Abstract. In this paper I propose a fundamental modification of standard type theory, produce a new kind of type theoretic language, and couch in this language a comprehensive theory of abstract individuals and abstract properties and relations of every type. I then suggest how to employ the theory to solve the four following philosophical problems: (A) the identification and ontological status of Frege's Senses; (B) the deviant behavior of terms in propositional attitude contexts; (C) the non-identity of necessarily equivalent propositions, and (D) the "paradox" of analysis. We can roughly describe these solutions as follows: (A) the senses of English names and descriptions which denote individuals will be modelled as abstract individuals; the senses of English relation denoting expressions of a given type will be modelled as abstract relations of that type. (B) Inside de dicto attitude contexts, these English expressions denote (the abstract objects which serve as) their senses. (C) Relations and propositions will not be identified with their extensions, nor with functions or sets of any kind. They will be taken as primitive, and precise "being" and identity conditions will be proposed consistent with the view that necessarily equivalent relations and propositions may be distinct. (D) With the modelling described in (A), the expressions "being a brother" and "being a male sibling" may both denote the same property, though (the abstract properties which serve as) their senses may differ. Just as Frege predicts, "being a brother just is being a male sibling" is an informative identity statement because the terms flanking the identity sign have the same denotation, though distinct senses.

In this paper, I propose a fundamental modification of standard type theory, produce a new kind of type-theoretic language, and couch in this language a comprehensive theory of abstract individuals and abstract properties and relations of every type. I then suggest how to use the theory to solve the four following philosophical problems: (A) the non-identity of necessarily equivalent propositions; (B) the identification and ontological status of Frege's Senses; (C) the deviant behavior of terms in propositional attitude contexts; and (D) the "paradox" of analysis. We can roughly describe these solutions as follows: (A) Relations and propositions will not be identified with their extensions, nor with functions or sets of any kind. They will be taken as primitive, and precise identity conditions will be proposed consistent with the view that necessarily equivalent relations and propositions may be distinct. (B) The senses of English names and descriptions which denote individuals will be modelled as abstract individuals; the senses of English relation denoting expressions of a given type will be modelled as abstract relations of that
type. (C) Inside de dicto attitude contexts, these English expressions denote (the abstract objects which serve as) their senses. (D) The expressions "being a brother" and "being a male sibling" may both denote the same property, though (the abstract properties which serve as) their senses may differ. Just as Frege predicts, "being a brother just is being a male sibling" is an informative identity statement because the terms flanking the identity sign have the same denotation, though distinct senses.

In Part I of the paper, I describe the theory and discuss (A). In Part II, I discuss (B), (C), and (D). I use Greek lower case letters to serve as metavariables which range over pieces of language and script letters to label semantic items. I use "object" to mean any individual, relation, property, or proposition.

1. We first specify the set types, TYPE. We let "i" be an element of TYPE, as the type of individuals. We also let "p" be an element of TYPE, as the type of propositions. Then we require that where \( t_1, \ldots, t_n \) are any elements of TYPE, then so is \( (t_1, \ldots, t_n)/p \). \( (t_1, \ldots, t_n)/p \) is the type of \( n \)-place relations ("relational type objects") whose arguments have types \( t_1, \ldots, t_n \), respectively. For our purposes, we may suppose that the elements of TYPE are symbols which will be employed to simultaneously categorize both the terms of the language and the entities they denote (or over which the variables range).

Next, we suppose there is an infinite, indexed supply of both constants and variables for each type. Officially we use "\( a_k^i \)" and "\( x_k \)" for all \( k \geq 1 \) and \( t \in TYPE \), as the constants and variables, respectively. However, we will frequently substitute other symbols for these terms, to make the formulas easier to read. Where \( t \) is any type, "\( E!t/p \)" is to be a distinguished constant of type \( t/p \). It will serve as the existence predicate for each type \( t \).

In addition to these primitive terms, we will use two standard connectives: \( \sim \), \&; a quantifier: \( \exists \); a box: \( \Box \); and parentheses to disambiguate.

The syntax of Meinongian type theory differs from standard type theory principally by the fact that there are three kinds, rather than two kinds, of atomic formulas.1 We have the familiar two kinds of "exemplification" formulas: (a) where \( e \) is any term of type \( p \), \( e \) is an atomic exemplification formula; and (b) where \( e \) is any term of type \( (t_1, \ldots, t_n)/p \) and \( \tau_1, \ldots, \tau_n \) are any terms of types \( t_1, \ldots, t_n \), respectively, \( e\tau_1, \ldots, \tau_n \) is an atomic exemplification formula. In addition to these, however, we add atomic "encoding" formulas: where \( e \) is any term of type \( t/p \) and \( \tau \) is any term of type \( t \), \( e\tau \) is an atomic encoding formula. Objects which have type \( t/p \) are properties of objects which have type \( t \). Consequently, atomic

1 The definition of formula which follows shall say nothing about whether formulas are to be considered terms of type \( p \). It will turn out that only certain formulas, those with no encoding subformulas, will also be terms of type \( p \) (therefore, only these formulas will denote propositions).