in \mathcal{A}) \subseteq \mathbf{D} \subseteq \text{REL}(\mathcal{A})\&\mathbf{D}$ is closed under seven$\}$

Algebraic definition of general structures

Definition by algebraic closure

A frame $\mathcal{A}$ is a general structure iff for every $n \geq 1$,

$\mathbf{A}_n = \wp\mathbf{A}^n \cap \text{ALG.DEF}(\mathcal{A})$

Chapter 5

TYPE THEORY

5.1 Introduction

General idea

So far we are familiar with propositional logic (i.e., zeroth order), first order logic and second order logic. In propositional logic no quantification is allowed; in first order logic we quantify over individuals; in second order logic we also quantify over sets and relations among individuals. Second order relations — as described in chapter I — only incidentally appear in this logic, since we have neither constants nor variables to refer to or to quantify over them. If we can refer to them using variables, we are in third order logic. When we quantify, as well, over second order relations, we are in fourth order logic. A logic which includes all logics of all finite orders (but only them) is known as finite type theory .

5.2 A Relational Theory of Finite Types

Definition (signature and alphabet)

Signature of the relational type theory By a signature Σ for the relational type theory RTT we mean an ordered pair $\Sigma = \langle \text{TS}, \text{FUNC} \rangle$ where:

1. TS is the set of type symbols, which is obtained by observing the following formation rules:

   (a) 1,0$\in$ TS (1 is the type symbol of individuals, while 0 is the type symbol of propositions).

   (b) For any natural number n : If $\alpha_1, ..., \alpha_n \in \text{TS} - \{0\}$, then $\langle 0, \alpha_1, ..., \alpha_n \rangle \in$ TS. (The new type $\langle 0, \alpha_1, ..., \alpha_n \rangle$ is for n-ary relations among types $\alpha_1, ..., \alpha_n$.)

2. FUNC = FUNC(Σ) is a function whose domain is OPER.CONS, the set of operation constants. This function tells you the type of your operation constants. In our present case, FUNC is empty because OPER.CONS is empty.

Alphabet of the RTT language T_ω Let us call T_ω the *Pure* RTT language.

The alphabet of this language includes :

Connectives, quantifiers, parentheses, equality for expressions of each nonzero type α (where equality itself has type $\langle 0, \alpha, \alpha\rangle$) and variables for each nonzero type. Namely,

For every $\alpha \in$TS$-\{0\}$, an infinite countable set $\mathcal{V}_\alpha$ of variables of type α : X^α, Y^α, Z^α, U^α, V^α, W^α, X_1^α , X_2^α , X_3^α , etc.

Expressions

The expressions of T_ω are built from the following rules:

- (E1) $\perp$ is an expression of T_ω whose type is 0.
- (E2) If $\alpha \in$ TS$-\{0\}$, X^α is an expression of T_ω whose type is α.
- (E3) For each $\alpha \in$ TS$-\{0\}$, $=^{\langle 0,\alpha,\alpha\rangle}$ is an expression whose type is the superscript.
- (E4) If A is an expression of type $\langle 0, \alpha_1, .., \alpha_n\rangle$ and if $B_1, ..., B_n$ are expressions of types $\alpha_1, ..., \alpha_n$, then $A(B_1...B_n)$ is an expression of type 0.
- (E5) If φ and ψ are expressions of type 0, then $\neg\varphi$, $(\varphi \vee \psi)$, $(\varphi \wedge \psi)$, $(\varphi \rightarrow \psi)$ and $(\varphi \leftrightarrow \psi)$ are expressions of type 0.
- (E6) If φ is an expression of type 0 and X^α is a variable of any type α, then $\forall X^\alpha\varphi$ and $\exists X^\alpha\varphi$ are expressions of type 0.

EXPR(T_ω) is the smallest set obtained by these rules.
FORM(T_ω) is the set of expressions of type 0.

The relational standard structure and the relational standard hierarchy of types

Definition

The language T_ω of signature Σ introduced so far is primarily designed to talk about *relational standard structures* of the same signature,

$$\mathcal{D} = \langle \mathbf{D}, \langle \mathbf{D}_\alpha\rangle_{\alpha\in \mathbf{TS}}\rangle$$

1. $\mathbf{D} \neq \emptyset$ is a non-empty set.
2. $\langle \mathbf{D}_\alpha\rangle_{\alpha\in \mathbf{TS}}$ is the *relational standard hierarchy of types*; i.e.
   (a) $\mathbf{D}_0 = \{T, F\}$ and $\mathbf{D}_1 = \mathbf{D}$,
   (b) $\mathbf{D}_{\langle 0,\alpha_1,...,\alpha_n\rangle} = \wp(\mathbf{D}_{\alpha_1} \times ... \times \mathbf{D}_{\alpha_n})$.

Assignments

As in second order logic (Chapter I), assignments are functions giving to every variable in the language a value in the hierarchy of types. Thus,

For every $X^\alpha : M(X^\alpha) \in \mathbf{D}_\alpha$.

Also, for all $\mathbf{X}_1 \in \mathbf{D}_{\alpha_1}$,...,$\mathbf{X}_n \in \mathbf{D}_{\alpha_n}$ and variables $X_1, ..., X_n$ of types $\alpha_1, ..., \alpha_n$, you will define $M^{\mathbf{X}_1 ... \mathbf{X}_n}_{X_1 ... X_n}$ in the expected way.

Interpretations

As in second order logic, an interpretation $\mathcal{I}$ over a relational standard structure $\mathcal{D}$ is a pair $\langle \mathcal{D}, \mathbf{M} \rangle$, where $\mathbf{M}$ is an assignment. The inductive definition of denotation of expressions under a given interpretation is the expected one. Since we have a universe $\mathbf{A}_0 = \{T, F\}$ of truth values, the formulas of RTT denote elements of $\mathbf{A}_0$.

RTT with lambda

The language $\lambda - T_\omega$

Our language $\lambda - T_\omega$ is the result of adding to T_ω the lambda's facilities with regard to defining new predicates. So, to the six rules given above for the formation of expressions, you add:

- (E7) If $X_1, ..., X_n$ are pairwise distinct variables of types $\alpha_1, ..., \alpha_n$ and φ is an expression of type 0, then $\lambda(X_1 ... X_n)\varphi$ is an expression of type $\langle 0, \alpha_1, ..., \alpha_n \rangle$.

Denotation of expressions

The denotation of expressions of $\lambda - T_\omega$ agrees basically with the definitions T_ω. We only have to add that we want

$\mathcal{I}(\lambda(X_1^{\alpha_1} ... X_n^{\alpha_n})\varphi) = \{\langle \mathbf{X}_1 ... \mathbf{X}_n \rangle \in \mathbf{D}_{\alpha_1} \times ... \times \mathbf{D}_{\alpha_n} \mid \mathcal{I}^{\mathbf{X}_1 ... \mathbf{X}_n}_{X_1^{\alpha_1} ... X_n^{\alpha_n}} \text{ sat } \varphi\}$

The calculus $\lambda - C_\omega$

This new calculus, to be used with language $\lambda - T_\omega$, is the obvious extension of the calculus $\lambda - C_2$. Thus, instead of the comprehension schema (CS) we will use the rules of (IAC) and (IAA) for introducing abstraction in the consequent and in the antecedent. Of course, we extend them to cover the new types.

Let $X_1, ..., X_n$ be pairwise distinct variables of types $\alpha_1, ..., \alpha_n$ and $\epsilon_1, ..., \epsilon_n$ be expressions of types $\alpha_1, ..., \alpha_n$ as well. We can state the rules as:

(IAC) Introducing abstraction in the consequent

$$\frac{\Omega \hookrightarrow \varphi \frac{\varepsilon_1 ... \varepsilon_n}{X_1 ... X_n}}{\Omega \hookrightarrow (\lambda(X_1 ... X_n)\varphi)(\varepsilon_1 ... \varepsilon_n)}$$

(IAA) Introducing abstraction in the antecedent

$$\frac{\Omega\ \varphi \frac{\varepsilon_1 ... \varepsilon_n}{X_1 ... X_n} \qquad \hookrightarrow \psi}{\Omega\ (\lambda(X_1 ... X_n)\varphi)(\varepsilon_1 ... \varepsilon_n) \hookrightarrow \psi}$$

5.3 A Functional Theory of Finite Types

This is the rather elegant formulation of the simple type theory offered by Church as early as 1940. The main features of FTT are that juxtaposition will have the usual interpretation as denoting the application of function to argument, and the abstractor operator is used to define a new function.

Definition (signature and alphabet)

Signature
Our signature Σ for the functional type theory FTT is an ordered pair $\Sigma = \langle \text{TS,FUNC} \rangle$ where:

1. TS is built by observing the following formation rules
   (a) $1, 0 \in \text{TS}$.
   (b) If $\alpha, \beta \in \text{TS}$, then $\langle \alpha\beta \rangle \in \text{TS}$.
2. FUNC = FUNC(Σ) is a function whose domain is OPER.CONS; where

$$\{\neg^{\langle 00 \rangle},\ \vee^{\langle \langle 00 \rangle 0 \rangle}\} \cup \{\Sigma^{\langle 0 \langle 0\alpha \rangle \rangle},\ =^{\langle \langle 0\alpha \rangle \alpha \rangle} |\ \alpha \in \text{TS}\} \subseteq \text{OPER.CONS}$$

   Among the members of OPER.CONS we very often have a selector operator, $t^{\langle \alpha \langle 0\alpha \rangle \rangle}$ for every $\alpha \in$ TS. The values of the function are specified in the superscripts.

Alphabet of the FTT languages $\lambda - F_\omega$ and $\lambda - cF_\omega$
Let us call $\lambda - F_\omega$ our language for FTT; this language does not include the selector operator. When this operator is in the language, we term it $\lambda - cF_\omega$. The alphabet of any of them includes:

1. All the symbols in OPER.CONS and parentheses *),(* and lambda λ as improper symbols.
2. Variables for all types; *i.e.*, including the propositional types.

Expressions

Definition The expressions of $\lambda - F_\omega$ (resp. of $\lambda - cF_\omega$) are built from the following rules:

- (E1) Any variable or constant alone is an expression whose type is indicated by the superscripts.
- (E2) If A is an expression of type $\langle \alpha\beta \rangle$ and if B is an expression of type β, then *(AB)* is an expression of type α.
- (E3) If A is an expression of type α and X^β is a variable of any type β, then $(\lambda X^\beta A)$ is an expression of type $\langle \alpha\beta \rangle$.

EXPR($\lambda - F_\omega$) is the smallest set obtained by these rules.
FORM($\lambda - F_\omega$) is the set of expressions of type 0.
EXPR($\lambda - cF_\omega$) and FORM($\lambda - cF_\omega$) also include the choice function.

Functional frames, functional general structures and functional standard structures

A functional frame $\mathcal{D}$ of type Σ is an ordered tuple

$$\mathcal{D} = \langle \mathbf{D}, \langle \mathbf{D}_\alpha \rangle_{\alpha \in \mathbf{TS}}, \langle C^{\mathcal{D}} \rangle_{C \in \text{OPER.CONS}} \rangle$$

where:

1. $\mathbf{D} \neq \emptyset$ is a set.
2. $\langle \mathbf{D}_\alpha \rangle_{\alpha \in TS}$ is a functional hierarchy of types; i.e.,
   (a) $\mathbf{D}_0 = \{T, F\}$ and $\mathbf{D}_1 = \mathbf{D}$.
   (b) $\mathbf{D}_{\langle \alpha\beta \rangle} \subseteq \{\mathbf{f} \mid \mathbf{f} : \mathbf{D}_\beta \rightarrow \mathbf{D}_\alpha\}$. That is, $\mathbf{D}_{\langle \alpha\beta \rangle}$ is a set of functions from $\mathbf{D}_\beta$ into $\mathbf{D}_\alpha$.
3. For any $C \in$ OPER.CONS with FUNC(C) = $\langle \alpha\beta \rangle$, $C^{\mathcal{D}} : \mathbf{D}_\beta \rightarrow \mathbf{D}_\alpha$. In particular,

$$(\neg^{\langle 00 \rangle})^{\mathcal{D}} : \begin{array}{l} \mathbf{D}_0 \longrightarrow \mathbf{D}_0 \\ T \longmapsto F \\ F \longmapsto T \end{array}$$

$$(\vee^{\langle \langle 00 \rangle 0 \rangle})^{\mathcal{D}} : \begin{array}{l} \mathbf{D}_0 \longrightarrow \mathbf{D}_0 \\ T \longmapsto (\vee^{\langle \langle 00 \rangle 0 \rangle})^{\mathcal{D}}(T) : \begin{array}{l} \mathbf{D}_0 \rightarrow \mathbf{D}_0 \\ \mathbf{X} \longmapsto T \end{array} \\ F \longmapsto (\vee^{\langle \langle 00 \rangle 0 \rangle})^{\mathcal{D}}(F) : \begin{array}{l} \mathbf{D}_0 \rightarrow \mathbf{D}_0 \\ \mathbf{X} \longmapsto \mathbf{X} \end{array} \end{array}$$

$$(\Sigma^{\langle 0\langle 0\alpha \rangle \rangle})^{\mathcal{D}} : \begin{array}{lll} \mathbf{D}_{\langle 0\alpha \rangle} & \longrightarrow & \mathbf{D}_0 \\ \mathbf{f} & \longmapsto & T\ (*) \\ \mathbf{g} & \longmapsto & F \text{ otherwise } (**) \end{array}$$

(*) if the function $\mathbf{f} : \begin{array}{l} \mathbf{D}_\alpha \rightarrow \mathbf{D}_\alpha \\ \mathbf{X} \longmapsto T \end{array}$ has a value T for at least one $\mathbf{X} \in D_\alpha$.

(**) That is, when $\mathbf{g} : \begin{array}{l} \mathbf{D}_\alpha \rightarrow \mathbf{D}_0 \\ \mathbf{X} \longmapsto F \end{array}$ is F for all

$$X \in D_\alpha.(=^{\langle \langle 0\alpha \rangle \alpha \rangle})^{\mathcal{D}} : \begin{array}{l} \mathbf{D}_\alpha \longrightarrow \mathbf{D}_{\langle 0\alpha \rangle} \\ \mathbf{X} \longmapsto (=^{\langle \langle 0\alpha \rangle \alpha \rangle})^{\mathcal{D}}(\mathbf{X}) : \begin{array}{l} \mathbf{D}_\alpha \rightarrow \mathbf{D}_0 \\ \mathbf{X} \longmapsto T \\ \mathbf{Y} \longmapsto F \text{ otherwisw } (*) \end{array} \end{array}$$

(*) that is, for all $\mathbf{Y} \neq \mathbf{X} \in \mathbf{D}_\alpha$.

For $\lambda - cF_\omega$ the interpretation of the selector operator is based on a choice function for every $\alpha \in$ TS.
$(t^{\langle\alpha\langle 0\alpha\rangle\rangle})^{\mathcal{D}}$ is an election function whose value for every singleton

$$\mathbf{f} : \mathbf{D}_\alpha \longrightarrow \mathbf{D}_0$$

is the only element mapped into T by $\mathbf{f}$. It is a fixed element of $\mathbf{D}_\alpha$ if $\mathbf{f}$ is always F for every $\mathbf{X} \in \mathbf{D}_\alpha$ or if there is more than one $\mathbf{X} \in \mathbf{D}_\alpha$ mapped into T by $\mathbf{f}$. Now we fix one element of $\mathbf{D}_\alpha$ for every type α using a choice function, $\mathbf{C}$, for type 1. This is done inductively by setting

$$\mathbf{a}^0 = F = \mathbf{C}(\mathbf{D}_1)$$

and, for any α and β, taking $\mathbf{a}^{\langle\alpha\beta\rangle}$ to be the function of $\mathbf{D}_{\langle\alpha\beta\rangle}$ such that

$$\mathbf{a}^{\langle\alpha\beta\rangle}(X) = \mathbf{a}^\alpha, \text{ for every } \quad \mathbf{X}\in\mathbf{D}_\beta$$

4. All the members of $\{C^{\mathcal{D}}\}_{C\in\text{OPER.CONS}}$ are in the corresponding universes for quantification needs.

5.4 Equational Presentation

Main features of ETT

Equational type theory (ETT) is nothing but a presentation of functional type theory. Its signature is that presented for FTT but the set OPER.CONS needs only to contain equality. The alphabet of our formal language for ETT, which we term $\lambda - E_\omega$, only contains variables and equality as proper symbols, and parentheses and abstractor as improper ones. The rules for forming the expressions are (E1), (E2) and (E3) of 4.2. Thus,

$$\text{FORM}(\lambda - E_\omega) = \text{FORM}((\lambda - F_\omega) - (\{\neg^{\langle 00\rangle}, \vee^{\langle\langle 00\rangle 0\rangle}\} \cup \{\Sigma^{\langle 0\langle 0\alpha\rangle\rangle} |\ \alpha \in TS\}))$$

Another language for ETT used here — termed $\lambda - cE_\omega$ — includes a selector operator, but restricted to certain types. We will use functional frames and functional standard structures and all the definitions introduced above also apply here. Being but a presentation of functional type theory, we only need to prove that FTT can be reduced to ETT; namely, that the connectors and quantifiers can be expressed with equality and abstraction.

Connectors and quantifiers in ETT

Since the only basic symbols of ETT are equality and abstraction and both of them are in FTT, we will show that they are sufficient by introducing the remaining basic symbols of FTT as abbreviations.

Basic definitions

For any formulas φ and ψ, $\varphi \leftrightarrow \psi$ can be used instead of $((=^{\langle\langle 00\rangle 0\rangle}\ \varphi)\psi)$.

1. $\top^0$ stands for $((=^{\langle\langle 0\langle 00\rangle\rangle\langle 00\rangle\rangle}\ (\lambda X^0 X^0))(\lambda X^0 X^0))$
   abbreviated as $\top :=_{Df} \lambda X^0 X^0 = \lambda X^0 X^0$

2. $\bot^0$ stands for $((=^{\langle\langle 0\langle 00\rangle\rangle\langle 00\rangle\rangle}\ (\lambda X^0 X^0))(\lambda X^0 \top^0))$
   abbreviated as $\bot :=_{Df} \lambda X^0 X^0 = \lambda X^0 \top$

3. $\neg^{\langle 00\rangle}$ stands for $(\lambda X^0((=^{\langle\langle 00\rangle 0\rangle 0\rangle}\bot^0)X^0))$
   abbreviated as $\neg :=_{Df} \lambda X^0(\bot = X^0)$

4. $\Sigma^{\langle 0\langle 0\alpha\rangle\rangle}$ stands for $(\lambda X^{\langle 0\alpha\rangle}(\neg^{\langle 00\rangle}((=^{\langle\langle 0\langle\alpha 0\rangle\rangle\langle\alpha 0\rangle\rangle}\ (\lambda X^\alpha(X^{\langle 0\alpha\rangle}X^\alpha)))(\lambda X^\alpha\ \bot^0$ $))))$
   abbreviated as $\Sigma :=_{Df} \lambda X^{\langle 0\alpha\rangle}\lambda X^\alpha(X^{\langle 0\alpha\rangle}X^\alpha) \neq \lambda X^\alpha\ \bot$

5. $\exists X^\alpha\varphi$ stands for $(\neg^{\langle 00\rangle}((=^{\langle\langle 0\langle\alpha 0\rangle\rangle\langle\alpha 0\rangle\rangle}\ (\lambda X^\alpha\varphi))(\lambda X^\alpha\ \bot^0)))$
   abbreviated as $\exists X^\alpha\varphi :=_{Df} \lambda X^\alpha\varphi \neq \lambda X^\alpha\ \bot$

6. $\vee^{\langle 0\langle 00\rangle\rangle}$ stands for

$$(\lambda X^0(\lambda Y^0(\exists X^{\langle 00\rangle}(\neg^{\langle 00\rangle}((=^{\langle\langle 00\rangle 0\rangle}\ ((=^{\langle\langle 00\rangle 0\rangle}\ (X^{\langle 00\rangle}(\neg^{\langle 00\rangle}X^0)))\ (X^{\langle 00\rangle}\top^0)))(\neg^{\langle 00\rangle}Y^0))))))$$

   abbreviated as $\vee :=_{Df} \lambda X^0(\lambda Y^0\exists X^{\langle 00\rangle}((X^{\langle 00\rangle}\neg X^0 = X^{\langle 00\rangle}\top) \neq \neg Y^0))$.

Explanation Let $\mathcal{I} = \langle \mathcal{D}, M\rangle$ be a frame-model. It is easy to see that using the six definitions given above, $\mathcal{I}(\top) = T$ while $\mathcal{I}(\bot) = F$. Moreover, $\mathcal{I}(\neg)$ is a function of type $\langle 00\rangle$ whose value is T iff $\mathcal{I}(X^0) = F$. Therefore, the symbol introduced in (1) is the name for truth, while in (2) we are defining falsehood. On the other hand, (3) is the negation function. Furthermore, $\mathcal{I}(\Sigma)$ is a function giving to sets of type $\langle 0\alpha\rangle$ the value T iff the set is not empty. Existential quantification can be reduced to this concept or be defined directly as in (5). Finally, even disjunction can be defined in terms of equality and lambda; as you see, the formula in (6) is a function which is F iff both $\mathcal{I}(X^0) = F$ and $\mathcal{I}(Y^0) = F$.

Let us examine this in detail:

1. When $\mathcal{I}(X^0) = T$ and $\mathcal{I}(Y^0) = T$, then $\mathcal{I}(\neg Y^0) = F$ and inequality is T because, for the function $\mathbf{g} : \mathbf{D}_0 \longrightarrow \mathbf{D}_0$ sending both T and F to T, the equation $X^{\langle 00\rangle}\neg X^0 = X^{\langle 00\rangle}\top$ is T.

2. When $\mathcal{I}(X^0) = T$ but $\mathcal{I}(Y^0) = F$, then $\mathcal{I}(\neg Y^0) = T$ and inequality is T because, for the identity function $\mathbf{i} : \mathbf{D}_0 \longrightarrow \mathbf{D}_0$, the equation $X^{\langle 00\rangle}\neg X^0 = X^{\langle 00\rangle}\top$ is F.

3. When $\mathcal{I}(X^0) = F$ and $\mathcal{I}(Y^0) = T$, then $\mathcal{I}(\neg Y^0) = F$ and inequality is T because, for any function $\mathbf{f} : \mathbf{D}_0 \longrightarrow \mathbf{D}_0$, the equation $X^{\langle 00\rangle}\neg X^0 = X^{\langle 00\rangle}\top$ is T.

4. Finally, when $\mathcal{I}(X^0) = F$ and $\mathcal{I}(Y^0) = F$, then $\mathcal{I}(\neg Y^0) = T$ and inequality can never be T because, for any function $\mathbf{h} : \mathbf{D}_0 \longrightarrow \mathbf{D}_0$, the equation $X^{\langle 00\rangle}\neg X^0 = X^{\langle 00\rangle}\top$ is T.

A calculus for ETT

The calculus we will present for ETT is an axiomatic one having four basic axioms and a rule of replacement. We will also present another calculus for a language with a selector operator which includes an axiom for descriptions.

Axioms

1. **(Ax.1). Only two truth values**

$$(X^{\langle 00\rangle} \perp \wedge X^{\langle 00\rangle} \perp) = \forall X^0(X^{\langle 00\rangle}X^0)$$

2. **(Ax.2). Equals substitution**

$$(X^\alpha = Y^\alpha) \rightarrow (Z^{\langle 0\alpha\rangle}X^\alpha = Z^{\langle 0\alpha\rangle}Y^\alpha)$$

3. **(Ax.3). Extensionality**

$$\forall X^\beta(X^{\langle \alpha\beta\rangle}X^\beta = Y^{\langle \alpha\beta\rangle}X^\beta) = (X^{\langle \alpha\beta\rangle} = Y^{\langle \alpha\beta\rangle})$$

4. **(Ax.4). Lambda conversion**

$$(\lambda X^\beta B^\alpha)A^\beta = B^\alpha\frac{A^\beta}{X^\beta}$$

5. **(Ax.5). Description**

$$t^{\langle 1\langle 01\rangle\rangle}(\lambda X^1(X^1 = Y^1)) = Y^1$$

Rule of inference

(R) Replacement

To infer from φ and $A^\alpha = B^\alpha$ any formula ψ obtained from φ by replacing in φ a part of the form A^α by occurrences of B^α.

Chapter 6

MANY-SORTED LOGIC

6.1 Structures

Definition (signature)

By a *signature* Σ we mean an ordered pair $\Sigma = \langle \text{SORT,FUNC} \rangle$ where:

1. $\text{SORT} = \text{SORT}(\Sigma)$ is an index set with $0 \in \text{SORT}$
2. $\text{FUNC} = \text{FUNC}(\Sigma)$ is a function whose values are in $[S_\omega(\text{SORT})] \cup [\omega - \{0\}]$
   Elements of $S_\omega(\text{SORT})$ are termed types while elements of $\omega - \{0\}$ are termed arities.
   We use OPER.SYM as Dom(FUNC) and call its elements operation symbols. OPER.SYM is partitioned into disjoint subsets
   $$[OPER.SYM]_\alpha = \{f \mid \text{FUNC}(f) = \alpha\}$$
   Every operation symbol in OPER.SYM is different from the rest and none is a string of other operation symbols.
3. $\neg, \vee, E \in$ OPER.SYM and $\text{FUNC}(\neg) = \langle 0, 0 \rangle$, $\text{FUNC}(\vee) = \langle 0, 0, 0 \rangle$, $\text{FUNC}(E) = 2$.
4. In fact, $\neg$ and $\vee$ are the only f such that $\text{FUNC}(f) = \langle i_0, i_1, ..., i_n \rangle$ with $0 \in \{i_1, ..., i_n\}$.

Definition (structure)

By a *structure* $\mathcal{A}$ of signature $\Sigma = \langle \text{SORT,FUNC} \rangle$ we mean a pair

$$\mathcal{A} = \langle \langle \mathbf{A}_i \rangle_{i \in \text{SORT}} \ , \ \langle f^{\mathcal{A}} \rangle_{f \in \text{OPER.SYM}} \rangle$$

as follows:

1. $\langle \mathbf{A}_i \rangle_{i \in SORT}$ is a family of non-empty sets. For each $i \in$SORT, $\mathbf{A}_i$ is the i^{th} universe of $\mathcal{A}$. We have $\mathbf{A}_0 = \{T, F\}$.

2. $\langle f^{\mathcal{A}} \rangle_{f \in OPER.SYM}$ is a family of functions.

   (a) For each $f \in$ OPER.SYM and FUNC $(f) = \langle i_0, ..., i_m \rangle$, we have

   $$f^{\mathcal{A}} : \mathbf{A}_{i_1} \times ... \times \mathbf{A}_{i_m} \longrightarrow \mathbf{A}_{i_0}$$

   In any structure $\mathcal{A}$ we will always have $\neg, \vee \in OPER.SYM$ and we want all these symbols, in the two-valued logic, to be interpreted in the classical way, therefore:

   $$\neg^{\mathcal{A}} : \begin{array}{l} \mathbf{A}_0 \longrightarrow \mathbf{A}_0 \\ T \longmapsto F \\ F \longmapsto T \end{array} \qquad \vee^{\mathcal{A}} : \begin{array}{l} \mathbf{A}_0 \times \mathbf{A}_0 \longrightarrow \mathbf{A}_0 \\ \langle T, T \rangle \longmapsto T \\ \langle T, F \rangle \longmapsto T \\ \langle F, T \rangle \longmapsto T \\ \langle F, F \rangle \longmapsto F \end{array}$$

   (b) For $f \in$ OPER.SYM and FUNC$(f) = n$ we have an n-ary untyped relation:

   $$f^{\mathcal{A}} : (\bigcup_{i \in SORT - \{0\}} \mathbf{A}_i)^n \longrightarrow \mathbf{A}_0$$

   In particular we have $E \in$ OPER.SYM, with FUNC$(E) = 2$, whose intended meaning is identity. Therefore, $E^{\mathcal{A}}(\mathbf{x}, \mathbf{y}) = T$ iff $\mathbf{x} = \mathbf{y}$. The structures which interpret equality as identity are called normal structures; all our structures are going to be normal, without explicitly indicating so.

6.2 Formal Many-Sorted Language

To talk about a many-sorted structure $\mathcal{A}$ of signature Σ we need a many-sorted language L of the same signature.

Alphabet

The alphabet of L includes all the operation symbols in OPER.SYM, the quantifiers and an infinite number of variables for each sort $i \in$SORT$-\{0\}$. In particular, L contains:

1. For each $i \in$SORT$-\{0\}$: v_0^i, v_1^i, v_2^i,... (variables of sort i and whose type is i too). The set of variables, $\mathcal{V}$, is itself partitioned into disjoint subsets, $\{\mathcal{V}_i \mid \ i \in$SORT$-\{0\}\}$.
   Thus, we let $\mathcal{V} = \bigcup\{\mathcal{V}_i | \ i \in$SORT$-\{0\}\}$.

2. All the operation symbols in OPER.SYM.

3. Quantifier: $\exists$.

Expressions: formulas and terms

Now, from the set of finite strings of elements of the alphabet we are going to select the expressions of L. We will use induction to define it.

- (E1) Each variable of sort i is an expression of the same type (*i.e.*, the single type i).
- (E2) If $f \in$OPER.SYM and $\text{FUNC}(f) = \langle i_0, i_1, ..., i_m \rangle$, and $\epsilon_1, ..., \epsilon_m$ are expressions of single types $i_1, ..., i_m$, then $f\ \epsilon_1 ... \epsilon_m$ is an expression of type i_0.
  If $R \in$OPER.SYM and $\text{FUNC}(R) = n$, then for all the expressions of nonzero arbitrary single types $i_1, ..., i_n$ the string $R\epsilon_1 ... \epsilon_n$ is an expression of type 0.
- (E3) If ϵ is an expression of type 0 and x^i is a variable of sort i, then $\exists x^i \epsilon$ is an expressions of type 0 as well.
- Finally, no other string is an expression.

We call $\text{TERM} = \text{TERM}(L)$ the set of all expressions of single nonzero type i (*i.e.* $i \in$SORT$-\{0\}$) and use the Greek letter τ as a metavariable to refer to a term. When needed, we put a superscript, τ^i, to show the type of the term.

We call $\text{FORM} = \text{FORM}(L)$ the set of all expressions of type 0 and use the Greek letters φ and ψ as metavariables to refer to formulas.

6.3 Semantics

Given a language L and a structure $\mathcal{A}$, both of them sharing the same signature Σ, each closed term in L will denote an element in $\mathcal{A}$ and each sentence in L is true or false in $\mathcal{A}$. Nevertheless, we want to widen the scope of our definitions so that each term or formula gets an interpretation in $\mathcal{A}$. To do that we need to define assignments, where in many-sorted logic an assignment is a mapping

$$\mathbf{M} : \bigcup_{i \in SORT - \{0\}} \mathcal{V}_i \longrightarrow \bigcup_{i \in \mathbf{SOR}T - \{0\}} \mathbf{A}_i$$

in such a way that $\mathbf{M}[\mathcal{V}_i] \subseteq \mathbf{A}_i$; that is to say that variables of sort i always get their value in $\mathbf{A}_i$.

An interpretation $\mathcal{I}$ over a structure $\mathcal{A}$ is a pair $\langle \mathcal{A}, \mathbf{M} \rangle$ where $\mathbf{M}$ is an assignment on $\mathcal{A}$.

Definitions

Denotation
Let $\mathcal{I} = \langle \mathcal{A}, \mathbf{M} \rangle$.

- (E1) $\mathcal{I}(x^i) = \mathbf{M}(x^i)$

- (E2) $\mathcal{I}(f\ \epsilon_1...\epsilon_n) = f^{\mathcal{A}}(\mathcal{I}(\epsilon_1)...\mathcal{I}(\epsilon_n))$
  As particular cases we have:
  - $\mathcal{I}(a_i) = a_i^{\mathcal{A}} \in \mathbf{A}_i$
  - $\mathcal{I}(f^{\beta}\tau_1...\tau_n) = (f^{\beta})^{\mathcal{A}}(\mathcal{I}(\tau_1), ..., \mathcal{I}(\tau_n))$
  - $\mathcal{I}(R^{\beta}\tau_1...\tau_n) = (R^{\beta})^{\mathcal{A}}(\mathcal{I}(\tau_1), ..., \mathcal{I}(\tau_n))$
    (It is T or F according to the value of $\langle \mathcal{I}(\tau_1), ..., \mathcal{I}(\tau_n)\rangle$ in the characteristic function $(R^{\beta})^{\mathcal{A}}$.)
  - $\mathcal{I}(R_n\tau_1...\tau_n) = R_n^{\mathcal{A}}(\mathcal{I}(\tau_1), ..., \mathcal{I}(\tau_1))$
  - $\mathcal{I}(E\tau_1\tau_2) = E^{\mathcal{A}}(\mathcal{I}(\tau_1), \mathcal{I}(\tau_2))$
  - $\mathcal{I}(\neg\varphi) = \neg^{\mathcal{A}}\mathcal{I}(\varphi)$
  - $\mathcal{I}(\varphi \vee \psi) = \vee^{\mathcal{A}}(\mathcal{I}(\varphi), \mathcal{I}(\psi))$
- (E3) $\mathcal{I}(\exists x^i\varphi) = T$ iff $\{x^i \in A^i | \langle \mathcal{A}, M_{x^i}^{\mathbf{x}^i}\rangle(\varphi) = T\} \neq \emptyset$
  (where $M_{x^i}^{\mathbf{x}^i} = (M - \{\langle x^i, M(x^i)\rangle\}) \cup \{\langle x^i, \mathbf{x}^i\rangle\}$)

6.4 Semanticis Theorems

Coincidence lemma

Lemma Let $\mathcal{A}$ be a many-sorted structure, and M_1 and M_2 two assignments on $\mathcal{A}$. For any expression ϵ such that $M_1 \restriction \text{FREE}(\epsilon) = M_2 \restriction \text{FREE}(\epsilon)$, we have

$$\langle \mathcal{A}, M_1\rangle(\epsilon) = \langle \mathcal{A}, M_2\rangle(\epsilon)$$

Substitution lemma

Given a many-sorted interpretation $\mathcal{I}$ (where $\mathcal{I} = \langle \mathcal{A}, M\rangle$), we have

$$\mathcal{I}_{x^i}^{\mathcal{I}(\tau)}(\epsilon) = \mathcal{I}(\epsilon\frac{\tau}{x^i})$$

for every expression ϵ of MSL and every term τ of type i.

Equals substitution lemma

Given a many-sorted interpretation $\mathcal{I}$,

$$\mathcal{I}(\epsilon\frac{\tau}{x^i}) = \mathcal{I}(\epsilon\frac{t}{x^i})$$

for every expression ϵ, whenever $\mathcal{I}(\tau) = \mathcal{I}(t)$, and x^i has the same type as τ iff x^i has the same type as t.

6.5 The Completeness of Many-Sorted Logic

Deductive calculus

Very briefly we will present a sequent calculus for many-sorted logic. As in second order logic, it is the sequent calculus of Ebbinghaus *et al* [1984]. Since this is the only calculus I will present for many-sorted logic, we will term it MSL.

Sequent rules

As in second order logic, a deduction is a finite non-empty sequence of lines, each of which is a finite non-empty sequence of formulas: $\langle \varphi_1 ... \varphi_n \psi \rangle$ is called a *sequent* with *antecedent* $\varphi_1, ..., \varphi_n$ and *consequent* ψ.

The rules are the following:

1. **(HI). Hypothesis introduction**
2. **(M). Monotony**
3. **(PC). Proof by cases**
4. **(NC). Non contradiction**
5. **(IDA). Introducing disjunction in the antecedent**
6. **(IDC). Introducing disjunction in the consequent**
7. **(IPA). Introducing particularization in the antecedent**

$$\frac{\Omega\ \varphi\frac{y^i}{x^i} \hookrightarrow \psi}{\Omega\ \exists x^i \varphi \hookrightarrow \psi} \quad , y^i \notin \text{FREE}(\Omega \cup \{\exists x^i \varphi, \psi\})$$

8. **(IPC). Introducing particularization in the consequent**

$$\frac{\Omega \hookrightarrow \varphi\frac{\tau}{x^i}}{\Omega \hookrightarrow \exists x^i \varphi}$$

(where x^i and τ are of the same type)

9. **(RE). Reflexivity of equality**
10. **(ES). Equals substitution**

$$\frac{\Omega \hookrightarrow \varphi\frac{\tau}{x^i}}{\Omega\ \tau = t \hookrightarrow \varphi\frac{t}{x^i}}$$

(where x^i is of the same type as τ iff x^i is of the same type as t)

Deductions The concepts of derivable sequent in MSL and of derivable formula form a set of formulas as in second order logic, see Chapter I for details. We will write the derivability sign $\vdash$ with a subscript when needed; namely, $\vdash_{MSL}$.

Syntactic notions

Preliminary definitions

1. $\Delta \subseteq$ FORM is *contradictory* iff for every $\varphi \in$FORM, $\Delta \vdash \varphi$.
2. $\Delta \subseteq$ FORM is *consistent* iff it is not contradictory.
3. $\Delta \subseteq$ FORM is a *maximally consistent* set iff Δ is consistent and whenever $\varphi \in$FORM and $\varphi \notin \Delta$, then $\Delta \cup \{\varphi\}$ is contradictory.
4. $\Delta \subseteq$ FORM *contains witnesses* iff for every existential formula, if $\exists x^i \varphi \in \Delta$, we have $\varphi \frac{\tau^i}{x^i} \in \Delta$, for some τ^i such that τ^i is a term of the same type as x^i.

Theorems on consistency

The following list includes some immediate corollaries of the definitions and the rules of the sequent calculus.

1. If Δ is consistent and $\Gamma \subseteq \Delta$, then Γ is consistent.
2. If Δ is contradictory and $\Delta \subseteq \Gamma$, then Γ is contradictory.
3. If Δ is consistent, $\exists x^i \varphi \in \Delta$ and $z^i \notin$ FREE(Δ), then $\Delta \cup \{\varphi \frac{z^i}{x^i}\}$ is also consistent.
4. $\Delta \subseteq$ FORM is contradictory iff there is a $\varphi \in$ FORM such that $\Delta \vdash \varphi$ and $\Delta \vdash \neg\varphi$.

Theorems on maximal consistency

Let $\Delta \subseteq$ FORM be a maximal consistent set and φ, $\psi \in$ FORM. Then all the following hold:

1. If $\Delta \vdash \varphi$ then $\varphi \in \Delta$.
2. If $\vdash \varphi$ then $\varphi \in \Delta$.
3. $\neg\varphi \in \Delta$ iff $\varphi \notin \Delta$.

(4) $\varphi \vee \psi \in \Delta$ iff $\varphi \in \Delta$ or $\psi \in \Delta$.
(5) If $\Delta \cup \{\varphi\} \vdash \psi$ and $\Delta \cup \{\psi\} \vdash \varphi$ then: $\varphi \in \Delta$ iff $\psi \in \Delta$.

Theorem on witnesses

Let $\Delta \subseteq$ FORM be a maximal consistent set such that Δ contains witnesses. Then:
$\exists x^i \varphi \in \Delta$ iff there is a term τ^i (of type i) such that $\varphi \frac{\tau^i}{x^i} \in \Delta$.

Lemma: finiteness of consistency

A set Δ of formulas is consistent if and only if every finite subset of Δ is consistent.

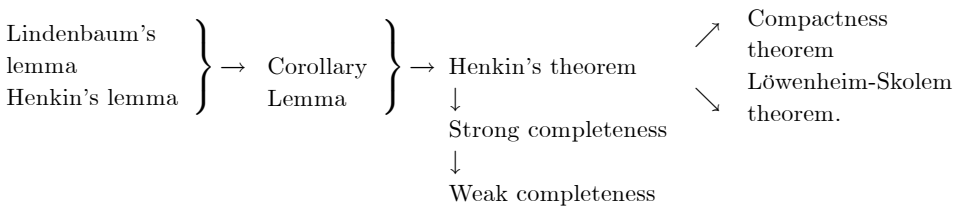
Soundness

Theorem For each $\Gamma \subseteq \text{FORM}$ and $\varphi \in \text{FORM}$, if $\Gamma \vdash \varphi$ then $\Gamma \models \varphi$.

Corollary If Δ has a model, then Δ is consistent.

Completeness theorem (countable language)

We will prove the completeness theorem in its strong sense, $\Gamma \models \varphi$ implies $\Gamma \vdash \varphi$ —for any Γ, φ such that $\Gamma \cup \{\varphi\} \subseteq \text{FORM}(L)$—. To prove completeness and its corollaries we will follow the path:



We will prove each of these theorems in the following form:

Lindenbaum lemma: If $\Gamma \subseteq \text{FORM}(L)$ is consistent and $\text{FREE}(\Gamma)$ finite, there exists $\Gamma*$ such that $\Gamma \subseteq \Gamma* \subseteq \text{FORM}(L)$, $\Gamma*$ is maximally consistent and contains witnesses.

Henkin's lemma: If $\Gamma*$ is maximally consistent and contains witnesses, then $\Gamma*$ has a countable model.

Corollary: If $\Gamma \subseteq \text{FORM}(L)$ is consistent and $\text{FREE}(\Gamma)$ is finite, then Γ has a countable model.

Lemma: If $\Gamma \subseteq \text{FORM}(L)$ is consistent and $\Delta \subseteq \text{SENT}(L*)$ is the class of sentences obtained from formulas in Γ by replacing every free variable by a new constant in $L*$ and if it happens that Δ has a countable model, then so has Γ.

Henkin's theorem: If $\Gamma \subseteq \text{FORM}(L)$ is consistent, then Γ has a model whose domain is countable.

Strong completeness: If $\Gamma \models \varphi$ then $\Gamma \vdash \varphi$

Weak completeness: If $\models \varphi$ then $\vdash \varphi$.

Compactness theorem: Γ has a model iff every finite subset of it has a model.

Löwenheim-Skolem: If Γ has a model, then it has a countable model.

Chapter 7

APPLYING MANY-SORTED LOGIC

7.1 General Plan

Representation theorem

The general plan, which has to be adapted to the particular cases, is as follows: The expressions of the logic to be studied — call it XL — are translated into a many-sorted language of a logic of a peculiar signature directly related to the signature of the original logic — call it $\Sigma\#$ and MSL# the logic, while the structures used to interpret the logic XL — call the class of them ST(XL) — are converted into structures for MSL# — call their class ST(MSL#).

Thus, we need to define a recursive function TRANS to do the translation and a direct conversion of structures, CONV_1.

Thus, we have to add two preliminary provisos:

(A) The function

$$\begin{array}{rrcl} \mathrm{TRANS}: & \mathrm{EXPR(XL)} & \longrightarrow & \mathrm{EXPR(MSL\#)} \\ & \varepsilon & \longmapsto & \mathrm{TRANS}(\varepsilon) \end{array}$$

must be defined by recursion on the formation of expressions of XL. We also want the translation to introduce at most a finite number of free variables in the closed expressions of XL.

On the other hand,

(B) The function

$$\begin{array}{rrcl} \mathrm{CONV}_1: & \mathrm{ST(XL)} & \longrightarrow & \mathrm{ST(MSL\#)} \\ & \varepsilon & \longmapsto & \mathrm{CONV}_1(\varepsilon) \end{array}$$

must be a conversion of structures for XL into many-sorted structures of a particular chosen signature. The aim is to be able to prove the direct semantic equivalence in the form of lemma 1 below.

Let me summarize what we expect to achieve with the direct conversion.

Goal: lemma 1

Lemma 1 For every $\mathcal{A} \in$ST(XL) there is a structure $\text{CONV}_1(\mathcal{A}) \in$ ST(MSL#) such that:

1. $\mathcal{A}, [\mathbf{v}_1 ... \mathbf{v}_n] \models \varphi$ in XL iff $\text{CONV}_1(\mathcal{A})[^{\mathbf{v}_1 \cdots \mathbf{v}_n}_{v_1 \ldots v_n}](\text{TRANS}(\varphi)[\mathbf{v}_1 ... \mathbf{v}_n]) = T$ in MSL#.
2. $\mathcal{A} \models \varphi$ in XL iff $\text{CONV}_1(\mathcal{A})$ is a model of $\forall\text{TRANS}(\varphi)$ in MSL#

for every sentence φ of XL.

From this lemma 1, or a slightly modified version of it, you can easily obtain as a corollary theorem 1.

Theorem 1 Let $\mathcal{S}* = CONV_1(ST(XL))$

$$\models \varphi \text{ in XL iff } \models_{\mathcal{S}*} \forall\text{TRANS}(\varphi) \text{ in MSL\#}$$

for all sentences φ of the logic XL.

Theorem Lemma 1 implies Theorem 1.

One of the first objectives is to be able to prove a representation theorem for the logic XL. That is, our aim is to prove the following theorem.

Goal: representation theorem

Representation theorem There is a recursive set of sentences Δ,

$$\Delta \subseteq \text{ SENT(MSL\#)}$$

with $\text{CONV}_1(\text{ST(XL)}) \subseteq \text{MOD}(\Delta)$ such that

$$\models \varphi \text{ in the logic XL iff } \Delta \models \forall\text{TRANS}(\varphi) \text{ in MSL\#}$$

for every sentence φ of XL.

From this theorem the enumerability of XL follows as a corollary.

Enumerability of XL The set VAL(XL) is recursively enumerable.

Theorem Representation theorem implies enumerability theorem for the logic XL.

Can we proceed further?

When the original logic XL has a defined concept of consequence we can try to prove a theorem stating the relationship between consequence in XL and consequence in MSL#. In fact, from the theorems proved already we can prove the following theorem.

Theorem 2 There is a recursive set of many-sorted sentences, $\Delta \subseteq \text{SENT}(\text{MSL}\#)$, with $\text{CONV}_1(\text{ST}(\text{XL})) \subseteq \text{MOD}(\Delta)$ such that:

$$\text{If } \text{TRANS}(\Pi) \cup \Delta \models \text{TRANS}(\varphi) \text{ in MSL}\#, \text{ then } \Pi \models \varphi \text{ in XL},$$

for all $\Pi \cup \{\varphi\} \subseteq \text{SENT}(XL)$. (Assuming that at most n variables appear in the translation.)

Theorem Lemma 1 and representation theorem imply theorem 2.

Main theorem

In order to prove the main theorem a reverse conversion of structures will be defined. Two possible situations are going to be considered here: (1) when $\mathcal{S}*$ is axiomatizable and (2) when it is axiomatizable only up to isomorphism or many-sorted equivalence. In either case, we have found a recursive Δ coding $\mathcal{S}*$ and we can define a reverse conversion of structures. In fact, a weaker requirement can be accepted because we only need that the models of Δ cannot be distinguished by translation of formulas of XL.

1. Let $\text{CONV}_1(\text{ST}(\text{XL})) = \text{MOD}(\Delta)$ for a recursive Δ. Thus, we have gotten an affirmative answer to the question of axiomatizability -i.e., our $\mathcal{S}* = \text{MOD}(\Delta)$−and we define a reverse conversion of structures

$$\begin{array}{rrl} \text{CONV}_2: & \text{ST}(\text{MSL}\#) \supseteq \text{MOD}(\Delta) & \longrightarrow \quad \text{ST}(\text{XL}) \\ & \mathcal{C} \in \text{MOD}(\Delta) & \longmapsto \quad \text{CONV}_2(\mathcal{C}) \end{array}$$

   from the relation $\text{CONV}\overline{1}^1$; where, with this definition, you want to be able to prove the reverse semantic equivalence in the form given below.

2. Let $\text{CONV}_1(\text{ST}(\text{XL})) \subseteq \text{MOD}(\Delta)$ for a recursive Δ and let $\text{MOD}(\Delta)$ be so close to $\mathcal{S}*$ that all the elements of this class are either isomorphic or equivalent to converted structures. In this case we can get rid of structures in $\text{MOD}(\Delta) - \mathcal{S}*$ because we can define a function

$$H: \text{ MOD}(\Delta) \longrightarrow \mathcal{S}*$$

   such that for each $\mathcal{C} \in \text{MOD}(\Delta)$, $\mathcal{C} \cong H(\mathcal{C})$ or $\mathcal{C} \equiv H(\mathcal{C})$ and $H(\mathcal{C}) \in \mathcal{S}*$. For each $H(\mathcal{C})$ obtained in this way, we try first to define a reverse structure in $\text{ST}(\text{XL})$. Our function CONV_2 is probably obtained by composition.

$$\begin{array}{rrl} \text{ST}(\text{MSL}\#) \supseteq & \text{MOD}(\Delta) & \text{CONV}_1(\text{ST}(\text{XL})) \subseteq \text{ST}(\text{MSL}\#) \\ & \text{CONV}_2 \searrow & \downarrow \\ & & \text{ST}(\text{XL}) \end{array}$$

   In any case we want the strong version of lemma 2 to be easily proved.

Goal: strong version of lemma 2

Strong version of lemma 2 There is a recursive set Δ of many-sorted sentences, $\Delta \subseteq \text{SENT}(\text{MSL}\#)$, with $\text{CONV}_1(\text{ST}(\text{XL})) \subseteq \text{MOD}(\Delta)$ satisfying: for every many-sorted structure $\mathcal{C} \in \text{MOD}(\Delta)$ there is an XL-structure

$$\text{CONV}_2(\mathcal{C}) \in \text{ST}(\text{XL})$$

such that:

1. $\text{CONV}_2(\mathcal{C}), [\mathbf{v}_1 ... \mathbf{v}_n] \models \varphi$ in XL iff $\mathcal{C}[^{\mathbf{v}_1 ... \mathbf{v}_n}_{v_1 ... v_n}](\text{TRANS}(\varphi)[\mathbf{v}_1 ... \mathbf{v}_n]) = T$ in MSL#
2. $\text{CONV}_2(\mathcal{C}) \models \varphi$ in XL iff $\mathcal{C}$ is a model of $\forall\text{TRANS}(\varphi)$ in MSL#

for all sentences of XL. (Here $v_1, ..., v_n$ are the free variables appearing in the translation, if any, and $v_1, ..., v_n$ are suitable elements in both the original and the converted structure. $\forall\text{TRANS}(\varphi)$ is the universal closure of the translation of φ. $\mathcal{A}, [v_1 ... v_n] \models \varphi$ stands for the truth of the formula φ with the parameters $v_1 ... v_n$, which is a concept of the logic XL. When the translation does not produce free variables, only the item (2) is applicable; in some other occasions (2) follows from (1)).

Now we will see how to use lemma 1 and the strong version of lemma 2 to obtain the main theorem.

Main Theorem

There is a recursive $\Delta \subseteq \text{SENT}(\text{MSL}\#)$ with

$$\text{CONV}_1(\text{ST}(\text{XL})) \subseteq \text{MOD}(\Delta)$$

such that:

$$\Pi \models \varphi \text{ in the logic XL iff } \text{TRANS}(\Pi) \cup \Delta \models \text{TRANS}(\varphi) \text{ in MSL}\#,$$

for all $\Pi \cup \{\varphi\} \subseteq \text{SENT}(XL)$. (Assuming that at most n variables appear in the translation.)

Theorem Lemma 1 and the strong version of lemma 2 imply the main theorem (with the additional assumption that at most n variables — namely, $v_1, ..., v_n$— are used in TRANS, the translation function).

Now we will see how to obtain compactness and Löwenheim-Skolem for XL using the main theorem.

Compactness of XL If $\Pi \models \varphi$ in XL, then there is a finite subset of it, $\Gamma \subseteq \Pi$, such that $\Gamma \models \varphi$.

Theorem The main theorem implies compactness of XL.

Theorem The main theorem implies Löwenheim-Skolem for XL. (With the additional assumptions that the language XL is countable and the reverse conversion, CONV_2, converts countable models into countable models.)

Theorem The main theorem implies enumerability for XL.

Testing a given calculus for XL

When we have a calculus in XL, which we term CAL(XL), we can also try to borrow soundness and completeness of MSL# by applying the results in XL. Therefore, we have to check whether or not the MSL# theory Δ is enough to produce as theorems the logical theorems of the logic XL. In what follows we will assume (1) CAL(XL) is finitary -i.e., deductions are finite, hypotheses of the rules are also finite — (2) CAL(XL) contains the classical propositional calculus and (3) the deduction theorem holds in CAL(XL).

There are several questions we would like to ask, the basic one being as follows: Are the axioms and/or rules of the calculus of XL easily converted by translation into theorems of Δ or derived rules of MSL#?

Therefore, we try to prove our lemma 3.

Goal: lemma 3

Lemma 3 If $\vdash \varphi$ in CAL(XL), then $\Delta \vdash \forall\text{TRANS}(\varphi)$ in MSL# (where Δ is the set of sentences obtained in any version of lemma 2).

From this lemma 3, theorem 3 follows easily.

Theorem 3 $\Pi \vdash \varphi$ in CAL(XL) implies $\text{TRANS}(\Pi) \cup \Delta \vdash \text{TRANS}(\varphi)$ in MSL#, for all $\Pi \cup \{\varphi\} \subseteq SENT(XL)$ (where Δ is the set of sentences obtained in any version of lemma 2).

Proof of Theorem 3 Lemma 3 implies theorem 3 (assuming the properties of the calculus and the behavior of the function TRANS stated above).

Theorem Theorem 2 and theorem 3 imply soundness of CAL(XL).

We have proved already the soundness of the calculus and we want to use the metaproperties of MSL# to prove completeness of CAL(XL) as well. When the calculus being tested is very similar to the many-sorted calculus, we can try to prove directly the lemma on calculus equivalence. Thus, we try to prove lemma 4.

Goal: lemma 4

Lemma 4 $\vdash \varphi$ in CAL(XL) if and only if $\Delta \vdash \forall\text{TRANS}(\varphi)$ in MSL#, all $\varphi \in \text{SENT(XL)}$ (where Δ comes from lemma 2, in any of its versions).

From lemma 4 and certain assumptions about the behavior of the calculus, theorem 4 follows easily.

Theorem 4 $\Pi \vdash \varphi$ in CAL(XL) if and only if $\text{TRANS}(\Pi) \cup \Delta \vdash \text{TRANS}(\varphi)$ in MSL#, all $\Pi \cup \{\varphi\} \subseteq \text{SENT(XL)}$ (where Δ is the set of sentences obtained in any version of lemma 2).

Theorem Lemma 4 implies theorem 4 (assuming the same as in 1.4.1).

Theorem Theorem 4 and the main theorem imply soundness and completeness of CAL(XL).

Proof

Assume theorem 4 and the main theorem and use completeness and soundness of MSL#.

$$\begin{array}{rcl} \Pi \models \varphi & \overset{(*)}{\Longleftrightarrow} & \text{TRANS}(\Pi) \cup \Delta \vDash \text{TRANS}(\varphi) \\ & & \Updownarrow (**) \\ \Pi \vdash_{CAL(XL)} \varphi & \underset{(***)}{\Longleftrightarrow} & \text{TRANS}(\Pi) \cup \Delta \vdash_{MSL\#} \text{TRANS}(\varphi) \end{array}$$

If you look at the schema, (*) is the main theorem, (**) is completeness and soundness of MSL, (***) is theorem 4. Therefore, following the arrow in the clockwise direction you obtain completeness,

$$\Pi \models \varphi \Rightarrow \Pi \vdash_{CAL(XL)} \varphi$$

and in the reverse sense, soundness,

$$\Pi \vdash_{CAL(XL)} \varphi \Rightarrow \Pi \models \varphi$$

Goal: lemma 5

Let $\mathcal{A}_{\mathbf{CAN}}$ be the canonical model of XL. If $\text{CONV}_1(\mathcal{A}_{\mathbf{CAN}}) \models \forall\text{TRANS}(\varphi)$, then $\vdash_{CAL(XL)}$.

From lemma 5 and lemma 3 we prove lemma 4.

Theorem Lemma 5 and lemma 3 imply lemma 4.

7.2 Higher-Order Logic as Many-Sorted Logic

The many-sorted signature Σ^{SOL}

Let SORT be the smallest set with the following properties:

1. $0, 1 \in$SORT. (0 is the Boolean and 1 is the individuals sort.)
2. For each natural number $n \geq 1$, we have $\langle 0, 1, \underset{n}{\ldots}, 1\rangle \in$ SORT. ($\langle 0, 1, \underset{n}{\ldots}, 1\rangle$ is the sort of domains of n-ary relations on individuals. To help your intuition, we take it to be the type of n-ary relations on individuals as well.)

The signature $\Sigma^{SOL} = \langle \text{SORT}, \text{FUNC} \rangle$ is such that:

1. SORT = SORT(Σ^{SOL}) is the set defined above.

2. FUNC = FUNC(Σ^{SOL}) is a function which takes as arguments the elements of OPER.CONS of second order logic (that means, only relation symbols on individuals, including equality for every sort; that is an equality of type $\langle 0, 1, 1 \rangle$, another of type $\langle 0, \langle 0, 1 \rangle, \langle 0, 1 \rangle \rangle$, etc.) enlarged with the new membership relation symbols (we also put in $\neg$ and $\vee$ as we did in many-sorted logic). Call the resulting set of symbols OPER.SYM^{SOL}.

Now we define

$$\text{FUNC: OPER.SYM}^{\mathbf{SOL}} \longrightarrow [\mathbf{S}_{\boldsymbol{\omega}}(\text{SORT})]$$

where:

• Almost all the values of FUNC are of the following kind: $\langle 0, 1, \overset{n}{\ldots}, 1 \rangle$. That is, they are relations among individuals whose interpretation will be a characteristic function. We have also values $\langle 0, 0 \rangle$ and $\langle 0, 0, 0 \rangle$ for $\neg$ and $\vee$.

•• For the new membership relations ϵ_n,

$$\text{FUNC}(\epsilon_n) = \langle 0, 1, \overset{n}{\ldots}, 1, \langle 0, 1 \overset{n}{\ldots}, 1 \rangle \rangle$$

Alphabet of MSL^{SOL}

The alphabet of many-sorted logic suited for SOL must contain all the symbols in OPER.SYM^{SOL} (that includes the membership relation symbols ϵ_n) and a countable set of variables for each $i \in$SORT$-\{0\}$. To simplify things, let us use the same variables as in SOL:

- x_1, x_2, ... for variables of sort 1; X_1^1, X_2^1, X_3^1, ... for variables of sort $\langle 0, 1 \rangle$; X_1^2, X_2^2, X_3^2 , ... for variables of sort $\langle 0, 1, 1 \rangle$; ...; X_1^n, X_2^n, X_3^n, ... for variables of sort $\langle 0, 1, \overset{n}{\ldots}, 1 \rangle$; etc.

- Besides all that, we have the existential quantifier: $\exists$.

Expressions The set of expressions of this many-sorted language, MSL^{SOL}, is defined in the same way as that of any other many-sorted language and therefore is different from the second order language only in the following:

1. $X^n x_1 ... x_n$ is no longer a formula in MSL^{SOL} (since it is only a string of variables).

2. $\epsilon_n x_1 ... x_n X^n$, unlike second order logic, is indeed a formula (of course, $x_1, ..., x_n$ should be variables of individual sort).

The syntactical translation

The syntactical translation from SOL to this MSL^{SOL} leaves every formula the same except the atomic formulas of the kind mentioned above; i.e.:

$$\text{TRANS}^{SOL}(X^n x_1 ... x_n) = \epsilon_n x_1 ... x_n X^n$$

Structures

Direct conversion of structures Let us define

$$\begin{array}{rcl} \text{CONV}_1 : \text{ST(SOL)} & \longrightarrow & \text{ST}(\text{MSL}^{SOL}) \\ \mathcal{A} & \longmapsto & \mathcal{A}* \end{array}$$

where, when

$$\mathcal{A} = \langle \mathbf{A}, \langle \mathbf{A}_n \rangle_{n \geq 1}, \langle C^{\mathcal{A}} \rangle_{C \in \text{OPER.CONS}} \rangle$$

we take

$$\mathcal{A}* = \langle \mathbf{A}, \langle \mathbf{A}_n \rangle_{n \geq 0}, \langle f^{\mathcal{A}} \rangle_{f \in OPER.SYM} \rangle$$

with $\mathbf{A}_0 = \{T, F\}$ and:

1. when $C \in \text{OPER.CONS} \cap \text{OPER.SYM}^{SOL}$ we put in $\mathcal{A}*$ the characteristic function of $\mathcal{C}^{\mathcal{A}}$,

$$C^{\mathcal{A}*} = \{\langle \mathbf{x}_1, ..., \mathbf{x}_n, \mathbf{X} \rangle \in \mathbf{A}^n \times \mathbf{A}_0 |\ \mathbf{X} = T \text{ iff } \langle \mathbf{x}_1, ..., \mathbf{x}_n \rangle \in C^{\mathcal{A}}\}.$$

2. when $C \in \text{OPER.SYM}^{SOL} - \text{OPER.CONS}$, C is either a connective or membership. Thus,

$$\epsilon_n^{\mathcal{A}*} = \{\langle \mathbf{x}_1, ..., \mathbf{x}_n, \mathbf{X}^n, \mathbf{Y} \rangle \in \mathbf{A}^n \times \mathbf{A}_n \times \mathbf{A}_0 |\ \mathbf{Y} = T \text{ iff } \langle \mathbf{x}_1, ..., \mathbf{x}_n \rangle \in \mathbf{X}^n\}$$

Besides this, all the connectives are standard.

Lemma 1 (b). For every second order general structure, $\mathcal{A} \in \text{ST(SOL)}$**,** there is a many-sorted structure, $\text{CONV}_1(\mathcal{A}) \in \text{CONV}_1(\text{ST(SOL)})$, such that

$$\mathcal{A} \text{ is a model of } \varphi \text{ iff } \text{CONV}_1(\mathcal{A}) \text{ is a model of } \text{TRANS}^{SOL}(\varphi)$$

for every sentence φ of SOL.

Theorem 1 For every sentence φ of SOL:

$$\models_{\mathcal{G}.\mathcal{S}} \varphi \text{ in SOL iff } \models_{\mathcal{S}*} \text{ TRANS}(\varphi) \text{ in } \text{MSL}^{SOL}$$

Reverse conversion of structures We will make the reverse conversion of structures in four steps.

(A) The structures we want to work with are MSL^{SOL} models of extensionality and comprehension with disjoint universes. So first we express these properties in MSL^{SOL}.

$$\Delta = \{\text{Ext}^{(n)} |\ n \geq 1\} \cup \{\forall\text{Comph}_{\varphi}^{(n)} |\ n \geq 1 \& \varphi \in \text{FORM(MSL)}\}$$
$$\cup \{\text{Disj}_{n,m} |\ n \neq m \&\ n, m \geq 1\}$$

where:

$$\text{Ext}^{(n)} = \forall X^n Y^n (\forall x_1 ... x_n (\epsilon_n x_1 ... x_n X^n \leftrightarrow \epsilon_n x_1 ... x_n Y^n) \rightarrow X^n = Y^n)$$

$$\forall\text{Comph}_{\varphi}^{(n)} = \forall \exists X^n \forall x_1 ... x_n (\epsilon_n x_1 ... x_n X^n \leftrightarrow \varphi), \text{ where } \quad X^n \notin FREE(\varphi)$$

$$\text{Disj}_{n,m} = \neg \exists X^n Y^m X^n = Y^m \text{ (for } n \neq m)$$

(B) From a many-sorted structure $\mathcal{A}$ of signature Σ^{SOL} which is a model of Δ, we pass to another many-sorted structure $\mathcal{B}$ of the same signature but where membership is the typical second order relation. Thus, we define a function

$$H : \text{MOD}(\Delta) \longrightarrow \text{CONV}_1(\text{ST(SOL)}) \tag{7.1}$$

If $\mathcal{A}$ is a model of Δ, $\mathcal{B} = H(\mathcal{A})$ is basically a general second order structure with the genuine membership relation.

(C) We define an isomorphism f from $\mathcal{A}$ onto $\mathcal{B}$ which is also the identity on universes of sort 0 and 1.

(D) From a MS-structure $\mathcal{B} \in \text{CONV}_1(\text{ST(SOL)})$ of the kind mentioned in step two, we define a SOL structure by an easy make-up transformation; namely, erasing the membership relation and adapting the signature to a second order one. Also characteristic functions turn back to relations. This is just to reverse the transformation done in CONV_1.

Theorem There is a function

$$H : \text{MOD}(\Delta) \longrightarrow \text{CONV}_1(\text{ST(SOL)})$$

sending every many-sorted model of Δ, $\mathcal{A} \in \text{MOD}(\Delta)$, to another many-sorted structure of the same signature such that $\mathcal{A} \cong H(\mathcal{A})$.

Definition of CONV_2 Let us define CONV_2 as $\text{CONV}\bar{1}^1 \circ H$, where H is the function (7.1) mentioned above.

Strong version of lemma 2 (item (b)) There is a recursive $\Delta \subseteq \text{SENT(MSL)}$ with $\text{CONV}_1(\text{ST}(SOL)) \subseteq \text{MOD}(\Delta)$ such that, for every many-sorted structure $\mathcal{C} \in \text{MOD}(\Delta)$ there is a SOL-structure $\text{CONV}_2(\mathcal{C}) \in \text{ST(SOL)}$ such that

$$\text{CONV}_2(\mathcal{C}) \text{ is a model of } \varphi \text{ iff } \mathcal{C} \text{ is a model of } \text{TRANS}^{SOL}(\varphi)$$

for all sentences of SOL.

The equivalence SOL-MSL^{SOL}

Main theorem $\Pi \models_{\mathcal{G}.\mathcal{S}} \varphi$ in second order logic iff $\text{TRANS}^{SOL}(\Pi) \cup \Delta \models \text{TRANS}^{SOL}(\varphi)$ in many-sorted logic.

Compactness of SOL with general semantics If $\Pi \models_{\mathcal{G}.\mathcal{S}} \varphi$ in second order logic, there is a finite Γ, $\Gamma \subseteq \Pi$, such that $\Gamma \models_{\mathcal{G}.\mathcal{S}} \varphi$.

Enumerability of SOL with general semantics The set $\text{VAL}_{\mathcal{G}.\mathcal{S}}(\text{SOL})$ of valid formulas in general semantics is recursively enumerable.

Löwenheim-Skolem of SOL with general semantics If a set of sentences of SOL has a general model, then it has a countable general model.

The enumerability and compactness theorems taken together tell us that a strongly complete calculus exists. But we want to know more, we want to test for completeness an actual one: our calculus C_2.

Theorem 4. $\Pi \vdash \varphi$ in second order calculus C_2 iff $\text{TRANS}^{SOL}(\Pi) \cup \Delta \vdash \text{TRANS}^{SOL}(\boldsymbol{\varphi})$ in many-sorted calculus, MSL.

Soundness and completeness of SOL

$$\Pi \models_{\mathcal{G}.\mathcal{S}} \varphi \text{ iff } \Pi \vdash_{C_2} \varphi \text{ for all } \Pi \cup \{\varphi\} \subseteq \text{SENT}(L_2)$$

7.3 Modal Logic

A formal language for PML

Alphabet: The alphabet of PML contains:

1. A set ATOM.PROP of atomic formulas with $\bot \in$ ATOM.PROP, $\bot$ is *falsity*. We will use the letter P for elements of ATOM.PROP; when needed, we will add subscripts: P_0, P_1, P_2, etc.
2. The connectives: $\neg, \vee, \wedge, \rightarrow$ and $\leftrightarrow$.
3. Modal operators: $\Diamond$ and $\Box$.
4. Parentheses:) and (. These are improper symbols.

Formulas: Formulas of PML are finite strings of symbols and they are inductively defined from the symbols in the alphabet by using the rules of the calculus of formulas given below. The set FORM(PML) of formulas of PML is the smallest set satisfying:

- (F1) ATOM.PROP $\subseteq$ FORM(PML)**.** In particular, $\perp \in$ FORM(PML).
- (F2) If $\varphi \in$FORM(PML) and $\psi \in$FORM(PML), then $\neg\varphi$, $(\varphi \vee \psi)$, $(\varphi \wedge \psi)$, $(\varphi \rightarrow \psi)$ and $(\varphi \leftrightarrow \psi)$ are in FORM(PML).
- (F3) If $\varphi \in$FORM(PML), then $\Diamond\varphi$ and $\Box\varphi$ are in FORM(PML).

Modal propositional logics

Definition A *modal logic* is any $\mathbf{B}$ such that $\vdash_{\mathbf{PC}}$ (PML) $\subseteq \mathbf{B} \subseteq$ FORM(PML) and $\mathbf{B}$ is closed under the rule of *modus ponens* (MP); *i.e.*, if $\varphi \in \mathbf{B}$ and $\varphi \rightarrow \psi \in \mathbf{B}$, then $\psi \in \mathbf{B}$.

Definition Any member of the modal logic $\mathbf{B}$ is a *theorem* of $\mathbf{B}$ and conversely. In symbols, $\varphi \in \mathbf{B}$ iff $\vdash_{\mathbf{B}} \varphi$.

Definition A formula φ is *deducible* in the logic $\mathbf{B}$ from the set Γ (written as $\Gamma \vdash_{\mathbf{B}} \varphi$) iff there is a finite subset of Γ, say $\{\gamma_1, ..., \gamma_n\}$, such that $\gamma_1 \wedge ... \wedge \gamma_n \rightarrow \varphi \in \mathbf{B}$. (When $n = 0$ we agree that $\gamma_1 \wedge ... \wedge \gamma_n \rightarrow \varphi \equiv \varphi$. Thus, we again get: $\vdash_{\mathbf{B}} \varphi$ iff $\varphi \in \mathbf{B}$.)

Proposition Some basic properties of the operator $\vdash_{\mathbf{B}}$ are:

1. *Monotony on hypotheses* : If $\Gamma \subseteq \Pi$ and $\Gamma \vdash_{\mathbf{B}} \varphi$, then $\Pi \vdash_{\mathbf{B}} \varphi$.
2. *Monotony on logics* : If $\mathbf{B} \subseteq A$ and $\Gamma \vdash_{\mathbf{B}} \varphi$, then $\Gamma \vdash_A \varphi$.
3. *Generalized (MP):* If $\Gamma \vdash_{\mathbf{B}} \varphi$ and $\Gamma \vdash_{\mathbf{B}} \varphi \rightarrow \psi$, then $\Gamma \vdash_{\mathbf{B}} \psi$.
4. *Deduction theorem :* $\Gamma \cup \{\varphi\} \vdash_{\mathbf{B}} \psi$ iff $\Gamma \vdash_{\mathbf{B}} \varphi \rightarrow \psi$.

Definition Given a decidable set of formulas Γ and a finite set of rules $\mathbf{H}$, a *proof of φ in the calculus* $\Gamma_{\mathbf{H}}$ is a finite sequence of formulas each of which either is in $\vdash_{\mathbf{PC}}$ (PML)$\cup\Gamma$ or has been obtained from previous formulas in the sequence by any of the rules of ($\mathbf{H}$) or (MP). When there is a proof of φ in $\Gamma_{\mathbf{H}}$ we write $\vdash_{\Gamma_{\mathbf{H}}} \varphi$.

Definition A logic $\mathbf{B}$ is *axiomatizable* iff there is a calculus $\Gamma_{\mathbf{H}}$ such that $\mathbf{B} = \{\varphi | \vdash_{\Gamma_{\mathbf{H}}} \varphi\}$.

A logic $\mathbf{B}$ is *finitely axiomatizable* iff $\mathbf{B}$ is axiomatizable by a calculus $\Gamma_{\mathbf{H}}$ and there is a finite set of schema axioms producing Γ as particular instances.

Definition A logic **B** is *consistent* iff no $\vdash_{\mathbf{B}} \bot$.

Consistency and maximal consistency
Let **B** be a modal logic and $\Pi \subseteq$ FORM(PML).

1. Π is **B**-*consistent* iff no $\Pi \vdash_{\mathbf{B}} \bot$.
2. Π is **B**-*contradictory* iff $\Pi \vdash_{\mathbf{B}} \bot$.
3. Π is *maximally B-consistent* iff Π is **B**-consistent and for all $\psi \in$ FORM(PML) :

$$If\ \psi \notin \Pi \text{ then } \Pi \cup \{\psi\} \text{ is } \mathbf{B}\text{-contradictory}$$

Theorems on consistency

1. If Π is **B**-consistent and $\Gamma \subseteq \Pi$ then Γ is **B**-consistent.
2. If Π is **B**-contradictory and $\Pi \subseteq \Gamma$ then Γ is **B**-contradictory.
3. Π is **B**-consistent iff every finite subset of Π is **B**-consistent.

Theorems on maximal consistency Let $\Delta \subseteq$ FORM(PML) be a maximal **B**-consistent set, then the items (1) through (5) stated in Chapter VI for MSL are also true for PML. Namely,

1. If $\Delta \vdash_{\mathbf{B}} \varphi$ then $\varphi \in \Delta$.
2. If $\vdash_{\mathbf{B}} \varphi$ then $\varphi \in \Delta$
3. $\neg\varphi \in \Delta$ iff $\varphi \notin \Delta$.
4. $\varphi \vee \psi \in \Delta$ iff $\varphi \in \Delta$ or $\psi \in \Delta$.
5. If $\Delta \cup \{\varphi\} \vdash_{\mathbf{B}} \psi$ and $\Delta \cup \{\psi\} \vdash_{\mathbf{B}} \varphi$, then $\varphi \in \Delta$ iff $\psi \in \Delta$.

Lindenbaum lemma
If Γ is **B**-consistent, then there is a maximally **B**-consistent set Δ such that $\Gamma \subseteq \Delta$.

More theorems on maximal consistency Let $\Delta \subseteq$ FORM(PML) be a maximal **B**-consistent set.

1. $\mathbf{B} = \bigcap\{\Gamma | \Gamma \text{ is maximally } \mathbf{B}\text{-consistent}\}$.
2. If $\Delta \subseteq \Gamma$ and Γ is maximally **B**-consistent, then $\Gamma = \Delta$.
3. Let $\Sigma \subseteq$ FORM(PML). If for every maximally **B**-consistent set Γ such that $\Sigma \subseteq \Gamma$, $\varphi \in \Gamma$ holds, then $\Sigma \vdash_{\mathbf{B}} \varphi$.

Normal modal logics

Normal modal logics form a class of modal logics whose distinguishing characteristic is to contain all the occurrences of the axiom schema **K** and **Df**$\Diamond$ and to be closed as well under the modal rule of necessitation. The smallest normal modal logic is the logic **K**, which we will define later.

Definition The *calculus* $\{\mathbf{K}, \mathbf{Df}\Diamond\}_{(\mathbf{N})}$ is obtained by adding to your favorite PC the schema axioms **K**, **Df**$\Diamond$ and the rule (**N**).

The new axioms and rules are:

- **K** *Distribution of* $\Box$ *with respect to* $\rightarrow$.

$$\Box(\boldsymbol{\varphi} \rightarrow \boldsymbol{\psi}) \rightarrow (\Box\boldsymbol{\varphi} \rightarrow \Box\boldsymbol{\psi})$$

- **Df**$\Diamond$ *Definition of the diamond*

$$\Diamond\varphi \leftrightarrow \neg\Box\neg\varphi$$

- (**N**) *Necessitation*

$$\text{If } \vdash \varphi, \text{ then } \vdash \Box\varphi$$

Definition The *logic* **K** is the smallest modal logic containing the schema axioms **Df**$\Diamond$ and **K** and being closed under the rule (**N**); i.e., if $\varphi \in K$, then $\Box\varphi \in K$. So

$$\mathbf{K} = \{\varphi | \vdash_{\{\mathbf{K},\mathbf{Df}\Diamond\}_{(\mathbf{N})}} \varphi\}$$

Definition A *normal modal logic* is any modal logic containing the logic K which is closed under the rules of the calculus $\{\mathbf{K}, \mathbf{Df}\Diamond\}_{(\mathbf{N})}$.

There are many normal logics, but the best known and studied are the axiomatizable normal modal logics obtained when adding to the logic **K** any combination of the schema axioms

D $\Box\varphi \rightarrow \Diamond\varphi$
T $\Box\varphi \rightarrow \varphi$
B $\varphi \rightarrow \Box\Diamond\varphi$
4 $\Box\varphi \rightarrow \Box\Box\varphi$
5 $\Diamond\varphi \rightarrow \Box\Diamond\varphi$
and closing the extended set over the rules of the calculus $\{\mathbf{K}, \mathbf{Df}\Diamond\}_{(\mathbf{N})}$.

Kripke models

Definition of Kripke's models

$\mathcal{M} = \langle \mathbf{W}, \mathbf{R}, \langle P^{\mathcal{M}} \rangle_{P \in \text{ATOM.PROP}} \rangle$ is a PM-structure or a Kripke model iff

1. $\mathbf{W} \neq \emptyset$ is a non-empty set whose elements are called worlds or states.

2. $\mathbf{R}$ is a binary relation on worlds; that is, $\mathbf{R} \subseteq \mathbf{W} \times \mathbf{W}$.

3. For each $P \in$ ATOM.PROP, $P^{\mathcal{M}}$ is a set of states; that is, $P^{\mathcal{M}} \subseteq \mathbf{W}$. (The idea behind this is that in $P^{\mathcal{M}}$ are all the worlds where P is true.)

Interpretation When a PM-structure $\mathcal{M}$ is given, each formula φ gets an interpretation, $\mathcal{M}(\varphi)$, easily seen as a truth value for every world in $\mathcal{M}$. The interpretation of φ, $\mathcal{M}(\varphi)$, is the set of worlds or states where φ is true. Clearly the structure $\mathcal{M}$ provides us with such an interpretation for the atomic formulas in ATOM.PROP. Recursively, we are going to extend the definition to cover any formula.

$$\begin{array}{ll} \mathcal{M}(P) = P^{\mathcal{M}} & \mathcal{M}(\bot) = \emptyset \\ \mathcal{M}(\neg\varphi) = \mathbf{W} - \mathcal{M}(\varphi) & \mathcal{M}(\varphi \vee \psi) = \mathcal{M}(\varphi) \cup \mathcal{M}(\psi) \\ \mathcal{M}(\varphi \wedge \psi) = \mathcal{M}(\varphi) \cap \mathcal{M}(\psi) & \mathcal{M}(\varphi \rightarrow \psi) = \mathcal{M}(\neg\varphi) \cup \mathcal{M}(\psi) \end{array}$$

$$\mathcal{M}(\varphi \leftrightarrow \psi) = (\mathcal{M}(\varphi) \cap \mathcal{M}(\psi)) \cup (\mathbf{W} - (\mathcal{M}(\varphi) \cup \mathcal{M}(\psi)))$$

$$\mathcal{M}(\Diamond\varphi) = \{s \in \mathbf{W} \mid \exists t \in \mathbf{W}(\langle s,t\rangle \in R \& t \in \mathcal{M}(\varphi))\}$$

$$\mathcal{M}(\Box\varphi) = \{s \in \mathbf{W} \mid \forall t \in \mathbf{W}(\langle s,t\rangle \in R \Rightarrow t \in \mathcal{M}(\varphi))\}$$

Definition of the canonical model of B. Let $\mathbf{B}$ be a consistent normal modal logic. We define the *canonical model of* B as the PM-structure
$\mathcal{A}_{\mathbf{B}} = \langle \mathbf{W_B}, \mathbf{R_B}, \langle P^{\mathcal{A}_{\mathbf{B}}}\rangle_{P \in \text{ATOM.PROP}}\rangle$
where:

1. $\mathbf{W_B} = \{\mathbf{s} \subseteq \text{FORM}(PML) \mid \mathbf{s}$ is maximally $\mathbf{B}$-consistent$\}$

2. $\mathbf{R_B} \subseteq \mathbf{W_B} \times \mathbf{W_B}$ is defined by: $\langle \mathbf{s}, \mathbf{t}\rangle \in \mathbf{R_B}$ iff $\{\varphi \mid \Box\varphi \in \mathbf{s}\} \subseteq \mathbf{t}$

3. $P^{\mathcal{A}_{\mathbf{B}}} = \{\mathbf{s} \in \mathbf{W_B} \mid P \in \mathbf{s}\}$ for all $P \in$ ATOM.PROP

Proposition Let $\mathbf{B}$ be a consistent normal modal logic and $\mathcal{A}_{\mathbf{B}}$ its canonical model. For every $\varphi \in \text{FORM}(PML)$ and $\mathbf{s} \in \mathbf{W_B}$:

$$\Box\varphi \in \mathbf{s} \text{ iff for every } \mathbf{t} \in \mathbf{W_B}, \text{ if } \langle \mathbf{s}, \mathbf{t}\rangle \in \mathbf{R_B} \text{ then } \varphi \in \mathbf{t}$$

Definitions

1. A normal modal logic $\mathbf{B}$ is: *sound with respect to a given class of structures* $\mathcal{C}$ iff:
$$\text{If } \Gamma \vdash_{\mathbf{B}} \varphi \text{ implies } \Gamma \models_{\mathcal{C}} \varphi$$

2. A normal modal logic $\mathbf{B}$ is: *strongly complete with respect to a given class of structures* $\mathcal{C}$ iff:
$$\text{If } \Gamma \models_{\mathcal{C}} \varphi \text{ implies } \Gamma \vdash_{\mathbf{B}} \varphi$$

3. A normal modal logic **B** is: *weakly complete with respect to a given class of structures* $\mathcal{C}$ iff:
$$\text{If } \models_{\mathcal{C}} \varphi \text{ implies } \vdash_{\mathbf{B}} \varphi$$

4. A normal modal logic **B** is: *determined by a given class of structures* $\mathcal{C}$ iff: **B** is sound and weakly complete with respect to $\mathcal{C}$.

Properties of the accessibility relation

1. **K** and **Df**$\Diamond$ are valid in the whole class of Kripke models. Furthermore, the rule (**N**) preserves consequence.

2. **D** is valid in the class of models where **R** is *serial* ; *i.e.*, for every $\mathbf{s} \in \mathbf{W}$, there is $\mathbf{t} \in \mathbf{W}$ such that $\langle \mathbf{s}, \mathbf{t} \rangle \in \mathbf{R}$.

3. **T** is valid in the class of models where **R** is *reflexive.*

4. **B** is valid when **R** is *symmetric.*

5. **4** is valid for *transitive* R.

6. **5** is valid for *euclidean* **R**; i.e., for every $\mathbf{s}, \mathbf{t}, \mathbf{z} \in \mathbf{W}$, if $\langle \mathbf{s}, \mathbf{t} \rangle \in \mathbf{R}$ and $\langle \mathbf{s}, \mathbf{z} \rangle \in \mathbf{R}$, then $\langle \mathbf{t}, \mathbf{z} \rangle \in \mathbf{R}$.

Corollary: soundness of the normal logics obtained from the axioms D, T, B, 4 and 5

1. The logic **K** is sound in the whole class of Kripke models.

2. The logic **K**D is sound in the class of models where **R** is *serial.*

3. The logic **K**T is sound in the class of models where **R** is *reflexive.*

4. The logic **K**B is sound in the class of models where **R** is *symmetric.*

5. The logic **K**4 is sound in the class of models where **R** is *transitive.*

6. The logic **K**5 is sound in the class of models where **R** is *euclidean.*

7.4 Propositional Modal Logic as Many-Sorted Logic

The formal many-sorted logic

The signature $\Sigma^{\blacksquare}$ Let SORT $= \{0, 1, \langle 0, 1 \rangle\}$.
The signature $\Sigma^{\blacksquare} = \langle$SORT,FUNC$\rangle$ is such that:

1. SORT is the set defined above.

2. FUNC$^{\blacksquare}$ is a function which takes as arguments the elements of OPER.SYM$^{\blacksquare}$ consisting of: ATOM.PROP of PML, the equalities among individuals and among sets, E_1 and E_2, a binary relation S among individuals, the connectives and the membership relation ϵ_1.

$$\text{FUNC}^{\blacksquare} : \text{OPER.SYM}^{\blacksquare} \longrightarrow [\mathbf{S}_{\boldsymbol{\omega}}(\text{SORT})]$$

where:

(a) FUNC$^{\blacksquare}(\bot) = \langle 0 \rangle$
(b) FUNC$^{\blacksquare}(\neg) = \langle 0, 0 \rangle$
(c) FUNC$^{\blacksquare}(\rightarrow) = \langle 0, 0, 0 \rangle$ and the same for the rest of the binary connectors
(d) FUNC$^{\blacksquare}(S) = \langle 0, 1, 1 \rangle$
(e) FUNC$^{\blacksquare}(P) = \langle 0, 1 \rangle$, for each $P \in$ ATOM.PROP
(f) FUNC$^{\blacksquare}(\epsilon_1) = \langle 0, 1, \langle 0, 1 \rangle \rangle$
(g) FUNC$^{\blacksquare}(E_1) = \langle 0, 1, 1 \rangle$
(h) FUNC$^{\blacksquare}(E_2) = \langle 0, \langle 0, 1 \rangle, \langle 0, 1 \rangle \rangle$

Alphabet of MSL$^{\blacksquare}$ The alphabet of a many-sorted language suited for PML must contain all operation symbols in OPER.SYM$^{\blacksquare}$ and a countable set of variables for each $i \in$ SORT$-\{0\}$.

We will use lower-case for individual variables $(u, v, w, ...,\ u_1,\ u_2,\ u_3, ...)$ and upper-case for set variables $(X, Y, Z, ...,\ X_1,\ X_2,\ X_3, ...for$ unary relation variables of sort $\langle 0, 1 \rangle)$.

As well as all that, we have the universal and existential quantifiers: $\forall$ and $\exists$.

Translating function

From PML we pass to a many-sorted language of signature $\Sigma^{\blacksquare}$.

Definition With every φ of PML and individual variable u we associate as its translation a formula TRANS$^{\blacksquare}(\varphi)[u]$ in the many-sorted language, defined by:

- (F1) TRANS$^{\blacksquare}(\bot)[u] \equiv u \neq u$
- TRANS$^{\blacksquare}(P)[u] \equiv Pu$ for each $P \in$ ATOM.PROP
- (F2) TRANS$^{\blacksquare}(\neg\varphi)[u] \equiv \neg$TRANS$^{\blacksquare}(\varphi)[u]$
- TRANS$^{\blacksquare}(\varphi \rightarrow \psi)[u] \equiv$ TRANS$^{\blacksquare}(\varphi)[u] \rightarrow$TRANS$^{\blacksquare}(\psi)[u]$ (similarly for the rest of the connectors)
- (F3) TRANS$^{\blacksquare}(\Box\varphi)[u] \equiv \forall w(Suw \rightarrow$TRANS$^{\blacksquare}(\varphi)[w])$ where w must be a new variable; the first one in a chosen ordering
- TRANS$^{\blacksquare}(\Diamond\varphi)[u] \equiv \exists w(Suw \wedge$TRANS$^{\blacksquare}(\varphi)[w])$ where w must be a new variable

Translation of the axioms for normal logics **K**. $\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$

$$\text{TRANS}^{\blacksquare}(\mathbf{K})[u] \equiv \forall v(Suv \to (\text{TRANS}^{\blacksquare}(\varphi)[v] \to \text{TRANS}^{\blacksquare}(\psi)[v]))$$
$$\to (\forall v(Suv \to \text{TRANS}^{\blacksquare}(\varphi)[v]) \to \forall v(Suv \to \text{TRANS}^{\blacksquare}(\psi)[v]))$$

Df$\Diamond$. $\Diamond\varphi \leftrightarrow \neg\Box\neg\varphi$

$$\text{TRANS}^{\blacksquare}(\mathbf{Df}\Diamond) \equiv \exists v(Suv \wedge TRANS^{\blacksquare}(\varphi)[v]) \leftrightarrow \neg\forall v(Suv \to \neg\text{TRANS}^{\blacksquare}(\varphi)[v])$$

D. $\Box\varphi \to \Diamond\varphi$

$$\text{TRANS}^{\blacksquare}(\mathbf{D})[u] \equiv \forall v(Suv \to \text{TRANS}^{\blacksquare}(\varphi)[v]) \to \exists v(Suv \wedge \text{TRANS}^{\blacksquare}(\varphi)[v])$$

T. $\Box\varphi \to \varphi$

$$\text{TRANS}^{\blacksquare}(\mathbf{T})[u] \equiv \forall v(Suv \to \text{TRANS}^{\blacksquare}(\varphi)[v]) \to \text{TRANS}^{\blacksquare}(\varphi)[u]$$

B. $\varphi \to \Box\Diamond\varphi$

$$\text{TRANS}^{\blacksquare}(B)[u] \equiv \text{TRANS}^{\blacksquare}(\varphi)[u] \to \forall v(Suv \to \exists w(Svw \wedge \text{TRANS}^{\blacksquare}(\varphi)[w]))$$

4. $\Box\varphi \to \Box\Box\varphi$

$$\text{TRANS}^{\blacksquare}(4)[u] \equiv \forall v(Suv \to \text{TRANS}^{\blacksquare}(\varphi)[v]) \to \forall v(Suv \to \forall w(Svw \to \text{TRANS}^{\blacksquare}(\varphi)[w]))$$

5. $\Diamond\varphi \to \Box\Diamond\varphi$

$$\text{TRANS}^{\blacksquare}(5)[u] \equiv \exists v(Suv \wedge \text{TRANS}^{\blacksquare}(\varphi)[v]) \to \forall v(Suv \to \exists w(Svw \wedge \text{TRANS}^{\blacksquare}(\varphi)[w]))$$

Theorems

1. $\vdash_{MSL} \forall u\ \text{TRANS}^{\blacksquare}(\mathbf{K})[u]$
2. $\vdash_{MSL} \forall u \text{TRANS}^{\blacksquare}(\mathbf{Df}\Diamond)[u]$
3. If $\vdash_{MSL} \forall u \text{TRANS}^{\blacksquare}(\varphi)[u]$, then $\vdash_{MSL} \forall u \text{TRANS}^{\blacksquare}(\Box\varphi)[u]$ (condition related to the modal rule of necessitation)
4. If $\vdash_{MSL} \forall u(\text{TRANS}^{\blacksquare}(\gamma_1)[u] \wedge ... \wedge \text{TRANS}^{\blacksquare}(\gamma_n)[u] \to \text{TRANS}^{\blacksquare}(\varphi)[u])$ then $\vdash_{MSL} \forall u(\text{TRANS}^{\blacksquare}(\Box\gamma_1)[u] \wedge ... \wedge \text{TRANS}^{\blacksquare}(\Box\gamma_n)[u] \to \text{TRANS}^{\blacksquare}(\Box\varphi)[u])$. (This is the condition related to the modal derived rule (KN))

General structures and frames built on PM-structures

Frames built on PM-structures Given a PM-structure $\mathcal{A}$,

$$\mathcal{A} = \langle \mathbf{W}, \mathbf{R}, \langle P^{\mathcal{A}} \rangle_{P \in \text{ATOM.PROP}} \rangle$$

we say that $\mathcal{AF}$ is a frame built on $\mathcal{A}$ iff

$$\mathcal{AF} = \langle \mathbf{W}, \mathbf{W}', \mathbf{R}, \langle P^{\mathcal{A}} \rangle_{P \in \text{ATOM.PROP}} \rangle$$

where:

1. $\mathbf{W}' \subseteq \wp\mathbf{W}$
2. $P^{\mathcal{A}} \in \mathbf{W}'$, for all $P \in$ATOM.PROP

General structures built on PM-structures Given a PM-structure $\mathcal{A}$ we say that $\mathcal{AG}$ is a general structure built on $\mathcal{A}$ iff

$$\mathcal{AG} = \langle \mathbf{W}, \mathbf{W}', \mathbf{R}, \langle P^{\mathcal{A}} \rangle_{P \in ATOM.\text{PROP}} \rangle$$

satisfies:

The set $\mathbf{W}'$ includes DEF($\mathbf{A}$,PML). If we abbreviate this set as DEF, the conditions are:

1. $\emptyset \in$ DEF and $\mathbf{W} \in$ DEF
2. $\{P^{\mathcal{A}} |\ P \in$ ATOM.PROP$\} \subseteq$ DEF
3. $\forall \mathbf{TS}(\mathbf{T}, \mathbf{S} \in \text{DEF} \Rightarrow \mathbf{T} \cup \mathbf{S},\ \mathbf{W} - \mathbf{T} \in \text{DEF})$
4. $\mathbf{T}(\mathbf{T} \in \text{DEF} \Rightarrow Dom(\mathbf{R} \cap (\mathbf{W} \times \mathbf{T})) \in \text{DEF})$
   In addition,
5. $\forall \mathbf{w}(\mathbf{w} \in \mathbf{W} \Rightarrow \{\mathbf{w}\} \in \mathbf{W}')$
6. $\forall \mathbf{w}(\mathbf{w} \in \mathbf{W} \Rightarrow Rec(\mathbf{R} \cap (\{\mathbf{w}\} \times \mathbf{W})) \in \mathbf{W}')$

Let CONV_1 be a function giving to each modal structure $\mathcal{A}$ the least general structure $\mathcal{AG}$ defined as above; *i.e.*, the intersection of general structures.

Let $\mathcal{F}$ be the class of all frames built on modal structures and $\mathcal{G}$ the class of all general structures defined as above.

Lemma Let $\mathcal{G}$ be the whole class of general structures built on modal structures, then

$$\mathcal{G} \subseteq \{\mathcal{B} \mid \exists \mathcal{A}(\mathcal{A} \text{ is a PM-structure } \&\ \mathcal{B} \in \mathcal{F} \text{ is built on } \mathcal{A} \\ \&\{\mathcal{A}(\varphi) |\ \varphi \in \text{ FORM}(PML)\} \subseteq \mathbf{W}')\}$$

Lemma Given a PM-structure $\mathcal{A}$ and a general structure $\mathcal{AG}$ built on it,

$$\mathcal{AG}(\lambda u TRANS^{\blacksquare}(\varphi[u])) = \mathcal{A}(\varphi), \text{ for all } \varphi \in \text{ FORM(PML)}$$

Corollary For any general structure $\mathcal{AG}$ built on a PM-structure $\mathcal{A}$

$$\mathcal{A}, s \models \varphi \text{ in PML iff } \mathcal{AG}[s](\text{TRANS}^{\blacksquare}(\varphi)[u]) = T \text{ in MSL}^{\blacksquare}$$

(where $\mathcal{AG}[s](\text{TRANS}^{\blacksquare}(\varphi)[u])$ is an abbreviation of $\langle \mathcal{AG}, M^s_u \rangle(\text{TRANS}^{\blacksquare}(\varphi)[u])$).

Corollary (Lemma 1 (a)) From the above corollary, it easily follows that for any PM-structure $\mathcal{A}$

$$\text{(a) } \mathcal{A}, s \models \varphi \text{ in PML iff } \text{CONV}_1(\mathcal{A})[s](\text{TRANS}^{\blacksquare}(\varphi)[u]) = T \text{ in MSL}^{\blacksquare}$$

Corollary (Lemma 1 (b))

(b) $\mathcal{A} \models \varphi$ in PML iff $\text{CONV}_1(\mathcal{A})$ is a model of $\forall u(\text{TRANS}^{\blacksquare}(\varphi)[u])$ in $\text{MSL}^{\blacksquare}$.

Theorem 1

$$\models \varphi \text{ in PML iff } \models_{\mathcal{G}} \forall u(\text{TRANS}^{\blacksquare}(\varphi)[u]) \text{ in } \text{MSL}^{\blacksquare}$$

The MODO theory

I will define a theory, MODO, which will be proven to be syntactically and semantically equivalent to the modal logic **K**.

Definition: MODO The set Δ of axioms of this theory are:

1. Comprehension sentences for a restricted class of formulas including the translations of PML formulas; that is, all of the form

   $$\forall \exists X \forall u(\epsilon_1 u X \leftrightarrow \varphi), \text{ for each } \quad \varphi \in TRANS^{\blacksquare}(PML)[u] \cup I \cup \Sigma$$

   where $I = \{u = v | v \text{ is any individual variable such that } u \not\equiv v\}$, $\Sigma = \{Suv | v \text{ is any individual variable such that } u \not\equiv v\}$.

2. Extensionality.

Definition: MODO(S4) To the set Δ of axioms in MODO we add the second order abstract conditions for the axioms **T** and 4,

MS(T) $\forall u Y(\forall v(Suv \rightarrow \epsilon_1 v Y) \rightarrow \epsilon_1 u Y)$

MS(4) $\forall u Y(\forall v(Suv \rightarrow \epsilon_1 v Y) \rightarrow \forall v(Suv \rightarrow \forall w(Svw \rightarrow \epsilon_1 w Y)))$

and call the resulting set MODO(S4).

Theorem The class $\mathcal{G}$ of general structures built on PM-structures is axiomatized by Δ; i.e., $MOD(\Delta) = \mathcal{G}$.

Representation theorem for K $\models \varphi$ in the whole class of Kripke structures (i.e., in the class of structures for the logic **K**) iff MODO $\models \forall u(\text{TRANS}^{\blacksquare}(\varphi)[u])$ in MSL.

The powerful theory MODO The theory MODO, MODO $= \Delta$, will enable us to prove the equivalence of the first order conditions for the axioms **D,T,B**, **4** and **5** with the second order abstract conditions for the translations of these axioms.

1. $\Delta \vdash \mathbf{MS}(\mathbf{D}) \leftrightarrow \alpha_D$

$$\begin{aligned} \mathbf{MS}(\mathbf{D}) &: = \forall uY(\forall v(Suv \to Yv) \to \exists v(Suv \wedge Yv)) \\ \alpha_D &: = \forall u \exists v Suv \end{aligned}$$

   The proof is easy

2. $\Delta \vdash \mathbf{MS}(\mathbf{T}) \leftrightarrow \alpha_T$

$$\begin{aligned} \mathbf{MS}(\mathbf{T}) &: = \forall uY(\forall v(Suv \to Yv) \to Yu) \\ \alpha_T &: = \forall u Suu \end{aligned}$$

3. $\Delta \vdash \mathbf{MS}(\mathbf{B}) \leftrightarrow \alpha_B$

$$\begin{aligned} \mathbf{MS}(\mathbf{B}) &: = \forall uY(Yu \to \forall v(Suv \to \exists w(Svw \wedge Yw))) \\ \alpha_B &: = \forall uv(Suv \to Svu) \end{aligned}$$

4. $\Delta \vdash \mathbf{MS}(\mathbf{4}) \leftrightarrow \alpha_4$

$$\begin{aligned} \mathbf{MS}(\mathbf{4}) &: = \forall uY(\forall v(Suv \to Yv) \to \forall v(Suv \to \forall w(Svw \to Yw))) \\ \alpha_4 &: = \forall uvw(Suv \wedge Svw \to Suw) \end{aligned}$$

5. $\Delta \vdash \mathbf{MS}(\mathbf{5}) \leftrightarrow \alpha_5$

$$\begin{aligned} \mathbf{MS}(\mathbf{5}) &: = \forall uY(\exists v(Suv \wedge Yv) \to \forall v(Suv \to \exists w(Svw \wedge Yw))) \\ \alpha_5 &: = \forall uv(Suv \wedge Suw \to Svw) \end{aligned}$$

Reverse conversion

Definition From a model $\mathcal{B}$ of Δ, which we are viewing as a SOL-structure, we obtain a modal structure by the simple procedure of deleting the universe of sets, $\mathbf{W}'$. So we can define

$$\mathrm{CONV}_2 = \mathrm{CONV}_1^{-1}$$

as a function from $\mathrm{MOD}(\Delta) \subseteq \mathrm{ST}(\mathrm{SOL})$ into $\mathrm{ST}(\mathrm{PML})$.

Strong version of lemma 2 There is a recursive $\Delta \subseteq \mathrm{SENT}(\mathrm{MSL}^{\blacksquare})$ such that $\mathrm{CONV}_1(\mathrm{ST}(\mathrm{PML})) \subseteq \mathrm{MOD}(\Delta)$ and for every $\mathcal{B} \in \mathrm{MOD}(\Delta)$ in $\mathrm{ST}(\mathrm{SOL})$ there is $\mathrm{CONV}_2(\mathcal{B}) \in \mathrm{ST}(\mathrm{PML})$. For every $\varphi \in \mathrm{FORM}(\mathrm{PML})$,

(a) $\mathrm{CONV}_2(\mathcal{B}), \mathbf{s} \models \varphi$ in PML iff $\mathcal{B}[\mathbf{s}]$ is a model of $\mathrm{TRANS}^{\blacksquare}(\varphi)[u]$ in $\mathrm{MSL}^{\blacksquare}$

(b) $\mathrm{CONV}_2(\mathcal{B}) \models \varphi$ in PML iff $\mathcal{B}$ is a model of $\forall u(\mathrm{TRANS}^{\blacksquare}(\varphi)[u])$ in $\mathrm{MSL}^{\blacksquare}$

Main theorem for K There is a recursive $\Delta \subseteq \text{SENT}(\text{MSL}^{\blacksquare})$ with

$$\text{CONV}_1(\text{ST}(\text{PML})) \subseteq \text{MOD}(\Delta)$$

such that $\Pi \models \varphi$ (in the whole class of Kripke structures) iff $\text{TRANS}^{\blacksquare}(\Pi)[u] \cup \Delta \models \text{TRANS}^{\blacksquare}(\varphi)[u]$ in MSL.

Compactness of K If $\Pi \models \varphi$ (in the whole class of Kripke structures), then there is a finite $\Gamma \subseteq \Pi$ such that $\Gamma \models \varphi$, for any $\Pi \cup \{\varphi\} \subseteq \text{FORM}(PML)$.

Enumerability of K. The set of all valid formulas in Kripke structures is recursively enumerable.

Löwenheim-Skolem for K. Let $\Pi \subseteq \text{FORM}(\text{PML})$. If Π has a Kripke model, then it has a countable model with a countable domain.

Representation theorem for S4 Let $\mathcal{C}$ be the class of reflexive and transitive Kripke structures.

$$\models_{\mathcal{C}} \varphi \text{ iff } \text{MODO}(\mathbf{S4}) \models \forall u(\text{TRANS}^{\blacksquare}(\varphi)[u])$$

for every formula φ of PML.

Main theorem for S4 Let $\mathcal{C}$ be the class of reflexive and transitive Kripke structures.

$$\Pi \models_{\mathcal{C}} \varphi \text{ iff } \text{TRANS}^{\blacksquare}(\Pi)[u] \cup \text{MODO}(\mathbf{S4}) \models \text{TRANS}^{\blacksquare}(\varphi)[u]$$

for any $\Pi \cup \{\varphi\} \subseteq \text{FORM}(\text{PML})$.

Compactness of S4 Let $\mathcal{C}$ be the class of reflexive and transitive Kripke structures.

$$\text{If } \Pi \models_{\mathcal{C}} \varphi \text{ then there is a finite } \Gamma \subseteq \Pi \text{ such that } \Gamma \models_{\mathcal{C}} \varphi$$

Enumerability of S4 The set of all valid formulas in reflexive Kripke structures is recursively enumerable.

Löwenheim-Skolem for S4 Let $\Pi \subseteq \text{FORM}(\text{PML})$. If Π has a reflexive and transitive Kripke model, then it has a countable one.

Testing the calculus

Lemma 3 for the logic K Let $\varphi \in \text{FORM}(PML)$.

$$\text{If } \vdash_{\mathbf{K}} \varphi, \text{ then } \text{MODO } \vdash_{MSL} \forall u(\text{TRANS}^{\blacksquare}(\varphi)[u]).$$

Theorem 3 for the logic K For all $\Pi \cup \{\varphi\} \subseteq \text{FORM(PML)}$:

$$\text{If } \Pi \vdash_{\mathbf{K}} \varphi, \text{ then } \text{TRANS}^{\blacksquare}(\Pi)[u] \cup \text{MODO} \vdash_{MSL} \text{TRANS}^{\blacksquare}(\varphi)[u]$$

Lemma 3 for the logic S4 For all $\varphi \in \text{FORM(PML)}$

$$\text{If } \vdash_{\mathbf{S4}} \varphi, \text{ then } \text{MODO}(S4) \vdash_{MSL} \forall u(\text{TRANS}^{\blacksquare}(\varphi)[u])$$

Theorem 3 for the logic S4 For all $\Pi \cup \{\varphi\} \subseteq \text{FORM}(PML)$,

$$\text{If } \Pi \vdash_{\mathbf{S4}} \varphi, \text{ then } \text{TRANS}^{\blacksquare}(\Pi)[u] \cup \text{MODO}(\mathbf{S4}) \vdash_{MSL} \text{TRANS}^{\blacksquare}(\varphi)[u]$$

Definition of the canonical general structure Given a consistent normal logic **B** and its canonical model

$$\mathcal{A}_{\mathbf{B}} = \langle \mathbf{W_B}, \mathbf{R_B}, \langle P^{\mathcal{A}_{\mathbf{B}}} \rangle_{P \in \text{ATOM.PROP}} \rangle$$

we define the canonical general structure, $\mathcal{A}_{\mathbf{B}}\mathcal{G}$, as $\text{CONV}_1(\mathcal{A}_{\mathbf{B}})$.

Proposition For every $\varphi \in \text{FORM}(PML)$ and $\mathbf{t} \in \mathbf{W_B}$:

$$\mathcal{A}_{\mathbf{B}}\mathcal{G}[\mathbf{t}] \models TRANS^{\blacksquare}(\varphi)[u] iff \varphi \in \mathbf{t}$$

Proposition For the consistent normal logic **K**, $\mathcal{A}_{\mathbf{K}}\mathcal{G}$ is a model of MODO.

Lemma 5 For all $\varphi \in \text{FORM}(PML)$,

$$\text{if } \mathcal{A}_{\mathbf{K}}\mathcal{G} \models \forall u(\text{TRANS}^{\blacksquare}(\varphi)[u]) \text{ then } \vdash_{\mathbf{K}} \varphi$$

Lemma 4 for the logic K For all $\varphi \in$FORM(PML),

$$\text{MODO} \vdash_{MSL} \forall u(\text{TRANS}^{\blacksquare}(\varphi)[u]) \text{ iff } \vdash_{\mathbf{K}} \varphi.$$

Theorem 4 for the logic K For all $\Pi \cup \{\varphi\} \subseteq \text{FORM}(PML)$,

$$\text{TRANS}^{\blacksquare}(\Pi)[u] \cup \text{MODO} \vdash_{MSL} \text{TRANS}^{\blacksquare}(\varphi)[u] \text{ iff } \Pi \vdash_{\mathbf{K}} \varphi$$

Completeness and soundness of the calculus K

For all $\Pi \cup \{\varphi\} \subseteq$ FORM(PML),

$$\Pi \vdash_K \varphi \textbf{ iff } \Pi \models \varphi$$

Proposition For the consistent normal logic S4, $\mathcal{A}_{S4}\mathcal{G}$ is a model of MODO(S4).

Lemma 5 For all $\varphi \in \text{FORM}(PML)$:

$$\text{If } \mathcal{A}_{\mathbf{S4}}\mathcal{G} \models \forall u(\text{TRANS}^{\blacksquare}(\varphi)[u]) \text{ then } \vdash_{S4} \varphi.$$

Lemma 4 for the logic S4 For all $\varphi \in \text{FORM}(PML)$:

$$\text{MODO}(S4) \vdash_{MSL} \forall u(\text{TRANS}^{\blacksquare}(\varphi)[u]) \text{ iff } \vdash_{S4} \varphi$$

Theorem 4 for the logic S4 For all $\Pi \cup \{\varphi\} \subseteq \text{FORM(PML)}$:

$$\text{TRANS}^{\blacksquare}(\Pi)[u] \cup \text{MODO}(\mathbf{S4}) \vdash_{MSL} \text{TRANS}^{\blacksquare}(\varphi)[u] \text{ iff } \Pi \vdash_{\mathbf{S4}} \varphi$$

Completeness and soundness of the calculus S4 For all $\Pi \cup \{\varphi\} \subseteq$ FORM(PML),

$$\Pi \vdash_{\mathbf{S4}} \varphi \text{ iff } \Pi \models_{\mathcal{C}} \varphi$$

in the class $\mathcal{C}$ of reflexive and transitive Kripke structures.

7.5 Dynamic Logic

A formal language for PDL

Alphabet The alphabet of PDL contains:

1. Two disjoint sets ATOM.PROP and ATOM.PROG, of atomic formulas and programs with $\perp \in$ATOM.PROP. We will use the letter P for elements of ATOM.PROP other than $\perp$, and Q for those in ATOM.PROG. When needed, we will add subscripts: P_0, P_1, $P_2, ...$ and the same for atomic programs.
2. The connectives: $\neg, \wedge, \vee, \rightarrow$ and $\leftrightarrow$.
3. Program operators: $\cup$, ; and *.
4. Test operator: ?
5. Modal operators: $\langle\ \rangle$ and $[\ ]$.
6. Parentheses:) and (.

Definition of formulas and programs Formulas and programs of PDL are inductively defined from the symbols in the alphabet.

FORM(PDL) and PROG(PDL) (respectively, formulas of PDL and programs of PDL) are the smallest sets satisfying:

- (F1) and (F2) as in definition 2 for modal logic.
- (F3) If $\varphi \in$ FORM(PDL) and $a \in$ PROG(PDL), then $\langle a \rangle \varphi$ and $[a]\varphi$ are in FORM(PDL).

- (G1) ATOM.PROG $\subseteq$ PROG(PDL).
- (G2) If $a \in$ PROG(PDL) and $b \in$ PROG(PDL) then $(a;b)$, $(a \cup b)$ and a^* are all in PROG(PDL).
- (G3) Whenever $\varphi \in$FORM(PDL) then $(\varphi?) \in$ PROG(PDL).

Semantics

In PDL, as in ML, the notion of truth is not an absolute one, but relative to certain states. PD structures are composed from a domain of states (or worlds), certain unary relations on states and certain binary relations on states.

Definition of Kripke structures

$\mathcal{A} = \langle \mathbf{W}, \langle P^{\mathcal{A}} \rangle_{P \in \text{ATOM.PROP}}, \langle Q^{\mathcal{A}} \rangle_{Q \in \text{ATOM.PROG}} \rangle$ is a PDL structure iff

1. $\mathbf{W} \neq \emptyset$ is a non-empty set whose elements are called states.
2. For each $P \in$ ATOM.PROP, $P^{\mathcal{A}}$ is a set of states; that is, $P^{\mathcal{A}} \subseteq \mathbf{W}$.
3. For every $Q \in$ ATOM.PROG, $Q^{\mathcal{A}}$ is a binary relation on states; that is, $Q^{\mathcal{A}} \subseteq \mathbf{W} \times \mathbf{W}$.
   (Each program gets as interpretation a set of pairs of states: initial and final states.)

Definition of interpretation When a PD-structure $\mathcal{A}$ is given, each formula $\varphi \in$ FORM(PDL) gets a value $\mathcal{A}(\varphi)$ and every program $\mathbf{a}$ of PROG(PDL) also gets a value $\mathcal{A}(a)$.

- (G1) $\mathcal{A}(Q) = Q^{\mathcal{A}}$
- (G2) $\mathcal{A}(a \cup b) = \mathcal{A}(a) \cup \mathcal{A}(b)$
  - $\mathcal{A}(a;b) = \mathcal{A}(a) \circ \mathcal{A}(b)$
  - $\mathcal{A}(a*) = (\mathcal{A}(a))*$
    (where $(\mathcal{A}(a))* = \{\langle \mathbf{s}, \mathbf{t} \rangle \in \mathbf{W}^2 | \exists k \exists \mathbf{s}_0, ..., \mathbf{s}_k (\mathbf{s}_0 = \mathbf{s} \& \mathbf{s}_k = \mathbf{t} \& \forall i (i \in \{1, ..., k\} \Rightarrow \langle \mathbf{s}_{i-1}, \mathbf{s}_i \rangle \in \mathcal{A}(a)))\}$.
- (G3) $\mathcal{A}(\varphi?) = \{\langle \mathbf{s}, \mathbf{s} \rangle | \ \mathbf{s} \in \mathcal{A}(\varphi)\}$

Definability in dynamic structures Given a structure $\mathcal{A}$ of signature σ, we say that the set $\mathbf{R} \subseteq \mathbf{W}$ is *dynamic definable* in $\mathcal{A}$ just if $\mathbf{R} = \mathcal{A}(\varphi)$ for a certain $\varphi \in$ FORM(PDL). And we say that the relation $\mathbf{R} \subseteq \mathbf{W} \times \mathbf{W}$ is *dynamic definable* in $\mathcal{A}$ when $\mathbf{R} = \mathcal{A}(a)$ for a certain $a \in$ PROG(PDL).

The logic PDL

PDL shall be described as a normal modal logic in the language of PDL; that is, a logic containing K and closed under the rules of the calculus $\{\mathbf{K}, \mathbf{Df}\Diamond\}_{(\mathbf{N})}$. In particular, PDL is the smallest normal logic containing the schema axioms listed below.

Axioms

A1 $\langle a;b\rangle\varphi \leftrightarrow \langle a\rangle\langle b\rangle\varphi$

A2 $\langle a \cup b\rangle\varphi \leftrightarrow (\langle a\rangle\varphi \vee \langle b\rangle\varphi)$

A3 $\langle a*\rangle\varphi \leftrightarrow (\varphi \vee \langle a\rangle\langle a*\rangle\varphi)$

A4 $\langle\varphi?\rangle\psi \leftrightarrow \varphi \wedge \psi$

A5 $[a]\varphi \leftrightarrow \neg\langle a\rangle\neg\varphi$

A6 $[a*](\varphi \rightarrow [a]\varphi) \rightarrow (\varphi \rightarrow [a*]\varphi)$

Deductive calculus PDL has the axioms for PC, the schema axiom **K**,

$$(\mathbf{K})[a](\varphi \rightarrow \psi) \rightarrow ([a]\varphi \rightarrow [a]\psi)$$

the axioms listed above and it is closed under modus ponens (MP) and necessitation in this general form:

$$(\mathbf{N})\ \text{If } \vdash \varphi, \text{ then } \vdash [a]\varphi$$

Known metaproperties of PDL A very well known result about PDL is that this calculus is sound and complete in the weak sense. That means:

$$\models \varphi \text{ iff } \vdash \varphi$$

for any formula φ .

Non-compactness of PDL PDL is not compact.

7.6 Propositional Dynamic Logic as Many-Sorted Logic

What I want to do now is to present a many-sorted theory, SOLO^2, which is equivalent to PDL, both syntactically and semantically. As we did before in the cases dealing with SOL, or when we translated modal logic into MSL, we are going to define a syntactical translation from PDL into MSL and a semantical conversion of structures. Furthermore, I will define a theory whose axioms are: comprehension sentences for translations of PDL formulas and program; and abstract conditions for certain axioms in PDL. As you will see, the treatment I will apply to PDL is similar to the one used for PML.

Formal many-sorted language MSL^{PDL}

The signature Σ^{PDL}. Let SORT $= \{0, 1, \langle 0, 1\rangle, \langle 0, 1, 1\rangle\}$.
The signature $\Sigma^{PDL} = \langle$SORT,FUNC$\rangle$ is such that:

1. SORT is the set defined above.
2. FUNC^{PDL} is a function which takes as arguments the elements of

$$\text{OPER.SYM}^{PDL}$$

consisting of: ATOM.PROP and ATOM.PROG of PDL, the equalities among individuals, sets and relations, E_1, E_2, E_3, the connective functions and the membership relations ϵ_1 and ϵ_2.

$$\text{FUNC}^{PDL} : \text{OPER.SYM}^{\mathbf{PDL}} \longrightarrow [\mathbf{S}_{\boldsymbol{\omega}}(\text{SORT})]$$

where: all the values are the expected ones; see the values given already by the "sister" function FUNC$^{\blacksquare}$. Of course, we add:

(a) FUNC$^{PDL}(Q) = \langle 0, 1, 1\rangle$, for $Q \in$ ATOM.PROG
(b) FUNC$^{PDL}(E_3) = \langle 0, \langle 0, 1, 1\rangle, \langle 0, 1, 1\rangle\rangle$
(c) FUNC$^{PDL}(\epsilon_2) = \langle 0, 1, 1, \langle 0, 1, 1\rangle\rangle$

Alphabet The alphabet of a many-sorted language suited for PDL must contain all operation symbols in OPER.SYM^{PDL} and a countable set of variables for each $i \in$ SORT$-\{0\}$.

Translating function

From PDL with ATOM.PROP $\cup$ ATOM.PROG as propositional and program constants, we pass to a many-sorted language of signature Σ^{PDL} having

$$\langle \langle P\rangle_{P\in\text{ATOM.PROP}}, \langle Q\rangle_{Q\in\text{ATOM.PROG}}\rangle$$

as unary and binary relation symbols.

Definition With every $\varphi \in$ FORM(PDL) and individual variable u we associate as its translation a formula TRANS$^{PDL}(\varphi)[u]$ in the many-sorted language. With every program $a \in$ PROG(PDL) and all variables u, v, we associate a formula TRANS$^{PDL}(a)[u, v]$:

- TRANS^{PDL}coincides with TRANS$^{\blacksquare}$ of the previous definition for the formulas obtained by the rules (F1) and (F2).
- (F3) TRANS$^{PDL}(\langle a\rangle\varphi)[u] :=$ $\exists v$(TRANS$^{PDL}(a)[u, v]\wedge$TRANS$^{PDL}(\varphi)[v]$) where v must be a new variable (the first one in a given ordering) and TRANS$^{PDL}(a)[u, v]$ is defined below. Similarly for $[a]\varphi$.

- (G1) $\text{TRANS}^{PDL}(Q)[u,v] := Quv$ for each $Q \in$ATOM.PROG
- (G2)

$$\text{TRANS}^{PDL}(a;b)[u,v] :=$$
$$\exists w(\text{TRANS}^{PDL}(a)[u,w] \wedge \text{TRANS}^{PDL}(b)[w,v])$$

  - $\text{TRANS}^{PDL}(a \cup b)[u,v] := TRANS^{PDL}(a)[u,v] \vee TRANS^{PDL}(b)[u,v]$
  - $\text{TRANS}^{PDL}(a*)[u,v] := \forall X^2(\text{TRANS}^{PDL}(a) \subseteq X^2 \wedge Refl\ X^2 \wedge Trans\ X^2 \rightarrow \epsilon_2 uv X^2)$
    which is an abbreviation for

$$\begin{aligned} \forall X^2(\forall uv(\text{TRANS}^{PDL}(a)[u,v] &\rightarrow \epsilon_2 uv X^2) \wedge \\ \forall u\ \epsilon_2 uu X^2 \wedge \forall wuv(\epsilon_2 wu X^2 \wedge \epsilon_2 uv X^2 &\rightarrow \epsilon_2 wv X^2) \rightarrow \epsilon_2 uv X^2) \end{aligned}$$

- (G3) $\text{TRANS}^{PDL}(\varphi?)[u,v] := \text{TRANS}^{PDL}(\varphi)[u] \wedge u = v$

Structures and frames built on PD-structures

Frames built on PD-structures of signature σ Given a PD-structure $\mathcal{A}$, of signature σ,

$$\mathcal{A} = \langle \mathbf{W}, \langle P^{\mathcal{A}} \rangle_{P \in \text{ATOM.PROP}}, \langle Q^{\mathcal{A}} \rangle_{Q \in \text{ATOM.PROG}} \rangle$$

we say that $\mathcal{AF}$ is a *frame built on* $\mathcal{A}$ iff

$$\mathcal{AF} = \langle \mathbf{W}, \mathbf{W}', \mathbf{W}'', \langle P^{\mathcal{A}} \rangle_{P \in \text{ATOM.PROP}}, \langle Q^{\mathcal{A}} \rangle_{Q \in \text{ATOM.PROG}} \rangle$$

where:

1. $\mathbf{W}' \subseteq \wp\mathbf{W}$ and $P^{\mathcal{A}} \in \mathbf{W}'$, for all $P \in$ATOM.PROP,
2. $\mathbf{W}'' \subseteq \wp(\mathbf{W} \times \mathbf{W})$ and $Q^{\mathcal{A}} \in \mathbf{W}''$, for all $Q \in$ATOM.PROG.

The class of frames built on structures of signature σ will be denoted by $\mathcal{F}$.

Extended dynamic frames Given a PD-structure $\mathcal{A}$ we say that $\mathcal{AE}$ is an extended dynamic frame built on $\mathcal{A}$ iff

$$\mathcal{AE} = \langle \mathbf{W}, \mathbf{W}', \mathbf{W}'', \langle P^{\mathcal{A}} \rangle_{P \in \text{PROP.CONS}}, \langle Q^{\mathcal{A}} \rangle_{Q \in \text{PROG.CONS}} \rangle$$

satisfies:

1. $\emptyset \in \mathbf{W}'$ and $\mathbf{W} \in \mathbf{W}'$
2. $\{P^{\mathcal{A}} |\ P \in \text{ATOM.PROP}\} \subseteq \mathbf{W}'$ and $\{Q^{\mathcal{A}} |\ Q \in \text{ATOM.PROG}\} \subseteq \mathbf{W}''$
3. $\forall \mathbf{RS}(\mathbf{R}, \mathbf{S} \in \mathbf{W}' \Rightarrow \mathbf{R} \cup \mathbf{S}, \mathbf{W} - \mathbf{R} \in \mathbf{W}')$

4. $\forall \mathbf{RS}(\mathbf{R}, \mathbf{S} \in \mathbf{W}'' \Rightarrow \mathbf{R} \cup \mathbf{S},\ \mathbf{R} \circ \mathbf{S},\ \mathbf{R}* \in \mathbf{W}'')$
5. $\forall \mathbf{R}(\mathbf{R} \in \mathbf{W}' \Rightarrow \mathbf{I_R} \in \mathbf{W}'')$ (where $\mathbf{I_R}$ is the identity on $\mathbf{R}$)
6. $\forall \mathbf{RS}(\mathbf{R} \in \mathbf{W}'' \& \mathbf{S} \in \mathbf{W}' \Rightarrow Dom(\mathbf{R} \cap (\mathbf{W} \times \mathbf{S})) \in \mathbf{W}')$

The class of extended dynamic frames built on structures of signature σ will be denoted by $\mathcal{E}$.

Let CONV_1 be a function giving to each $\mathcal{A} \in \text{ST(PDL)}$ as value a unique extended dynamic frame built on $\mathcal{A}$; namely, the least extended dynamic frame built on $\mathcal{A}$.

Lemma

$$\begin{array}{rcl} \mathcal{E} & \subseteq & \{\mathcal{B} \mid \exists \mathcal{A}(\mathcal{A} \text{ of signature } \sigma \ \&\ \mathcal{B} \in \mathcal{F} \text{ is built on } \mathcal{A} \\ \&\{\mathcal{A}(\varphi) & \mid & \varphi \in \text{ FORM(PDL)}\} \subseteq \mathbf{W}' \& \{\mathcal{A}(a) | a \in \text{ PROG(PDL)}\} \subseteq \mathbf{W}'')\} \end{array}$$

Lemma
Given a structure $\mathcal{A}$ of signature σ and an extended dynamic frame $\mathcal{AE}$ built on it,
$\mathcal{AE}(\lambda u TRANS^{PDL}(\varphi)[u]) = \mathcal{A}(\varphi)$, for all $\varphi \in \text{FORM(PDL)}$
$\mathcal{AE}(\lambda uv TRANS^{PDL}(a)[u, v]) = \mathcal{A}(a)$, for all $a \in \text{PROG(PDL)}$

Corollary For every extended dynamic frame $\mathcal{AE}$ built on a PD-structure $\mathcal{A}$ of signature σ

$$\mathcal{A}, s \models \varphi \text{ in PDL iff } \mathcal{AE}[s](\text{TRANS}^{PDL}(\varphi)[u]) = T \text{ in MSL}$$

Corollary: lemma 1 (b) for PDL $\mathcal{A} \models \varphi$ in PDL iff $CONV_1(\text{TRANS}^{PDL}(\varphi)[u])$ in MSL.

Corollary: theorem 1 $\models \varphi$ in PDL iff $\models_{\mathcal{E}} \forall u \text{TRANS}^{PDL}(\varphi)[u]$ in MSL.

The SOLO^2 theory

Definition We will define a theory which will be proven to be syntactically and semantically equivalent to PDL. The axioms of this theory are:

1. Comprehension sentences for translations of PDL formulas and programs; that is, all of the form

$$\exists X^2 \forall uv(\epsilon_2 uv X^2 \leftrightarrow \text{TRANS}^{PDL} a)[u, v]), \text{ for each } \quad a \in \text{PROG(PDL)}$$

$$\exists X \forall u(\epsilon_1 u X \leftrightarrow \text{TRANS}^{PDL}(\varphi)[u]), \text{ for each } \quad \varphi \in \text{FORM(PDL)}$$

2. Extensionality; for both, sets and relations.

3. The Σ^{PDL} many-sorted abstract conditions for PDL axioms A4 and A6.

$$\mathrm{MS}(\mathbf{A4}) := \forall X^2 Y u(\exists w(\epsilon_2 uw X^2 * \wedge \epsilon_1 wY) \leftrightarrow (\epsilon_1 uY \vee \exists v(\epsilon_2 uv X^2 \wedge \exists w(\epsilon_2 vw X^2 * \wedge \epsilon_1 wY))))$$

$$\mathrm{MS}(\mathbf{A6}) := \forall X^2 Y u(\forall w(\epsilon_2 uw X^2 * \rightarrow (\epsilon_1 wY \rightarrow \forall v(\epsilon_2 wv X^2 \rightarrow \epsilon_1 vY))) \rightarrow (\epsilon_1 uY \rightarrow \forall w(\epsilon_2 uw X^2 * \rightarrow \epsilon_1 wY)))$$

(where $\epsilon_2 uw X^2 *$ is short for $\forall Z^2(X^2 \subseteq Z^2 \wedge Refl\ Z^2 \wedge Trans\ Z^2 \rightarrow \epsilon_2 uw Z^2)$).

Lemma For each extended dynamic frame, $\mathcal{AE} \in \mathcal{E}$, $\mathcal{AE}$ is a model of SOLO^2.

Corollary: Weak version of lemma 2 $\mathrm{CONV}_1(\mathrm{ST}(\mathrm{PDL})) \subseteq \mathrm{MOD}(\mathrm{SOLO}^2)$.

Lemma 3 For each $\varphi \in \mathrm{FORM}(\mathrm{PDL})$:
whenever $\vdash \varphi$ in PDL then we have $\mathrm{SOLO}^2 \vdash \forall u \mathrm{TRANS}^{PDL}(\varphi)[u]$ in MSL.

Conclusion

$$\begin{array}{ccccc} \models \varphi & \overset{(1)}{\Longleftrightarrow} & \models_{\mathcal{E}} \forall u \mathrm{TRANS}^{PDL}(\varphi)[u] & \overset{(2)}{\Longleftarrow} & \mathrm{SOLO}^2 \models_{\mathcal{F}} \forall u \mathrm{TRANS}^{PDL}(\varphi)[u] \\ \Updownarrow (5) & & & & \Updownarrow (3) \\ \vdash \varphi & & \underset{(4)}{\Longrightarrow} & & \mathrm{SOLO}^2 \vdash_{MSL^{PDL}} \forall u \mathrm{TRANS}^{PDL}(\varphi)[u] \end{array}$$

The double arrow in (1) is lemma 1 (above)
The single left arrow in (2) follows from the fact that $\mathcal{E} \subseteq \mathcal{F}$.
The double arrow in (3) is the soundness and completeness of MSL.
The single right arrow in (4) is lemma 3 (above)
The double arrow in (5) is completeness and soundness of PDL.
Therefore, we have proven that

$$\models \varphi \Longleftrightarrow \mathrm{SOLO}^2 \models_{\mathcal{F}} \forall u \mathrm{TRANS}^{PDL}(\varphi)[u]$$

and that

$$\vdash \varphi \Longleftrightarrow \mathrm{SOLO}^2 \vdash_{MSL^{PDL}} \forall u \mathrm{TRANS}^{PDL}(\varphi)[u]$$

Bibliography

[1] Manzano, M [1996]. **Extensions of First Order Logic.** Cambridge: Cambridge University Press.