Adjective based inference*

Marilisa Amoia
INRIA/Université de Nancy 1 &
University of the Saarland
Saarbrücken Germany
amoia@coli.uni-sb.de

Claire Gardent
CNRS/Loria
Campus Scientifique BP 239
54506 Vandoeuvre-les-Nancy, France
claire.gardent@loria.fr

Abstract

In this paper, we propose a fine grained classification of english adjectives geared at modeling the distinct inference patterns licensed by each adjective class. We show how it can be implemented in description logic and illustrate the predictions made by a series of examples. The proposal has been implemented using Description logic as a semantic representation language and the prediction verified using the DL theorem prover RACER.

Topics: Textual Entailment, Adjectival Semantics

1 Introduction

Understanding a text is one of the ultimate goals of computational linguistics. To achieve this goal, systems need to be developed which can construct a meaning representation for any given text and which furthermore, can reason about the meaning of a text. As is convincingly argued in (rte, 2005), one of the major inference task involved in that reasoning is the entailment recognition task:

Does text T_1 entail text T_2?

Indeed entailment recognition can be used to determine whether a text fragment answers a question (e.g., in question answering application), whether a query is entailed by a relevant document (in information retrieval), whether a text fragment entails a specific information nugget (in information extraction), etc.

Because the Pascal RTE challenge focuses on real text, the participating systems must be robust that is, they must be able to handle unconstrained input. Most systems therefore are based on statistical methods (e.g., stochastic parsing and lexical distance or word overlap for semantic similarity) and few provide for a principled integration of lexical and compositional semantics. On the other hand, one of the participant teams has shown that roughly 50% of the RTE cases could be handled correctly by a system that would adequately cover semantic entailments that are either syntax based (e.g., active/passive) or lexical semantics based (e.g., bicycle/bike). Given that the overall system accuracies hovered between 50 and 60 percent with a baseline of 50 %[1], this suggests that a better integration of syntax, compositional and lexical semantics might improve entailment recognition accuracy.

In this paper, we focus on the case of adjectives and explore the entailment patterns that are supported by the interaction of their lexical and of their compositional semantics. We start by defining a classification schema for adjectives based on their syntactic and semantic properties. We then associate with each class a set of axioms schemas and of semantic construction rules and we show that these correctly predicts the observed entailment patterns. For instance, the approach will account for the following (non)-entailment cases:

(1) a. John frightened the child
$\models$ The child is afraid

b. John is an alledged murderer
$\models$ Peter claims that John is a murderer
$\not\models$ John is a murderer

c. This is a fake bicycle
$\models$ This is a false bike
$\models$ This is not a real bike
$\not\models$ This is a bike

*We thank la Région Lorraine, INRIA and the University of Sarrebruecken for partially funding the research presented in this paper.

[1] 50% of the cases were true entailment and 50% were false ones, hence tossing a coin would get half of the cases right.

d. John is not awake
$\models$ John sleeps
$\not\models$ John does not sleep

The approach is implemented using Description Logic as a semantic representation language and tested on a hand-built semantic test suite of approximately 1 000 items. In the latter part of the paper we discuss this testsuite and the philosophy behind it.

2 A fine grained classification for adjectives

As mentioned above, we semantically classify adjectives based on their lexical, their model theoretic and their morpho-derivational properties. To facilitate the link with compositional semantics (the construction of a meaning representation for sentences containing adjectives), we also take into account syntactic properties such as the predicative/attributive or the static/dynamic distinction. We now detail each of these properties. The overall categorisation system is given in Figure 1.

2.1 Model theoretic properties

The main criteria for classification are given by (Kamp, 1975; Kamp and Partee, 1995) semantic classification of adjectives which is based on whether it is possible to infer from the Adj+N combination the Adj or the N denotation.

Intersective adjectives (e.g., *red*) licence the following inference inference patterns:

$$A + N \models A$$

$$A + N \models N$$

For instance, if *X is a red car* then *X is a car* and *X is red*

Subsective adjectives (e.g., *big*) licence the following inference pattern:

$$A + N \models N$$

For instance, if *X is a big mouse*, then *X is a mouse* but it is not necessarily true *X is big*

Privative adjectives licence the inference pattern:

$$A + N \models \neg N$$

For instance, if *X is a fake gun* then *X is not a gun*

Plain non-subsective adjectives (e.g., *alledged*) do not licence any inference
For instance, if *X is an alleged murderer* then it is unknown whether *X is a murderer or not*

2.2 Lexical semantics

From the lexical semantics literature, we take one additional classification criterion namely antonymy. This term covers different kinds of opposite polarity relations between adjectives namly, binary opposition, contraries and multiple oppositions.

Binary oppositions covers pairs such as *wet/dry* which license the following inference pattern:

$$A1 \equiv \neg A2 \ \wedge \ \neg A1 \equiv A2$$

So that in particular:

$$wet \equiv \neg dry \ \wedge \ \neg wet \equiv dry$$

Contraries are pairs such as *long/short* where the implication is unidirectional:

$$A1 \models \neg A2 \ \wedge \ \neg A1 \not\models A2$$
$$A2 \models \neg A1 \ \wedge \ \neg A2 \not\models A1$$

and in particular:

$$long \models \neg short \wedge \neg long \not\models short$$
$$short \models \neg long \wedge \neg short \not\models long$$

Multiple oppositions involve a finite set of adjectives (e.g., *linguistic/economic/mathematical/...*) which are pairwise mutually exclusive. For a set of opposed adjectives $A_1 \ldots A_n$, the following axioms schemas will be licensed:

$$\forall i, j \ s.t.\ 1 \leq i, j \leq and\ i \neq j$$
$$A_i \models \neg A_j \quad and \quad \neg A_i \not\models A_j$$

2.2.1 Derivational morphology

We also take into account related forms that is, whether there exists a noun or a verb that is semantically related to the adjectives being considered. Moreover, for nominalizations we distinguish whether the morphologically related noun is an event noun or a noun denoting a theta role of the related verb.

As we shall see, this permits capturing entailment relations between sentences containing morphoderivational variants such as for instance :

(2) a. John is asleep (Adjective → Verb)
$\models$ John sleeps

b. John is absent (Adj. → θ−role Noun)
$\models$ John is the absentee

c. John is deeply asleep (Adj. → evt N.)
$\models$ John's sleep is deep

2.2.2 Syntactic properties

To better support the syntax/semantic interface, we refine the semantic classes distinguishable on the basis of the above criteria with the following syntactic ones taken from (Quirk et al., 1985).

Attributiveness/Predicativeness. English adjectives can be divided in adjectives which can be used only predicatively (such as *alone*), adjectives which can be used only attributively (such as *mechanical* in *mechanical enginner*) and adjectives which can be used in both constructions such as *red*.

Modifi ability by *very*. We distinguish between adjectives such as *nice* which can be modified by *very* (i.e. *very nice*) and adjectives such as *alleged* which cannot (**very alleged*).

Staticity/Dynamicity. Dynamic adjectives can be used in imperative constructions and in the progressive form (*Be reasonable, He is being reasonable*), static adjectives cannot (**Be short, He is being short*).

3 Semantic Classes and textual entailment recognition

As Figure 1 shows, the proposed classification includes 15 adjective classes, each with distinct syntactic and semantic properties.

To account for these differences, we define for each class a set of axiom schemas capturing the model theoretic, lexical semantics and morpho-derivational properties of that class. Based on some basic syntactic patterns, we then show that these axioms predict the observed textual entailment patterns for that class.

Before we illustrate this approach by means of an example, we first show how we capture logical entailment between NL semantic representations in a description logic setting.

3.1 Using description logic to check entailment between NL sentences

As argued in (Gardent and Jacquey, 2003), description logic (DL) is an intuitive framework within which to perform lexical reasoning: it is efficient (basic versions of description logics are decidable), it is tailored to reason about complex taxonomies (taxonomies of descriptions) and it is equipped with powerful, freely available automated provers (such as RACER, (Volker Haarslev, 2001)). For these reasons, we are here exploring a DL encoding of the entailment recognition task for the set of examples we are considering.The particular language we assume has the following syntax.

$$C, D \rightarrow A|\top|\bot|\neg A \mid C \sqcap D \mid C \sqcup D \mid \forall R.C \mid \exists R.C$$

The semantics of this language is given below with Δ the domain of interpretation and I the interpretation function which assigns to every atomic concept A, a set $A^I \subseteq \Delta$ and to every atomic role R a binary relation $R^I \subseteq \Delta \times \Delta$.

$$\begin{array}{rcl}
\top^I & = & \Delta \\
\bot^I & = & \emptyset \\
(\neg A)^I & = & \Delta \backslash A^I \\
(C \sqcap D)^I & = & C^I \cap D^I \\
(C \sqcup D)^I & = & C^I \cup D^I \\
(\forall R.C)^I & = & \{a \in \Delta \mid \forall b(a,b) \in R^I \rightarrow b \in C^I\} \\
(\exists R.C)^I & = & \{a \in \Delta \mid \exists b \in C^I \wedge (a,b) \in R^I n\}
\end{array}$$

Now one basic problem with using DL to check entailment between NL expressions, is that DL formulae are "directional" in that they refer to a given set of individuals. For instance the sentence *The boat is floating* might be represented by either of the two formulae given in 3 but these two formulae do not stand in an entailment relation (since they refer to different kind of objects namely floating event of a boat in 3a and boats that float in 3b).

(3) a. float $\sqcap\exists$theme.boat

b. boat $\sqcap\exists$theme^{-1}.float

To remedy this shortcoming, we introduce the notion of a *rotation*. Given a DL formula which only contains conjunction (disjunction is translated in DL as different formulas)
Φ = $\sqcap_{i=1,n}$ Event$_i$ $\sqcap_{j=1,m}$ $\exists$R$_j$.Type$_j$
a rotation of this formula is defined as:

1. Φ
2. $\forall j \in \{1, ..., m\}$:
$Type_j$ $\sqcap$ $\exists$R$_j^{-1}$.($\sqcap_{i=1,n}$Event$_i$ $\sqcap_{1<k<j,j<k<m}$ $\exists$R$_k$.Type$_k$)

Adjective Class	Predicative/Attributive	Modifi able by *very*	static/dynamic	Antonymy	Related forms	Semantic class
Class 1: *afloat*	predicative-only	-	static	multi-opposition	V_a, N_e, N_θ	intersective
Class 2: *asleep*	predicative-only	+	static	binary-opposition	V_a, N_e, N_θ	intersective
Class 3: *polite*	both	+	dynamic	contraries	N_a	intersective
Class 4: *dry*	both	+	static	binary-opposition	V_a, N_e, N_θ	intersective
Class 5: *open*	both	-	dynamic	binary-opposition	V_a, N_e, N_θ	intersective
Class 6: *male*	both	-	static	multi-opposition	$N_a, N_e,$	intersective
Class 7: *authentic*	both	+	static	binary-opposition	N_e	intersective
Class 8: *big*	both	+	static	contraries	N_e	subsective
Class 9: *good*	both	+	dynamic	contraries	N_e	subsective
Class 10: *cultural*	attributive-only	-	static	multi-opposition	N_a	subsective
Class 11: *recent*	attributive-only	+	static	multi-opposition	N_e	subsective
Class 12: *fake*	both	-	static	binary-opposition	V_a,N_e	privative
Class 13: *former*	attributive-only	-	static	multi-opposition	-	privative
Class 14: *questionable*	both	+	static	contraries	V_a, N_e	plain non-subsective
Class 15: *alleged*	attributive-only	-	static	contraries	V_a	plain non-subsective

Figure 1: Classes of Adjectives (in column 6, N_e indicates a related noun that denotes an event and N_θ noun denoting a theta role of a related verb).

so that the formula:
$\text{Event}_1 \sqcap \text{Event}_2 \sqcap ... \sqcap \text{Event}_n \sqcap \exists R_1.\text{Type}_1 \sqcap \exists R_2.\text{Type}_2 ... \sqcap \exists R_n.\text{Type}_n$
corresponds to the following n Rotations each of which describe the same situation from the point of view of a particular type

0. Event $\sqcap \exists R_1.\text{Type}_1 \sqcap \exists R_2.\text{Type}_2 ... \sqcap \exists R_n.\text{Type}_n$
   $\subseteq$ Event
1. $\text{Type}_1 \sqcap \exists R_1^{-1}.(\text{Event} \sqcap \exists R_2.\text{Type}_2 ... \sqcap \exists R_n.\text{Type}_n)$
   $\subseteq \text{Type}_1$
2. $\text{Type}_2 \sqcap \exists R_2^{-1}.(\text{Event} \sqcap \exists R_1.\text{Type}_1 ... \sqcap \exists R_n.\text{Type}_n)$
   $\subseteq \text{Type}_2$
   ...

n. $\text{Type}_n \sqcap \exists R_n^{-1}.(\text{Event} \sqcap \exists R_1.\text{Type}_1 ... \sqcap \exists R_{n-1}.\text{Type}_{n-1})$
   $\subseteq \text{Type}_n$

So for example, the sentence *Mary knows that John is the inventor of the radio* will be represented as a predicate logic formula
$\exists x_1 mary(x_1) \wedge \exists x_2 john(x_2) \wedge \exists x_3 radio(x_3) \wedge \exists e_1 know(e_1) \wedge \exists agent(e_1, x_1) \wedge \exists topic(e_1, e_2) \wedge \exists e_2 invent(e_2) \wedge agent(e_2, x_2) \wedge patient(e_2, x_3)$
the denotation of this PL formula corresponds to the set of individual $\{x_1, x_2, x_3\} \cup \{e_1, e_2\}$. The corresponding DL representation will be the underspecified representation
know $\sqcap\exists$ agent.mary $\sqcap\exists$ topic.(invent $\sqcap\exists$agent.john $\sqcap\exists$ patient.radio)
the denotation of which corresponds to the set $\{e_1\}$ and all its rotations which permitt to access the other sets of individuals asserted in the sentence:
Rotation_0: know $\sqcap\exists$ agent.mary $\sqcap\exists$ topic.(invent $\sqcap\exists$agent.john $\sqcap\exists$ patient.radio)
Rotation_1: mary $\sqcap\exists$ agent^{-1}.(know $\sqcap\exists$ topic.(invent $\sqcap\exists$agent.john $\sqcap\exists$ patient.radio))
Rotation_2: (invent $\sqcap\exists$agent.john $\sqcap\exists$ patient.radio) $\sqcap\exists\text{topic}^{-1}$.(know $\sqcap\exists$agent.mary)
Rotation_3: john $\sqcap\exists\text{agent}^{-1}$.(invent $\sqcap\exists$patient.radio $\sqcap\exists\text{topic}^{-1}$.(know $\sqcap\exists$agent.mary))
Rotation_4: radio $\sqcap\exists\text{patient}^{-1}$.(invent $\sqcap\exists$agent.john $\sqcap\exists\text{topic}^{-1}$.(know $\sqcap\exists$agent.mary))

Finally, we say that an arbitrary formula/representation Φ_1 implies the formula Φ_2 iff it is possible to find a rotation $Rotation_i$ of Φ_1 the denotation of which describes a subset of the denotation of Φ_2:

Defi nition

$$\Phi_1 \models \Phi_2 \; iff \; \exists i.Rotation_i(\Phi_1) \sqsubseteq \Phi_2 \qquad (1)$$

3.2 Example class axioms and derivations

We now illustrate our approach by looking at two classes in more detail namely, class 1 and class 8.

3.2.1 Class 1

Syntactically, Class 1 contains adjectives like *adrift*,*afloat*,*aground* which can only be used predicatively, are non gradable and cannot be modified by *very*. Semantically, they are intersective adjectives which enter in multiple opposition relations with other adjectives. They are furthermore morphologically derived from verbs and can be nominalized. To reflect these semantic properties we use the following axioms.

Model theoretic semantics. Adjectives of class 1 are intersective adjective. They will thus licence the correponding inference patterns namely:

$$A + N \models A \qquad (2)$$
$$A + N \models N \qquad (3)$$

Lexical semantics. Adjectives of class 1 enter in multiple opposition relations. Hence For instance:

afloat $\models$ ¬ aground ∧¬ afloat $\not\models$ aground
aground $\models$ ¬ afloat ∧¬ aground $\not\models$ afloat
sunken $\models$ ¬ afloat ∧¬ afloat $\not\models$ sunken
afloat $\models$ ¬ sunken ∧¬ sunken $\not\models$ afloat

Morpho-derivational semantics. Adjectives in Class 1 can be related to both nouns and verbs. This is encoded in the following axiom schemas:

MDR 1. Adj1 $\sqsubset$ ¬ Adj2 If Adj1 = Anto(Adj2)
e.g., afbat $\sqsubset$ ¬ sunken

MDR 2. Adj1 $\equiv$ ∃ Theme^{-1}.V1 If Adj1 is related to V1
e.g.,afbat $\equiv$ ∃ Theme $^{-1}$.fbat

MDR 3. V1 $\sqsubset$ ¬ V2 If V1 = Anto(V2)
e.g., fbat $\sqsubset$ ¬ sink

MDR 4. N1 $\equiv$ V1 If Adj1 is related to an evt denoting N1
e.g., fbating $\equiv$ fbat

MDR 5. N1 $\sqsubset$ ¬ N2 If N1 is an antonym of N2
e.g., fbating $\sqsubset$ ¬ sinking

MDR 6. N11 $\equiv$ ∃ Theme^{-1}.V1 If Adj1 is related to a noun N11 denoting the theme role of the verb V1
e.g., fbater $\equiv$ ∃ Theme $^{-1}$.fbat

We make the following assumptions about the syntax/semantic interface that is, about the semantic representations associated with given sentence patterns.

SCR 1. NP toBe Adj
$ADJ \sqcap NP$

SCR 2. NP toBe clearly Adj
$ADJ \sqcap NP$

SCR 3. N_i[+event] of NP is clear
$V_i \sqcap \exists theme.NP$

SCR 4. N_{ii}[-event] is clear
$\exists theme^{-1}.V_i$

SCR 5. NP toBe V[+ing].
$V \sqcap \exists Theme.NP$

Given the above axiom schemas and semantic constructions rules, the following inference patterns can be handled:

1. ADJ1 + N $\models$ N
   Ex. This boat is afloat. $\models$ This is a boat.

2. ADJ1 + N $\models$ ADJ1
   Ex. This boat is afloat. $\models$ This is afloat.

3. ADJ1 + N $\not\models$ ¬ N
   Ex. The boat is afloat. $\not\models$ This not a boat.

4. ADJ1 + N $\models$ ¬ ADJ2 $\sqcap$ N
   Ex. The boat is afloat. $\models$ The boat is not sunken.

5. ¬ ADJ1 + N $\not\models$ ADJ2 $\sqcap$ N
   Ex. The boat is not afloat. $\not\models$ The boat is sunken.

6. ADJ1 + N $\models$ N $\sqcap\exists$theme^{-1}.$V1$
   Ex. The boat is afloat. $\models$ The boat is the floater.

7. ADJ1 + N $\models$ V1 $\sqcap\exists$theme.N
   Ex. The boat is afloat. $\models$ The boat is floating.

8. ADJ1 + N $\models$ N1 $\sqcap\exists$theme.N
   Ex. This boat is clearly afloat. $\models$ The floating of the boat is clear.

9. ADJ1 + N $\models$ N $\sqcap\exists$theme^{-1}.N1
   Ex. This boat is clearly afloat. $\models$ The floating of the boat is clear (or the boat is the floating object).

10. ¬ (ADJ1 + N) $\models$ ¬ (V1 $\sqcap\exists$theme.N) $\not\models$ ¬ N
    Ex. This is not a floating boat. $\not\models$ This is not a boat.

11. ¬ (ADJ1 + N) $\not\models$ ¬ Adj1
    Ex. This is not a floating boat. $\not\models$ This is not afloat.

12. ¬ (ADJ1 + N) $\not\models$ ¬ V1
    Ex. This is not a floating boat. $\not\models$ This is not floating.

13. ¬ (ADJ1 + N) $\not\models$ ¬ N1
    Ex. This is not a floating boat. $\not\models$ This is not a floating.

14. ¬ (ADJ1 + N) $\not\models$ ¬ ∃ theme^{-1}.V1
    Ex. This is not a floating boat. $\not\models$ This is not the floater.

15. ¬ (ADJ1 + N) $\not\models$ ¬ ∃ theme.N
    Ex. This is not a floating boat. $\not\models$ This is not a floating.

In the inference patterns 10 to 15, the negation of the adjective-noun compound ¬ (ADJ1 + N) is syntactically blocked, as the adjectives in this class are used predicative only, however the equivalent representation V1 $\sqcap\exists$theme.N can be used to motivate the inferences.

The following show in more detail how the first three of the above (non) entailments are recognised.

Example 1.

(4) a. The boat is afloat.

b. $\models$ The boat is floating.

4a	$\equiv$ Boat $\sqcap$ Afbat	(by SCR 1)	A
4b	$\equiv$ Float $\sqcap \exists Theme$.Boat	(by SCR 5)	B
Afbat	$\equiv \exists Theme^{-1}.Float$	(by MDR 2)	C
1	$\equiv$ Boat $\sqcap \exists Theme^{-1}.Float$	(from A and C)	D
	D $\models$ B	(By Defn 1)	E

Example 2.

(5) a. The boat is afloat.

b. $\models$ The boat is the floater.

5a	$\equiv$ Boat $\sqcap$ Afbat	(by SCR 1)	A
5b	$\equiv$ Boat $\sqcap \exists Theme^{-1}.float$	(by SCR 4)	B
Afbat	$\equiv \exists Theme^{-1}.Float$	(by MDR 2)	C
	A $\models$ B	(from B und C)	D

Example 3.

(6) a. The boat is afloat.

b. $\models$ The boat is not sinking.

6a	$\equiv$ Boat $\sqcap$ Afbat	(by SCR 1)	A
6b	$\equiv \neg$ sink $\sqcap \exists Theme.boat$	(by SCR 5)	B
Afbat	$\equiv \exists Theme^{-1}.Float$	(by MDR 2)	C
	Boat $\sqcap \exists Theme^{-1}.Float$	(from A and C)	D
	fbat $\sqcap \exists Theme.boat$	(By Defn 1)	E
	E $\models$ B	(by MDR 1)	F

3.2.2 Class 8.

Class 8 contains adjectives like *big,fast,tall,deep* which can be used attributively and predicatively, are gradable, can be modified by *very*. Semantically, they are classified as subsective adjectives and their antonyms are contraries. They are morphologically related to nouns which describe the particular property denoted by the adjectives and to nouns of which they are attributes.

Model theoretic semantics. Adjectives of class 8 are subsective adjective. They will thus licence the correponding inference patterns namely:

$$A + N \not\models A \quad (4)$$
$$A + N \models N \quad (5)$$

Lexical semantics. The Adjectives of class 8 enter in contrary opposition relations. Hence, the following axioms schemas will be licensed:

$$A_i \models \neg Anto(A_i) \quad and \quad \neg A_i \not\models Anto(A_i) \quad (6)$$

For instance:

long $\models$ ¬ small ∧¬ long $\not\models$ small
deep $\models$ ¬ shallow ∧¬ deep $\not\models$ shallow

Morpho-derivational semantics. Adjectives in Class 8 can be related to nouns but not to verbs. This is encoded in the following axiom schemas:

MDR 1. Adj1 $\sqsubset$ ¬ Adj2 If Adj1 = Anto(Adj2)
Ex. tall $\sqsubset$ ¬ short

MDR 2. Adj1 $\sqsubset$ ∃ has_property.(N1 $\sqcap$∃has_measure.Top)
If Adj1 is related to a noun N1 denoting the property described by Adj1
Ex. tall $\sqsubset$ ∃ has_property.(tallness $\sqcap$∃has_measure.Top)

MDR 3. N1 $\sqsubset$ ¬ N2 If N1=Anto(N2)
Ex. tallness $\sqsubset$ ¬ shortness

MDR 4. N1 ≡ N' $\sqcap$∃has_value.Adj1 $\sqcap$∃has_measure.Top
If Adj1 is an attribute of the noun N'
Ex. tallness ≡ height $\sqcap$∃has_value.tall ∃has_measure.Top

MDR 5. N2 ≡ N' $\sqcap$∃has_value.Adj2 $\sqcap$∃has_measure.Top
If Adj2 is an attribute of the noun N'
Ex. shortness ≡ height $\sqcap$∃has_value.short ∃has_measure.Top

MDR 6. N1 $\sqsubset$ N' If N1 is an hyponym of N'
Ex. tallness $\sqsubset$ height

MDR 7. N2 $\sqsubset$ N' If N2 is an hyponym of N'
Ex. shortness $\sqsubset$ height

MDR 8. Adj11 $\sqsubset$ Adj1 If Adj1 is a scalar attribute with value less then Adj11 (hyponymy is not defi ned for adjectives)
Ex. giant $\sqsubset$ tall

We make the following assumptions about the semantic representations associated with basic sentence patterns.

SCR 1. NP toBe Adj
NP $\sqcap$∃ has_property.(N1 $\sqcap$∃has_measure.NP)

SCR 2. That toBe Det Adj NP
NP $\sqcap$∃ has_property.(N1 $\sqcap$∃has_measure.NP)

SCR 3. NP toBe clearly Adj
NP $\sqcap$∃ has_property.(N1 $\sqcap$∃has_measure.NP)

SCR 4. N1 of NP is clear
NP $\sqcap$∃ has_property.(N1 $\sqcap$∃has_measure.NP)

SCR 5. The Adj N' of NP
NP $\sqcap$∃ has_property.(N' $\sqcap$∃ has_value.Adj $\sqcap$∃has_measure.NP)

SCR 6. NP1 toBe Adj as a N
NP1 $\sqcap$ N $\sqcap\exists$has_property.(N' $\sqcap\exists$ value.Adj $\sqcap\exists$ has_measure.N)

SCR 7. NP1 toBe NP2[+measure] Adj
NP1 $\sqcap\exists$has_property.(N' $\sqcap\exists$ value.Adj $\sqcap\exists$ has_measure.NP2)

SCR 8. NP1 toBe NP2[+measure] Adj N
NP1 $\sqcap$ N $\sqcap\exists$has_property.(N' $\sqcap\exists$has_value.Adj $\sqcap\exists$ has_measure.NP2)

Given the above axioms, the following inference patterns can be handled:

1. ADJ1 + N $\models$ N
   This animal is tall. $\models$ This is an animal.

2. ADJ1 + N $\not\models$ ADJ1
   This animal is tall. $\not\models$ This is tall.

3. ADJ1 + N $\not\models$ $\neg$ N
   This animal is tall. $\not\models$ This not an animal.

4. ADJ1 + N $\models$ $\neg$ ADJ2 $\sqcap$ N
   This animal is tall. $\models$ This animal is not small.

5. $\neg$ ADJ1 + N $\not\models$ ADJ2 $\sqcap$ N
   This animal is not tall. $\not\models$ This animal is small.

6. ADJ1 + N $\models$ N $\sqcap\exists$has_property.(N1 $\sqcap\exists$has_measure.N)
   This animal is tall. $\models$ This animal has tallness.

7. ADJ1 + N $\not\models$ $\exists$has_property.(N1 $\sqcap\exists$has_measure.Top)
   This animal is tall. $\models$ tallness.

8. ADJ1 + N $\models$ N $\sqcap\exists$has_property.(N' $\sqcap\exists$ has_value.ADJ1 $\sqcap\exists$has_measure.N)
   This animal is tall. $\models$ This animal has a tall stature.

9. ADJ1 + N $\not\models$ $\exists$has_property.(N' $\sqcap\exists$ has_value.ADJ1 $\sqcap\exists$has_measure.Top)
   This animal is tall. $\not\models$ tall stature.

10. $\neg$ (ADJ1 + N) $\not\models$ $\neg$ N
    This is not a tall animal. $\not\models$ This is not an animal.

11. $\neg$ (ADJ1 + N) $\not\models$ $\neg$ ADJ1
    This is not a tall animal. $\not\models$ This is not tall.

12. $\neg$ (ADJ1 + N) $\not\models$ $\neg$ $\exists$ has_property.(N1 $\sqcap\exists$has_measure.N)
    This is not a tall animal. $\not\models$ This animal has not tallness.

13. $\neg$ (ADJ1 + N) $\not\models$ $\neg$ $\exists$ has_property.(N' $\sqcap\exists$ has_value.Adj1 $\sqcap\exists$has_measure.N)
    This is not a tall animal. $\not\models$ This animal has not a tall stature.

Example 1.

(7) (a) John is a 2 meter tall man. $\models$ (b) John is 2 meter tall.

7a	$\equiv$ John $\sqcap$ Man $\sqcap\exists$has_property.(height $\sqcap$has_value.tall $\sqcap$has_measure(2 meter)) (by SCR 8)	A
7b	$\models$ John $\sqcap\exists$has_property.(height $\sqcap$has_value.tall $\sqcap$has_measure(2 meter)) (by SCR 7 and from A)	B
	A $\models$ B	C

(8) (a) John is a 2 meter tall man. $\not\models$ (b) John is a tall man.

8a	$\equiv$ John $\sqcap$ Man $\sqcap\exists$has_property.(height $\sqcap$has_value.tall $\sqcap$has_measure(2 meter)) (by SCR 8)	A
8b	$\models$ John $\sqcap$ Man $\sqcap\exists$has_property.(height $\sqcap$ has_value.tall $\sqcap$has_measure(man)) (by SCR1 and from A)	B
	A $\not\models$ B	C

4 Implementation

For each of the 15 classes, we have specified a set of axioms schemas, some basic semantic construction rules and a set of inference patterns which could be deduced to follow from both of these. The axioms schemas were implemented in Description Logic using RACER and for each inference pattern identified, the corresponding Description Logic query was checked to verify that the proposed axioms and semantic construction rules did indeed correctly predict the deduced inference patterns.

5 Further work and evaluation

The main contribution of this work is a detailed analysis of the interactions between derivational morphology, lexical and compositional semantics and of their impact on the entailment patterns licensed by sentences containing adjective or their related nouns/verbs.

To turn this analysis into a computational system, its components need to be integrated into a

```
<pair id="1" value="TRUE" class="[CLASS1]" inference="Adj/Verb">
        <t>The boat is <sn n="1"> afloat </sn>.</t>
        <h>The boat is floating.</h>
</pair>
<pair id="2" value="FALSE" class="[CLASS6]" inference="Antonymy">
        <t>This is not a  <sn n="1"> rectangular </sn> table.</t>
        <h>This is a  <sn n="1"> round </sn> table </h>
</pair>
<pair id="3" value="TRUE" class="[CLASS8]" inference="Adj/Noun">
        <t>The line is 2 meter  <sn n="1"> long </sn>.</t>
        <h>The length of the line is 2 meter.</h>
</pair>
<pair id="4" value="FALSE" class "[subs/intersective]" inference="Attr/Pred">
        <t>The treasurer is  <sn n="2"> present </sn>.</t>
        <h>This is the  <sn n="1"> present </sn> treasurer.</h>
</pair>
```

Figure 2: TestSuite

semantic analyser and the behaviour of that analyser tested against a collection of data. We are currently working on developing such an analyser within a symbolic grammar framework. We have also started to develop an evaluation test suite geared towards entailment recognition between sentence pairs containing adjectives. At the moment, the test suite contains about 1 000 inference pairs in xml format. Each item in the TestSuite (see fig. 2) is annotated with a judgement about the truth of the entailment between the pair of sentences, with the type of inference involved and with the specification of adjective involved. Moreover, each adjective is annotated with the WordNet sense corresponding to the given class.

The idea behind this test suite is similar to that underlying the creation of the TSNLP (Test suite for natural language processing) or the Eurotra testsuites namely, to provide a benchmark against which to evaluate and compare existing semantic analyzers. Thus this test suite illustrates the semantic and syntactic behaviour of adjectives and their related verbs/nouns with respect to textual entailment. One could imagine other test suites illustrating the semantic behaviour of verbs, of quantifiers, of discourse connectives, etc. Just as the TSNLP still proves useful in supporting the development of new symbolic parsers/grammars, hand built test suites of artificial examples might prove useful in improving the accuracy of semantic analyser wrt textual entailment. Indeed the Pascal RTE challenge has shown that existing systems fares rather poortly at the textual entailment task. Providing a set of hand crafted semantic test suites might help in remedying this shortcoming.

Beside implementing and evaluating the analysis of adjectives presented in this paper, we are also working on refining this analysis by combining it with a detailed analysis of noun semantics so as to handle (non) entailments such as:

Lyon is the gastronomical capital of France
$\not\models$ *Lyon is the capital of France*

References

Claire Gardent and Evelyne Jacquey. 2003. Lexical reasoning. In *Proceedings of the ICON'03 (International Conference on Natural Language Processing)*, Mysore, India.

Hans Kamp and Barbara Partee. 1995. Prototype theory and compositionality. *Cognition*, (57):129–191.

Hans Kamp. 1975. Two theories about adjectives. In Edward L. Keenan (ed.), *Formal Semantics of Natural Language*, pages 123–155. Cambridge University Press.

R. Quirk, S. Greenbaum, G. Leech, and J. Svartvik. 1985. *A Comprehensive Grammar of the English Language*. Longman.

2005. *Proceedings of the First Challenge Workshop Recognizing Textual Entailment*, Southampton, UK.

Ralf Möller Volker Haarslev. 2001. Description of the racer system and its applications. In *Proceedings International Workshop on Description Logics (DL-2001*, Stanford, USA.