Technical Report

UCAM-CL-TR-661
ISSN 1476-2986

Number 661

UNIVERSITY OF CAMBRIDGE

Computer Laboratory

Compatible RMRS representations from RASP and the ERG

Anna Ritchie

March 2006

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

Technical reports published by the University of Cambridge
Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/TechReports/

ISSN 1476-2986

Compatible RMRS representations from RASP and the ERG

Anna Ritchie
University of Cambridge
Computer Laboratory
Anna.Ritchie@cl.cam.ac.uk

Abstract

Various applications could potentially benefit from the integration of deep and shallow processing techniques. A universal representation, compatible between deep and shallow parsers, would enable such integration, allowing the advantages of both to be combined. This paper describes efforts to make RMRS such a representation. This work was done as part of DeepThought, funded under the 5th Framework Program of the European Commission (contract reference IST-2001-37836).

1 Introduction

Two aims of DeepThought (Hybrid Deep and Shallow Methods for Knowledge-intensive Information Extraction) are to research the potential of deep semantic processing if combined with shallow methods for robustness and to demonstrate this potential by applying the novel approach to three particular applications (*DeepThought* 2002). While a number of deep processing systems have been developed that have broad coverage of linguistic phenomena and can produce detailed semantic representations for this language, such systems are noted to have limited usefulness in practice. Shallower systems that do not require such a high level of lexical information nor assume grammaticality of their input boast the advantage of increased robustness and speed over deeper systems. The expectation is that, through integration, a single application can offer the advantages of both.

In order to successfully combine deep and shallow techniques, the notion of a universal representation becomes important; a standard representation that both deep and shallow processors can produce and that is compatible between both. Copestake (2004) argues the suitability of a semantic representation as this interface; in particular, Robust Minimal Recursion Semantics (RMRS).

With this end-goal in mind, work has been undertaken to make RMRSs produced by particular deep and shallow processors compatible; namely, the deep LinGO English Resource Grammar (ERG) (Copestake & Flickinger 2000) with the Robust Accurate Statistical Parser (RASP) (Briscoe & Carroll 2002) as its shallower counterpart. A test suite of over one hundred English sentences was developed to exhibit a wide range of semantic phenomena handled by the ERG and these sentences parsed by both systems i.e. the ERG loaded into the Linguistic Knowledge Building system (LKB) (Copestake 2002) and the RASP system.

New functionality recently added to the LKB was then used to extract RMRSs from the RASP trees and compare these to the corresponding ERG RMRSs, with the ERG representations taken to be normative. Based on the ERG RMRSs produced from the semantic test suite, the mechanism used to convert RASP trees to RMRSs was gradually modified and developed to try to ensure compatibility between ERG and RASP RMRSs; aiming to make the two as similar as possible while maintaining correctness wherever possible.

The aim of this document is to record to what extent the desired compatibility is possible and the ways in which discrepancies arose between the RMRSs[1]. Section 2 describes the process of extracting RMRSs from RASP trees in more detail, while Section 3 gives a brief introduction to the new RMRS comparison functionality in the LKB. Section 4 then lists particular issues that arose preventing 100% identicalness between the ERG and RASP RMRSs.

2 RASP to RMRS conversion

The conversion from RASP output to RMRS works off its syntactic trees, rather than one of its alternative output forms. Two sets of XML-encoded rules define what components of RMRS are generated from what parts of trees;

[1] The resources used in this work are under active development and details may well change.

```
<le>
<tag>NN1</tag>
<comment>14k sg nouns</comment>
<semstruct>
<hook><index>X</index><label>H</label></hook>
<ep><pred/><label>H</label><var>X</var></ep>
</semstruct>
</le>
```

Figure 1: Lexical rule for RASP's NN1 tag

```
<rule>
<name>N1/n_n1</name>
<comment>ANNA - noun-noun compound treatment (sem.ex.48,49)</comment>
<dtrs><dtr>N</dtr><dtr>OPT</dtr><dtr>N1</dtr></dtrs>
<head>RULE</head>
<semstruct>
<hook><index>X</index><label>H</label></hook>
<ep><gpred>COMPOUND_REL</gpred><label>H</label><var>E</var></ep>
<ep><gpred>UDEF_Q_REL</gpred><label>H1</label><var>X1</var></ep>
<rarg><rargname>ARG1</rargname><label>H</label><var>X</var></rarg>
<rarg><rargname>ARG2</rargname><label>H</label><var>X1</var></rarg>
<rarg><rargname>RSTR</rargname><label>H1</label><var>H4</var></rarg>
<rarg><rargname>BODY</rargname><label>H1</label><var>H3</var></rarg>
<ing><ing-a><var>H</var></ing-a><ing-b><var>H2</var></ing-b></ing>
</semstruct>
<equalities><rv>X</rv><dh><dtr>N1</dtr><he>INDEX</he></dh></equalities>
<equalities><rv>X1</rv><dh><dtr>N</dtr><he>INDEX</he></dh></equalities>
<equalities><rv>H2</rv><dh><dtr>N1</dtr><he>LABEL</he></dh></equalities>
<equalities><rv>H4</rv><dh><dtr>N</dtr><he>LABEL</he></dh></equalities>
</rule>
```

Figure 2: Grammar rule for RASP's N1/n_n1 rule

specifically, from part-of-speech (POS) tags and RASP grammar rules. These rules were initially created by an assortment of manual, semi-automatic and automatic techniques before being manually edited for the purposes of this work. An example of both a rule defining RMRS created from a RASP grammar rule (henceforth, a *grammar rule*) and one for creating RMRS from a POS tag (a *lexical rule*) are discussed to illustrate their syntax and function.

Figure 1 shows the rule for words tagged as singular common nouns. The name of the POS tag that the rule governs (NN1 in this case) is given within the ⟨tag⟩ tags and a description of the words with this POS is found between the ⟨comment⟩s. The interesting part of the rule is the ⟨semstruct⟩, the XML description of the semantic structure arising from a word of this POS, that generally consists of the *hook* and one or more *elementary predications* (EPs)[2]. This shows that a noun of this type has a normal-type variable (X) for its *index* and that this, together with an identifying *label* (H), constitutes the semantic entity's hook[3]. A single EP is introduced, by the one ⟨ep⟩ line in the rule. The ⟨pred/⟩ tag indicates that the lemma tagged NN1 should be extracted from the tree and incorporated into the resultant predicate name according to a fixed format (the word delimited by underscores and followed by a broad POS tag i.e. n for nouns, v for verbs, j for adjectives, r for adverbs and x otherwise). The entire rule is enclosed in ⟨le⟩ tags.

Figure 2 shows the rule for applications of the noun-noun compound rule. The grammar rules are largely similar to lexical rules: the rule is contained within ⟨rule⟩ rather than ⟨le⟩ tags, ⟨name⟩ tags replace ⟨tag⟩ in enclosing the identity of the tree component to which the rule applies (here N1/n_n1), the ⟨comment⟩ tags hold relevant information about the rule and the ⟨semstruct⟩ defines the semantic structure of the syntactic constituent to which the rule is applied.

However, these rules differ in a number of respects. Firstly, note the necessary addition of the ⟨dtrs⟩ and

[2] A minority of tags supply no EP or supply semantics additional to EPs.

[3] For an explanation of these terms and the semantic algebra from which they arise, the reader is referred to (Copestake, Lascarides & Flickinger 2001).

```
compound_rel(h7,e9)
  ARG1(h7,x2)
  ARG2(h7,x4)
  ING(h7,h5)
udef_q_rel(h10,x4)
  RSTR(h10,h3)
  BODY(h10,h13)
_garden_n(h3,x4)
_dog_n(h5,x2)
```

Figure 3: RMRS for "garden dog"

⟨head⟩ lines to identify the daughters and semantic head of the rule respectively. When generated automatically, the head of a rule is set to the syntactic head[4](unless one cannot be found e.g. in co-ordinate structures) although this may be altered manually (e.g. to the right-most daughter in the case of co-ordinates). The head may also be set to RULE to indicate that the head should be supplied by the ⟨semstruct⟩.

Secondly, the ⟨semstruct⟩ may contain more than the hook and EPs. Since the grammar rules generally deal with more than a single word, RMRS components describing the relationship between the rule daughters may be introduced, such as arguments, in-groups and qeqs. In terms of XML tags, these are ⟨rarg⟩s, ⟨ing⟩s and ⟨hcons⟩s (not present in the N1/n_n1 rule). An *anchor*[5] may also be defined within ⟨slots⟩ tags.

Finally, grammar rules may define some number of ⟨equalities⟩ outside the ⟨semstruct⟩ that define which variables inside it are equated with which variables in the daughters.

To more adequately demonstrate the function of these rules, the RMRS that results from their application to the phrase *garden dog* is given in Figure 3. The two bottom-most EPs are those introduced by two applications of the NN1 lexical rule; one for *garden*, one for *dog*. In each case, the ⟨pred/⟩ tag takes the word tagged NN1 and creates the predicate name shown, attaching to it its label (h5 for *dog*) and ARG0 variable (x2). These variables are co-indexed with the ⟨label⟩ and ⟨index⟩ in the ⟨hook⟩ line of the lexical rule, thus defining h5 and x2 as the externally visible parts of the semantic entity.

The words *garden* and *dog* then become the N and N1 daughters of the N1/n_n1 rule respectively[6]. The two remaining EPs are similarly introduced by ⟨ep⟩ lines, this time in the grammar rule. However, these are grammatical predicates, rather than the lexical predicates introduced to represent specific words, and as such are named explicitly in the rule, within the ⟨gpred⟩ tags. Again, each EP is attached to a label and variable, defined in the rule. Then, the four ⟨rarg⟩ lines each introduce one argument into the RMRS: the ARG1, ARG2, RSTR and BODY. These too have each a label and variable. Note that the ARG1 and ARG2 share the label h7 with the **compound_rel** (and are consequently aligned with it). This is due to the co-indexation of the variable H in the rule. Similarly, the ⟨ing⟩ defines H to be the first label in an in-group and thus introduces the ING aligned with the **compound_rel**.

Further note that the ARG1's variable is x2, the ARG0 of the **_dog_n**. This is the result of the first of the ⟨equalities⟩, that equates X (the ARG1's variable) with the N1 daughter's index. As mentioned, the daughter's index is co-indexed with its ARG0 variable inside the lexical rule that created the **_dog_n** EP and so the compound_rel's ARG1 becomes x2. Likewise, the second label in the ING is h5, the label of the **_dog_n**, due to the third of the ⟨equalities⟩ between H2, the variable name used for the ⟨ing-b⟩, and N1's label.

Thus, as lexical and grammar rules are applied to POS-tagged words and gradually larger constituents in a sentence, successive layers of RMRS components are incrementally added to the sentence's RASP RMRS and relationships between constituents forged.

3 RMRS comparison functionality

The LKB now provides menu options to select and compare pairs of RMRSs. The comparison window displays the two RMRSs side-by-side, using coloured lines to indicate pairs of components that are judged to correspond.

[4]The extraction of the syntactic head is often straightforward because RASP rules follow the GPSG convention that the head daughter has an identifier that starts with an 'H'.

[5]Again, the reader is referred to (Copestake et al. 2001) for details.

[6]The OPT is an optional punctuation daughter

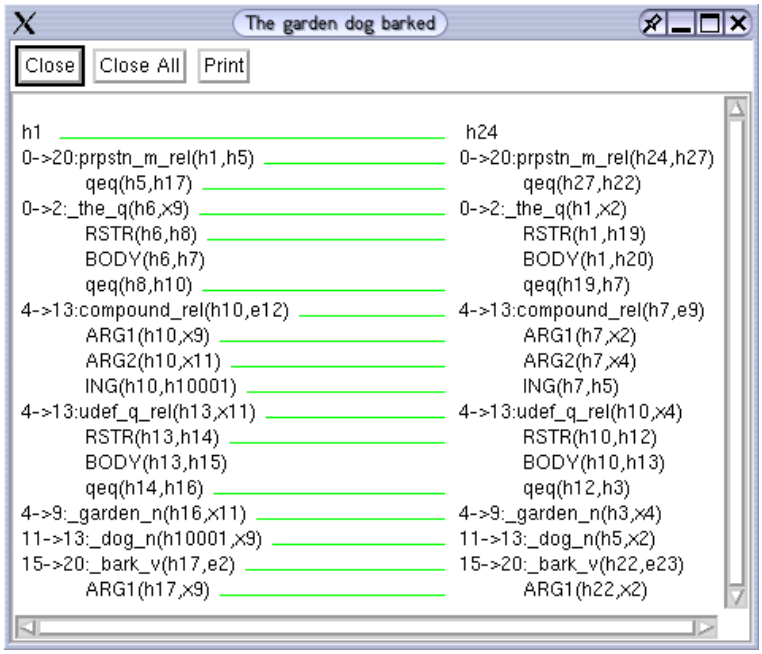

Figure 4: Comparison window for *The garden dog barked.*

Throughout this work, comparisons were made between ERG and RASP RMRSs from the same sentence and Figure 4 shows the comparison for *The garden dog barked.* The ERG RMRS appears on the left, with RASP's on the right.

A green line between a pair of components indicates a perfect match (above the level of variable name identity) and inspection of the two RMRSs reveals that they are, in fact, identical. Those components that do not display a match are quantifier BODYs; these will never match since their handle arguments are always unattached. Thus, this sentence serves as an example of one for which the RMRSs produced by the ERG and RASP are perfectly compatible.

This, however, is not the case for many of the sentences in the semantic test suite. There are a number of ways in which the comparison displayed between two RMRSs may appear differently to the perfect example given. For instance, the correspondence line may be blue or red (rather than green) indicating a *subsumptive* match. Fundamental to RMRS is the notion of specificity: RMRSs produced by shallow systems can be taken to be underspecified versions of their deeper equivalents. In the context of this work, this means that it is possible for one RMRS to contain an RMRS component that subsumes the one to which it corresponds and, in these cases, a blue or red correspondence line will be displayed when the ERG or RASP, respectively, is more specific.

Another possible mismatch is manifest as the absence of a correspondence line between two corresponding components. This may occur because the two components are unrelated in terms of specificity or, in the case of predicates, because their *characterisation* information does not match. Alongside each predicate is displayed the range of character positions of the substring of a sentence from which that predicate has arisen. This information may be used to allow for efficient comparison between RMRSs; a potentially exponential problem otherwise. If characterisation is ignored during comparison, as it may be, ambiguity is increased and multiple pairings may be judged possible for a given predicate. Thus, multiple comparison windows are displayed, some with correspondence lines between components that do not correspond; the final manifestation of a mismatch.

Mismatches between ERG and RASP RMRSs may occur for a number of reasons and the following section details the issues brought to light by the semantic test suite, describing how particular mismatches are displayed by the comparison window, explaining why they occur and reasoning about how they might be rectified, if at all. Some issues will be apparent in more than one example but an issue, once identified and discussed, will not be mentioned again.

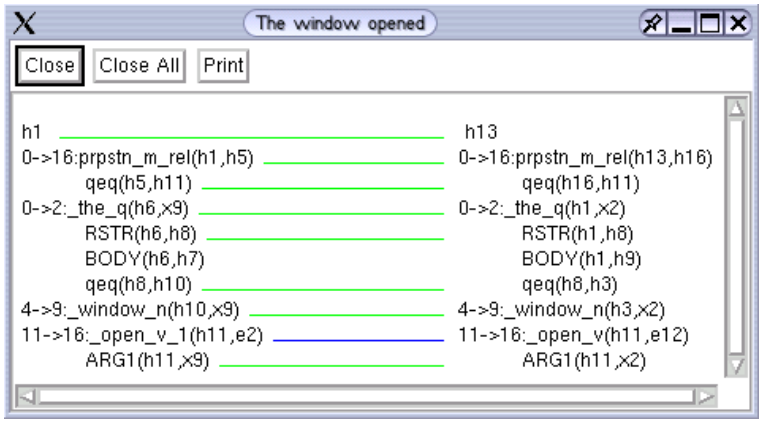

Figure 5: Comparison window for *The window opened.*

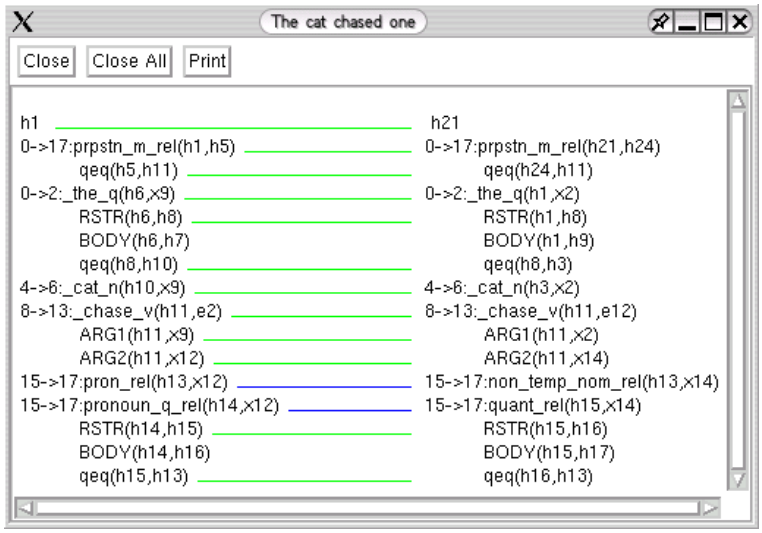

Figure 6: Comparison window for *The cat chased one.*

4 ERG vs RASP RMRS issues

4.1 Subsumptive predicate matches

The formalism of RMRS specificity includes a partial order on predicates, defined in terms of their syntax, and a hierarchy of grammatical predicates. *The window opened* is an example of a sentence where the ERG can produce a more specific lexical predicate, for *opened*, than RASP since it can distinguish between multiple senses of the word *open* whereas RASP, with no lexical information other than potential POS tags, cannot. Figure 5 demonstrates the blue line that indicates that the ERG's **_open_v_1** predicate is subsumed by RASP's **_open_v**, while Figure 6 illustrates the ERG producing more specific grammatical predicates for the pronoun *one* in *The cat chased one*. This is due to the fact that RASP uses one POS tag for all singular indefinite pronouns, not all of which produce the **pron_rel** and **pronoun_q_rel**: *somebody*, for instance, produces a **person_rel** and **some_q_rel**. Thus the lexical rule for this tag must introduce the least general predicates that subsume the predicates produced by each of these pronouns.

While subsumptive matches in the RMRSs illustrate one respect in which the ERG and RASP RMRSs fail to match perfectly, they are not a problem: they simply highlight the fundamental difference in depth of the systems that produce them and do not render the RMRSs incompatible.

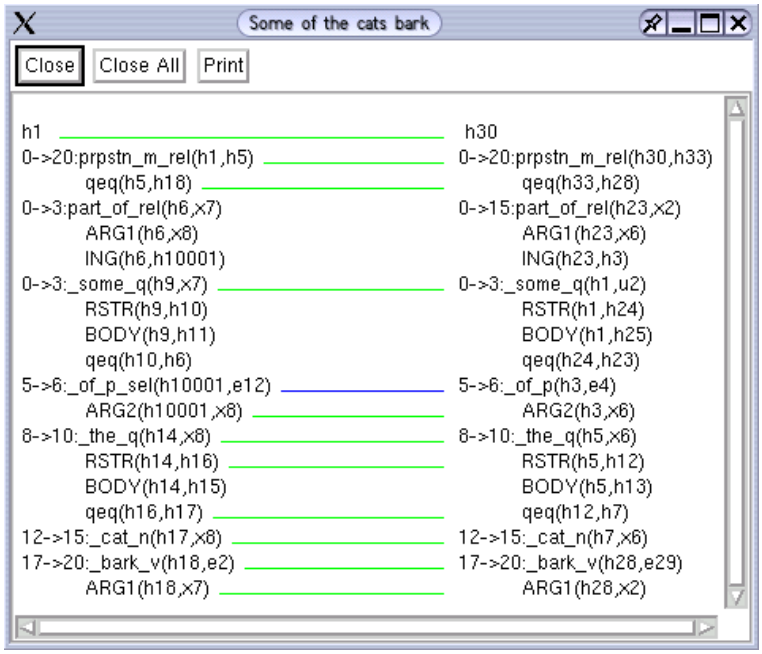

Figure 7: Comparison window for *Some of the cats bark.*

4.2 Characterisation mismatches

The predominant cause of characterisation mismatches is that the ERG and RASP produce the same predicate from a slightly different substring of the sentence. For example, Figure 7 shows the comparison for *Some of the cats bark* when characterisation is taken into account. As can be seen, both RMRSs contain a **_some_q** predicate arising from characters 0 to 3, i.e. the word *Some*, and the green line between these indicates that these identical predicates correspond. Both too contain a **part_of_rel** whose ARG0 variable links it to the **_some_q**. Despite the seeming identicalness of these predicates, however, no correspondence line is present. Inspection reveals that this is because the ERG's **part_of_rel** stems again from the word *Some* itself, whereas RASP's **part_of_rel** is introduced by its NP/np-pro_pp-of grammar rule i.e. the entire *Some of the cats* phrase (characters 0 to 15). As a consequence, the ARG1 and ING attached to the **part_of_rel** and the **_some_q**'s RSTR and qeq (that lead to the **part_of_rel**) are likewise judged not to correspond.

Punctuation was a second cause of characterisation mismatches. Consider the sentence *Abrams, Browne and the dog arrived* in Figure 38. Initially, only those components originating from the word *Abrams* (the **proper_q_rel**, the **named_rel** and their arguments) displayed a match, since the ERG considered character positions after a pre-processor had stripped away punctuation characters. RASP, on the other hand, leaves punctuation in place when calculating character positions. Hence, the position of each character after the comma was misaligned between the ERG and RASP's RMRSs, and each predicate from substrings including characters after the comma were judged to mismatch. This particular problem was fixed by altering the ERG's pre-processing to include punctuation in its characterisation.

Nevertheless, since the sole purpose of including characterisation is to increase efficiency, mismatches of this nature, again, do not signify any incompatibility between ERG and RASP RMRSs and thus characterisation will be ignored in all following comparisons to focus on other issues.

4.3 CARG mismatches

Certain classes of words produce predicates requiring a CARG; a string argument that describes which particular word of that class is in question. Proper names are one example already met, *Abrams* appearing in Figure 38 as a **named_rel** with a CARG containing *abrams*. While these present no problem in terms of compatibility, CARGs for closed class words are more problematic since the ERG alters the word as it appears as a CARG, whereas RASP at present simply extracts the lemma from the tree. Figure 8 shows CARGS for cardinals, days and months as they

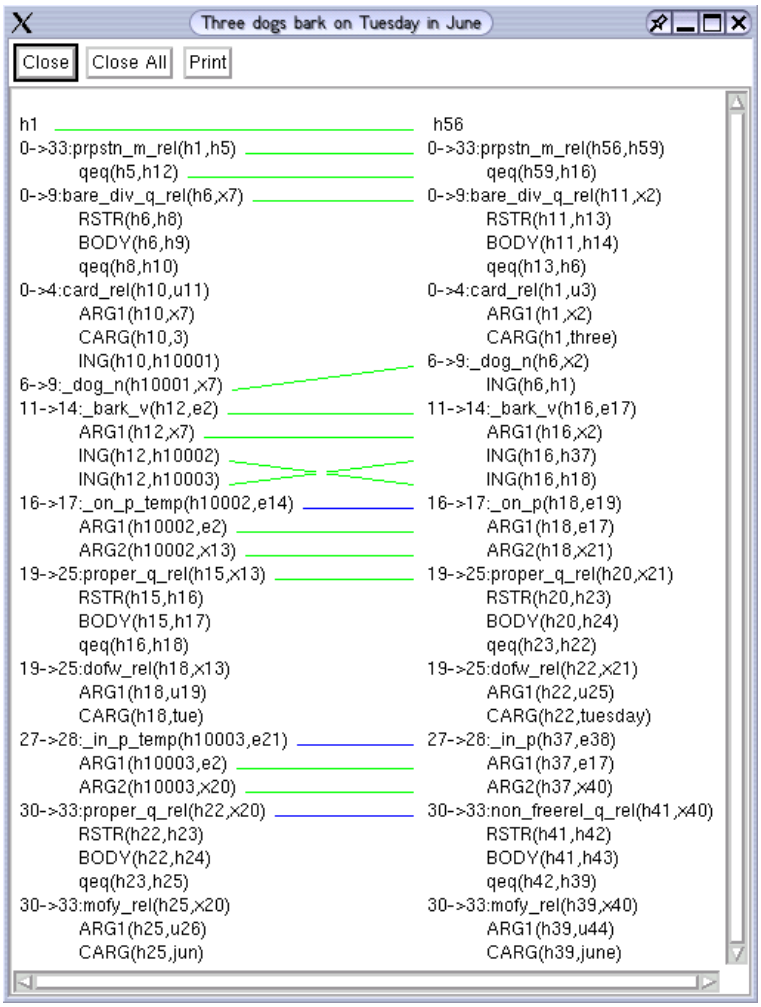

Figure 8: Comparison window for *Three dogs bark on Tuesday in June.*

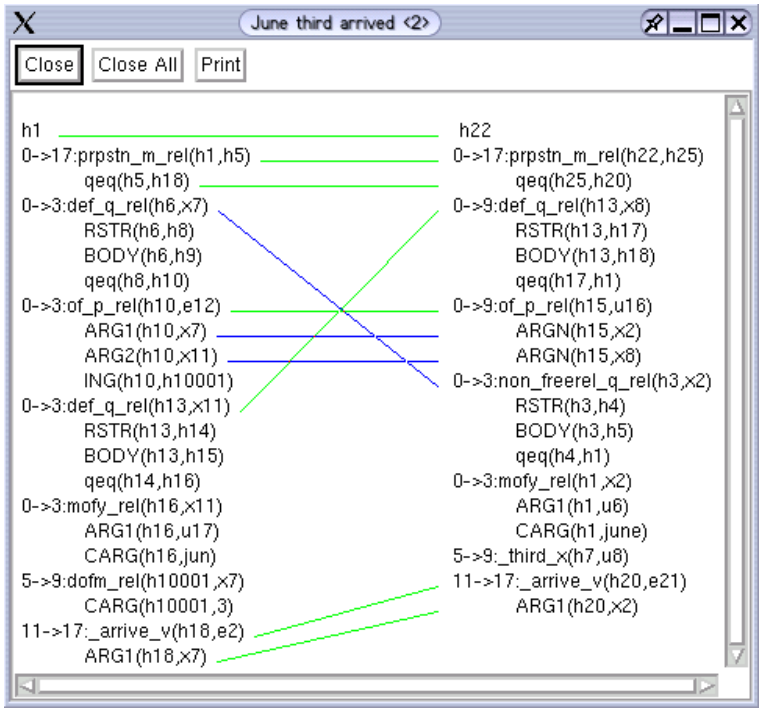

Figure 9: Comparison window for *June third arrived.*

appear (differently) for the ERG and RASP, resulting in mismatches between not only the CARGs themselves but also the predicates to which the CARGs are attached.

This problem should be easily resolved, for the very reason that the ERG can alter the form of some CARGs in the first place: since the words are closed class, a fixed number of rules are required to be added to the RASP-RMRS mechanism to perform the same alterations.

4.4 Dates

Dates illustrate of a number of issues that also arise in other contexts. Consider Figure 9 of the sentence *June third arrived*. As can be seen, the ERG produces a **dofm_rel** and **mofy_rel** (each with associated **def_q_rel**, arguments etc.) for *third* and *June* respectively. A further **of_p_rel** represents the day-of-the-month relationship between these words. Inspecting RASP's RMRS, several mismatches can be identified.

Most strikingly, there is a **third_x** from the word *third* rather than the required **dofm_rel**. This predicate is introduced by the lexical rule for ordinal numbers so one solution would be to replace this predicate with a **dofm_rel** here. However, this is far from ideal since these words can appear in different contexts where they have a different semantics and, thus, require a different predicate. For instance, in *The third dog barks* (Figure 10), *third* is a plain ordinal and requires an **ord_rel** rather than **dofm_rel**. Since the same POS tag is used for both cases, however, the lexical rule cannot introduce these predicates and the default **third_x** is left in place. The same problem occurs with time phrases, as in Figure 11, where the ERG represents times differently from cardinals but RASP, using one POS tag for these numbers, simply applies its grammar rules for numbers and cannot make this distinction.

Furthermore, the rule that combines the *June* and *third* is NP/date, meaning that RASP successfully identifies this phrase as a date. However, this rule simply accepts a temporal noun (*June*) followed by a number noun (*third*) and optionally surrounded by additional number daughters (not present in this example). Given this definition of the NP/date, however, this rule would equally be applied to the phrase *June 1993* as in Figure 12. As this necessitates a **yofc_rel** in place of the **dofm_rel** (and further differences), these predicates can neither be introduced by the grammar rule.

The second issue with dates is the **of_p_rel**'s arguments. As discussed, there is no fixed relationship between the two daughters in the NP/date rule: Figure 9 shows an **of_p_rel** whose ARG1 is the day and ARG2 is the

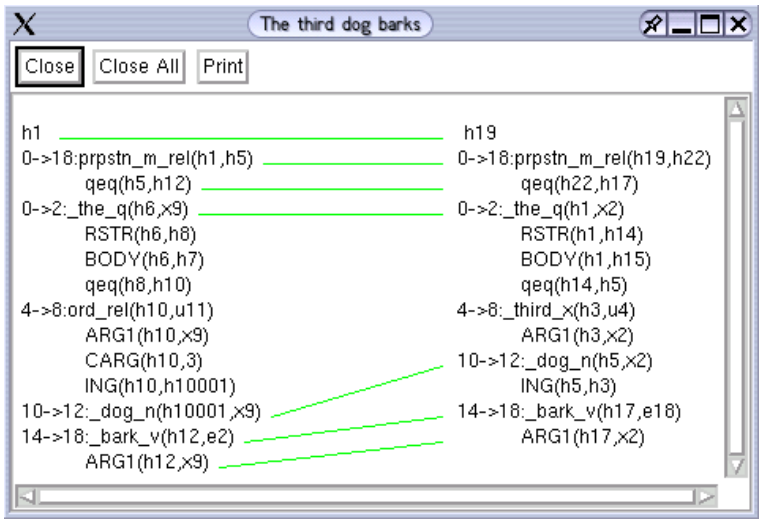

Figure 10: Comparison window for *The third dog arrived.*

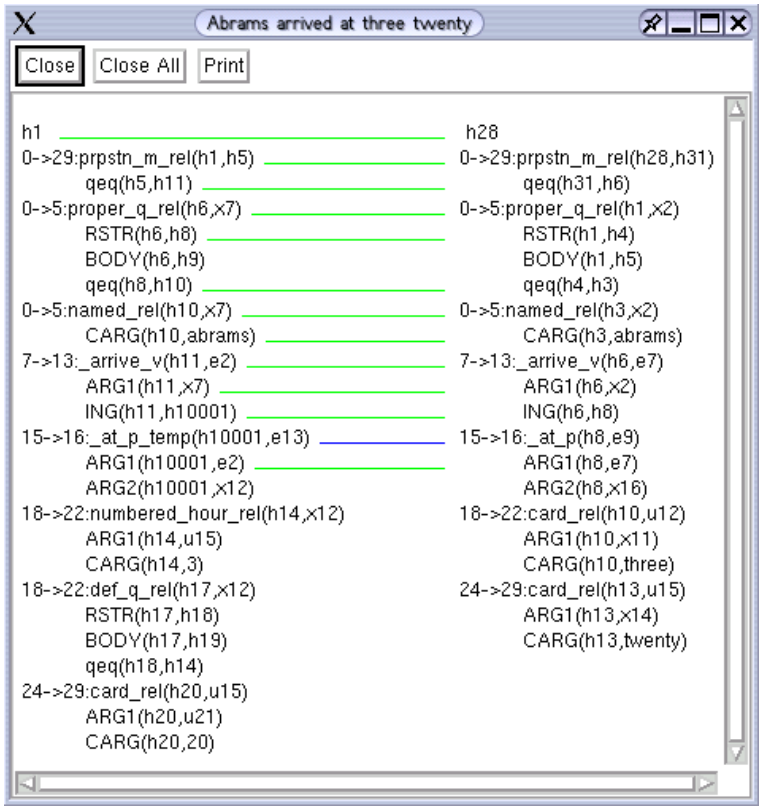

Figure 11: Comparison window for *Abrams arrived at three twenty.*

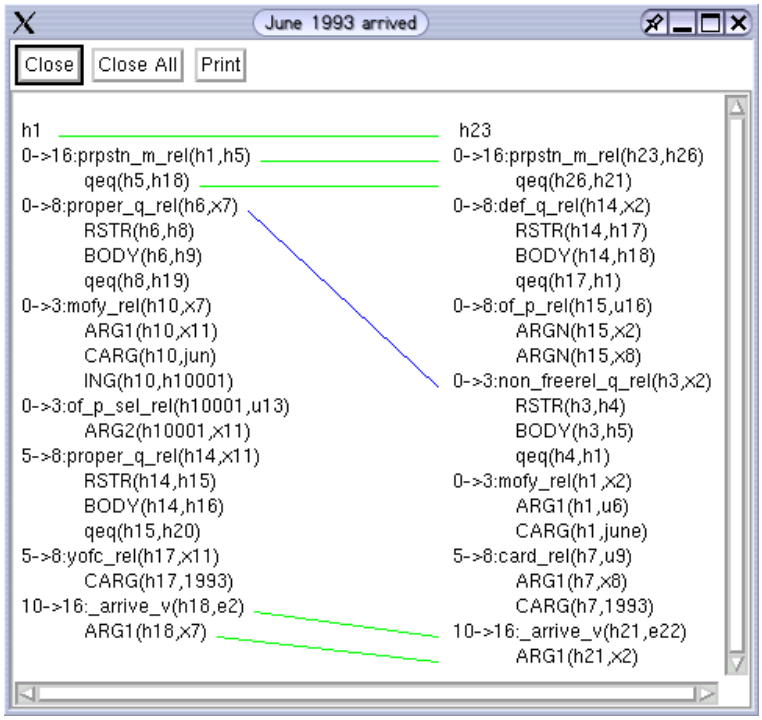

Figure 12: Comparison window for *June 1993 arrived.*

month, whereas Figure 12 shows the year is the ARG2 i.e. the relationship has been reversed in terms of the rule daughters. Thus, in constructing the RMRS for applications of NP/date, the best the rule can do is to introduce an **of_p_rel** with two ARGNs, one of which is the month daughter and the other, the number daughter. A similar issue arises with composite cardinals, where the ERG introduces predicates that specify the relationships between parts of numbers. For instance, the RMRS for the number phrase *two hundred twenty* (Figure 13) would co-index *two* and *hundred* (variable x7), would include a **plus_rel** linking *two hundred* to *twenty* and would further include a **times_rel** linking *two* and *hundred*. However, since the NP/cplx-num rule that covers such phrases accepts a simple sequence of numbers, with no information about how they are related, these predicates cannot be introduced by the rule. Note, however, that RASP does introduce the **times_rel**, though without its FACTOR1 argument. This is done by the lexical rule for RASP's tag for (number neutral) numeral nouns, like *hundred* and *dozen* (all of which appear to require this **times_rel**), but from which there is no access to the first FACTOR.

Furthermore, this grammar rule is peculiar in that it, while it is applied to sequences of numbers, it is defined such that it has a single (non-optional) daughter called CPLX-NUM. The effect of this is that the individual numbers (their indices, labels etc..) cannot be accessed within the rule. So while NP/cplx-num defines the hook of the overall composite number, it cannot equate this with the hooks of any of the comprising numbers and, effectively, there is a missing link between the numbers and the rest of the sentence. This is demonstrated by the INGs attached to **_dog_n** in Figure 8 and Figure 13: the former shows how the noun predicate should be INGed with the quantity predicate (via their labels) but the latter involves a composite number and an application of NP/cplx-num, meaning the label INGed with **_dog_n** is not the label of any of the predicates from the quantity. It is unclear whether anything can be done to solve this problem.

4.5 Unwanted predicates

Another issue that appears in several contexts is that of lexical predicates that are introduced by RASP where they are not required. This is the case for words that can occur in contexts where they are semantically empty. Auxiliary verbs, for example, do not require predicates (*is* in Figure 14) but have contentful equivalents that do (Figure 15). As illustrated, however, the RASP RMRS includes this predicate (**_be_v**) in both cases, since there is a single POS tag for *be* and the predicate originates from this lexical rule. Although the conversion mechanism could be extended in theory to e.g. remove such predicates when they are identified as unwanted, it is not straightforward to

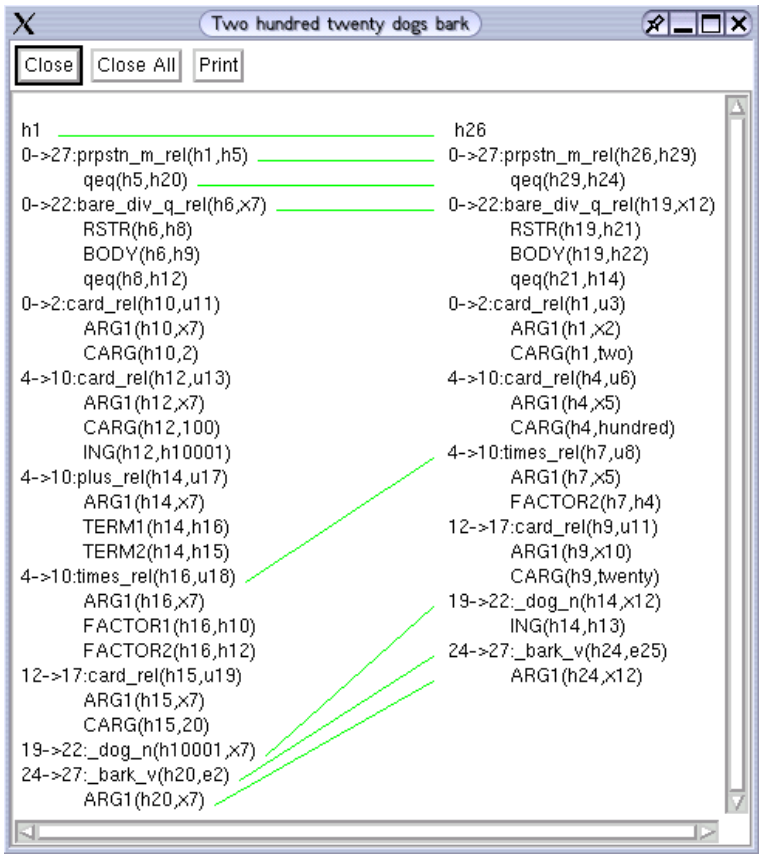

Figure 13: Comparison window for *Twon hundred twenty dogs bark.*

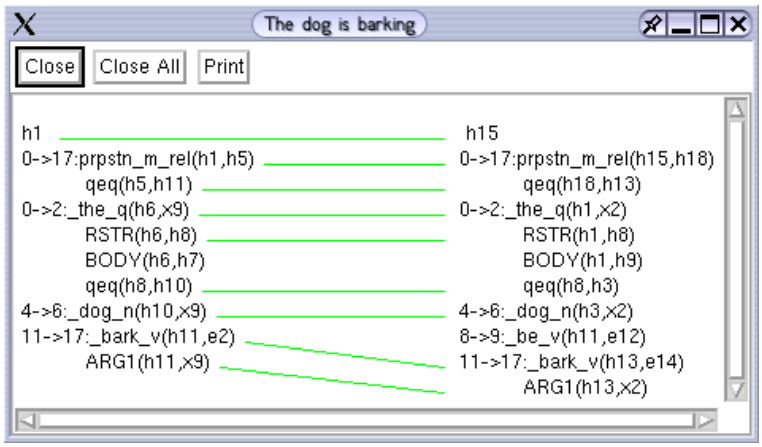

Figure 14: Comparison window for *The dog is barking.*

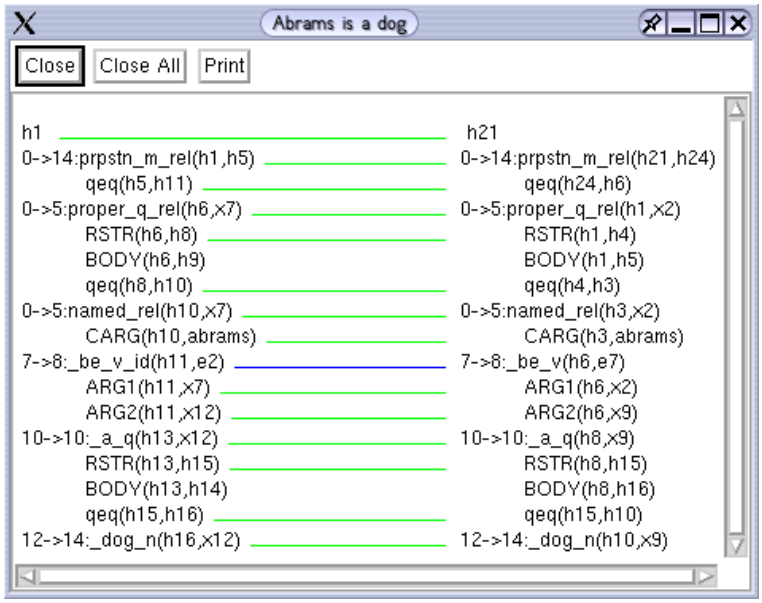

Figure 15: Comparison window for *Abrams is a dog.*

distinguish auxiliaries from their contentful counterparts in practice and this issue may have to be left unresolved.

Modal verbs are a second example of this problem, although one where a solution will be easily attainable. In general, the ERG introduces predicates for modal verbs, such as *could* in Figure 16. The one exception to this rule is *will*: the ERG treats this as a future tense marked rather than a modal, so no predicate is introduced (Figure 17). Since RASP tags *will* the same as all other modal verbs, however, the same lexical rule is applied to it and an unwanted **_will_v** produced. Because this predicate is never required for modal *will* (and only *will*), it should be straightforward to remove the predicate (and its arguments) in this one case.

4.6 Expletive "it" and existential "there"

RASP's lack of lexical information means that it cannot recognise expletive *it*. Hence, in sentences like *It rained* (Figure 18), semantics for a spurious pronoun *it* are added, this entity also appearing as the ARG1 of the raining event. It is unlikely that a solution will be found for this due to the necessity of lexical information in distinguishing the expletive case.

Part of this problem is resolved for existential *there*, since RASP has a special POS tag for this sense of the word and can successfully recognise its occurence. Therefore, this lexical rule can avoid adding any predicate. However, the problem of the spurious entity remains: Figure 19 shows how the RASP RMRS for *There were cats in the garden* has an 'invisible' subject (x2, the index of the removed **_there_x**) for *were*, rather than the index of **_cat_n**. This is because, despite recognising existential *there*, the sentence is still covered by the standard S/np_vp rule that makes the NP (*there*)'s index the subject of the VP's main verb (*were*). Again, there is the potential to rectify this by altering the arguments in cases where the NP has been recognised as existential *there*. It would also be possible to alter the S/np_vp rule so that it does not pick out the NP as the main verb's ARG1 but, since this semantics is correct for the vast majority of sentences and removing it would render their RMRSs quite unhelpful, this would not be a particularly constructive solution.

4.7 Scopal vs non-scopal adverbs

The ERG produces slightly different semantics for scopal and non-scopal adverbs, as illustrated in Figure 20 and Figure 21, respectively. But since RASP's tagset does not make this distinction, it applies the same (RASP) grammar rules for both adverb types and, thus, the (XML) grammar rules cannot strictly introduce either semantics. Note, however, that RASP does, in fact, reproduce the non-scopal adverb semantics for *barked softly*. This is because the ERG disallows scopals to follow the verb so, in the context of this work, it is 'safe' for the relevant grammar rule (V1/vp_advp) to be presumed to apply only to non-scopals.

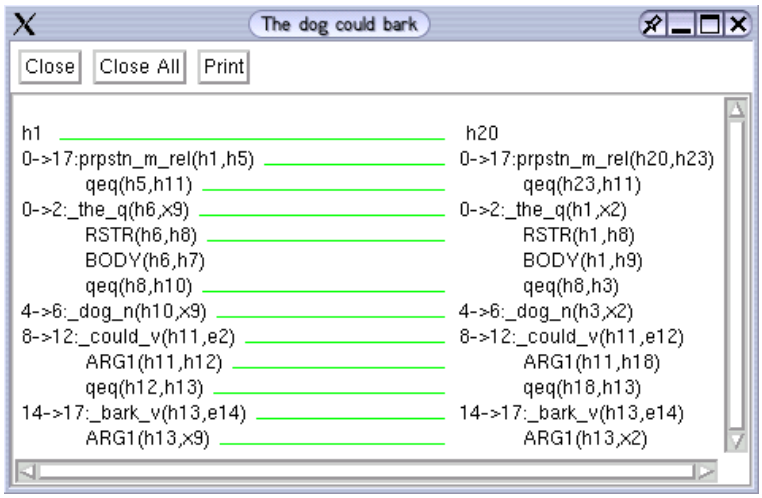

Figure 16: Comparison window for *The dog could bark.*

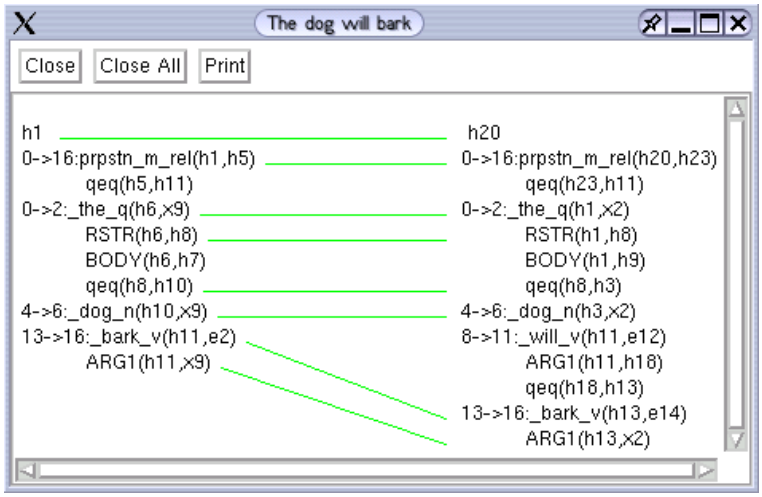

Figure 17: Comparison window for *The dog will bark.*

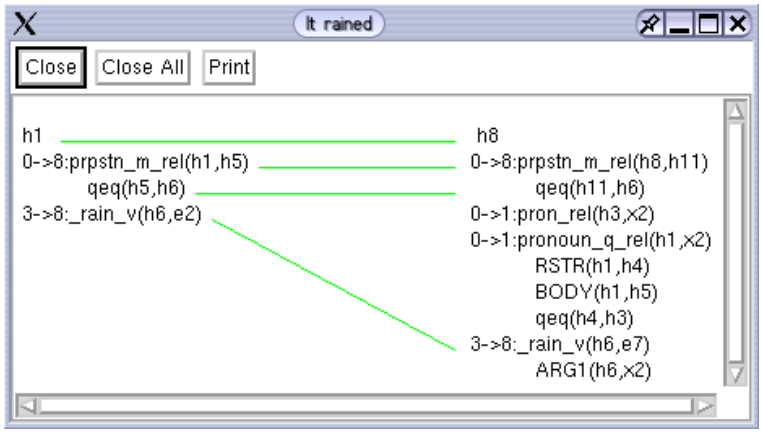

Figure 18: Comparison window for *It rained.*

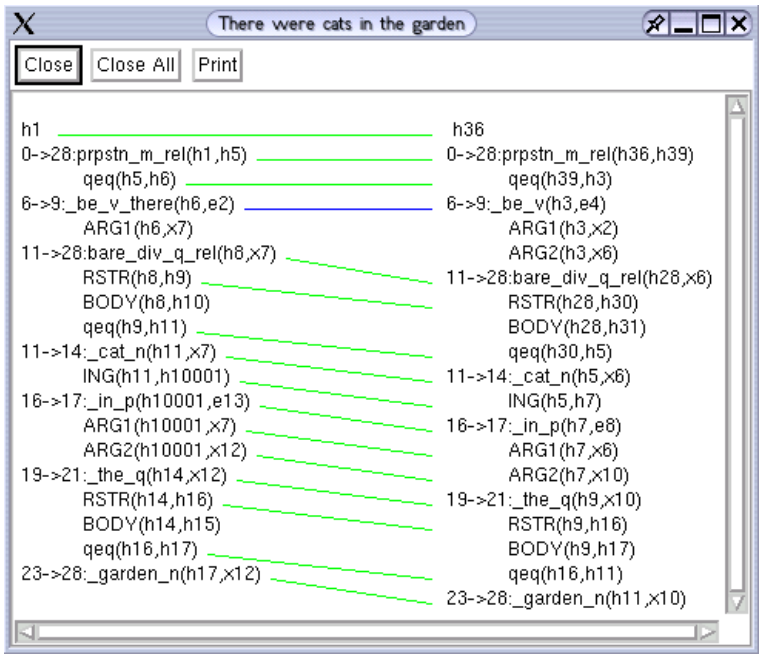

Figure 19: Comparison window for *There were cats in the garden.*

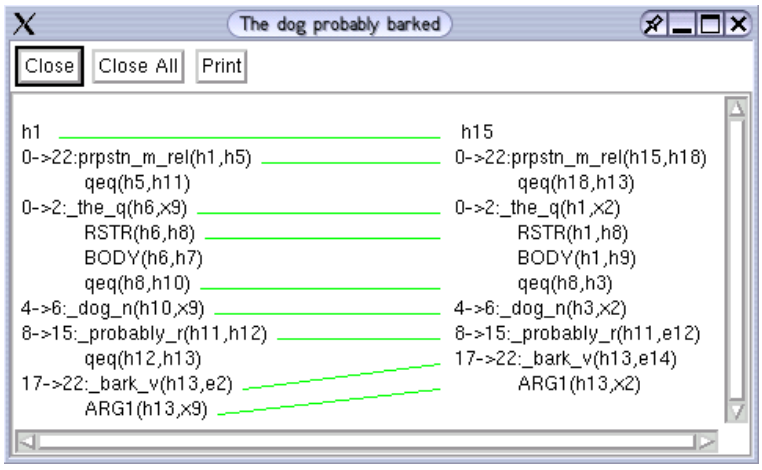

Figure 20: Comparison window for *The dog probably barked.*

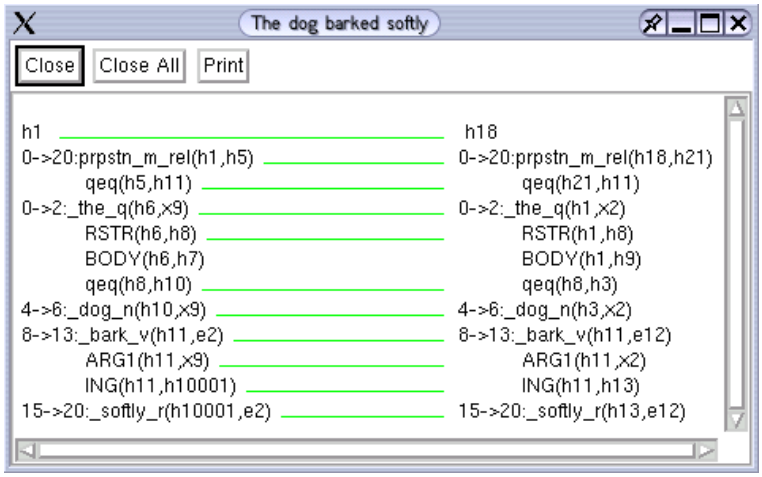

Figure 21: Comparison window for *The dog barked softly.*

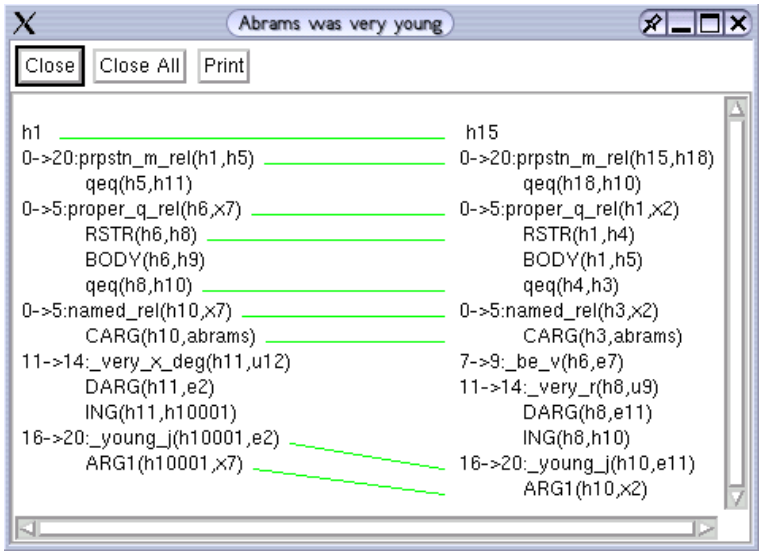

Figure 22: Comparison window for *Abrams was very young.*

4.8 Mismatching predicates

There are a number of situations where the ERG and RASP produce slightly different predicates (for the same word) that do not match even subsumptively, according to the syntax rules for specificity. One case of this is degree adverbs, as in *Abrams was very young* (Figure 22) where the ERG produces **_very_x_deg** whereas RASP can only produce **_very_r** at present. This should be resolvable as the RASP tagset does distinguish degree adverbs.

A further example of mismatching predicates is verbs that appear as adjectives or nouns. RASP can correctly identify these cases and, consequently, produces **_barking_j** and **_chasing_n** for *the barking dog* and *Browne's chasing of cats*, for instance (Figure 23 and Figure 55, respectively). The ERG, on the other hand, treats them as derived forms and, consequently, produces **_bark_v** and **_chase_v**. Similarly, temporal adverbs such as *now* (Figure 24) appear as adjectives (**_now_j**) whereas RASP produces the usual adverbial predicate (**_now_r**). Furthermore, the ERG uses the base form of comparative adjectives in formulating their predicates (**_happy_j**) whereas RASP simply extracts these words, like any other, from the tree (**_happier_j**) as in Figure 25[7].

[7]Note that RASP's **_happier_j** shares its label (h3) with the **comp_rel**. This equivalence was made in order to allow the predicates to share their ARG1, as necessitated by the ERG RMRS. Since no other components are associated with these predicates, sharing the label does not have any adverse effects in this case.

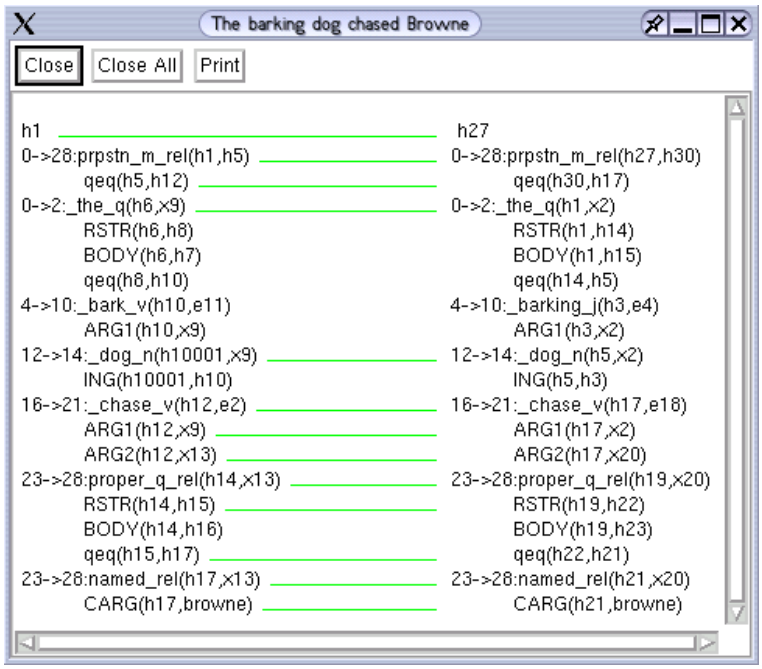

Figure 23: Comparison window for *The barking dog chased Browne.*

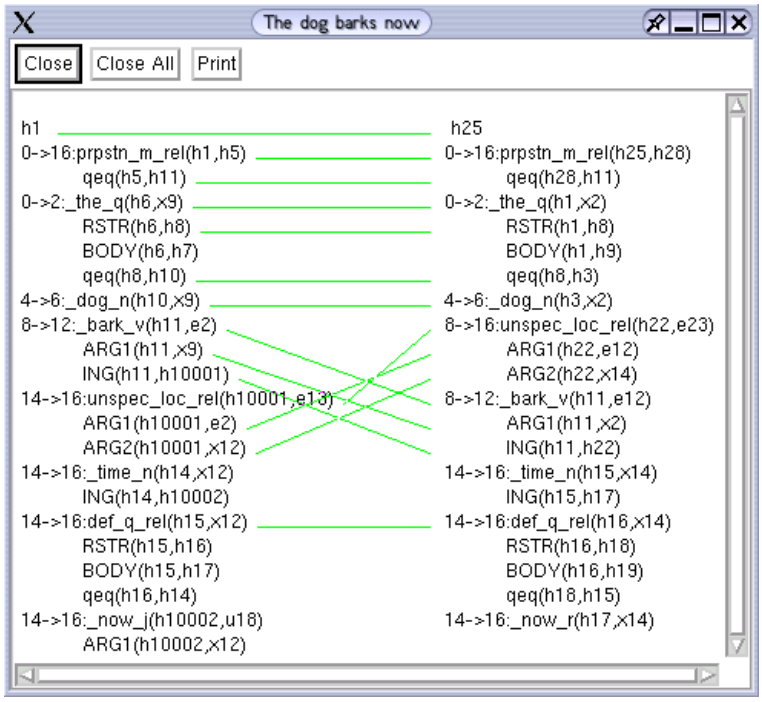

Figure 24: Comparison window for *The dog barks now.*

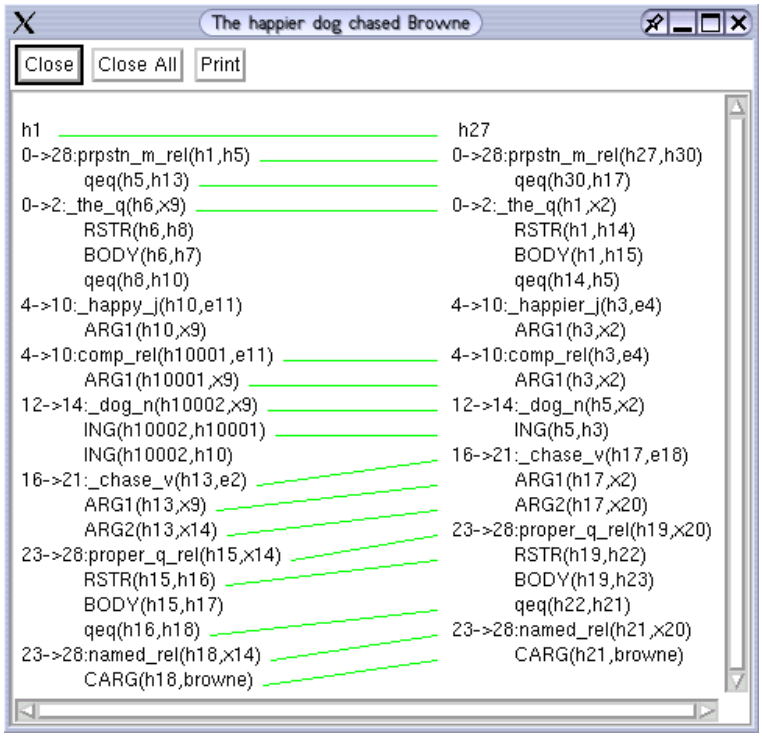

Figure 25: Comparison window for *The happier dog chased Browne.*

It would not be straightforward to change the predicates for *barking* and *chasing* to appear as their verb forms since RASP simply tags them as plain adjectives and nouns, making no distinction between ones that have verb stems and ones that do not. However, RASP does have separate tags for both comparative adjectives and temporal adverbs so it may be possible to replicate the ERG's conversions.

4.9 Composite predicates

Each prepositional phrase rule in RASP's grammar is of the form P1/p_constituent i.e. contains a single preposition. Therefore, whereas the ERG recognises the complicity of *from-* and *to-* in *The dog barked from ten to three* (Figure 26) and creates a **_from_p_to** and **_to_p_sel** predicate, this is simply impossible for RASP as it sees only two distinct PPs.

Similarly, phrasal modal verbs result in composite predicates, as in *The dog is going to bark* (Figure 27). Here, the ERG treats *going to* as a fixed phrase that behaves as a modal verb and produces a **_going+to_v** predicate accordingly. RASP, with no lexical information, simply applies its VP/v_inf rule that covers verbs with infinitival complements and produces the usual lexical predicate for the main verb (**_go_v**) and **prpstn_m_rel** for the barking proposition. The argument structures for the corresponding predicates also differ as a consequence of these alternative interpretations.

4.10 Determiners

Determiners present a couple of problems for RASP to RMRS conversion. Firstly, since the majority of determiners require the same default semantics linking them to their noun phrases, shown in Figure 28, the NP/det_n1 grammar rule introduces this semantics. Certain determiners, on the other hand, need more specialised semantics, such as possessive pronouns. Figure 29 shows how the phrase *my cat* produces a **pro_poss_rel**, **pronoun_q_rel** and **pron_rel**, as well as the usual **def_explicit_q_rel** that is now qeq the **pro_poss_rel** rather than the **_cat_n**. RASP has a separate tag for these determiners but, although this enables RASP to introduce the required predicates in the lexical rule, inspection reveals a number of mismatches. Firstly, the ING linking the **pro_poss_rel** to the **_cat_n** is missing and, secondly, the **def_explicit_q_rel** is still qeq the **_cat_n**. This is due to the fact that the lexical rule has no access to the NP so cannot introduce the ING. Thus the default link to the NP, through the **def_explicit_q_rel**'s

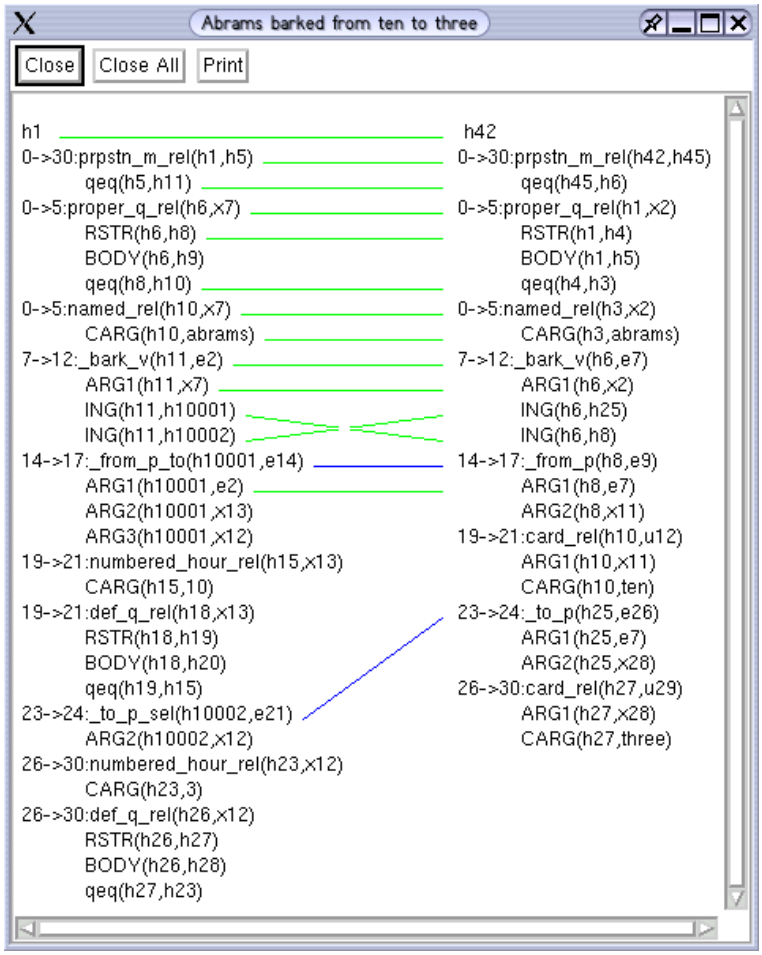

Figure 26: Comparison window for *The dog barked from ten to three.*

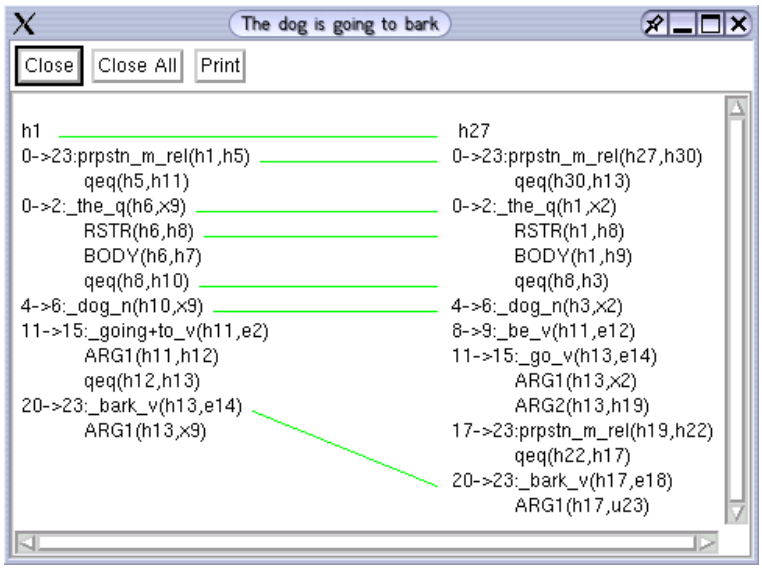

Figure 27: Comparison window for *The dog is going to bark.*

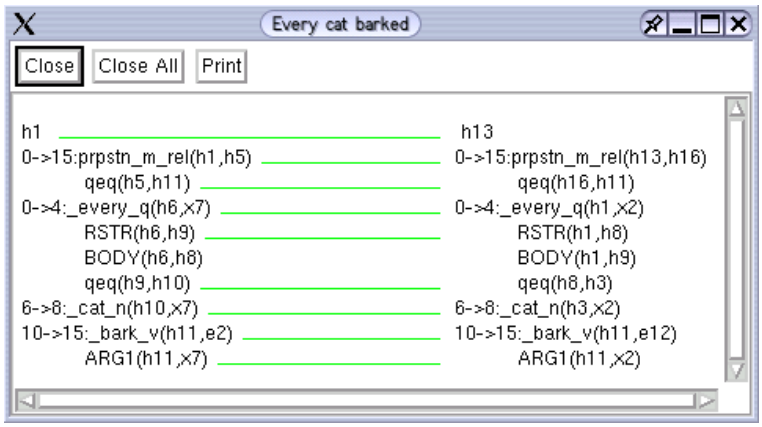

Figure 28: Comparison window for *Every dog barked.*

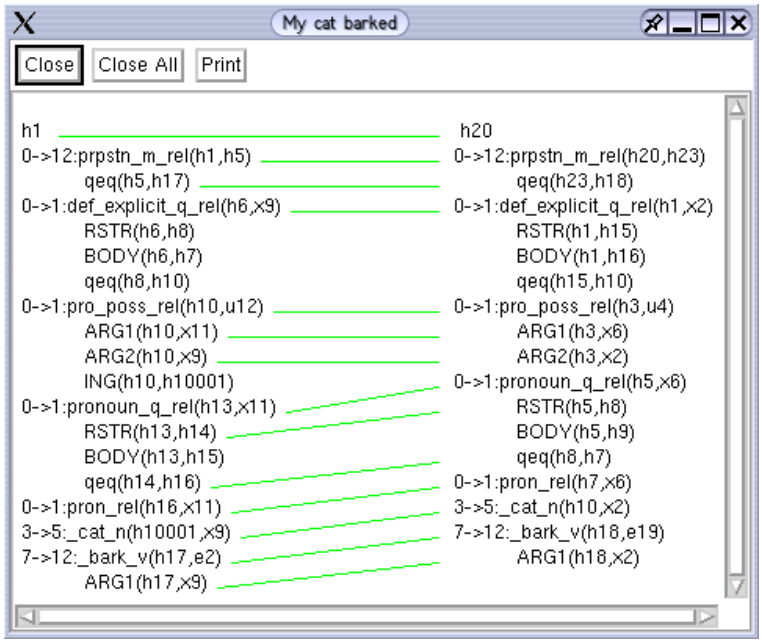

Figure 29: Comparison window for *My cat barked.*

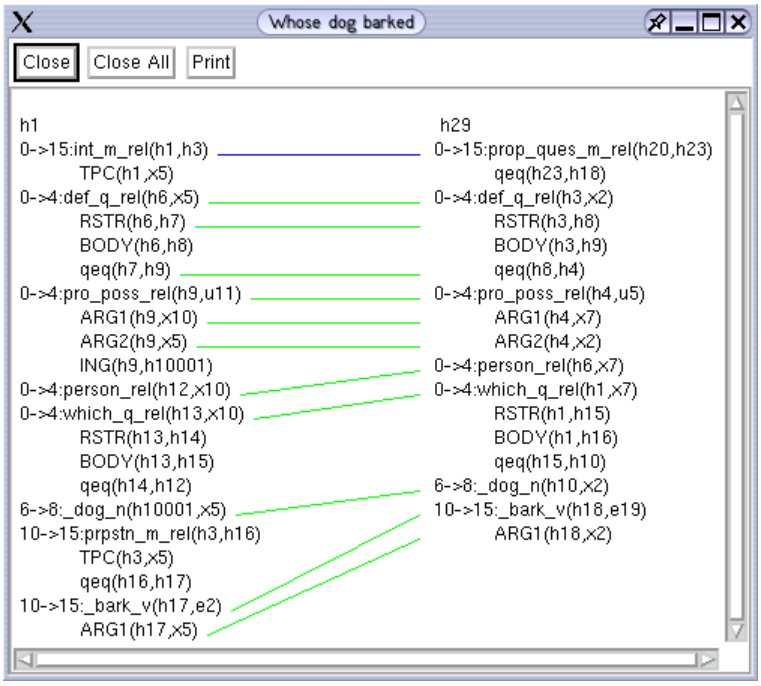

Figure 30: Comparison window for *Whose dog barked?*

RSTR, must remain. Similar problems are encountered with the interrogative determiner in *Whose dog barked?* in Figure 30.

A second issue comes from determiners that can occur as complete NPs. Figure 31 exemplifies this with the sentence *Some begin.* In such cases, the ERG introduces a **generic_*_rel** (here, **generic_nonpro_rel**) to denote the 'missing' noun entity in the normal determiner-noun semantics. RASP, on the other hand, never applies its NP/det_n1, since there is no N1, where this semantics would normally be introduced. Moreover, no alternative rule is applied that would indicate that this type of NP has been constructed (the word *some* is simply accepted straight into the S/np_vp rule), so there is nowhere where the ERG's semantics can be replicated, including the 'dummy' predicate.

4.11 Ellipsis

Elliptical constructions are particularly problematic. For some sentences, such as *Abrams could* (Figure 32), RASP can only produce a fragment parse, since it expects modals to always precede a main verb. Consequently, the resultant RMRS is likewise fragmented: note the lack of any **prpstn_m_rel** indicating a complete proposition. It does slightly better for *Browne tried to* (Figure 33) in that a complete parse tree is produced. However, *tried to* is simply interpreted as a complete VP, where *to* is treated as a particle and the V1/v_pp rule applied, because RASP has no information about the subcategorisation frames of the verb *try* and that the *to* is part of an elliptical infinitival clause. Thus, in both cases, the desired **ellipsis_rel** is never introduced by RASP.

4.12 Negation

At present, three mismatches are caused by negation e.g. *The dog couldn't bark* Figure 34). Firstly, there are characterisation mismatches for the **_could_v** and **neg_rel**: the ERG distinguishes between *couldn't* and *could not* so introduces both predicates from the characters *couldn't*, whereas RASP treats the two constructions as identical, translates the contracted version to *could not* then introduces the **_could_v** from *could* and the **neg_rel** from *not*. This difference in treatment results in the second mismatch: a difference in scoping. The ERG has the **neg_rel** scoping over the **_could_rel** and vice versa for RASP. Finally, the argument structures for the **neg_rel**s do not match since there is unresolved debate over the 'correct' semantics for negation: should there be a negation event?

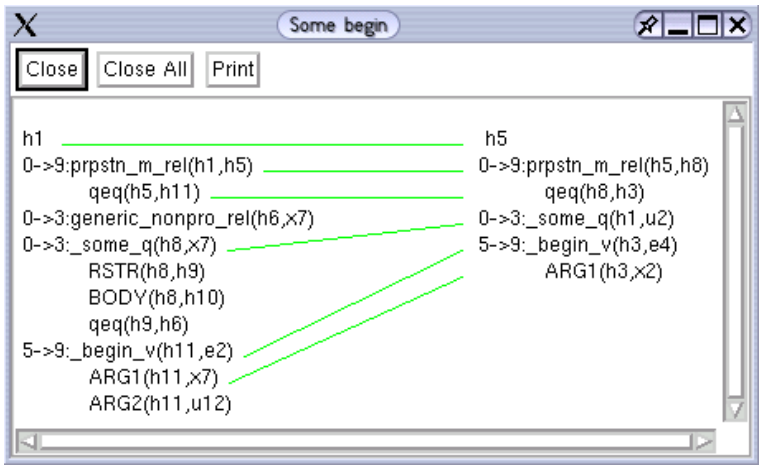

Figure 31: Comparison window for *Some begin*.

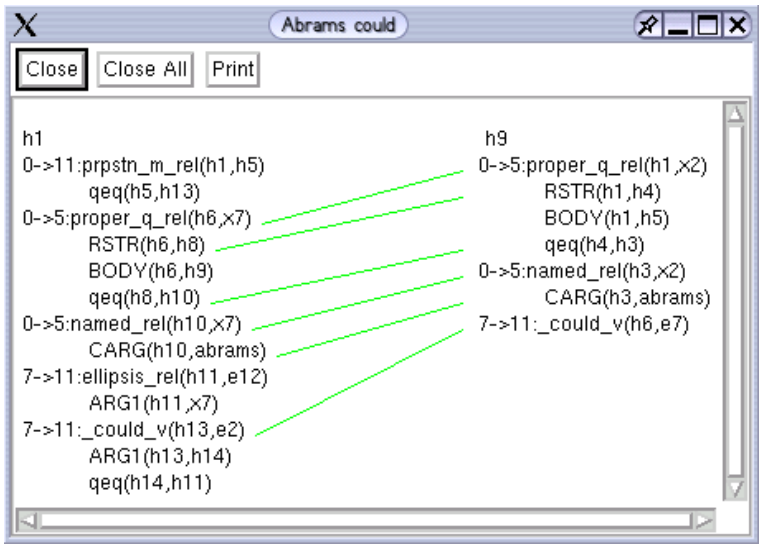

Figure 32: Comparison window for *Abrams could.*

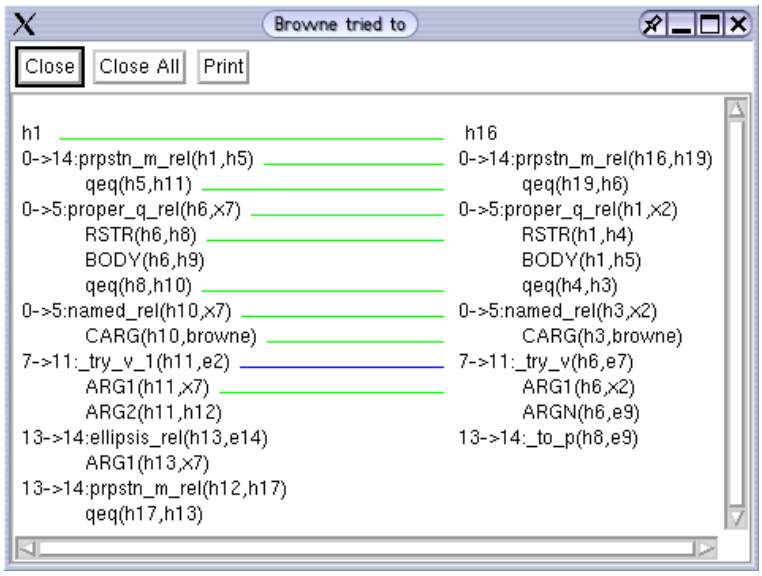

Figure 33: Comparison window for *Browne tried to.*

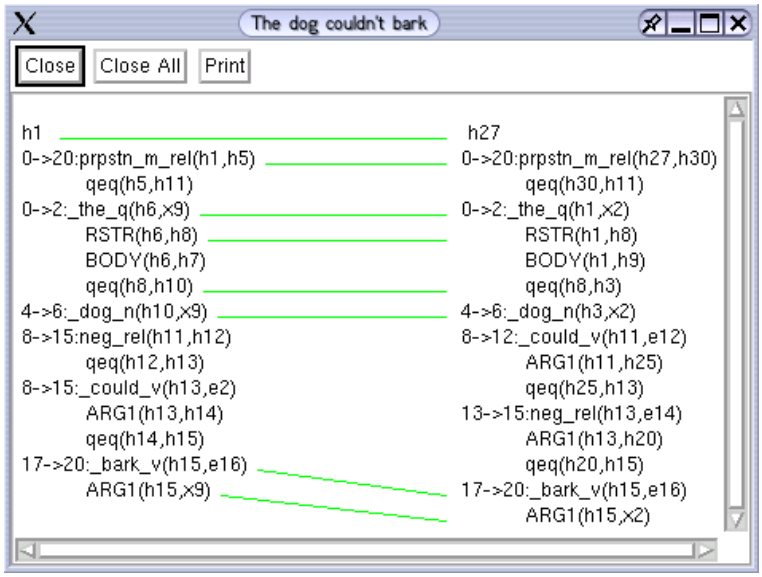

Figure 34: Comparison window for *The dog couldn't bark.*

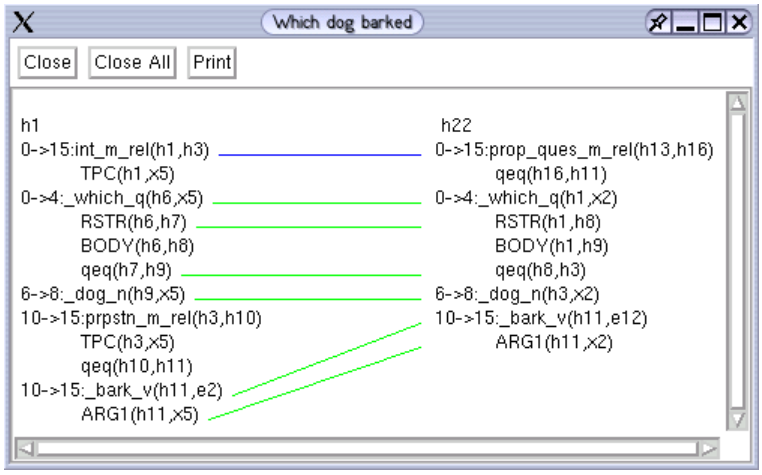

Figure 35: Comparison window for *Which dog barked?*

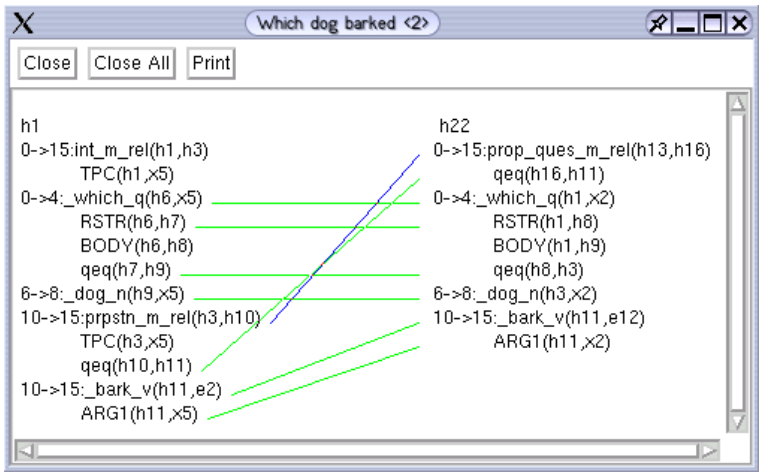

Figure 36: Comparison window for *Which dog barked?*

4.13 Interrogatives vs relative clauses

The interrogative *Which dog barked?* (Figure 35 and Figure 36) and the relative clause *which barked* (Figure 37) are both covered by RASP's S/np-wh_vp rule. The ERG produces a **prpstn_m_rel** attached to the **_bark_v** in both cases but also an overall **int_m_rel** for the interrogative. It is, thus, impossible for RASP to introduce this predicate only in the interrogative case. The solution to this problem is for the grammar rule to rather introduce a (single) **prop_ques_m_rel**, as this subsumes both the **prpstn_m_rel** and **int_m_rel**. Currently, the comparison functionality can only allow RASP's **prop_ques_m_rel** to correspond to one or the other of the ERG's (hence the two comparison windows for *Which dog barked?*) but this may be revised to allow for such cases of multiple correspondence.

4.14 Co-ordination

Figure 38 shows the semantics required for both co-ordination by an explicit conjunction (*and*) and by a comma. The former requires a lexical predicate (**_and_c**) and the latter, an **implicit_conj_rel**, both with arguments linking them to the appropriate entities. Although both cases are covered by RASP's NP/np_np-coord, the comma is an optional daughter in this rule. Thus, when the comma is present, the rule appears in the tree as NP/np_np-coord/+ (otherwise, NP/np_np-coord/-) and the conjunction semantics can be approximated by creating separate (XML) grammar rules for these cases and adding the **implicit_conj_rel** and its arguments in the '+' case and only the arguments for the **_and_c** (added by the appropriate lexical rule) in the '-' case. However, Figure 39 illustrates how

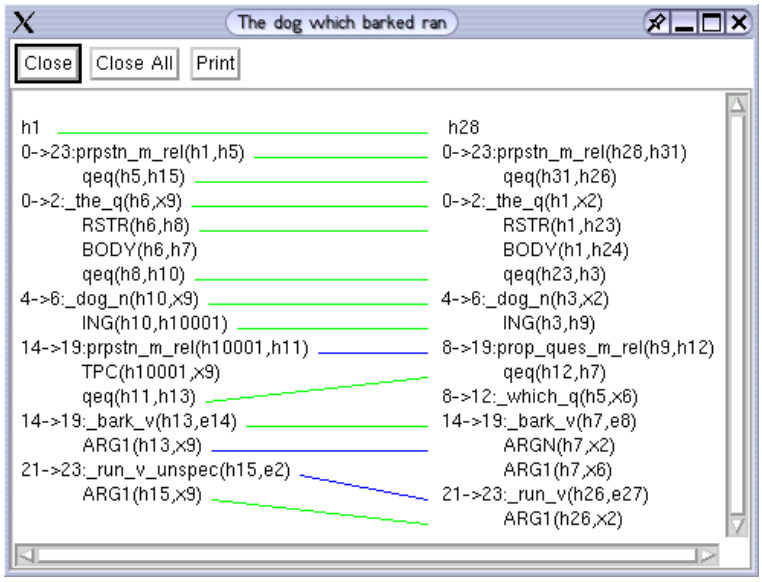

Figure 37: Comparison window for *The dog which barked ran.*

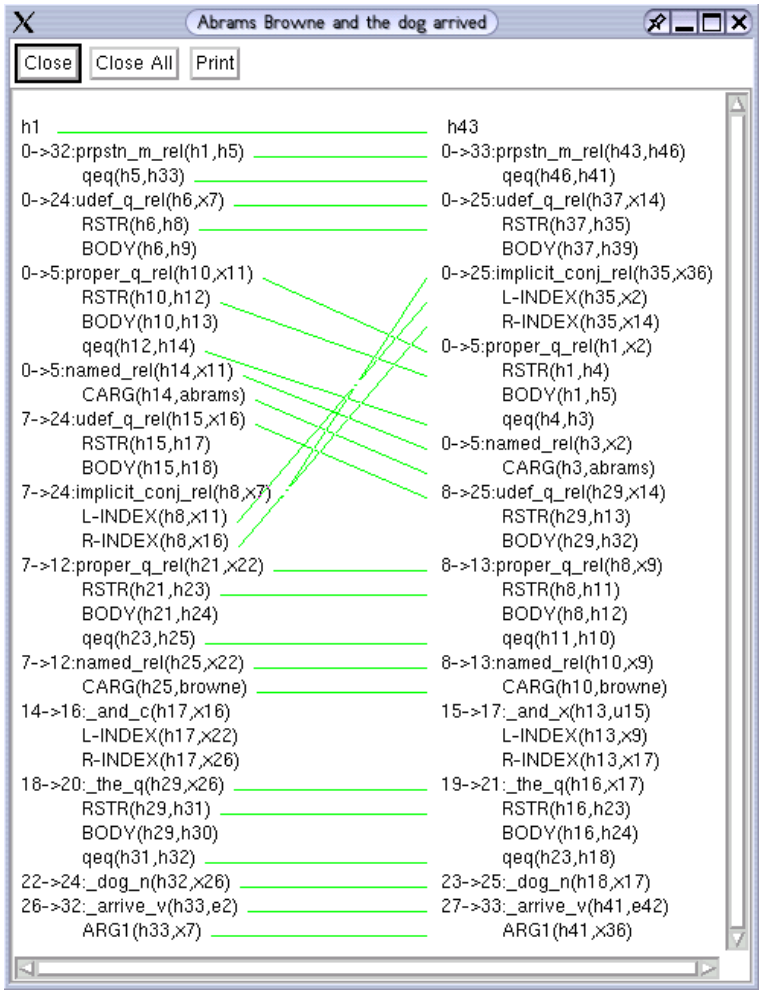

Figure 38: Comparison window for *Abrams, Browne and the dog arrived.*

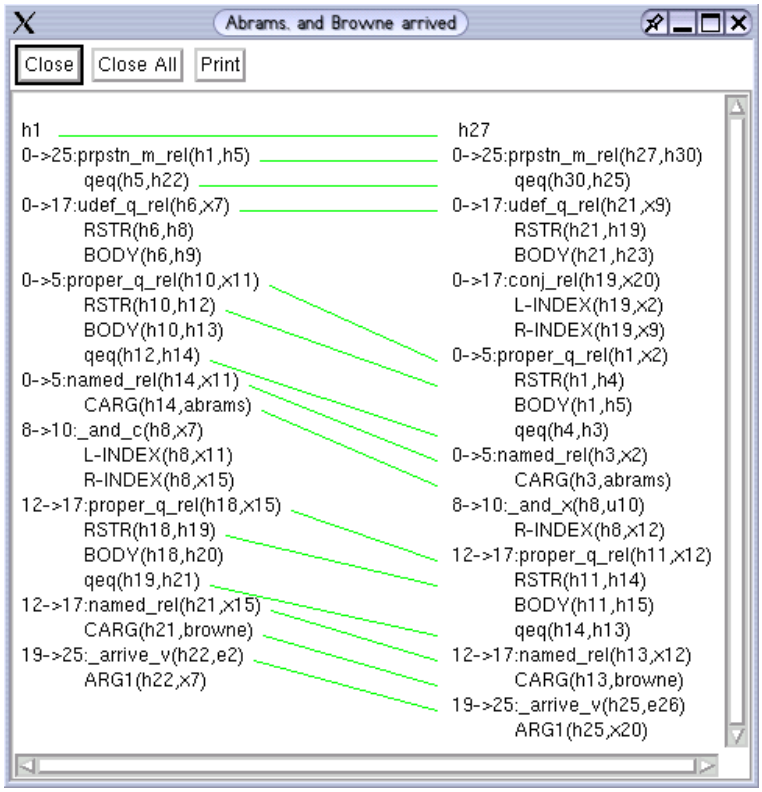

Figure 39: Comparison window for *Abrams, and Browne arrived.*

the Oxford comma results in an unwanted **implicit_conj_rel** being introduced alongside the necessary **_and_c**.

Notice also that, while the ERG's conjunction predicates have normal variable ARG0s, RASP's are underspecified. This is since the same lexical rule is applied to all conjunctions whether they co-ordinate NPs, VPs or sentences and, in the last two cases, the ARG0 is an event (Figure 40 and Figure 41, respectively).

VP co-ordination presents a further problem in that the co-ordinated verbs should share the same subject. However, since the allocation of subjects is handled by S/np_vp and this simply attaches an ARG1 to the main verb of the VP (i.e. the predicate in the VP attributed with the label), only one of the verb predicates are given the correct subject by this rule. The best RASP can do, at present, is to use the grammar rules that co-ordinate the VPs to introduce an ARG1 for the second verb and say it is *some* normal variable but, since this rule has no access to the sentence's NP, the correct variable cannot be found. Thus, while the ARG1 of **_bark_v** in Figure 40 is x2 (the dog), that of **_arrive_v** is a floating x26.

The same problem occurs with verbs that take progressive verb complements, as in *Abrams kept barking* (Figure 42). Again, the ERG successfully associates the same subject with *kept* and *barking*, whereas RASP can only access the **_keep_v** from S/np_vp and its V1/v_ing cannot access the NP. The solution to this problem would seem to be to use a second anchor and to replicate the main verb's subject for the anchored predicate. However, it may not be straightforward to implement such a scheme, since it would require the S/np_vp to introduce this second ARG1 only in *some* cases and the rules currently introduce semantics for all cases or do not introduce the semantics at all.

4.15 Infinitival verb subjects

A different take on the missing subject problem was initially apparent with verbs taking infinitival VP complements, although in this case an (imperfect) solution has been found. Consider the sentences *Abrams intended to bark* and *Abrams intended Browne to bark*, illustrated in Figure 43 and Figure 44, respectively. In both cases, the **_bark_v** requires an ARG1 but, in the former, it shares *Abrams* as the subject with the **_intend_v** and, in the latter, it is *Browne*. First appearances suggest that RASP's use of different grammar rules in these cases (V1/v_inf and V1/v_np_inf, respectively) could be used to correctly identify the subject. Thus, the V1/v_np_inf would make its

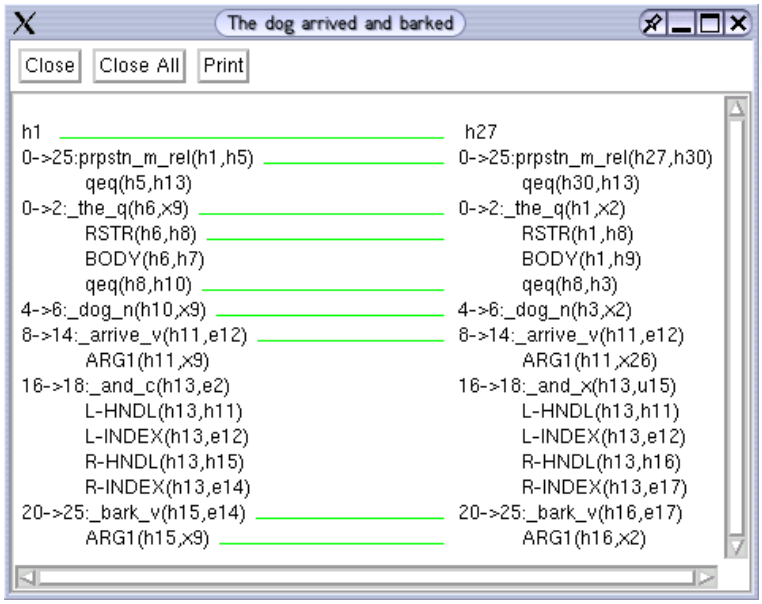

Figure 40: Comparison window for *The dog arrived and barked.*

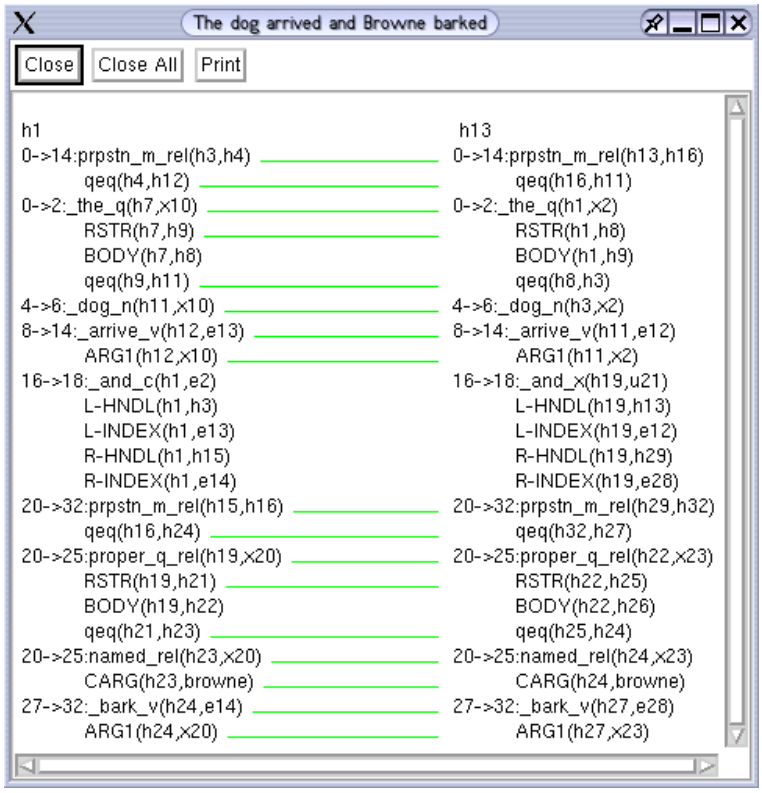

Figure 41: Comparison window for *The dog arrived and Browne barked.*

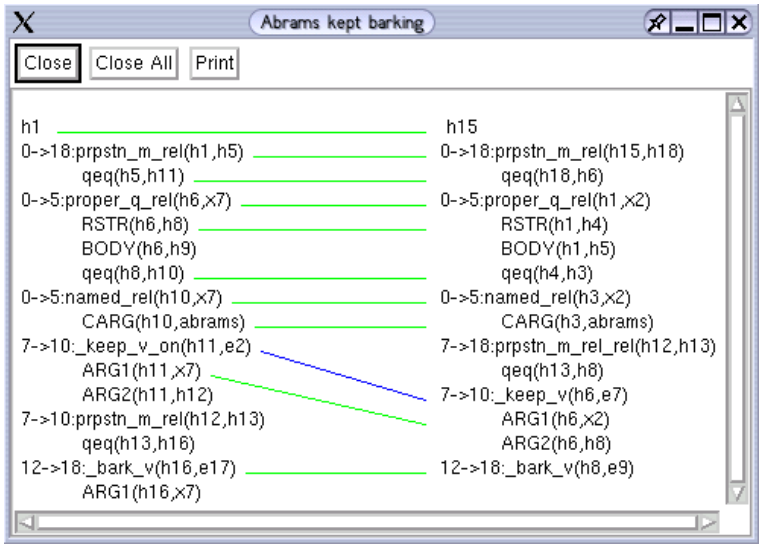

Figure 42: Comparison window for *Abrams kept barking.*

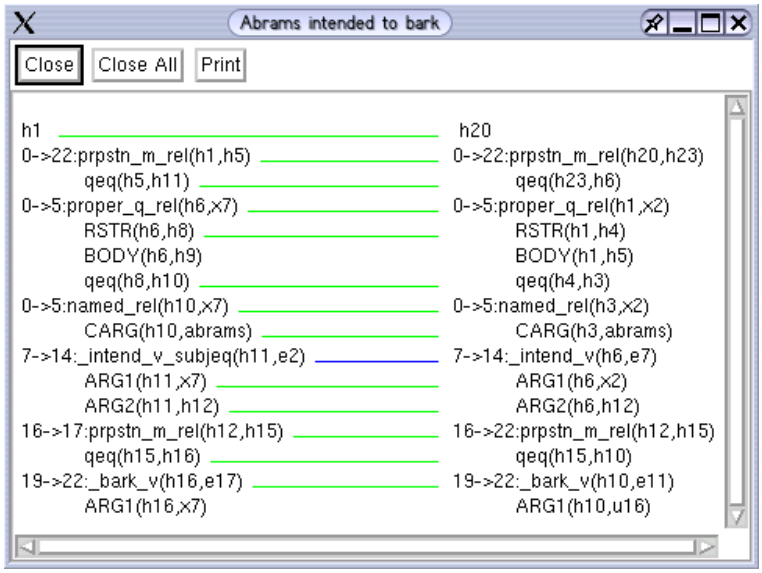

Figure 43: Comparison window for *Abrams intended to bark.*

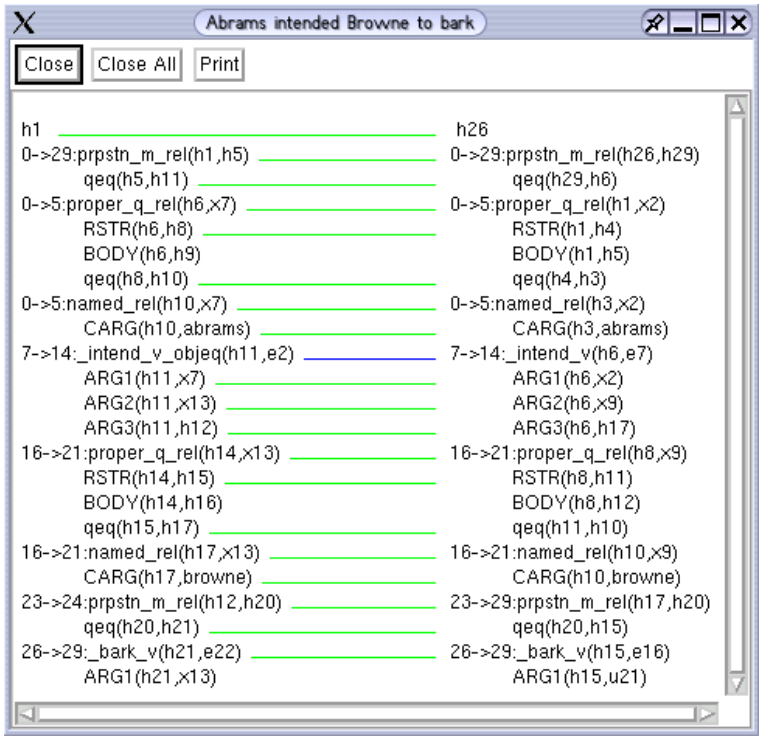

Figure 44: Comparison window for *Abrams intended Browne to bark.*

NP the ARG1 of the infinitival VP. This approach is flawed, though, since not all VPs with this sub-categorisation frame will have the NP as their infinitival VP's subject e.g. *Abrams promised Browne to bark* (Figure 45).

The preferred solution is to introduce an underspecified ARG1 for *all* infinitival VPs (i.e. in the V1/to_bse rule). Not only does this avoid the problem of sometimes getting the subject wrong (or just never adding a subject at all), it is also exactly the semantics required for some infinitival VPs, e.g. *The dog to chase is barking* (Figure 46).

4.16 Control vs raising verbs

Figure 43 also serves as an illustration for a separate problem. Here, *intended* is a control verb and the S/np_vp rule correctly picks out the NP as its subject. Raising verbs like *seems*, on the other hand, do not have a subject. However, the sentence *Abrams seems to bark* (Figure 47) is syntactically identical to *Abrams intended to bark* and the same RASP rules are applied. Thus, with no lexical information, RASP cannot distinguish between these types of verbs and the S/np_vp rule attaches an unwanted ARG1 to the **_seem_v**.

4.17 Subordinate clauses

Subordinate clauses generally produce two **prpstn_m_rel**s: one for the embedded sentence and one for the entire clause (Figure 48). RASP treats subordinating conjunctions as prepositions and, therefore, these clauses are covered by PP grammar rules. Since one particular rule applies to the subordinate clauses (and only to these constructions), the subordinate clause semantics can be introduced by that P1/p_s rule, including the SUBORD argument of the subordinating conjunction.

This solution is not ideal, however, because certain subordinating conjunctions produce slightly different semantics e.g. *whether* requires instead an **int_m_rel** overall, no SUBORD argument and an ARG2 linking the subordinate clause to the main clause via its head verb, as shown in Figure 49. Since RASP has a separate tag for *whether*, the required **int_m_rel** can replace the default **_whether_x** here but the usual P1/p_s rule is applied so an unwanted **prpstn_m_rel** and SUBORD argument are added, while the **_wonder_v**'s ARG2 is missing.

Furthermore, RASP's treatment of subordinating conjunctions as prepositions results in an unwanted ARG1 (the verb event) being attached to their predicates, as is required for true prepositions, by the V1/vp_pp rule.

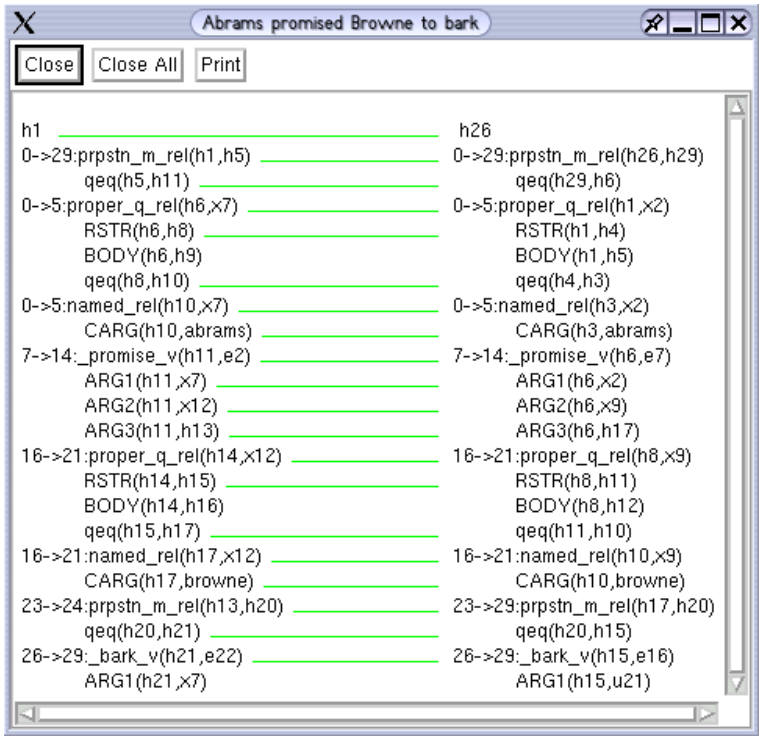

Figure 45: Comparison window for *Abrams promised Browne to bark.*

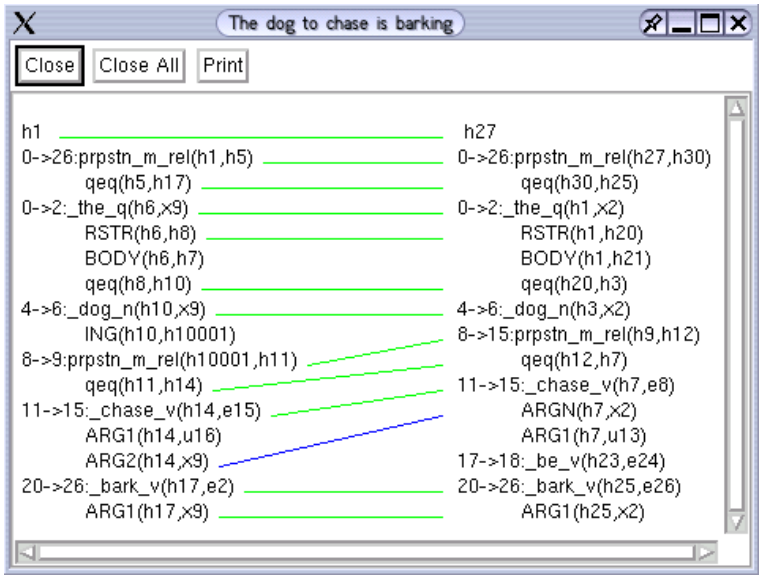

Figure 46: Comparison window for *The dog to chase is barking.*

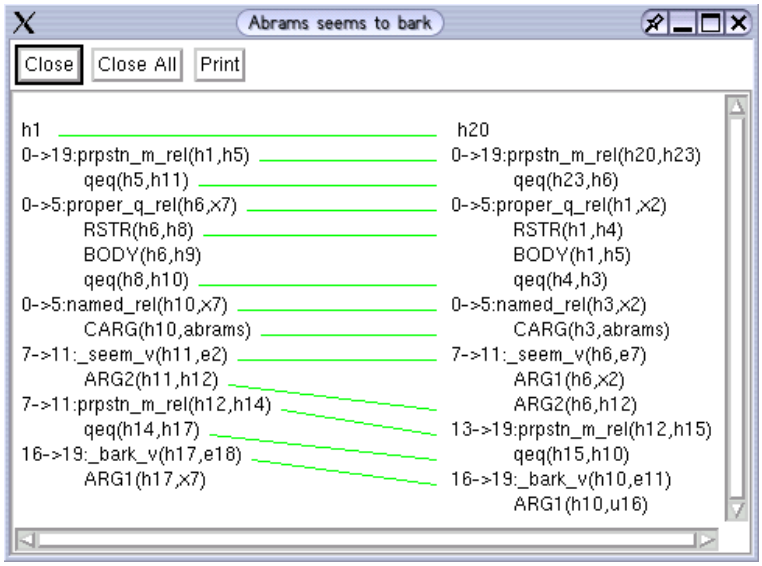

Figure 47: Comparison window for *Abrams seems to bark.*

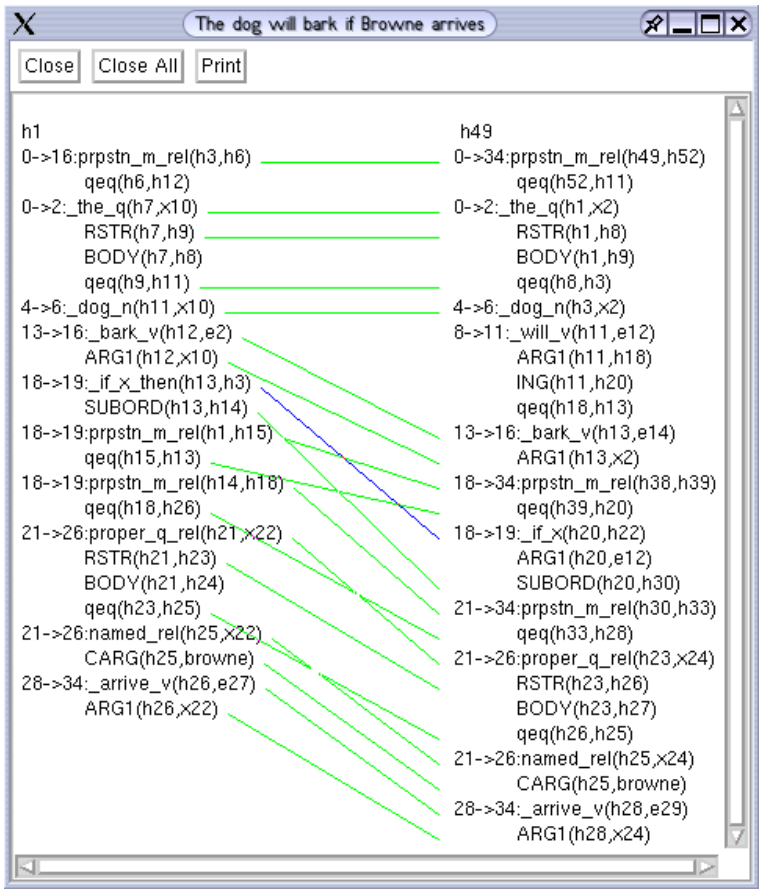

Figure 48: Comparison window for *The dog will bark if Browne arrives.*

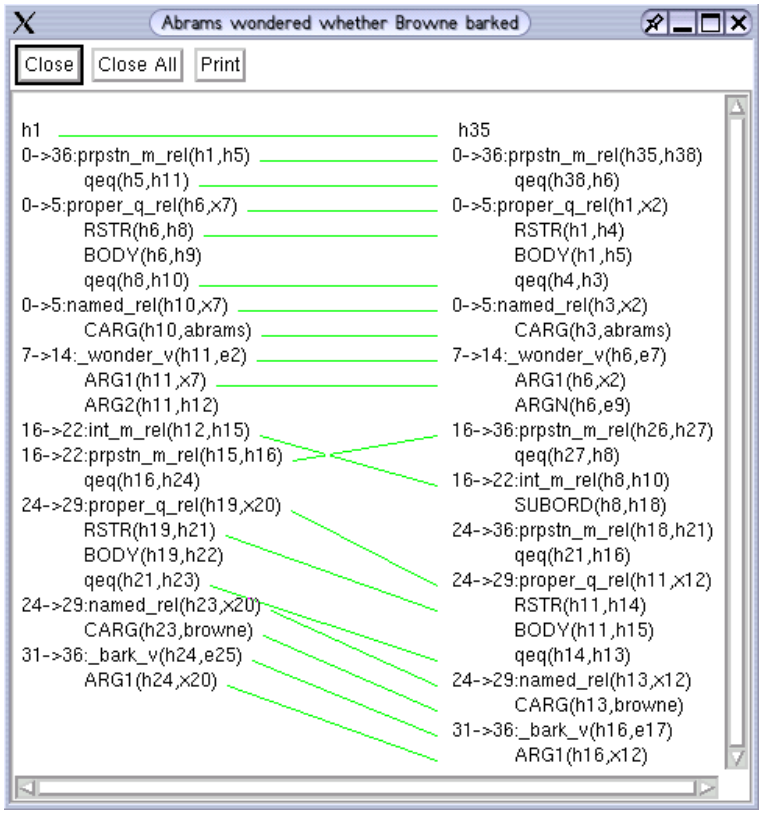

Figure 49: Comparison window for *Abrams wondered whether Browne barked.*

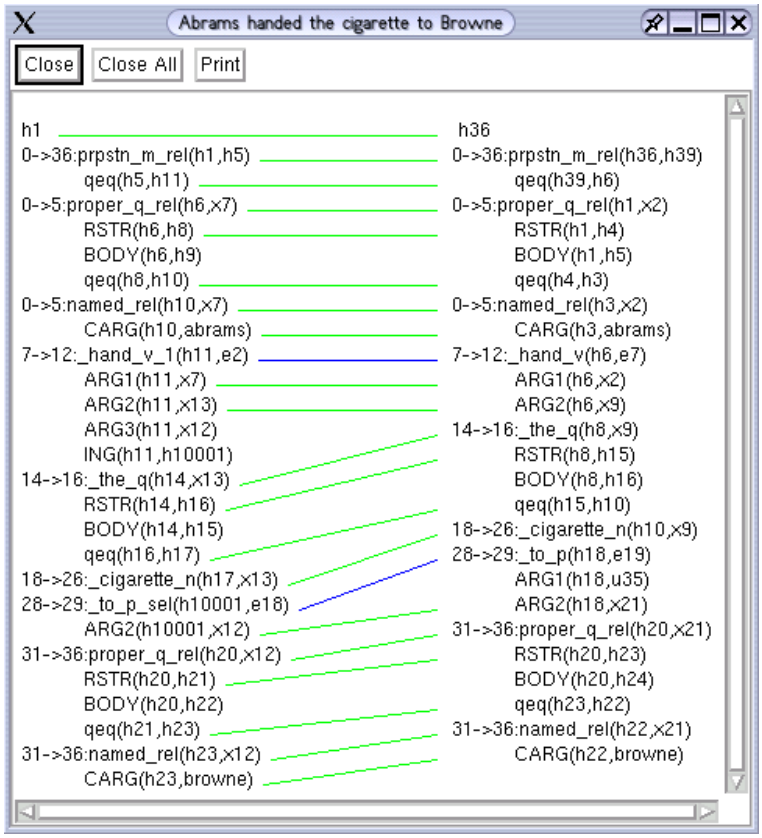

Figure 50: Comparison window for *Abrams handed the cigarette to Browne.*

4.18 PP-taking verbs

The same issue reoccurs with verbs that take PP complements. The ERG distinguishes between many types of PP-taking verbs and produces different semantics for them. For instance, the VPs in each of *Abrams handed the cigarette to Browne*, *Abrams put Browne in the garden* and *Abrams strikes Browne as young* (Figure 50, Figure 51 and Figure 52, respectively) are represented differently. In terms of RASP syntax, however, these VPs are identical and are each covered by its V1/v_np_pp rule. Thus it is impossible for RASP, with no lexical information, to know which semantics is required and, thus, none can be added by the rule.

4.19 "That-" clause subjects

Yet another instance of this issue is found in *that*- clauses. Figure 53 and Figure 54 show the RMRSs for *The dog that Browne chased barked* and *Abrams liked the idea that Browne could*, respectively. Both *that*- clauses are the S in RASP's N1/n_s rule but the ERG requires different semantics for the relationships between the N and the S. Since the relative clause case is likely to be the more common (N1/n_s also applies to e.g. *which*- relative clauses), its semantics are added in this rule, giving the **_bark_v** in Figure 54 a spurious ARGN (*the idea*) and having an ING linking **_idea_n** to the **prpstn_m_rel** for *Browne could bark* rather than an ARG1.

4.20 Incorrect argument placing

RASP's lack of lexical information is manifest in several other areas, including cases where predicate arguments are placed wrongly. Considering Figure 52 once more, RASP has the subject and direct object of the verb *strikes* confused in this application of V1/v_np_pp since, in most cases, the embedded NP would be the direct object (and the NP in the outer S/np_vp, the subject) and, thus, this is the semantics that is produced by these rules.

Similarly, Figure 55 and Figure 56 show *Browne's chasing of cats bothered Abrams* and *It bothered Abrams that Browne chased cats*, respectively; two subcategorisation frames for the verb *bother*. Whereas in the former,

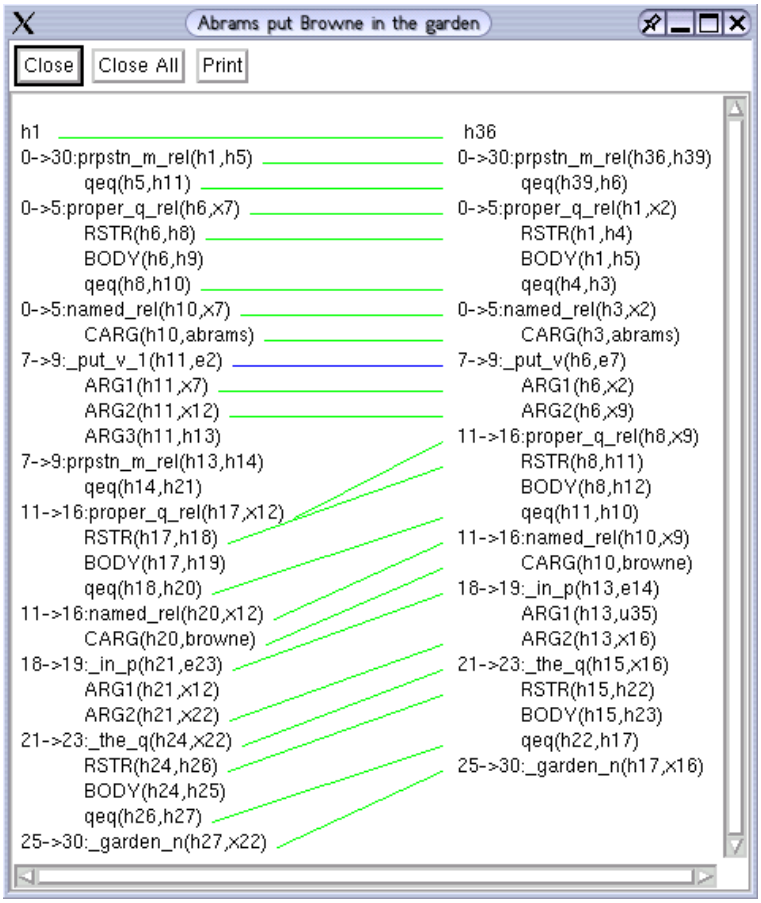

Figure 51: Comparison window for *Abrams put Browne in the garden.*

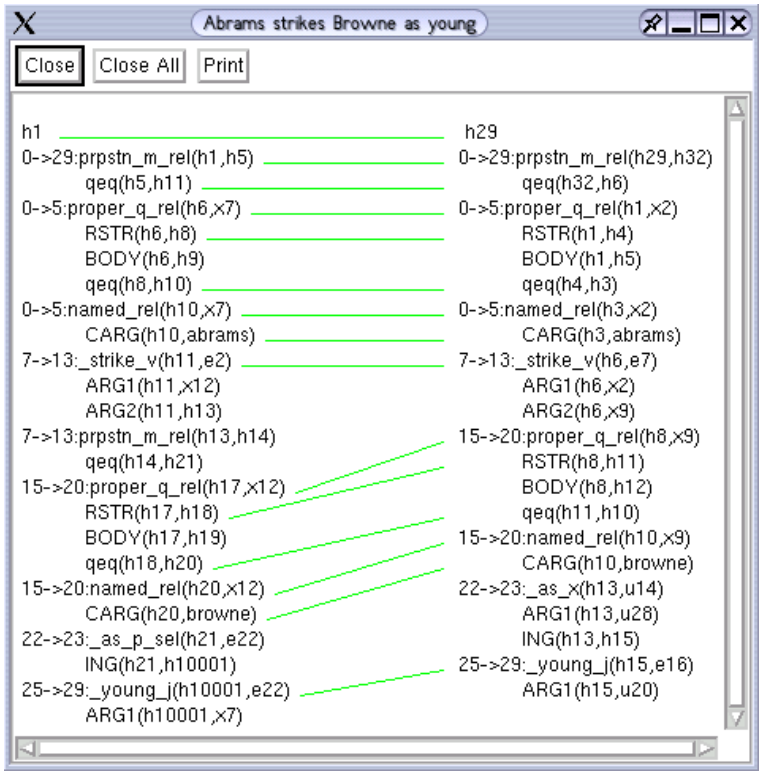

Figure 52: Comparison window for *Abrams strikes Browne as young.*

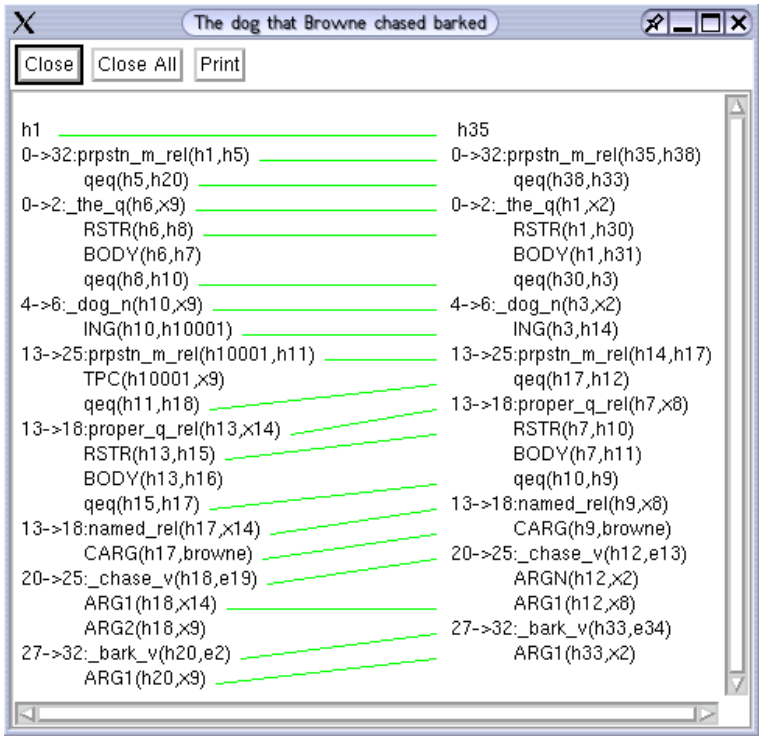

Figure 53: Comparison window for *The dog that Browne chased barked.*

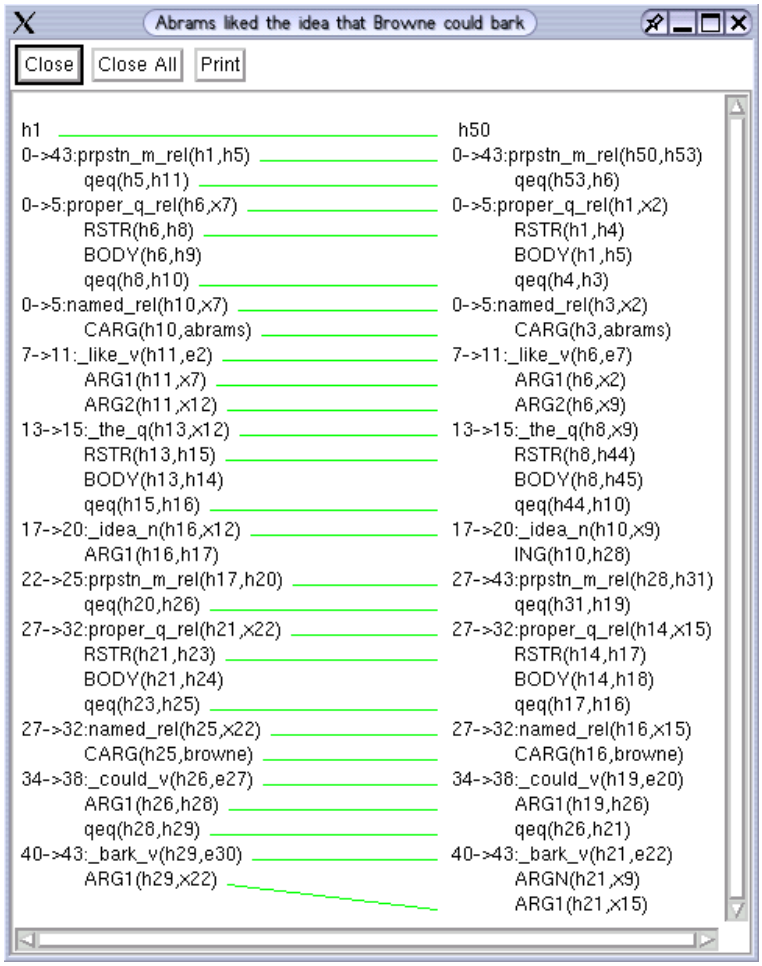

Figure 54: Comparison window for *Abrams liked the idea that Browne could.*

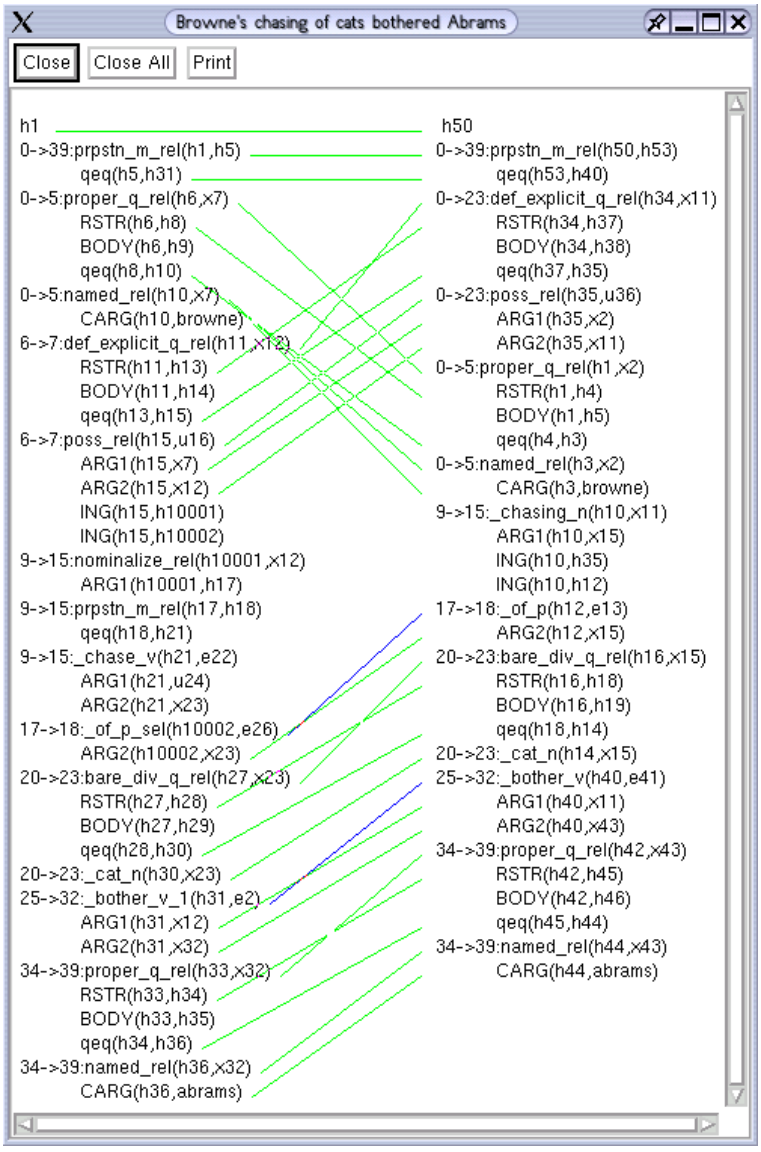

Figure 55: Comparison window for *Browne's chasing of cats bothered Abrams.*

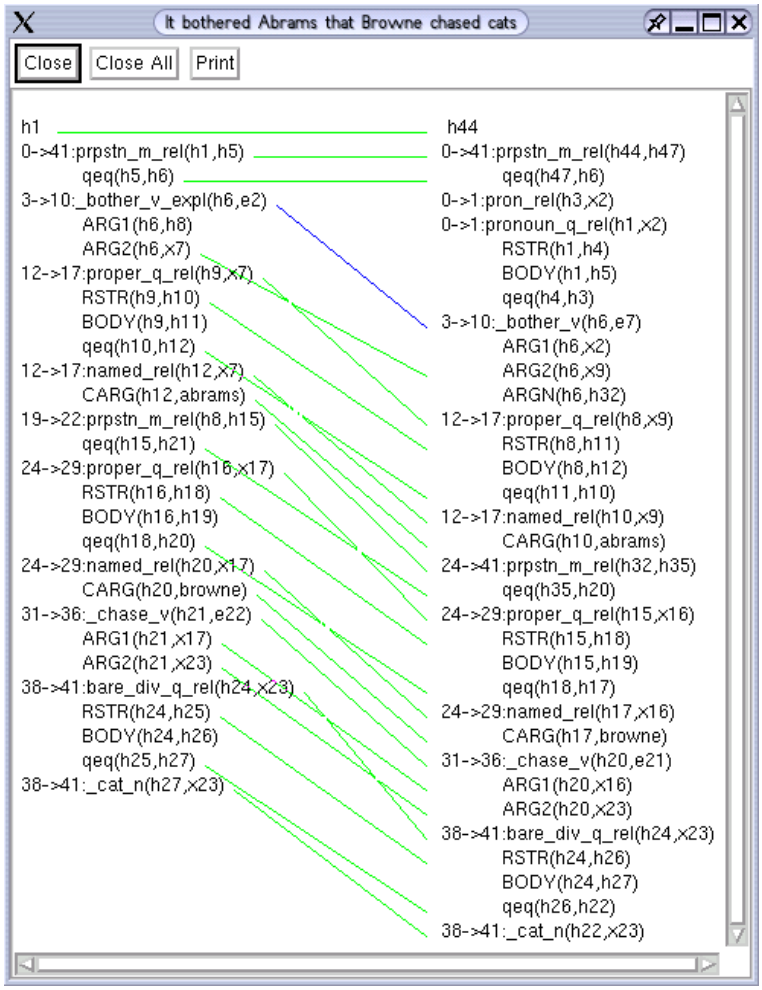

Figure 56: Comparison window for *It bothered Abrams that Browne chased cats.*

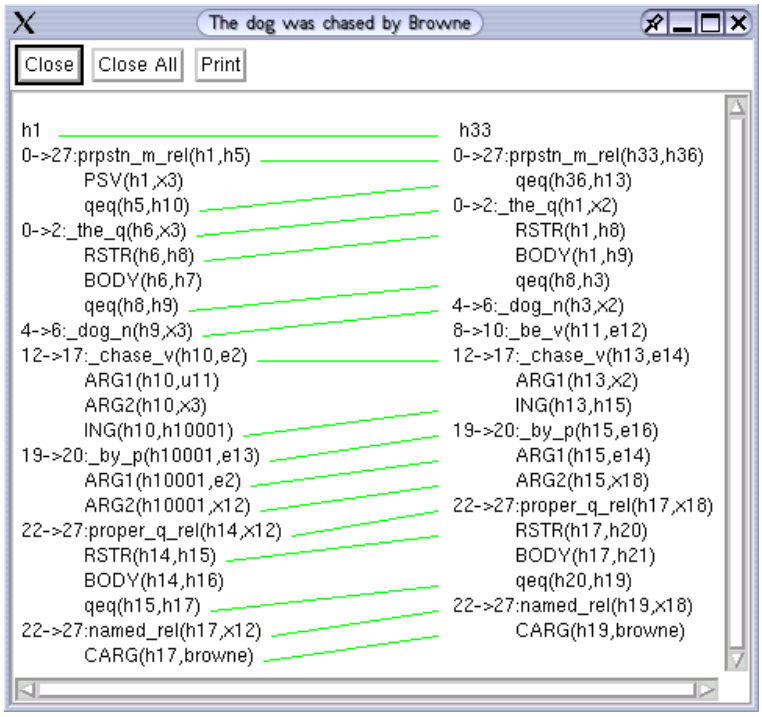

Figure 57: Comparison window for *The dog was chased by Browne.*

RASP correctly identifies *Abrams* as the direct object of the verb and the chasing of cats as its subject, in the latter (expletive) case, these ARGs are displaced by one due to RASP's incorrect selection of *it* as the subject.

Passive constructions are a further instance where RASP misplaces verb arguments. Figure 57 shows the comparison window for *The dog was chased by Browne.* Although the ERG cannot identify *Browne* as the subject of its **_chase_v**, it recognises the passive voice[8] and correctly labels *the dog* as the direct object, leaving the subject underspecified. RASP, on the other hand, treats the construction in the same way as an active one and picks out *the dog* as the subject. While this allocation of subject is done, as usual, by the S/np_vp rule, RASP's use of its V1/be_ppart rule for *was chased* is a reliable indicator of the passive voice and, therefore, it should be possible to overrule this default choice of subject whenever this rule is applied. (This would also signal that *was* is an auxiliary and its predicate should be removed.)

4.21 Wrong RASP parses

Despite RASP's robustness, there remain some phenomena that can prevent it producing the correct parse tree for a sentence (or even any parse at all). The sentence *Some bark*, for instance, is only interpreted as a fragment (a noun phrase, in this case) since *bark* is tagged as a noun. In *Browne squeezed the cat in*, the word *in* is erroneously treated as an abbreviation for *inch* (since it immediately precedes a full stop) and is therefore interpreted as part of a measure type NP with *the* and *cat*. It is not robust to phrasal adverbs, as in *Boys kind of suck* or *It is sort of cold*: in the first case, a fragment parse is produced with *boys kind* treated as noun-noun compound, *of* as an isolated preposition and *suck* as an imperative sentence. In the second, *sort* is treated as a noun and the N1/n_pp-of rule applied (meant for constructions like *jar of honey*). And, as mentioned previously, interrogatives and ellipsis are particularly problematic for RASP since it was not developed to deal with these sorts of constructions.

Without the correct syntax, it is impossible for RASP to get the semantics right.

[8]Note the ERG's PSV marker indicating passive voice: this is a device used in generating sentences from RMRS rather than a semantic component. Thus, no attempt has been made to introduce this into the RASP-RMRS.

5 Conclusions and further work

Having documented here the mismatches between ERG and RASP RMRSs, it would appear that they can be grouped into two broad categories. On the one hand, there are those cosmetic issues that generally do not mark any true incompatibility between the representations. Though they may result in irritating differences between the RMRSs, these differences are purely at the surface level and, in some cases, should be easily fixed. The ERG's formatting of CARGs and particular predicate names, for instance, should be straightforward to replicate.

The second type of issue presents more of a problem. These are products of the fundamental difference between deep and shallow processors and, as such, are irresolvable. The lack of lexical information inherent in RASP means that it simply cannot rely on 'knowing' words and their sub-categorisation frames and thus identifying how they interact with the rest of a sentence and the semantics that results. It must, instead, guess from the syntactic structure of the sentence. As detailed, though, the same syntactic structure will underlie constructions with very different semantics and, hence, RASP is limited in the guesses it can safely make.

Having said that, some grammar rules were found to be applied to constructions with the same semantic structure in the vast majority of cases. Whether it is worthwhile making the guess and risking getting the semantics wrong some of the time probably depends on the end application. Future work could include an investigation into which aspects of RMRS generation it might be useful to make user-configurable.

In the course of this work, a number of RASP's grammar rules and POS tags were studied and conversion rules developed accordingly[9]. Yet only the surface has been scraped: the grammar in use contains almost 600 phrase structure rules; the tagset, around 150 tags. Some proportion of those rules still to be investigated will be ones for rare constructions, some will be near duplicates of investigated rules that do not exhibit anything new (from a semantic point-of-view) and some will simply apply to constructions that do not affect the semantics of a sentence at all. Nevertheless, the semantic test suite is certainly not comprehensive and remains to be extended to more fully demonstrate the semantic phenomena in English. In doing this, more of RASP's rules and tags should be exercised and, hopefully, more of its potential for generating useful RMRSs achieved.

Finally, in documenting only problems, this paper has necessarily painted a somewhat bleak picture of ERG and RASP RMRS compatibility. It is worth noting, however, that a number of the semantic test suite examples failed to make it into this discussion by virtue of being 'perfect'. Discounting characterisation mismatches and subsumptive matches, as discussed, 46 of the 124 (originally 107) sentences are perfectly compatible. A further proportion exhibited only problems that occurred elswhere and, as has been demonstrated, many of these problems should be easily remedied. The conclusion of this work, therefore, must be to verify the potential for integrating deep and shallow processing through the RMRS representation and to motivate further work needed to confirm its benefits.

References

Briscoe, E. & Carroll, J. (2002), Robust accurate statistical annotation of general text, *in* 'Proceedings of the 3rd International Conference on Language Resources and Evaluation'. *citeseer.ist.psu.edu/564499.html.

Copestake, A. (2002), *Implementing Typed Feature Structure Grammars*, CSLI Stanford, California.

Copestake, A. (2004), Robust minimal recursion semantics.

Copestake, A. & Flickinger, D. (2000), An open source grammar development environment and broad-coverage English grammar using HPSG, *in* 'Proceedings of the 2nd International Conference on Language Resources and Evaluation'. *citeseer.ist.psu.edu/copestake00open.html.

Copestake, A., Lascarides, A. & Flickinger, D. (2001), An algebra for semantic construction in constraint-based grammars, *in* 'Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics', pp. 132–139. *citeseer.ist.psu.edu/456077.html.

DeepThought (2002). *www.project-deepthought.net/.

[9]The XML rule files from which this work began were CVS version 1.10. The versions of the resources used throughout were RASP grammar tsg14.1-t8qa, ERG 11-Jun-04 and LKB 2004/06/22.