Blocking

Ted Briscoe & Ann Copestake
Computer Laboratory, University of Cambridge, Pembroke Street, Cambridge, CB2 3QG, UK
ejb / aac @cl.cam.ac.uk

Alex Lascarides
Human Communication Research Centre, University of Edinburgh
2 Buccleuch Place, Edinburgh, EH8 9LW, UK
alex@cogsci.ed.ac.uk

1 Introduction

A major motivation for the introduction of default inheritance mechanisms into theories of lexical organisation has been to account for the prevalence of the family of phenomena variously described as blocking (Aronoff, 1976:43), the elsewhere condition (Kiparsky, 1973), or preemption by synonymy (Clark & Clark, 1979:798). In Copestake & Briscoe (1991) we argued that productive processes of sense extension also undergo the same process, suggesting that an integrated account of lexical semantic and morphological processes must allow for blocking. In this paper, we review extant accounts which follow from theories of lexical organisation based on default inheritance, such as Paradigmatic Morphology (Calder, 1989), DATR (Evans & Gazdar, 1989), ELU (Russell et al., 1991, in press), Word Grammar (Hudson, 1990; Fraser & Hudson, 1992), or the LKB (Copestake 1992; this volume; Copestake et al., in press). We argue that these theories fail to capture the full complexity of even the simplest cases of blocking and sketch a more adequate framework, based on a nonmonotonic logic that incorporates more powerful mechanisms for resolving conflict among defeasible knowledge resources (Commonsense Entailment, Asher & Morreau, 1991). Finally, we explore the similarities and differences between various phenomena which have been intuitively felt to be cases of blocking within this formal framework, and discuss the manner in which such processes might interact with more general interpretative strategies during language comprehension. Our presentation is necessarily brief and rather informal; we are primarily concerned to point out the potential advantages using a more expressive default logic for remedying some of the inadequacies of current theories of lexical description.

2 Data

The type of phenomenon which is typically discussed under the rubric of blocking involves inflectional morphological irregularities; for example, the past participle of *walk* is *walked* and we can describe this regular morphological operation (roughly) in terms of a rule of concatenation $\langle$stem$\rangle$+*ed*[1]. However, this rule should not be applied either to non-verbal stems (e.g. *beast, *beasted*) or to the subset of verbal stems which exhibit irregular behaviour (e.g. *sink, sunk, *sinked; bring, brought, *bringed; make, made, *maked*). These observations are captured in default inheritance based approaches to the lexicon by specifying that the rule in question applies to the subclass of verb stems, but allowing subclasses of this class to override its application (where in the limit a subclass may consist of a specific word). Thus, the rule is interpreted as a default within a class and more specific subclasses can prevent its application via stipulative statements. In the case where a rule does not apply to a subclass or single member of the class over which it is defined the rule is said to be blocked. There are cases though, where both a morphologically irregular and regular form coexist, so that blocking cannot be treated as an absolute property of lexical organisation (e.g. *dream, dreamt, dreamed; burn, burnt, burned*).

[1] Throughout this paper we will ignore complications of morphographemics and morphophonemics. For proposals on how such complications can be dealt with within the accounts of the lexicon we consider see e.g. Calder (1989), Cahill (1990), Russell et al. (1991) or Bird (1992).

Proposals for accounting for such exceptions to blocking have not treated blocking itself as a default, but have either, in effect, denied that any such general principle is at work in the lexicon by stipulating its effect on a word-by-word basis (Fraser & Hudson, 1992) or treated those verbs which allow both irregular and irregular forms as a further subclass of 'dual-class' verbs which specify an irregular variant but also inherit the regular pattern (Russell et al., 1991). Even in the clear cases of blocking the productive forms surface as common errors in the language of first and second language learners and, though we may not be happy to accept *sleeped* as a word of English, we are able to interpret utterances such as *I sleeped well* with considerably more ease than many other forms of putative ungrammaticality.

Blocking is not restricted to inflectional morphology, but appears to be pervasive throughout the lexicon: we can see its effects in derivational morphology, conversion and metonymic processes of sense extension. It appears to account for some of the semi-productivity of most synchronic, generative lexical rules (see Bauer, 1983:84f for a discussion of other factors restricting productivity). However, blocking rarely if ever appears to be an absolute constraint on word (sense) formation. For example, there is a derivational rule of noun formation from adjectives by suffixation of *+ity* (e.g. *curious, curiosity; grammatical, grammaticality*). This rule is generally blocked by the existence of a non-derived synonymous noun (e.g. *glorious, glory, *gloriousity; tropical, tropic(s), *tropicality*). The similar rule of noun formation from adjectives by suffixation of *+ness* is apparently not blocked or not blocked to the same extent under these circumstances (e.g. *?curiousness, ?grammaticalness, ?gloriousness, ?tropicalness*). Nevertheless, there is a sense of markedness or awkwardness about such examples which is not present with forms which do not undergo noun formation with *+ity* or compete with a non-derived synonymous form (e.g. *awkwardness, markedness, weakness, kindness*). Aronoff (1976) argues that noun formation with *+ness*, in contrast to *+ity*, is fully productive, hence the lack of blocking with *+ness* forms. This account does not address the difference in acceptability between those *+ness* forms which are preempted by a synonymous underived form or one formed with *+ity*.

In general, even clearly productive derivational rules can be, at least partially, blocked. Rappaport & Levin (1990) argue convincingly that noun formation via suffixation with *+er* on verbs is highly productive and creates a meaning change where the derived noun denotes the class of objects which can serve as the verb's subject (or external argument in their terminology). Thus *teacher* denotes the agent of *teach*, whilst *opener* can denote the agent or more usually instrument of *open*. Bolinger (1975:109) cites *stealer* as an example of a form blocked by *thief*. In fact, *stealer* is fairly clearly blocked in contexts where its meaning is synonymous with *thief*, but may be used in contexts where its productively derived meaning has been specialised or modified; for example, Bauer (1983:87-8) points out that *stealer* can be used metonymically as in Shakespeare's *the ten stealers* (fingers) or when the denotation is restricted via mention of the object (e.g. *stealer of hearts / fast sports cars*).

Nominalisation with *+er* also illustrates another apparent type of blocking which cannot be described as preemption by synonymy but which can be characterised as preemption by equality at some level of representation. Rappaport & Levin (1990) point out that the class of words which allow middle formation (e.g. *John stuck the poster (to the wall), The poster stuck easily (to the wall)*) tend to form nominalisations with *+er* which denote stereotypical objects, rather than subjects (e.g. *sticker, broiler, bestseller*)[2]. It is difficult for these nominalisations to also denote subjects and, where this can be forced, the meaning is inherently specialised; for example, *sticker* can be used to refer to a person who is determined, as in *He is a sticker when it comes to crosswords*, but is very odd in *?He works as a sticker of posters*. Productive nominalisation with *+er* can also be blocked by the existence of a non-productive sense of the derived form;

[2]Whether these cases can be subsumed under a general rule based on subject / external argument nominalisation is controversial, but it seems clear that this (sub-)pattern is less productive in terms of the restrictions imposed on the denotation of the derived nominal.

for example, *stationer* means a person who sells stationary and not a person who stations, say, troops, and *banker (at Barclays)* refers to a person who works in a (senior) position in a bank, and not to one who banks his money there. In these cases, what is blocked is generation of an orthographically or phonologically identical word form with the productive meaning associated with the relevant derivational rule.

In addition to rules of derivation there are similarly productive processes of conversion and sense extension which also appear to be subject to blocking. In Briscoe & Copestake (1991) and Copestake & Briscoe (1991) we argued in detail that derivation, coercion and sense extension should be treated within the same general framework. Conversion processes are often highly productive as the host of noun-verb conversions such as *hoover* illustrate. However, derived forms such as *arrival* do not undergo this process because of the existence of *arrive*. Briscoe & Copestake (1991), Copestake & Briscoe (1991) and Ostler & Atkins (1991) argue that metonymic and metaphorical processes of sense extension must be treated as lexical rules which can undergo blocking; for example, the productive sense extension of 'grinding' (see e.g. Pelletier & Schubert, 1986), whereby a count noun denoting an individuated object becomes a mass noun denoting a substance derived from that object, when applied to animals typically denotes the meat or edible flesh of that animal (e.g. *lamb, haddock, rabbit*). However, this process is blocked by the existence of a synonymous form (e.g. *pork, pig; beef, cow*). In these cases, blocking does not prevent the more general process of grinding but makes the typical interpretation highly marked.

It seems that blocking is pervasive, or at least that use of the term in the literature is pervasive. However, all the cases we have considered appear to share some properties: blocking itself is rarely if ever absolute, and blocking is preemption of rule application by equality at some level of lexical representation. In what follows, we explore the hypothesis that a uniform account of blocking can be provided which subsumes both the morphological and lexical semantic cases.

3 Inadequacies of Current Approaches

Extant accounts of blocking in recent theories of lexical organisation based on default inheritance have mostly treated blocking as an absolute principle of lexical organisation. That is, if some derivational behaviour is given for a class, specifying some distinct behaviour for a subclass *always* has the effect of preventing the more general behaviour from applying. The sole exception that we are aware of is the theory of Word Grammar (Hudson, 1990; Fraser & Hudson, 1992) which treats blocking as the exception rather than the rule. Thus, in Word Grammar, by default, the lexical rules specified for a class apply to all subclasses, even those that have been stated to have exceptional behaviour — the more specific information augments, rather than overrides, the more general information. This approach would be natural if it were the case that the lexicon appeared to tolerate the rule-governed production of synonymous or homophonous words, but the cases considered in §2 suggest that this is not so. As a result, it is necessary to stipulate that blocking does occur with forms such as *sleeped*, whilst cases like *dreamed* or *curiousness* are predicted to be the norm rather than the exception. At the very least, this obliges the lexicographer to include many idiosyncratic exception statements in a lexical description (e.g. Cahill, 1992). In addition, there are theoretical reasons for believing blocking to be a general principle of lexical organisation: a lexicon which did not impose some such constraint would be in danger of becoming dysfunctional with language change and development, resulting in too many distinct word forms conveying the same meaning or too many distinct meanings associated with homophonous forms.

All other theories of lexical organisation which account for blocking impose it as an absolute principle. We will illustrate some approaches to blocking within default inheritance based accounts by discussing how inflectional morphology may be described within the LKB system and

LEX_SIGN: $\langle\mathrm{ORTH}\rangle\ (\mathrm{orth}) \wedge \langle\mathrm{SYN}\rangle(\mathrm{syn}) \wedge \langle\mathrm{SEM}\rangle(\mathrm{sem})$

Verb: $\langle\mathrm{SYN}\rangle\langle\mathrm{CAT}\rangle(\mathrm{v}) \wedge \langle\mathrm{SYN}\rangle\langle\mathrm{VFORM}\rangle(\mathrm{base})$

Past-Verb: $\langle\mathrm{SYN}\rangle\langle\mathrm{VFORM}\rangle(\mathrm{pastprt}) \wedge \langle\mathrm{ORTH}\rangle\langle\mathrm{SUFFIX}\rangle(+\mathrm{ed})$

Irreg-Past-Verb: $\langle\mathrm{ORTH}\rangle\langle\mathrm{SUFFIX}\rangle(+\mathrm{t})$

walk:	Verb
walked:	Past-Verb
slept:	Irreg-Past-Verb
dreamed:	Past-Verb
dreamt:	Irreg-Past-Verb

Figure 1: LKB psort hierarchy and lexical entries

compare this to some other formalisms, concentrating on those which are similar to the LKB, in that lexical information is represented as feature structures (FSs). The LKB, in effect, incorporates two inheritance mechanisms — the type system, based on Carpenter (1992), enforces constraints on FSs which cannot be overridden, and psorts provide a default inheritance mechanism. Psorts are named FSs which are used in the LKB description language in a manner similar to the use of templates in PATR-II. Psorts may be specifically defined, or be ordinary lexical entries or rules, since these are also represented as FSs. In the LKB, psorts may be combined using either ordinary unification or default unification, the latter giving default inheritance (Copestake et al., in press). In this case the FS description may specify information which conflicts with that of the psort and which will override it.

For ease of comparison with other untyped systems, the examples which follow are based on the use of inheritance via the psort mechanism, rather than the type mechanism. We will simply assume a most general type **lex-sign** which will constrain all the FSs representing lexical signs to have the three features ORTH, SYN and SEM. Thus, if we specify the hierarchy of psorts illustrated in Figure 1, `Past-Verb` and `Irreg-Past-Verb` introduce incompatible values for the feature SUFFIX so that FSs for verbs in the latter class will not be subsumed by those for `Past-Verb`.[3]

There are clearly some infelicities in this approach, since we are forced to associate one and only one type of FS with each type or psort and therefore cannot capture the morphological relationships between inflectional variants of a verb. For this reason, the LKB includes a notion of lexical rule which can encode essentially arbitrary mappings between FSs, and it would be more natural to treat past participle formation as a rule of this type. One possible treatment is shown in Figure 2. Here the inflectional variants are generated by rule application but we need to introduce a new feature MORPH on lexical signs, in order to specify the appropriate irregular forms. We also must ensure that only the correct rules apply and this is done by typing the stem values as either **regular** or **irregular**. `Verb`, by default, specifies that the value of $\langle\mathrm{ORTH}\rangle\langle\mathrm{STEM}\rangle$ has to be a subtype of **regular**, but this is overridden by the psort `Irregbase`.

A lexical rule of past participle formation, shown in Figure 2, maps regular base verbs to regular past participle forms. In the LKB, lexical rules are themselves FSs, and thus can be treated as psorts and inherit information from each other, thus the lexical rule for irregular

[3] For consistency we use a notation based on Blackburn (1992a,b) to encode constraints on FSs. We discuss this further in §3.1

LEX_SIGN: ⟨ORTH⟩ (orth) ∧ ⟨SYN⟩(syn) ∧ ⟨SEM⟩(sem)

Verb: ⟨SYN⟩⟨CAT⟩(v) ∧ ⟨SYN⟩⟨VFORM⟩(base) ∧ ⟨ORTH⟩⟨STEM⟩(regular)

Irregbase: ⟨MORPH⟩⟨PASTPRT⟩(+t) ∧ ⟨ORTH⟩⟨STEM⟩(irregular)

Dualpast: ⟨ORTH⟩⟨STEM⟩(reg/irreg)

Past_formation: [1 : [⟨SYN⟩⟨VFORM⟩(base) ∧ ⟨ORTH⟩⟨STEM⟩(regular)]
0 : [⟨SYN⟩⟨VFORM⟩(pastprt) ∧ ⟨ORTH⟩⟨SUFFIX⟩(+ed)]]

Irreg_suffixation: [1 : [⟨MORPH⟩⟨PASTPRT⟩(α) ∧ ⟨ORTH⟩⟨STEM⟩(irregular)]
0 : [⟨ORTH⟩⟨SUFFIX⟩(α)]]

walk: Verb
sleep: Irregbase
dream: Dualpast

Figure 2: LKB lexical rules (α is a variable indicating reentrancy)

suffixation default inherits from the standard rule. A lexical rule is applicable to a FS if its input half, that is the section of the lexical rule found by following the path ⟨1⟩, is unifiable with that FS. The result of lexical rule application is indicated by the path ⟨0⟩. In Figure 2, we assume that unspecified parts of the FS are the same in the input and output. Thus the effect of the rule Past_formation is to add the suffix *ed* and to change the value of ⟨SYN⟩⟨VFORM⟩. The lexical entry for *walk* simply inherits from `Verb` and *sleep* inherits from `Irregbase`. The regular lexical rule applies to *walk* but the irregular one does not, because the types **regular** and **irregular** are disjoint. Note that we have to explicitly specify the stem of *walk* as **regular** to avoid the incorrect lexical rule applying. In order to allow the regular past participle form of *dream* to be produced, we have to ensure that both lexical rules can apply, and we do this by specifying that the value for ⟨ORTH⟩⟨STEM⟩ for the psort `Dualpast` is the type **reg/irreg** which subsumes both **regular** and **irreg**. Thus *dream* has both an irregular suffix and a regular one.[4]

Although this approach is better than the first, it is still far from perfect. The lexical rule is not directly blocked by the existence of a 'competing' form, instead we have associated the existence of the irregular form with a class which also specifies that the stem is irregular, and it is this specification that blocks the application of the lexical rule. This, in turn, is overridden for dual class verbs such as *dream*. Thus we have made blocking non-absolute at the cost of increasing the complexity of the representation. A third option is to give the suffixes themselves lexical entries which are combined with the base forms in more or less the same way as word forms are combined in the grammar. This approach however suffers from much the same problem with respect to blocking.

[4]The value of ⟨ORTH⟩⟨STEM⟩ has actually to be constrained to be equal to **reg/irreg** rather than just subsumed by it, in order that `Dualpast` can inherit from `Irregstem`, without ⟨ORTH⟩⟨STEM⟩ being specified to be **irreg**. The LKB incorporates a notion of an equality or mutual subsumption specification as well as ordinary value specification, which allows this to work.

Verb: $\langle \mathrm{SYN} \rangle \langle \mathrm{CAT} \rangle (\mathrm{v}) \wedge \langle \mathrm{MORPH} \rangle \langle \mathrm{PASTPRT} \rangle (+\mathrm{ed})$
$|$
$\langle \mathrm{VFORM} \rangle (\mathrm{base})$
$|$
$\langle \mathrm{VFORM} \rangle (\mathrm{pastprt}) \wedge \langle \mathrm{ORTH} \rangle \langle \mathrm{SUFFIX} \rangle (\alpha) \wedge \langle \mathrm{MORPH} \rangle \langle \mathrm{PASTPRT} \rangle (\alpha)$

Dual-past: $|$
$\langle \mathrm{VFORM} \rangle (\mathrm{pastprt}) \wedge \langle \mathrm{ORTH} \rangle \langle \mathrm{SUFFIX} \rangle (+\mathrm{t})$
$|$

walk: Verb

sleep: Verb
$\langle \mathrm{MORPH} \rangle \langle \mathrm{PASTPRT} \rangle (+\mathrm{t})$

dream: Dual-past Verb

Figure 3: ELU-like fragment

In contrast to the LKB, ELU and DATR represent inflectional paradigms by inheritance, without introducing the concept of a lexical rule. ELU makes use of a variant set mechanism which causes a verbal class or paradigm to generate separate FSs for each inflectional variant. Constraints on FSs are specified as path equations and hierarchical class membership is used to enforce default inheritance so that blocking is absolute whenever a subclass introduces a constraint which clashes with that introduced by a superclass. Therefore, Russell et al. (1991) introduce a separate class of dual past verbs and treat *dream* as a member of both this and the regular class. A slight variant of their analysis is reproduced in Figure 3.[5] Vertical bars separating constraints indicate ELU variant sets, that is, disjunctive constraints on (distinct) FSs. The empty variant set in `Dual-past` allows `dream` to inherit all the information from `Verb` as well as the variant specified by `Dual-past`. The order of inheritance in this example is dependent on textual order — that is `dream` inherits first from `Dual-past` and then from `Verb`. An equivalent approach would be to make `Dual-past` inherit from `Verb`. This would be slightly more elegant, because it would remove the need to specify that the lexical entries inherited from two classes, and avoid the dependency on textual order. However, in both cases the same form of redundancy is present in the analysis as in the LKB description considered above: the irregular past specification has been made in several parts of the lexical description in order to circumvent the absolute effects of blocking.

DATR (Evans & Gazdar, 1989) also employs a notion of hierarchical class membership to specify lexical entries and to control default inheritance. However, unlike the LKB and ELU, DATR is a language which is intended to be specific to lexical representation and to be usable by any grammar which can be encoded in terms of attributes and values. DATR does not specify or manipulate FSs directly — DATR queries result in the values of paths at particular nodes being returned. There are essentially two components — a monotonic component which explicitly specified values for paths and a nonmonotonic operation of path closure which can intuitively be thought of as 'filling in the gaps' to give values for all node/path pairs not defined in the theory. However, despite the differences between DATR and the other theories, similar redundancy arises as in the analyses sketched above when we attempt to account in a straightforward way for alternative forms such as *dreamed* and *dreamt*, since the specification of a value for a node

[5] We have modified their notation and analysis to make it more consistent with Blackburn (1992a,b) and the LKB descriptions above.

blocks any alternative inherited value.

To summarise, the inelegancy and redundancy in these descriptions arises from the fact that each is forced to treat blocking as an absolute constraint enforced via default inheritance. The undesirable effects of this are circumvented by introducing additional classes which are otherwise unmotivated. Less restricted approaches such as those of Daelemans (1987) or de Smedt (1984) might, in principle, be able to use additional object-oriented techniques to achieve a less redundant lexical description, by allowing the option of accumulating rather than overriding inherited information, for example. However, this is difficult to evaluate in view of the open-ended nature of these techniques. Furthermore, it is far from clear that any inheritance based treatment of blocking can allow for the examples discussed in the previous section where the lexical item responsible for the blocking of the productive process is morphologically unrelated (e.g. *stealer/thief*, *pig/pork*). This would imply that we would have to structure the lexicon so that, for example, *thief* was related to *steal* by some form of the process of *+er* nominalisation. Clearly there is some semantic relationship, but requiring that a relationship between the lexical items as a whole be specified directly as though it were a variant of the productive process is very unappealing.

4 A New Framework

In this section, we outline a novel approach to lexical description in which we utilise a default logic as a constraint language on possible lexical entries specified as FSs. (For a detailed description of constraint-based languages in linguistic description, see Shieber (1992).) Within this framework we believe that it is possible to characterise blocking more satisfactorily. The default logic that we utilise is the propositional variant of Commonsense Entailment (CE, Asher & Morreau, 1991). This logic is appropriate because we can encode FSs using nominal modal propositional logic (Blackburn, 1992a,b); CE supports the patterns of default reasoning required to characterise blocking; and the propositional variant is decidable which is one crucial (and minimal) requirement for a theory of lexical description.

4.1 A Language for describing FSs

Blackburn (1992a,b) demonstrates that attribute value structures expressed as directed graphs are, in effect, languages of propositional modal logic in which attributes are modal operators and values are propositional variables. Furthermore, he proves that the subset of attribute value structures that can be expressed as directed acyclic graphs (DAGs) incorporating re-entrancy are characterised by the nominal propositional modal logic (NPL). This is a modal logic augmented with an atomic propositional variable sort called *nominals*, where a nominal is constrained to be true on a unique node. Blackburn (1992b) discusses why modal logics are particularly natural ways of describing FSs. In effect, the standard notations for describing FSs, such as AVMs, assume an internal view: than describing what happens at arbitrary nodes, the graph structure is described in terms of transitions from some particular node. Thus these notations are equivalent to a language where all quantifiers are bounded and the richness of the variable binding machinery of the first order languages is unnecessary. Modal operators and AVMs both precisely capture this restriction. Blackburn's results extend those of Kaspar & Rounds (1990) and others concerning the logical characterisation of unification-based formalisms utilising DAGs (e.g. Shieber, 1986).

In what follows, we develop a theory in which lexical signs are encoded as FSs; that is DAGs incorporating re-entrancy. And we utilise a notational variant of NPL to describe FSs. An example of a lexical entry represented as a FS in AVM notation is given in Figure 4. We will mostly refer to such FSs as lexical signs and assume that all such signs are minimally specified for ORTH, SYN and SEM (e.g. Pollard & Sag, 1987).

$$\begin{bmatrix} \text{ORTH} = \begin{bmatrix} \text{STEM} = \textbf{sleep} \end{bmatrix} \\ \text{SYN} = \begin{bmatrix} \text{CAT} = \textbf{verb} \\ \text{SUBJ} = \begin{bmatrix} \text{SYN} = \begin{bmatrix} \text{CAT} = \textbf{np} \end{bmatrix} \\ \text{SEM} = \boxed{\text{i}}\ \textbf{obj} \end{bmatrix} \\ \text{VFORM} = \textbf{base} \end{bmatrix} \\ \text{SEM} = \begin{bmatrix} \text{PRED} = \textbf{sleep} \\ \text{ARG1} = \boxed{\text{i}} \end{bmatrix} \end{bmatrix}$$

Figure 4: FS for *sleep*

Descriptions in NPL can be partial in the sense that more than one FS may satisfy a set of NPL formulas; we will use a sorted version of NPL, in order to achieve the required fine-grainedness of descriptions, in which a partial specificity ordering induces a lattice over propositional variables from which we can derive a subsumption relation (Mellish, 1988). The minimal model which satisfies a lexical description will correspond to the set of well-formed FSs or lexical signs according to that description. The sortal axioms in (1) define some useful sorts and values for verbal features.

(1) a $vform \equiv base \sqcup finite \sqcup infinite \sqcup pastprt \sqcup pass \sqcup presprt$
b $suffix \equiv +ed \sqcup +t \sqcup ...$
c $stem \equiv vstem \sqcup nstem \sqcup ...$
d $vstem \equiv walk \sqcup dream \sqcup sleep \sqcup ...$
e $irregvstem \equiv dream \sqcup sleep \sqcup ...$
f $dualvstem \equiv dream \sqcup ...$
g $cat \equiv verb \sqcup noun \sqcup ...$

A partial description of the class of FSs that we have been assuming for verbs is given in (2).

(2) $\langle \text{SYN} \rangle (\langle \text{CAT} \rangle(\text{verb}) \wedge \langle \text{VFORM} \rangle(\text{vform})) \wedge \langle \text{ORTH} \rangle \langle \text{STEM} \rangle(\text{vstem})$

Modal operators representing attributes are written in upper case between angle brackets, propositional variables are lower case, and round brackets, delimiting the scope of operators, are mostly suppressed where this is unambiguous. This description is satisfied by any of the FSs containing values for CAT, VFORM and STEM subsumed by the propositional variables 'verb', 'vform' and 'vstem', such as that in Figure 4. We can define the closure of a lexical description as the set of FSs which satisfy all theorems of that description. An algorithm for computing (membership of) this set can be defined in terms of substitutions of atomic propositional variables (such as 'verb' in the sort system defined in (1)) guided by the subsumption relation. Whether the closure of a lexical description remains finite will depend, firstly, on the finiteness of the sort system and, secondly, on the manner in which we restrict the inference mechanisms available via the constraint language.

In general, to properly constrain the set of lexical signs compatible with a description, it will be necessary to specify re-entrancy or structure sharing within and between lexical signs. Blackburn (1992a) uses nominals, denoted by alphabetical indices, to describe re-entrant FSs. Thus, the FS in Figure 4 specifies that ARG1 of the predicate 'sleep' is the same as the value of SEM under SUBJ. A constraint enforcing this equivalence is given in (3).

(3) $\langle \text{SEM} \rangle \langle \text{ARG1} \rangle(\text{i}) \wedge \langle \text{SYN} \rangle \langle \text{SUBJ} \rangle \langle \text{SEM} \rangle(\text{i})$

We will use NPL to model the lexicon as a whole, including the relationships between individual lexical signs. We will employ a notion of lexical rule similar to that presented in Pollard & Sag (1987:209f) in which such rules are treated as conditional assertions relating one (basic) lexical sign to another (derived) lexical sign. (This notion of lexical rule is essentially equivalent in expressive power to that utilised in the LKB, allowing arbitrary relationships between FSs to be specified.) We represent this in terms of a structure where derived signs 'fan out' from the basic signs, using a conventionally-named set of modal operators, LR1, LR2 ... LRn, in order to denote the transitions described by the different lexical rules. In NPL, we can express the fact that conjunctive constraints must hold for a single lexical sign by naming the superordinate node in the DAG corresponding to the appropriate FS. For example, (4) illustrates a simple lexical rule for past participle formation expressed in NPL: here i, j and k are metavariables ranging over nominals. Models corresponding to descriptions which include lexical rules may include more than one FS to which the rule has been applied.

(4) $$(i \rightarrow (\langle \textsc{sem}\rangle(\mathrm{j}) \wedge \langle \textsc{orth}\rangle\langle \textsc{stem}\rangle(\mathrm{k}) \wedge \langle \textsc{syn}\rangle\langle \textsc{vform}\rangle(\mathrm{base})))$$
$$\rightarrow$$
$$(i \rightarrow$$
$$(\langle \mathrm{LR1}\rangle((\langle \textsc{syn}\rangle\langle \textsc{vform}\rangle(\mathrm{pastprt}) \wedge \langle \textsc{sem}\rangle(\mathrm{j}) \wedge \langle \textsc{orth}\rangle((\langle \textsc{stem}\rangle(\mathrm{k}) \wedge \langle \textsc{suffix}\rangle(+\mathrm{ed})))))$$

In this framework, we are treating lexical rules as constraints on connected FSs containing one basic and further derived lexical entries connected by a distinguished attribute naming the relevant rule. This notation is verbose, so we will represent lexical rules using more familiar AVM notation in which boxed values translate as nominals and standard AVM formatting conventions are used to indicate the scope of modal operators, conjunction, and so forth (Blackburn, 1992a). Rather than explicitly indicating the conventional modal operators, LR1 etc, we will use the syntax $\rightarrow_{LR1}$ etc to indicate that the implied feature structure is connected by an LR operator. We will also omit the explicit specification of the outermost nominal (i) in the AVM diagrams; thus (4) will become (5).

(5) $$\begin{bmatrix} \mathrm{ORTH} = \begin{bmatrix} \mathrm{STEM} = \boxed{\mathrm{k}} \end{bmatrix} \\ \mathrm{SYN} = \begin{bmatrix} \mathrm{CAT} = \mathit{verb} \\ \mathrm{VFORM} = \mathit{base} \end{bmatrix} \\ \mathrm{SEM} = \boxed{\mathrm{j}} \end{bmatrix} \rightarrow_{LR1} \begin{bmatrix} \mathrm{ORTH} = \begin{bmatrix} \mathrm{STEM} = \boxed{\mathrm{k}} \\ \mathrm{SUFFIX} = +\mathit{ed} \end{bmatrix} \\ \mathrm{SYN} = \begin{bmatrix} \mathrm{CAT} = \mathit{verb} \\ \mathrm{VFORM} = \mathit{pastprt} \end{bmatrix} \\ \mathrm{SEM} = \boxed{\mathrm{j}} \end{bmatrix}$$

4.2 Default Logic as a Constraint Language

The lexical rule of past participle formation in (5) is far too strong since it does not hold of all (English) verbs and would therefore produce many linguistically unmotivated FSs. The approach taken in theories of lexical description based on default inheritance is, firstly, to restrict the domain of application of such a constraint to a subclass of verbs and, secondly, to allow the constraint to be overridden on stipulated further sub-classes or cases. We can restrict the application of a rule such as (5) to an appropriate subclass via the sort system and we can express its default nature by relaxing $\rightarrow$ to default implication ($>$), as in (6) ($\phi > \psi$ is to be read as *If* ϕ *then by default,* ψ).

(6) $$\begin{bmatrix} \mathrm{ORTH} = \begin{bmatrix} \mathrm{STEM} = \boxed{\mathrm{k}} \end{bmatrix} \\ \mathrm{SYN} = \begin{bmatrix} \mathrm{CAT} = \mathit{verb} \\ \mathrm{VFORM} = \mathit{base} \end{bmatrix} \\ \mathrm{SEM} = \boxed{\mathrm{j}} \end{bmatrix} >_{LR1} \begin{bmatrix} \mathrm{ORTH} = \begin{bmatrix} \mathrm{STEM} = \boxed{\mathrm{k}} \\ \mathrm{SUFFIX} = +\mathit{ed} \end{bmatrix} \\ \mathrm{SYN} = \begin{bmatrix} \mathrm{CAT} = \mathit{verb} \\ \mathrm{VFORM} = \mathit{pastprt} \end{bmatrix} \\ \mathrm{SEM} = \boxed{\mathrm{j}} \end{bmatrix}$$

Default logics incorporate a rule of defeasible modus ponens (DMP: $\phi > \psi, \phi \mathrel{|\!\approx} \psi$, where $\mathrel{|\!\approx}$ stands for nonmonotonic validity) which ensures that within the class of verbs *vstem*, (6) will apply unless the result is inconsistent with the premises. We cannot, though, prevent the application of (6) to irregular verbs merely by adding the more specific rule in (7).

$$
(7)\quad \begin{bmatrix} \text{ORTH} = \begin{bmatrix} \text{STEM} = \textit{irregvstem} \wedge \boxed{k} \end{bmatrix} \\ \text{SYN} = \begin{bmatrix} \text{CAT} = \textit{verb} \\ \text{VFORM} = \textit{base} \end{bmatrix} \\ \text{SEM} = \boxed{j} \end{bmatrix} >_{LR2} \begin{bmatrix} \text{ORTH} = \begin{bmatrix} \text{STEM} = \boxed{k} \\ \text{SUFFIX} = +t \end{bmatrix} \\ \text{SYN} = \begin{bmatrix} \text{CAT} = \textit{verb} \\ \text{VFORM} = \textit{pastprt} \end{bmatrix} \\ \text{SEM} = \boxed{j} \end{bmatrix}
$$

This is because as things stand, there is no statement in the premises asserting that the consequents of (6) and (7) don't usually both hold in any given knowledge base (KB). So no inconsistency arises through the application of DMP to both (6) and (7) (which we can represent schematically as $\phi > \psi, \chi > \zeta, \phi, \chi \approx\!\!| \ \psi, \zeta$). Therefore, as the system stands we do not incorporate blocking. And, in fact, this is the result that we want because, although we intuitively understand these values for SUFFIX to be in conflict, the existence of forms such as *dreamed* and *dreamt* tells us that we do not want this conflict to be as strong as logical inconsistency. In order to create a conflict, we can include a further constraint (8), which represents the effect of blocking in irregular past participle formation.

(8) $\langle \text{LR2} \rangle(\langle \text{ORTH} \rangle(\langle \text{STEM} \rangle(\text{irregvstem}) \wedge \langle \text{SUFFIX} \rangle(+\text{t}))) > \neg\langle \text{LR1} \rangle(\langle \text{ORTH} \rangle(\langle \text{STEM} \rangle(\text{irregvstem}) \wedge \langle \text{SUFFIX} \rangle(+\text{ed})))$

(8) captures the intuition that the SUFFIXs *+t* and *+ed* are usually, but not always, in conflict. The motivation for including (8) is to override the default application of (6) in appropriate circumstances. In fact, this is an inference that will not follow in all theories of default logic. However, we postpone detailed discussion of the precise theory of nonmonotonic reasoning required to validate the desired patterns of reasoning until §4.4. In addition, we want irregular past participle formation to continue to apply. The (minimal) pattern of reasoning required is thus schematised in (9).

(9) a $\phi_{ls} > \psi_{ss} \wedge \psi_o$
b $\chi_{ls} > \zeta_{ss} \wedge \zeta_o$
c $\zeta_o > \neg\psi_o$
d $\phi_{ls} \sqsupset \chi_{ls}$
e χ_{ls}
f $\models\!\!\!\approx \zeta_{ls}$
$\not\models\!\!\!\approx \psi_{ls}$

Lower case Greek symbols range over lexical signs and subscripts are used to refer to the values of paths, SYN and SEM (ϕ_{ss}) and ORTH (ϕ_o) within a derived lexical sign (ϕ_{ls}). In this case, (9a,b) represent the rules of regular and irregular past participle formation respectively. (9d,e) follow from the sort system, and (9c) represents the blocking constraint for past participle formation (i.e., the default rule (8) above). However, note that $>_{LR1}$ is a special case of $>$ so that the schema can be thought of as generalising over all lexical rules (and other defeasible conditionals) in the description language. When we have an irregular verb (χ) we wish to conclude that we have an irregular derived past participle (ζ) and to prevent the derivation of a regular past participle (ψ); that is, not infer ψ and optionally infer $\neg\psi$. (9c) is a specific default rule relevant to the blocking of regular past formation by irregular past participle formation. In §5 we argue that this and many other blocking rules can be derived from an indefeasible principle of blocking, so it is not necessary to stipulate all the specific rules required to implement every individual case of blocking.

4.3 Unblocking

The fact that specific blocking rules, such as (8), are defaults can be exploited to develop a more elegant and conceptually clearer account of 'unblocking', where 'competing' rules do both fire. In the case of dual past verbs, such as *dream*, given the sort system defined in (5) the dual past class of verbs are subsumed by the class of irregular verbs. Therefore, these verbs will be treated like *sleep* with respect to past participle formation. In order to unblock regular past participle formation for these verbs we can override the blocking default by explicitly adding the assertion that dual past verbs are exceptional in that they tolerate both regular and irregular past participle forms. This information is represented in (10).

(10) $\langle \mathrm{LR2} \rangle (\langle \textsc{orth} \rangle (\langle \textsc{stem} \rangle (\mathrm{dualvstem}) \wedge \langle \textsc{suffix} \rangle (+\mathrm{t}))) \rightarrow \langle \mathrm{LR1} \rangle (\langle \textsc{orth} \rangle (\langle \textsc{suffix} \rangle (+\mathrm{ed})))$

By adding (10), we introduce an indefeasible assertion concerning dual class verbs' orthographic forms which will override the blocking default rule of (8), since indefeasible information always has priority over defeasible information (this is supported in all nonmonotonic logics). The schematic pattern of inference involves the addition of the subsumption relation and (10) to the pattern given in (9) above, as (11) illustrates.

(11) a $\phi_{ls} > \psi_{ss} \wedge \psi_o$
b $\chi_{ls} > \zeta_{ss} \wedge \zeta_o$
c $\zeta_o > \neg\psi_o$
d $\delta_o \wedge \psi_o$
e $\phi_{ls} \sqsupset \chi_{ls}$
f $\chi_{ls} \sqsupset \delta_{ls}$
g δ_{ls}
h $\approx\!\!\models \zeta_{ls}, \psi_{ls}$

The unblocking constraint is represented in (11d) and will prevent application of the blocking default in (11c), thus both (11a) and (11b) will apply to dual class verbs. So far, we have developed an account in which dual past verbs can be represented as a subclass of irregular verbs and the regular and irregular rules of past participle formation need only be stated once.

4.4 Choosing a Suitable Logic

We have proposed that to account for blocking, the statements about conflict among feature values must be defeasible. This had ramifications on the inferences we need to validate, and hence on the kind of logic that will be suitable. In this section, we introduce a logic for defeasible reasoning called commonsense entailment (or CE) (Asher and Morreau, 1991), and argue that for our purposes it has certain features that make it more attractive than other candidate logics. In particular, we'll suggest that CE's logical consequence relation supports the patterns of inference (9) and (11) in a straightforward and more elegant way than the other candidate logics.

The default constraint language that we use is a sorted version of NPL, augmented with a nonmonotonic conditional operator $>$; CE supplies a modal semantics for such default formulas. Intuitively, $\phi > \psi$ is read as *If ϕ then normally ψ*. The constraint language incorporates two logical consequence relations. The first relation is represented as $\models$; it is monotonic and supra-classical. It applies to the truth conditional component of the logic, and supports intuitively compelling monotonic patterns of inference involving defeasible statements, such as Facticity.

- `Facticity`: $\models \phi > \phi$

In words, Facticity validates *If ϕ then normally ϕ*.

The second logical consequence relation is nonmonotonic and written as $\approx\!\!\models$. It underlies a partial dynamic theory of belief, which is defined on top of the truth conditional semantics for defeasible statements. It supports intuitively compelling patterns of defeasible inference, such as those given below:

- `Defeasible Modus Ponens:` $\phi > \psi, \phi \mathrel{|\!\approx} \psi$
  (e.g., birds normally fly, Tweety is a bird $\mathrel{|\!\approx}$ Tweety flies)
- `The Penguin Principle:` $\phi > \psi, \chi > \neg\psi, \chi \rightarrow \phi, \chi \mathrel{|\!\approx} \neg\psi$
  (e.g., birds fly, penguins don't fly, penguins are birds, Tweety is a penguin $\mathrel{|\!\approx}$ Tweety doesn't fly.)
- `The Nixon Diamond:` $\phi > \psi, \chi > \neg\psi, \phi, \chi \mathrel{|\!\not\approx} \psi$ (and $\mathrel{|\!\not\approx} \neg\psi$)
  (e.g., Quakers are pacificists, Republicans are non-pacifists, Nixon is a Quaker and Republican $\mathrel{|\!\not\approx}$ Nixon is a pacifist, and $\mathrel{|\!\not\approx}$ Nixon is a non-pacifist.)

Defeasible Modus Ponens captures the intuition that one can infer the consequent of a default rule if it is consistent with what is already known. The Penguin Principle contains two conflicting default rules in the premises, in that the consequents of both laws cannot hold in a consistent knowledge base (KB). The antecedents of these rules are logically related because χ entails ϕ; in this sense $\chi > \neg\psi$ is more specific than $\phi > \psi$. The Penguin Principle therefore captures the intuition that one prefers specific defaults to general ones when they conflict. The Nixon Diamond is like the Penguin Principle in that two conflicting default laws apply. However it differs in that these default laws do not stand in a relation of specificity to each other. CE is a logic where no conclusion is drawn under these circumstances without further information. In other words, it supports the intuition that conflict among default laws is irresolvable when they're unrelated.

The way CE is set up gives at least two advantages over other logics for defeasible reasoning. First, it splits the monotonic component of the theory from the nonmonotonic component, and this can, on occasion, prove conceptually useful when the premises form complex logical structures. Second, unlike Hierarchical Autoepistemic Logic (Konolige, 1988) and Prioritised Circumscription (McCarthy, 1980; Lifschitz, 1984), it captures the Penguin Principle without the intervention of a device which is extraneous to the semantic machinery. We find this latter property of CE attractive, since intuitively, whatever kind of reasoning default reasoning is, that more specific information takes precedence is intrinsic to it and so should be captured in the semantics of defaults.

We will shortly give a brief description of the way the dynamic partial theory of belief supports nonmonotonic patterns of inference, and in particular how it supports the inferences (9) and (11). But first, we consider the truth conditional component of the theory. The semantics of defeasible statements are defined in terms of a function $*$ from worlds and propositions to propositions. This function forms part of the model. And intuitively, $*(w, p)$ is the set of worlds where the proposition p holds together with everything else which, in world w, is normally the case when p holds. So $*$ encodes assumptions about normality. The truth conditions of defeasible statements are defined as follows:

- $M, w \models \phi > \psi$ if and only if $*(w, [\![\phi]\!]) \subseteq [\![\psi]\!]$

In words, this says that *If* ϕ *then normally* ψ is true with respect to a model M at a possible world w if the set of worlds that defines what is normally the case when ϕ is true in w all contain the information that ψ is also true. Thus CE is a conditional logic in the Acqvist (1972) tradition; it differs from previous conditional logics in the constraints it puts on $*$.

There are two constraints: the first is Facticity, which intuitively states that however few propositions normally hold when p is the case, one of them is p.

- `Facticity:` $*(w, p) \subseteq p$

This constraint enables the monotonic inference Facticity stated above to be verified. The second constraint is Specificity: it is essential for capturing the Penguin Principle as we will see shortly.

It expresses the intuition that penguins aren't normal birds, because penguins don't have the flying property associated with normal birds. To see this, think of p as *Tweety is a penguin*, and q as *Tweety is a bird*; so $*(w,p)$ is *Tweety is a normal penguin* and $*(w,q)$ is *Tweety is a normal bird.*

- `Specificity`: If $p \subseteq q$ and $*(w,p) \cap *(w,q) = \emptyset$, then $*(w,q) \cap p = \emptyset$

The dynamic semantics of information states is set up so that intuitively, Defeasible Modus Ponens goes as follows: first, one assumes the premises $\phi > \psi$ and ϕ and no more than this. Second, one assumes that ϕ is as normal as is consistent with these premises. Finally, from these two assumptions, one concludes that ψ. The theory makes precise the notion of assuming no more than the premises of an argument (in terms of an update function), and assuming everything is as normal as is consistent with those premises (in terms of a normalisation function). Information states are considered to be sets of possible worlds. The update function $+$ is defined on an information state s and a set of WFFS Γ, and it outputs a new information state that supports everything in the old information state as well as supporting Γ.

- `Update Function`: $s + \Gamma = \{w \in s : w \models \Gamma\}$

The normalisation function N uses a definition of $*$ on information states (as opposed to possible worlds):

- `Definition of * on information states`: $*(s,p) = \cup_{w \in s} *(w,p)$

N takes an information state s and a WFF ϕ as input, and it outputs a new information state s'. Intuitively, s' isolates those worlds in s where either ϕ isn't true at all, or its true but normal, if there are any worlds where ϕ is normal; otherwise it 'gives up' and simply returns the original state s.

- `The Normalisation function`:
$$\begin{aligned} N(s,\phi) &= \{w \in s : w \notin \llbracket\phi\rrbracket \setminus *(s,\llbracket\phi\rrbracket)\} \text{ if } *(s,\llbracket\phi\rrbracket) \cap s \neq \emptyset \\ &= s \text{ otherwise} \end{aligned}$$

Or to put it another way, if it is consistent in s to assume that ϕ is normal (and therefore by Facticity it is also consistent to assume ϕ is true), then $N(s,\phi)$ eliminates all those worlds from s where ϕ is true but abnormal. So normalisation is the process where one revises the information state to include the assumption that ϕ is as normal as possible, given what is already known. Pictorially, we can represent the effect of normalisation as in Figure 5, where the shaded areas define the value of $N(s,\phi)$.

Suppose one knows the premises Γ. Then from the normalisation function, one can build a normalisation *chain* for Γ. First, one orders the antecedents of the default rules in Γ into a sequence μ. One then builds the normalisation chain by repeated applications of N: one applies N to the current information state and the first element in μ; then one applies N to the result of this first application of N, and the *second* element in μ; and so on until one reaches a fixed point (such a point will be reached since N is non-increasing). In order to see what nonmonotonically follows from the premises Γ, one starts with the information state $\smiley + \Gamma$, where $\smiley$ is the information state containing nothing but the laws of logic, and one sees what holds in all the fixed points of all the normalisation chains for all possible enumerations μ.[6] In propositional CE, $\models$ and $\approx\!\!\!\!|$ are both sound, complete and decidable (see Lascarides and Asher, in press).

[6] $\smiley$ is defined to be the set of worlds in the canonical model, that's used to prove completeness of $\models$. Since this corresponds to knowing nothing but the laws of logic, $\smiley + \Gamma$ corresponds to knowing nothing but Γ and the laws of logic.

To clarify this, let's see informally how $*$, $+$ and N interact to ensure that $\mathrel{|}\!\approx$ supports The Penguin Principle. Let $\Gamma = \{\phi > \psi, \chi > \neg\psi, \chi \rightarrow \phi, \chi\}$ (i.e., the premises of a Penguin Principle). Then there are essentially two possible enumerations μ_1 and μ_2, where in the former we normalise on χ first and then on ϕ, and in the latter we normalise on ϕ first and then χ. Consider the normalisation chain for μ_1. We must first calculate the value of the function $N(\smiley +\Gamma, \chi)$. Note that $\smiley+\Gamma$ contains worlds where χ is normal, and so the first clause in the definition of N applies. Thus, given the truth definition of $\chi > \neg\psi$, $N(\smiley+\Gamma, \chi) = s$ for some information state s that is contained in $\llbracket\neg\psi\rrbracket$. Pictorially, s corresponds to the shaded areas in the diagram in Figure 6.

Now we must work out the value s' of $N(s, \phi)$. Since $s \subseteq \llbracket\neg\psi\rrbracket$, and by the definition of $\phi > \psi$ $*(w, \llbracket\phi\rrbracket) \subseteq \llbracket\psi\rrbracket$, there are no worlds in s where ϕ is normal. This situation is depicted in Figure 7: the second clause in the definition of N applies. Therefore $s' = s$, and so it also supports $\neg\psi$. Since N is non-increasing, it should be clear that $\neg\psi$ will be in the fixed point of this normalisation chain.

Now consider the enumeration μ_2. We must first work out the value of $N(\smiley+\Gamma, \phi)$. By the truth conditions of $\phi > \psi$ and $\chi > \neg\psi$, $*(w, \llbracket\phi\rrbracket) \cap *(w, \llbracket\chi\rrbracket) = \emptyset$. And $\chi \rightarrow \phi$ means that $\llbracket\chi\rrbracket \subseteq \llbracket\phi\rrbracket$. So the Specificity Constraint on $*$ applies here; for all worlds w in $\smiley + \Gamma$, $*(w, \llbracket\phi\rrbracket) \cap \llbracket\chi\rrbracket = \emptyset$. But $\smiley+\Gamma \subseteq \llbracket\chi\rrbracket$. So there are no worlds in $\smiley + \Gamma$ where ϕ is normal. Therefore $N(\smiley + \Gamma, \phi) = \smiley+\Gamma$. Now we calculate the value of $N(\smiley + \Gamma, \chi)$, and as before, $N(\smiley+\Gamma, \chi) \subseteq \llbracket\neg\psi\rrbracket$. So this normalisation chain also supports $\neg\psi$ in its fixed point. From this brief sketch, one can intuitively see that $\mathrel{|}\!\approx$ supports the Penguin Principle, since $\neg\psi$ holds in all fixed points of all normalisation chains. What's more, it has done so without having to assume a particular order of application of the default rules. This is in sharp contrast to HAEL and Prioritised Circumscription. In these logics, one must constrain the reasoning process so that one applies the more specific default rule first, which then blocks applying the second more general rule.

The advantages of having specificity baked into the semantics of default statements really come into their own when one considers the pattern of inference in (12), which we claim underlies blocking.

(12) a $\phi_{ls} > \psi_{ss} \wedge \psi_o$
b $\chi_{ls} > \zeta_{ss} \wedge \zeta_o$
c $\zeta_o > \neg\psi_o$
d $\phi_{ls} \sqsupset \chi_{ls}$
e χ_{ls}
f $\mathrel{|}\!\approx \zeta_{ls}$
$\mathrel{|}\!\not\approx \psi_{ls}$

To see this, we will indicate how CE supports (12), and then briefly consider how HAEL and prioritised circumscription would deal with it. Let the premises in (12) be Γ. There are three default rules in Γ, making six possible enumerations. First consider the enumeration μ_1 where we normalise in the order: ϕ_{ls}, χ_{ls}, ζ_o. The Specificity Constraint doesn't apply here, since even though the premises entail $\llbracket\chi_{ls}\rrbracket \subseteq \llbracket\phi_{ls}\rrbracket$, they don't allow us to assume $*(w, \llbracket\phi_{ls}\rrbracket) \cap *(w, \llbracket\chi_{ls}\rrbracket) = \emptyset$. So we cannot assume $*(w, \llbracket\phi_{ls}\rrbracket) \cap \llbracket\chi_{ls}\rrbracket = \emptyset$. Therefore, the first clause in the definition of N applies when calculating $N(\smiley + \Gamma, \phi_{ls})$, for there are worlds in $\smiley+\Gamma$ where ϕ_{ls} is normal. So $N(\smiley+\Gamma, \phi_{ls}) = s$ where $s \subseteq \llbracket\psi_{ss} \wedge \psi_o\rrbracket$. This normalisation is depicted in figure 8, where s is characterised by the shaded areas.

Now s must contain worlds where χ_{ls} is normal, since this is consistent with what is known, and therefore $N(s, \chi_{ls}) = s'$, where $s' \subseteq \llbracket\zeta_{ss} \wedge \zeta_o\rrbracket$. Normalisation on the specific default is given in figure 9: s' is the shaded area.

Furthermore N is non-increasing, and so $s' \subseteq s \subseteq \llbracket\psi_o\rrbracket$. By the truth conditions of (12c),

$*(w, [\![\zeta_o]\!]) \cap [\![\psi_o]\!] = \emptyset$. So there are no worlds in s' where ζ_o is normal. Therefore $N(s', \zeta_o) = s'$. Thus the final link in the normalisation chain is depicted in figure 10. In fact, the fixed point of this normalisation chain μ_1 contains $[\![\zeta_{ls}]\!]$ and $[\![\psi_{ls}]\!]$.

Now consider the enumeration μ_2, where we normalise in the order: χ_{ls}, ζ_o, ϕ_{ls}. $N(☺ + \Gamma, \chi_{ls}) = s$, where $s \subseteq [\![\zeta_{ss} \wedge \zeta_o]\!]$. And $N(s, \zeta_o) = s'$, where $s' \subseteq [\![\neg\psi_o]\!]$. By the truth definition of (12a), $*(w, \phi_{ls}) \cap [\![\neg\psi_{ls}]\!] = \emptyset$. So s' doesn't contain any worlds where ϕ_{ls} is normal (because $s' \subseteq [\![\neg\psi_o]\!] \subseteq [\![\neg\psi_{ls}]\!]$). So $N(s', \phi_{ls}) = s'$. In fact, the fixed point of this normalisation chain μ_2 contains $[\![\zeta_{ls}]\!]$ and $[\![\neg\psi_{ls}]\!]$.

These normalisation chains show that the following holds, as we required in (12):

- $\Gamma \not\approx \psi_{ls}$
  $\Gamma \not\approx \neg\psi_{ls}$

In fact, using a similar line of reasoning, the other four normalisation chains lead to fixed points that all support ζ_{ls} (the details are omitted for reasons of space). So the following also holds, as we require:

- $\Gamma \approx \zeta_{ls}$

Hence the pattern of inference (12), which underlies lexical blocking, is supported in CE. So CE provides a suitable default logic with which to formalise the constraint language.

There are other candidate logics for defeasible reasoning, so why choose CE? Let's assess how the other logics deal with (12). A few comments about the logical structure of (12) will help clarify the discussion. The default rule (12a) is more specific than (12b) because of (12d). Furthermore, because of (12c), the consequents of these rules *conflict by default.* The inference in (12) resolves this conflict in favour of the more specific rule. So a logic will be adequate for our purposes only if it has the means to resolve conflict. Not all logics for defeasible reasoning have this facility; for example, Reiter's (1980) default logic, autoepistemic logic (Moore 1984) and circumscription (McCarthy 1980). There are at least two logics apart from CE that have the facility to resolve conflict among defeasible rules: HAEL, and prioritised circumscription. As we've mentioned, these logics resolve the conflict by constraining the reasoning processes with machinery that is extraneous to the semantics of default statements.

HAEL is an extension of autoepistemic logic motivated in part by the need to validate the Penguin Principle. An autoepistemic logic is a theory of KB extension: if the consequent of a default law is consistent with the rest of the contents of the KB, then it is added to that KB. In HAEL, certain KB expansions are preferred on the basis that the defaults used in them have a higher priority in some well-defined sense than the ones used in the alternative expansions. Informally, the contents of a KB are structured into a hierarchy. The information at each level in the hierarchy represents different sources of information available to an agent, while the hierarchy expresses the way in which this information is combined. Facts and indefeasible laws are placed at the first level, and more specific default laws are placed at a lower level in the hierarchy than less specific default laws; the way the logic works ensures that information at level n has priority over information at level m where $n < m$; in other words the effects of default laws at lower levels in the hierarchy can block the action of more general defaults, placed higher in the hierarchy. The only constraint on the hierarchy is that more specific defaults must be at a lower level. Since (12b) and (12c) have unrelated antecedents, they can appear in any relative order in the hierarchy, even though they conflict. This proves problematic, if we wish to support (12) in HAEL. For suppose we put (12b) at a lower level in the hierarchy than (12c). This means that in the KB extension, (12b) will have priority over (12c). Hence ψ_{ls} will follow from the premises (12a-e), contrary to our requirements. To avoid this undesirable result, we would have to constrain the way we construct the hierarchy, in order to ensure that (12c) is always placed at a lower level than (12b), (if they're at the same level, we would in fact still infer ψ_{ls}). But this has serious ramifications on the translation process of default statements into the object language. We would be committed to constraining the hierarchy in this case so that two defaults with logically unrelated antecedents appear in a particular order. But it is not always the case that we wish to constrain the order of two such defaults (cf. the premises of a Nixon Diamond). So we have sacrificed the means to translate default statements into the logic in a uniform way.

A similar problem arises in prioritised circumscription, where the abnormality predicates are ordered into a hierarchy, and one prefers models that satisfy the premises and which minimise on abnormality predicates higher in the hierarchy than those lower down. Again, we would have to be careful about the order in which we place the abnormality predicates associated with the default rules (12b) and (12c), if the logic is to capture the pattern of inference we need to do blocking. Again, we would fail to supply a uniform translation of default statements into the formal language. By contrast, our formalisation of blocking in CE captures the pattern of inference we need, while at the same time maintaining a uniform translation of default statements into the object language. This is desirable because it allows for a more direct, perspicuous and elegant encoding of lexical rules and other constraints without the need to impose any external, linguistically-unmotivated ordering.

5 Blocking – A Unified Phenomenon?

The account we have developed would be unsatisfactory if we were unable to derive any general principle of blocking, since intuitively the same principle is involved in many other examples including those cited in §2. We argue that specific blocking defaults, such as that for irregular past verbs, follow from an indefeasible rule which can express a version of 'preemption by synonymy' restricted via the subsumption relation. This rule enforces the principle that, if there are two lexical rules which validate derived signs with equivalent SYN and SEM components but distinct ORTH components, and the antecedents of these rules stand in a subsumption relation, then a blocking default is validated negating the derived orthography of the more general rule. We express the first version of the blocking principle as a deductive propositional schema; that is, as a 'metalevel' indefeasible implication over lexical descriptions, in (13).[7]

(13) a) $(\phi_{ls} > \psi_{ls}) \wedge (\chi_{ls} > \zeta_{ls}) \wedge$
b) $(\chi_{ls} \rightarrow \phi_{ls}) \wedge \neg(\phi_{ls} \rightarrow \chi_{ls}) \wedge$
c) $\psi_{ss} \leftrightarrow \zeta_{ss} \wedge \neg(\psi_o \leftrightarrow \zeta_o))$
$\rightarrow (\zeta_o > \neg\psi_o)$

The schema in (13a) represents any two lexical rules, and (13b) states that there must be an (indirect) unidirectional subsumption relation between the antecedents of these rules (inferrable from the sort system). (13c) states that these rules must result in derived lexical signs with equivalent values for SYN and SEM and distinct values for ORTH. If the conditions expressed in the antecedent of (13) are met the principle validates the deduction of a specific blocking default negating the orthography of the more general derived rule. This deduction takes place in the monotonic component of the constraint language (since it is an indefeasible implication). Hence if (12a,b,d) are verified, the default statement (12c) of conflict is inferred *before* normalisation starts. And as we saw in §4.4, the nonmonotonic component can use this default statement of conflict to achieve the desired effect of blocking. Moreover, it's important to stress that the statement of conflict is a default; so even if its antecedent is verified by the KB, its effects can be overridden by indefeasible unblocking assertions of the form $\psi_o \wedge \zeta_o$ (as in the case of the dual class verbs in §4.3).

[7] It's important to stress that (13) is a metalevel rule only in the sense that it is a schema; its desired effects are captured in the logic without have to assume metalevels of reasoning.

The blocking principle in (13) is adequate to generate the blocking default required to prevent **sleeped* discussed in §4.2 and many others. The case of *+ness* and *+ity*, introduced in §2, can be treated in a manner analogous to past participle formation, since we have two lexical rules which clearly stand in a subsumption relation and generate differing forms with the same syntax (and meaning). However, an account of the *curious, curiosity; glorious, glory, *gloriousity* case cannot be obtained for the blocking principle as stated in (13) because we require the blocking of a (derivational) morphological rule in the presence of an underived synonymous but orthographically distinct lexical sign. A similar situation obtains with the rule of 'animal grinding' sense extension or conversion which produces the meat reading of *lamb*, and so forth, but is blocked in the case of *pig* by *pork*. In these cases, it appears that we must state that 'preemption by synonymy' also occurs when a non rule-generated synonymous word exists. We will concentrate on developing an account of the blocking of animal grinding, since this putative lexical rule has motivated much of our interest in blocking (Copestake & Briscoe, 1991). We can recast the rules of grinding and animal grinding as lexical rules, as in (14a,b).

(14) a
$$\begin{bmatrix} \text{ORTH} = \boxed{j} \\ \text{SYN} = \begin{bmatrix} \text{CAT} = \textit{noun} \\ \text{CNT} = + \end{bmatrix} \\ \text{SEM} = \begin{bmatrix} \text{REF} = \textit{ind\_obj} \wedge \boxed{k} \end{bmatrix} \end{bmatrix} \rightarrow_{LR3} \begin{bmatrix} \text{ORTH} = \boxed{j} \\ \text{SYN} = \begin{bmatrix} \text{CAT} = \textit{noun} \\ \text{CNT} = - \end{bmatrix} \\ \text{SEM} = \begin{bmatrix} \text{REF} = \textit{substance} \\ \text{ORIGIN} = \boxed{k} \end{bmatrix} \end{bmatrix}$$

b
$$\begin{bmatrix} \text{ORTH} = \boxed{j} \\ \text{SYN} = \begin{bmatrix} \text{CAT} = \textit{noun} \\ \text{CNT} = + \end{bmatrix} \\ \text{SEM} = \begin{bmatrix} \text{REF} = \textit{animal} \wedge \boxed{k} \end{bmatrix} \end{bmatrix} >_{LR4} \begin{bmatrix} \text{ORTH} = \boxed{j} \\ \text{SYN} = \begin{bmatrix} \text{CAT} = \textit{noun} \\ \text{CNT} = - \end{bmatrix} \\ \text{SEM} = \begin{bmatrix} \text{REF} = \textit{nat\_food\_subst} \\ \text{ORIGIN} = \boxed{k} \end{bmatrix} \end{bmatrix}$$

We assume that the class of 'animals' is subsumed by the class of 'ind(ividuated)_obj(ects)', that (14a) is an indefeasible rule, and that (14b) applies by default to further specify the interpretation of grinding applied to animals (see Copestake & Briscoe, 1991) for a more detailed justification of this analysis, explication of ORIGIN, and so forth). In order to not generate this more specified meaning for *pig*, by default, we must express the manner in which *pork* takes precedence over *pig*. As it stands the blocking principle apparently states that blocking applies when two lexical rules are involved, but in this case blocking is caused by the presence of an underived lexical sign (standing in the same general relationship to the derived sign). But the underived sign does not stand in a subsumption relation to either sign in the lexical rule. Intuitively, though the underived sign is more 'reliable' than the derived one because it represents a fact, rather than a (nonmonotonic) inference. We can incorporate this intuition into the blocking principle without modification, given that we use CE, because the axioms of CE in (15a,b) allow us to make this pattern of inference parallel to the earlier case.

(15) a $\models \bot \rightarrow \psi$
b $\models (\psi \rightarrow \phi) \rightarrow (\psi > \phi)$

In (16) χ_{ls} represents a basic lexical sign such as *pork* and we use the axioms in (15a,b) to obtain the formula in (16b) and (15a) to obtain (16c).

(16) a $\phi_{ls} > \psi_{ls}$
b $\bot > \chi_{ls}$
c $\bot \rightarrow \phi_{ls} \wedge \neg(\phi_{ls} \rightarrow \bot)$
d $\psi_{ss} \leftrightarrow \chi_{ss} \wedge \neg(\psi_o \leftrightarrow \chi_o))$

If (16a) represents animal grinding then application of this rule to *pig* will produce a lexical sign which stands in the relation to *pork* (i.e., χ_{ls}) defined by (16d), and the formulas of (16) together will instantiate the blocking principle given in (13). Therefore, we will derive the specific blocking default that we require to prevent animal grinding validating *pig*, meaning meat; schematically, $\chi_o > \neg\psi_o$; and the application of animal grinding in this case will be blocked because the premises will be analogous to the case of irregular past participle formation in §4.2. Thus, these axioms allow us to formalise the similarity between these two examples of blocking without stating that *pork* is subsumed by *pig* (meat), as we would have been forced to do in a purely inheritance based account.

The type of blocking which is occurring in the case of words such as *sticker* or *banker* is apparently rather different to 'preemption by synonymy'. In these cases, a different meaning for an orthographically identical form is blocked. Ostler & Atkins (1991) refer to this as 'lexical preemption'. Nevertheless, the pattern of inference required to block the derived, productive sign resulting from *+er* nominalisation is very similar to the cases of semantic preemption we have considered. We assume that the non-productive meaning of forms such as *sticker* means that underived lexical signs are present expressing such senses. Application of *+er* nominalisation will be blocked by a specific blocking default which asserts that the 'more reliable' monotonically derivable semantics of the basic sign for *sticker* negates the productive default semantics produced by the rule. In (17) we give a (simplified) rule of *+er* nominalisation.

(17)
$$\left[\begin{array}{ll} \text{ORTH} = & \left[\text{STEM} = \boxed{j} \wedge vstem\right] \\ \text{SYN} = & \left[\begin{array}{ll} \text{CAT} = verb & \\ \text{SUBJ} = & \left[\begin{array}{l} \text{SYN} = \left[\text{CAT} = np\right] \\ \text{SEM} = \left[\text{REF} = \boxed{k}\right] \end{array}\right] \\ \text{VFORM} = base & \end{array}\right] \\ \text{SEM} = & \left[\begin{array}{l} \text{PRED} = pred \\ \text{ARG1} = \boxed{k} \wedge anim\_obj \end{array}\right] \end{array}\right] >_{LR5} \left[\begin{array}{ll} \text{ORTH} = & \left[\begin{array}{l} \text{STEM} = \boxed{j} \\ \text{SUFFIX} = +er \end{array}\right] \\ \text{SYN} = & \left[\text{CAT} = noun\right] \\ \text{SEM} = & \left[\text{REF} = \boxed{k}\right] \end{array}\right]$$

This rule will produce a meaning for *sticker* which can be glossed as 'person who sticks (things)'. We represent the pattern of inference required to block derivation of this sign in (18).

(18) a $\phi_{ls} > \psi_{se} \wedge \psi_{sy} \wedge \psi_o$
b $\chi_{se} > \neg\psi_{se}$
c ϕ_{ls}
d χ_{ls}
e $\not\approx\ \psi_{ls}$

This pattern results in ψ_{ls} being blocked, because there is irresolvable conflict between the default rules (18a) and (18b). In order to derive the blocking default in (18b) we need to introduce a second related blocking condition which introduces semantic blocking defaults when the conditions for 'lexical preemption' are satisfied. We give this version of the blocking principle in (19).

(19) a) $(\phi_{ls} > \psi_{ls}) \wedge (\chi_{ls} > \zeta_{ls}) \wedge$
b) $(\chi_{ls} \rightarrow \phi_{ls}) \wedge \neg(\phi_{ls} \rightarrow \chi_{ls}) \wedge$
c) $\psi_{os} \leftrightarrow \zeta_{os} \wedge \neg(\psi_{se} \leftrightarrow \zeta_{se}))$
$\rightarrow (\zeta_{se} > \neg\psi_{se})$

This second principle is symmetric with the first in (13) above, but reverses the conditions for preemption. ψ_{os} abbreviates values for ORTH and SYN on a lexical sign and ψ_{se} those for SEM.

As before, (19) will generate the required blocking default when an underived lexical sign bears the relation defined in (19c) to a lexical sign derived by default. So in particular, (19) will generate (18b).

It might be thought that the availability of the axioms in (17) makes the blocking principles too unconstrained. For example, what is there to stop us blocking one sense of a homonymous form such as *bank* by applying the axioms of (17) to two basic signs which satisfy the conditions of the second principle in (19)? In fact, condition b) in the antecedent of the blocking principles prevents two underived forms satisfying them, despite the availability of axioms which allow any underived sign to be expressed as the consequent of a default implication. Condition b) fails to be satisfied in such a situation because $\neg(\bot \rightarrow \bot)$ can never be true in any model.

The symmetry between the two versions of the blocking principle, encoding the conditions of semantic and lexical pre-emption mean that it is possible to merge the two principles into one schematic blocking principle, which illustrates more clearly the similarities between the two subcases. (20) is the final version of the blocking principle and replaces both previous versions.

(20) a) $(\psi_{ls} > \phi_{ls}) \wedge (\chi_{ls} > \zeta_{ls}) \wedge$
b) $(\chi_{ls} \rightarrow \psi_{ls}) \wedge \neg(\psi_{ls} \rightarrow \chi_{ls}) \wedge$
c) $\phi_{syn} \leftrightarrow \zeta_{syn} \wedge$
d) $\phi_x \leftrightarrow \zeta_x \wedge$
e) $(\neg(\phi_y \leftrightarrow \zeta_y))$
$\rightarrow (\zeta_y > \neg\phi_y)$

Clauses c) – e) express two subcases: for preemption, syntactic identity is always required, x and y are variables ranging over SEM and ORTH. If x is instantiated to SEM and y to ORTH, we have semantic preemption, otherwise we have lexical preemption.

We've seen how a general blocking principle can be encoded in our constraint language. It is interesting to note that such a principle wouldn't have the desired effect in HAEL. This is because in HAEL, the default information is 'drip-fed' into the reasoning process as one climbs the hierarchy. (20) will be known from the start (i.e., at level 0) since it is indefeasible knowledge, but its antecedent won't be verified until the reasoning process reaches the level where the general default $\psi_{ls} > \phi_{ls}$ is introduced. Therefore, given the way HAEL works, the consequents of both (20) and this general default will be inferred at the next level in the hierarchy. Consequently, the blocking default is eventually inferred, but it is too late, because ϕ_{ls} has also been inferred: the blocking principle has failed to block it. Here, we see another advantage in the way CE is set up. In contrast to HAEL, *all* the defeasible information is known from the start of the reasoning process, and all monotonic inferences are carried out before the nonmonotonic ones.

6 (Un)Blocking and Interactions with Interpretation

A major motivation for developing a default account of blocking was to provide an account of interactions during interpretation, which would allow, say, *pig* to have the meaning of meat in an appropriate context; for example, *I ate pig last night in this awful restaurant.* In the extant theories of lexical organisation which we have considered in §3, there is no way of blocking the application of animal grinding to a form such as *pig* deriving a meaning synonymous with *pork.* In addition, in those theories, if it were possible to block *pig* meaning meat via the existence of the underived form *pork*, then there would be no way of unblocking it; just as in those theories there is no way of unblocking *dreamed* in a manner which also captures the intuition that *dreamed* is a subclass of the irregular past participle formation verbs.

A full account of the unblocking of animal grinding is beyond the scope of this paper. However, we can illustrate the approach that might be taken within this framework with respect to the blocked form **sleeped*. In a context where a (first or second) language learner utters, say, *I sleeped well*, it seems plausible that listeners infer an unblocking default 'on-line'. The problem is: why do we interpret *sleeped* as the past participle of *sleep* in this case? Intuitively, one recognises that the more general default for constructing past participle verb forms applies. One can express this intuition declaratively in CE plus NPL. Let Γ be (21a-e); i.e., the premises of the inference that enables us to infer that the past participle form of *sleep* is *slept*.

(21) a $\phi_{ls} > \psi_{ss} \wedge \psi_o$ where $\psi_o = i \rightarrow LRn\langle \text{ORTH} \rangle(\alpha_o)$
b $\chi_{ls} > \zeta_{ss} \wedge \zeta_o$
c $\zeta_o > \neg\psi_o$
d $\phi_{ls} \sqsupset \chi_{ls}$
e χ_{ls}
f $\mathrel{|}\!\approx \zeta_{ls}$
$\not\mathrel{|}\!\approx \psi_{ls}$

We formalise the effect of the utterance of *sleeped* as adding $\diamond\alpha_o$ to these premises. The possibility operator has the effect that we are asserting that α_o (i.e., the orthography "sleeped") is true at some node in the feature structure, since $\diamond\alpha_o$ means that α_o is true at some node, but not necessarily the current one. Then the embedded default in (22) captures the intuition that, upon learning that α_o must hold at some node (through the utterance of *sleeped*), one assumes that by default, the consequent of the more general default rule holds and thus α_o is true as the value of the orthography for ψ_{ls}.

(22) $(\Gamma \wedge \diamond\alpha_o) > \psi_{ls}$

So, suppose the KB verifies Γ and $\diamond\alpha_o$. Then (22) is the most specific default that applies. So by the Penguin Principle its consequent is inferred. This means that in the case of hearing *sleeped*, one infers that it is to be interpreted as the past participle form of *sleep*.

Our use of a default logic incorporating a dynamic theory of belief guarantees us intertranslatibility between the constraint language we use to describe the lexicon and the logic in which we characterise processes of interpretation (e.g. Lascarides & Asher, in press). This enables us to elegantly capture such interactions by seemlessly extending the lexical KB into the more general modular, but interacting KB underlying language interpretation. In addition, the fact that we model the inference underlying the interpretation of *sleeped* using an embedded default has ramifications on the kind of logic that can account for these phenomena. Embedded defaults are syntactically ill-formed in HAEL and default logic, but well-formed in circumscription and CE.

7 Conclusion

In this paper, we have shown that it is possible to derive a default account of blocking. This account is more compatible with linguistic facts. By representing conflict among feature values as default statements, we were able to support inferences that underly blocking without making blocking an absolute principle of lexical organisation. This paves the way for more insightful accounts of the status of words, such as *pig* meaning meat, *?curiousness*, and so forth. In this paper, we have sketched preliminary accounts of a few of these cases in the logic CE, and argued that this is a more appropriate for this application. In addition we have argued that our approach provides a basis for characterising interactions between lexical and wider interpretative principles.

Much work remains to be done: we think that the approach could be improved by being recast within a typed framework (Carpenter 1992) which would disallow arbitrary extensions of feature structures. Within such a treatment the notion of lexical rule could be constrained: here we have assumed an arbitrary mapping between feature structures is possible, but we would like to limit this to monotonic operations plus restricted mappings between particular types in the hierarchy. In general the computational complexity and consequent tractability of this approach needs to be investigated. We intend to explore the applicability of this approach to other examples of blocking; and to examine interactions between lexical blocking and other knowledge resources used in interpretation.

References

Aronoff, M. (1976) *Word Formation in Generative Grammar,* Linguistic Inquiry Monograph 1. MIT Press, Cambridge, Mass.

Aqvist, L. (1972) 'Logic of Conditionals', *Journal of Philosophical Logic, vol.1,* .

Asher, N. and M. Morreau (1991) 'Commonsense Entailment: A Modal Theory fo Nonmonotonic Reasoning', *Proceedings of the Proceedings to the 12th International Joint Conference on Artificial Intelligence,* Sydney, Australia, pp. 387–392.

Bauer, L. (1983) *English Word-formation,* CUP.

Bird, S. (1992) 'Finite-state phonology in HPSG', *Proceedings of the 15th International Conference on Computational Linguistics (COLING-92),* Nantes, France, pp. 74–80.

Blackburn, P. (1992a) 'Modal Logic and Attribute Value Structures' in M. de Rijke (eds.), *Diamonds and Defaults,* Studies in Logic, Language and Information. Kluwer, Dordrecht, Holland (available as University of Amsterdam, ITLI, LP-92-02).

Blackburn, P. (1992b) *Structures, Languages and Translations: the Structural Approach to Feature Logic,* Research Paper HCRC/RP-33. HCRC Publications, University of Edinburgh.

Bolinger, D. L. (1975) *Aspects of Language,* Harcourt, Brace and Jovanovich: New York.

Briscoe, E. J., and A. Copestake (1991) 'Sense extensions as lexical rules', *Proceedings of the IJCAI Workshop on Computational Approaches to Non-Literal Language,* Sydney, Australia, pp. 12–20.

Cahill L (1990) 'Syllable based morphology', *Proceedings of the 13th Coling,* Helsinki, pp. 48–54.

Cahill, L. (1992) 'Review of 'English Word Grammar'', *Computational Linguistics, vol.18 (1),* 92–94.

Calder, J. (1989) 'Paradigmatic morphology', *Proceedings of the 4th Conference of the European Chapter of the Association for Computational Linguistics (EACL-89),* Manchester, England, pp. 58–65.

Carpenter, R. (1992) *The logic of typed feature structures,* (Tracts in Theoretical Computer Science), Cambridge University Press, Cambridge, England.

Clark, E. V. and H. H. Clark (1979) 'When nouns surface as verbs', *Language, vol.55,* 767–811.

Copestake, A. (1992) 'The ACQUILEX LKB: representation issues in semi-automatic acquisition of large lexicons', *Proceedings of the 3rd Conference on Applied Natural Language Processing (ANLP-92),* Trento, Italy, pp. 88–96.

Copestake, A., A. Sanfilippo, E. J. Briscoe and V. de Paiva (in press) 'The ACQUILEX LKB: an Introduction' in E. J. Briscoe, A. Copestake and V. de Paiva (eds.), *Defaults, Inheritance and the Lexicon,* Cambridge University Press, Cambridge, England.

Copestake, A. and E.J. Briscoe (1991) 'Lexical Operations in a Unification Based Framework', *Proceedings of the ACL SIGLEX Workshop on Lexical Semantics and Knowledge Representation,* Berkeley, California, pp. 88-101.

Daelemans, W. (1987) 'Inheritance in Object-Oriented Natural Language Processing', *Proceedings of the 3rd Conference of the European Chapter of the Association for Computational Linguistics (EACL-1987),* Copenhagen, Denmark, pp. 70–74.

Evans, R. and G. Gazdar (1989) 'Inference in DATR', *Proceedings of the 4th Conference of the European Chapter of the Association for Computational Linguistics (EACL-1989),* Manchester, England, pp. 66–71.

Fraser, N. M. and R. A. Hudson (1992) 'Inheritance in Word Grammar', *Computational Linguistics, vol.18 (2),* 133–158.

Hudson, R. A. (1990) *Word Grammar,* Blackwell, Oxford.

Kasper, R. T. and W. C. Rounds (1990) 'The logic of unification in grammar', *Linguistics and Philosophy, vol.13 (1),* 35–58.

Konolige, K. (1988) *Hierarchic Autoepistemic Theories for Nonmonotonic Reasoning: Preliminary Report,* Technical Note No. 446, SRI International.

Lascarides, A. and Asher, N. (in press) 'Temporal Interpretation, Discourse Relations and Commonse Sense Entailment', *Linguistics and Philosophy, vol.,* .

Lifschitz, V. (1984) 'Some Results on Circumscription', *Proceedings of the Proceedings of the Nonmonotonic Reasoning Workshop,* AAAI, New York, pp. 151–164.

McCarthy, J. (1980) 'Circumscription: A Form of Nonmonotonic Reasoning', *Artificial Intelligence, vol.13,* 27–39.

Mellish, C. M. (1988) 'Implementing systemic classification by unification', *Computational Linguistics, vol.14(1),* 40–51.

Moore, R. C. (1984) *A Formal Theory of Knowledge and Action,* Technical Note, SRI International Number 320.

Ostler, N. and B. T. S. Atkins (1991) 'Predictable meaning shift: some linguistic properties of lexical implication rules', *Proceedings of the ACL SIGLEX Workshop on Lexical Semantics and Knowledge Representation,* Berkeley, California, pp. 76–87.

Pelletier, F.J., and Schubert, L. K. (1986) 'Mass Expressions.' in Gabbay and Guenthner (eds.), *Handbook of Philosophical Logic, Vol 4,* Reidel, Dordrecht.

Pollard, C. and I. Sag (1987) *An information-based approach to syntax and semantics: Volume 1 fundamentals,* CSLI Lecture Notes 13, Stanford CA.

Rappaport, M. and B. Levin (1990) '*-er* Nominals: Implications for the theory of argument structure' in E. Wehrli and T. Stowell (eds.), *Syntax and the Lexicon,* Syntax and Semantics 26, Academic Press, New York.

Reiter, R. (1980) 'A Logic for Default Reasoning', *Artificial Intelligence, vol.13,* 81–132.

Russell, G., J. Carroll and S. Warwick (1991) 'Multiple default inheritance in a unification-based lexicon', *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL-91),* Berkeley, California, pp. 215–221.

Russell, G., A. Ballim, J. Carroll and S. Warwick-Armstrong (in press) 'Practical Multiple Default Inheritance' in E. J. Briscoe, A. Copestake and V. de Paiva (eds.), *Defaults, Inheritance and the Lexicon,* Cambridge University Press, Cambridge, England.

Shieber, S. M. (1986) *An introduction to unification-based approaches to grammar,* CSLI Lecture Notes 4, Stanford CA.

Shieber, S. M. (1992) *Constraint-based grammar formalisms,* MIT Press, Cambridge, Mass..

de Smedt, K. (1984) 'Using object-oriented knowledge-representation techniques in morphology and syntax programming', *Proceedings of the ECAI-84,* Pisa, pp. 181–184.