On Types, Intension and Compositionality

Walid S. Saba

Department of Computer Science
American University of Technology (AUT)
Byblos-Highway, Byblos, Lebanon P. O. Box 20
walid.saba@aut.edu

Abstract. In this paper we demonstrate that a number of challenging problems in the semantics of natural language, namely the treatment of the so-called intensional verbs and the semantics of nominal compounds, can be adequately resolved in the framework of compositional semantics, if a strongly-typed ontological structure is assumed. In addition to suggesting a proper treatment of nominal compounds and intensional verbs within the framework of compositional semantics, we briefly discuss the nature of this ontological type system and how it may be constructed.

1 The Semantics of Nominal Compounds

The semantics of nominal compounds have received considerable attention by a number of authors, most notably (Kamp & Partee, 1995; Fodor & Lepore, 1996; Pustejovsky, 2001), and to our knowledge, the question of what is an appropriate semantics for nominal compounds has not yet been settled. In fact, it seems that the problem of nominal compounds has presented a major challenge to the general program of compositional semantics in the Montague (1973) tradition, where the meaning of a compound nominal such as $[N_1\ N_2]$ is generally given as follows:

$$(1) \quad \|N_1\ N_2\| = F(\|N_1\|, \|N_2\|)$$

In the simplest of cases, the compositional function F is usually taken to be a conjunction (or intersection) of predicates (or sets). For example, assuming that red(x) and apple(x) represent the meanings of red and apple, respectively, then the meaning of a nominal such as red apple is usually given as

$$(2) \quad \|\text{red apple}\| = \{x \mid red(x) \wedge apple(x)\}$$

What (2) says is that something is a red apple if it is red and apple. This simplistic model, while seems adequate in this case (and indeed in many other instances of

© A. Gelbukh (Editor)
Advances in Natural Language Processing
Research in Computing Science 18, 2006, pp. 93-102

similar ontological nature), clearly fails in the following cases, all of which involve an adjective and a noun:

(3) former senator
(4) fake gun
(5) alleged thief

Clearly, the simple conjunctive model, while seems to be adequate for situations similar to those in (2), fails here, as it cannot be accepted that something is former senator if it is former and senator, and similarly for (4) and (5). Thus, while conjunction is one possible function that can be used to attain a compositional meaning, there are in general more complex functions that might be needed for other types of ontological categories. In particular, what we seem to have is something like the following:

(6) $\|\text{red apple}\| = \{x \mid x \text{ is red and } x \text{ is apple}\}$
(7) $\|\text{former senator}\| = \{x \mid x \text{ was but is not now a senator}\}$
(8) $\|\text{fake gun}\| = \{x \mid x \text{ looks like but is not actually a gun}\}$
(9) $\|\text{alleged thief}\| = \{x \mid x \text{ could possibly turn out to be a thief}\}$

It would seem, then, that different ontological categories require different compositional functions to compute the meaning of the whole from the meanings of the parts. In fact, the meaning (intension) of some compound might not be captured without resorting to temporal and/or modal operators. This has generally been taken as an argument against compositionality, in that there does not seem to be an answer as to what the compositional semantic function F in $\|N_1\ N_2\| = F(\|N_1\|, \|N_2\|)$ might be. We believe, however, that this is a fallacious argument in that the problem is not due to compositionality but in 'discovering' a number of semantic functions that could account for all nominal compounds of different ontological categories. Moreover, we believe that the answer lies in assuming a richer type structure than the flat type system typically assumed in Montague-style semantics.

2 Ontology and the Semantics of Adjectives

In (2) we stated that the meaning of some adjectives. The question however is what "kinds" of adjectives are specifically intersective. It would seem that for constructions of the form [A N] where A is a physical property (such as red, large, heavy, etc.) and N is a object of type PhysicalThing (such as car, person, desk, etc.), the meaning of [A N] can be obtained as follows:

(10) $\|A\ N\| = \{x \mid A_{PhysicalProperty}(x) \wedge N_{PhysicalThing}(x)\}$

Note here that the above expression is not a statement about the meaning of any particular adjective. Instead, what (10) simply states is that some adjectives, such as large, heavy, etc. are intersective. Thus, in $\|\text{large table}\| = \{x \mid large(x) \wedge table(x)\}$.

for example, it is assumed that the meaning of large, namely the predicate large(x) has been defined. Although the semantics of such adjectives is not our immediate concern here, it must be pointed out that semantics of such (intersective) adjectives, which are presumably the simplest, can be quite involved, as these adjectives are very context-sensitive – clearly the sense of 'large' in 'large elephant' is quite different from the sense of 'large' in 'large bird'. Assuming a predicate $typical_{a,T}(x)$, which is true of some object x of type T if x is a typical object with respect to one of its attributes a is defined, then the meanings of such adjectives as large and heavy, for example, could be defined as follows, where $x :: T$ refers to an object x of type T:

(11) $$large \Rightarrow (\forall x :: PhysicalThing)(large(x) \equiv_{df} \lambda P[P(x) \wedge (\exists y :: PhysicalThing)(P(y) \wedge typical_{size}(y) \wedge size(x,s_1) \wedge size(y,s_2) \wedge (s_1 >> s_2))])$$

(12) $$heavy \Rightarrow (\forall x :: PhysicalThing)(heavy(x) \equiv_{df} \lambda P[P(x) \wedge (\exists y :: PhysicalThing)(P(y) \wedge typical_{weight}(y) \wedge weight(x,w_1) \wedge weight(y,w_2) \wedge (w_1 >> w_2))])$$

What (1) and (2) say is the following: that some P object x, is a large (heavy) P, iff it has a size (weight) which is larger than the size (weight) of another P object, y, which has a typical size (weight) as far as P objects go. It would seem, then, that the meaning of such adjectives is tightly related to some attribute (large/size, heavy/weight, etc.) of the corresponding concept. Thus, such adjectives, while they are intersective, are context-dependent: their meaning is fully specified only in the context of a specific concept.

One of the main points that we like to make in this paper is that, like intersective adjectives, non-intersective adjectives also have a compositional meaning, although the compositional function might be more involved than simple conjunction. For example, we argue that the following are reasonable definitions for fake, former and alleged:

(13) $$(\forall x :: \textbf{PhysicalArtifact})(fake(x) \equiv_{df} \lambda P[(\exists y :: \textbf{Physical})(\neg P(x) \wedge P(y) \wedge similar_{\{shape,size\}}(x,y))]$$

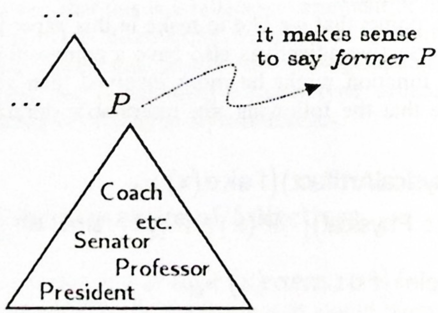
(14) $$(\forall x :: \textbf{Role})(former(x) \equiv_{df} \lambda P[(\exists t)((t < now) \wedge P(x,t) \wedge \neg P(x,now))])$$

(15) $$(\forall x :: \textbf{Role})(alleged(x) \equiv_{df} \lambda P[(\exists t)((t > now) \wedge \neg P(x,now) \wedge \Diamond P(x,t))]$$

That is, 'fake' applies to some concept P as follows: a certain physical object x is a fake P iff it is not a P, but looks like (in certain respects) to something else, say y, which is actually a P. On the other hand, what (14) says is the following: a certain x is a former P iff x was a P at some point in time in the past and is not now a P. Finally, what (15) says is that something is an 'alleged' P iff it is not now known to be a P, but could possibly turn out to be a P at some point in the future.

It is interesting to note here that the intension of fake and that of former and alleged was in one case represented by recourse to possible worlds semantics (the case of (14) and (15)), while in (13) the intension uses something like structured semantics, assuming that $similar_{\{A_1,A_2,\ldots,A_n\}}(x,y)$ which is true of some x and some y if x and y share a number of important features, is defined. What is interesting in this is that it suggests that possible-worlds semantics and structured semantics are not two distinct alternatives to representing intensionality, as has been suggested in the literature, but that in fact they should co-exist.

Additionally, several points should also be made here. First, the representation of the meaning of fake given in (13) suggests that fake expects a concept which is of type PhysicalArtifact, and thus something like fake idea, or fake song, for example, should sound meaningless, from the standpoint of commonsense[1]. Second, the representation of the meaning of former given in (14) suggests that former expects a concept which has a time dimension, i.e. is a temporal concept. Finally, we should note here that our ultimate goal of this type of analysis is to discover the ontological categories that share the same behavior. For example, an analysis of the meaning of former, given in (14), suggests that there are a number of ontological categories that seem to share the same behavior, and could thus replace P in (14), as implied by the fragment hierarchy below.



3 Types, Predicates and Logical Concepts

In "Logic and Ontology" Cocchirarella (2001) argues for a view of logic as a language in contrast with the view of logic as a language. In the latter, logic is

[1] One can of course say fake smile, but this is clearly another sense of fake. While fake gun refers to a gun (which is of type Artifact) that is not real, fake smile refers to a dishonest smile, or a smile that is not genuine.

viewed as an "abstract calculus that has no content of its own, and which depends on set theory as a background framework by which such a calculus might be syntactically described and semantically interpreted." In the view of logic as a language, however, logic has content, and "ontological content in particular." This view however necessitates the use of type theory, as opposed to set theory as the background framework. It is this view that we advocate here, and in our opinion, problems in the semantics of natural language cannot be resolved until a logic that is grounded in type theory and predication (as opposed to set membership) is properly formulated. In this section we discuss the building blocks of such a program.

3.1 Types vs. Predicates

In formal (programming) languages we write statements such as '`int x`', which is a type declaration statement meaning that `x` is an object of type `int`. However, in programming languages we do not have procedures that verify (somehow) if some object is of a certain type - that is we do not have a predicate such as `int(x)` that takes some object `x` and returns 'true' if `x` is an `int` and 'false' otherwise. Clearly, the type and the corresponding predicate are related, and in particular, a predicate such as `int(x)` is true of some object `x` if has all the properties of the type `int`.

Like objects in formal (programming) languages, commonsense objects have a type, and a corresponding predicate that verifies if a certain object is of a specific type. For instance, our ontology has a type hierarchy that contains the following fragment:

(16) $\mathbf{Piano} \supset \mathbf{Instrument} \supset \ldots \supset \mathbf{Artifact} \supset \ldots \supset \mathbf{PhysicalThing} \supset \mathbf{Thing}$

Corresponding to these types there are predicates such as `piano(x)`, `instrument(x)`, etc. Moreover, a predicate such as `piano(x)` is true of some object `x` just in case `x` happens to be a piano. That is, such concepts correspond to what Cocchirarella (2001) refers to as 'first intentions', i.e., concepts abstracted directly from physical reality. The point here is that what makes some object `x` a `piano`, for example (or, what makes `piano(x)` true of some object `x`) is determined directly from physical reality. Such 'first intentions' concepts should be contrasted with concepts that are about 'second intentions', which, according to Cocchirarella are "concepts abstracted wholly from the 'material' content of first intentions", using the logical apparatus. Thus first intention concepts are in some sense 'ontological concepts', while second intention concepts can be thought of as 'logical concepts'.

Continuing with our example, `piano(x)` would be an ontological concept, while `pianist(x)`, for example, is a concept that is logically defined using the concept `piano(x)`, and perhaps other 'first intention' concepts. In other words, what makes `pianist(x)` true of some `x` is not physical reality but some set of logical conditions. This can be stated as follows:

(17) $(\forall x :: \mathbf{Artifact})(\mathrm{piano}(x) \equiv_{df} \mathrm{NueralNetPatternRecogProc}(x))$

(18) $(\forall x :: \textbf{Human})(\text{pianist}(x)$
$$\equiv_{df} (\exists a :: \textbf{Activity})(\exists p :: \textbf{Artifact})(\text{playing}(a) \wedge \text{piano}(p) \wedge \text{agent}(a,x) \wedge \text{object}(a,\textbf{Music}) \wedge \text{instrument}(a,p))$$

That is, something is a piano if it looks like, sounds like, feels like, etc. what we call 'piano'. On the other hand, what (18) says is the following: any Human x, is a pianist, iff there is some playing activity and some Artifact which is a piano where the object of this playing activity is Music and the instrument of this activity is a piano.

3.2 Compound Nominals Revisited

The problem of compound nominals in the case of noun-noun combinations has traditionally been due to the various relations that are usually implicit between the nouns (see Weiskopf, forthcoming). For example, consider the following:

(19) $\|\text{brick house}\| = \{x \mid x \text{ is a house that is } \textbf{made of} \text{ brick}\}$
(20) $\|\text{dog house}\| = \{x \mid x \text{ is a house that is } \textbf{made for} \text{ a dog}\}$
(21) $\|\text{beer drinker}\| = \{x \mid x \text{ often drinks beer}\}$
(22) $\|\text{beer factory}\| = \{x \mid x \text{ is a factory that } \textbf{makes} \text{ beer}\}$

Thus, while a brick house is a house 'made of' bricks, a dog house is a house that is 'made for' a dog. It would seem, then, that the relation implicitly implied between the two nouns differ with different noun-noun combinations. However, assuming the existence of a strongly-typed ontology might result in identifying a handful of implicit relations that can account for all patterns. Consider for example the following:

(23)
$$\|\text{brick house}\| = \{x : \textbf{Artifact} \mid \text{house}(x) \wedge (\exists y : \textbf{Substance})(\text{brick}(y) \wedge \text{madeOf}(x,y))\}$$

(24)
$$\|\text{paper cup}\| = \{x : \textbf{Artifact} \mid \text{cup}(x) \wedge (\exists y : \textbf{Substance})(\text{paper}(y) \wedge \text{madeOf}(x,y))\}$$

(25)
$$\|\text{plastic knife}\| = \{x : \textbf{Artifact} \mid \text{knife}(x) \wedge (\exists y : \textbf{Substance})(\text{plastic}(y) \wedge \text{madeOf}(x,y))\}$$

It would seem, therefore, that the same semantic relation, namely madeOf, is the relation that is implicit between all $[N_1\ N_2]$ combinations when N_1 is an Artifact and N_2 is a Substance. Similarly, it would seem that the same semantic relation underlies all $[N_1\ N_2]$ combinations when N_1 is a Human and N_2 is a Substance, where P should be read as 'it is often the case that P', or 'generally, P':

(26) $\|\text{beer drinker}\| = \{x :: \textbf{Human} \mid (\exists y :: \textbf{Substance})(\text{beer}(y) \wedge \Delta(\text{drinks}(x,y)))\}$

(27) $\|\text{cigar smoker}\| = \{x :: \textbf{Human} \mid (\exists y :: \textbf{Substance})(\text{cigar}(y) \wedge \Delta(\text{smokes}(x,y)))\}$

4 The So-Called Intensional Verbs

In (Montague, 1969) Montague discusses a puzzle pointed out to him by Quine which can be illustrated by the following examples:

(16) $\|\text{John painted a unicorn}\| = (\exists x)(\text{unicorn}(x) \wedge \text{painted}(j,x))$

(17) $\|\text{John found a unicorn}\| = (\exists x)(\text{unicorn}(x) \wedge \text{found}(j,x))$

The puzzle Quine was referring to was the following: both translations admit the inference $(\exists x)(\text{unicorn}(x))$ – that is, both sentences imply the existence of a unicorn, although it is quite clear that such an inference should not be admitted in the case of (17). According to Montague, the obvious difference between (16) and (17) must be reflected in an ontological difference between find and paint in that the extensional type $(e \rightarrow (e \rightarrow t))$ both transitive verbs are typically assigned is too simplistic. Montague was implicitly suggesting that a much more sophisticated ontology (i.e., a more complex type system) is needed, one that would in fact yield different types for find and paint. One reasonable suggestion for the types of find and paint, for example, could be as follows:

(18) find :: $(e_{\textbf{Animal}} \rightarrow (e_{\textbf{Thing}} \rightarrow t))$

(19) paint :: $(e_{\textbf{Human}} \rightarrow (e_{\textbf{Representation}} \rightarrow t))$

Thus instead of the flat type structure implied by $(e \rightarrow (e \rightarrow t))$, the types of find and paint should reflect our commonsense belief that we can always speak of some **Animal** that found something (i.e., any **Thing** whatsoever), and of a **Human** that painted some illustration, or as we called it here a Representation. Before we proceed, however, we point out that throughout, we will use this Font for concept types in the ontology, and this font for predicate names. Thus, **x:LivingThing** means x is an object/entity of type LivingThing and apple(x) means the predicate or property apple is true of x. Note, further, that in a flat-type system, the expression $(\exists x)(\text{unicorn}(x) \wedge \text{found}(j,x))$ is equivalent to the typed expression $(\exists x :: \textbf{Entity})(\text{unicorn}(x) \wedge \text{found}(j :: \textbf{Entity}, x))$ since in flat type system there is only one type of entity. With this background, the correct translations of (18) and (19) and the corresponding inferences can now be given as follows:

(20) $(\exists x{:}\textbf{Thing})(\text{unicorn}(x) \wedge \text{found}(j{:}\textbf{Rational}, x))$

$\Rightarrow (\exists x{:}\textbf{Thing})(\text{unicorn}(x))$

$\Rightarrow (\exists x{:}\textbf{Thing})(\text{found}(j{:}\textbf{Rational}, x))$

(21) $(\exists x{:}\mathrm{Representation})(\mathrm{unicorn}(x) \wedge \mathrm{painted}(j{:}\mathrm{Rational}, x))$
$\Rightarrow (\exists x{:}\mathrm{Representation})(\mathrm{unicorn}(x))$
$\Rightarrow (\exists x{:}\mathrm{Representation})(\mathrm{painted}(j{:}\mathrm{Rational}, x))$

Adding a rich type structure to the semantics, it seems, provides a reasonable solution to Quine's puzzle, as the correct inferences can now be made: if John found a unicorn, then one can indeed infer that an actual unicorn exists[2]. However, the painting of a unicorn only implies the existence of a representation (an illustration) of something we call a unicorn! Stated yet in other words, (7) implies that a unicorn Thing (including perhaps a unicorn Toy) exists, while (8) implies a unicorn Representation exists. There are two points that this discussion intends to emphasize: (i) is the need for a rich type structure to solve a number of problems in the semantics of natural language; and (ii) that this type structure is actually systematically discovered by an analysis of how ordinary language is used to talk about the world.

5 Language, Logic, Ontology and Commonsense

Our work here has been motivated by the (rather strong) claim of Richard Montague (see the paper on ELF in (Thomasson, 1974)) that there is no theoretical difference between formal and natural languages. If does turn out that Montague is correct (as we believe to be the case), then there should exist a formal system, much like arithmetic, or any other algebra, for concepts, as has been advocated by a number of authors, such as Cresswell (1973) and Barwise (1989), among others. What we are arguing for here is a formal system that explains how concepts of various types combine, forming more complex concepts in a formal, strongly-typed system. To illustrate, consider the following:

(22) artificial :: NaturalKind → Artifact
(23) flower :: Plant
(24) flower :: Plant ⊃ LivingThing
(25) flower :: Plant ⊃ LivingThing ⊃ NaturalKind
(26) artificial flower :: Artifact

What the above says is the following: artificial is a function that takes a NaturalKind and returns an Artifact (22); a flower is a Plant (23); a flower is a Plant which in turn is a LivingThing (24); a flower is a Plant, which is a LivingThing, which in turn is a NaturalKind (25); and, finally, an artificial flower is an Artifact (26). Therefore, 'artificial c', for some NaturalKind c, should in the final analysis have the same properties that any other Artifact has. Thus, while a flower, which is of type Plant, and is therefore a LivingThing, grows, lives and dies like any other LivingThing, an

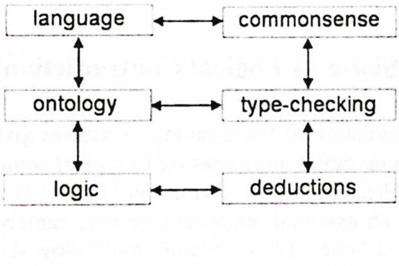
[2] Of course, in such a type system we would have Rational ⊃ Animal and therefore John, an entity of type Rational, can be the subject of found which expects an entity of type Animal.

artificial flower, and like any other Artifact, is something that is manufactured, does not grow, does not die, but can be assembled, destroyed, etc. The concept algebra we have in mind should also systematically explain the interplay between what is considered commonsense at the linguistic level, type checking at the ontological level, and deduction at the logical level. For example, the concept artificial car, which is a meaningless concept from the standpoint of commonsense, is ill-typed since Car is an Artifact, and Artifact does not unify with NaturalKind – neither type is a sub-type of the other.

The concept former father, on the other hand, which is also a meaningless concept from the standpoint of commonsense, escapes type-checking since father, which is a Role, is a type that former expects as shown in (29) below.

(29) former :: **Role** → **Role**

However, although former father escapes type-checking, the fact that this a meaningless concept from the standpoint of commonsense, is ultimately detected at the logical level by resulting in a contradiction as shown in the appendix. Thus what is meaningless at the linguistic level should be flagged at the type-checking level, or, if happens to escapes type-checking, such as former father, it should eventually result in a logical contradiction at the logical level (see the appendix concerning former father). The picture we have in mind can therefore be summarized as shown in the figure below.



6 Concluding Remarks

A number of problems in the semantics of natural language can be resolved in a compositional semantic framework if a rich type system that models an ontology of commonsense concepts can be assumed. If this where to happen, it would mean that there is a formal system that underlies natural language and that a concept algebra must exist. This subsequently means that the ontology we have in mind must be systematically discovered and cannot be invented, as has been argued by Saba (2001). In this paper we have shown that assuming such a rich type systems can help resolving a number of challenging problems in the semantics of natural language. For lack of space, in this paper we could not discuss the nature of this ontological structure, the corresponding strongly-typed meaning algebra, and how this structure might be discovered rather than invented, using natural language itself as a guide in this process. Some of these issues are discussed in some detail in Saba (2006).

References

1. Barwise, J. (1989). The Situation in Logic. CSLI Publications, Stanford.
2. Cocchiarella, N. B. (2001). Logic and Ontology, Axiomathes 12, pp. 117-150.
3. Cresswell, M. J. (1973). Logics and Languages. Methuen & Co., London.
4. Kamp, H. & B. Partee. 1995. Prototype theory and Compositionality, Cognition 57, pp. 129–191.
5. Montague, R., 1960. On the Nature of Certain Philosophical Entities, The Monist, 53, pp. 159–194
6. Montague, R., 1974. In Thomason, R. (1974) (Ed.), Formal Philosophy: Selected Papers of Richard Montague. Yale University Press.
7. Pustejovsky, J. (2001). Type Construction and the Logic of Concepts, In P. Bouillon and F. Busa (eds.), The Syntax of Word Meanings, Cambridge University Press.
8. Saba, W. S. (2006). The Structure of Commonsense Knowledge, in Paolo Valore (Ed.), Topics on General and Formal Ontology, pp. 221-243, Polimetrica International Scientific Publisher.
9. Saba, W. S. (2001). Language and Commonsense Knowledge, In Brooks, M., Corbett, D., Stumptner, M., (Eds.), AI 2001: Advances in Artificial Intelligence, LNAI, Vol. 2256, pp. 426-437, Springer.
10. Weiskopf, D. A. (forthcoming). Compound nominals, Context and Compositionality, to appear in Synthese.

Appendix: (Non-Sense → Logical Contradiction)

Using the logical formulation of the meaning of former given above, we show here how the concept 'former father' translates into a logical contradiction.

First, we reiterate the meaning of 'former' in (1). In (2) we state the fact that the role type Father has an essential temporal property, namely that once someone is a father they are always a father. The deductions that follow should be obvious.

1. $(\forall x : \text{Role})\left(former(x) \equiv_{df} \lambda P\left[(\exists t)((t < now) \wedge P(x,t) \wedge \neg P(x, now))\right]\right)$
2. $(\forall x)\left((\exists t_1)\left(father(x,t_1) \supset (\forall t_2)((t_2 > t_1) \supset father(x,t_2))\right)\right)$
3. $(\exists t)((t < now) \wedge father(x,t) \wedge \neg father(x, now))$ (1) applied on father
4. $(t < now) \wedge father(x,t) \wedge \neg father(x, now)$ EI of (3)
5. $father(x,t)$ $\wedge$ - elimination of (4)
6. $(\exists t_1)\left(father(x,t_1) \supset (\forall t_2)((t_2 > t_1) \supset father(x,t_2))\right)$ UG of (2)
7. $father(x,u) \supset (\forall t_2)((t_2 \geq u) \supset father(x,t_2))$ EI of (6)
8. $(\forall t_2)((t_2 > t) \supset father(x,t_2))$ (5), (7) and MP
9. $(t_2 > t) \supset father(x,t_2)$ UG of (8)
10. $(t < now)$ $\wedge$ - elimination of (4)
11. $father(x, now)$ (9), (10) and MP
12. $\neg father(x, now)$ $\wedge$ - elimination of (4)
13. $\perp$ (11) and (12)