Institute for Language, Logic and Information

THE CONSISTENCY OF AN EXTENDED NaDSet

Paul C. Gilmore

ITLI Prepublication Series
for Mathematical Logic and Foundations ML-90-11

University of Amsterdam

The ITLI Prepublication Series

1986
86-01 The Institute of Language, Logic and Information
86-02 Peter van Emde Boas — A Semantical Model for Integration and Modularization of Rules
86-03 Johan van Benthem — Categorial Grammar and Lambda Calculus
86-04 Reinhard Muskens — A Relational Formulation of the Theory of Types
86-05 Kenneth A. Bowen, Dick de Jongh — Some Complete Logics for Branched Time, Part I Well-founded Time, Forward looking Operators
86-06 Johan van Benthem — Logical Syntax

1987
87-01 Jeroen Groenendijk, Martin Stokhof — Type shifting Rules and the Semantics of Interrogatives
87-02 Renate Bartsch — Frame Representations and Discourse Representations
87-03 Jan Willem Klop, Roel de Vrijer — Unique Normal Forms for Lambda Calculus with Surjective Pairing
87-04 Johan van Benthem — Polyadic quantifiers
87-05 Víctor Sánchez Valencia — Traditional Logicians and de Morgan's Example
87-06 Eleonore Oversteegen — Temporal Adverbials in the Two Track Theory of Time
87-07 Johan van Benthem — Categorial Grammar and Type Theory
87-08 Renate Bartsch — The Construction of Properties under Perspectives
87-09 Herman Hendriks — Type Change in Semantics: The Scope of Quantification and Coordination

1988
Logic, Semantics and Philosophy of Language:
LP-88-01 Michiel van Lambalgen — Algorithmic Information Theory
LP-88-02 Yde Venema — Expressiveness and Completeness of an Interval Tense Logic
LP-88-03 — Year Report 1987
LP-88-04 Reinhard Muskens — Going partial in Montague Grammar
LP-88-05 Johan van Benthem — Logical Constants across Varying Types
LP-88-06 Johan van Benthem — Semantic Parallels in Natural Language and Computation
LP-88-07 Renate Bartsch — Tenses, Aspects, and their Scopes in Discourse
LP-88-08 Jeroen Groenendijk, Martin Stokhof — Context and Information in Dynamic Semantics
LP-88-09 Theo M.V. Janssen — A mathematical model for the CAT framework of Eurotra
LP-88-10 Anneke Kleppe — A Blissymbolics Translation Program

Mathematical Logic and Foundations:
ML-88-01 Jaap van Oosten — Lifschitz' Realizabiility
ML-88-02 M.D.G. Swaen — The Arithmetical Fragment of Martin Löfs Type Theories with weak Σ-elimination
ML-88-03 Dick de Jongh, Frank Veltman — Provability Logics for Relative Interpretability
ML-88-04 A.S. Troelstra — On the Early History of Intuitionistic Logic
ML-88-05 A.S. Troelstra — Remarks on Intuitionism and the Philosophy of Mathematics

Computation and Complexity Theory:
CT-88-01 Ming Li, Paul M.B.Vitanyi — Two Decades of Applied Kolmogorov Complexity
CT-88-02 Michiel H.M. Smid — General Lower Bounds for the Partitioning of Range Trees
CT-88-03 Michiel H.M. Smid, Mark H. Overmars, Leen Torenvliet, Peter van Emde Boas — Maintaining Multiple Representations of Dynamic Data Structures
CT-88-04 Dick de Jongh, Lex Hendriks, Gerard R. Renardel de Lavalette — Computations in Fragments of Intuitionistic Propositional Logic
CT-88-05 Peter van Emde Boas — Machine Models and Simulations (revised version)
CT-88-06 Michiel H.M. Smid — A Data Structure for the Union-find Problem having good Single-Operation Complexity
CT-88-07 Johan van Benthem — Time, Logic and Computation
CT-88-08 Michiel H.M. Smid, Mark H. Overmars, Leen Torenvliet, Peter van Emde Boas — Multiple Representations of Dynamic Data Structures
CT-88-09 Theo M.V. Janssen — Towards a Universal Parsing Algorithm for Functional Grammar
CT-88-10 Edith Spaan, Leen Torenvliet, Peter van Emde Boas — Nondeterminism, Fairness and a Fundamental Analogy
CT-88-11 Sieger van Denneheuvel, Peter van Emde Boas — Towards implementing RL

Other prepublications:
X-88-01 Marc Jumelet — On Solovay's Completeness Theorem

1989
Logic, Semantics and Philosophy of Language:
LP-89-01 Johan van Benthem — The Fine-Structure of Categorial Semantics
LP-89-02 Jeroen Groenendijk, Martin Stokhof — Dynamic Predicate Logic, towards a compositional, non-representational semantics of discourse
LP-89-03 Yde Venema — Two-dimensional Modal Logics for Relation Algebras and Temporal Logic of Intervals
LP-89-04 Johan van Benthem — Language in Action
LP-89-05 Johan van Benthem — Modal Logic as a Theory of Information
LP-89-06 Andreja Prijatelj — Intensional Lambek Calculi: Theory and Application
LP-89-07 Heinrich Wansing — The Adequacy Problem for Sequential Propositional Logic
LP-89-08 Víctor Sánchez Valencia — Peirce's Propositional Logic: From Algebra to Graphs
LP-89-09 Zhisheng Huang — Dependency of Belief in Distributed Systems

Mathematical Logic and Foundations:
ML-89-01 Dick de Jongh, Albert Visser — Explicit Fixed Points for Interpretability Logic
ML-89-02 Roel de Vrijer — Extending the Lambda Calculus with Surjective Pairing is conservative
ML-89-03 Dick de Jongh, Franco Montagna — Rosser Orderings and Free Variables
ML-89-04 Dick de Jongh, Marc Jumelet, Franco Montagna — On the Proof of Solovay's Theorem
ML-89-05 Rineke Verbrugge — Σ-completeness and Bounded Arithmetic
ML-89-06 Michiel van Lambalgen — The Axiomatization of Randomness
ML-89-07 Dirk Roorda — Elementary Inductive Definitions in HA: from Strictly Positive towards Monotone
ML-89-08 Dirk Roorda — Investigations into Classical Linear Logic
ML-89-09 Alessandra Carbone — Provable Fixed points in $I\Delta_0+\Omega_1$

Computation and Complexity Theory:
CT-89-01 Michiel H.M. Smid — Dynamic Deferred Data Structures
CT-89-02 Peter van Emde Boas — Machine Models and Simulations
CT-89-03 Ming Li, Herman Neuféglise, Leen Torenvliet, Peter van Emde Boas — On Space Efficient Simulations
CT-89-04 Harry Buhrman, Leen Torenvliet — A Comparison of Reductions on Nondeterministic Space
CT-89-05 Pieter H. Hartel, Michiel H.M. Smid, Leen Torenvliet, Willem G. Vree — A Parallel Functional Implementation of Range Queries
CT-89-06 H.W. Lenstra, Jr. — Finding Isomorphisms between Finite Fields
CT-89-07 Ming Li, Paul M.B. Vitanyi — A Theory of Learning Simple Concepts under Simple Distributions and Average Case Complexity for the Universal Distribution (Prel. Version)
CT-89-08 Harry Buhrman, Steven Homer, Leen Torenvliet — Honest Reductions, Completeness and Nondeterminstic Complexity Classes
CT-89-09 Harry Buhrman, Edith Spaan, Leen Torenvliet — On Adaptive Resource Bounded Computations
CT-89-10 Sieger van Denneheuvel — The Rule Language RL/1
CT-89-11 Zhisheng Huang, Sieger van Denneheuvel, Peter van Emde Boas — Towards Functional Classification of Recursive Query Processing

Other Prepublications:
X-89-01 Marianne Kalsbeek — An Orey Sentence for Predicative Arithmetic
X-89-02 G. Wagemakers — New Foundations: a Survey of Quine's Set Theory
X-89-03 A.S. Troelstra — Index of the Heyting Nachlass
X-89-04 Jeroen Groenendijk, Martin Stokhof — Dynamic Montague Grammar, a first sketch
X-89-05 Maarten de Rijke — The Modal Theory of Inequality
X-89-06 Peter van Emde Boas — Een Relationele Semantiek voor Conceptueel Modelleren: Het RL-project

1990 SEE INSIDE BACK COVER

Instituut voor Taal, Logica en Informatie
Institute for Language, Logic and Information

Faculteit der Wiskunde en Informatica
(Department of Mathematics and Computer Science)
Plantage Muidergracht 24
1018TV Amsterdam

Faculteit der Wijsbegeerte
(Department of Philosophy)
Nieuwe Doelenstraat 15
1012CP Amsterdam

THE CONSISTENCY OF AN EXTENDED NaDSet

Paul C. Gilmore
Department of Mathematics and Computer Science
University of Amsterdam
Department of Computer Science
University of British Columbia

ITLI Prepublications
for Mathematical Logic and Foundations
ISSN 0924-2090

Received December 1990

THE CONSISTENCY OF AN EXTENDED NaDSet

Paul C. Gilmore
Department of Computer Science
University of British Columbia
Vancouver, B.C. V6T1W5, Canada
(e-mail: gilmore@cs.ubc.ca)

1. INTRODUCTION

NaDSet, as described in this paper, is a natural deduction based logic and set theory that is an extension of the one described in [Gilmore 71,80,86]. This introduction motivates the need for a new logic, summarizes the distinguishing features of the extended NaDSet, and describes some of its applications elaborated in [Gilmore 87a,b,88,89] and [Gilmore&Tsiknis 90a,b,c,d]. In [Gilmore89] NaDSet is compared with some similarly motivated logics.

In sections 2 and 3 the elementary syntax and the logical syntax, or proof theory, for NaDSet is described. In section 4 some definitions and results of a combinatorial nature are provided which depend largely on the elementary syntax. The same is done in section 5 for definitions and results depending upon the logical syntax. In section 6 the consistency of the logic is proved in the manner of Gentzen's second proof of the consistency of elementary number theory [Gentzen38][Szabo69]; namely, a transfinite induction argument over the ordinals less than ε_0 demonstrates that the empty sequent has no derivation in NaDSet.

1.1. Why a New Logic is Needed

Mathematics has traditionally used a process of abstraction to generalize and simplify structures: A property of objects is regarded as an object that may itself have properties. The traditional set theories are attempts to codify acceptable abstractions to ensure that undesirable conclusions are not drawn from sound premisses. But the concern of these set theories with what sets may correctly exist has given them an ad hoc character which may account for why "[they] have never been of particular interest to mathematicians. They now function mainly as a talismen to ward off evil "[Gray84]. This ad hoc character, as well as the complexity of their proof theory, make these set theories unsuitable for most uses within computer science.

The need for abstractions in computer science not available within traditional set theories has been argued many times. For example, [Scott70] describes the problems of self-application that can arise when interpreting programming languages and proposes a solution that has led to the development of denotational semantics. In Scott's foreword to [Stoy77], he concludes "For the future the problems of an adequate proof theory and of explaining non-determinism loom very large." This quote can be interpreted as a call for a new formal logic within which the mathematical constructions of denotational semantics can be developed. But other needs for such a logic can be identified.

Horn clause programming, as introduced in Prolog, provides a computational model, but not a deductive model, for its programs. In NaDSet, the definition of a predicate by Horn clauses is an abstraction term that is complete in the sense that two predicates with different but equivalent definitions can be proved identical without additional axioms. In short, a proof theory for the semantics of Horn clause programming can be provided. For this reason, NaDSet may suggest extensions to Prolog that incorporate second order concepts.

The increasing levels of abstraction required for the conceptual models used in enterprise modelling for database design, knowledge engineering and object-oriented systems, demands a logic within which such abstractions can be defined as objects and reasoned about. [Gilmore87a, 87b,88] describes applications of the earlier form of NaDSet to some of these problems, while [Gilmore&Tsiknis90b,d] describes two applications of NaDSet for the specification of programming semantics.

Despite its widespread appeal, no suitable logic has been available within which category theory can be properly formalized [Feferman77,84]. As demonstrated in [Gilmore&Tsiknis90a,90b], category theory can be formalized within NaDSet, and a derivation provided for the theorem that the set of all categories is itself a category.

Finally, no suitable logic is available within which computational conjectures such as P≠NP may be explored. NaDSet may prove to be such a logic.

A logic that can satisfy the above demands must of necessity offer an elementary resolution of the paradoxes of set theory. The resolution offered by NaDSet is described in the next section; briefly, it suggests that the underlying source of the paradoxes is an abuse of use and mention.

1.2. Features of NaDSet

Classical first order logic provides a formalization of two of the three fundamental concepts of modern logic, namely truth functions and quantification. In classical set theories the third fundamental concept, namely abstraction, is formalized by adding axioms to first order logic. In NaDSet the three concepts are formalized in the same manner, namely through rules of deduction in a natural deduction presentation of the logic. This is the first of four distinguishing features of

NaDSet which will be discussed.

1.2.1. Natural Deduction based Set Theory

Although the sequent calculus of [Gentzen35][Szabo69] is used for this paper, any natural deduction formalization of first order logic, such as those presented in [Beth55], [Prawitz65], or [Fitch 52] would do as well, for they can be simply extended to be a formalization of NaDSet. [Tsiknis90]

Natural deduction presentations of logic, but in particular the Gentsen sequent calculus, provide a transparent formalization of the traditional reductionist semantics of [Tarski36], in which the truth value of a complex formula depends upon the truth values of simpler formulas, and eventually upon the truth values of atomic sentences. Formalizing abstractions in this way has the effect of replacing an unrestricted comprehension axiom scheme by a comprehension rule of deduction. This replacement is not novel to NaDSet; for example, several of the theories described in [Schütte60] or the set theory of Fitch described in [Prawitz65] or [Fitch 52] have this feature. It is also implicit in the description of the logic in section 21 of [Church41]. This replacement is, however, not enough to ensure consistency; the theory described in [Gilmore68], for example, is inconsistent because of an improper definition of 'atomic formula'.

The interpretation of atomic formulas is critical for the reductionist semantics of Tarski. A second distinguishing feature of NaDSet is its interpretation of atomic formulas.

1.2.2. A Nominalist Interpretation of Atomic Formulas

In NaDSet, only names of sets, not sets may be members of sets. To emphasize that this interpretation is distinct from the interpretation of atomic formulas in classical set theory, ':' is used in place of '∈' to denote the membership relationship. For example, the atomic formula

(i) {u | ~u:u }:C

is true in an interpretation if the term '{u | ~u:u }' is in the set assigned to 'C', and is false otherwise. Note that the term '{u | ~u:u }' is being mentioned in the formula while 'C' is being used.

To avoid confusions of use and mention warned against in [Tarski36] and [Church56], NaDSet must be in effect a second order logic. The first order domain for the logic is the set $\mathbb{D}$ of all closed terms in which no parameter occurs, as defined in 2.1 below. For example, the term '{u | ~u:u }' is a member of $\mathbb{D}$. The second order domain for the logic is the set of all subsets of $\mathbb{D}$. Thus if 'C' is a second order constant, then an interpretation will assign it a subset of $\mathbb{D}$, so that (i) will be true or false in the interpretation.

Although NaDSet is in effect a second order logic, the elementary syntax requires only one kind of quantifier used for quantification over both the first order and second order domains. This is the third distinguishing feature of NadSet

1.2.3. One Universal Quantifier Instead of Two

In classical logic, existential quantification can be defined in terms of universal quantification and negation for both first and second order quantifiers. This opportunity for simplification is exploited in NaDSet as well; but the elementary syntax requires only one universal quantifier, not one for first order quantification and one for second order quantification. However, the second order nature of the logic is maintained in the two kinds of parameters that are required.

In [Gentzen35,38][Szabo69] a syntactic distinction is drawn between free and bound variables; substitutions of terms can thereby be greatly simplified since a free variable can never become bound. In NaDSet the practice of [Prawitz65] is followed in calling free variables parameters. Thus an occurrence of a parameter in a formula or term of NaDSet plays the role of a variable not bound by a quantifier or an abstraction term.

In an interpretation of NaDSet, first order parameters are assigned members of $\mathbb{D}$, while second order parameters are assigned subsets of $\mathbb{D}$.

1.2.4. A Generalized Abstraction

The term '{u | ~u:u }' introduced in 1.2.2 is a typical abstraction term for a set theory that admits such terms; they take the form **{v | F }**, where **v** is a variable, and **F** is a formula in which the variable may have a free occurrence. The term is understood to represent the set of **v** satisfying **F**. In NaDSet, however, **v** may be replaced by any term in which there is at least one free occurrence of a variable and there are no occurrences of parameters (clause 6 of the definition of elementary syntax in section 2). A term satisfying these conditions is the ordered pair term defined for variables **u** and **v** that are distinct from w as follows:

<u,v> for {w |(**u**:C ↓ v:C)}

Here '↓' is the single primitive logical connective of joint denial, in terms of which all other logical connectives are defined, and C is a second order constant. That this simple term has the desired properties of the ordered pair is demonstrated in [Gilmore89]. The ordered pair term is used, for example, to define the Cartesian product of two sets **A** and **B**:

[AxB] for {<u,v> | (u:**A** ∧ v:**B**)}

The rules of deduction for the introduction of abstraction terms such as these are natural generalizations of the rules of deduction for abstraction terms of the form {**v** | **F**}. These abstraction rules determine what are appropriate uses of abstraction terms in mathematical arguments, rather than determine what sets may consistently coexist. For example, the arguments Russell used to show that the empty set is a member of the Russell set and that the universal set is not, are arguments that can be shown to be correct in NaDSet, while the arguments demonstrating that the Russell set is and is not a member of itself cannot be justified in NaDSet. Thus it can be said that NaDSet provides an answer to the question

What constitutes a sound argument?

rather than to the question

What sets exist?

which is a concern of the classical set theories. This is stressed in [Gilmore89] where it is demonstrated that the general diagonal argument of Cantor is not a sound argument, although the commonly used instances of it in computer science are sound.

1.5. Acknowledgements

I am grateful to Johan van Benthem for the invitation to spend a term in the Department of Mathematics and Computer Science of the University of Amsterdam and for the material and technical help I received during the writing of this paper. Conversations with and suggestions from my colleagues Kees Doets, Dick de Jongh, and Anne Troelstra have been particularly helpful.

The financial support of the Natural Sciences and Engineering Research Council of Canada is acknowledged with gratitude.

2. ELEMENTARY SYNTAX OF NaDSet

Five different kinds of sytactical objects are used in the elementary syntax, namely variables, first and second order constants, and first and second order parameters. It is assumed that there are denumerably many objects of each kind, and that any object of one kind is distinct from any object of any other kind.

2.1. Definition of Elementary Syntax

.1. A variable is a term. The single occurrence of the variable in the term is a free occurrence in the term.

.2. Any parameter or constant is a term. No variable has a free occurrence in the term.

.3. If **r** and **s** are any terms, then **r**:**s** is a formula. A free occurrence of a variable in **r** or in **s**, is a free occurrence of the variable in the formula.

.4. If **G** and **H** are formulas then (**G↓H**) is a formula. A free occurrence of a variable in **G** or in **H** is a free occurrence in (**G↓H**).

.5. If **F** is a formula and **v** a variable, then ∀**vF** is a formula. A free occurrence of a variable other than **v** in **F**, is a free occurrence in ∀**vF**; no occurrence of **v** is free in ∀**vF**.

.6. Let **t** be any term in which there is at least one free occurrence of a variable and no occurrence of a parameter. Let **F** be any formula. Then {**t**|**F**} is an abstraction term and a term. A free occurrence of a variable in **F** which does not also have a free occurrence in **ta**, is a free occurrence in {**t**|**F**}. A variable with a free occurrence in **ta** has no free occurrence in {**t**|**F**}.

.7. A term is first order if no second order parameter occurs in it. A formula **r**:**T** is atomic if **r** is first order, and **T** is a variable or a second order parameter or constant. A term or formula in which no variable has a free occurrence is said to be closed.

Note that the first sentence of 2.1.7 applies to second order constants and to the abstraction terms defined in 2.1.6; for example, the second order constant 'B' is a first order term, as is also the abstraction term {<u,v> | u:v ∧ <v,y>:B}

2.2. Free and Bound Occurrences of Variables

It is important to understand what are free and not free, that is bound occurrences of variables in a term {**t**|**F**}. Consider the formula:

<u,v>:{<u,v> | u:v ∧ <v,w>:B}

The first occurrence of each of the variables 'u' and 'v' in this formula are free occurrences; all other occurrences of these variables are not free. The single occurrence of the variable 'w' is a free occurrence. Therefore in the formula

[∀u][∀w](<u,v>:{<u,v> | u:v ∧ <v,w>:B})

only the first occurrence of 'v' is free.

2.3. Closed Formulas

A closed formula must take one of the following forms:

a) (**G↓H**), where **G** and **H** are closed formulas.

b) ∀**vF**, where **F** is a formula in which at most the variable **v** has a free occurrence.

c) **r**:**T**, where both **r** and **T** are closed terms.

The only subforms of the latter are the following three:

i) **T** is {**t**|**F**}, where at most the variables with a free occurrence in **t** have a free occurrence in **F**.

ii) **r**:**T** is atomic; that is, **r** is first order and **T** is a second order parameter or constant.

iii) **r** is second order or **T** is a first order parameter or constant.

3. LOGICAL SYNTAX

NaDSet is presented as a Gentzen Sequent Calculus. Familiarity with the Gentzen sequent calculus as described in [Gentzen35], [Szabo69], [Kleene52], or [Prawitz65] is presumed. As noted in the introduction, this natural deduction calculus is chosen for the formalization of NaDSet because it is one of the least complicated to describe and justify. However, any natural deduction formalization of first order logic, such as those presented in [Beth55], [Prawitz65], or [Fitch52], can be simply extended to be a formalization of NaDSet.

A sequent in NaDSet takes the form

$$\Gamma \to \Theta,$$

where Γ and Θ are finite, possibly empty, sequences of closed formulas. The formulas Γ form the <u>antecedent</u> of the sequent, and the formulas of Θ the <u>succedent</u>. A sequent can be interpreted as asserting that one of the formulas of its antecedent is false, or one of the formulas of its succedent is true.

3.1. Definition of Logical Syntax

In the following, **F**, **G**, and **H** are closed formulas, unless otherwise stated.

3.1.1. Axioms

$$\mathbf{G} \to \mathbf{G}$$

where **G** is atomic

3.1.2. Logical Rules

Propositional

$$\frac{\Gamma, \mathbf{G} \to \Theta \qquad \Gamma, \mathbf{H} \to \Theta}{\Gamma \to (\mathbf{G}\downarrow\mathbf{H}), \Theta} \qquad\qquad \frac{\Gamma \to \mathbf{G}, \Theta}{\Gamma, (\mathbf{G}\downarrow\mathbf{H}) \to \Theta} \qquad \frac{\Gamma \to \mathbf{H}, \Theta}{\Gamma, (\mathbf{G}\downarrow\mathbf{H}) \to \Theta}$$

Quantification

$$\frac{\Gamma \rightarrow [\mathbf{p}/\mathbf{u}]\mathbf{F}, \Theta}{\Gamma \rightarrow \forall \mathbf{u}\mathbf{F}, \Theta} \qquad \frac{\Gamma, [\mathbf{t}/\mathbf{u}]\mathbf{F} \rightarrow \Theta}{\Gamma, \forall \mathbf{u}\mathbf{F} \rightarrow \Theta}$$

- **F** is any formula in which at most the variable **u** has a free occurrence.
- **p** is a parameter that does not occur in **F**, or in any formula of Γ or Θ of the first rule.
- **t** is any closed term.

Abstraction

$$\frac{\Gamma \rightarrow [\underline{\mathbf{r}}/\underline{\mathbf{u}}]\mathbf{F}, \Theta}{\Gamma \rightarrow [\underline{\mathbf{r}}/\underline{\mathbf{u}}]\mathbf{t}{:}\{\mathbf{t}|\mathbf{F}\}, \Theta} \qquad \frac{\Gamma, [\underline{\mathbf{r}}/\underline{\mathbf{u}}]\mathbf{F} \rightarrow \Theta}{\Gamma, [\underline{\mathbf{r}}/\underline{\mathbf{u}}]\mathbf{t}{:}\{\mathbf{t}|\mathbf{F}\} \rightarrow \Theta}$$

- $\underline{\mathbf{u}}$ is a sequence of the distinct variables with free occurrences in the term **ta**.
- **F** is a formula in which no variable, other than one of $\underline{\mathbf{u}}$, has a free occurrence.
- $\underline{\mathbf{r}}$ is a sequence of closed terms, one for each variable in $\underline{\mathbf{u}}$.
- $[\underline{\mathbf{r}}/\underline{\mathbf{u}}]$ is a simultaneous substitution operator that replaces each occurrence of the variables $\underline{\mathbf{u}}$, respectively, with the corresponding terms $\underline{\mathbf{r}}$.

3.1.3. Structural Rules

Thinning

$$\frac{\Gamma \rightarrow \Theta}{\Gamma \rightarrow \Theta, \mathbf{F}} \qquad \frac{\Gamma \rightarrow \Theta}{\mathbf{F}, \Gamma \rightarrow \Theta}$$

Contraction

$$\frac{\Gamma \rightarrow \Theta, \mathbf{F}, \mathbf{F}}{\Gamma \rightarrow \Theta, \mathbf{F}} \qquad \frac{\mathbf{F}, \mathbf{F}, \Gamma \rightarrow \Theta}{\mathbf{F}, \Gamma \rightarrow \Theta}$$

Interchange

$$\frac{\Gamma \rightarrow \Theta, \mathbf{F}, \mathbf{G}, \Delta}{\Gamma \rightarrow \Theta, \mathbf{G}, \mathbf{F}, \Delta} \qquad \frac{\Delta, \mathbf{F}, \mathbf{G}, \Gamma \rightarrow \Theta}{\Delta, \mathbf{G}, \mathbf{F}, \Gamma \rightarrow \Theta}$$

3.1.4. Cut Rule

$$\frac{\Gamma \rightarrow \Theta, \mathbf{G} \qquad \mathbf{G}, \Gamma \rightarrow \Theta}{\Gamma \rightarrow \Theta}$$

End of definition

The rules are defined as schemas. In an application of a rule in a derivation, particular formulas are used.

From the restrictions placed on the formulas appearing in the axioms and rules, it follows that only sequents of closed formulas are derivable in NaDSet.

The propositional, quantification and abstraction rules will be denoted respectively by:

$\rightarrow\downarrow$, $\downarrow\rightarrow$, $\rightarrow\forall$, $\forall\rightarrow$, $\rightarrow\{\}$ and $\{\}\rightarrow$.

When it is necessary to distinguish between the two $\downarrow\rightarrow$ rules, they will be referred to as the L$\downarrow\rightarrow$ and the R$\downarrow\rightarrow$ rules. The structural and cut rules will be referred to by name.

Although all the usual logical connectives $\sim$, $\wedge$, $\vee$, $\supset$ and $\equiv$ and the existential quantifier $\exists$ can be defined using $\downarrow$ and $\forall$, and corresponding rules of deduction derived, they will not be used in this paper.

Note that, unlike [Gentzen35,38] and [Szabo69], the cut rule is not included among the structural rules.

3.2. Eigenparameters and Eigenterms

The parameter **p** used in an application of $\rightarrow\forall$ is called the eigenparameter (briefly e-par) of the application, and the term **t** used in an application of $\forall\rightarrow$ is called the eigenterm (briefly e-term) of the application.

It is easy to confirm that the e-par **p** of an application of the $\rightarrow\forall$ rule in a derivation can be changed to any other parameter of the same order that does not occur in the derivation. It is only necessary to replace all occurrences of **p** in the derivation by the new parameter.

3.3. Failure of Parameter by Term Replacement

Gentzen's proof of the consistency of arithmetic made use of a simple property of the axioms of his theory: Any parameter occurring in a logical axiom $\mathbf{F} \rightarrow \mathbf{F}$ can be replaced in all of its occurrences by a term, and the resulting sequent will still be an axiom. As a consequence, the replacement of all occurrences of a parameter in a derivation by a term results in a derivation.

Because of the definition of atomic, and the restriction of axioms $\mathbf{G} \rightarrow \mathbf{G}$ to those for which $\mathbf{G}$ is atomic, it is not possible in general to transform a derivation to a derivation by replacing parameters by terms. This failure of parameter by term replacement is the main source of complications in the proof of consistency of NaDSet given below.

4. COMBINATORIAL PRELIMINARIES

The definitions and results presented in this section do not require more than a superficial knowledge of section 3, although they do require a better knowledge of section 2. They all have a combinatorial, rather than a logical character.

4.1. Global Substitutions

Each application of $\rightarrow\forall$ has an e-par and each application of $\forall\rightarrow$ has an e-term. During the reduction process to be described for a derivation of the empty sequent, a one-to-one mapping of applications of $\rightarrow\forall$ onto applications of $\forall\rightarrow$ is constructed which leads to a mapping of e-pars onto e-terms. In Gentzen's proof of consistency, each e-par is replaced by its corresponding e-term. But in NaDSet this is not possible, so a record must be kept of the substitutions that would be made if they could be made. The record is kept as a global substitution consisting of zero or more components of the form $[\mathbf{t}_i/\mathbf{p}_i]$, where $\mathbf{t}_i$ is a closed term and $\mathbf{p}_i$ is a parameter, satisfying the condition:

- A parameter $\mathbf{p}_i$ occurs in a term $\mathbf{t}_j$ only if $i < j$.

The order of the components is therefore significant; it is determined by the order in which the parameters $\mathbf{p}_i$ are encountered in the reducation process as e-pars for applications of $\rightarrow\forall$, with $\mathbf{t}_i$ being the e-term of the corresponding application of $\forall\rightarrow$.

Let a global substitution σ have components $[\mathbf{t}_1/\mathbf{p}_1]$, $[\mathbf{t}_2/\mathbf{p}_2]$, ... , $[\mathbf{t}_k/\mathbf{p}_k]$. Each component $[\mathbf{t}_i/\mathbf{p}_i]$ of σ has the effect of replacing every occurrence of $\mathbf{p}_i$ in a term or formula to which it is applied by $\mathbf{t}_i$. The result $\sigma(\mathbf{F})$ of applying σ to a term or formula $\mathbf{F}$ is the result of successively applying the components in the reverse order: first $[\mathbf{t}_k/\mathbf{p}_k]$, then $[\mathbf{t}_{k-1}/\mathbf{p}_{k-1}]$, ... , and then finally $[\mathbf{t}_1/\mathbf{p}_1]$.

4.2. Labelled Graphs

Cuts are eliminated from a derivation of the empty sequent in much the same manner as Gentzen. Cuts are accumulated as edges of a labelled graph as the reduction of the derivation proceeds, and then eliminated in pairs when the middle sequent of the two cuts is an axiom.

4.2.1. Terminology

A graph $\mathbb{G}$ consists of a nonempty set of vertices V and a set E of edges consisting of ordered pairs <v1,v2> of distinct vertices v1and v2, said to be the vertices of the edge; v1 is the tail of the edge, while v2 is the head. An undirected path of $\mathbb{G}$ is a sequence $e_1, \ldots, e_n$ of distinct edges such that successive edges e_i and e_{i+1} have a common vertex. $\mathbb{G}$ is said to be connected acyclic if there is exactly one undirected path connecting each pair of distinct vertices of $\mathbb{G}$. A vertex of $\mathbb{G}$ is said to be a leaf if it is a vertex of at most one edge.

Note that if $\mathbb{G}$ is a connected acyclic graph then not both <v1,v2> and <v2,v1> are edges of $\mathbb{G}$ for any vertices v1 and v2. The following lemmas state two easily established properties of such graphs.

4.2.1.1. Lemma: Every connected acyclic graph has at least one leaf vertex.

4.2.1.2. Lemma: Let <v1,v2> be an edge of a connected acyclic graph $\mathbb{G}$. If the edge is removed, then $\mathbb{G}$ is split into two connected acyclic graphs, one in which v1 is a vertex and one in which v2 is a vertex.

For the remainder of the paper, a connected acyclic graph will be referred to simply as a graph.

Given a global substitution σ, a labelled graph for σ is a graph for which each edge <v1,v2> is assigned a single label consisting of a pair **<F1, F2>** of closed formulas for which

σ(**F1**) is σ(**F2**).

The label assigned to an edge is said to be the label of the edge. The formula **F1** is called the tail formula for the edge, and the formula **F2**, the head formula.

Let **F1** and **F2** be respectively the tail and head formulas of an edge of a labelled graph. Both **F1** and **F2** together must be of one of the three forms (a), (b), or (c) described in 2.3. In the first two cases the edge is said to be respectively a ↓ edge or a ∀ edge. In the third case the edge is called a { } edge if both **F1** and **F2** have the form (ci), an atomic edge if at least one of them has the form (cii) and the other does not have the form (ciii), and a thinned edge if either has the form (ciii). An atomic edge for which **F1** (respectively **F2**) is atomic is said to have an atomic tail (respectively an atomic head). A thinned edge for which **F1** (respectively **F2**) has the form (ciii) is said to have a thinned tail (respectively a thinned head).

4.2.2. Reducible Vertices

A vertex v of a labelled graph is said to be reducible if it satisfies one of the following four conditions:

R0: v is the only vertex of the graph;

R1: v is the head or tail of a ↓ or { } edge or the thinned head or tail of a thinned edge;

R2: there are one or more ∀ edges of which v is head, every other edge of which v is head is atomic with atomic head, and every edge of which v is tail is atomic with atomic tail; or

R3: each edge of which v is the head is atomic with an atomic head, and each edge of which v is the tail is atomic with an atomic tail.

Lemma: Every labelled graph has a reducible vertex.

Proof: By induction on the number of edges of a labelled graph $\mathbb{G}$. If $\mathbb{G}$ has no edges then its single vertex is reducible by R0.

Assume the lemma proved for all graphs $\mathbb{G}$ with n or fewer edges, where $n \geq 0$. Consider a graph $\mathbb{G}'$ with n+1 edges. By the lemma of 4.2.1, $\mathbb{G}'$ has at least one leaf vertex v'. Let e be the sole edge of which v' is a vertex, and let v be the vertex of $\mathbb{G}$ which is the other vertex of e.

If e is a ↓ or { } edge then both v and v' satisfy R1 in $\mathbb{G}'$, and one of them does if e is a thinned edge. Assume, therefore, that e is a ∀ edge or an atomic edge with an atomic head or tail.

If e is a ∀ edge with head v', then v' satisfies R2 in $\mathbb{G}'$. If e is an atomic edge with v' its atomic head or tail, then v' satisfies R3 in $\mathbb{G}'$. Assume, therefore, that either

a) e is a ∀ edge with head v, or

b) e is an atomic edge with v its atomic head or tail.

By the induction assumption $\mathbb{G}$ has a reducible vertex. If v is not a reducible vertex of $\mathbb{G}$, then some other vertex of $\mathbb{G}$ is reducible and is a reducible vertex of $\mathbb{G}'$. Assume therefore that v is a reducible vertex of $\mathbb{G}$. There are four cases to consider:

- v is the only vertex of $\mathbb{G}$. If (a) is the case, then v satisfies R2 in $\mathbb{G}'$. If (b) is the case, then v satisfies R3 in $\mathbb{G}'$.
- v satisfies R1 in $\mathbb{G}$. Then v satisfies R1 in $\mathbb{G}'$.
- v satisfies R2 in $\mathbb{G}$. Then in each of the cases (a) and (b), v satisfies R2 in $\mathbb{G}'$.
- v satisfies R3 in $\mathbb{G}$. Then if (a) is the case, v satisfies R2 in $\mathbb{G}'$, while if (b) is the case, v satisfies R3 in $\mathbb{G}'$.

End of proof

4.3. Contradiction Graphs

Associated with each vertex v of a labelled graph is a sequent $\Gamma \rightarrow \Theta$, called the sequent of the vertex, defined as follows:

- Each occurrence of a formula in Γ is an occurrence of the head formula of an edge with head v; and
- Each occurrence of a formula in Θ is an occurrence of the tail formula of an edge with tail v.

It is assumed that the order of the occurrences of formulas in Γ and Θ is fixed by an ordering of all closed formulas of NaDSet. Note that the sequent of the single vertex of the labelled graph without edges is the empty sequent.

A labelled graph is called a contradiction graph if the sequent for each of its vertices is derivable. It will be assumed that for a given contradiction graph the derivations of the sequents of its vertices are fixed. By the derivation of a vertex of a contradiction graph is meant the given derivation for the sequent of the vertex.

In summary, the vertices of a contradiction graph with global substitution σ can be thought of as derivations while the edges are ordered pairs <**G1**, **G2**> of formulas satisfying the following conditions:

1. σ(**G1**) is σ(**G2**);
2. The endsequent of the derivation of a vertex is the sequent of the vertex.

5. LOGICAL PRELIMINARIES

This section introduces some terminology as well as some lemmas needed in the proof of consistency given in section 6. A knowledge of section 3, describing the logical syntax of NaDSet, is presumed.

The consistency proof is an adaptation of Gentzen's second proof of consistency of elementary number theory [Gentzen38]. But the terminolgy used differs at times from that used in the translation of that paper offered in chapter 8 of [Szabo69]; the differences are largely all noted in section 5.1. In sections 5.2, 5.3, and 5.4 some logical lemmas are stated and proved.

5.1. Terminology and Elementary Observations

5.1.1. Endsequent and Branch of a Derivation

A derivation in a Gentzen sequent calculus takes the form of a tree with leaves that are axioms, and with a single sequent at the root of the tree called the endsequent of the derivation. Each sequent in the tree, other than an axiom at a leaf, is the conclusion of an application of a rule of deduction with the premiss or premisses of the application immediately above the conclusion in the tree. (In [Szabo69], an axiom is called a basic sequence, a rule of deduction an inference figure schemata, and an application of a rule of deduction an inference figure.) When there is no risk of confusion, an application of a rule of deduction in a derivation will be referred to simply as a rule of deduction.

A branch of a given derivation is a sequence $Sq_1, \dots, Sq_n$ of sequents Sq_i, $i \geq 1$, where Sq_{i+1} is a premiss of a rule of deduction with conclusion Sq_i. Thus the order of sequents in a branch is upwards in the tree.

5.1.2. Principal and Corresponding Formulas

In an application of a rule of deduction, specific formulas must replace the metasystem variables printed in bold in the description of the rule, and specific sequences of formulas must replace the sequences denoted by uppercase Greek letters. The specific formulas replacing the metasystem variables are called the principal formulas of the application. An application of any logical rule, or of any structural rule other than interchange, has a single principal formula in its conclusion, while an application of interchange has two principal formulas in its conclusion. The premiss of an application of contraction has two principal formulas. Each premiss of an application of a logical rule has a single principal formula. Each application of a thinning rule has no principal formula in its premiss, while each application of cut has no principal formula in its conclusion. (In [Szabo69] only an application of a logical rule has a principal formula, and it is the principal formula of the conclusion)

Each principal formula in the conclusion of an application of a rule has a corresponding principal formula in each premiss of the rule. Each formula that is not a principal formula of the conclusion of an application has a corresponding identical formula in each premiss of the rule.

5.1.3. Blocked Applications of $\forall\rightarrow$

The principal formula of the premiss of an application of the $\rightarrow\forall$ rule takes the form **[p/u]F**, where **p** is the e-par of the application. The requirement that the e-par of an application cannot occur in any formula in the conclusion is referred to briefly as the e-par restriction. The principal formula of the premiss of an application of the ∀→ rule takes the form **[t/u]F**, where **t** is the e-term of the application.

There is no e-term restriction similar to the e-par restriction. However, complications arise from interactions between applications of ∀→ and →∀. Consider the following example derivation in which the horizontal bars between premiss and conclusion have been omitted:

c:P → c:P	axiom
∀x c:x → c:P	∀→
∀x c:x → ∀x c:x	→∀
∀x c:x, (∀x c:x↓∀x c:x) →	↓→
∀x c:x, c:{u ∣ (∀x u:x↓∀x u:x) } →	{ }→
∀x c:x, ∀x c:x →	∀→
∀x c:x →	contraction

The first rule applied is ∀→; its e-term P is the e-par of the application of →∀ that follows it. Thus the ∀→ rule removes an occurrence of the parameter P in the antecedent of the premiss of the →∀ rule, permitting that application to satisfy the e-par restriction. The removal of the application of ∀→ would prevent the application of →∀ from being made; the application of ∀→ is said to be blocked by the application of →∀. The second application of ∀→ is however not blocked by any application of →∀.

The full definition of blocking follows: Consider a branch of a derivation ending in the premiss

Γ → **[p /u]F**, Θ

of an application of →∀, and consider an application of ∀→ occuring in the branch. The application of ∀→ is said to be blocked by the application of →∀ if the e-par **p** occurs in the e-term of the application of ∀→. An application of ∀→ is said to be blocked in a derivation if it is blocked by some application of →∀ in the derivation.

One elementary result concerning blocking is used in the consistency proof:

5.1.3.1. Lemma: Let Γ → Θ be a derivable sequent for which each formula of Θ is atomic, and for which each formula of Γ is atomic or is of the form ∀**vG**. Further, let at least one of the formulas ∀**vG** of the endsequent have a top identical predecessor that is the principal formula in the conclusion of an application of ∀→. Then the last application of ∀→ in the derivation is not blocked.

The endsequent of the derivation in 5.1.3 is an example of a sequent satisfying the conditions of the lemma. That the last application of ∀→ is not blocked follows from the fact that no application of →∀ follows it.

5.1.4. Predecessor, Logical Predecessor, and (Top) Identical Predecessor

Consider an occurrence of a formula in a derivation. A predecessor of the occurrence is the occurrence itself, or a formula in a premiss of an application of a rule of deduction corresponding to a predecessor in the conclusion. The principal formula of a premiss of an application of a logical rule is a logical predecessor of the principal formula of its conclusion.

A predecessor **H** of an occurrence **F** is called an identical predecessor if **H**, and every predecessor **G** of **F** of which **H** is a predecessor, is an occurrence of the same formula as **F**. A top identical predecessor is an identical predecessor that is the antecedent or succedent of an axiom, or the principal formula in the conclusion of an application of thinning or of a logical rule.

5.2. Transformations of Contradiction Graphs

For all but one of the following transformations, the global substitution for the given contradiction graph will be the global substitution of the transformed graph.

5.2.1. A Cut-Transformation

Let $\Gamma \rightarrow \Theta$ be the sequent of a vertex v of a contradiction graph $\mathbb{G}$, and let the derivation of $\Gamma \rightarrow \Theta$ have an application of cut which is not followed by any application of a logical rule. Let **G** be the principal formula of the premisses of the cut.

The cut-transformation of the contradication graph $\mathbb{G}$ proceeds as follows:

- A duplicate $\mathbb{G}'$ of $\mathbb{G}$ is formed.
- $\mathbb{G}$ and $\mathbb{G}'$ are joined by an edge with tail the vertex v of $\mathbb{G}$ and with head the duplicate of v in $\mathbb{G}'$.
- The label for the new edge is < **G**, **G** >.

The endsequents of the derivations for v and v' in the transformed graph are

$\Gamma \rightarrow \mathbf{G}, \Theta$ and $\Gamma, \mathbf{G} \rightarrow \Theta$

Since derivations of these sequents can be obtained from the derivations of the premisses of the cut, the transformed graph is a contradiction graph.

5.2.2. A Thinning-Transformation

Let <v1,v2> be an edge with label < **F1**, **F2** > of a contradiction graph $\mathbb{G}$. Let $\Gamma \rightarrow \mathbf{F1}, \Theta$ be the sequent of v1, and $\Delta, \mathbf{F2} \rightarrow \Lambda$ the sequent of v2.

- If every top identical formula of **F1** is the principal formula in the conclusion of thinning, then the thinning-transformation of $\mathbb{G}$ is $\mathbb{G}1$.
- If every top identical formula of **F2** is the principal formula in the conclusion of thinning, then the thinning-transformation of $\mathbb{G}$ is $\mathbb{G}2$.

In the first case $\mathbb{G}1$ is a contradiction graph with $\Gamma \rightarrow \Theta$ the sequent for v1, and in the second case $\mathbb{G}2$ is a contradiction graph with $\Gamma \rightarrow \Theta$ the sequent for v2. In each case a derivation of the sequent of the vertex can be obtained from the derivation of the premiss of the thinning.

5.2.3. A ↓-Transformation

Let <v1,v2> be an edge of a contradiction graph $\mathbb{G}$ with label < **(G1↓H1)**, **(G2↓H2)** >. Let $\Gamma \rightarrow$ **(G1↓H1)**, Θ be the sequent for v1 and Δ, **(G2↓H2)** $\rightarrow \Lambda$ the sequent for v2. Let **(G1↓H1)** have a top identical predecessor that is the principal formula in the conclusion of an application of →↓, and let **(G2↓H2)** have a top identical predecessor that is the principal formula in the conclusion of an application of the L↓→ rule. The derivations Derv1 and Derv2 for the sequents of the vertices v1 and v2 may be assumed to have the following forms:

<u>Derv1</u>

$$\dfrac{\begin{matrix}\cdots & & \cdots \\ \Gamma', \mathbf{G1} \rightarrow \Theta' & & \Gamma', \mathbf{H1} \rightarrow \Theta'\end{matrix}}{\cdots \quad \Gamma' \rightarrow \mathbf{(G1{\downarrow}H1)}, \Theta' \quad \cdots}$$
$$\overline{\overline{\qquad\qquad\qquad\qquad\qquad\qquad}}$$
$$\Gamma \rightarrow \Theta, \mathbf{(G1{\downarrow}H1)}$$

<u>Derv2</u>

$$\dfrac{\begin{matrix}\cdots \\ \Delta' \rightarrow \Lambda', \mathbf{G2}\end{matrix}}{\cdots \quad \Delta', \mathbf{(G2{\downarrow}H2)} \rightarrow \Lambda' \quad \cdots}$$
$$\overline{\overline{\qquad\qquad\qquad\qquad\qquad\qquad}}$$
$$\mathbf{(G2{\downarrow}H2)}, \Delta \rightarrow \Lambda$$

Here the occurrences of '. . .' represent portions of the derivation not explicitly displayed, while the double dotted lines represent zero or more applications of rules.

Two derivations Derv1' and Derv2' may be obtained from these by dropping the displayed applications of respectively →↓ and L↓→ and replacing them with applications of thinning:

<u>Derv1'</u>

$$\dfrac{\begin{matrix}\cdots \\ \Gamma', \mathbf{G1} \rightarrow \Theta'\end{matrix}}{\cdots \quad \Gamma', \mathbf{G1} \rightarrow \mathbf{(G1{\downarrow}H1)}, \Theta' \quad \cdots}$$
$$\overline{\overline{\qquad\qquad\qquad\qquad\qquad\qquad}}$$
$$\Gamma, \mathbf{G1} \rightarrow \Theta, \mathbf{(G1{\downarrow}H1)}$$

<u>Derv2'</u>

$$\dfrac{\begin{matrix}\cdots \\ \Delta' \rightarrow \Lambda', \mathbf{G2}\end{matrix}}{\cdots \quad \Delta', \mathbf{(G2{\downarrow}H2)} \rightarrow \Lambda', \mathbf{G2} \quad \cdots}$$
$$\overline{\overline{\qquad\qquad\qquad\qquad\qquad\qquad}}$$
$$\mathbf{(G2{\downarrow}H2)}, \Delta \rightarrow \Lambda, \mathbf{G2}$$

The edge of $\mathbb{G}$ under consideration can be represented as follows:

```
                     <(G1↓H1), (G2↓H2)>
                • ---------------->----------------- •
                v1                                   v2
                Derv1                                Derv2
```

The ↓-transformation of $\mathbb{G}$ is obtained as follows:

1. By lemma 4.2.1.2, two graphs $\mathbb{G}1$ and $\mathbb{G}2$ are obtained from $\mathbb{G}$ by removing the edge <v1,v2>; $\mathbb{G}1$ is the graph in which v1 is a vertex and $\mathbb{G}2$ the graph in which v2 is a vertex.
2. A duplicate $\mathbb{G}1'$ of $\mathbb{G}1$ is made with v1 named v1', and a duplicate $\mathbb{G}2'$ of $\mathbb{G}2$ is made with v2 being named v2'.
3. Three new edges <v1,v2'>, <v2',v1'>, and <v1',v2> are added with labels respectively <**(G1↓H1), (G2↓H2)**>, <**G1,G2**>, and <**(G1↓H1), (G2↓H2)**> to form a connected acyclic labelled graph from $\mathbb{G}1$, $\mathbb{G}1'$, $\mathbb{G}2'$, and $\mathbb{G}2$.

The new edges with their labels and derivations can be represented as follows:

```
   <(G1↓H1), (G2↓H2)>               <G2,G1>              <(G1↓H1), (G2↓H2)>
• ---------------->----------------- • ---------------->----------------- • ---------------->---------------- •
v1                                   v2'                                  v1'                                 v2
Derv1                                Derv2'                               Derv1'                              Derv2
```

The transformed graph is clearly a contradiction graph. A similar transformation can be defined if the R↓→ rule rather than the L↓→ rule is used in Derv2.

5.2.4. A ∀-Transformation

Let $\mathbb{G}$ be a contradiction graph with global substitution σ, and let <v1,v2> be an edge of $\mathbb{G}$ with label < ∀v**G1**,∀v**G2** >. Let Γ→∀v**G1**, Θ be the sequent of v1, and let ∀v**G1** have a top identical predecessor which is the principal formula in the conclusion of an application of →∀ with e-par **p** satisfying the e-par restriction; therefore in particular, **p** does not occur in **G1**. Similarly let Δ, ∀v**G2** → Λ be the sequent of v2, and let ∀v**G2** have a top identical formula which is the principal formula in the conclusion of an application of ∀→ that is not blocked, and that has e-term **t**. By 3.2, it may be assumed that **p** is distinct from any parameter occurring in the components of σ or in the derivation of Δ, ∀v**G2** → Λ. In particular, it may be assumed that **p** does not occur in **t** or in **G2**.

Let Derv1 and Derv2 be respectively the derivations for the sequents of the vertices v1 and v2. Derivations Derv1' and Derv2' can be obtained from these by dropping the applications of →∀ and ∀→, and replacing them by thinnings, just as was done in 5.2.3 for applications of the ↓

rules. The endsequents for Derv1' and Derv2' are then respectively
$\Gamma \rightarrow \forall v\mathbf{G1}, [\mathbf{p}/v]\mathbf{G1}, \Theta$ and $\Delta, [\mathbf{t}/v]\mathbf{G2}, \forall v\mathbf{G2} \rightarrow \Lambda$.

The edge of $\mathbb{G}$ under consideration can be represented as follows:

```
          < ∀vG1,∀vG2 >
• ---------------->---------------- •
v1                                  v2
Derv1                               Derv2
```

The $\forall$-transformation of $\mathbb{G}$ is obtained by steps (1) and (2) of 5.2.3, followed by the following replacement for step 3:

- Three new edges <v1,v2'>, <v1',v2'>, and <v1',v2> are added with labels respectively $<\forall v\mathbf{G1},\forall v\mathbf{G2}>$, $<[\mathbf{t}/v]\mathbf{G2},[\mathbf{p}/v]\mathbf{G1}>$, and $<\forall v\mathbf{G1},\forall v\mathbf{G2}>$ to form a connected acyclic labelled graph from $\mathbb{G}1$, $\mathbb{G}1'$, $\mathbb{G}2'$, and $\mathbb{G}2$.

The new edges with their labels and derivations can be represented as follows:

```
      < ∀vG1,∀vG2 >               < [t/v]G2,[p/v]G1 >            < ∀vG1,∀vG2 >
• ---------------->---------------- • ----------------<---------------- • ---------------->---------------- •
v1                                  v2'                                 v1'                                v2
Derv1                               Derv2'                              Derv1'                             Derv2
```

The global substitution for the new graph is defined to be $\sigma[\mathbf{t}/\mathbf{p}]$. Since $\mathbf{p}$ does not occur in $\mathbf{G1}$, $\mathbf{t}$, or in $\mathbf{G2}$, $\sigma[\mathbf{t}/\mathbf{p}]([\mathbf{p}/v]\mathbf{G1})$ is $\sigma([\mathbf{t}/v]\mathbf{G1})$ is $[\sigma(\mathbf{t})/v]\sigma(\mathbf{G1})$. But since $\sigma(\forall v\mathbf{G1})$ is $\sigma(\forall v\mathbf{G2})$, $\sigma(\mathbf{G1})$ is $\sigma(\mathbf{G2})$. Therefore, $[\sigma(\mathbf{t})/v]\sigma(\mathbf{G1})$ is $[\sigma(\mathbf{t})/v]\sigma(\mathbf{G2})$ is $\sigma([\mathbf{t}/v]\mathbf{G2})$ is $\sigma[\mathbf{t}/\mathbf{p}]([\mathbf{t}/v]\mathbf{G2})$, and the transformed graph is a contradiction graph.

5.2.5. A {}-Transformation

Let $\mathbb{G}$ be a contradiction graph with global substitution σ, and let <v1,v2> be an edge with label $<\mathbf{r1}:\mathbf{T1}, \mathbf{r2}:\mathbf{T2}>$, where both $\mathbf{T1}$ and $\mathbf{T2}$ are abstraction terms. Since $\sigma(\mathbf{T1})$ is $\sigma(\mathbf{T2})$, there is a term $\mathbf{t}$ and formulas $\mathbf{G1}$ and $\mathbf{G2}$, such that $\mathbf{T1}$ and $\mathbf{T2}$ are respectively $\{\mathbf{t}|\mathbf{G1}\}$ and $\{\mathbf{t}|\mathbf{G2}\}$, for by 2.1.6 no parameter may occur in $\mathbf{t}$.

Let $\Gamma \rightarrow \mathbf{r1}:\{\mathbf{t}|\mathbf{G1}\}, \Theta$ be the sequent of v1, and let $\mathbf{r1}:\{\mathbf{t}|\mathbf{G1}\}$ have a top identical predecessor that is the principal formula in the conclusion of an application of $\rightarrow\{\}$. Similarly, let $\Delta, \mathbf{r2}:\{\mathbf{t}|\mathbf{G2}\} \rightarrow \Lambda$ be the sequent of v2, and let $\mathbf{r2}:\{\mathbf{t}|\mathbf{G2}$ have a top identical predecessor that is the principal formula in the conclusion of an application of $\{\}\rightarrow$.

It follows that $\mathbf{r1}$ must be of the form $[\underline{\mathbf{r1}}/\underline{\mathbf{u}}]\mathbf{t}$, and that the principal formula in the premiss of the application of →{ } has the form [**r1**/**u**]**G1**, where **u** is a sequence of all the variables with free occurrences in **t**, and **r1** is a sequence of terms of the same length. Similarly it follows that **r2** must be of the form [**r2**/**u**]**t**, and that the principal formula in the premiss of the application of { }→ has the form [**r2**/**u**]**G2**, where **r2** is a sequence of terms of the same length as **u**.

As in 5.2.3 and 5.2.4, let Derv1 and Derv2 be the derivations for v1 and v2, and let Derv1' and Derv2' be obtined from them by replacing applications of { } rules by thinnings. The endsequents for Derv1' and Derv2' are then respectively
Γ → [**r1**/**u**]**t**:{**t**|**G1**}, [**r1**/**u**]**G1**, Θ and Δ, [**r2**/**u**]**t**:{**t**|**G2**}, [**r2**/**u**]**G2** → Λ.

The edge of 𝔾 under consideration can be represented as follows:

```
< [r1/u]t:{t|G1},[r2/u]t:{t|G2}>
• ---------------->----------------- •
v1                                  v2
Derv1                               Derv2
```

The { }-transformation of 𝔾 is obtained by steps (1) and (2) of 5.2.3, followed by the following replacement for step 3:

- Three new edges <v1,v2'>, <v1',v2'>, and <v1',v2> are added with labels respectively <[**r1**/**u**]**t**:{**t**|**G1**},[**r2**/**u**]**t**:{**t**|**G2**}>, <[**r1**/**u**]**G1**,[**r2**/**u**]**G2** >, and <[**r1**/**u**]**t**:{**t**|**G1**},[**r2**/**u**]**t**:{**t**|**G2**}> to form a connected acyclic labelled graph from 𝔾1, 𝔾1', 𝔾2', and 𝔾2.

The new edges with their labels and derivations can be represented as follows:

```
<[r1/u]t:{t|G1},[r2/u]t:{t|G2}>   <[r2/u]G2,[r1/u]G1>   <[r1/u]t:{t|G1},[r2/u]t:{t|G2}>
• ---------------->----------------- • ----------------<----------------- • ---------------->---------------- •
v1                                   v2' v1'                               v2
Derv1                                Derv2'                                Derv1'                          Derv2
```

5.2.6. An Axiom-Transformation

Let 𝔾 be a contradiction graph with global substitution σ, and let the vertex v of 𝔾 have the axiom **A** → **A** as its sequent. There are therefore distinct vertices v1 and v2 such that <v1,v> and <v,v2> are edges of 𝔾 with labels respectively < **A1**, **A** > and < **A**, **A2** > for which
σ**A1** is σ**A** is σ**A2**.

Let 𝔾1 be the subgraph of 𝔾 in which v1 is a vertex obtained by removing the edge <v1,v> and let 𝔾2 be the subgraph in which v2 is a vertex obtained by removing the edge <v,v2>.

The transformed contradiction graph is obtained from $\mathbb{G}1$ and $\mathbb{G}2$ by joining the two graphs with an edge <v1,v2> labelled with < **A1, A2** >.

The edges of $\mathbb{G}$ under consideration with their labels and derivations can be represented as follows:

```
              <A1,A>                              <A,A2 >
• ---------------->----------------- • ---------------->------------------ •
v1                                   v                                     v2
Derv1                              A → A                                   Derv2
```

These edges are replaced by the single edge:

```
              <A1,A2>
• ---------------->---------------- •
v1                                  v2
Derv1                               Derv2
```

The transformed graph is clearly a contradiction graph.

6. CONSISTENCY PROOF

A proof of the following theorem will be provided in the subsections of this section:
Theorem: The empty sequent is not derivable in NaDSet.

The proof of the theorem will proceed by contradiction. A derivation of the empty sequent defines a contradiction graph with a single vertex and no edges. It will be shown in 6.1 that given any contradiction graph, at least one of the transformations defined in 5.2 can be applied to it. In 6.2 a definition is given of the degree of an occurrence of a formula in a derivation of the sequent of a vertex of a contradiction graph. This definition is used in 6.3 to define the ordinal number of a contradiction graph. This ordinal is always less than ε_0. The proof of the theorem is completed in 6.4 where it is shown that the ordinal number of a contradiction graph can always be decreased by applying a finite number of the transformations defined in 5.2.

6.1. A Transformation Can Always be Applied

Let $\mathbb{G}$ be a contradiction graph. By the lemma of 4.2.2, $\mathbb{G}$ has a reducible vertex v. No matter which of the conditions R0, R1, R2, or R3 v satisfies, one of the transformations defined in 5.2 can be applied to $\mathbb{G}$:

6.1.1. v satisfies R0

The sequent for v is the empty sequent. The last rule of deduction applied in any derivation of the empty sequent is necessarily cut. Therefore the cut -transformation described in 5.2.1 can be applied to $\mathbb{G}$.

6.1.2. v satisfies R1

It follows that $\mathbb{G}$ has an edge <v1,v> or <v,v2> with a label one of the following:

<(**G1**↓**H1**), (**G2**↓**H2**)>, <**r1**:{**t** | **G1**}, **r2**:{**t** | **G2**}>, or <**F1**, **F2**>,

where at least one of **F1** and **F2** has the form (ciii) of 2.3.

Assume that no thinning-transformation can be applied. In this case, only the first two labels are possible. If the first of these is the label, then necessarily a ↓-transformation can be applied, while if the second is the label, then a { }-transformation can be applied.

6.1.3. v satisfies R2

It follows that $\mathbb{G}$ has an edge <v1,v> with label < ∀**vG1**,∀**vG2** >. If no thinning-transformation applies, then by lemma 4.1.4.1 a ∀-transformation can be applied.

6.1.4. v satisfies R3

The formulas in the sequent $\Gamma \rightarrow \Theta$ of v are all atomic. Assume that no thinning transformation can be applied. If $\Gamma \rightarrow \Theta$ is an axiom, then an axiom transformationcan be applied. If $\Gamma \rightarrow \Theta$ is not an axiom, then necessarily the derivation of $\Gamma \rightarrow \Theta$ has an application of cut below which no application of a logical rule is applied. Then a cut transformation can be applied.

6.2. <u>The Degree of an Occurrence of a Formula in a Derivation</u>

The usual definition of the degree of a formula is simply a count of the number of occurrences of logical connectives and quantifiers in the formula. This definition is, however, no longer useful in NaDSet because of the interaction between the { } and ∀→ rules. For example, the last application of ∀→ in the derivation of 5.1.3 removes a term that has been introduced by a { } rule. The definition provided here is that of the degree of an <u>occurrence</u> of a formula in a contradiction graph.

A <u>degree path</u> in a contradiction graph $\mathbb{G}$ is a sequence $\mathbf{F}_1, \dots, \mathbf{F}_m$, $m \geq 1$, of distinct occurrences of formulas for which, for $i < m$, $\mathbf{F}_i$ and $\mathbf{F}_{i+1}$ satisfy one of the following conditions:

1. $\mathbf{F}_{i+1}$ is a logical predecessor of $\mathbf{F}_i$;
2. $\mathbf{F}_i$ is a logical predecessor of $\mathbf{F}_{i+1}$;
3. One of $\mathbf{F}_i$ and $\mathbf{F}_{i+1}$ is a distinct immediate predecessor of the other;
4. $\mathbf{F}_i$ and $\mathbf{F}_{i+1}$ are the principal formulas in the premisses of an application of cut in the derivation of a vertex of $\mathbb{G}$; or
5. $<\mathbf{F}_i,\mathbf{F}_{i+1}>$ or $<\mathbf{F}_{i+1},\mathbf{F}_i>$ is a label of an edge of $\mathbb{G}$.
6. One of $\mathbf{F}_i$ and $\mathbf{F}_{i+1}$ is the antecedent and the other is the succedent formulas of an axiom.

The degree of a degree path $\mathbf{F}_1, \dots, \mathbf{F}_m$ is defined recursively as follows:

- The degree of the path $\mathbf{F}_1$ is 0;
- Let the degree of $\mathbf{F}_1, \dots, \mathbf{F}_i$ be d. Then the degree of $\mathbf{F}_1, \dots, \mathbf{F}_i, \mathbf{F}_{i+1}$ is d', where the value of d' depends upon which of the six conditions $\mathbf{F}_i$ and $\mathbf{F}_{i+1}$ satisfy:
  d'= d+1 in case (1), d'= d-1 in case (2) and d'= d in the remaining cases.

Since a given occurrence of a formula can be at most one element of a degree path, there are at most finitely many degree paths with a given occurrence as first element. The degree of an occurrence is the maximum of the degrees of degree paths with the occurrence as first element.

The following notation will be used for degrees: Given a degree path dp and a occurrence **F** of a contradiction graph $\mathbb{G}$, the degree of dp is deg(dp, $\mathbb{G}$) and the degree of **F** is deg(**F**, $\mathbb{G}$)

The following lemma summarizes some of the obvious properties of the degree assignment:

6.2.1. Lemma: Let $\mathbb{G}$ be a contradiction graph, and **F** and **G** occurrences of formulas in derivations of vertices of $\mathbb{G}$. Then

1. deg(**F**, $\mathbb{G}$) = max{ deg(dp, $\mathbb{G}$) | dp a degree path with first element **F**} ≥ 0.
2. Let **F** be the principal formula in the conclusion of an application of a logical rule and let **G** be a principal formula in a premiss. Then deg(**F**, $\mathbb{G}$) = deg(**G**, $\mathbb{G}$) + 1.
3. deg(**F**, $\mathbb{G}$) = deg(**G**, $\mathbb{G}$) if **F** and **G** satisfy one of the following conditions:
   .1. One is an identical predecessor of the other;
   .2. They are the principal formulas in the premisses of an application of cut;
   .3. They are the occurrences of formulas referenced by the label of an edge of $\mathbb{G}$; or
   .4. They are antecedent and succedent formulas of an axiom.

Note that with the exception of (2), these conditions are all satisfied by the degrees in the traditional sense of formulas in a derivation in first order logic; an inequality is satisfied in this case. They are all assumed in Gentzen's consistency proof for arithmetic.

6.2.2. The Effect on Degrees of the Transformations

Unlike the traditional degrees of formulas in first order logic, the degree of an occurrence of a formula in a contradiction graph can be affected by the transformations described in 5.2, for depending upon the transformation, degree paths may be added to or removed from the transformed contradiction graph. If degree paths are added, the degree of an occurrence may be increased, while if they are removed, the degree may be decreased.

6.2.2.1. A Cut-Transformation

Since a cut with cut formulas **G** and **G** is replaced with an edge with label **<G,G>**, the degree of no occurrence of a formula is affected by the transformation.

6.2.2.2. A Thinning-Transformation

The transformed graph has at least one fewer edges than the given graph, and therefore the degree of an occurrence of a formula cannot be increased, but it may be decreased.

6.2.2.3. A Logical Transformation

By a logical transformation is meant a ↓-, ∀-, or { }-transformation. When such a transformation can be applied it transforms a contradiction graph $\mathbb{G}$ into the contradiction grah $\mathbb{G}^*$ described in 5.2.3, 5.2.4, and 5.2.5. Since the effect on degrees is similar for the other transformations, it is sufficient to consider the case of a ↓-transformation only.

A ↓-transformation replaces a single edge of $\mathbb{G}$ with three edges in $\mathbb{G}^*$ as follows:

An edge

```
                      <(G1↓H1), (G2↓H2)>
                    • ---------------->---------------- •
                    v1                                  v2
                    Derv1                               Derv2
```

is replaced by the edges

```
   <(G1↓H1), (G2↓H2)>                 <G2,G1>                  <(G1↓H1), (G2↓H2)>
• ---------------->---------------- • ---------------->---------------- • ---------------->---------------- •
v1                                  v2'                                 v1'                               v2
Derv1                               Derv2'                              Derv1'                            Derv2
```

The only way that the subgraph $\mathbb{G}1'$ of $\mathbb{G}^*$ differs from the subgraph $\mathbb{G}1$ of $\mathbb{G}$ and $\mathbb{G}^*$ is that v1 has the derivation Derv1, while v1' has the derivation Derv1'. The relationship between the derivations Derv1 and Derv1' are described in 5.2.3. With the exception of the possible additional occurrences of **G1** in Derv1', there is a one-to-one map of the occurrences of formulas in Derv1' with the occurrences in Derv1. However the possible additional occurrences of **G1** have an identical predecessor, namely the occurrence of **G1** in the premiss of thinning, that has a corresponding occurrence in Derv1, namely as the principal formula in the premiss of the application of $\rightarrow\downarrow$.

Let dp1'* be a degree path of $\mathbb{G}^*$ with elements entirely within $\mathbb{G}1'$. It follows from the observations of the previous paragraph that there is a degree path dp1 of $\mathbb{G}$ with elements entirely within $\mathbb{G}1$ for which

deg(dp1'*, $\mathbb{G}^*$) = deg(dp1, $\mathbb{G}$).

Similarly it follows from analogous observations, that if dp2'* is a degree path of $\mathbb{G}^*$ with elements entirely within $\mathbb{G}2'$, then there is a degree path dp2 of $\mathbb{G}$ with elements entirely within $\mathbb{G}2$ for which

deg(dp2'*, $\mathbb{G}^*$) = deg(dp2, $\mathbb{G}$).

These conclusions are essential to a proof of the following lemma:

Lemma: Let **F*** be an occurrence of a formula in $\mathbb{G}^*$, and let **F** be the corresponding occurrence in $\mathbb{G}$. Then deg(**F***, $\mathbb{G}^*$) $\leq$ deg(**F**, $\mathbb{G}$).

Proof: It is sufficient to prove that for any degree path dp* of $\mathbb{G}^*$ with first element **F***, there is a degree path dp of $\mathbb{G}$ with first element **F** for which deg(dp*, $\mathbb{G}^*$) $\leq$ deg(dp, $\mathbb{G}$).

If all the elements of dp* are entirely within any of the pairs of subgraphs $\mathbb{G}1$ and $\mathbb{G}2'$, $\mathbb{G}2'$ and $\mathbb{G}1'$, or $\mathbb{G}1'$ and $\mathbb{G}2$, then the result is immediate. For assume that the elements are entirely within the pair $\mathbb{G}2'$ and $\mathbb{G}1'$. Then dp* is a sequence <dp2'*, dp1'*> with subsequences dp2'* and dp1'* with elements entirely within respectively $\mathbb{G}2'$ and $\mathbb{G}1'$, and having the forms respectivley

F*, ... , **G2***

G1*, ... , **H***.

But deg(dp*, $\mathbb{G}^*$) = deg(dp2'*, $\mathbb{G}^*$) + deg(dp1'*, $\mathbb{G}^*$)

$\leq$ deg(dp2, $\mathbb{G}$) + deg(dp1, $\mathbb{G}$) = deg(dp, $\mathbb{G}$), where dp is

F, ... , G2, (G2↓H2), (G1↓H1), G1, ... , H;

since the pair **G2, (G2↓H2)** contributes -1 to the degree, the pair **(G2↓H2), (G1↓H1)** 0, and the pair **(G1↓H1), G1** +1.

A similar argument can be advanced for the cases when the elements of dp* are entirely within other pairs of subgraphs.

Assume therefore that the elements of dp* are entirely within at least three of the subgraphs. Consider first the case that they are within all four of the subgraphs, so that dp* consists of four degree paths dp1*, dp2'*, dp1'*, and dp2* with elements entirely within respectivley $\mathbb{G}1$, $\mathbb{G}2'$, $\mathbb{G}1'$, and $\mathbb{G}2$.

Since dp1* has elements entirely within the subgraph $\mathbb{G}1$ of both $\mathbb{G}^*$ and $\mathbb{G}$, **F*** may be assumed to be **F**. Similarly the last element of dp2* is in both $\mathbb{G}^*$ and $\mathbb{G}$. Therefore the four paths may be assumed to have the following forms respectively:

F, ... , (G1↓H1)

(G2↓H2)*, ... , G2*

G1*, ... , (G1↓H1)*

(G2↓H2), ... , H

Let dp2 and dp1 be the degree paths of $\mathbb{G}2$ and $\mathbb{G}1$ corresponding to the degree paths dp2'* and dp1'* of $\mathbb{G}2'$ and $\mathbb{G}1'$. Then

$$deg(dp2'^*, \mathbb{G}^*) = deg(dp2, \mathbb{G}) \leq deg((\mathbf{G2}{\downarrow}\mathbf{H2}), \mathbb{G}) - deg(\mathbf{G2}, \mathbb{G}) = 1$$

Similarly, using analagous notation,

$$deg(dp1'^*, \mathbb{G}^*) = deg(dp1, \mathbb{G}) \leq deg(\mathbf{G1}, \mathbb{G}) - deg((\mathbf{G1}{\downarrow}\mathbf{H1}), \mathbb{G}) = -1.$$

Therefore

$$\begin{aligned} deg(dp^*, \mathbb{G}^*) &= deg(dp1^*, \mathbb{G}^*) + deg(dp2'^*, \mathbb{G}^*) + deg(dp1'^*, \mathbb{G}^*) + deg(dp2^*, \mathbb{G}^*) \\ &\leq deg(dp1^*, \mathbb{G}^*) + deg(dp2^*, \mathbb{G}^*) = deg(dp1^*, \mathbb{G}) + deg(dp2^*, \mathbb{G}) \\ &\leq deg(\langle dp1^*, dp2^*\rangle, \mathbb{G}), \end{aligned}$$

where as before $\langle dp1^*, dp2^*\rangle$ is the sequence consisting of first the elements of dp1* followed by the elements of dp2*.

Consider now the case that the degree path dp* has elements entirely within three of the subgraphs. It is sufficient to consider the case that the elements are entirely within $\mathbb{G}1$, $\mathbb{G}2'$, and

$\mathbb{G}1'$, for the case that they are entirely within $\mathbb{G}2'$, $\mathbb{G}1'$, and $\mathbb{G}2$ is similar.

The degree paths dp1*, dp2'*, and dp1'* may be assumed to take the forms respectively:

F, ... , **(G1↓H1)**
(G2↓H2)*, ... , **G2***
G1*, ... , **H***

Using the same argument as in the previous case,

$$\deg(dp2'^*, \mathbb{G}^*) = \deg(dp2, \mathbb{G}) \leq \deg(\,(\mathbf{G2}{\downarrow}\mathbf{H2}), \mathbb{G}) - \deg(\mathbf{G2}, \mathbb{G}) = 1,$$

so that

$$\deg(dp^*, \mathbb{G}^*) \leq \deg(dp1^*, \mathbb{G}^*) + \deg(dp1'^*, \mathbb{G}^*) + 1$$
$$\leq \deg(dp1^*, \mathbb{G}) + \deg(dp1, \mathbb{G}) + 1,$$

where dp1 is the sequence of elements **G1**, ... , **H** in $\mathbb{G}1$ corresponding to the sequence dp1'* of elements **G1***, ... , **H*** in $\mathbb{G}1'$. Consider the sequence <dp1*, dp1> of occurrences in $\mathbb{G}1$ consisting of first those of dp1*, followed by those of dp1. The degree deg(<dp1*, dp1>, $\mathbb{G}$) of such a sequence can be defined in the same manner as the degree of a degree path, even though occurrences may be repeated as elements.

The elements of <dp1*, dp1> are

F, ... , **(G1↓H1)**, **G1**, ... , **H**

Since **G1** is a principal formula of a premiss of an application of →↓ with **(G1↓H1)** the principal formula of its conclusion, **(G1↓H1)** and **G1** can be successive elements in a degree path, and contribute +1 to the degree of the sequence. Therefore

$$\deg(\langle dp1^*, dp1\rangle, \mathbb{G}) = \deg(dp1^*, \mathbb{G}) + \deg(dp1, \mathbb{G}) + 1.$$

To complete the proof of this case it is sufficient to prove that a degree path dp can be formed from the sequence <dp1*, dp1> by removing repeated occurrences while not changing the degree of the sequent.

Let the first element of **F**, ... , **(G1↓H1)** which also appears in **G1**, ... , **H** be $\mathbf{F}_k$ so that the degree path has the form:

$\mathbf{F}, \ldots, \mathbf{F}_k, \mathbf{F}_{k+1}, \ldots, \mathbf{F}_{k+m}, (\mathbf{G1}{\downarrow}\mathbf{H1})$.

Since the only two immediate predecessors of **(G1↓H1)** are **G1** and **H1**, it may be assumed that $\mathbf{F}_{k+m}$ is **G1**, for if it were **H1**, the two sequences **F**, ... , **(G1↓H1)** and **G1**, ... , **H** would not have any occurrences in common and would form the desired degree path. Thus the sequence of occurrences

$\mathbf{F}, \ldots, \mathbf{F}_{k}, \mathbf{F}_{k+1}, \ldots, \mathbf{F}_{k+m}, \ldots, \mathbf{H}$

has the same degree as the earlier, and has one less repeated occurrence of a formula. By repeating this process of eliminating repeated occurrences, a degree path dp is obtained for which

$\deg(dp^*, \mathbb{G}^*) \leq \deg(dp, \mathbb{G})$.

End of proof of lemma

Thus the degree of an occurrence of a formula in a contradiction graph cannot be increased by a logical transformation.

6.2.2.3. An Axiom-Transformation

In the original graph, the occurrence of **A1** in the endsequent of the derivation of the vertex v1, the two occurrences of **A** in the axiom that is the derivation of v, and the occurrence of **A2** in the endsequent of the derivation of the vertex v2, all have the same degree by (4) and (5) of lemma 6.2.1. The degrees of **A1** and **A2** remain unchanged in the transformed graph, as do the degrees of all other occurrences.

6.3. The Ordinal Number of a Contradiction Graph

The method employed here for assigning an ordinal number to a contradiction graph $\mathbb{G}$ is an adaptation of the method Gentzen used to assign an ordinal number to a derivation of the empty sequent as described in [Gentzen38] and [Szabo69]. An adaptation is necessary since NaDSet has no explicit induction rule of deduction, and no basic mathematical sequents. In addition a contradiction graph has edges that are in effect cuts, but to which no order of application has been defined. However, assuming that an order of application has been specified for the edges of a contradiction graph, the graph can be regarded as a derivation of the empty sequent and therefore is assigned an ordinal by the method of Gentzen. Some definitions of [Gentzen38] and [Takeuti75] are adapted here.

The <u>degree</u> of a cut, is the degree of the cut formulas in its premiss, while the <u>degree</u> of an edge is the degree of the occurrences referenced in its label. The <u>height</u> of a sequent is the maximum of the degrees of cuts with conclusions appearing below the sequent. (In [Szabo69], the height of a sequent is called the level.)

By an <u>ordinal</u> is meant any ordinal number less than ε_0. For an ordinal μ and an integer k, $k \geq 0$, $\omega_k(\mu)$ is defined recursively: $\omega_0(\mu) = \mu$; $\omega_1(\mu) = \omega^{\mu}$; and $\omega_{k+1}(\mu) = \omega_1(\omega_k(\mu))$.

Each ordinal μ, $\mu > 0$, has a unique representation in the following normal form:

$$\mu = \omega_1(\delta_1) + \omega_1(\delta_2) + ... + \omega_1(\delta_m),$$

where $\mu > \delta_1 \geq \delta_2 \geq ... \geq \delta_m$, $m \geq 1$.

Let μ and ν be two ordinal numbers with normal forms as follows:

$$\mu = \omega_1(\delta_1) + \omega_1(\delta_2) + ... + \omega_1(\delta_m)$$

$$\nu = \omega_1(\gamma_1) + \omega_1(\gamma_2) + ... + \omega_1(\gamma_n)$$

The natural sum of the two ordinals μ and ν is

$$\mu \# \nu = \omega_1(\lambda_1) + \omega_1(\lambda_2) + ... + \omega_1(\lambda_{m+n}),$$

where $\lambda_1, \lambda_2, ... , \lambda_{m+n}$ is the sequence obtained by merging the sequences $\delta_1, \delta_2, ... , \delta_m$ and $\gamma_1, \gamma_2, ... , \gamma_n$ with duplicates maintained, and then reordering the resulting sequence so that

$$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_{m+n}.$$

Two properties of ordinals that will be used in the proof of consistency are:

1. $\mu < \mu + 1$;
2. If $\mu, \nu < \gamma$ and $k \geq 1$, then $\omega_k(\mu \# \nu) < \omega_k(\gamma)$.

6.3.1. The Ordinal of a Contradiction Graph

The ordinal number assigned to a contradiction graph $\mathbb{G}$, assuming a given order of application for the cuts represented by its edges, is defined as follows:

1. The ordinal 1 is assigned to each axiom used in a derivation of a vertex of $\mathbb{G}$;
2. The ordinal number of the conclusion of a structural rule is the ordinal of its premiss;
3. The ordinal number of the conclusion of an application of a single premiss logical rule, that is $\downarrow\rightarrow$, or either of the $\forall$ or { } rules, is $\mu + 1$, where μ is the ordinal of the premiss;
4. The ordinal number of the conclusion of an application of the two premiss logical rule $\rightarrow\downarrow$ is $\mu \# \nu$, where μ and ν are the ordinals of the premisses;
5. The ordinal number of the conclusion of an application of cut is $\omega_{h1\text{-}h2}((\mu \# \nu) + 1)$, where μ and ν are the ordinals of the premisses, and h1 and h2 are respectively the height of the premisses and of the conclusion of the cut;
6. The ordinal number of the conclusion of an application of cut represented by an edge of $\mathbb{G}$ is $\omega_{h1\text{-}h2}(\mu \# \nu)$, where μ and ν are the ordinals of the premisses, and h1 and h2 are respectively the height of the premisses and of the conclusion;
7. The ordinal number of $\mathbb{G}$, for the given order of application of the cuts represented by its edges, is the ordinal number of the empty sequence which is the endsequent of the derivation

represented by the graph.

The ordinal number of a contradiction graph is the minimum of the ordinals assigned to it for given orders of application for the cuts represented by its edges.

6.4. The Ordinal of a Contradiction Graph Can Always be Reduced

To complete the proof of consistency, it is sufficient to prove that each of the transformations defined in 5.2 decreases the ordinal of any contradiction graph. The transformations will be considered in turn.

6.4.1. Cut-transformation

The ordinal number of the conclusion of an application of cut is

$$\omega_{h1-h2}((\mu \# \nu)+1)$$

where μ and ν are the ordinals of the premiss, h1 is the height of the premisses, and h2 is the height of the conclusion. Necessarily h1 $\geq$ h2.

The cut is replaced by a labelled edge. To calculate the ordinal of the resulting contradiction graph, the minimum of the ordinals for given orders of application of the cuts corresponding to edges must be calculated. To show that ordinal of the original graph is decreased, it is sufficient to show it is decreased if the cut corresponding to the new edge is applied in the same order as the cut it replaces was applied. But with that order assumed, the ordinal of the conclusion is

$$\omega_{h1-h2}(\mu \# \nu)$$

Thus a cut-transformation always decreases the ordinal of a contradiction graph.

6.4.2. Thinning-transformation

The cut corresponding to the edge that is removed in this transformation has the ordinal

$$\omega_{h1-h2}(\mu \# \nu)$$

assigned to its conclusion. Removing the edge means that in place of this ordinal, an ordinal no greater than one of the ordinals μ and ν appears, since the possible decrease in the degrees of occurrences in the transformed graph cannot increase the ordinal of its endsequent. Therefore a thinning-transformation always decreases the ordinal of a contradiction graph.

6.4.3. Logical Transformations

For any of the three logical transformations, a single edge is replaced with three edges. The conclusion of the cut corresponding to the single edge has the ordinal $\omega_{h1-h2}(\mu \# \nu)$ assigned to its conclusion, where μ and ν are the ordinals of the premisses of the cut, h1 the height of its premisses, and h2 the height of its conclusion.

The form of the "derivation" constructed from the cuts corresponding to the edges of the given contradiction graph $\mathbb{G}$, can be represented as follows:

$$\begin{array}{ccl}
\text{(Derv1)} & \text{(Derv2)} & \text{Heights} \\
\dots & \dots & \\
\Gamma \to \Theta, (\mathbf{G1{\downarrow}H1}) & (\mathbf{G2{\downarrow}H2}), \Delta \to \Lambda & \\
=========== & =========== \text{ (thinnings)} & \\
\Sigma \to \Pi, (\mathbf{G1{\downarrow}H1}) & (\mathbf{G2{\downarrow}H2}), \Sigma \to \Pi & \text{h1} \\
\hline
\multicolumn{2}{c}{\Sigma \to \Pi} & \text{h2} \\
\multicolumn{2}{c}{\dots} & \\
\multicolumn{2}{c}{\dots \quad \Sigma' \to \Pi' \quad \dots} & \text{h (h < h1)} \\
\multicolumn{2}{c}{=================} & \\
\multicolumn{2}{c}{\to} & 0
\end{array}$$

Here the first pair of double lines represent the thinnings necessary to add formulas to Γ and Θ that are not in Δ and Λ, and to add formulas to Δ and Λ that are not in Γ and Θ. The sequent $\Sigma' \to \Pi'$ is the first sequent below the premisses of the displayed cut with a height h < h1. There must exist such a sequent since the height of the endsequent is 0. Since $\Sigma \to \Pi$ may be such a sequent, either h < h2 = h1, or h = h2 < h1.*

Assume that the two cuts corresponding to the new edges <v1,v2'> and <v1',v2> are executed in that order in the same place as the cut for the edge <v1,v2> they are replacing. Thus their premisses have the same height h1 as does the premisses for the cut they are replacing. Further assume that the remaining cut corresponding to the new edge <v2',v1'> is executed at the location of the sequent $\Sigma' \to \Pi'$.

The form of the new "derivation" constructed from the cuts corresponding to the edges of the transformed contradiction graph $\mathbb{G}$*, can be represented as follows:

$$\begin{array}{cccc}
\text{(Derv1)} & \text{(Derv2')} & \text{(Derv1')} & \text{(Derv2)} \\
\dots & \dots & \dots & \dots \\
\Gamma \to \Theta, (\mathbf{G1{\downarrow}H1}) & (\mathbf{G2{\downarrow}H2}), \Delta \to \Lambda, \mathbf{G2} & \mathbf{G1}, \Gamma \to \Theta, (\mathbf{G1{\downarrow}H1}) & (\mathbf{G2{\downarrow}H2}), \Delta \to \Lambda \\
\multicolumn{2}{c}{=========================} & \multicolumn{2}{c}{=========================} \\
 & \Sigma \to \Pi, \mathbf{G2} & \mathbf{G1}, \Sigma \to \Pi & \\
 & \dots & \dots & \\
 & \Sigma' \to \Pi', \mathbf{G2} & \mathbf{G1}, \Sigma' \to \Pi' & \\
 & \multicolumn{2}{c}{================} & \\
 & \multicolumn{2}{c}{\dots \quad \Sigma' \to \Pi' \quad \dots} & \\
 & \multicolumn{2}{c}{==================} & \\
 & \multicolumn{2}{c}{\to} &
\end{array}$$

The lemma of 6.2.2.3 asserts that

$$\deg(\mathbf{F}^*, \mathbb{G}^*) \leq \deg(\mathbf{F}, \mathbb{G})$$

where $\mathbf{F}^*$ is an occurrence in $\mathbb{G}^*$ and $\mathbf{F}$ is the corresponding occurrence in $\mathbb{G}$. Should

$$\deg(\mathbf{F}^*, \mathbb{G}^*) = \deg(\mathbf{F}, \mathbb{G})$$

then by lemma 6.2.1,

$$\deg(\mathbf{G1}, \mathbb{G}^*) + 1 = \deg(\mathbf{G2}, \mathbb{G}^*) + 1 = \deg((\mathbf{G1}{\downarrow}\mathbf{H1}), \mathbb{G}^*) = \deg((\mathbf{G2}{\downarrow}\mathbf{H2}), \mathbb{G}^*)$$

The argument provided in section 4.3 of [Gentzen38], or its translation in [Szabo69], or the more detailed argument provided in section 12 of [Takeuti75] can now be used to prove that the ordinal number of the transformed contradiction graph is decreased. A similar argument can be used to show that if the degrees of cut formulas are decreased by the transformation, then the ordinal is not increased.

6.4.4. Axiom-transformation

In this transformation, two edges are replaced with a single edge.

	Height
$\Gamma \to \Theta, \mathbf{A1} \quad \mathbf{A} \to \mathbf{A}$	h1
—————————	
$\Gamma \to \Theta, \mathbf{A}$	
=========	
$\Gamma' \to \Theta', \mathbf{A} \quad \mathbf{A2}, \Delta \to \Lambda$	h1
—————————	
$\Gamma', \Delta \to \Theta', \Lambda$	h2

Let μ be the ordinal of the sequent $\Gamma \to \Theta, \mathbf{A1}$, ν the ordinal of the sequent $\mathbf{A2}, \Delta \to \Lambda$, and μ' the ordinal of the sequent $\Gamma' \to \Theta', \mathbf{A}$. Then $\mu' \geq \mu+1$, and the ordinal of the endsequent is

$$\omega_{h1-h2}(\mu' \# (\nu+1)).$$

The above derivation becomes

	Height
$\Gamma \to \Theta, \mathbf{A1}$	
=========	
$\Gamma' \to \Theta', \mathbf{A1} \quad \mathbf{A2}, \Delta \to \Lambda$	h1
—————————	
$\Gamma', \Delta \to \Theta', \Lambda$	h2

If μ'' is the ordinal of $\Gamma' \to \Theta', \mathbf{A1}$, then $\mu'' < \mu'$, and the ordinal of the endsequent is

$$\omega_{h1-h2}(\mu'' \# \nu).$$

Thus an axiom-transformation reduces the ordinal of a contradiction graph.

BIBLIOGRAPHY

The numbers in parentheses refer to the date of publication. (xx) is the year 19xx.

Beth, E.W.

(55) Semantic Entailment and Formal Derivability, *Mededelingen van de Koninklijke Nederlandse Akademie der Wetenschappen, Afdeling Letterkunde, Nieuwe Reeks,* 18, no.13, 309-342.

Church, Alonzo

(41) The Calculi of Lambda Conversion, Princeton University Press.

(56) Introduction to Mathematical Logic, Vol I, Princeton University Press.

Feferman, Solomon

(77) Categorical Foundations and Foundations of Category Theory, Logic, Foundations of Mathematics and Computability Theory, Editors Butts and Hintikka, D. Reidel, 149-169.

(84) Towards Useful Type-Free Theories, I, *Journal of Symbolic Logic*, March, 75-111.

Fitch, Frederick B.

(52) Symbolic Logic: An Introduction, Ronald Press, New York.

Gentzen, Gerhard

(35) Untersuchungen über das logische Schliessen, *Mathematische Zeitschrift*, 39, 176-210, 405-431.

(38) Neue Fassung des Widerspruchsfreiheitsbeweises für die reine Zahlentheories, *Forschungen zur Logik und zur Grundlegung der exakten Wissenschaften*, New Series, No. 4, 19-44.

Gilmore, P.C. (Paul C.),

(67) The Consistency of Partial Set Theory Without Extensionality, A paper presented at the Summer Institute on Axiomatic Set Theory, Los Angeles, California. It appears in Axiomatic Set Theory, Proceedings of Symposia in Pure Mathematics, 13, part 2, American Mathematical Society, Providence, Rhode Island, 1974, 147-153.

(68) A Formalized Naive Set Theory, a paper presented at the Summer Conference on Intuitionism and Proof Theory, Buffalo, New York.

(71) A Consistent Naive Set Theory: Foundations for a Formal Theory of Computation, IBM Research Report RC 3413, June 22.

(80) Combining Unrestricted Abstraction with Universal Quantification, To H.B. Curry: Essays on Combinatorial Logic, Lambda Calculus and Formalism, Editors J.P. Seldin, J.R. Hindley, Academic Press, 99-123. This is a revised version of [Gilmore71].

(86) Natural Deduction Based Set Theories: A New Resolution of the Old Paradoxes, *Journal of Symbolic Logic*, 51, 393-411.

(87a) The SET Conceptual Model and the Domain Graph Method of Table Design, UBC Computer Science Department Technical Report 87-7.

(87b) Justifications and Applications of the SET Conceptual Model, UBC Computer Science Department Technical Report 87-9.

(88) A Foundation for the Entity Relationship Approach: How and Why, Proceedings of the 6th Entity Relationship Conference, editor S.T. March, North-Holland 95-113.

(89) How Many Real Numbers are There?, UBC Computer Science Department Technical Report 89-7, Revised December, 1990.

Gilmore, Paul C. & Tsiknis, George K.

(90a) A Logic for Category Theory, UBC Computer Science Department Technical Report 90-2, May, 1990.

(90b) Logical Foundations for Programming Semantics, a paper presented to the Sixth Workshop on the Mathematical Foundations of Programming Semantics, Kingston, Ontario, Canada, May 15-19. UBC Computer Science Department Technical Report 90-22, August, 1990.

(90c) A Formalization of Category Theory in NaDSet, a paper presented to the Sixth Workshop on the Mathematical Foundations of Programming Semantics, Kingston, Ontario, Canada, May 15-19. UBC Computer Science Department Technical Report 90-23, August 1990.

(90d) The Lambda Calculus in NaDSet, an extended abstract submitted to MFPS91, Mathematical Foundations of Programming Semantics, Carnegie Mellon University, March 25-28, 1991.

Gödel, Kurt

(40) The Consistency of the Continuum Hypothesis, Annals of Mathematics Studies Number 3, Princeton University Press.

Gray, J.W.

(84) Editor, Mathematical Applications of Category Theory, *Contemporary Mathematics*, 30, American Mathematical Society.

Prawitz, Dag

(65) Natural Deduction, A Proof-Theoretical Study, Stockholm Studies in Philosphy 3, Almquist & Wiksell, Stockholm.

Schütte, K

(60) Beweisetheorie, Springer.

Szabo, M.E. (editor)

(69) The Collected Papers of Gerhard Gentzen, North-Holland

Takeuti, Gaisi

(75) Proof Theory, North-Holland

Tarski, Alfred

(36) Der Wahrheitsbegriff in den formalisierten Sprachen, *Studia Philosophica*, vol. 1. 261-405. English translation appears in Logic, Semantics, Metamathematics, Papers from 1923 to 1938, Oxford University Press, 152-278, 1956.

Tsiknis, George K.

(90) Applications of a Natural Deduction Based Set Theory, UBC Department of Computer Science PhD thesis. In preparation.

The ITLI Prepublication Series

1990

Logic, Semantics and Philosophy of Language

LP-90-01 Jaap van der Does	A Generalized Quantifier Logic for Naked Infinitives
LP-90-02 Jeroen Groenendijk, Martin Stokhof	Dynamic Montague Grammar
LP-90-03 Renate Bartsch	Concept Formation and Concept Composition
LP-90-04 Aarne Ranta	Intuitionistic Categorial Grammar
LP-90-05 Patrick Blackburn	Nominal Tense Logic
LP-90-06 Gennaro Chierchia	The Variablity of Impersonal Subjects
LP-90-07 Gennaro Chierchia	Anaphora and Dynamic Logic
LP-90-08 Herman Hendriks	Flexible Montague Grammar
LP-90-09 Paul Dekker	The Scope of Negation in Discourse, towards a flexible dynamic Montague grammar
LP-90-10 Theo M.V. Janssen	Models for Discourse Markers
LP-90-11 Johan van Benthem	General Dynamics
LP-90-12 Serge Lapierre	A Functional Partial Semantics for Intensional Logic
LP-90-13 Zhisheng Huang	Logics for Belief Dependence
LP-90-14 Jeroen Groenendijk, Martin Stokhof	Two Theories of Dynamic Semantics
LP-90-15 Maarten de Rijke	The Modal Logic of Inequality
LP-90-16 Zhisheng Huang, Karen Kwast	Awareness, Negation and Logical Omniscience

Mathematical Logic and Foundations

ML-90-01 Harold Schellinx	Isomorphisms and Non-Isomorphisms of Graph Models
ML-90-02 Jaap van Oosten	A Semantical Proof of De Jongh's Theorem
ML-90-03 Yde Venema	Relational Games
ML-90-04 Maarten de Rijke	Unary Interpretability Logic
ML-90-05 Domenico Zambella	Sequences with Simple Initial Segments
ML-90-06 Jaap van Oosten	Extension of Lifschitz' Realizability to Higher Order Arithmetic, and a Solution to a Problem of F. Richman
ML-90-07 Maarten de Rijke	A Note on the Interpretability Logic of Finitely Axiomatized Theories
ML-90-08 Harold Schellinx	Some Syntactical Observations on Linear Logic
ML-90-09 Dick de Jongh, Duccio Pianigiani	Solution of a Problem of David Guaspari
ML-90-10 Michiel van Lambalgen	Randomness in Set Theory
ML-90-11 Paul C. Gilmore	The Consistency of an Extended NaDSet

Computation and Complexity Theory

CT-90-01 John Tromp, Peter van Emde Boas	Associative Storage Modification Machines
CT-90-02 Sieger van Denneheuvel, Gerard R. Renardel de Lavalette	A Normal Form for PCSJ Expressions
CT-90-03 Ricard Gavaldà, Leen Torenvliet, Osamu Watanabe, José L. Balcázar	Generalized Kolmogorov Complexity in Relativized Separations
CT-90-04 Harry Buhrman, Edith Spaan, Leen Torenvliet	Bounded Reductions
CT-90-05 Sieger van Denneheuvel, Karen Kwast	Efficient Normalization of Database and Constraint Expressions
CT-90-06 Michiel Smid, Peter van Emde Boas	Dynamic Data Structures on Multiple Storage Media, a Tutorial
CT-90-07 Kees Doets	Greatest Fixed Points of Logic Programs`

Other Prepublications

X-90-01 A.S. Troelstra	Remarks on Intuitionism and the Philosophy of Mathematics, Revised Version
X-90-02 Maarten de Rijke	Some Chapters on Interpretability Logic
X-90-03 L.D. Beklemishev	On the Complexity of Arithmetical Interpretations of Modal Formulae
X-90-04	Annual Report 1989
X-90-05 Valentin Shehtman	Derived Sets in Euclidean Spaces and Modal Logic
X-90-06 Valentin Goranko, Solomon Passy	Using the Universal Modality: Gains and Questions
X-90-07 V.Yu. Shavrukov	The Lindenbaum Fixed Point Algebra is Undecidable
X-90-08 L.D. Beklemishev	Provability Logics for Natural Turing Progressions of Arithmetical Theories
X-90-09 V.Yu. Shavrukov	On Rosser's Provability Predicate
X-90-10 Sieger van Denneheuvel, Peter van Emde Boas	An Overview of the Rule Language RL/1
X-90-11 Alessandra Carbone	Provable Fixed points in $I\Delta_0+\Omega_1$, revised version
X-90-12 Maarten de Rijke	Bi-Unary Interpretability Logic
X-90-13 K.N. Ignatiev	Dzhaparidze's Polymodal Logic: Arithmetical Completeness, Fixed Point Property, Craig's Property
X-90-14 L.A. Chagrova	Undecidable Problems in Correspondence Theory
X-90-15 A.S. Troelstra	Lectures on Linear Logic