CHAPTER 20

Dynamic Syntax

Foundations and Developments

RONNIE CANN, RUTH KEMPSON, ELENI GREGOROMICHELAKI, AND CHRISTINE HOWES

20.1 Introduction: Language as Process

Unlike most other current linguistic theories, Dynamic Syntax (DS) provides an incremental, process-driven model of language and linguistic behaviour. Gone are multiple levels or structures of representations for putatively distinct aspects of grammar such as syntax, semantics, and pragmatics. Gone, too, is a purely static conception of such representations as derived from whatever operations are defined as central to the model presented, such as Merge and Move in minimalist theories of transformational-generative grammar, unification in Head-driven Phrase Structure Grammar or combinatory operations or type raising in Categorial Grammar. Instead, a single level of representation, that of the conceptual content of an utterance, is derived time linearly as that utterance is produced. These representations may also be subject to further modification in the light of aspects of the context in which the utterance occurs. As such the framework presents a radical departure from the theoretical tradition of syntax which developed from Chomsky (1957) and the American structuralists who preceded. This shift in perspective~~,~~ has, in its turn, led to further radical questioning of apparently foundational concepts of linguistic theory over the relatively short period since the inception of the framework. The intent of this chapter is to discuss the basic philosophical ideas behind the framework as set out in Kempson et al. (2001) and Cann et al. (2005) and chart the various conceptual and philosophical shifts in perspective provided by later work.

20.2 The Framework: Structures and Actions

In its earliest conception, DS was intended as a formal model of utterance interpretation to underpin pragmatic theories such as Relevance Theory (Sperber & Wilson 1985/95). The initial goal was to provide a formal model of the process of logical form construction of utterances in context. This model directly paired strings with interpretations without prior construction of syntactic structure as traditionally defined over strings of words. Thus processes of parsing and generalized concepts of underspecification and update have formed the core of the framework from the beginning.

In work that pre-dates the formal specification of what became Dynamic Syntax in Kempson et al. (2001), Kempson set out two criteria that were the basic driving force behind the development of DS as a theoretical framework (Kempson 1995: 492):

Ronnie Cann et al., *Dynamic Syntax*. In: *The Oxford Handbook of Philosophy of Linguistics*. Edited by: Ryan M. Nefdt, Gabe Dupre, and Kate Hazel Stanton, Oxford University Press. © Oxford University Press (2026).
DOI: 10.1093/oso/9780198879640.003.0020

(i) Utterance interpretation is a time-linear inferential process in which, by the information encoded with each word in sequence, the hearer progressively builds a structure which they take to correspond to the interpretation intended by the speaker.

(ii) The input to interpretation provided by words underdetermines the content attributed to them in context, and as part of the interpretation process, the hearer has to make choices for all underdetermined aspects.

Following these criteria, the processing system is defined as a set of conditional actions constituting a goal-driven process of parsing an utterance incrementally as each word in the utterance is encountered. These goals are represented as requirements to achieve some end. In the original formulation, the requirement that starts off any grammatical processing is that of extracting some propositional formula representing the content of an utterance in context. To articulate this goal-driven, step-by-step process, a proof-theoretic perspective was adopted using the *Labelled Deductive System* (LDS) methodology (Kempson 1995; Gabbay 1996; Kempson et al. 1998). An LDS is a composite logical system in which proof transitions are defined for label-formula pairs in tandem: the labelling language and formula language are separately defined but in systematic correspondence with each other. Proof steps thus involve not just formulae/propositions as in standard natural deduction systems, but formula plus label (a *declarative unit*; DU). So *modus ponens* in an LDS appears as below, assuming the type (e.g. $P \to Q$) as formula (invoking a generalized form of the 'propositions as types' principle) and the label as a term representing some metalevel or semantic information about the derivation of the formula (e.g. in case of natural language content, α, β could stand for a 'proof object' or witness for the proposition or a time/world index or some control constraints):

(1) $\alpha : P,\ \beta : P \to Q \vdash \beta(\alpha) : Q$

This general LDS methodology is extended and expanded with various novel architectural features in the framework of Dynamic Syntax (Kempson et al. 2001). The data structures that a grammar manipulates are binary trees that can be partial and built or linearized incrementally as strings of words are consumed on a word-by-word basis. A logical language is defined for describing such trees, the DU language. The formulation of the grammar in this language can be conceived in the sense of a set of constraints whose cumulative effect is to allow for models which are sets of trees in various states of completion. In this sense, DS is a 'model-theoretic' (as opposed to 'generative-enumerative') grammar formalism (Pullum & Scholz 2001). This is important because defining a set of constraints on models, as opposed to building the data structures directly, allows for unlimited, but constrained, context dependency and underspecification, which addresses to a large extent the so-called 'frame problem' in AI and epistemology (see, e.g., Bickhard 2009).

The DU language describes the information residing on tree nodes as well as predicting and constraining future developments of the trees. These predictions for upcoming information are defined as *requirements* that determine potential developments and exclude others. DS binary trees are described by employing a modal logic (LOFT, the Logic of Finite Trees; Blackburn & Meyer-Viol 1994). This description language describes tree structure and provides modalities for describing any node of a binary tree from the perspective of any other node, so defining a set of tree node relations. The current node under development is indicated by a *pointer*, $\Diamond$, which has a unique position on a tree at each processing stage, thus allowing the modelling of word order effects. The nodes of the tree are ordered left to right and are addressable by a special label *Tn* consisting of sequences 0s and 1s to define the node position, e.g. $Tn(0)$, the root node, $Tn(00)$, the left daughter, $Tn(01)$, the right daughter, etc. Moreover, the tree address might be temporarily underspecified, e.g. $Tn(a)$ can be used to talk about a node that does not yet have a fully specific position on the tree. But the LOFT modalities allow much more expressiveness than this global view of tree structure. Through LOFT we can take a local point of view and we can talk about daughters of the *current* node with the modality $\langle\downarrow\rangle$, with variants $\langle\downarrow_0\rangle$ and $\langle\downarrow_1\rangle$ for the left and right daughters respectively, or the mother of the current node ($\langle\uparrow\rangle$). On this basis, the general concepts of 'dominate' and 'be dominated' are defined in terms of the Kleene star, indicating 0 or more mother-daughter relations as $\langle\uparrow *\rangle Tn(a)$, 'somewhere above this node is Tn(a)', and $\langle\downarrow *\rangle Tn(a)$, 'somewhere below this node is Tn(a)'.

The nodes of these binary trees are inhabited by DUs plus requirements. A DS DU follows the LDS idea of a label-formula pair but in a more elaborate manner in that the labels assume a substantial burden for executing control of the incremental natural language processing. For this reason, the LDS label-formula articulation shown in (1) is reversed with the type (e.g. $P \rightarrow Q$) now being a label value that controls the composition of conceptual content, the latter appearing as a formula in the language of the lambda calculus. The inhabitants of tree nodes are now feature structures that can be recursive, with the features being partial functions indicating the various labels and the formula inhabiting the node. Features are functions like *Fo*, *Ty*, *Tn*, standing for 'formula', 'type', and 'tree node' respectively, as well as any potentially useful linguistic features that might be relevant to the processing task.[1] The values of these features can also be partial and underspecified like any structure in DS.

In general, values are provided by the linear arrangement of words in a string, with words taken to be procedures that induce the creation and expansion of tree structures and node annotations. This extension of Relevance Theoretic assumptions means that lexical specifications are thus taken not to be direct content bearers, but procedural constraints on conceptual structure. Such constraints expressed in a grammar that includes the lexicon of a language are defined as conditional context-relative actions that, as a result, deliver the equivalent of ad-hoc concepts (Carston 2002) as content. Different categories of words, such as *Jeremy* and *cries* will provide conceptual content values for *Fo* in the form of the function-argument articulation of the lambda calculus. The lambda term that eventually becomes the *Fo* value of the root node is built bottom-up from the terms residing in the leaf and intermediate nodes of the tree by means of function application from the daughter nodes to the mother node. Noun phrases (NPs or DPs) thus generally provide content of the argument type (conventionally shown as, e.g., $Fo(Jeremy')$ for the word *Jeremy*) whereas verbs provide functions (e.g. $Fo(\lambda x.Cry'(x))$ for *cry*). Function application is controlled by the type information (values of the *Ty* label), which is expressed, as in type theory, in the form of propositional formulae in a conditional logic: there is the type e ($Ty(e)$) for individuals, and the type t ($Ty(t)$) for propositions, while the semantic type $<e, t>$ corresponds to the formula $e \rightarrow t$, $<e, <e, t>>$ to $(e \rightarrow (e \rightarrow t))$, etc. The grammatical processing tasks are controlled by type predictions (requirements in the form of, e.g., $?Ty(e)$, $?Ty(t)$) seeking to instantiate these types from processing of words as they are encountered or from actions on the node annotations. In Kempson et al. (2001), the initial requirement that starts a processing task, dubbed the Axiom, is shown as $?Ty(t)$, the goal of constructing a formula of propositional type, $Ty(t)$.[2]

The procedural structure of the framework relies on another modal logic, a customized version of Propositional Dynamic Logic (PDL).[3] This model of algorithmic process is applied

[1] Although, at least in principle, specific linguistic features are avoided as much as possible in favour of using context and process to account for phenomena typically accounted for by the postulation of such primitives. See Cann (2011) for an account of the English auxiliary system that uses no syntactic features.

[2] In later work, it is recognized that not all utterances lead, or need to lead, to some propositional outcome, see, e.g., Gregoromichelaki & Kempson (2019).

[3] PDL is a logic which was originally designed for modelling the behaviour of computer programs. 'Actions' or 'programs' are used interchangeably in this context.

to the development of binary trees 'decorated' with feature structures indicating a content formula and control labels. The DS model defines a space of stages of partial trees in various states of completion both structurally and in terms of node decorations (i.e. feature structures). Progressive transitions within this space are provided by PDL modalities licensing traversal of this space driven by the constraints contributed by the particular natural language under consideration. The PDL basic modalities thus stand for *actions* enabling tree development both structurally and in terms of node decorations. Some of the basic actions are $\texttt{make}(X)$ for creating new nodes, $\texttt{go}(X)$ for moving the pointer, and $\texttt{put}(Y)$ for annotating nodes, where X and Y are tree addresses and node decorations respectively. These basic actions can be combined by regular operations like sequential composition or iteration and can also be expressed in the form of conditional actions: $\langle IF\ \ \alpha,\ THEN\ \ \beta,\ ELSE\ \ \gamma\rangle$.

Moreover, basic actions and their regular combinations can be collected as packages of (routinized) sequences of actions (*macros*) given some triggering environment. Such macros are presented as *transition rules*, developing one tree description into another or conditional actions associated with parsing words. *Computational Rules* concern the development of a tree without consuming any lexical input. So a rule of INTRODUCTION that divides one goal into two subgoals may be defined as:[4]

$$
\text{(2)}\quad
\begin{array}{ll}
\text{IF} & ?Ty(\alpha) \\
\text{THEN} & \texttt{make}(\langle\downarrow_1\rangle), \texttt{go}(\langle\downarrow_1\rangle), \texttt{put}(?Ty(\beta \rightarrow \alpha)), \texttt{go}(\langle\uparrow_1\rangle), \\
& \texttt{make}(\langle\downarrow_0\rangle), \texttt{go}(\langle\downarrow_0\rangle), \texttt{put}(?Ty(\beta)) \\
\text{ELSE} & \texttt{Abort}
\end{array}
$$

This can be used to induce a partial tree structure that is rooted in $?Ty(t)$ with no other structure into one with two immediately dominated nodes bearing the requirements $?Ty(e)$ and $?Ty(e \rightarrow t)$, respectively.

While computational actions provide top-down procedures for structure building, parsing words induce bottom-up actions that may simply annotate a node with formula and type information, as with the actions associated with the proper name *Fiona* in (3a) or may induce more structure and annotate more than one node with requirements and labels as in the actions associated with a transitive verb such as *upset* in (3b).[5]

$$
\text{(3)}\quad \text{a. } \textit{Fiona}\ \left|
\begin{array}{ll}
\text{IF} & ?Ty(e) \\
\text{THEN} & \texttt{put}(Ty(e), Fo(\textit{Fiona}')) \\
\text{ELSE} & \texttt{Abort}
\end{array}\right.
$$

$$
\text{b. } \textit{upset}\ \left|
\begin{array}{ll}
\text{IF} & ?Ty(t) \\
\text{THEN} & \texttt{make}(\langle\downarrow_0\rangle), \texttt{go}(\langle\downarrow_0\rangle), \texttt{put}(Ty(e_s), Fo(\mathbf{s}_{\text{past}})), \texttt{go}(\langle\uparrow_0\rangle), \\
& \texttt{make}(\langle\downarrow_1\rangle), \texttt{go}(\langle\downarrow_1\rangle), \texttt{put}(?Ty(e_s \rightarrow t)), \\
& \texttt{make}(\langle\downarrow_0\rangle), \texttt{go}(\langle\downarrow_0\rangle), \texttt{put}(?Ty(e)), \texttt{go}(\langle\uparrow_0\rangle) \\
& \texttt{make}(\langle\downarrow_1\rangle), \texttt{go}(\langle\downarrow_1\rangle), \texttt{put}(?Ty(e \rightarrow (e_s \rightarrow t))), \\
& \texttt{make}(\langle\downarrow_1\rangle), \texttt{go}(\langle\downarrow_1\rangle), \\
& \texttt{put}(Ty(e \rightarrow (e \rightarrow (e_s \rightarrow t))), Fo(\textit{Upset}')), \texttt{go}(\langle\uparrow_1\rangle), \\
& \texttt{make}(\langle\downarrow_0\rangle), \texttt{go}(\langle\downarrow_0\rangle), \texttt{put}(?Ty(e)), \\
\text{ELSE} & \texttt{Abort}
\end{array}\right.
$$

[4] This is a simplification of the rules in Kempson et al. (2001).

[5] Note that s_{past} represents a past tense situation variable standing in lieu of a structured situation term; e_s is the type of such a term in a sorted type domain of individuals (see Gregoromichelaki 2006 and Cann 2011). Note further that exact details of how subjects are introduced and identified as the highest individual argument are suppressed (Cann 2011).

Notice that the actions in (3b) construct the predicate-argument structure of the verb *upset* so that a parse of *Fiona upset* yields the structure in Figure 20.1 where the pointer is left on the logical object node in anticipation of further lexical input.

In terms of combining information, DS uses primarily *modus ponens* on the side of the *Ty* label and function application on the formula, *Fo*, side. This reflects the Curry-Howard isomorphism of type deduction and function application (Howard 1980). Conventionally, the left daughter of a node always accommodates the argument term (the 'argument daughter') while the right daughter is inhabited by the function ('the functor daughter'). Despite using a familiar typed lambda calculus (Montague 1973) as the language of conceptual content, DS deliberately only uses a very small and restricted subset of types and eschews general rules of type inference. The idea is to keep typing as close as possible to the sort of thing that words intuitively describe. In particular, all forms of type raising. are avoided and that means that noun phrases (NPs or DPs)[6] are taken to contribute entities to the conceptual structure, not generalized quantifiers (Montague 1973; Barwise & Cooper 1981; Keenan & Stavi 1986; etc.). Amongst other things this allows predicates of whatever arity to express properties and relations directly over individuals without the need for complex systems of meaning postulates or logical decomposition to reduce higher types to lower ones.[7]

To achieve this while, at the same time, accounting for the effects of quantification, Kempson et al. employ the *epsilon calculus* of Hilbert & Bernays (1939) as developed in Meyer-Viol (1995) (cf. Avigad & Zach 2020). Although developed specifically for providing a philosophical foundation for mathematics, the calculus has been adopted into formal theories of language, specifically to account for certain puzzles in anaphora construal (Chierchia 1992; von Heusinger 1994, 1997, Egli & von Heusinger 1995, Chierchia 1992; i.a.). In essence, the epsilon calculus is the logic of the use of arbitrary names in predicate logic proofs. It utilizes

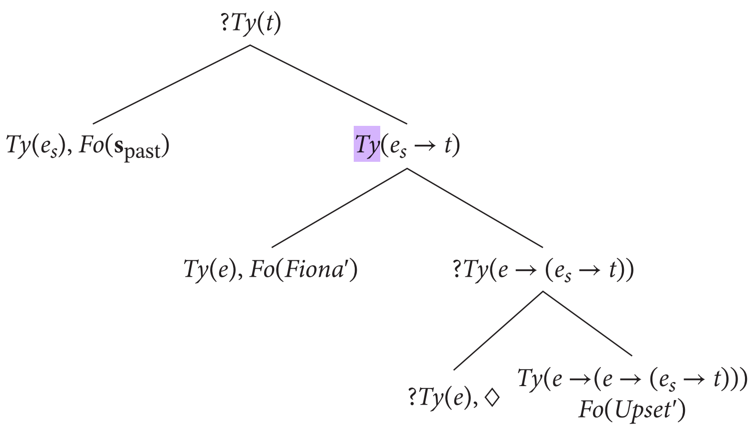

FIG. 20.1 Processing *Fiona upset*

[6] Note that these labels of purely syntactic categories are not expressed in DS since a separate level of syntactic representation is eschewed.

[7] While there is little formal difference between the use of high types with meaning postulate and the use of low types and the epsilon calculus (see below), DS avowedly aims to model human cognition in relation to language and linguistic behaviour. In that sense DS can be regarded as a formaal version of 'cognitive grammar' albeit very different from theories that are typically labelled as such (see, for example, Langacker 2008).

an operator ε and its dual τ which bind a free variable and map a proposition(al function) P into a term of the form $\varepsilon x.P$ or $\tau x.P$ of type e. Semantically, an *epsilon term* identifies a witness that satisfies its predicate scope, if such a witness exists, or, if not, any arbitrary entity. Syntactically, this allows the quantifier-free epsilon calculus to be directly related to first-order predicate logic with quantifiers by the following definitions:

$$\exists x.A(x) \equiv A(\varepsilon x.A(x))$$
$$\forall x.A(x) \equiv A(\varepsilon x.\neg A(x)) \equiv A(\tau x.A(x))$$

Accordingly, in DS, the content of the noun phrase *a dog* is shown as $(\varepsilon x.Dog'(x))$ while the content of *every dog* is represented as $(\tau x.Dog'(x))$.

The decision to adopt the epsilon calculus offers the advantage that natural language quantifiers can be captured as names, hence of low entity type, with whatever complexity of scope variability, etc., being captured through the semantics of these epsilon (and tau) terms assigned to the relevant natural language phrases. In this way, the framework is able to sustain the match between argument terms of predicates in the conceptual structure and the configurational behaviour of quantified expressions in natural language which standardly occupy clause-internal argument positions. This decision in its turn, like the adoption of the LDS methodology, enabled DS to achieve modelling the dynamics of the step-by-step construction process, a combination which has proved essential to capturing the 'syntactic' generalizations that DS is able to express.

The DS commitment to maintaining a low type system might seem to pose a problem in accounting for certain constructions that are typically assumed to require the postulation of more internally complex types. One such is modification of various sorts which has often been analysed in terms of types of the form $A \rightarrow A$ $(<A, A>)$ where A is the type of the expression being modified. This basic approach can be found with respect to adverbials and adjectives from the early days of Montague Semantics (Montague 1970). For example, Thomason & Stalnaker (1973) define an operator ξ which maps one predicate into another, while Keenan & Faltz (1984) provide a completely general and variable typing system of $A \rightarrow A$ functors in their Boolean approach to semantics. However, instead of utilizing such higher types, DS makes recourse to an additional tree relation beyond simple dominance and node ordering. This is the Link relation, which connects one partial tree to another such that both trees share a common term. Conceptually, these Link structures correspond to subproofs in an unfolding main proof and, in natural language, they allow the foregrounded processing to be interrupted in order to enrich the context with further relevant information, before continuing the processing of the primary structure. Accordingly, Linked trees were used initially to provide an account of relative clause modification (Kempson et al. 2001), but this relation has a wide range of application to a number of different phenomena, such as hanging topics, afterthoughts, Right Node Raising, agreement in Swahili, and much more (Cann et al. 2005, i.a.).

Processes involving Linked structures target a node of a certain type, often but not exclusively e or t, with a formula value α and create a new Linked structure rooted in a node of the same or different type with a requirement for a copy of the formula α somewhere within the new structure. As a very simple example, consider a possible analysis of *Fido ran quickly*. With the introduction of a situation node as the first argument of a tensed clause (Gregoromichelaki 2006), a typical Neo-Davidsonian event analysis (Parsons 1990) can be given by projecting a propositional Link structure from the situation node, with a copy of the content of that node as the first argument of a one-place predicate requirement, allowing a parse of the adverb. This is illustrated up to the point of parsing the adverb in Figure 20.2 in simplified form.

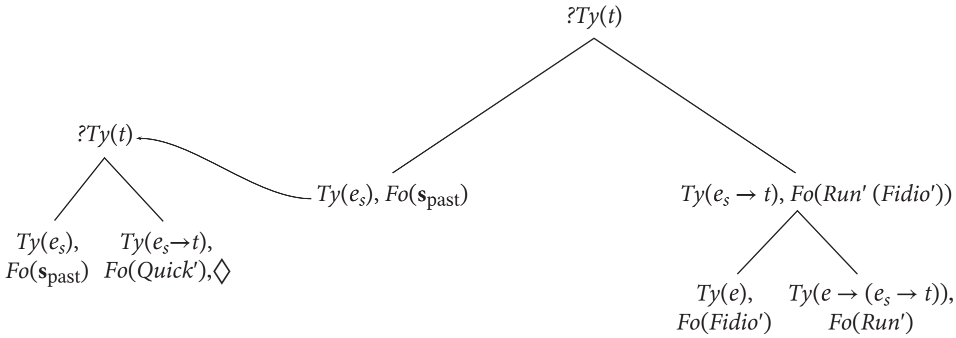

FIG. 20.2 Processing *Fido ran quickly*

A rule of LINK EVALUATION combines the completed formulae on the two propositional nodes to yield a final output formula: $Quick'(\mathbf{s}_{past}) \wedge (Run'(Fido'))(\mathbf{s}_{past})$.[8]

So far, we have noted the requirement $?Ty(\alpha)$, indicating a goal to specify some content of the appropriate type on the current node. However, requirements may take the form of any possible annotation on a node. In this way, they are anticipatory of possible developments of the current node, at least one of which must lead to a successful satisfaction of the requirement. Moreover, requirements employing the LOFT modalities can be used to add constraints to the way a node can successfully develop while the pointer resides at some other node or from the perspective of another node, even for nodes or features that have not yet appeared on the tree. For example, a modal requirement $?\langle\uparrow_0\rangle Ty(e_s \rightarrow t)$ can be used to constrain where in a structure the content of a nominative pronoun in English may appear, i.e. as the first $Ty(e)$ (non-event) individual argument of some predicate-argument structure. These requirements thus drive the subsequent tree-construction process in the sense that the search for further processing routes will be goal directed to satisfy the predicted values that requirements have introduced. Quite generally, since 'syntax' is seen in procedural terms and not as a separate level of representation, it is actions that are basic, not representations, with structural patterns normally identified as 'syntactic' being expressed through the action vocabulary. For example, to identify the requisite concepts of locality constraining antecedenthood for both reflexives and pronouns, all that is needed is a characterization of the constraints on the steps the pointer is licensed to make in seeking a potential antecedent. This characterization derives as an epiphenomenon a notion of 'co-argument': any formula value that can be found by moving up one argument-relation step plus a (possibly empty) sequence of functor-relation steps and then one argument step below. All concepts of syntactic command may be restated in these terms (see Gregoromichelaki, 2006, chs. 7–8, 2013).

In addition to the underspecification of node decorations indicated by type requirements and associated structural constraints, there is also underspecification in terms of content. This is particularly (although not exclusively, see, e.g., Cann 2011) useful in characterizing all forms of pronominal resolution, including WH pronouns. A parse of a form like *it* in the context of $?Ty(e)$ satisfies the type requirement, but fails to provide a suitable content specification. So the formula value provided by the pronoun is underspecified. In DS, this is represented as

[8] Note here the lack of formal differentiation between adverbs and adjective and indeed prepositions which are all of type $e \rightarrow t$ where the type e expression may be a situation or individual term depending on the particular lexical item. Differences in behaviour that, in other frameworks, are defined in terms of separate category labels are, in DS, handled by differing triggering environments and action sequences.

a *metavariable*, conventionally represented as **U**, **V**, etc., an expression that is not part of the content language but a 'placeholder' in the DU language that needs to be replaced by such content. This is ensured by the requirement $?\exists\mathbf{x}.Fo(\mathbf{x})$ that can only be satisfied through the substitution of the metavariable by the recovery of content from the context which may be immediately available as in (4a) or only later in a parse as in (4b):

(4) a. I was horrified by the state of the oven. It was filthy.
b. When I saw it, I realised the oven was too filthy to use.

What is perhaps more striking for a putative grammar system is the modelling of long-distance dependency, the bastion of transformational grammars, standardly used as an argument for the 'autonomy of syntax' hypothesis. In DS, these dependencies are reformulated in action-based terms of underspecification and update. Classic left-peripheral dislocation is analysed in terms of a rule called *Adjunction that constructs a structural relation for a node whose position within an unfolding structure is as yet unspecified. As noted above, such an *unfixed node* is introduced along with a generalized dominance dependency $\langle\uparrow_*\rangle\alpha$ and a requirement $?\exists\mathbf{x}.Tn(\mathbf{x})$ to fix the position within a tree that must be dominated by α. This perspective on long-distance dependency is again no more than the concept of underspecification plus update taken from Relevance Theory, generalized to capture not merely the emergence of content, but also the emergence of tree structure via the same dynamic. In both cases this is driven by the anticipatory twist given by the concept of requirement for update.

20.3 The DS Framework as a Grammatical Formalism

Although, as noted above, developed as a formal model of recovering content from utterances in context, the DS framework clearly has obvious applicability to solving a wide range of syntactic puzzles, from Right Node Raising to clitic clusters in a number of languages, from multiple scrambling in Korean and Japanese to agreement in Bantu and much more besides. In this section we briefly describe applications of the framework to a number of syntactic constructions, illustrative of how a process-oriented syntax driven by underspecification and update can be used to analyse grammatical phenomena.

The action-based account of long-distance dependency and other complex constructions provides natural explanations for the different variant types of relative clause structures, question types, cleft structures, and left-right-peripheral asymmetries. These had previously each been seen as having wholly distinct subtypes apparently necessitating the positing of multiple ambiguities right across the board.[9] Yet all of these somewhat disparate construction types can be given an integrated explanation as straightforward variants from a processing standpoint (for discussion, see Kempson et al. 2001; Cann et al. 2004, 2005; Gregoromichelaki et al. 2011; Purver et al. 2021).

One particularly striking result has been to find that underpinning all such structure-building processes is a universal constraint that derives from the underlying and interacting PDL and LOFT formalisms, but may be further reducible to fundamental properties of

[9] This methodological assumption is extremely widespread and almost never questioned in any framework, so that phenomena such as pronouns, relative clauses, questions, negations, determiners, etc., are all taken to involve multiple discrete specifications.

processing and context. This is the constraint that there can be no construction of two distinct arguments resulting from a single transition as it is the transitions that define the resulting node, so any multiplicity of nodes with the same relation to the root node is simply excluded. This restriction guarantees that no more than one unfixed node of a type can be constructed from any node at any interim point in the construction process. The striking effect of this is to constrain dislocation to a single instance at a time, without having to posit the existence of some permanent specialized node in a tree such as [Spec,CP] to which an expression is moved. Such a constraint might seem to be falsified by the many verb-final languages in which the rigidly final verb may be preceded by a number of nominal expressions within a clause in any order which would seem to involve whole sequences of unfixed nominal nodes as their fixed positions are not determined until the verb is parsed, as illustrated in the Japanese example in (5):

(5)	*syorui-o*	*supai-ni*	*zyaaranisuto-ga*	*watasita*
	document-OBJ	spy-DAT	journalist-SUBJ	gave

'The journalist gave the document to the spy'

Yet languages that allow such very free constituent ordering within the clause without any support from anaphoric devices (Japanese, Korean, Greek, German in the Mittelfeld, etc.) may be treated as using case marking to induce the requisite structure (Kempson & Kiaer 2009, 2010, following insights in Lexical Functional Grammar in Nordlinger 1997). Modelling this constructive view of case in DS terms, we can assume that a noun phrase such as *syorui-o* introduces an unfixed term node with the case marker predicting and ensuring that the unfixed node becomes the internal argument node through a requirement $?Ty(e \rightarrow (e_s \rightarrow t))$. This radical curtailing of further processing options leads to the immediate induction of the relevant structure fixing that hitherto underspecified tree relation. The effect of this immediate structural enrichment is to offer up the possibility of a second application of building an unfixed node without violating the overall tree-structure restriction. However, this phenomenon is essentially local, so in the account of Kempson & Kiaer (2010), this is expressed by defining a local variant of *ADJUNCTION, LOCAL*ADJUNCTION. This licenses the introduction of a locally unfixed node, with an address defined along a tree path $\langle \downarrow_* ^1 \downarrow_0 \rangle$ which ranges over 0 or more functor nodes plus one argument node. Presuming such an action macrolicenses an arbitrary sequence of nominal expressions to induce a progressively enriched structure, with the pointer in each such update returning to the dominating $?Ty(t)$ node (this being the trigger for the several steps of LOCAL*ADJUNCTION). Following this, the verb can be parsed to complete the structure. The partial tree configuration already constructed as triggered by any such sequence of case-marked expressions will simply collapse with the predicate-argument structure subsequently induced by the verb. This is because the local partial structure which case marking and structural inference induce is identical to that built by the actions triggered by the verb.[10]

There is striking confirmation of this style of analysis from a process called multiple long-distance movement which occurs in these languages, which is subject to an inviolable restriction that such multiple instances be local to each other within that configuration from which they are dislocated:

[10] Given that trees are formally defined in terms of tree node relations, it is necessarily the case that two identical sets of such relations define the same tree.

(6)

yorui-o	*supai-ni*	*kaizatsu-wa*	*zyaaranisuto-ga*	*watasita*	*to*	*itta*
[document-OBJ	spy-DAT]	police-TOP	journalist-SUBJ	gave	COMP	said

'the police said that the journalist gave the document to the spy'.

These present a major challenge to movement analyses, as they appear to motivate multiple applications of long-distance movement with no means of stating any locality restriction on the rule itself. However, as generally licensed in DS given the model-theoretic nature of the formalism, these facts follow from the interaction between the two types of underspecified tree relation. *ADJUNCTION, as noted, can only be triggered once and only from a root node, so any use of this action may feed the local variant but not vice versa. Hence, a parse may begin with the construction of an unfixed propositional node. At this point, LOCAL*ADJUNCTION can apply, allowing the parse of any dislocated nominals whose case markers license structure building as described above. Note, however, that all such dislocated nominals will inevitably be within a single local propositional domain. The remainder of the input can then be parsed with the unfixed propositional node merging with the internal argument of *itta* 'said' shown as the dashed arrow illustrated in Figure 20.3.[11]

One of the other DS mechanisms that has surprising uses beyond the initial use to model a substitution view of pronoun resolution is the use of metavariables licensing subsequent update. An early extension is found in Cann et al. (2005) and Kempson et al. (2005) ~~in~~ an account of so-called Right Node Raising (Ross 1967). Here, an example like (7) is analysed as having a metavariable decorate the missing internal argument in the first conjunct which is copied into the second conjunct on an unfixed node. The second conjunct is parsed as normal with the unfixed metavariable unifying with the internal argument of *read*. A process of LATE *ADJUNCTION from this node allows the parse of the overt noun phrase which substitutes for the metavariale and instantiates the metavariable in the first conjunct.

(7) I do own but haven't read any books by James Joyce

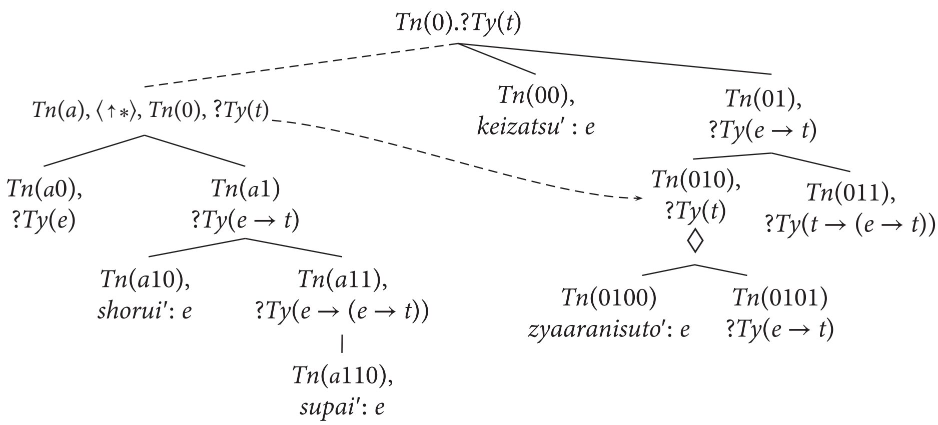

FIG. 20.3 Parsing *yorui-o supai-ni keisatsu-wa zyaaranisuto-ga*

[11] All effects of tense construal have been omitted from Figure 20.3 for clarity of display.

The puzzle posed by examples like (7) is that, while the negative polarity item *any* is licensed in the second conjunct, it is not so licensed in the first, posing a problem for any copy theory of the resolution of the gap in the first conjunct, given the ungrammaticality of **I do own any books by James Joyce*. However, in this analysis the problem simply does not arise. This is so because the constraint is imposed on the *process* of building up the structure of the second conjunct, not the resulting structure itself. So the point at which *any* is parsed, the current node is in the scope of the negation, licensing the NPI.

An interesting consequence emerges from this analysis which has posed serious problems for alternative accounts: a negative polarity item can appear if the licensing context is provided by the second conjunct, but not if only the first conjunct contains the appropriate licensor:

(8) *I haven't managed to read but do own any books by James Joyce.

Unlike more conservative accounts in which all such constraints are a constraint on the output structure, in DS the constraint is a constraint on the process of constructing the structure of the second conjunct in which no negator appears, leaving *any* with no licensor.

To end this section, we briefly review a further extension of the use of metavariables: the use of predicate metavariables to account for VP ellipsis where again the form of the linguistic input severely underdetermines the construal assigned to what is said. The challenge here is how to reflect the pairing between such input forms and the different types of content these give rise to, either as richer input structures matching these contents to be deleted in their surface realization or in more procedural terms. The precise array of contents has proven problematic for the conventional static assumptions of pairings of strings and their assigned content, there being two identifiably distinct interpretations assignable from an unambiguous antecedent, labelled as strict vs sloppy readings (see (9), below). If the account of them is taken to require identity of structure with their antecedent, it will require different input structures not only for the elliptical form but also for its antecedent (Hankamer & Sag 1976). For semantic accounts (presuming some selected abstraction from the antecedent's content), the problem is how to appropriately limit the abstractions to be applied to the antecedent (Dalrymple et al. 1991). In DS terms however, the problem vanishes: this is no more than the anticipated availability within the context of not only the content of the antecedent as projected in context, but also the actions leading to such content. This corresponds directly with the strict vs sloppy construals of ellipsis in terms of the potential for reconstruction from context of both content and procedures (Kempson et al. 2001, 2015, 2016).

In this vein, to capture (9) with its strict and sloppy readings as indicated, auxiliaries in English are taken to project not explicit content but predicate metavariables.[12]

(9) a. Bill loves his mother and so does John.
b. Bill loves Bill's mother and John loves Bill's mother. STRICT
c. Bill loves Bill's mother and John loves John's mother. SLOPPY

Hence, auxiliary *do* projects a predicate metavariable decorating the predicate node in the second conjunct. As with term metavariables, this is associated with a formula requirement $?\exists\mathbf{x}.Fo(\mathbf{x})$ which can be provided directly by substituting the predicate formula in the first conjunct $Fo(Love'(\epsilon(x.Mother\text{-}of'(x, Bill'))))$ to yield the strict reading. To ensure the availability of the sloppy reading, an alternative to direct substitution is employed that does

[12] Any explicit modal or other meaning is taken to involve manipulation of the content of the situation term (Cann 2011).

not copy content but re-runs a sequence of actions used to construct some partial tree in the context (Kempson et al. 2001, 2015, 2019). This type of substitution then re-runs the actions used to construct the predicate of the first conjunct in (9), targeting $Fo(John')$ as the relevant element in the new emergent context. Crucially this relies on the action of assigning the value of the anaphoric 'his' applied to the antecedent string, rather than its resulting content, to yield the sloppy reading $Fo(Love'(\epsilon(x, Mother\text{-}of'(x, John'))))$ as the interpretation of the elliptical VP.

Stepping back from these details, it is notable that the analyses set out here turn on the transitions involved in the dynamics of emergent structure building following a processing timeline, departing from the standard conservative methodology of syntactic theorizing which allows no reference to such a time-linear process in accounting for structural constraints. Moreover, being able to provide an integrated explanation of at first glance wholly unrelated morphosyntactic and syntactic constraints is easily expressible within a dynamic and process-oriented perspective and much less straightforward in static systems with strict separation of syntax, semantics, and pragmatics. This strongly suggests that linguists should take heed of the need to express the dynamics of such emergent structure within the grammar formalism, radically undermining the assumptions underpinning the competence-performance divide.

20.4 Dialogue, Generation, and Parsing

While the tenets of DS and underlying assumptions depart quite radically from more conventional theories, some of the early DS assumptions were quite conservative, as the initial aim was to test the framework against the methodology adopted by other frameworks. Perhaps the most striking of these is the sentential basis of analysis, reflected in the Axiom determining that the initial goal is to construct some propositional formula, even despite the acknowledgement that context may have a direct effect on the content, and indeed syntax, of an utterance. So at this early stage of DS development, the assumption of needing to characterize isolated complete sentences remained a working assumption. Yet there are manifold data from conversational dialogue which cast serious doubt on the soundness of the centrality of the sentence. It can be shown from corpus data that participants in a conversational exchange can take up seamlessly from each other's partial or initiating offering, apparently splitting over all types of morphosyntactic, syntactic, and semantic dependencies (Purver 2010). This phenomenon is little short of a mystery for all frameworks delimited by a methodology which considers only data associated with an individual speaker. Illustrative is (10) in which an initiating utterance is extended successively by a number of participants, even shifting across one type of speech act to another as the structure unfolds, with no one speaker having the complete emergent structure in mind *ab initio*:

(10) Alex (mum): We're going to
Hugh (dad): to Burbage, to see Ann, Auntie Ann
Eliot (son): with the dogs?
Hugh (dad): if you take care of them.

Here we have a preposition picked up and repeated by Hugh with an adjunct purpose clause. This interchange is taken over by Eliot by adding a fragment to be taken both as a continuation of the previous utterance but, in addition, as a clarificatory query. This instance of shared utterances illustrates the multifunctionality of such segments in speech act terms (Gregoromichelaki 2012; Gregoromichelaki & Kempson 2016a). The clarification is answered within that overall emergent structure by the addition of a conditional protasis, which needs to be taken as taking scope over the entire previous structure. Notice in particular here that stringing all the segments together creates a string whose infelicity only resides in the early repeated preposition, which is a common phenomenon in single speaker utterances. In addition, the interpretation of indexicals needs to be adjusted (e.g. the *you* in the conditional protasis, which indicates the need for radical incremental interpretation as the words unfold (see, e.g., Gregoromichelaki 2018).

The efficiency and coordinative function of shared utterances cannot rely on a grammar that glues together phrases to derive a sentence/proposition as the fundamental unit of analysis. Indeed, in (11) where the resulting string is *Did you burn myself?*, it is not this which provides the basis for determining that the exchange is perfectly acceptable, as any grammar would uncontentiously debar the string as not well formed:

(11) A (leaving a smoke-filled kitchen): I've had an accident. The kitchen is badly burned.
B: Did you burn
A: Myself? Fortunately not.

In (12), furthermore, there is not only a shift in pronoun where the surface reconstruction would again yield the wrong interpretation, but there is a speech act shift from the lawyer's question to the client's assertion. Note also the break between possessive determiner and head noun:

(12) Lawyer: Will you have your son as the executor of your will or your
Client: wife.

This sharing of roles in the construction process is wholly general, potentially able to distribute any syntactic or semantic dependency across participants, as witness the split between the left-peripheral expression and the gap where it needs to be reconstructed in (13), with its follow-on notably having to respect whatever local morphological dependencies have been set up by the previous speaker:

(13) A: Which department did you think we will have to
B: axe/*axed/*axes/*axing.

There are also exchanges composed entirely of fragments without any necessary proposition being expressed or any sentence being reconstructible from the utterances as a whole:

(14) A: Covent Garden?
B: Right at the lights. Then straight on up.

And, even more problematically for other frameworks, there might even not be agreement between the parties as to the intended continuation:

(15) A: All this shows is
B: that you are wrong.

With utterance shifts being possible across all syntactic dependencies, including islands and morphosyntactic dependencies, it is hard to escape the conclusion that the in principle distinction between competence and performance introduced in Chomsky (1965) must be untenable. It is further difficult also not to conclude that speaking and understanding, production and perception, must be in lockstep and definable in the same terms as each other. A further conclusion is that the sentence, a concept that develops only with writing and literacy (Miller & Weinert 1998), has no special status and, by extension, intended meanings need not be propositional as illustrated in (14). Indeed, utterances by a single individual can be of any syntactic category performing various speech acts without need of propositional reconstruction, but just by enabling action coordination in a multimodal context.

To enable a successful characterization of the interactive nature of these data relative to DS assumptions, the only assumption that needs to be abandoned is that the Axiom is an essential starting point. To the contrary, an utterance can be instigated by a requirement of any type, making the context relativity of utterance production and construal be essentially context dependent. Furthermore, given that DS is at this juncture to be construed as a grammar formalism, underpinning both production as well as perception, the very same actions determining construal will equally form the basis of production. The basis of just such a theory was proposed in a series of papers in the early 2000s (Otsuka & Purver 2003; Purver & Kempson 2004a, b; Purver et al. 2006), with *generation* defined as the process of incrementally building a partial tree as an agent utters words in sequence, just like a hearer does on hearing an utterance. However, there is one crucial difference: a speaker has in mind a partial tree expressing the intended content of the utterance, the *Goal Tree*. The Goal Tree is at least one step ahead of the generation task. At each stage of generating an utterance, partial strings and trees are tentatively extended using some word form plus its associated actions from the lexicon. Candidate processing states which include trees that subsume the Goal Tree are pursued for further extension.

Coordination between the parties, whether as hearer or as speaker, and the resultant interactivity in such exchanges, are the immediate consequences of such a joint co-construction process.[13] Moreover, what determines the success of such exchanges does not have to be seen as requiring higher order reasoning involving mind reading, as the interactive effect is directly achieved through incremental action coordination which accommodates feedback at each step-by-step subsentential stage (Gregoromichelaki & Kempson ~~2013;~~ 2013, 2016a,b).

So, without any necessary constraint on the type to be assumed as goal, and with the context being presumed to include a record of actions taken in the immediate preceding sequence of actions within an exchange, the formal definition of the generation/parsing process utilizes a restricted notion of linguistic context. A parser state is defined as a triple <*T*, *W*, *A*> where *T* is a (possibly partial) semantic tree, *W* the sequence of words and *A* the sequence of lexical and computational actions that have been used in building the tree *T*. When parsing or generating some utterance, therefore, some triple reflecting the previous utterance acts as the context for the next one. It is the presence of, and associations between, all three parts of the triple, available equally to speaker and hearer, that allow a straightforward model of dialogue phenomena. So when participant B in (16) utters a correction to A's assertion, she has available in the context, not just the word *give* but also the set of lexical actions of the word that construct the dative construction rather than the double object variant, thus bypassing completely any search of the lexicon:

(16) A: Joanne gave the only copy of her new book to Lawrence.
B: No. She gave one of the copies to Lawrence, not the only one.
(No. She gave Lawrence one of the copies, not the only one.)

[13] An explicit modelling of this cross-party check is provided in Howes & Eshghi (2021).

Instead of priming and some process of undiscerning structural alignment (Pickering & Garrod 2004, 2013; cf. Gregoromichelaki & Kempson 2019), the close coupling of parsing and generation processes, their sharing of a suitable model of context, and quotation mechanisms of general applicability (Gregoromichelaki 2018) thus allow shared utterances and various ellipsis and dialogue phenomena like subsententially pinpointed correction to be modelled in an entirely straightforward fashion.

As a formalism grounded in processing potentials, DS, especially in its application to building dialogue affordances, the 'common ground', cannot be illustrated just with exemplar trajectories of individual derivational sequences. Instead, the full range of possibilities made available by the system provide an account of how options within the unfolding derivations are orchestrated and either pursued or abandoned at each successive stage (Kempson et al. 2016). In the computational implementation of DS, the process of unfolding action potential and subsequent realization is modelled as an extended parse state representing a parse forest (Eshghi 2015; Eshghi et al. 2015) in the form of a direct acyclic graph (DAG), constituted by branching potential trajectories (*affordances*) emanating from nodes in the DAG called *Information Control States* (ICSs). In parsing a sequence of words, the parser starts from an initial tree T_0 (a requirement to construct a complete tree of a particular type), and applies optional computational and lexical actions associated with the words uttered. A simple example is shown in Figure 20.4 showing a parse of *John lost*.

In Figure 20.4, the DAG is rooted at T_0, with partial trees as nodes, and computational and lexical actions as edges (i.e. transitions between trees) (Sato 2011). L*adj, link-adjoin, thin and comp correspond to the computational actions Local *Adjunction, Link adjunction, Thinning, and Completion respectively, while 'john' is a lexical action. Different DAG paths represent different parsing strategies, which may succeed or fail depending on how the utterance is continued. Here, the path $T_0 - -T_4$ succeeds if 'John' is an argument of a verb (*John lost his keys*); $T_0 - -T_3$ would succeed if 'John' is a hanging topic with the associated constraint that the rest of the utterance contains a co-referential pronoun ~~(~~*(As for) John, he lost his keys*); and so on. This incrementally constructed DAG makes up the entire parse state at any point. The current tree along any path is the rightmost node on that path and the paths themselves constitute the linguistic context for resolving phenomena like ellipsis and pronominal anaphora (Purver et al. 2011) as well as phenomena like clarifications, corrections, and so-called 'disfluencies'. This concept of a state space DAG affording multiple routes for its traversal gives a full model of what options are licensed as possible at each stage, how some of these may atrophy at any subsequent stage, and how other options can be taken up as subsequent words are processed.[14]

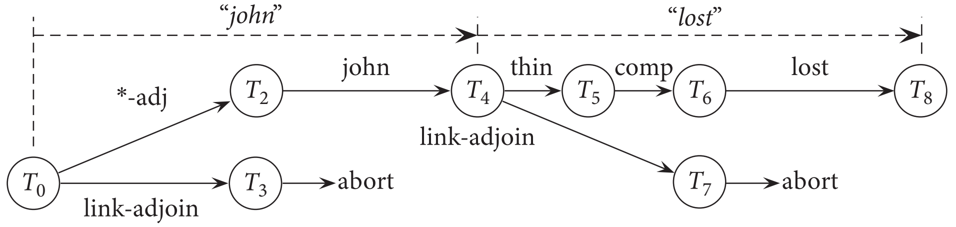

FIG. 20.4 A context DAG (parse state) for *John lost*

[14] This approach to defining an implementation for DS was first developed by Sato (2011), and then further elaborated and extended by Eshghi, Purver, & Hough (Eshghi et al. 2012, Eshghi 2015, Eshghi et al. 2023; Eshghi 2015; Purver et al. 2010, 2011; Hough 2015; Hough & Purver 2014, 2017).

In addition, this conception of the space of possible paths to a successful outcome highlights the notion of *potential* grammaticality (Cann et al. 2007) as equivalent to *acceptability* (cf. Chomsky 1965), grounding also the processing of split utterances ((10)–(11)) as triggered by 'syntactic conditional relevance' (Gregoromichelaki & Kempson 2013). In the case of potentially grammatical sequences, there is a successful sequence of steps up to a given point, although the tree is not yet complete and there may be outstanding requirements. In contrast, if there is no possible sequence of steps which allows development up to a given point, the sequence is ungrammatical and the process aborted. Grammaticality in DS therefore depends on the ability for one partial tree to unfold into another partial tree.

In this model, dialogue phenomena like false starts, self-correction, backtracking, or other repair strategies can be directly accounted for as part of the grammar formalism itself, rather than as peripheral 'performance' issues. So the framework stands in stark contrast to those in which all such phenomena are dismissed as 'noise', and not regarded as having anything to do with theories of grammar. Any such recourse to differentiation between subtheories inevitably necessitates the existence of a separate subtheory (rarely, if ever, defined) that relates decontextualized, sentence-based syntax and semantics, to the use of a language to communicate. The fact that DS needs no recourse to any such extra theoretical machinery beyond that of parsing and process undermines the whole Chomskyan assertion that the dichotomy between competence as some sort of disembodied, abstract system of language, and performance as the actual use of language to communicate is somehow necessary for a true understanding of human language and grammar. If Ockham's Razor tells us anything, it is that if a single definition captures the whole of a phenomenon, it is not necessary, indeed wrong, to define different ways of describing subparts of that phenomenon. The fact that one may account for contextualized dialogue as well as decontextualized data representing specific constructions of a language in the usual fashion of theoretical linguistics surely vindicates the DS model of human linguistic ability, and indeed knowledge, as based on parsing and production, rather than abstract concepts like Universal Grammar and innateness.

But it is not merely in its greater breadth of data that DS departs from other frameworks. It is the process-orientation which marks it out as a radical departure. On the one hand, the concepts of incrementality, underspecification, and predictivity are properties of grammar in general, not just of some subsystem of processing, and not just a genre-specific grammar for dialogue either. On the other hand, in virtue of the fact that it is processes that lies at the heart of the explanation, the concept of 'syntax' is grounded solely in the articulation of interactions, replacing in its entirety the concept of syntactic structure over words, with even tree structures being conceived as structures on an affordance landscape (Gregoromichelaki et al. 2020; Gregoromichelaki 2022). So there are no separate syntactic or morphosyntactic levels of representation; no syntactic categories for strings of words; no phrase structure rules inducing structure inhabited by sequences of words; and sequences of words are not sequences of symbols (picking out objects) but sequences of triggers opening up further actions. This stance leads DS to the view that putative universals can only be cognition-general constraints (Gregoromichelaki et al. 2020). Taking a step back from the data, there is much further work to be done. To substantiate the claim of DS of providing an integrated framework for both production and perception, the system must be demonstrably able in principle to be scaled up to apply in general contexts and across the full range of types of data. Recent developments in DS have turned to the Type Theory with Records (TTR) framework in order to provide a vocabulary generalizable to all structural types and all types of quantification in order to take up this challenge (Purver et al. 2010, 2011; Eshghi et al. 2015; Gregoromichelaki 2018).

20.5 Modelling the Interface between Cognitive and Social Interaction

More important for placing DS within the panoply of linguistic frameworks as a means of its evaluation is the natural correspondence it offers to other ongoing cognitive science endeavours. Confirmation of the need for a radical shift of perspective away from the isolationist Chomskyan paradigm is now manifold across many areas of cognitive science. First of all, there is empirical evidence from cognitive psychology, and cognitive neuroscience that emergent cognitive development builds on interactivity both well before and in the emergence of language acquisition (di Paolo & de Jaegher 2013; Heyes 2018; Rączaszek-Leonardi et al. 2018; Rącaczaszek-Leonardi & Deacon 2018; Mirski & Bickhard 2019, 2021; Tomasello 2019; Mirski & Gut 2020). There is also strong supporting evidence for an interactivist language perspective from ecological psychology in which frameworks initially developed by Gibson (1957, 1982) have been extended within the Skilled Intentionality Framework of Rietveld et al. (2018) and Bruineberg & Rietveld (2019) as an interactionist view of perception/action. This involves focus on information available in the environment, re-addressing the balance between cognition-internal constraints and the structure of the temporal/spatial environment, in so doing side-stepping the need to invoke the multiple richness of higher order and social inference. This is a perspective within which DS offers a unique and natural niche, rounding out that perspective by providing an explanation of the strength of natural language as a tool for interaction.

What DS thus offers is a '*distributed cognition*' approach (Hutchins 1995) rather than a model where autonomous agents interact only by conceptualizing others' activities and the environment. Instead, it is actions that constitute the primary cognitive processes establishing relations between an agent and other sociomaterial systems in its environment. Normativity (corresponding to 'well-formedness' or 'acceptability' in the case of language theorizing) is thus a property of the complementarity of actions that is induced by emerging regularities expressed as social affordances in a particular sociomaterial environment (Gregoromichelaki & Kempson 2015; 2016b; Gregoromichelaki et al. 2022). A grammar within such a framework is a model of what underpins participants' interactions in social exchange, building on what is available in context as affordances, giving rise to either feedback leading to modification or context update, possibly by others, in short, in spirit, a Wittgensteinian perspective on language fleshed out as a substantive formal framework (Wittgenstein 1953; Weichold & Thonhauser 2019).

Against such a back drop, the distributional semantics methodology as applied to building vector-style analyses of natural language interpretation offers a rich potential for enriching the Dynamic Syntax explication of the interpretability of natural language as essentially context dependent, but yet intrinsically flexible (Sadrzadeh et al. 2018; Purver et al. 2021), a further departure from both the original Fodorian stance and the assumption of direct truth-theoretic content to natural language expressions.

We also find here striking parallels with Bickhard's interactivist framework. Bickhard's focus is to explain the emergence of human cognition and human language capacity from more elementary forms of life grounding the explanation in naturalistic terms rather Cartesian dualistic assumptions. The appeal to processes rather than objects/substances, as the fundamental elements of reality as explanatory of both language and cognition, is common between DS and Bickhard, but also Rovelli (e.g. Rovelli 2021; Laudisa & Rovelli 2021) in his relational interpretation of quantum mechanics.

Bickhard's notion of contextually driven anticipation has the same Gibsonian provenance as DS's conception of the grammar as sets of affordances, albeit in DS such affordances are defined at the social interpersonal level of interactions. As a result, coordination, adaptation, and learning relative to the local sociomaterial context are seen as emergent products of interactions grounded in complementarity of abilities and contributions by interlocutors and without imposing the necessity of mind reading, metarepresentations, or identity of content inside people's heads. However, as Bickhard outlines (e.g. Bickhard 2009), representations can be established via adopting a reflective stance towards such interactions and a grammar (in the Wittgensteinian sense) should be able to model such conceptualizations. As in Bickhard's account, error detected by a DS system geared towards action coordination and local adjustment can for guiding learning for the participants in this system allowing them to improve their grasp of the prevalent sociocultural norms as far as linguistic regularities are concerned (see also Bruineberg & Rietveld 2014). Indeed, more strongly, it serves as the underpinning enabling the emergence of language itself (cf. Clark 2020). So, like Bickhard, albeit from a rather different direction, we urge the need for radical re-evaluation of the foundations upon which cognitive modelling has taken place over past decades. And, with DS enjoying both ecological underpinnings and grounding in a general interactivist perspective, the assumption that linguiztic theorising about syntax has to be rethought in terms grounded purely in process is begging for further development.

References

Avigad, J. & Zach, R. 2020. The epsilon calculus. In E. N. Zalta (ed.) *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University.

Barwise J. & Cooper, R. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophy* 5:159–219.

Bickhard, M. 2009. The interactivist model. *Synthese* 166(3):547–591.

Blackburn, S. & Meyer-Viol, W. 1994. Linguistics, logic and finite trees. *Bulletin of Interest Group of Pure and Applied Logics* 2:2–39.

Bruineberg, J. & Rietveld, E. 2019. Self-Organization, Free Energy Minimization, and Optimal Grip on a Field of Affordances. *Frontiers in Human Neuroscience*, 8, 599. doi: 10.3389/fnhum.2014.00599

Bruineberg, J. & Rietveld, E. 2019. What's inside your head once you've figured out what your head's inside of. *Ecological Psychology* 31(3): 198–217.

Cann, R. 2011. Towards an account of the English auxiliary system. In R. Kempson, E. Gregoromichelaki, & C. Howes (eds.) *The Dynamics of Lexical Interfaces*, pp. 279–318. CSLI Publications.

Cann, R., Kempson, R., Marten, L., Otsuka, M., & Swinburne, D. 2004. What goes left and what goes right. In D. Adger, C. de Cat, and G. Tsoulas (eds.) *Peripheries: Syntactic Edges and Their Effects*, pp. 19–47. Kluwer.

Cann, R., Kempson, R., & Marten, L. 2005. *The Dynamics of Language: An Introduction*, Vol. 35: *Syntax and Semantics*. Elsevier.

Cann, R., R. Kempson, & M. Purver 2007. Context and well-formedness: The dynamics of ellipsis. *Research on language and Computation* 5(3):333–358.

Carston, R. 2002. *Thoughts and Utterances: The Pragmatics of Explicit Communication*. Blackwell.

Chierchia, G., 1992. Anaphora and dynamic logic. *Linguistics and Philosophy*, 15:111–183.

Chomsky, N. 1965. *Aspects of the Theory of Syntax*. MIT Press.

Chomsky, Noam. 1957. *Syntactic Structures*. Mouton, The Hague.

Clark, E. V. 2020. Conversational repair and the acquisition of language. *Discourse Processes* 57(5–6): 441–459.

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

Di Paolo, Ezequiel & De Jaegher, Hanne 2013. The interactive brain hypothesis. *Frontiers in Human Neuroscience* 6:163.

Egli, U. & von Heusinger, K. 1995, The epsilon operator and E-type pronouns. In U. Egli et al. (eds.) *Lexical Knowledge in the Organization of Language*, pp. 121–141. Benjamins.

Eshghi, A. 2015. DyLan: An incremental, semantic, contextual parser for dialogue. In SEMDIAL Proceedings 2015 (GoDIAL).

Eshghi, A., Howes, C., Gregoromichelaki, E., Hough, J., & Purver, M. 2015. Feedback in conversation as incremental semantic update. In *Proceedings of the 11th International Conference on Computational Semantics*, pp. 261–271. Association for Computational Linguistics

Eshghi, A., Hough, J., Purver, M., Kempson, R., & Gregoromichelaki, E. 2012. Conversational interactions: Capturing dialogue dynamics. In S. Larsson & L. Borin (eds.) *From Quantification to Conversation, Festschrift for Robin Cooper on the Occasion of his 65th Birthday*. Tributes 19, 325–349. College Publications.

Eshghi, A., Howes, C., & Gregoromichelaki, E. 2023. Action coordination and learning in dialogue. In J.-P. Bernardy, R. Blanck, S. Chatzikyriakidis et al. (eds.), *Probabilistic Approaches to Linguistic Theory*. CSLI Publications.

Gabbay, D. 1996. *Labelled Deductive Systems*. Oxford University Press.

Gibson, J.J. 1957. *The Ecological Approach to Visual Perception*. Houghton-Mifflin, Boston.

Gibson, E. J. 1982. The concept of affordances in development: The renascence of functionalism. In J. Gibson & J. W. A. Collins (ed.) *The Concept of Development.* The Minnesota Symposium on Child Psychology, Vol. 15, 55–81. L. Erlbaum.

Gregoromichelaki, E. 2006. *Conditionals: A Dynamic Syntax Account*, PhD, King's College London.

Gregoromichelaki, E. 2012. Review of J. Ginzburg (2012) *The Interactive Stance*. Invited review for *Folia Linguistica* 47(1):293–316.

Gregoromichelaki, Eleni. 2018. Quotation in dialogue. In P. Saka and M. Johnson (eds.) *The Pragmatics of Quotation*. Springer.

Gregoromichelak 2013. clitic left dislocation and clitic doubling: A dynamic perspective on left–right asymmetries in Greek. In G. Webelhuth, M. Sailer, & H. Walker (eds.) Rightward Movement in a Comparative Perspective. John Benjamins.

Gregoromichelaki, E. & Kempson, R. 2013. Grammars as processes for interactive language use: Incrementality and the emergence of joint intentionality. In A. Capone, L. F. Piparo, & M. Carapezza (eds.) *Perspectives on Linguistic Pragmatics*. Springer.

Gregoromichelaki, E., & Kempson, R. 2015. Joint utterances and the (split-) turn-taking puzzle. In *Interdisciplinary Studies in Pragmatics, Culture and Society* (pp. 703–743). Springer International Publishing.

Gregoromichelaki, E. & Kempson, R. 2016a. Joint utterances and the (split-) turn taking puzzle. In Jacob L. Mey & A. Capone (eds.) *Interdisciplinary Studies in Pragmatics, Culture and Society*. Springer Verlag.

Gregoromichelaki, E. & Kempson, R. 2016b. Reporting, dialogue, and the role of grammar. In A. Capone, F. Kiefer, & F. Lo Piparo (eds.) *Indirect Reports and Pragmatics: Interdisciplinary Studies*. Springer.

Gregoromichelaki, E. & Kempson, R. 2019. Procedural syntax. In K. Scott, W. Clark, & R. Carston (eds.) *Relevance, Pragmatics and Interpretation: Essays in Honour of Deirdre Wilson*. Cambridge University Press.

Gregoromichelaki, E., Chatzikyriakidis, S., Eshghi, A., Hough, J., Howes, C., Kempson, R., Kiaer, J., Purver, M., Sadrzadeh, M., & White, G. 2020. Affordance competition in dialogue: The case of syntactic universals. In S. Malamud, J. Pustejovsky, & J. Ginzburg (eds.) *Proceedings of SemDial (WatchDial2020)*, Waltham, MA.

Gregoromichelaki, E., Eshghi, A., Howes, C., Mills, G., Kempson, R., Hough, J., Healey, P., & Purver, M. 2022. Language and cognition as distributed process interactions. *SemDial* 2022.

Gregoromichelaki, E., Kempson, R., Purver, M., Mills, G., & Cann, R. 2011. Incrementality and intention-recognition in utterance processing In: *Dialogue and Discourse* 2, pp. 199–233, No 1, special issue on *Incrementality in Dialogue*.

Hankamer J. & Sag, I. 1976. Deep and surface anaphora. *Linguistic Inquiry* 7(3):391–426.

Heyes, C. 2018. *Cognitive Gadgets: The Cultural Evolution of Thinking*. Harvard University Press.

Hilbert, D. & Bernays, P. 1939. *Grundlagen der Mathematik II*. Springer.

Hough, J. 2015. *Modelling Incremental Self-Repair Processing in Dialogue*. Phd thesis, Queen Mary University of London.

Hough, J. & Purver, M. 2014. Strongly incremental repair detection. In *Proceedings of EMNLP*, Doha, Qatar, October.

Hough, J. & Purver, M. 2017. Probabilistic record type lattices for incremental reference processing. In Chatzikyriakidis, S. & Luo, Z. (eds.) *Modern Perspectives in Type-Theoretical Semantics*. Springer.

Howard, William A. 1980. The formulae-as-types notion of construction. In Jonathan P. Seldin & J. Roger Hindley (eds.) *To H.B. Curry: Essays on Combinatory Logic, Lambda s Calculus and Formalism*, pp. 479–490. Academic Press.

Howes, C. & Eshghi, A. 2021. Feedback relevance spaces: Interactional constraints on processing contexts in Dynamic Syntax. *Journal of Logic, Language and Information*, 30(2), 331–362. Special issue: Dynamic Syntax.

Hutchins, E. 1995. *Cognition in the Wild*. MIT Press.

Keenan, E. & Faltz, L. 1984. *Boolean Semantics for Natural Language*, Studies in Linguistics and Philosophy 23. Springer.

Keenan, E. & Stavi, J. 1986. A semantic characteristion of natural language determiners. *Linguistics and Philosophy* 9:253–326.

Kempson, R. 1995. Ellipsis in a labelled deductive system. *Bulletin of the Interest Group in Pure and Applied Logics* 3:489–526.

Kempson, R. & Kiaer, J. 2009. Japanese scrambling and the dynamics of on-line processing. In H. Hoshi (ed.) *Language, Mind and Brain: Perspectives from Linguistics and Cognitive Neuroscience*. Kuroshio.

Kempson, R. & Kiaer, J. 2010. Multiple long-distance scrambling: Syntax as reflections of processing. *Journal of Linguistics* 46(1):127–192.

Kempson, R. Meyer-Viol, W., & Gabbay, D. 1998. VP ellipsis: Towards a dynamic, structural account. In S. Lappin & A. Benmamoun (eds.) *Fragments: Studies in Ellipsis and Gapping*, 227–290. Oxford University Press.

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

Kempson, R., Marten, L., & Otsuka, M. 2005. Right node raising, coordination and the dynamics of language processing. *Lingua* 115(4):503–525.

Kempson, R., Cann, R., Eshghi, A., Gregoromichelaki, E., & Purver, M. 2015. Ellipsis. In S. Lappin & C. Fox (ed.) *Handbook of Contemporary Semantic Theory*. Wiley.

Kempson, R., Cann, R., Gregoromichelaki, E., & Chatzikyriakidis, S. 2016. Language as mechanisms for interaction. *Theoretical Linguistics* 42(3–4):203–276.

Kempson, R., Gregoromichelaki, E., Eshghi, A., & Hough, J. 2019. Ellipsis in dynamic syntax. In Jeroen Van Craenenbroeck & T. Temmerman (eds.) *The Oxford Handbook of Ellipsis*. Oxford University Press.

Langacker, R. W. 2008. *Cognitive Grammar: A Basic Introduction*. Oxford University Press.

Laudisa, F. & Rovelli, C. 2021. Relational quantum mechanics. In E. N. Zalta (ed.) *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University.

Meyer Viol, W. P. M. 1995. *Instantial Logic: An Investigation into Reasoning with Instances*. PhD thesis, University of Utrecht. ILLC Dissertation Series 1995–11.

Miller, J. & Weinert, R. 1998. *Spontaneous Spoken Language: Syntax and Discourse*. Clarendon.

Mirski, R. & M. H. Bickhard 2019. Encodingism is not just a bad metaphor. *Behavioral and Brain Sciences* 42. DOI: 10.1017/S0140525X19001286.

Mirski, R. & Bickhard, M. H. 2021. Conventional minds: An interactivist perspective on social cognition and its enculturation. *New Ideas in Psychology* 62. https://doi.org/10.1016/j.newideapsych.2021.100856.

Mirski, R. & Gut, A. 2020. Action-based versus cognitivist perspectives on socio-cognitive development: Culture, language and social experience within the two paradigms. *Synthese* 197, 5511–5537.

Montague, R. 1970. Universal grammar. *Theoria* 36(3):373–398.

Montague, R. 1973. The proper treatment of quantification in ordinary English. In J. Hintikka, U. Moravcsik, & P. Suppes (eds.) *Approaches to Natural Language*. pp. 221–242. Reidel.

Nordlinger, R. 1997. *Constructive Case: Evidence From Australian Languages*. CSLI Publications.

Otsuka, M. & Purver, M. 2003. Incremental generation by incremental parsing. *Proceedings of Sixth Annual CLUK Research Colloquium*, 93–100.

Parsons, T. 1990. *Events* in the semantics of English: A study in subatomic semantics. In Keyser, S. J. (ed.) *Current Studies in Linguistics Series No.19*, p. 334. MIT Press.

Pickering, M. & Garrod, S. 2004. Toward a mechanistic account of dialogue. *Behavioral and Brain Sciences* 27:169–226.

Pickering, M. & Garrod, S. 2013. An integrated theory of language production and comprehension. *Behavioral and Brain Sciences* 36(4), 329–347.

Pullum, G. & Scholz, B. 2001. On the distinction between model-theoretic and generative-enumerative syntactic frameworks. In *Proceedings of Logical Aspects of Computational Linguistics 2001*, pp. 17–43.

Purver, M. & Kempson, R. 2004a. Incremental context-based generation for dialogue. In A. Belz, R. Evans, & P. Piwek (eds.) *Proceedings of the 3rd International Conference on Natural Language Generation (INLG04)*, pp. 151–160. Springer.

Purver, M. & Kempson, R. 2004b. Incrementality, alignment and shared utterances. In J. Ginzburg and E. Vallduví (eds.) *Proceedings of the 8th Workshop on the Semantics and Pragmatics of Dialogue (SEMDIAL)*, pp. 85–92. Barcelona.

Purver, M., Cann, R., & Kempson, R. 2006. Gramamrs as parsers: Meeting the dialogue challenge. *Journal of Research on Language and Computation* 4:289–326.

Purver, M., Howes, C., Gregoromichelaki, E., & Healey, P. 2009. Split utterances in dialogue: A corpus study. *Proceedings of SIGdial (Special Interest Group on Discourse and Dialogue)* 3(4).

Purver, M., Gregoromichelaki, E., Meyer-Viol, W., & Cann, R. 2010. Splitting the Is and crossing the Yous: Context, speech acts and grammar. *Proceedings of 14th workshop on Aspects of Semantics and Pragmatics in Dialogue*, 43–50.

Purver, M., Eshghi, A., & Hough, J. 2011. Incremental semantic construction in a dialogue system. *Proceedings of International. Conference on Computational Semantics*, pp. 365–369.

Purver, M., Sadrzadeh, M., Wijnholds, G., Kempson, R., & Hough, Julian. 2021. Incremental composition in distributional semantics. *Journal of Logic, Language and Information* 30:379–406.

Rączaszek-Leonardi, J. & T. W. Deacon. 2018. Ungrounding symbols in language development: Implications for modeling emergent symbolic communication in artificial systems. *Joint IEEE 8th International Conference on Development and Learning and Epigenetic Robotics*, pp, 232–237. Tokyo.

Rączaszek-Leonardi, J., Nomikou, I., Rohlfing, K., & Deacon, T. (eds.) 2018. Language development an ecological perspective: Ecologically valid ways to abstract symbols. *Ecological Psychology* 30(1), 39–73.

Rietveld, E., Denys, D., & Van Westen, M. 2018. Ecological-enactive cognition as engaging with a field of relevant affordances: The Skilled Intentionality Framework (SIF). In A. Newen, L. L. de Bruin, and S. Gallagher (eds.) *Oxford Handbook of 4E Cognition*, pp. 41–70. Oxford University Press.

Ross, John Robert. 1967. *Constraints on Variables in Syntax*. Doctoral dissertation, MIT.

Rovelli, C. 2021. *Helgoland*. Translated by Segre E. & S. Carnell. Allen Lane.

Sadrzadeh M., Purver, M., Hough, J., & Kempson, R. 2018. Exploring semantic incrementality with Dynamic Syntax and Vector Space Semantics. In *Proceedings of AixDial (SemDial)*, Aix-en-Provence.

Sato, Y. 2011. Local ambiguity, search strategies and parsing in Dynamic Syntax. In R. Kempson, E. Gregoromichelaki, & C. Howes (eds.) *The Dynamics of Lexical Interfaces*, pp. 205–34. CSLI Publications.

Sperber, D. & Wilson, D. 1986/95. *Relevance: Communication and Cognition*. Blackwell.

Thomason, R. H. & Stalnaker, R. C. 1973. A semantic theory of adverbs. *Linguistic Inquiry* 4:195–220.

Tomasello, M. 2019. The moral psychology of obligation. *The Behavioral and Brain Sciences* 43, 1–33.

von Heusinger, K. 1994. Review of Neale (1990). *Linguistics* 32:378–385.

von Heusinger, K. 1997. Definite descriptions and choice functions. In S. Akama (ed.) *Logic, Language and Computation*, 61–91. Kluwer.

Weichold, M. & Thonhauser, G. 2019. Collective affordances. *Ecological Psychology* 32(1), 329–347.

Wittgenstein, L. 1953. *Philosophical Investigations*. Blackwell.