Anaphora Resolution Involving Interactive Knowledge Acquisition

Rolf Schwitter

Centre for Language Technology
Macquarie University
Sydney 2109 NSW, Australia
Rolf.Schwitter@mq.edu.au

Abstract. Anaphora resolution in current computer-processable controlled natural languages relies mainly on syntactic information, accessibility constraints and the distance of the anaphoric expression to its antecedent. This design decision has the advantage that a text can be processed automatically without any additional ontological knowledge, but it has the disadvantage that the author is severely restricted in using anaphoric expressions while writing a text. I will argue that we can allow for a wider range of anaphoric expressions whose resolution requires inference-supporting knowledge, if we consider the anaphora resolution process as an interactive knowledge acquisition process in those cases where no suitable noun phrase antecedent can be found. In particular, I will focus on definite descriptions that stand in a synonymy, subclass/superclass or part-whole relation to their noun phrase antecedent, and show how the original anaphora resolution algorithm of PENG Light can be extended in a systematic way in order to take care of these bridging definite descriptions. The solution to this problem also sheds some light on the adequate treatment of part-whole relations in a controlled natural language context.

1 Introduction

Computer-processable controlled natural languages such as Attempto Controlled English [12] and PENG Light [34] are **engineered** subsets of natural languages designed to get rid of ambiguity and vagueness that is inherent in full natural language. These controlled natural languages **look like** natural languages such as English but are in fact **formal** languages that can be translated unambiguously into the input language of an automated reasoner and used for several reasoning tasks, among them question answering. Similar to unrestricted natural language, these controlled natural languages allow for anaphoric expressions but their form and usage are considerably restricted. An anaphoric expression (= anaphor) is a word or a phrase that points back to an expression (= antecedent) that has been previously introduced in the text (see [19] for an overview). The two most important types of anaphoric relations that are used in sentences and between sentences in controlled natural languages are pronominal anaphora and definite

N.E. Fuchs (Ed.): CNL 2009 Workshop, LNAI 5972, pp. 36–55, 2010.

noun phrase anaphora. Definite noun phrase anaphora take the form of definite descriptions and proper names. Computer-processable controlled natural languages use relatively "safe" anaphora resolution algorithms that rely mainly on syntactic information, accessibility constraints and the distance of the anaphor to its antecedent [12,29]. This makes it easy for the machine to resolve anaphoric expressions automatically but difficult for the human author to remember the approved forms of anaphoric expressions. In the following discussion, I will focus on definite descriptions, in particular on bridging definite descriptions [21], and discuss how the resolution of these anaphoric expressions that requires inference-supporting knowledge can be managed in a controlled natural language context where not all relevant knowledge is available in advance.

The reminder of this paper is organised as follows: In Section 2, I distinguish different usages of definite descriptions that a controlled natural language processor needs to be able to handle in order to process a text and briefly discuss some related research and contemporary theories. In Section 3, I look at a number of important semantic relations that build the foundation for linking bridging definite descriptions to their noun phrase antecedents via inference. In Section 4, I investigate how well an existing linguistic resource (WordNet) supports the resolution of definite descriptions and discuss alternative approaches to construct the relevant background knowledge for anaphora resolution. In Section 5, I show how a first-order model builder can be used for constructing a model that contains the required background knowledge to resolve definite descriptions during the writing process. In Section 6, I discuss how the language processor of the controlled natural language PENG Light communicates with the model builder and how the author can interact with the language processor in order to augment the text with additional background knowledge that is required to resolve anaphoric definite descriptions. In Section 7, I summarise the advantages and benefits of using an interactive approach to anaphora resolution in a controlled natural language context.

2 Definite Descriptions

Definite descriptions are noun phrases of the form *the F* that begin with the definite determiner *the* followed by the remaining noun phrase *F*. This definition captures one widespread use of the term. However, there are many kinds of expressions that have this surface form in unrestricted natural language (for example generics and plurals) but they have semantic properties that are clearly different from definite descriptions, and there are many expressions that have different surface forms (for example possessives) that could count as being definite descriptions [17]. Definite descriptions have played a central role in logic, semantics, and philosophy of language and their (controversial) discussion goes back to Frege [11], Russell [26], Strawson [30], and Donnellan [9].

2.1 Definite Descriptions and Anaphora Resolution

In context of PENG Light [34], we need a flexible approach to process definite descriptions that pays attention to the various ways individuals can be identified. This approach must leave room for information expressed by definite descriptions to be linked directly to previous information, to be linked indirectly via inference to previous information or to be accommodated as discourse-new information during the interpretation process.

In the simplest case, a definite description that is used anaphorically matches syntactically fully or partially with its noun phrase antecedent and constitutes an example of a direct anaphor. For instance, the head of the noun phrase antecedent in (1) directly matches with the head of the anaphor in (2), and it is relatively easy to establish a direct link between the two expressions:

1. <u>An academic</u> who teaches COMP448 in E6A owns a computer.
2. *The academic* supervises Robert Black and leads the LT Center.

However, the relation between the anaphor and its noun phrase antecedent is often more complex than that of identity. The relation may be a synonymy relation as in (3), a subclass/superclass relation as in (4), or a part-whole relation as in (5):

3. *The scholar* supervises Robert Black ...
4. *The educator* supervises Robert Black ...
5. *The mother board* is faulty and the computer does not start up.

These definite descriptions point back to a noun phrase antecedent that has already been introduced in (1), but they are characterised by a different head noun. The resolution of these definite descriptions requires additional ontological knowledge and some specific kind of inference (also known as implicatures). Note that the definite noun phrases in (3) and (4) refer to the same individual as the noun phrase antecedent in (1). But this is not the case in (5) where the discourse referent of *the mother board* is only "associated" with the one of *a computer* previously introduced in (1). Definite descriptions that have a noun phrase antecedent which uses a different head noun and are related to the antecedent by a relation other than identity are called bridging descriptions.[1]

For some semantic relations it is less obvious that they can establish a link between a definite description and a noun phrase antecedent than for others. While a bridging definite description like (4) sounds felicitous in the context of (1), the following one sounds odd:

6. ?*The professor* supervises Robert Black ...

The potential noun phrase antecedent in (1) and the definite description in (6) stand in a superclass/subclass relation while the antecedent in (1) and the

[1] Clark [8] also considers direct anaphora as "bridging", but the implicature required for resolving these direct references is straightforward.

definite description in (4) stand in a subclass/superclass relation. A possible explanation for the infelicity of (6) is that a subsequent definite description can **in general** not introduce new information about an existing referent (although unrestricted natural language offers of course other ways to do this), so using a more general expression to establish a bridge as in (4) is fine, but using a more specific one as in (6) is unusual and would require additional contextual support. In other words, the definite description in (6) has more descriptive content than the one in (4), and it this fact that makes accommodation [16] more likely than bridging. That means if no suitable noun phrase antecedent can be found for a definite description, we need to jump into repair mode and accommodate the use of the definite description in a similar way as if this expression had been introduced by an indefinite noun phrase. This process will result in a new discourse referent that we can anaphorically refer to by subsequent noun phrases provided that we take a number of accessibility constraints into consideration (for details see Section 6).

2.2 Related Research and Theories

Hobbs et al. [13] use weighted abduction to interpret discourse and to resolve anaphoric references. Abduction is inference to the best explanation and explains the data at least cost from a knowledge base. Bos et al. [6] develop a theory of bridging for unrestricted natural language in the context of discourse representation theory [14] by extending van der Sandt's theory of presupposition projection [27] with lexical knowledge that is available in form of a rich generative lexicon [24]. Asher and Lascarides [1] show that this theory has shortcomings to model bridging inferences in the absence of presupposition triggers and that lexical semantics is not enough for modeling bridging. They present a sophisticated theory that demonstrates how rhetorical information interacts with compositional and lexical semantics during the update of segmented discourse representation structures and see bridging inferences as a byproduct of this discourse update process. Baumgartner and Kühn [3] develop an abductive solution to resolve anaphoric references in a model generating framework that is in theory able to deal with several discourse histories but the answer to the question how to select between alternative histories is left open.

It is important to note that all these theories assume unrestricted natural language as object of research. The authors gloss over the problem of parsing natural language and take the availability of the relevant (background) knowledge sources for granted.

3 Tailoring Semantic Relations

The resolving of bridging references requires domain-specific terminological background knowledge since the noun phrase antecedent and the anaphor can stand in a number of different semantic relations (see [8] for an overview). I will focus here on three important types of semantic relations: synonymy, subclass and part-whole relations. Both subclass and part-whole relations are inclusion relations that are transitive and structure the semantic space in a hierarchical fashion. While transitivity of the subclass relation supports valid syllogistic inferences, this does not always seem to be the case for part-whole relations and requires a closer investigation. Before we do this, let us first have a look at how synonymy relations are handled in our controlled natural language context and then progress to the thornier issues that inclusion relations present.

3.1 Synonymy Relations

In unrestricted natural language, synonyms are different words or phrases with identical meaning (= strict synonyms) or very similar meaning (= near-synonyms). In PENG Light (see [34] for an introduction), we only allow for strict synonyms where a word or phrase has exactly the same meaning as another expression. Synonymy relations are binary relations that are transitive, symmetric and reflexive. Noun phrases that use these synonyms have always the same discourse referent. In PENG Light, the author can specify a synonymy relation via a definition that asserts the identity between two expressions, for example:

7. Every scholar is defined as an academic.

It is the author who decides whether two expressions can stand in a synonymy relation or not, and there is nothing that can hinder the author to define, for example, *wumpus* as a synonym of *academic* – but of course this is bad style, and it should be avoided. Another problem that we face is that more than one candidate antecedent might occur in the text – as illustrated in (8) and (9) – that can serve as an antecedent for a definite description:

8. <u>An academic A</u> leads the LT Centre.
9. <u>An academic B</u> teaches COMP249.
10. *The scholar* supervises Robert Black ...

I will discuss a solution to this problem in Section 6 where I show how the ordering of the sentences affects the anaphora resolution process in PENG Light.

3.2 Subclass Relations

A subclass relation (or class inclusion) is often expressed in unrestricted natural language in the form: *An A is a B* or *An A is a kind of B* whereas *A* is referred to as the specific entity type and *B* the generic entity type. In linguistics, *A* is called a hyponym and *B* a hypernym. A hyponym is a word whose semantic range is included within that of another word, its hypernym. In PENG Light, we distinguish between subclass relations such as in (11) and (12), and class membership such as in (13):

11. Every professor is an academic.
12. Every academic is an educator.
13. Anna Grau is a professor.

Note that in contrast to class inclusion in (11) and (12), the class membership relation in (13) relates an individual to an entity type. In the case of class inclusion, we can make the underlying universal quantifier explicit on the surface level of the controlled natural language and use for this purpose the form *Every A is a B* instead of the more implicit version *An A is a B*.

Class inclusion relations are usually understood to express strict partial ordering relations. Strict partial ordering relations are transitive, irreflexive, and antisymmetrical. We can express these semantic properties directly in PENG Light via two conditional statements:

14. If X is a subclass of Y and Y is a subclass of Z
    then X is a subclass of Z.
15. If X is a subclass of Y then X is not equal to Y.

3.3 Part-Whole Relations

A part-whole relation (or meronymic relation) has similar semantic properties as the class inclusion relation: it is a transitive, hierarchical, inclusion relation and can provide structure to the terminology. In English, the term *part of* is the most general of a large number of terms that can be used to express various kinds of meronymic relations [35]. The vagueness and generality of this term glosses over more specific meronymic relations and creates problems with transitivity and, as a consequence, makes it difficult to resolve bridging definite descriptions. Here is an example (in unrestricted natural language) that illustrates this problem:

16. A mother board is part of a computer.
17. A ship is part of a fleet.

In (16), the *part of* relation describes a relation between an object (*computer*) and one of its components (*mother board*). This component stands in a specific functional relation to the whole. This is in contrast to (17) where the meronymic relation describes a relation between a member (*ship*) and a collection (*fleet*). Membership in a collection is determined by spatial proximity (or a social relation) between the whole and its parts but does not show a functional relation to the whole. On the other side, membership in a collection and membership in a class (as discussed in Section 3.2) differ in that class membership is based on the similarity of its elements. We can clearly distinguish between (16) and (17) on the surface level of the controlled natural language by replacing the term *part of* in both cases by a more specific term (additionally, we can make the universal quantifier in the subject position of the statement explicit, in a similar way as we have done this for subclass relations before; note that *every* and *each* are predefined function words in PENG Light and have the same meaning):

18. Each mother board is a component of a computer.
19. Each ship is a member of a fleet.

Meronymic relations appear to be transitive but transitivity often seems to fail in "meronymic syllogisms" if the *part of* relation is used in an incompatible way in unrestricted natural language; compare (20) with (21):

20. A CPU is part of a mother board.
    A mother board is part of a computer.
    A CPU is part of a computer.

21. An engine is part of a ship.
    A ship is part of a fleet.
    *An engine is part of a fleet.

In (20), we can observe that the *part of* relation describes a semantic relation between a component and an object. But this is not the case in (21) where two different types of meronymic relations are used: the first one describes a relation between a component and an integral object and the second one a relation between a member and a collection. As a result of this mix, the part-whole relation is not transitive and the inference is not valid. We can make the cause of this problem transparent on the surface level of the controlled natural language by replacing the general term *part of* by the two more specific expressions *component of* and *member of* that clearly distinguish between the intended interpretations:

22. Each CPU is a component of a mother board.
    Each mother board is a component of a computer.
    Each CPU is a component of a computer.

23. Each engine is a component of a ship.
    Each ship is a member of a fleet.
    *Each engine is a (component OR member) of a fleet.

In [7,35], Chaffin and his colleagues present a taxonomy that distinguishes seven types of part-whole relations: 1. component-integral object (*pedal – bike*), 2. feature-event (*trapeze act – circus*), 3. member-collection (*ship – fleet*), 4. portion-mass (*slice – pie*), 5. phase-activity (*paying – shopping*), 6. place-area (*Everglades – Florida*), and 7. stuff-object (*steel – car*). These part-whole relations can be described via four relation elements which characterise the nature of the connection between the part and whole: (a) whether the relation of a part to the whole is functional or not; (b) whether the part is separable from the whole or not; (c) whether the parts are similar to each other and to the whole or not, and (d) whether the whole possesses (most of) its parts at the same time. Furthermore, the authors show that this taxonomy can explain quite accurately the failure of transitivity in "part-whole syllogisms" and that humans **can distinguish** between these semantic relations if they need to do so.

We have taken the semantic relations (1-4, 6 and 7) of this meronymic taxonomy as a **starting point** for the treatment of part-whole relations in PENG Light. The *phase-activity* relation (5) is currently not used since PENG Light does not allow to anaphorically refer back to a verbal event. For each other type in the taxonomy, we use a default term in controlled language to clarify the semantic relation so that we can better support the resolution of bridging definite descriptions. As a consequence, the term *part of* is not allowed in PENG Light and needs to be replaced always by a more specific term. The following examples (24-29) list these default terms which overlap with a set of expressions that have been frequently used in an empirical relation naming experiment [7]:

24. **Component-Integral Object**
    ENG: A CPU is part of a computer.
    CNL: Each CPU is **a component of** a computer.

25. **Feature-Event**
    ENG: A self-timer is a feature of a digital camera.
    CNL: Each self-timer is **a feature of** a digital camera.

26. **Member-Collection**
    ENG: A computer is part of a domain.
    CNL: Each computer is **a member of** a domain.

27. **Portion-Mass**
    ENG: A buffer is part of a memory.
    CNL: Each buffer is **a portion of** a memory.

28. **Area-Place**
    ENG: A partition is part of a hard disk.
    CNL: Each partition is **an area of** a hard disk.

29. **Stuff-Object**
    ENG: Aluminum is part of a hard disk.
    CNL: Each hard disk is **made of** aluminum.

These default terms can be related to a number of user-defined synonyms in PENG Light. That means each synonym set uniquely identifies a type in the above-mentioned meronymic taxonomy.

4 Knowledge Sources for Anaphora Resolution

WordNet [10] has been used as an approximation of a knowledge base for resolving bridging definite descriptions in unrestricted texts [21,22,23,32]. WordNet groups English words into sets of synonyms and specifies various semantic relations between these synonym sets. These semantic relations depend on the type of the word and include for nouns – among other things – hyponymy/hypernymy and holonymy/meronymy relations. WordNet distinguishes only three subtypes of meronymic relations: component part (*part of*), member-collection (*member of*) and substance (*substance of*). In WordNet, we can, for example, find the information that *computer* is a part of *platform*, that *professor* is a member of *staff* and that *oxygen* is a substance of *air*. That means we can only find information about semantic relations of nouns in WordNet that belong to three of those six meronymic types that are currently available in PENG Light.

WordNet proved to be not a very reliable resource for the automatic identification of correct semantic relations (synonymy, hyponymy, and meronymy) and consequently for the resolution of anaphoric references in unrestricted texts. In almost 40% of the cases, no semantic relation could be found that relates a definite description to its antecedent [21,32]. The situation gets even worse if we work in an application domain where a very specific vocabulary is required and where the semantic relations between the entries of the vocabulary need to be established first.

Alternatively, we can try to construct a (linguistically motivated) formal ontology for a particular application domain that contains the required terminological knowledge for resolving definite descriptions. Note that this terminological knowledge can be specified directly in PENG Light and then be translated automatically into an expressive description logic [2]. In this scenario, the knowledge base can immediately be used for resolving anaphoric definite descriptions by checking if a discourse referent already exists in the knowledge base and if this discourse referent stands in an approved semantic relation to the information expressed by the definite noun phrase. The information in the knowledge base can even be used to guide the writing process in a predictive way (using similar techniques as in [15,28]) since all background information has been carefully specified in advance.

However, there exists another scenario where a domain expert might want to assert new factual information, but the terminological knowledge is **not yet** available. For example, the domain expert might want to assert first the sentence (1) (see Section 2.1) and then the sentence (4), but the correct resolution of the definite description in (4) would require additional terminological information that probably does not yet exist. This suggests an approach where the domain expert supports the anaphora resolution process and specifies additional terminological knowledge while a text is written. In the next section, I look into the suitability of a state-of-the-art first-order model builder for providing the required background knowledge to support such an interactive approach, in particular, because I am interested in using the expressivity of full first-order logic for knowledge representation rather than a version of description logic.

5 Model Generation

Model generation was introduced as a new reasoning paradigm in the mid-1980's and is able to handle a wider range of specifications than the traditional refutation-based paradigm [18]. Surprisingly, model generation received only little attention within computational semantics and the broader field of language technology although automatic model builders have very attractive properties: they provide a positive way to deal with the satisfiability problem and are able to construct concrete flat models – if there is one – for first-order theories (see [6,25] for an introduction).

5.1 E-KRHyper

PENG Light currently uses E-KRHyper [20] as a reasoning service. E-KRHyper is a general purpose theorem prover and model generator for first-order logic with equality and has been designed for the use in knowledge representation applications (supporting arithmetic evaluation and non-monotonic negation) [4].

E-KRHyper can be seen as an extension of a tableaux-based theorem prover. Tableaux-based provers can be used in two different ways: on the one hand, they can be used for proving that some goal follows from a set of axioms by adding the negation of the goal to the axioms and showing that the resulting set of formulas has no models; on the other hand, they can also be used for model generation, by adding the goal itself to the axioms and showing that the resulting tableau has an open branch which forms a model consisting of a set of ground formulas.

E-KRHyper accepts a first-order logic specification in clausal form (Protein format). That means the output of PENG Light – in our case a discourses representation structure – needs to be translated first into a first-order formula and then with the help of the TPTP tools [31] into the Protein format before the theory can be processed by E-KRHyper. E-KRHyper then computes a E-hyper tableau working always on a single branch. If the computation of a branch reaches a fixed point, then a model has been found. The Herbrand model that E-KRHyper outputs consists of all ground instances of the derived formulas. Since the derived formulas must not necessarily be ground, they characterise in some cases an infinite model (see [4,20,33] for details).

5.2 Model Building for Text Understanding

Models can make correct predictions about the accessibility of bridging definite descriptions provided the information derived from the text is represented in a suitable way. Let's illustrate this claim and start the discussion with a case where the semantic relation between the definite description and the noun phrase antecedent is a synonymy relation. The first sentence in the following mini-discourse (30) asserts terminological information, the second sentence specifies information about a particular application domain, and finally the subsequent sentence starts with a definite description:

30. Every scholar is defined as an academic.
    An academic teaches COMP448 in E6A.
    The scholar ...

In principle, the model builder can take a first-order representation of the first two sentences as input and generate a model that contains the following variable-free atomic formulas:

31. `academic(sk1). scholar(sk1).`
    `teaching(sk2, sk1, comp448). location(sk2, e6a).`

This model can be accessed by an anaphora resolution algorithm and be searched for an antecedent that stands in a synonymy relation to the formula `scholar(X)` that has been derived from the definite description in (30). Note that `sk1` and `sk2` are two Skolem constants: `sk1` stands for an individual and `sk2` stands for a teaching event. The variable substitution `X/sk1` returns information about the existence of a suitable discourse referent and licenses the anaphoric interpretation of the definite description.

However, things get a bit more complex when the noun phrase antecedent and the bridging definite description stand in a subclass/superclass relation:

32. Every academic is an educator.
    An academic teaches COMP448 in E6A.
    The educator ...

If the model builder processes the first two sentences of (32) in a naive way, we will end up with the following model:

33. `academic(sk1). educator(sk1).`
    `teaching(sk2, sk1, comp448). location(sk2, e6a).`

As the result shows, we lose the information that an academic is a kind of educator since the model only contains extensional information. The model (33) does not contain any information about the semantic relation between these two formulas, and it is not possible anymore to distinguish between a synonymy relation and a subclass relation.

PENG Light uses a flat notation that relies on reification and on a small number of predefined predicates. These predicates distinguish between objects, events, states, names, thematic relations and inclusion relations. This flat notation makes it easy to formulate axioms for these predicates. Taking the rules for subclass inclusion in (14 and 15) into consideration, the model builder is now able to generate a model that contains also the relevant intensional information about subclass relations, for example:

34. `object(sk1, academic). object(sk1, educator).`
    `theta(sk3, agent, sk1). theta(sk3, theme, sk2).`
    `event(sk3, teaching). named(sk2, com448).`
    `theta(sk3, location, sk4). named(sk4, e6a).`
    `subclass(academic, educator).`

Given the formula `object(X, educator)` for the definite description in (32), the anaphora resolution algorithm can now search the model for an antecedent and check additionally whether this antecedent stands in a subclass relationship to the anaphor or not. This information can then be used on the interface level to indicate how the definite description has been interpreted by the anaphora resolution algorithm.

In a similar way, we add information about synonymy relations to the model, for example in the case of (31) the atomic formula `synonym(academic,scholar)`. This makes the relation explicit and – as we will see – the checking for synonymy between an anaphor and its antecedent easier.

So far so good –, but this representation is still not optimal for our purpose; in particular, if the model contains more than one potential antecedent. The following mini-discourse (35) contains, for example, two potential antecedents for the definite description:

35. Every academic is an educator.
    An academic A teaches COMP448 in E6A.
    An academic B supervises Mary on Monday.
    The educator ...

The current representation loses the information about the textual ordering of these antecedents (see the two object-predicates in bold face in (36)). The anaphora resolution algorithm could rely on the numbering convention that is used for naming the Skolem constants but this does not provide adequate information about the distance between the candidate antecedents and the anaphor:

36. `object(sk1, academic).` **object(sk1, educator).**
    `theta(sk3, agent, sk1). theta(sk3, theme, sk2).`
    `event(sk3, teaching). named(sk2, com448).`
    `location(sk3, sk4). named(sk4, e6a).`
    `object(sk5, academic).` **object(sk5, educator).**
    `theta(sk7, agent, sk5). theta(sk7, theme, sk6).`
    `event(sk7, supervising). named(sk6, mary).`
    `location(sk7, sk8). named(sk8, e7b).`
    `subclass(academic, educator).`

We solve this problem by adding an additional argument to each formula that has been derived from a content word. This argument contains information about the offset position of the word in the text. This additional information makes it easy for the anaphora resolution algorithm to select the most recent noun phrase antecedent from a number of options (note: we will not display this auxiliary argument anymore in the subsequent examples):

37. `object(sk1, academic, '8').`
    `object(sk1, educator, '8'). ...`

As we have already seen in Section 2.1, superclass/subclass relations between a noun phrase antecedent and a definite description do usually not establish a semantic bridge for anaphora resolution since these definite descriptions introduce new information. Note that the model builder does not generate any information for superclass/subclass relations as long as the preconditions of such terminological statements are not fulfilled. The following example (38) illustrates this case:

38. Every academic is an educator.
    Every professor is an academic.
    An academic teaches COMP448 in E6A.
    The professor ...

Here only the rule for the subclass relation in the first sentence is triggered but not the one for the subclass relation in the second sentence since the third sentence does not contain an instance of a professor. This results in the following model (39):

39. `object(sk1, academic).`
    `theta(sk3, agent, sk1). theta(sk3, theme, sk2).`
    `event(sk3, teaching). named(sk2, comp448).`
    `theta(sk5, location, sk6). named(sk6, e6a).`
    `subclass(academic, educator).`

That means there is no atomic formula for *professor* available in the model that could serve as an antecedent for the definite description, and therefore a new discourse referent needs to be introduced in this case.

As explained in Section 3.3, the *part of* relation is polysemous and therefore not allowed in PENG Light; as a consequence, the author has to select one of the available alternatives. For all these alternatives, axioms exist similar to those in (14 and 15) that specify these meronymic relations as transitive, irreflexive and antisymmetrical. Let us assume that these axioms are available in PENG Light and used together with the following mini-discourse in (40):

40. Each mother board is a component of a computer.
    Anna Grau owns a computer.
    The mother board ...

In this case, the model builder generates the model in (41) for the second sentence:

41. `named(sk1, anna_grau).`
    `theta(sk3, theme, sk1). theta(sk3, theme, sk2).`
    `state(sk3, possessing). object(sk2, computer).`

Note that the rules for the meronymic relation will be only triggered after the definite description in (40) has been processed and after a new discourse referent has been introduced for this definite description. This results in two additional atomic formulas in the model: one that contains the information about the definite description (`object(sk4, mother_board)`) and one that contains the information about the componenthood (`component(mother_board, computer)`).

Now let us have a look at a sentence that contains a disjunction:

42. A politician meets ?an academic or talks to ?a student.
    The scholar ...

The first sentence has two models and both models provide potential discourse referents; however, disjunction should block anaphoric reference. E-KRHyper generates the first model and stores which formulas have been derived from the selected disjunct (e.g.: `selected_disjunct(object(sk2, academic), _)`). This information can then be used to exclude discourse referents that occur under disjunction as candidate antecedents (see Section 6.3).

Note that in the case of a conditional sentence a model is first constructed for the antecedent part taking the existing context into consideration in order to be able to bind anaphoric expressions in the consequent of the conditional.

6 Anaphora Resolution in PENG Light

PENG Light is a computer-processable controlled natural language that can be used for knowledge representation [34]. The language processor of the PENG Light system translates texts incrementally into TPTP notation [31] with the help of discourse representation structures (DRSs) [14]. The grammar of PENG Light is written in a DCG-style notation that is transformed via term expansion (a well-known logic programming technique) into a format that can be processed by a chart parser. Since the processing of the input is incremental, the chart parser can provide lookahead information that informs the author which word forms or which categories of words can follow the current input string [15]. The language processor of PENG Light communicates with the model builder E-KRHyper for various reasoning tasks. The model builder can be used, among other things, for consistency checking whenever a new sentence has been added to the text or for question answering whenever the author wants to inspect an emerging or existing textual specification [29]. The language processor does not only communicate with the model builder on the discourse level but also on the phrasal level whenever the extended anaphora resolution algorithm (see Section 6.3) needs to access and reason with the terminological knowledge.

6.1 DRSs in PENG Light

In PENG Light, DRSs are built up incrementally during the parsing process using difference lists. In our case, a DRS is a term of the form `drs(U,Con)` whereas `U` is a list of discourse referents that stand for the entities introduced in the discourse and `Con` is a list of simple or complex conditions for these discourse referents. Simple conditions consist of predicates or equations while complex conditions are built up recursively from other DRSs with the help of logical connectives. The key idea underlying discourse representation theory [14] is that an anaphor can only refer to a discourse referent that occurs in the current DRS or in a DRS immediately superordinate to the current DRS but not to a discourse referent in an embedded DRS. These restrictions make correct predictions about the accessibility of antecedents to anaphora and make DRSs a convenient intermediate knowledge representation format. DRSs can be translated in linear time into TPTP notation (or into any other first-order notation).

The grammar of PENG Light contains not only feature structures for DRSs but also feature structures for syntactic and pragmatic information. The construction of a DRS always runs in parallel with the construction of a syntax tree and a paraphrase. The paraphrase clarifies how the input has been interpreted by the grammar and shows the author all relevant substitutions and provides additional explanations. Some of the conditions in the DRS are annotated with syntactic information in order to improve their processability by other system components. These conditions have the form `Pred#Anno` whereas `Pred` is a predefined predicate and `Anno` is a list that contains syntactic information. In the case of object-predicates, these annotations consist of syntactic information about person, number and gender of the noun and information about the form of the surface string, for example:

43. `object(X,mother_board,23)#[[third,sg,neut],[mother,board]]`

This information can directly be used by the anaphora resolution algorithm that operates over the DRS and communicates with other components of the system (for example the model builder).

6.2 The Original Anaphora Resolution Algorithm

The original anaphora resolution algorithm resolves an anaphorically used noun phrase with the most recent accessible noun phrase antecedent that matches fully or partially with the anaphor and that agrees in person, number and gender with that anaphor [29]. This algorithm has similar coverage as the anaphora resolution of Attempto Controlled English (ACE) [12] and cannot resolve bridging definite descriptions, since it is syntax-based. However, in contrast to ACE where the anaphora resolution algorithm is activated only after the entire DRS for a text has been constructed, the algorithm of PENG Light is embedded into the grammar and triggered whenever a definite expression has been processed.

Representations for definite expressions are built up in an anaphoric DRS that can be seen as a temporary store within the main DRS. This anaphoric DRS consists of unresolved discourse referents (in our case variables) and conditions that contain these discourse referents. Once the processing of a definite expression is complete, the anaphoric DRS is resolved against the main DRS taking syntactic constraints, accessibility constraints and the distance between the anaphor and the antecedent into consideration. If the anaphora resolution algorithm finds a suitable antecedent, then the anaphoric discourse referent `Y` is related to the accessible discourse referent `X` via an equation of the form `Y=X` and this equation is added locally to the main DRS. If no suitable antecedent for an anaphoric expression can be found in the main DRS, then the discourse referent and the corresponding conditions are accommodated as highly as possible to the main DRS. Note that non-anaphoric definite descriptions are processed in PENG Light exactly in the same way as indefinite noun phrases. The result of the anaphora resolution process is always reflected in a paraphrase and immediately visible for the author.

6.3 The Extended Anaphora Resolution Algorithm

The extended anaphora resolution algorithm takes care of bridging definite descriptions and modifies the original algorithm of PENG Light in a systematic way. This new algorithm relies on interactivity and allows the author to specify semantic relations between a noun phrase antecedent and a bridging definite description as part of the specification if this information is not already available. This solution is compatible with the predictive authoring approach that is supported by PENG Light since the anaphora resolution process is machine-guided and the author selects among a number of options [15].

In a first step, the new algorithm searches for syntactic matches in the main DRS in a similar way as the original algorithm, selects the most recent accessible discourse referent but additionally collects all other candidate antecedents in a list. For example, the mini-discourse in (44) contains two possible noun phrase antecedents:

44. An academic A teaches COMP448 in E6A.
    An academic B supervises Mary on Monday.
    The academic ...

The anaphora resolution algorithm selects the most recent discourse referent as antecedent and replaces the definite description by the noun phrase antecedent in the paraphrase:

45. { The academic B } ...

If the author is not happy with this solution, then he can access one of the other candidates in the list via a keyboard shortcut. Each repeated keystroke replaces the current solution with an alternative binding and updates the main DRS. This basically allows the author to iterate through the list of candidate antecedents and view the result immediately in the paraphrase.

As explained above, the anaphora resolution algorithm of PENG Light is embedded into the grammar and triggers whenever a noun phrase has been processed. In order to process bridging definite descriptions, the algorithm communicates with the model builder and this process can be interpreted as a form of question answering whereas the anaphoric DRS forms the query and the model supplies the facts. If the conditions in the anaphoric DRS do not fully or partially match with the accessible conditions in the main DRS, then the anaphora resolution algorithm sends the main DRS first to the DRS-to-TPTP translator and subsequently to E-KRHyper. The DRS-to-TPTP translator takes care of the skolemization and stores the information about the substitution of variables (e.g. `X/sk1`). This is an important technical point, since the anaphora resolution algorithm needs to remember how these variables have been replaced because the main DRS will be updated according to the outcome of the anaphora resolution process. E-KRHyper takes the skolemized formula as input and tries to construct a model. The output of E-KRHyper is loaded into the Prolog knowledge base and is then available for further processing. The anaphora resolution algorithm can then search the model for a suitable noun phrase antecedent. Since the discourse referents in the anaphoric DRS are represented as variables, the anaphora resolution algorithm has to work with a copy of these variables in the query. As already discussed in Section 5.2, the model can make correct predictions about the accessibility of discourse referents.

Once the model is available, the anaphora resolution algorithm checks first if the anaphor and the noun phrase antecedent stand in a synonymy relation and then if they stand in a subclass relation. Let us assume that the mini-discourse in (46) has just been processed and let us further assume two different situations: one where the author continues the discourse with the definite description

The scholar that stands in a synonymy relation to the noun phrase antecedent *An academic*, and one where the definite description *The educator* stands in a subclass relation to the same antecedent:

46. Every scholar is defined as an academic.
    Every academic is an educator.
    An academic who teaches COMP448 in E6A owns a computer.

Since we are interested in all solutions, the anaphora resolution algorithm uses the `setof/3` predicate of Prolog and searches first for synonymy in both cases. The search for synonymy is successful in the first case but not in the second case. Therefore, the anaphora resolution algorithm continues in the second case and searches for a subclass relation between the anaphor and a noun phrase antecedent. The following examples (47) illustrate how the `setof/3` predicate is used to check for subclass relations:

47.
```
setof([X,P], (subclass(C,educator), object(X,C,P),
      \+ selected_disjunct(object(X,C,P))), Res).
```

Note that the search for synonymy is done in a similar way exploiting potential synonymy relations and constraints in the model. In both cases, the relevant information about the offset position and the Skolem constant is collected. If a solution is found, then the main DRS is updated and the paraphrase will reflect the binding via an explanation displayed in square brackets, for example:

48. The scholar [is a synonym of an academic] ...
49. The educator [is a subclass of an academic] ...

Additionally, all alternative solutions are collected in both cases and the author can – as explained above – jump into "relinking" mode and iterate over the alternatives in the list. Each iteration will update the DRS and generate the corresponding paraphrase.

If no antecedent can be found, then the conditions in the anaphoric DRS are accommodated and a new model is generated. That means a bridging definite description that stands in a meronymic relation to its antecedent is treated in the same way as an indefinite noun phrase and results in a new discourse referent but is marked as "associated" with another term in the paraphrase once a corresponding part-whole relation has been found in the model:

50. Each mother board is a component of a computer.
    Anna Grau owns a computer.
    The mother board [is a component of a computer] ...

If the author is not happy with a solution, then he can specify a suitable semantic relation as part of the discourse on the interface level. Let us assume that the author continues the discourse in (50) with the definite description *The CPU* but that the relevant background knowledge is not yet available:

51. Each mother board is a component of a computer.
    Anna Grau owns a computer.
    The CPU ...

In this case the author has to switch to the "terminology insertion" mode via a keyboard shortcut that suspends the processing of the current sentence and accepts terminological statements on the fly, for example:

52. Each CPU is a component of a mother board.

Once the terminological information has been specified, the author can switch back to the normal input mode via another keyboard shortcut. The inserted terminological statement is processed as a separate DRS and added to the main DRS. Once this has been done, the model builder updates the model, and the new bridging definite description is licensed in the discourse.

53. Each CPU is a component of a mother board.
    Each mother board is a component of a computer.
    Anna Grau owns a computer.
    The CPU [is a component of a computer] ...

The resulting paraphrase (51) then illustrates the outcome of this knowledge acquisition process, and the author can continue with the writing process.

7 Conclusions

In this paper I showed that anaphora resolution in existing controlled natural languages is quite restricted and argued that we can allow for a wider range of anaphoric expressions if we consider the anaphora resolution process as an interactive knowledge acquisition process. I focused, in particular, on bridging definite descriptions and illustrated what kind of inference-supporting background knowledge is required in order to establish a bridge between an anaphor and a suitable noun phrase antecedent. I discussed three important types of semantic relations (synonymy relations, subclass relations and part-whole relations) that are required in order to support the resolution of bridging definite descriptions. I showed that the naive treatment of part-whole relations results in invalid inferences and therefore potentially wrong resolutions of bridging references. To solve this problem the current implementation of PENG Light uses a more fine-grained representation of meronymic relations than is supported in WordNet. I further argued that a model builder can not only be used as a tool for consistency checking or question answering in a controlled natural language context, but also for resolving anaphoric references that require inference and inference-supporting knowledge. The new anaphora resolution algorithm of PENG Light extends the existing syntax-based algorithm in a systematic way and allows the author to choose among alternative noun phrase antecedents and to add additional terminological knowledge in order to establish an explicit link between an anaphor and its antecedent while a specification is written. The presented approach fits well together with the predictive interface approach of PENG Light that guides the writing process and allows for adding terminological knowledge in an interactive way. It is important to note that this approach relies on the author of the text – in our case on a domain expert – who works in **collaboration** with the machine and provides the required ontological knowledge while a text is written – this approach brings the human author back into the loop.

Acknowledgments. I would like to thank Norbert E. Fuchs for organising the Workshop on Controlled Natural Language (CNL 2009) on Marettimo Island west of Sicily (Italy), and also three anonymous reviewers of this paper for their constructive and valuable suggestions.

References

1. Asher, N., Lascarides, A.: Bridging. Journal of Semantics 15, 83–113 (1998)
2. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F.: The Description Logic Handbook. Cambridge University Press, Cambridge (2003)
3. Baumgartner, P., Kühn, M.: Abducing Coreference by Model Construction. Journal of Language and Computation 1(2), 175–190 (2000)
4. Baumgartner, P., Furbach, U., Pelzer, B.: Hyper Tableaux with Equality. In: Pfenning, F. (ed.) CADE 2007. LNCS (LNAI), vol. 4603, pp. 492–507. Springer, Heidelberg (2007)
5. Bos, J., Buitelaar, P., Mineur, A.: Bridging as Coercive Accommodation. In: Proceedings of CLNLP 1995, pp. 1–16 (1995)
6. Bos, J.: Exploring Model Building for Natural Language Understanding. In: Proceedings of ICoS-4, pp. 41–55 (2003)
7. Chaffin, R., Herrmann, D.J., Winston, M.: An Empirical Taxonomy of Part-Whole Relations: Effects of Part-Whole Relations of Relation Identification. Language and Cognitive Processes 3(1), 17–48 (1988)
8. Clark, H.H.: "Bridging". In: Schank, R., Nash-Weber, B. (eds.) Theoretical Issues in Natural Language Processing, pp. 169–174. Cambridge, Massachusetts (1975)
9. Donnellan, K.: Reference and Definite Descriptions. Philosophical Review 77, 281–304 (1966)
10. Fellbaum, C.: WordNet, An Electronic Lexical Database. MIT Press, Cambridge (1998)
11. Frege, G.: Über Sinn und Bedeutung. In: Zeitschrift für Philosophie und philosophische Kritik, NF 100, pp. 25–50 (1892)
12. Fuchs, N.E., Kaljurand, K., Kuhn, T.: Attempto Controlled English for Knowledge Representation. In: Baroglio, C., Bonatti, P.A., Małuszyński, J., Marchiori, M., Polleres, A., Schaffert, S. (eds.) Reasoning Web. LNCS, vol. 5224, pp. 104–124. Springer, Heidelberg (2008)
13. Hobbs, J.R., Stickel, M.E., Appelt, E., Martin, P.: Interpretation as Abduction. Artificial Intelligence 63, 69–142 (1993)
14. Kamp, H., Reyle, U.: From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT. Kluwer, Dordrecht (1993)
15. Kuhn, T., Schwitter, R.: Writing Support for Controlled Natural Languages. In: Proceedings of ALTA 2008, Tasmania, Australia, pp. 46–54 (2008)

16. Lewis, D.: Scorekeeping in a language game. Journal of Philosophical Logic 8, 339–359 (1979)
17. Ludlow, P.: Descriptions. In: Stanford Encyclopedia of Philosophy, July 26 (2007), `http://plato.stanford.edu/entries/descriptions/`
18. Manthey, R., Bry, F.: Satchmo: a theorem prover implemented in prolog. In: Lusk, E., Overbeek, R. (eds.) CADE 1988. LNCS, vol. 310, pp. 415–434. Springer, Heidelberg (1988)
19. Mitkov, R.: Anaphora Resolution. In: Mitkov, R. (ed.) Oxford Handbook of Computational Linguistics, pp. 266–283. Oxford University Press, Oxford (2003)
20. Pelzer, B., Wernhard, C.: System Description: E-KRHyper. In: Pfenning, F. (ed.) CADE 2007. LNCS (LNAI), vol. 4603, pp. 508–513. Springer, Heidelberg (2007)
21. Poesio, M., Vieira, R., Teufel, S.: Resolving Bridging References in Unrestricted Text. In: Proceedings of ACL 1997 Workshop on Operational Factors in Practical, Robust, Anaphora Resolution for Unrestricted Texts, July 7-11, pp. 1–6 (1997)
22. Poesio, M., Ishikawa, T., Schulte im Walde, S., Vieira, R.: Acquiring lexical knowledge for anaphora resolution. In: Proceedings of LREC, Las Palmas, May 2002, pp. 1220–1224 (2002)
23. Poesio, M.: Associative descriptions and salience: a preliminary investigation. In: Proceedings of the ACL Workshop on Anaphora, Budapest, April 2003, pp. 31–38 (2003)
24. Pustejovsky, J.: The Generative Lexicon: A theory of computational lexical semantics. MIT Press, Cambridge (1995)
25. Ramsey, A.M., Seville, H.L.: Models and Discourse Models. Journal of Language and Computation 1(2), 159–174 (2000)
26. Russell, B.: On denoting. Mind 14(56), 479–493 (1905)
27. van der Sandt, R.A.: Presupposition Projection as Anaphora Resolution. Journal of Semantics 9(4), 333–377 (1992)
28. Schwitter, R., Ljungberg, A., Hood, D.: ECOLE – A Look-ahead Editor for a Controlled Language. In: Proceedings of EAMT-CLAW 2003, May 2003, pp. 141–150 (2003)
29. Schwitter, R., Tilbrook, M.: Meaningful Web Annotations for Humans and Machines using CNL. Expert Systems 25(3), 253–267 (2008)
30. Strawson, P.F.: On Referring. Mind 59(235), 320–344 (1950)
31. Sutcliffe, G., Suttner, C.B.: The TPTP Problem Library: CNF Release v1.2.1. Journal of Automated Reasoning 21(2), 177–203 (1998)
32. Vieira, R., Poesio, M.: An Empirically-Based System for Processing Definite Descriptions. Computational Linguistics 26(4), 539–593 (2000)
33. Wernhard, C.: System Description: KRHyper. In: Fachberichte Informatik 14/2003, Universität Koblenz-Landau (2003)
34. White, C., Schwitter, R.: An Update on PENG Light. In: Proceedings of ALTA 2009, vol. 7, pp. 80–88 (2009)
35. Winston, M.E., Chaffin, R., Herrmann, D.: A Taxonomy of Part-Whole Relations. Cognitive Science 11(4), 417–444 (1987)