Dialogue and Discourse 2(1) (2011) 171–197 doi: 10.5087/dad.2011.108

Incrementality and the Dynamics of Routines in Dialogue

Andrew Gargett ANDREW.GARGETT@UAEU.AC.AE
Department of Linguistics
U.A.E. University
Al Ain, U.A.E.

Editor: Hannes Reiser and David Schlangen

Abstract

We propose a novel dual processing model of linguistic routinisation, specifically formulaic expressions (from relatively fixed idioms, all the way through to looser collocational phenomena). This model is formalised using the Dynamic Syntax (DS) formal account of language processing, whereby we make a specific extension to the core DS lexical architecture to capture the dynamics of linguistic routinisation. This extension is inspired by work within cognitive science more broadly. DS has a range of attractive modelling features, such as full incrementality, as well as recent accounts of using resources of the core grammar for modelling a range of dialogue phenomena, all of which we deploy in our account. This leads to not only a fully incremental model of formulaic language, but further, this straightforwardly extends to routinised dialogue phenomena. We consider this approach to be a proof of concept of how interdisciplinary work within cognitive science holds out the promise of meeting challenges faced by modellers of dialogue and discourse.

Submitted 1/2010; Accepted 11/2010; Published online 5/2011

1. Orientation

In this paper we propose a unified approach to the relation between linguistic knowledge and linguistic experience, specifically, we present a new approach to modelling linguistic routinisation which, among other things, offers a way of capturing within the same framework the use of both formulaic and non-formulaic language in dialogue. As we discuss below, these phenomena have numerous distinct properties, yet they also share features directly relevant for formal modelling (see Nunberg et al. (1994)). Focusing on actual language use, we formally model the relative incrementality of both formulaic and non-formulaic language.

Our theoretical framework is inspired by work on dual process models of cognitive phenomena, in particular interaction within dialogue (Barr and Keysar (2006) provide in part a recent overview). Specifically, we model the interaction between rule-based and memory-based processing of natural language, formally implementing this within Dynamic Syntax (DS, Kempson et al. (2001), Cann et al. (2005)). While DS has typically focused on rule-based processing, we seek to extend this by arguing that actual patterns of linguistic phenomena, as found for example in dialogue, emerge out of the interaction between these distinct processes. For us, processing formulaic language is more likely to involve retrieval of items stored as wholes than computed online (details about this below), compared to the rule-driven processing underlying non-formulaic language.

For this formalisation of dual processing, we extend the lexical architecture of DS, so that lexical entries incorporate both the usual DS lexical actions, but also include the output semantic structures which result from employing such lexical actions. This sets up two competing processes for updating the representation being constructed for a speaker's utterance, a slower one based on lexical actions, and a faster one based on stored semantic structure (see Gargett (2010) for details). Importantly, in making this extension, we are concerned with retaining attractive properties of the DS account. In particular, we aim to preserve the dynamics of the model, retaining fully incremental processing within the context of interaction. We are able to effectively account for the dynamics of routinised dialogue phenomena via modifications to the DS model at the lexical level alone. This might be viewed as one in a line of recent proposals for lexicalist modelling of dialogue phenomena (e.g. Kecskes (2008)).

2. Motivations

Our approach is distinct from previous dual process models of dialogue in that we focus on core grammatical resources to model interaction, in line with previous Dynamic Syntax work (Purver et al. (2006), Gargett et al. (2009)), aiming to investigate the extent to which dialogue can be modelled using mechanisms specified by the grammar.[1] This mechanistic approach to modelling formulaic language, or *multi-word expressions* (MWEs), makes a break with the orthodoxy of property-list approaches which define MWEs via a list of linguistic properties, such as relative compositionality, idiosyncrasy of meaning, selectivity for discontinuous formulaic expressions, etc (e.g. Nunberg et al. (1994)). We avoid modelling directly in terms of properties, following the reasoning in Rawson (2004), regarding drawbacks of this. Instead, Rawson suggests mechanistic models (specifically those of Logan (1988) and Anderson (1992)), which we find fit remarkably well to our adopted approach to incremental processing.

For our purposes, linguistic routinisation involves long-term storing and reuse of context, with context taken to be the previous words plus their mode of construal (details to be made precise). Interlocutors may routinise the grammatical or semantic aspects of words or phrases, within a single turn or across multiple turns. Such routinisation is highly sensitive to specific features of the context of an interaction, such features typically triggering the routine. A note: defining the time periods relevant for the emergence of routines is somewhat problematic (given the fuzziness of notions of language, language use and context); here we will define short-term reuse as reuse of words and their construal immediately following the initial use, medium term as reuse later in the interaction, and long-term as reuse on some subsequent occasion of interaction.

2.1 Dynamics of incrementality

Milward (1994) proposes that incrementality involves:

(i) as much information being extracted as soon as possible

(ii) carried out in small steps approximately as each word is encountered

In a way that we will make more precise below, we can think of incremental processing of an incoming string of words as a kind of *stepping through* the string of items while constructing the unfolding representation. Then the processing of formulaic language can be seen as *skipping over* chunks of items rather than stepping through every possible individual item. Consider how a hearer might process the information provided by an utterance of:

(1) ‘Bob left’

Natural language utterances, like all kinds of natural phenomena, unfold in time, such temporal dimension of processing being a feature of *dynamical systems* (Ward (2002)). Such systems are inherently incremental, with input updating one state to the next. First, the occurrence of the name ‘Bob’:

(2) $\textsc{state}1 \overset{Bob}{\to} \textsc{state}2$

[1] To a first approximation, and for ease of exposition, grammatical mechanisms are here simply taken to be the rules and representations of an adopted formal grammar model, in this case DS (analogous to the second representational level in Marr (1982), that of the algorithms and data structures of some formal model).

is the initial term in the unfolding utterance, and also licenses the hearer to expect an upcoming predicate, among other terms, in the construal of that utterance. Further, the hearer could even try to make a guess as to which Bob is being referred to (assuming a place-holder model of names). Such considerations suggest this kind of processing is goal-driven.

Presuming an intuitive, evidence-driven model of communication,[2] interlocutors are presumed to operate in a goal-driven way, guided by expectations about what is coming up (determined via current evidence from the ongoing interaction). The expectation of a predicate is satisfied by the occurrence of 'left', triggering a transition to the next state:

(3) $\textsc{state}2 \overset{left}{\to} \textsc{state}3$

However, note that prior to the actual occurrence of this item, and the particular information it imparts (e.g. that no other items are required for a saturated proposition), any of a number of other ways of completing the utterance are possible (e.g. a transitive predicate such as *likes*). Another possibility is that of completing the utterance with the idiomatic 'kicked himself', processed as a chunk:

(4) $\textsc{state}2 \overset{kicked\ himself}{\longrightarrow} \textsc{state}3$

Further, we might also wonder whether alternatives such as 'kicked herself', 'kicked' (non-idiomatic), etc, are available, and at what point (see discussion of implementation issues below).

Dynamical modelling involves observing changes to some phenomenon over time at discrete time-steps (Ward (2002)), whereby complex systems can be modelled via "snapshots" of successive stages in the system's progress, these being idealised as successive states of the system, plus transitions between these states (Milward (1994)). As a dynamic process, parsing can be characterised in terms of observed states and transitions between such states. Some parsing research suggests such transitions are made *eagerly* rather than delayed (e.g. until more information is available), with problems arising if a chosen search path turns out to be wrong with respect to upcoming linguistic information (e.g. the so-called garden-pathing of expressions like 'The horse raced past the barn fell') (Sturt and Lombardo (2005)).

So in this more abstract view of incrementality as stepping through stages in the development of an output process of some system, key implementation issues include (e.g. Crocker (2010)): (i) the degree of eagerness, (ii) whether parallel or serial parsing strategies are employed, (iii) whether/not the process is monotonic. This suggests classifying incremental approaches depending on whether/not they exhibit such features. To this end, we aim for a unified account of formulaic and non-formulaic language with a parallel flavour (see Section (2.3) below for details of how we go about this).

2.2 Empirical details

2.2.1 IDIOMS IN DIALOGUE

Let's start with some clearly context-dependent set of phenomena provided by the following elliptical dialogue fragments ((a) – (e)):

[2] Which is to say, interlocutors communicate by cycling through stages of presenting evidence of the information they wish to impart (in the case of speakers), and evidence of the information they have accepted (in the case of hearers).

(5)
A: Have you seen Mary?
(a) B: Mary?
(b) B: No, I haven't.
(c) B: She's out I think.
(d) B: I have, Bill.
(e) B: No, nor Bob.

The interpretation of such fragments requires at least linguistic context (A's question), as well as non-linguistic context. Indeed, context either directly provides linguistic evidence for completion, or else this is more indirect. The examples in (5) above are all examples where the context provides direct linguistic input for processing the fragments, for example, A's original utterance forming the context for B's utterance in (a), providing predicate and subject information.

A complication arises with a more indirect mode of construal occurring in so-called "sloppy" ellipsis. Consider the following examples:[3]

(6)
John took his clients to the cleaners, and so did Bill
"Bill took John's clients to the cleaners" (strict)
"Bill took Bill's clients to the cleaners" (sloppy)

So in (6), we have a strict interpretation in which some content is taken directly from context, but in addition, a sloppy reading, via reusing some aspect of meaning from context (the referent resolving the anaphoric reference of 'his') but nevertheless with distinct content.[4]

Now, placing the idiom in example (6) within an elliptical context illustrates how idioms are compositional and sensitive to items they combine with (see Nunberg et al. (1994)):

(7) John took his clients to the cleaners, but never his shirts

In example (7), substituting an inanimate for an animate nominal (i.e. 'shirts' for 'clients') removes the cues which trigger the idiom, resulting in a switch to non-formulaic processing, despite *the very same representation* being reused for the second clause (see the analysis of ellipsis in Cann et al. (2007)).

Example (7) demonstrates one kind of incrementality of formulaic language, and here is another in the context of dialogue:

(8)
A: Bob took his clients to...
B: the cleaners.
A: Actually, I was going to say to his uncle's restaurant.

Splitting idioms across dialogue turns suggests their processing is as incremental as non-formulaic language. Note, this involves not only the processing of strings, but also the construction of representations.This perhaps would not come as a surprise to those accounts which for the last decade have been arguing for the compositionality of formulaic language (e.g. Nunberg et al. (1994)).

[3] The meaning of 'take NP to the cleaners' being "To take a significant quantity of NP's money or valuables, through gambling, unfavorable investing, fraud, litigation, etc.' (http://en.wiktionary.org/wiki/take_to_the_cleaners).

[4] These two readings are given a formally unitary account through the use of abstraction and higher order unification by Dalrymple et al. (Dalrymple et al. (1991)), but the DS account proposed by Purver et al. (2006) is that in which the actions used in building up interpretation are themselves stored in context and, in the sloppy forms of interpretation, re-used to yield distinct denotational content.

2.2.2 PRE-FABS IN MAZE DESCRIPTIONS

A key focus here is the emergence within dialogue of looser forms of routinised language, such as pre-fab (i.e. conventionalised collocation, see Bybee (2006), more details in Section (2.3) below). Pickering and Garrod (2005) present an influential model of linguistic routinisation, based in large part on patterns discovered in maze task experiments. Now, a robust observation about these experiments is that participants tend to produce a restricted number of so-called *description types*, from more concrete (e.g. *figural* "the sticking out bit at the bottom", *line* "at the end of bottom row") to more abstract (e.g. *path* "two across, one up", *matrix* "2,9") ways of describing/conceptualising the maze.

However, whether these categorisations in fact capture distinct, independent *types* of language is not entirely clear, given the complexity of the various constraints under which the linguistic system operates during dialogue.[5] For example, figurative language might be employed in conditions of greater uncertainty (as proposed in Bavelas (2009)). Consider this in terms of minimal effort vs. maximal effect: a more concrete yet elaborate description (e.g. "the left indicator bit on the right top corner") could be harder to produce yet more likely understood by someone, whereas a more truncated and specialised form (e.g. "two across, one up") will be easier to produce but understanding it may require task-relevant experience.

An experimental task that naturally leads to routinisation is the maze location description task reported in Healey (1997). Here interlocutors communicate with each other in order to identify a set of twenty maze locations. Given the repetitive nature of this task, over the course of interaction, they tend to routinise these descriptions in predictable ways. Consider the following example of one such maze location description:

(i) A: ummm, it's the top right hand corner two down,

(ii) B: two down right so [it's the]

(iii) A: [so it's] kind of three down really but it's only two down,

(9) (iv) B: okay

(v) A: if there isn't one in the top left hand corner,

(vi) B: right,

(vii) A: right?

Here the meaning of 'down' is negotiated quite explicitly, with respect to the specific context, namely the shape of the particular maze A is describing to B. This kind of dialogue routinisation involves linguistically encoding some aspect of non-linguistic context which the interlocutors have made mutually salient through their interaction (using specific linguistic resources useful for picking out bits of the world). Importantly, routinisation adds to this the possibility that future similar interaction will reflect/reuse aspects of this particular interaction. Yet, it has been observed that this trend is not unidirectional, and can in fact reverse, especially when problems arise (e.g. Healey (2008)).

For example, in subsequent interaction, 'some number *down*' is more likely to refer to a point from the top of the actual maze than, say, the putative top of the smallest square the entire maze fits into. Interestingly, A and B might employ 'down' in this way in subsequent interactions with other interlocutors, and some times this may work, particularly if they are doing this against a background

[5] At the very least, there is likely interaction between linguistic and conceptual systems, involving not only preferred ways of talking in interaction, but also preferred ways of conceptualising mazes.

community-wide set of interactions. However, if in egocentric fashion they wrongly presume the same routine should work with any subsequent interlocutor,[6] they may well run into difficulty that causes such routines to be ineffective. We are here interested in the question of what might they do next. One answer is that they will typically switch to other, more computational (rather than memory-based) kinds of linguistic processing, which we discuss elsewhere (e.g. Gargett (2010)).

2.3 Simulating routinisation

Let's bring out some issues more distinctly by considering the life-cycle of a linguistic routine, adapting the mechanistic approaches in Logan (1988), Anderson (1992), Rawson (2004). Imagine a (perhaps unusual) adult speaker of English encountering the following for the first time:

(10) Bob pulled some strings at work.

Given the capacity to construct a representation of this string, complete with a suitable construal (e.g. determining that Bob does not physically handle string for a living), with repeated exposure to this phrase our adult speaker can store this representation which she can later re-use. For this speaker, 'pull strings' has become a pre-fab (i.e. conventionalised collocation, see Bybee (2006)) with the sense of (broadly) "manipulate". In line with our dual processing perspective, we further assume that updating proceeds via either the immediate use of lexical actions for constructing representations from scratch, or else the reuse of stored representations outputted from previous use of lexical actions. Following Logan, such interaction between competing options can be modelled as a race between them to effect update.

At the beginning of the cycle of routinisation, rules may be favoured in such races due to their generality (cf. Logan (1988), Rawson (2004)). But over time, richer contexts accumulate for the output representations (via the surrounding words, etc) and within which processing takes place, so that specialisation of these representations to specific contexts in which processing *typically* occurs, leads to their being favoured over rules.[7] The intermediate term is marked by a period of shifting between one kind of processing and another, and over-specialisation of semantic structures can lead to these failing to respond in novel contexts. Importantly, *computational processing is still available* when such failure occurs, and may in fact reappear in such cases.[8]

However, what about the longer term? It is here that rules make a resurgence, in the form of routines, or complexes of actions. Idioms are a good example of this, and various earlier accounts have looked at the relative compositionality of these (e.g. Nunberg et al. (1994)). An implication of Logan's model seldom taken up is that computational processing never actually abandons the race, and might even somehow gain a competitive edge (even after some period of dominance by memory-based processing).[9] We would suggest that various ways of effecting computational efficiency, such as production compilation (see Taatgen et al. (2008)) could provide computational

[6] Or even if it is more mechanistic than this, say, triggered by cues within memory.

[7] We are of course talking about a probabilistic phenomenon, and a probabilistic version of the DS parser is currently underway.

[8] Our approach to coordinating the two processes in this way, albeit indirectly, attempts to extend the model in Logan (1988) for linguistic purposes, and is in line with suggestions in the linguistics literature (e.g. Wray (2002) on holophrastic processing operating in tandem with more analytical processing, Rawson and Middleton (2009) on novelty and automaticity in text comprehension).

[9] Recently, Rawson and Middleton (2009) has discussed Logan's theory in similar terms (interestingly, by way of proposing an account of the response of automatic processing to novelty).

processing with the necessary boost to eventually win out over memory-based processing, thus linking linguistic routinisation with models of automaticity in cognitive psychology. Production compilation essentially involves linking together otherwise sequential, separate productions (each with their own triggering conditions and output effects) into a complex unit with a single triggering condition and combined output effects.[10]

2.4 Previous accounts

There is extensive work on routinisation (the process leading to the formation of routines) throughout the cognitive sciences. Common to many such theories is the idea that routines arise from practice, becoming points of stability in the face of contextual exigencies, typically expressed as a list of various properties, such as their being: (i) rigid in form, (ii) somewhat truncated, (iii) relatively non-compositional, (iv) yet highly specialised to the context (Ruh et al. (2005), Chernova and Arkin (2007)). Such *repetition effects* are familiar enough from everyday experience, and this list only hints at the complexity involved with routinisation. Within cognitive psychology, routinisation surfaces chiefly in models of automaticity (e.g. Logan (1988), Anderson (1992), Bargh (1992)), ranging from the more common property listing accounts, to the recently emerging mechanistic approaches (recently critiqued in Rawson (2004)). Within linguistics, routinisation surfaces chiefly in the vast literature on formulaic language (e.g. Bolinger (1976), Jackendoff (1997), Erman and Warren (2000), Sag et al. (2002)), much of which employs property-listing of one sort or another as a key component of their modelling strategy.

Despite detailed accounts of routinisation phenomena within dialogue (e.g. Kuiper (1996), Aijmer (1996)), these also involve extensive listing of properties, with little formal and computational work in this area. Pickering and Garrod (2005) present an explicit, testable model of linguistic routinisation.[11] However, their account lacks the formal details we require, despite their suggestions for adapting ideas from Jackendoff (2003), the latter being a form of the property-list approach to routinisation, which we are trying to avoid.

Rawson (2004) presents a non-property-listing alternative, demonstrating the usefulness of mechanistic approaches to linguistic modelling (see also Logan (1997)), comparing the rule-based account of Anderson (1992) to memory-based models of automaticity, such as that of Logan (1988), in a series of reading experiments. While the results were complex, with memory-based processing clearly driving the bulk of speed-up effects associated with practice (so that there were typically reduced reading times for the same texts, but elevated times for novel texts), she did find evidence for some involvement of rule-based processing, particularly in response to novel items. In our account, rather than devising our own models and possibly re-inventing several wheels, we look to such models from psychology to provide the basis for our account of linguistic routinisation.

To capture earlier observations in Section (2.2) regarding the relative incrementality of linguistic routines, as well as proposals for how such routines might emerge in Section (2.3), we suggest that the dynamics of routinisation stems from the race between memory-based vs. rule-based processes. Thus, the shift from the mid-term where memory-based processing holds sway, to the longer-term

[10] Schematically, compiling two productions triggered by CONDITION1 and CONDITION2, and which lead to effects UPDATE1 and UPDATE2, respectively, may lead to a rule with compound effects, but which does not require CONDITION2 as a trigger, something like:

$$IF[\texttt{CONDITION1}]\ THEN[\texttt{UPDATE1} \wedge \texttt{UPDATE2}]$$

[11] Within Dynamic Syntax, Bouzouita (2008) proposes a way of formally modelling aspects of their account.

where rule-based processing, via production compilation, becomes more competitive, is externally verified by proposals from specific cognitive psychology accounts (chiefly Anderson (1992), Logan (1988), Rawson (2004)). In this way, we formulate a novel account of the emergence of formulaic language, in terms of the dynamics of the linguistic system, by extending the framework of Dynamic Syntax (DS, e.g. Kempson et al. (2001), Cann et al. (2005)), especially as this has been applied to dialogue (e.g. Purver et al. (2006)). We show how this yields not only a new explanation of the continuum from the context specialisation of pre-fabs to selectivity of discontinuous idioms, but further, we can employ the DS model of language processing as applied to dialogue phenomena, to potentially model all manner of formulaic dialogue phenomena.

3. Formal modelling

3.1 Formal details of the model

Informally, the Dynamic Syntax (DS, Kempson et al. (2001), Cann et al. (2005)) account of how contextual information can be incorporated *as it arises* with linguistic information during dialogue, has three main characteristics: fully incremental processing, modelling update as cycles of the enrichment of underspecified representations, plus parity of parsing and production (formal details for each below).

DS provides a fully incremental parsing model, with update modelled as transitions between succeeding parse states, essentially, enrichment of partial tree structures. Parsing is then the sequence of pairings of natural language strings of terms s with the logical formula R representing the semantic structure of those terms:

(11) $\{\langle s(i), \mathrm{R}_i\rangle, \langle s(i+1), \mathrm{R}_{i+1}\rangle, \ldots\}$

Thus, R_i results from parsing $s(i)$. More generally, these successive parse states are modelled as triples

(12) $\langle PT, F_s, F_a\rangle$

of (partial) tree structures PT, function F_s mapping partial tree structures to items of the formal language, and function F_a mapping actions (from sets of actions, A) for transition between trees to pairs of partial trees.

Structured logical formulae representing (predicate-argument) content are mapped to decorated (binary) *finite partial trees*. Thus, parsing the string 'Bob left' results in the unreduced lambda term $((\lambda x Left(x)), Bob)$, represented by the following decorated finite partial tree which includes the decorations for the topmost, $Ty(t)$ mother node (both formula Fo and typing information Ty included):[12]

$$[_0 Fo(Left'(Bob')), Ty(t)[_0 Fo(Bob'), Ty(e)][_1 Fo(\lambda x Left'(x)), Ty(e > t)]]$$

Dominance relations between nodes specify tree structure, from more local *argument-daughter* ($\prec_0$) and *function-daughter* ($\prec_1$) relations between immediate neighbour nodes, to more global relations holding over collections of nested sub-trees (neighbours of neighbours of nodes). Another crucial component of the framework is a so-called LINK mechanism for constructing pairs of trees, effectively conjoining the information contained in trees so linked (more below).[13]

Transitions from one PT to the next (in the sequence of updates) are effected through three main kinds of actions:

- **Lexical actions**, a finite set of incremental actions associated with every word in the language, the occurrence of this word effectively triggering this instruction set,
- **Computational actions**, a finite set of actions of a more general nature for building linguistic structure, which are triggered independently of the occurrence of individual words, and finally

[12] Following the bracketed format of Meyer-Viol (2001) signifying predicate-argument tree structure, where outermost $[_0 ...]$ is the top level, subsequent $[_0 ...]$ enclose argument daughters within the tree, and any $[_1 ...]$ enclose function daughters.

[13] Linking symbolised via an $\langle \mathrm{L} \rangle$ modality.

- **Pragmatic actions**, a finite set of actions which operate to connect contextual information with the current *PT* under construction.

Each of these "macro"-level actions are actually composed of lower-level constructional actions for creating new nodes, decorating nodes, substitution at nodes, or else pointer movement.[14] Regarding this latter, the **pointer** device (symbolised by $\diamondsuit$) is central to modelling transitions, singling out the specific node in the tree which is the current focus of update, pairing this node with the partial tree currently under construction (Kempson et al. 2001, p. 275),[15] this pairing then being an element of the set of **Pointed Partial Trees** (*PPT*s).

An important feature of the eventual dialogue account (detailed below) is that, while parsing may begin in the simplest cases with the so-called axiom, $?Ty(t)$, an initial state expressing the requirement to simply build a propositional object, modelling sequences of contributions by succeeding interlocutors in dialogue may potentially require starting from any point along the partial order of states (more below). Indeed, parsing could then start from a context provided by the previous speaker's utterance (as we will see).

Summarising the presentation so far, the basic units of the framework (Kempson et al. 2001, p. 269) are decorated *PPTs*, described using a language pairing elements from the set ***F*** of labels or features, like Ty ("type"), Fo ("formula") or Tn ("tree node"), with elements from the set ***D*** of formulae ("decorations"), like $e > t$, $\lambda x Left'(x)$ or 01, the latter effectively values for these features (cf. attribute value matrices), but also including a set ***MV*** of metavariables (details of these latter below).[16]

The enrichment of underspecified *PPTs* is crucial to the account of goal-directed information growth for any dimension of tree structures and decorations (formula and type information). Three kinds of underspecification are involved, structural, formula, and type value, the goal-directedness of enrichment/update modelled explicitly in terms of *requirements*, so that for any label X, adding a requirement $?X$ imposes a goal to establish X. All aspects of underspecification have an associated requirement for update, so that requirements may take the form $?Ty(t)$, $?Ty(e)$, $?Ty(e > t)$, $?\langle\downarrow\rangle Ty(e > t)$, $?\exists x Fo(x)$, $?\exists x Tn(x)$, etc. Pronouns illustrate formula underspecification, for example, the pronoun *he* licensing projection of a metavariable $Fo(\mathbf{U}_{Male'(\mathbf{U})})$ of $Ty(e)$ with requirement $?\exists x Fo(x)$ (the latter a requirement for a fully specified formula). Such metavariables are replaced by a SUBSTITUTION process from a term available in context. Names too are modelled as projecting a metavariable, so that the occurrence of "Bill" projects a metavariable annotated as $Fo(\mathbf{U}_{Bill'(\mathbf{U})})$, with instruction to construct a LINK transition to a linked tree of topnode $Ty(t)$ decorated with the formula value $Bill'(\mathbf{U})$, characterising the predicate *being named Bill*, this constituting a constraint on the logical constant to be assigned as a construal of the use of that name in the particular context. All such metavariable-based terms are then enriched via the pragmatic action of Substitution, and which may itself be suitably constrained (e.g. see discussion in Purver et al. (2006)).

[14] Such basic actions can combine into complex ones. Suppressing some formal details (for which see (Meyer-Viol 2001, p. 171), (Kempson et al. 2001, p. 308)), basic actions include ones like *make*, *put* and *go*, which are very low level actions for constructing and decorating nodes, and moving between sub-trees under construction. Macros of actions can be assembled from basic actions, chained together with the Dynamic Logic concatenation operator **;**. For example, the procedure for moving to a particular node to decorate it with some fact or requirement is specified as: $make(R); go(R); put(\phi); go(R^{-1})$.

[15] Thus, $Tn \prec_i T'n'$ if $T = T'$ and $n \prec_i n'$ (for some i belonging to the set of dominance relations presented above).

[16] This is somewhat simplified, for details see Kempson et al. (2001).

3.2 Parity, dialogue and bi-directional grammars

In DS, interactional dynamics during dialogue can be captured directly in terms of the core resources of the grammar, whereby the transition between speakers is modelled as the transition between parsing states. DS models generation as driven by the same underlying processing mechanisms as parsing, in common with many bi-directional grammars (e.g. Shieber (1988)), thereby ensuring that generation is as incremental and goal-directed as parsing. In DS, a hearer builds a succession of partial parse trees representing components of the speaker's utterance. Speaking is modelled in the same way, with the addition of a *goal tree* representing what the speaker wishes to say.[17] Thus, a hearer can switch to speaking immediately and work from the same representation for both. Further, parsing and generation can start from any point (*Axiom* being only one possibility), so that interlocutors may in fact work off the immediately preceding context, or else from some store of structures, both strategies being crucial for modelling routinisation in dialogue. We can then model the switching back-and-forth between parsing and generation in a fully incremental way, this providing a mechanism for the emergence of *parity* during dialogue.[18] Such tight coupling of goal-directed parsing and generation captures how interlocutors can make micro-adjustments in and through the interaction itself, thus directly modelling how the emergent dialogue is shaped on the fly through such fine-grained interaction between interlocutors.

More formally, Purver et al. (2006) define a parse state P as the triple $\langle T, W, A\rangle$, with T a (possibly partial) tree, W the associated sequence of words, and A the associated sequence of lexical and computational actions. At any point in the parsing process, the context $\mathcal{C}$ for a particular partial tree T in the set P can be taken to consist of the results of previous parses $\{\ldots, \langle T_i, W_i, A_i\rangle, \ldots\}$. Later we draw on this for our model of tripartite minimal exchanges, with the context consisting of a parse state P' resulting from some utterance initiating the exchange, any partial trees established in subsequent parsed fragments associated with some clarification or extension of aspects of P', and finally partial trees established for the response of the initiating speaker.

The generation model consists of incremental (word-by-word) parsing, and lexicon search for words which provide appropriate tree-update relative to a *goal tree* (what it is the speaker wishes to say), and through this process speakers produce the natural language string associated with the goal tree (Purver et al. (2006)). A generator state G is thus a pair $\langle T_G, X\rangle$ consisting of:

(i) a goal tree T_G, and

(ii) a set X of pairs $\langle S, P\rangle$, S a candidate partial string, P an associated parser state.[19]

The context-dependence of generation comes to the fore where lexical search can include context wherever possible, the effect being to reduce the production task. Such reuse of context drives coordination between speakers via generation as well as parsing, the dynamics of this process arising indirectly out of the interaction. The following example demonstrates reuse for simple question-answer:

[17] Being a so-called tactical generation model only, the question of how this is arrived at is put on hold for now.

[18] Within cognitive science, parity involves sharing representations across processes within the same individual, such as where representations for both speaking and hearing are built using the same underlying mechanism (Pickering and Garrod (2004)). Bi-directionality is then an important ingredient within the present account of how parity across understanding and generation systems is achieved, and is central to our modelling of dialogue phenomena based on the DS grammar model.

[19] As defined in Section (3.1) above.

(13)
A: Who left?
B: Bill did.

From a DS perspective (e.g. Cann et al. (2007)), for their answer, B reuses the context provided by interpreting A's question (details below). Note that for A to understand B's question, A needs to understand it against the context of their own initial utterance.

Moreover, the combined incremental, goal-directed framework enables processing linguistic strings **sub-propositionally**, making the DS dialogue model radically different to established approaches to dialogue modelling (e.g. Traum (1994), Asher and Lascarides (2003)) which retain commitment to rather coarse-grained units of analysis, typically propositions.[20] DS models the processing of fragments in contexts as steps toward the construction of a fully propositional term, so that grounding (the process whereby interlocutors arrive at mutual understanding of their utterances) is driven sub-propositionally. There is ample evidence that during dialogue people quite happily interact sub-propositionally, such as:[21]

(14)
TRAINS91, Dialogue 1.2, Lines 12.1 to 18.1
S: okay
: so we'll say
M: send
S: E2
: I guess
: ... from Elmira
M: tshh
: yeah
S: and send them ...
M: to Corning
S: to Corning

Here, M and S switch turns, seemingly together constructing the eventual proposition. Of course, each must understand the whole and where their own contribution fits, so each must separately entertain some proposition commensurate with that expressed by the final utterance (or if not, then they would know they were mistaken). For DS, such sub-propositional units are grammatical,[22] and interlocutors may be not only working toward constructing a propositional term (i.e. the output of a complete tree), they may also be engaged in more partial interpretive work and processing material at sub-propositional levels below this.

Our modelling strategy for dealing with the complex of dialogue phenomena is to recast this in terms of a minimal exchange model, focusing directly on the *initiative* which dialogue agents display in the following manner:

[20] Although others have recently claimed to also be modelling sub-propositionally, e.g. Matheson et al. (2000), Poesio and Rieser (2010).

[21] Note that the format of the following dialogue from the TRAINS91 dialogue corpus (http://www.cs.rochester.edu/research/speech/trains.html) is simplified for expository purposes. ... represents an extended pause of noticeable duration.

[22] In the sense that grammaticality is relative to context, if when combined in suitable ways with some information from context the result is a grammatical unit (for explicit definition of grammatical, outside our scope here, see Cann et al. (2007).

(i) **A context tree**: the START STATE - this is the tree the initiator is starting their parse from

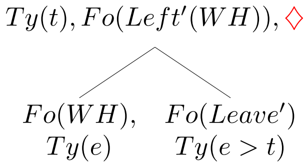
(ii) **A goal tree**: the FINAL STATE - the initiator will end up with a tree matching this, if grounding is successful

(iii) **A construction tree**: an INTERMEDIATE STATE bridging (i) and (ii), and which replaces the context tree after every update step

This covers a range of core dialogue phenomena we are interested in modelling, such as clarifications, reformulations, and corrections, and we have demonstrated (e.g. Gargett et al. (2009)) that our approach accounts for such core dialogue phenomena. However, it is also important to point out that we are not currently explicitly modelling the content of such exchanges, and we are not claiming to have (as yet) a complete dialogue model in this sense.

Let's consider example (13) using a more schematic presentation of these three, inter-related kinds of trees. In Table (1), the context for step 1 is the following already formed tree structure resulting from B's parse of A's question, including both the subject node decorated with WH, and predicate node decorated with $Left'$:

(15)



In our example, the subject node is updated with information licensed by occurrence of 'Bob', this reflecting B's analysis of A's question (plus relevant wider knowledge that Bob is the correct answer in this case). Next, occurrence of 'did' licenses update in accordance with the following lexical actions (see Purver et al. (2006) for details):[23]

(16)

'Do'	IF	$?Ty(e > t)$
	THEN	$put(Fo(\mathbf{U})); put(Ty(e > t))$
		$put(?\exists x.Fo(x))$
	ELSE	ABORT

Having uttered 'did', the next update step 2 in Table (1) requires specifying the metavariable $Fo(\mathbf{U})$ decorating the predicate node, by enabling reuse of the formula decorating the $Ty(e > t)$ node of the tree in (15).

The dynamics of the process stem from how construction trees iteratively become in turn context trees for subsequent construction trees, with a progression of construction trees recycled as context. These cycles of contribution-response-contribution enable narrowing of focus to a specific point in the representation under construction, providing interlocutors the opportunity for quite fine-grained adjustments of understanding.[24]

[23] For convenience, we are ignoring tense information.

[24] So that such continual switching of speakers does not necessarily indicate misunderstanding and breakdown of communication.

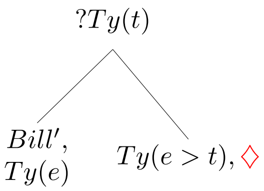
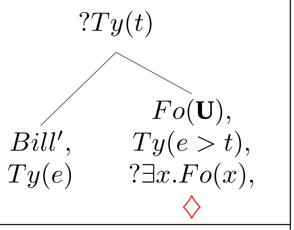
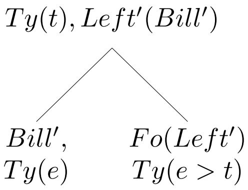
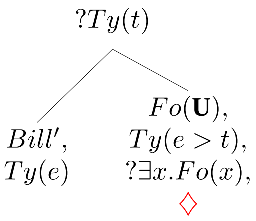
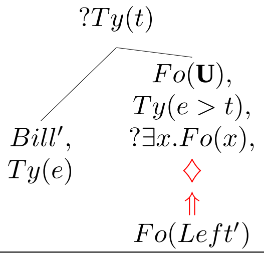
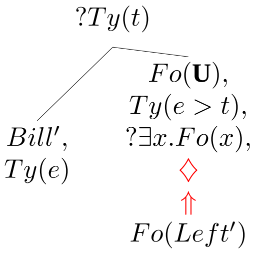
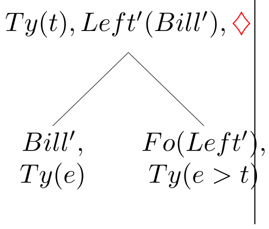
CONTEXT$_B$:	CONSTRUCTION$_B$:	GOAL$_B$:
STEP 1  	 $?Ty(t)$ $Bill'$, $Ty(e)$ $Fo(\mathbf{U})$, $Ty(e > t)$, $?\exists x.Fo(x)$, $\Diamond$	 $Ty(t), Left'(Bill')$ $Bill'$, $Ty(e)$ $Fo(Left')$ $Ty(e > t)$
STEP 2  $?Ty(t)$ $Bill'$, $Ty(e)$ $Fo(\mathbf{U})$, $Ty(e > t)$, $?\exists x.Fo(x)$, $\Diamond$	 $?Ty(t)$ $Bill'$, $Ty(e)$ $Fo(\mathbf{U})$, $Ty(e > t)$, $?\exists x.Fo(x)$, $\Diamond$ $\Uparrow$ $Fo(Left')$	
STEP 3  $?Ty(t)$ $Bill'$, $Ty(e)$ $Fo(\mathbf{U})$, $Ty(e > t)$, $?\exists x.Fo(x)$, $\Diamond$ $\Uparrow$ $Fo(Left')$	 $Ty(t), Left'(Bill'), \Diamond$ $Bill'$, $Ty(e)$ $Fo(Left')$, $Ty(e > t)$	

Table 1: Tripartite model of minimal exchanges: simple question and answer. Note that the predicate information $Fo(Left')$ substituted into the construction tree in Step 2, is provided by B's own representation of A's previous utterance of 'Who left?' (see discussion of (15) for details).

4. From prefabs to idioms in language use

This section attempts a formal demonstration of the approach to routinisation suggested in Sections (2.3) and (2.4), whereby pre-fabs emerge within focused dialogue over the intermediate term, with increasing efficiency of lexical actions over the longer term. Another source of contact with our hybrid modelling approach in the literature are the results from the psycholinguistic investigation of idioms by Sprenger et al. (2006). Their model is hybrid between idioms being unitary, at a conceptual level, and compositional, at a lexemic level. Thus, the particular lemma ‘bucket’ will be activated for both the literal and idiomatic meanings of ‘kicked the bucket’, but the source of activation of the lemma is different in each case. However, the formal model in Sprenger et al. (2006) focuses on linking distinct syntactic and semantic/conceptual levels, whereas we focus on the additional aspects of the dynamics of such links in contexts of interaction.

In what follows, we provide details of our extension to the core DS framework, incorporating hybrid rule-based and memory-based modelling, and formally detailing a dual process model of the emergence of formulaic language. We finish by drawing out the other thread pursued in this paper, that of incremental context-dependence of formulaic language, demonstrating this in a dialogue setting. Note that while the analyses in what follows involve constructed data, we are currently extending our analyses to actual dialogue data (such as that reported in Healey (1997), discussed in Section (2.2.2) above).

4.1 Extending the core DS account

Our aim here is to extend the DS framework, by modelling lexical entries as nodes within a network of such entries, consisting of tuples $\langle w, T, A \rangle$ of phonological information w, semantic structure T and lexical actions A. These nodes are accessed primarily through recognising/producing sequences of phonological strings $w_i w_j w_k \ldots$, so that nodes may themselves consist in part of string sets such as those for ‘kick’, ‘kick himself’, ‘kick herself’, ‘kick themselves’, etc (more details on the structure of lexical entries below). This leads to transitions between states being effected either via lexical actions, triggering building of tree structure by basic actions (as noted in Section (3.1) above), or else by directly contributing (previously stored) structure at the appropriate place in the unfolding tree structure.

In what follows, we first consider the formal modelling issues which formulaic language presents, then suggest how fixed idiomatic forms may emerge from relatively less fixed pre-fabs. Finally, we will show how relevant lexical entries can be extended, before finishing this subsection with a schematic presentation of our formal proposal for extending DS lexical entries to lexical nodes.

4.1.1 BASIC LEXICAL ENTRIES

As mentioned, nodes may consist of sets of phonological strings, together with some associated lexical actions, such as (in all following examples we ignore tense for convenience):[25]

[25] Note the use of relational operators for various purposes, including locating the current node within the larger tree structure, such as $\langle \downarrow_1 \rangle Ty(e > t)$, which specifies that at the function daughter below the current node is one of $Ty(e > t)$, or even for pointing out a direction, as in $go(\langle \uparrow_0 \rangle)$, which is to say, “go up the argument (’0’) branch from here”.

(17)
$$\begin{array}{lll} \textbf{kick} & \text{IF} & ?Ty(e>t) \\ & \text{THEN} & make(\langle\downarrow_1\rangle), go(\langle\downarrow_1\rangle), \\ & & put(Fo(\lambda xy.Kick'(xy)), Ty(e>(e>t))) \\ & & go(\langle\uparrow_1\rangle), make(\langle\downarrow_0\rangle), go(\langle\downarrow_0\rangle), put(?Ty(e)) \\ & \text{ELSE} & \text{ABORT} \end{array}$$

(18)
$$\begin{array}{lll} \textbf{Xself} & \text{IF} & ?Ty(e>t) \\ & \text{THEN} & put(Fo(\mathbf{U}_{Anaph}) \wedge Ty(e)) \\ & \text{ELSE} & \text{ABORT} \end{array}$$

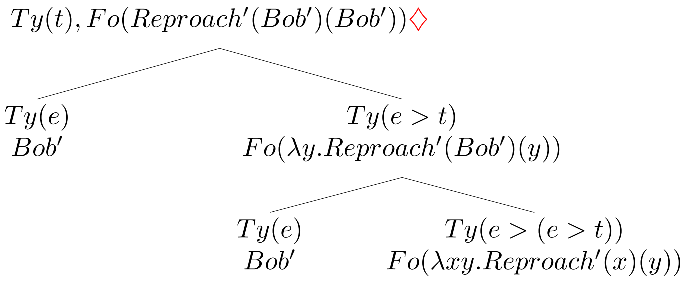
Note that the operation of the reflexive also relies on a special local version of the pragmatic action Substitution, which enriches the metavariable decorating the object node with the formula information decorating the subject node (following the analysis in Cann et al. (2005)).

Now, applying the lexical entries in (17) and (18) capture non-idiomatic meaning only - but consider the occurrence of idiomatic ‘Bob kicked himself’(=“Bob reproached himself”). Recall from Section (2.1) the following schematic sequence of transitions for this idiomatic expression, repeating here the previous analysis that parsing this expression involves stepping through one less state than the non-idiomatic expression:

(19) $S_1 \overset{Bob}{\rightarrow} S_2 \xrightarrow{kicked\ himself} S_3$

Upon completing a parse of this sequence we might arrive at the following final state (the formulation of names here is simplified, but see Gargett (2010) for details):

(20)


$Ty(t), Fo(Reproach'(Bob')(Bob'))\diamondsuit$
$Ty(e)$
Bob'
$Ty(e>t)$
$Fo(\lambda y.Reproach'(Bob')(y))$
$Ty(e)$
Bob'
$Ty(e>(e>t))$
$Fo(\lambda xy.Reproach'(x)(y))$

We need to show how our model captures the entire sequence of updates leading to (20) by providing lexical actions for the idiomatic expression. Rather than simply stipulating these directly, the advantage of the dual process account we are taking is that we are able to model the process underlying the emergence of these entries (specifically, the memory-based processing of these in terms of their access and retrieval).

4.1.2 THE EMERGENCE AND STORAGE OF SEMANTIC STRUCTURES

We propose that the sequence of the verb *kick* plus reflexive *himself* becomes routinised over time, with storage of the semantic structure outputted at the associated parse states S_2 and S_3, essentially the tree in example (20). Over time, there will be an accumulation of many instances of such output, for example, both of the following:

(21)
$$\begin{array}{l}
\{Tn(0), Fo(\mathit{Reproach}'(\mathit{Mary}')(\mathit{Mary}')), Ty(t)\}\\
\{\langle\uparrow_0\rangle\, Tn(0), Fo(\mathit{Mary}'), Ty(e)\}\\
\{\langle\uparrow_1\rangle\, Tn(0), Fo(\lambda x.\mathit{Reproach}'(\mathit{Mary}')(x)), Ty(e > t)\}\\
\{\langle\uparrow_0\uparrow_1\rangle\, Tn(0), Fo(\mathit{Mary}), Ty(e)\}\\
\{\langle\uparrow_1\uparrow_1\rangle\, Tn(0), Fo(\lambda xy.\mathit{Reproach}'(y)(x)), Ty(e > (e > t))\}
\end{array}$$

(22)
$$\begin{array}{l}
\{Tn(0), Fo(\mathit{Reproach}'(\mathit{Bob}')(\mathit{Bob}')), Ty(t)\}\\
\{\langle\uparrow_0\rangle\, Tn(0), Fo(\mathit{Bob}'), Ty(e)\}\\
\{\langle\uparrow_1\rangle\, Tn(0), Fo(\lambda x.\mathit{Reproach}'(\mathit{Bob}')(x)), Ty(e > t)\}\\
\{\langle\uparrow_0\uparrow_1\rangle\, Tn(0), Fo(\mathit{Bob}), Ty(e)\}\\
\{\langle\uparrow_1\uparrow_1\rangle\, Tn(0), Fo(\lambda xy.\mathit{Reproach}'(y)(x)), Ty(e > (e > t))\}
\end{array}$$

By virtue of being stored locally, these largely similar semantic structures may be related via a process of *tree abstraction*. Here we adapt a proposal by Wilfried Meyer-Viol to formalise how such abstraction might proceed.[26] Recall that update via the transition function involves moving the unfolding tree structure along the partial order $\leq$ from less to more specified states. The basic idea of abstraction involves moving *backwards* along $\leq$, effectively unwinding the complete tree to an earlier point at whichever nodes the information of the source trees differs, replacing any formulae at these nodes with metavariables and requirements for an Fo. Thus, the structures in examples (21) and (22) can be abstracted as follows:

(23)
$$\begin{array}{l}
\{Tn(0), ?Ty(t)\}\\
\{\langle\uparrow_0\rangle\, Tn(0), Fo(\mathbf{U}), ?\exists y Fo(y), ?Ty(e)\}\\
\{\langle\uparrow_1\rangle\, Tn(0), ?Ty(e > t)\}\\
\{\langle\uparrow_0\uparrow_1\rangle\, Tn(0), Fo(\mathbf{U}), ?\exists x Fo(x), ?Ty(e)\}\\
\{\langle\uparrow_1\uparrow_1\rangle\, Tn(0), Fo(\lambda xy.\mathit{Reproach}'(y)(x)), Ty(e > (e > t))\}
\end{array}$$

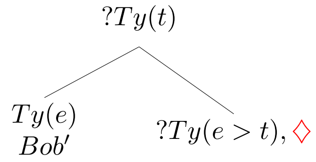
The resulting abstracted tree in (23) is not an expected output of parsing some utterance in English: although the verb node is fixed and decorated with a fully specified formula, the subject node is fixed yet also decorated by an underspecified formula expression.[27] Such abstraction essentially pinpoints the similarities in these structures, and might be expected from structures being stored locally within some network of such structures, as a memory-based effect (discussed further below). The metavariables here represent kinds of abstractions over the places they are the holder for. These places were originally occupied by items which had some similarity with respect to each other, in most general terms (employing featural definition of categories) this involved [+animacy], and more specifically, it involved [+human].

Note that metavariables at both subject and object nodes in (23) are identical, and this captures the identity of the formulae at these nodes resulting from occurrence of the reflexive (see (18) above). Yet, as it stands, our analysis is incomplete, since we need to derive a structure which can be employed incrementally at the appropriate point in the parse. Recall the schematic of this sequence in (19): after the occurrence of 'Bob', the parse state is as follows:

[26] Personal communication, Ms.

[27] A node can of course be unfixed and decorated with underspecified formula, as occurs in the case of left-dislocation, see Kempson et al. (2001) for details.

(24)



Proceeding to the next update step via stored semantic structure requires retrieval of a sub-tree with topmost node of type $?Ty(e > t)$. However, the tree in example (23) contains a subject node, which is superfluous since we require only information stored for the combined predicate and object (i.e. in DS terms, the sub-tree with topmost node of type $Ty(e > t)$). At this point, we do not have a detailed theoretical account of how such superfluous information might be discarded.[28] For now, we simply prune structure above the $Ty(e > t)$ node, with the result as follows:

(25)
$$\{\langle\uparrow_1\rangle Tn(0), ?Ty(e > t)\}$$
$$\{\langle\uparrow_0\uparrow_1\rangle Tn(0), Fo(Fo(\mathbf{U}_{Anaph}) \wedge Ty(e))\}$$
$$\{\langle\uparrow_1\uparrow_1\rangle Tn(0), Fo(\lambda xy.Reproach'(y)(x)), Ty(e > (e > t))\}$$

The final version in (25) models the structure that would provide update of the partial tree in (24). Note that this additional step extends the original abstraction operation by decorating the object node with a reflexive anaphor (to which can be applied the special local version of Substitution advocated by Cann et al. (2005) for reflexives), this being triggered during the pruning process by identical metavariables occurring on subject and object nodes of (23).

4.1.3 EXTENDING LEXICAL ENTRIES

Crucial to our proposed account of the dynamics of the emergence of formulaic language (as discussed in Section (2.3) above) is the competition between lexical actions and semantic structures to update the tree currently under construction, in response to occurrences of the phonological string. Thus, the occurrence of ‘kick himself’ sets in train a race between the processes underlying both lexical actions and semantic structure, to produce the material which updates the unfolding tree structure through the sequence of transitions represented above in (19). Depending on the outcome of this race, it may be the abstracted semantic structure in (25) which updates the unfolding tree, or it may be the lexical actions triggered by ‘kick’ followed by those triggered by ‘himself’. We consider this account to be essentially a linguistic recasting of Logan’s model of automaticity (see Section (2.4) for details).

We saw in Section (4.1.2) how semantic structures might emerge and provide structure for updating the tree currently under construction. We propose that a semantic structure suitably optimised (such as after undergoing the pruning process described in Section (4.1.2)), would win the race to provide update. Indeed, the degree of specialisation of the semantic structure for the particular context, leads to it taking over update of the parse state.

Now, keeping with our linguistically inspired extension of Logan’s model, the only way that the lexical action can become competitive again, and thus take over processing in response to the occurrence of, say, idiomatic ‘kick himself’ or ‘kick herself’, is if somehow there is a reduction in processing time. An obvious mechanism for this is the use of procedural compilation in various models of working memory (e.g. ACT-R, Taatgen et al. (2008), discussed further below).

[28] For example, some way of restricting extraction to that information from context which is “relevant” is likely needed, although we do not currently have a way of operationalising such a notion of *relevance of information*.

We propose then that lexical rules may undergo their own form of optimisation in order to become once again competitive in this process, in particular through a process of procedural compilation, whereby lexical actions are chained together to provide complexes of such actions. The resulting complex lexical action for idiomatic ‘kick oneself’ is as follows:

(26)

kick oneself	IF	$?Ty(e > t)$
	THEN	$make(\langle\downarrow_1\rangle), go(\langle\downarrow_1\rangle),$ $put(Fo(\lambda xy.Reproach'(x)(y)), Ty(e > (e > t))),$ $go(\langle\uparrow_1\rangle), make(\langle\downarrow_0\rangle), go(\langle\downarrow_0\rangle),$ $put(Fo(\mathbf{U}_{Anaph}) \wedge Ty(e))$
	ELSE	ABORT

Note that the lexical entry in (26) for ‘kick oneself’ contains additional procedures for decorating the tree node with the 010 address (the object node),[29] these additional procedures being deployed by extending the formalism for lexical actions with the AND structuring device for bundling together procedures dealing with both ‘kick’ as well as ‘oneself’ into a more complex lexical action for ‘kick oneself’.

4.1.4 INTERIM SUMMARY

In summary, the mechanism we are proposing for the emergence of linguistic routines is quite indirect, driven by the *competition* between semantic structure and lexical actions.[30] It should be emphasised that of course DS provides the possibility that either actions or structures can be used as possible updates for the unfolding tree structure, so of course we could well have represented (23) in terms of actions rather than structure. However, what we are seeking here is a way of using these formal mechanisms to model processes underlying the patterns we see in dialogue in cognitive terms. To this end, these structures are employed to suggest a memory-based account for how outputs of the parser may be stored and reused on subsequent occasions (perhaps even over the much longer term in the case of stable forms of formulaic language).

Further, we have shown that, despite idioms involving skipping over some state/s, rather than stepping through each and every possible individual state (recall discussion in Section (2.1)), the process is still incremental, just that there are overall fewer actual transitions between states, this being the effect of more the complex lexical actions for idioms.[31]

4.1.5 ARCHITECTURE OF LEXICAL NODES

Now, while our model integrates rules and stored structures, both are essentially computational.[32] The final component in our dual process account is to fully implement retrieval of structures in

[29] Reading the address from right to left, and thereby “back up” the tree: argument daughter (0) of function daughter (1) of the root node (0).

[30] Note how close we are to the account in Kecskes (2008) of the dynamics of linguistic meaning arising from the interplay between current knowledge and growing experience.

[31] In fact, as pointed out by an astute reviewer, as this entry stands, it may suggest that no concept of ‘kick’ is accessed when invoking the entry in (26). This is not intended by our account, although at this stage we are unable to directly address this issue. As noted immediately below, we intend revisiting these and other issues, in particular in light of a recent model proposed by Sprenger et al. (2006).

[32] Via lexical procedures, on the one hand, and via tree abstraction, on the other.

a properly memory-based fashion, in order to simulate competition between update by either rule-based computation or via retrieval of structure. In Gargett (2010), we propose modelling this competition in terms of retrieval of semantic structures conditioned through the manipulation of activation weights, and also in terms of utilities assigned to productions that govern how these operate (e.g. their speed).[33] These aspects of our model are currently being implemented, and details of these are not included here.

Our dual processing account of lexical architecture models lexical nodes as consisting not only of strings and lexical actions for computing representations, but also stored semantic structure. Figure (1) presents a schematic model of lexical nodes: each lexical node being bundles of phonological strings, lexical actions and semantic structures. These semantic structures are essentially the structures outputted from previous uses of the associated lexical actions. Accessing the information in these nodes is essentially via phonological strings (noted above), ranging from more compositional (like 'kick'), to more formulaic (like 'kick himself', 'kick herself', 'kick themselves'). By making both rules and structures available via some string set, our proposal aims to reflect the hybrid compositional/formulaic nature of the lexicon (e.g. Sprenger et al. (2006)).[34]

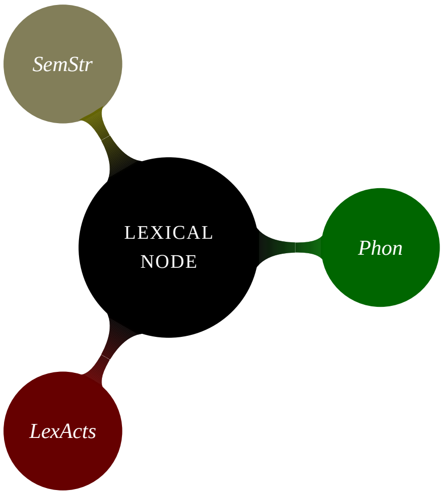


Figure 1: Representation of lexical nodes (*Phon* = phonological material, *LexActs* = lexical actions, *SemStr* = stored semantic structures)

In summary, we have shown with our series of examples employing tree abstraction, that this enables modelling the interaction between the processes that give rise to semantic structures as well as the process whereby these structures may be stored and subsequently retrieved. The result is a

[33] Such weights degrade over time, thereby modelling recency effects, so that both storing and retrieval of semantic structures, say, boosts their activation levels, and analogously, for the utility levels of productions when successfully firing.

[34] Indeed, a crucial element of Sprenger et al.'s account is the relationship between the lexical and conceptual levels, which is beyond the scope of our proposal here. We are not here proposing a full-fledged lexical architecture, and our highly schematic model presented here does not detail the links between specific phonological forms and specific actions and/or semantic representations, required to model how uttering 'kick himself' increases the likelihood of use of the semantic representation for this specific string. Indeed, in order to develop a more detailed account, we will aim for a model along the lines of the proposal in Sprenger et al. (2006), but bringing this in line with our more procedural approach.

unified story of formulaic and non-formulaic language,[35] extending the DS lexicon to consist of a network of nodes, each consisting of strings (as the locus of lexical entries), with their associated lexical actions (core linguistic knowledge) and semantic structures (accumulating via "linguistic" experiences).

4.2 Idioms in context

Finally, we demonstrate how our approach draws on aspects of the DS grammar model, to model idioms in dialogue. Consider the following examples of the idiom 'kick oneself':

(27)
A: Bob kicked himself
B: and so did Jill?

By way of demonstrating how dialogue phenomena are modelled in DS via core grammatical resources, Table (2) displays an analysis of example (27). First, on our extended model, given the idiomatic reading of A's utterance, there are two possible routes to constructing a representation of A's utterance: (i) use of the complex lexical actions for idiomatic reading of A's utterance (as set out in (26)), (ii) retrieval of long-term stored structure for the idiomatic reading (displayed in (25)). In accordance with our dual process account, each of these are viable alternatives which compete against one another to provide update. Second, B's response in (27) cannot mean that Jill kicked Bob, so that if immediate context is reused here, this cannot consist of the output structure, complete with its value for the metavariable, since this would wrongly allow the meaning Jill kicked Bob. However, depending on which was initially used, complex actions or else stored semantic structure, this would be available for reuse in this case.

This analysis reveals where we need to focus future work. For B's response to A, a DS analysis of "do" posits a metavariable of $Ty(e > t)$, enabling substitution of predicate information from context, with an obvious candidate being the stored semantic structure retrieved for parsing A's utterance (see (25)). Another candidate may in fact be the structure immediately following application of the complex lexical action triggered by the idiom (see (26)). For the analysis in Table (2), both alternatives, reuse of complex actions and retrieval of semantic structure, are theoretically possible given our dual process account (see Section (4.1)). Determining the strategy actually selected in this competition is an implementation issue; in future work we aim to implement the proposal in Section (4.1.5), modelling the competition between lexical actions and semantic structure in these terms.

[35] While we focus processing at the level of lexical nodes, others have suggested distributing this across multiple lexica (contra Becker (1975), Wray (2002), among others).

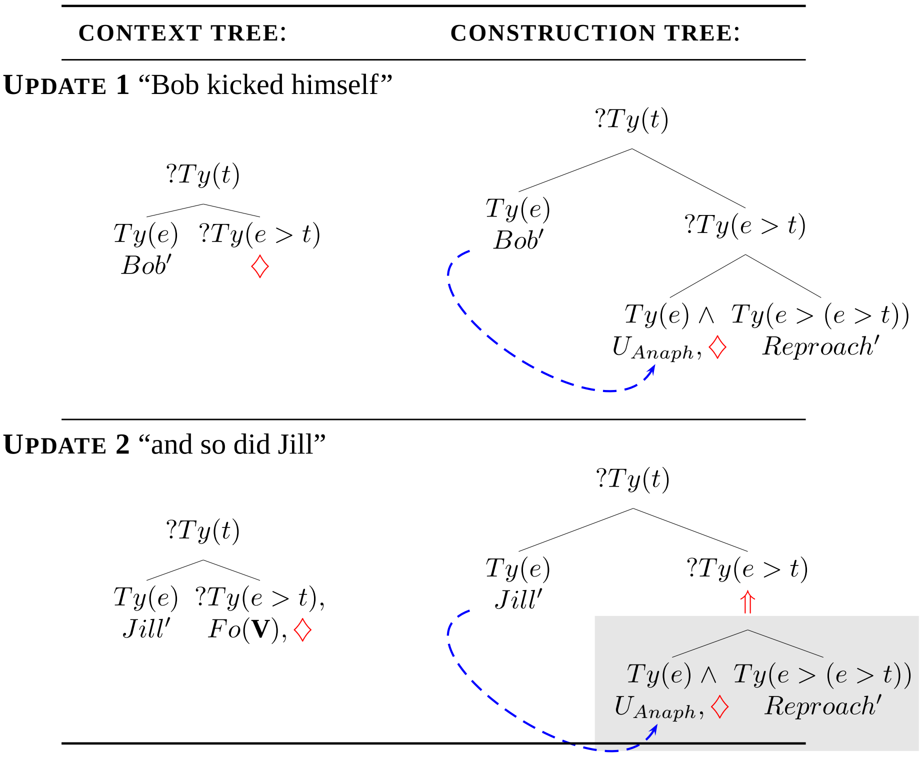


Table 2: B's Context and Construction trees for example (27). Updates to the context tree licensed by idiomatic reading of the string are symbolised by the blue dashed line (although these dashed lines are for expository purposes, and have no formal significance). Note that for Update 2, the separate steps of update from context and then resolving reference via a local form of Substitution to ensure identical formulae on subject and object nodes, are placed together on the same tree diagram for convenience, but they are in fact separate steps. The grayed section in Update 2 highlights the subtree drawn from context (in fact the $Ty(e > t)$ subtree from the construction tree in Update 1).

5. Conclusions: modelling context and routinisation in dialogue

We have provided a novel dual process account of modelling the dynamics of formulaic language, as an alternative to the more common property listing approach. By setting our account firmly within the DS framework, we retain features of this framework useful for modelling. Taking this approach, we can straightforwardly extend the framework to model *routine* dialogue phenomena. We extend the DS lexical architecture in order to model the emergence of formulaic language indirectly, within a model of language that focuses on replicating the processes underlying patterns of usage.

Our focus on the relationship between idioms and prefabs demonstrates the DS model of language as a system flexible enough to provide multiple strategies for a single form. An additional novel aspect of our approach to linguistic routinisation, is that we provide a unified account of processing, focusing this at the level of lexical nodes, rather than distributing this across multiple lexica as has been proposed elsewhere.

In sum, our contributions are fourfold: (1) a unified approach to formulaic and non-formulaic language, (2) a novel dual process account of formulaic language, (3) an extension of DS, in particular with respect to lexical architecture, and (4) a model of the dynamics of linguistic routinisation in dialogue. As an added bonus, our account turns out to be a linguistic implementation of the model for routinisation originally proposed by Logan from within cognitive psychology, directly demonstrating how the complexity of dialogue can be tackled by integrating insights across disciplines within cognitive science.

Acknowledgments. I would like to acknowledge Ruth Kempson, Eleni Gregoromichelaki and Matt Purver for insightful discussions which helped in clarifying the ideas presented here (Matt also kindly commented on a very early draft). Also many thanks to David Schlangen and Hannes Rieser for their expert editorial guidance, as well as three anonymous reviewers who generously provided acutely critical comments which vastly improved the final version (normal disclaimers apply).

References

K. Aijmer. *Conversational Routines in English: Convention and Creativity*. Longman, 1996.

J.R. Anderson. Automaticity and the ACT* theory. *American Journal of Psychology*, 105:165–180, 1992.

N. Asher and A. Lascarides. *Logics of Conversation*. Cambridge University Press, 2003.

J.A. Bargh. The ecology of automaticity: Toward establishing the conditions needed to produce automatic processing effects. *American Journal of Psychology*, 105:181–199, 1992.

D.J. Barr and B. Keysar. Perspective-taking and the coordination of meaning in language use. In M.J. Traxler and M.A. Gernsbacher, editors, *Handbook of Psycholinguistics: Second Edition*, pages 901–938. Amsterdam: Elsevier, 2006.

J.B. Bavelas. What's unique about dialogue? hand gestures, figurative language, facial displays, and direct quotation. In *Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue*, 2009.

J.D. Becker. The phrasal lexicon. In *Proceedings of the 1975 workshop on Theoretical issues in natural language processing*, 1975.

D. Bolinger. Meaning and memory. *Forum Linguisticum*, 1:1–14, 1976.

M. Bouzouita. At the syntax-pragmatics interface: clitics in the history of spanish. In R. Cooper and R. Kempson, editors, *Language Evolution and Change*. Cambridge University Press, 2008.

J.L. Bybee. From usage to grammar: The mind's response to repetition. *Language*, 82:711–733, 2006.

R. Cann, R. Kempson, and L. Marten. *The Dynamics of Language: An Introduction*. Elsevier, 2005.

R. Cann, R. Kempson, and M. Purver. Context and wellformedness: the dynamics of ellipsis. *Research on Language and Computation*, 5, 2007.

S. Chernova and R.C. Arkin. From deliberative to routine behaviors: a cognitively-inspired action selection mechanism for routine behavior capture. *Adaptive Behavior Journal*, 15(2):199–216, 2007.

M. Crocker. Computational psycholinguistics. In A. Clark, C. Fox, and S. Lappin, editors, *The Handbook of Computational Linguistics and Natural Language Processing*. Wiley-Blackwell, 2010.

M. Dalrymple, S.M. Shieber, and F.C.N. Pereira. Ellipsis and higher-order unification. *Linguistics and Philosophy*, 14(4):399–452, 1991.

B. Erman and B. Warren. The idiom principle and the open-choice principle. *Text*, 20:29–62, 2000.

A. Gargett. *Context and routinisation in dialogue*. Ph.d., Kings College London, 2010.

A. Gargett, E. Gregoromichelaki, R. Kempson, M. Purver, and Y. Sato. Grammar resources for modelling dialogue dynamically. *Cognitive Neurodynamics*, 3(4):347–63, 2009.

P. Healey. Expertise or expert-ese: The emergence of task-oriented sub-languages. In M.D. Shafto and P. Langley, editors, *Proceedings of the 19th Annual Conference of the Cognitive Science Society*, pages 301–306, Stanford, California, August 1997. Stanford University.

P. Healey. Interactive misalignment: The role of repair in the development of groxford university press sub-languages. In R. Cooper and R. Kempson, editors, *Language in Flux: Dialogue Coordination, Language Variation, Change and Evolution*. College Publications, 2008.

R. Jackendoff. *The Architecture of the Language Faculty*. MIT Press, 1997.

R. Jackendoff. *Foundations of Language: Brain, Meaning, Grammar, and Evolution*. Oxford University Press, 2003.

I. Kecskes. Dueling context: A dynamic model of meaning. *Journal of Pragmatics*, 2008.

R. Kempson, W. Meyer-Viol, and D. Gabbay. *Dynamic Syntax: The Flow of Language Understanding*. Blackwell, 2001.

K. Kuiper. *Smooth Talkers: The Linguistic Performance of Auctioneers and Sportscasters*. Lawrence Erlbaum Associates, 1996.

G.D. Logan. Toward an instance theory of automatization. *Psychological Review*, 95:492–527, 1988.

G.D. Logan. Automaticity and reading: Perspectives from the instance theory of automatization. *Reading and Writing Quarterly*, 13:123–146, 1997.

D. Marr. *Vision*. W.H. Freeman, 1982.

C. Matheson, M. Poesio, and D. Traum. Modeling grounding and discourse obligations using update rules. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Seattle, Washington, April 2000.

W. Meyer-Viol. Sequential construction of logical forms. *LNAI 2014*, pages 159–178, 2001.

D. Milward. Dynamic dependency grammar. *Linguistics and Philosophy*, 17:561–605, 1994.

G. Nunberg, I.A. Sag, and T. Wasow. Idioms. *Language*, 70:491–538, 1994.

M.J. Pickering and S. Garrod. Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, 27:169–225, 2004.

M.J. Pickering and S. Garrod. Establishing and using routines during dialogue: Implications for psychology and linguistics. In A. Cutler, editor, *Twenty-first Century Psycholinguistics: Four Cornerstones*, pages 85–101. Lawrence Erlbaum Associates, 2005.

M. Poesio and H. Rieser. Completions, coordination, and alignment in dialogue. *Dialogue & Discourse*, 1(1), 2010.

M. Purver, R. Cann, and R. Kempson. Grammars as parsers: Meeting the dialogue challenge. *Research on Language and Computation*, 4(2-3):289–326, 2006.

K.A. Rawson. Exploring automaticity in text processing: Syntactic ambiguity as a test case. *Cognitive Psychology*, 49:333–369, 2004.

K.A. Rawson and E.L. Middleton. Memory-based processing as a mechanism of automaticity in text comprehension. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 35 (2):353–370, 2009.

N. Ruh, R.P. Cooper, and D. Mareschal. A reinforcement model of sequential routine action. In *Proceedings of International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*, pages 65–70, 2005.

I.A. Sag, T. Baldwin, F. Bond, A.A. Copestake, and D. Flickinger. Multiword expressions: A pain in the neck for nlp. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, pages 1–15, 2002.

S. M. Shieber. A uniform architecture for parsing and generation. In *Proceedings of the 12th International Conference on Computational Linguistics*, pages 614–619, 1988.

S.A. Sprenger, W.J.M. Levelt, and G. Kempen. Lexical access during the production of idiomatic phrases. *Journal of Memory and Language*, 54(2):161–184, 2006.

P. Sturt and V. Lombardo. Processing coordinated structures: Incrementality and connectedness. *Cognitive Science*, 29(2):291–305, 2005.

N.A. Taatgen, D. Huss, D. Dickison, and J.R. Anderson. The acquisition of robust and flexible cognitive skills. *Journal of Experimental Psychology: General*, 137(3):548–565, 2008.

D. Traum. *A Computational Theory of Grounding in Natural Language Conversation*. PhD thesis, University of Rochester, 1994.

L. M. Ward. *Dynamical cognitive science*. MIT Press, 2002.

A. Wray. *Formulaic language and the lexicon*. Cambridge University Press, 2002.