Constructions emerging : a usage-based model of the acquisition of grammar

Beekhuizen, B.F.

CHAPTER 3

The Syntagmatic-Paradigmatic Learner

3.1 Introduction

In this chapter, I introduce the Syntagmatic-Paradigmatic Learner (SPL for short), a computational model of the acquisition of linguistic representations that constitutes the culmination of my inquiries presented in chapter 2. In that chapter, I discussed several desiderata and explananda for a usage-based learner. With those in mind, I developed a model that satisfies many desiderata and explains most of the explananda (as we will see in the later chapters) with a limited set of mechanisms and representations.

Globally speaking, SPL is an incremental learner that processes input items one by one. Each input item consists of an utterance, paired with a set of situations to which the utterance can refer. SPL tries to analyze the utterance on the basis of the situational context, its current state of linguistic knowledge, and several general processing operations. Using the resulting analysis, SPL updates and expands its linguistic knowledge. The learning gets off the ground by a procedure of analogical reasoning over recent exemplars. Using this procedure, the model is able to learn initial lexical mappings between form and meaning.

Unique features of SPL are that it performs the full comprehension and production task (desideratum D2), and acquires lexical and grammatical constructions at the same time (D3). The gradual build-up of the representations in the model through the syntagmatization and paradigmatization operations (defined below) furthermore makes SPL a faithful implementation of the usage-based conception. At the same time, it addresses those aspect of the

usage-based approach that I deemed unsatisfactorily worked out (cf. section 2.2).

As a note for readers not accustomed to reading set-theoretical and graph-theoretical definitions, and probability calculations: I will only employ the high degree of formalization in this chapter, and try to explain and motivate it in the text surrounding the formalization. The formalization is meant to show how one can operationalize certain usage-based notions.

3.2 General properties of input items to the model

3.2.1 Input items: utterances and conceptualizations of situations

SPL takes as its input pairings of an utterance and a number of conceptualizations of situations that the learner considers to be the possible conveyed meanings. The idea that the language-learning child has a conception of the possible meaning of an utterance (in a conceptualization of a situation) is a logical necessity for symbol acquisition to get started. At the very least, not all of the possible concepts a child can entertain should be considered to be signified by every utterance, as this would disallow any correct associations to be formed.

The assumption that the meaning of an utterance can be independently obtained has been commonly made, and has been labeled the Interpretatibility Requirement (O'Grady 1997, 260), put forward most eloquently by MacNamara:

> It is not too fanciful to think of the infant as treating the sentences he hears as glosses on the events that occur about him. The grammar he writes is not in Latin or in any other language, but in some neurological code of which as yet not a single letter has been deciphered. (Macnamara 1972, 12)

The most obvious source of this language-independent understanding is the perception of the situation in which the language is used (Gleitman et al. 2005, 28). In fact, the primary external source of obtaining a set of candidate meanings is experience. As we know from work such as Tomasello & Farrar (1986) and Baldwin (1993), this does not necessarily mean the *perceptual* experience of the *immediate* situation in which the utterance is produced (although that is the simplest imaginable source); it can also include concepts *inferred* on the basis of perceived situations (e.g., mental states such as intentions and attitudes) as well as non-immediate situations (concepts not present in the here and now of the speech situation, or in the child's visual field, but nevertheless deemed relevant by the child because of these inferential mechanisms). Nonetheless, the simplest source of potentially signified concepts is the perception of the situation that is spatially and temporally contiguous with the

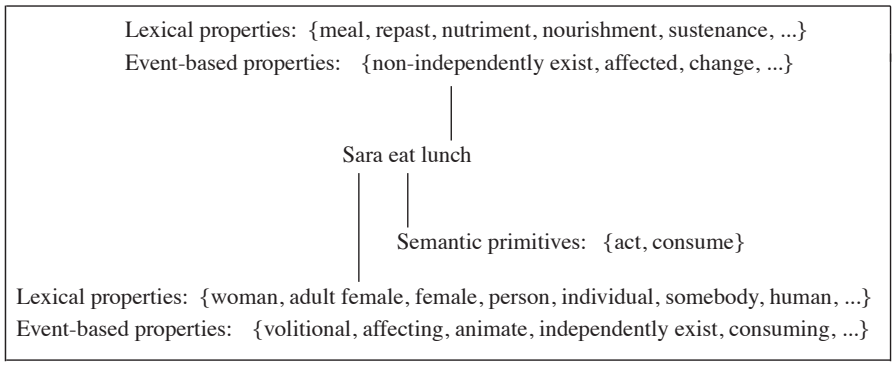


Figure 3.1: Semantic features extracted on the basis of the utterance in Alishahi & Stevenson (2010, 59).

utterance, as this is a source that requires little further cognitive sophistication to arrive at, and that is attested in other species as well (Goodall 1986, Savage-Rumbaugh, Murphy, Sevcik, Brakke, Williams & Rumbaugh 1993, Kaminski, Call & Fischer 2004).

The Interpretability Requirement may, however, be too strong compared to the situations the child finds herself in. It may be that the correct situation is not observed, for instance. Furthermore, there may be many situations besides the correct one that are initially equally likely to be the situation the utterance refers to. These issues constitute a topic that many computational models discuss, but the empirical grounding on the eventual decision they make concerning the frequency with which the correct situation is absent and the number of 'distracting' situations being co-present, is thin. For that reason, I decided to venture into this topic empirically by looking at videotaped caregiver-child interaction. The results of that exploration and an answer to the question how to provide the computational model with realistic input items are discussed in chapter 4.

3.2.2 The structure of the conceptual representations

In dealing with the acquisition of a constructicon, hierarchical representations of meaning are required. Rather than taking recourse to approaches to meaning based on formal logic, I do so by using a graphical structure with sets of features on the nodes that reflect the subtleties of conceptualization better. To do so, I make use of Alishahi & Stevenson's (2010) input-generation procedure. In their procedure, an utterance with a conceptual frame is generated. An example is given in figure 3.1. We can automatically extract hierarchical

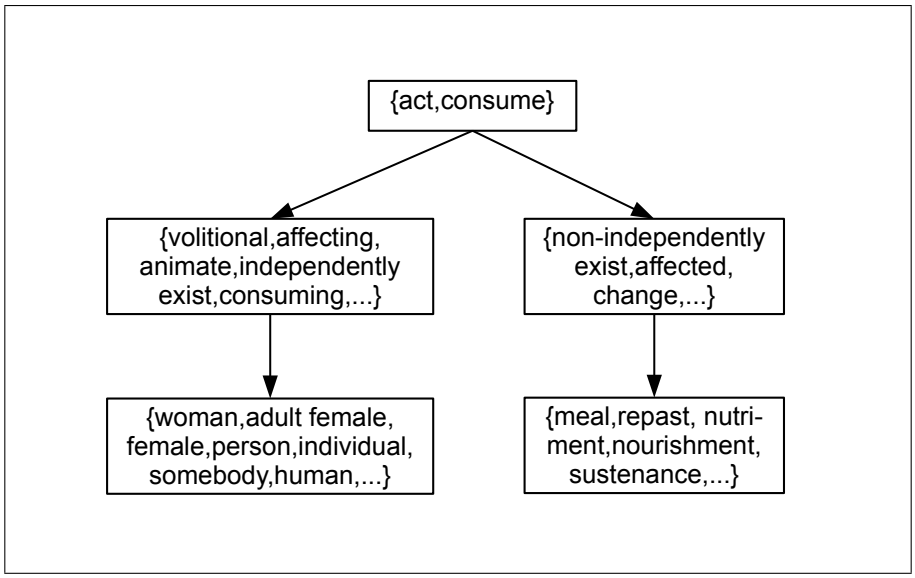


Figure 3.2: An example of a situation.

conceptual representations from Alishahi and Stevenson's procedure given the following three basic rules:

- The event node is the root node.
- The semantic role nodes, or event-based properties are daughters of the root node.
- The semantic argument nodes, or lexical properties, are daughters of a semantic role node.

For the example in figure 3.1, we obtain the structure found in figure 3.2. This structure constitutes (a conceptualization of) a situation s. As we will use conceptual graphs more in the model, it is useful to have some general definition. A situation is a graph G, which consists of a pair of a set of vertices V (or nodes), each of which contains a set of conceptual features, and a set of unlabeled directed edges (or links) E, connecting pairs of vertices in V. As we will see, the meanings of linguistic representations consist of meaning graphs as well.

In Alishahi & Stevenson's (2010) procedure, every generated situation is paired with a linguistic argument structure and a set of words filling the main

predicate and argument positions. Together, these constitute the utterance U. The argument structure for the situation in figure 3.2 would be ARGUMENT$_1$ + PREDICATE + ARGUMENT$_2$, but prepositions can also be part of the argument structure, for example in ARGUMENT$_1$ + PREDICATE + ARGUMENT$_2$ + *on* + ARGUMENT$_3$.

For now, this short exposition suffices to give an idea of the structure of the representations. Chapter 4 will deal with the exact properties of the input generation procedure.

3.3 Constructions

3.3.1 Constructions as representational primitives

The only representational unit of linguistic knowledge employed in SPL is the construction. While there are many perspectives on what a construction is within the theory of construction grammar, I start off from Verhagen's (2009) vantage point (cf. the discussion in section 2.1.1). Recall that Verhagen argues for the importance of the conceptual distinction between the contents of constructions and the roles these contents play. Crucially, a construction is a symbol, that is: a conventional pairing of a signifier and a signified. Signification entails that when the hearer observes a signifier, he infers that the speaker intends him to conceptualize the signified. The conventionality means that the signification process relies on a mutual understanding of the inferential process of signification between any two members of a language community.

The next question is what kinds of elements we assume to be present as the signifying and signified roles of a construction. Following desideratum D4-1, we assume only phonological and conceptual structure to be the elements out of which constructions are built. In the simplest case, that of words, the signifying element is a phonological string, and the signified element a conceptual representation. Grammatical constructions, however, often have non-phonological signifiers. As Verhagen argues, conceptual structure can be taken to fufill the role of a signifying element as well, and it is this content type that constitutes the signifying element in many grammatical constructions in SPL.

In grammatical constructions, we can also see a second property of the signifier, namely that it can be complex, that is: consisting of multiple elements. It is this property that allows language its expressivity: signification processes can be recursively applied to the outcomes of other signification processes, and multiple signifieds can function together as the signifier of a larger, more encompassing construction, effectively giving rise to a hierarchical interpretation of a phonological string.

We therefore assume that the signifier of a construction consists of a number of constituents, each specifying what kind of element (a phonological string, a conceptual representation, or both) should be satisfied for that con-

stituent to be recognized. The signified element of a construction is taken to be a conceptual graph that is a subgraph of a situation: as we will see, all signified conceptual structures are grounded in the situations, and as such, the meaning of the constructions is qualitatively grounded in linguistic usage events as well. Importantly, no matter how abstract, all constructions in SPL have a semantic representation as a signified. That is: I assume that there are no conventions in a language that are completely devoid of meaning.[1]

Finally, it has to be noted that this research just forms a proof of concept of the feasibility of operationalizing a usage-based constructivist approach to grammar learning, early production and comprehension. In order to model more complex phenomena than the ones we study here, richer conceptual representations and further extensions to the definition of a construction have to be assumed. To name a few: the current definition of conceptual structure as a graph without re-entrances is not suited for addressing issues of co-referentiality within sentences, as this would require multiple links in the conceptual graph to connect to the same node. In principle, there is no reason why this cannot be implemented in SPL. Furthermore, the signifiers of constructions are now strictly linearly ordered. This is unproblematic for a language like English, that relies heavily on word order and constrains the possible word orders rather strictly, but for languages with freer word order, we may want to loosen the strict linearity constraint and define constructions in terms of sets of signifying constituents which may or may not have some ordering constraints on them.

3.3.2 A formal definition of constructions and the constructicon

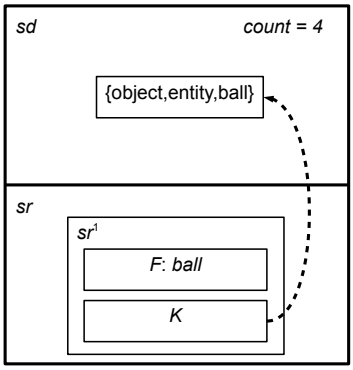
Formalizing these assumptions, we arrive at the following definition of a construction and a constructicon:

[1] This is an issue that has drawn some attention in the constructivist literature. Concerning the case of subject-auxiliary inversion in English, which is often considered to be a purely structural generalization, Goldberg (2006, ch. 8) argued that a common functional element to all cases can be found. A full treatment of the question whether purely structural generalizations exist falls outside the scope of this research, but one option that is rarely considered is that a generalization such as subject-auxiliary inversion in English may only be a *linguist's* generalization. This means that linguists may observe structural commonalities in several grammatical patterns, but that the language user does not have any sort of mental representation corresponding to these structural commonalities. That is to say: the abstraction over the various patterns is not made by the language user, but she rather maintains a number of semantically non-vacuous lower-level constructions.

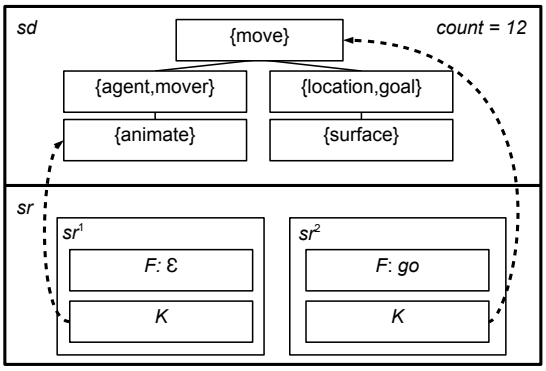
Definition of a construction and a constructicon

- Let α be a phonological element from the speaker's phonological inventory. In principle, there are no constraints on the size of α: it can consist of a single phoneme, or a string longer than a word. For the purposes of the experiments here, we set the lower bound on α to be a word in the input generation procedure.
- A construction c is a pair of a signifier sr_c and a signified sd_c, where:
  - sd_c is a conceptual graph G (as defined in section 3.2).
  - sr_c is an n-tuple (where $n \geq 1$) of constituents. Each constituent (denoted: sr_c^i for the i^{th} constituent) is a pair of a conceptual constraint K and a phonological constraint F, where
    - $K(sr_c^i)$ is a single vertex in G
    - $F(sr_c^i)$ is a string of phonological elements of any length greater than or equal to one ($F(sr_c^i) = \alpha^+$, where $^+$ denotes the Kleene plus) or unspecified ($F(sr_c^i) = \epsilon$)
  - A construction is furthermore associated with a $count_c = [0, \infty]$ reflecting how often that construction has been processed.
- We define the **head** constituent of a construction sr_c^{head} to be the constituent that has a conceptual constraint $K(sr_c^{\text{head}})$ such that $K(sr_c^{\text{head}}) = v_{\text{root}}(G)$.
- We define a **lexical** construction to be a construction c that has a single signifier, i.e., $|sr_c| = 1$.
- A constructicon Γ^t is a set of constructions $c_1, \ldots, c_n$, including their counts, at some time t

Figure 3.3 gives two examples of possible constructions. In Figure 3.3a we can see a lexical construction, containing a single signifying constituent. The construction's meaning is a conceptual graph consisting of a single vertex and no edges. The signifier consists of a conceptual constraint (K) pointing to the root vertex of G, and the phonological constraint (F) specifying that this construction can be recognized with the phonological string *ball*. Figure 3.3b, next, displays a grammatical construction, that is: a construction consisting of more than one signifying constituent. The signified meaning is a meaning graph G consisting of four vertices, each containing a set of conceptual features. The first signifying constituent sr^1 has a phonological constraint that is empty (represented as $F : \epsilon$) and a semantic constraint pointing to the vertex



(a) An example of a lexical construction.



(b) An example of a grammatical construction.

Figure 3.3: Two examples of constructions.

of G that contains the feature ENTITY. The second constituent sr^2 has a specified phonological constraint, viz. *go*, and a conceptual constraint stating that whatever is combined with this constituent must somehow combine with the feature set {EVENT,MOVE}. This second constituent, furthermore, is the head constituent of the construction, as its semantic constraint points to the root vertex of the constructional meaning.

As the box-diagrammatic format is often unwieldy, we will make use of a modified version of Langacker's (1987) bracket notation format, as introduced in chapter 1. The two constructions in figure 3.3 would be represented as follows in this format:

(24) [BALL / *ball*]

(25) [[ANIMATE] [MOVE / *go*]] |
MOVE(MOVER(ANIMATE),LOCATION(SURFACE))

3.4 Defining the space of possible analyses

When presented with an input item, the model employs its inventory of constructions and processing mechanisms to analyze it. Constructions can be applied if the string of signifying constituents is found to be present and if their meaning 'makes sense' in the context of one of the co-present situations.

I conceptualize the analysis of an utterance with constructions as a derivation process in which a fixed set of rules[2] is applied to an utterance. This is perceived for explanatory purposes as a top-down branching process (starting with a TOP-node, and terminating in the words of the utterance). However, as we will see in section 3.5.4, the model employs in the implementation a more realistic bottom-up process in which it does not keep track of all logical possibilities.

As the learner starts with no knowledge of the linguistic conventions, and as in the early stages of learning, the constructicon does not allow for full analyses of the utterance, the model will have to be robust enough to interpret parts of the utterance and situation on the basis of little knowledge. To this end, I define several rules that allow the model to create analyses of the utterance despite having little or no knowledge of linguistic constructions.

3.4.1 Mapping constructions to situations

I assume that SPL always interprets an utterance in the light of the observed situations in the input item. This means that the meaning of every used construction has to 'make sense' given at least one of the situations, or in other words: the model has to establish how the meaning of a construction is contextually resolved. In the simple case of a noun-like lexical construction, the

[2] I will call them 'rules' or 'mechanisms': these should be taken to be equivalent.

model can apply that construction if there is at least one element in the context to which the constructional meaning can refer. Potentially, there are more: a word may refer to a number of entities, possibly in multiple situations, in which case the hearer has to disambiguate to which entity the construction refers. What the model needs is a means of finding out what parts of the situational context S can be expressed by each of the constructions $c \in \Gamma$.

In order to link the constructions to the situations, subset mappings between signified meaning of the constructions and parts of situations are made. The (possibly empty) set of subset mappings $\mathbf{M}$ between the signified conceptual graph sd_c of a construction c and the situational context S consists of all legal subset mappings $\mathbf{map}_{\text{subset}}$ between sd_c and situations in the context. A mapping $\mathbf{map}_{\text{subset}}$ between sd_c and a subgraph of a situation $s \in S$ is established if and only if sd_c and the subgraph of s have the same edge structure and if the sets of conceptual features on the vertices of the subgraph of s are supersets of the sets of features on the vertices of sd_c.

Definition of subset mapping

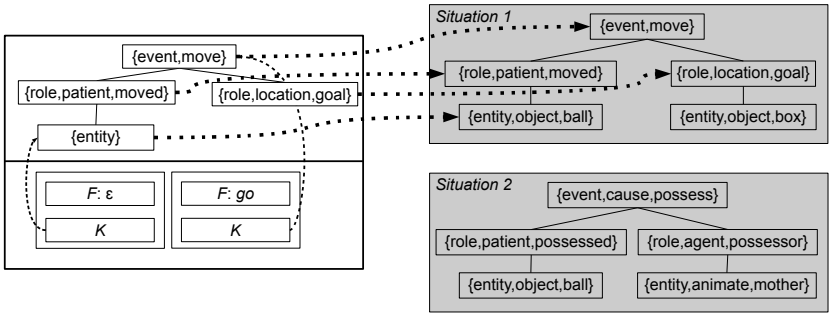
A subset mapping is an injective structure-preserving function $\mathbf{map}_{\text{subset}} = f : sd_c \rightarrow s$ between a signified constructional meaning sd_c of a construction c and a situation $s \in S$ such that

- $\mathbf{map}_{\text{subset}}(sd_c)$ is a connected subgraph of s
- The feature sets of all vertices in sd_c are subsets of the feature sets of the vertices $\mathbf{map}_{\text{subset}}(v) \in s$ they correspond with (i.e., $\forall v \in V(sd_c).v \subseteq \mathbf{map}_{\text{subset}}(v)$)

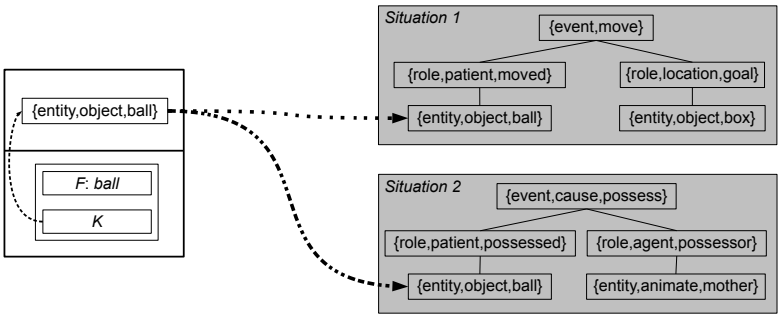
For any construction c, the set of possible subset mappings holding between sd_c and any $s \in S$ is denoted as $\mathbf{M}(sd_c)$. As per convention, we leave out the subscript when talking about subset maps, i.e., $\mathbf{map} = \mathbf{map}_{\text{subset}}$ (as opposed to other kinds of maps which we will encounter later).

Some examples of subgraph mappings are presented in figure 3.4. In the first example, we see a semi-open construction mapped to a subgraph of situation 1. Each of the four vertices of the constructional meaning maps to another vertex in situation 1, and all of the edges are preserved in the mapping. Furthermore, each vertex to which the vertices of the constructional meaning map is a subset of the conceptual features in the subgraph of the situation.

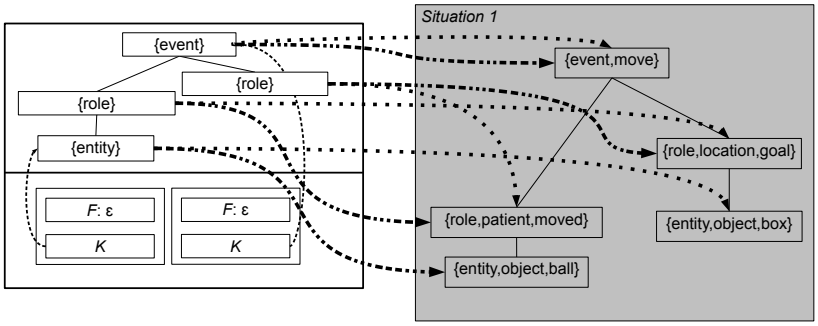
The second example shows a mapping of a lexical construction to two subgraphs, each in a different situation. The first mapping, represented as a dotted line, maps the vertex containing the feature set {ENTITY, OBJECT, BALL} to a vertex with an identical feature set in the first situation. The second mapping, represented as a triple-dash triple-dot line, maps that vertex to a vertex



(a) A subset mapping between a semi-open construction and the situational context.



(b) Two subset mappings between a lexical construction and the situational context.



(c) Two subset mappings between an open construction and the situational context.

Figure 3.4: Three examples of subset mappings. Different subgraph mappings are represented with differently patterned lines.

with an identical feature set in the second situation. In both cases, the edge structure is (trivially) preserved, and the content of the single vertex in the constructional meaning is a subgraph of each of the two vertices it maps to.

Finally, in the third example, we see what happens when we have a relatively abstract construction. Because the sets of conceptual features on the vertices of the constructional meaning are small, they have the potential of being the subset of many vertices in the situational context. In this case, the first vertex containing the conceptual feature set {ROLE} can be mapped onto either the vertex containing {ROLE, PATIENT, MOVED} of situation 1, or onto the vertex containing {ROLE, LOCATION, GOAL}, and similarly for the other vertices. Because of this, two subgraphs of situation 1 can stand in a subgraph mapping relationship with the construction.

The constraint that there needs to be at least one situational mapping in order to apply a construction is obviously an oversimplification: if a construction has a meaning that is not among the meanings considered to be relevant for communication, the model simply does not consider it. In adult linguistic communication, however, constructions can be referring to entities and events beyond what the hearer assumes the speaker to be considering, which means that these can nonetheless be retrieved and the communicative intent can be understood. Nonetheless, I believe that much of the infant's communication is based in the here-and-now of the situational context, and that, therefore, she will consider those primarily.

3.4.2 Three general constraints

Two general constraints on derivations furthermore hold. The first is that all constructions used in a derivation must be mapped, via a subset mapping, to the same situation. This is the principle of **coherence**, which ensures that the interpretation of the utterance is coherent. It relies on a communicative assumption that the speaker is trying to refer to a specific situation with her message.

The second constraint, **isomorphy**, states that the root vertices of the meanings of any two constructions used in a derivation may not be mapped to the same vertex in the situation. The principle of isomorphy constitutes a strong case of mutual exclusivity on the level of the sentence, similar to models of the acquisition of word meaning such as Fazly, Alishahi & Stevenson (2010) and Siskind (1996). It takes an intermediate position: whereas Fazly et al.'s notion of mutual exclusivity is a soft constraint, Siskind (1996, 43) goes further, stating that no two words may refer to the same part of a situation at all.

I believe Siskind's approach to be too strong: two words *can* refer to the same semantic elements. A verb like *leave* signifies a Source-Path-Goal image schema (Lakoff 1987), and a preposition like *from* does so as well. The fact that both refer to aspects of the same frame, does not preclude language users from using both in the same sentence (*I left from my house this morning*). The **isomorphy** constraint I define is non-probabilistic, but weaker than Siskind's

approach. It only states that the root vertices of the meanings of the used constructions may not map to the same vertex of a situation. In order to combine constructions in our model, we require the two constructions to share one vertex in a s, as we will see in section 3.4.5. Because of this, we need different constructions to be able to stand in a subset-mapping relationship to the same vertex in a situation (as in the case of *leave* and *from*), whereas Siskind's notion of isomorphy would preclude this.

One exception to **isomorphy** is the case in which the head constituent of a construction c is filled with another construction c'. In that case, the root vertex of $sd_{c'}$ points by necessity to the same vertex as the root vertex of sd_c. The other constituents of c and c' still have to obey **isomorphy**. The reason for this exception is that we want to allow for abstract argument structure constructions to be combined with verbs and more generally: for abstract valency patterns (i.e., without a phonological specification of the head constituent) to be combinable with lexical constructions giving a phonological specification of the head.

Unlike for **coherence**, the discrete nature of the **isomorphy** constraint is not self-evident. Exploring a more probabilistic version of isomorphy (in which multiple coverage of the same situational vertex is directly or indirectly penalized) may constitute an interesting future extension of the model.

A final constraint concerning heads is the the **single-dependent-distribution** constraint. This constraint states that the head constituent of a construction cannot be combined with another construction that has the same head, unless it is a lexical construction. This constraint prevents the recursive application of highly abstract constructions early on, which would otherwise lead to spurious bootstrapping behavior. The motivation for this constraint comes from the connection with dependency parsing the model has: given a head, certain patterns of dependents (other constituents) can be selected, but the selection can to be made only once. Cognitively, one could argue that, when selecting a verb, the speaker selects only a single, and not multiple, argument-structure constructions to express that verb with.

3.4.3 Starting a derivation: concatenation

Derivations are built using a set of processing mechanisms that are given to the model before it has any contentive knowledge of the grammatical constructions. As such, they should be considered 'innate' to the model, or at least existing prior to any linguistic input. However, they should be considered to be very general structure-building operations rather than domain-specific rules. The four operations defined by them (concatenation, rule application, ignoring, and bootstrapping) can be seen as general operations on information.

All processing mechanisms are applied to the left-most non-terminal symbol of a current derivation. A derivation starts with the TOP symbol. From a TOP symbol, we can start any number of concatenated derivations:

i TOP $\rightarrow$ START^{+}

The START symbol, then, forms the starting point for the application of pairings of a construction and a subset mapping (c, **map** pairings). We therefore add the following processing mechanism to the set:

ii START $\rightarrow (c, \mathbf{map})$

With mechanism **i**, any number of derivations can be concatenated as long as they obey the **coherence** and **isomorphy** constraints. Mechanism **i** gives the model the robustness to jointly interpret several partial analyses in early stages when it has little linguistic knowledge. As such, it can be seen as a general inferential strategy: the model understands several parts, assumes they are parts of the same message, and so interprets them jointly. Importantly, this processing mechanism remains available to the model throughout development (desideratum D6-4), although its relative importance may decrease.

3.4.4 Ignoring words

Furthermore, this concatenative top rule allows the model to integrate words that it cannot analyze into the derivation. This behavior, too, is needed in early stages, as the model simply does not have constructions to analyze all the words in the utterance. For ignoring words, we define the following rule, given that α is a minimal phonological string (in our case defined as a presegmented word).

iii START $\rightarrow \alpha$

Importantly, any $\alpha \in U$ can be ignored with rule **iii**. This is important in allowing the model the robustness to interpret complex constructions whose constituents are disjunct, i.e., by ignoring the intermediate words (applying rule **iii** for each ignored word).

3.4.5 Applying construction-mapping pairings

When applying a c, **map** pairing with rule **ii**, the constraints on its signifying constituents sr_c have to be satisfied in order to create a legal derivation. The processing mechanism **iv** specifies this, by instructing the model to replace c with its constituents sr_c.

iv $(c, \mathbf{map}) \rightarrow sr_c^1, \ldots sr_c^n$

Satisfying the constraints on each sr_c^i can be done in three ways, depending on the constraints on sr_c^i.

v $sr_c^i \rightarrow \alpha^{+}$ (if $F(sr_c^i) \neq \epsilon$)

vi $sr_c^i \rightarrow \alpha^+$ (if $F(sr_c^i) = \epsilon$)

vii $sr_c^i \rightarrow (c', \mathbf{map}')$ (only if c is not a **lexical** construction)

Rule **v**, firstly, terminates the derivation with any number of terminal nodes. In the generative process, these terminal nodes are specified by the non-empty phonological constraint $F(sr_c^i)$. In parsing, this phonological string α^+ has to be a substring of U. When the phonological constraint is not specified (i.e., $F(sr_c^i) = \epsilon$), we can bootstrap a substring of U into that constituent with rule **vi**. This is another operation, besides the concatenation process of rule **i** and ignoring words with rule **iii**, allowing the model to apply constructions despite not knowing certain lexical constructions.

Rule **vii**, finally, allows the model to fill any constituent of a construction c with another pairing of a construction c' and a subset mapping $\mathbf{map}'$. Apart from having to satisfy the general constraints of **isomorphy** and **coherence**, the new pairing $c', \mathbf{map}'$ has to satisfy the phonological and semantic constraints on sr_c^i.

Satisfying semantic constraints

Satisfying a semantic constraint means that whatever fills a constituent (from a top-down perspective) or whatever is used to recognize a constituent (from a bottom-up perspective) has a meaning that is compatible with the content of the semantic constraint. Recall that a semantic constraint on a signifier $K(sr_c^i)$ is a pointer to a single vertex v in the meaning of c. As such, it can be mapped, via the subset mapping **map** to a vertex in one of the situations.

Semantic constraint satisfaction is defined as the situation in which the root vertex of the meaning of the construction filling a constituent is mapped to the *same* vertex in one of the situations to which the semantic constraint on the constituent is mapped. More formally:

Definition of semantic constraint satisfaction

A semantic constraint $K(sr_c^i)$ of a construction c with a mapping **map** is satisfied by a pairing of a construction and a mapping $c', \mathbf{map}'$ iff

- $\mathbf{map}(K(sr_c^i)) = \mathbf{map}'(v_{\text{root}}(sd_{c'}))$

Satisfying phonological constraints

Satisfying phonological constraints in rule **vii** is a slightly more complex matter. After all, the construction c' itself is not a phonological element. However, if the head constituent of c' terminates into a phonological string α^+ that is identical $F(sr_c^i)$, we consider $F(sr_c^i)$ satisfied.

Formally, phonological constraint satisfaction works as follows:

Definition of phonological constraint satisfaction

A phonological constraint $F(sr_c^i)$ of a construction c with a mapping $\mathbf{map}_{\text{subset}}$ is satisfied by a pairing of a construction and a mapping $c', \mathbf{map}'_{\text{subset}}$ iff

- $F(sr_c^i) = \mathbf{yield}(sr_{c'}^{\text{head}})$, where the yield of a signifier $\mathbf{yield}(sr_c^i)$ is defined as the string of phonological elements α^+ governed by the derivation at sr_c^i.

The motivation for allowing derivations themselves to satisfy phonological constraints is that it allows us, for adult language, to parse modified idioms like *pull some family strings* or *pull political strings*. In those cases, the *strings* is a lexical constituent of a phonologically specified construction [[*pull*] [*strings*]]. I propose that the analysis of *pull political strings* is that the [[*pull*] [*strings*]] construction is combined with something like an [[PROPERTY] [ENTITY]] construction, where the [PROPERTY] constituent is replaced with the lexical element *political*. Similarly, if the child starts out with highly lexically specified constructions, as usage-based theory has it, allowing for the modification of a lexically specified constituent is a desirable feature of the model.

Finally, a special constraint on head constituents sr^{head} of constructions is that rule **vii** can only apply if c' is a lexical construction. I assume that a head can only distribute its roles once, meaning that if a construction is applied in which the dependent constituents of a head constituent are given, the head constituent of this construction cannot be filled with another construction which again gives the dependent constituents of the same head. We call this constraint the **single-dependent-distribution** constraint. One could argue that this constraint is overly strict: if a learner knows an [[AGENT] [*kicks*]] as well as a [[*kicks*] [PATIENT]] construction, why could we not apply both subsequently? It would give the learner more robustness for interpreting full(er) utterances early on, for instance in cases where the learner does not have an [[AGENT] [*kicks*] [PATIENT]] construction, but does have an [[AGENT] [*kicks*]] and a [[*kicks*] [PATIENT]] construction (for a proposal along those lines, see Langacker 2009).

One apparent problem is that this approach would allow for a lot of overgeneration: the head constituent of, say, a transitive construction can be filled with a transitive construction, whose head constituent can be filled with another transitive construction, and so forth. Of course, the **isomorphy** constraint limits this, and in practice it would not pose that much of a problem.

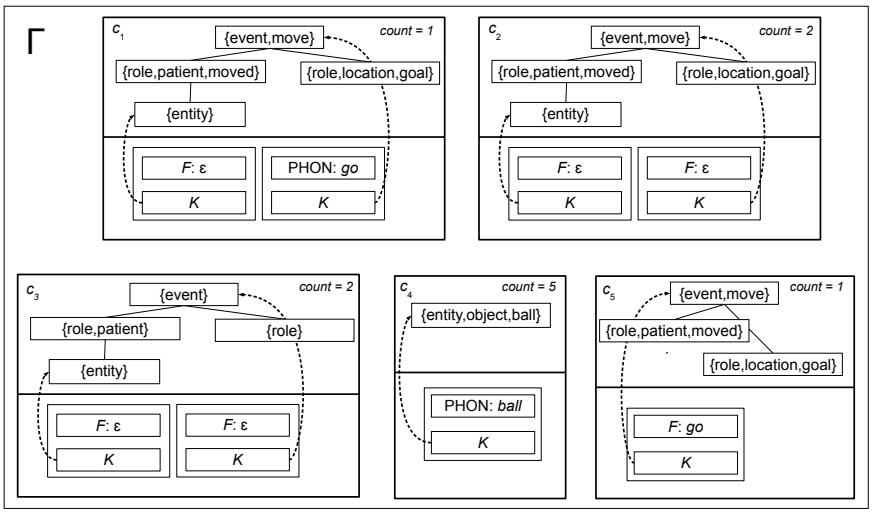


Figure 3.5: A constructicon Γ consisting of 5 constructions.

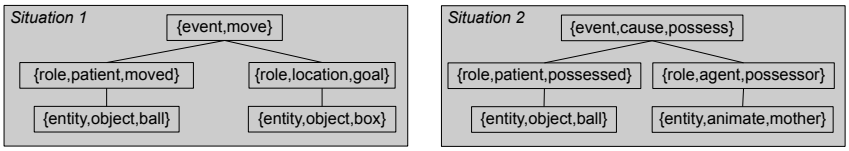
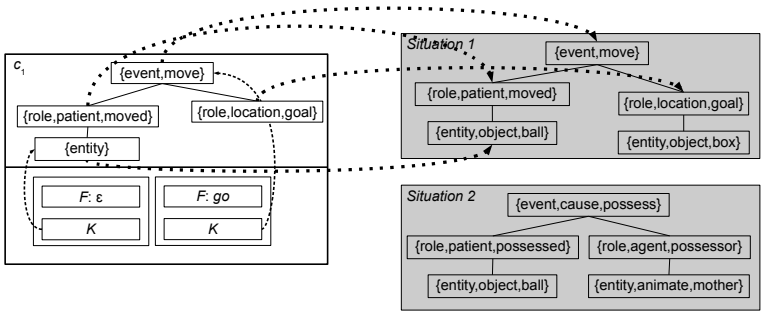


Figure 3.6: Two situations in the input item.

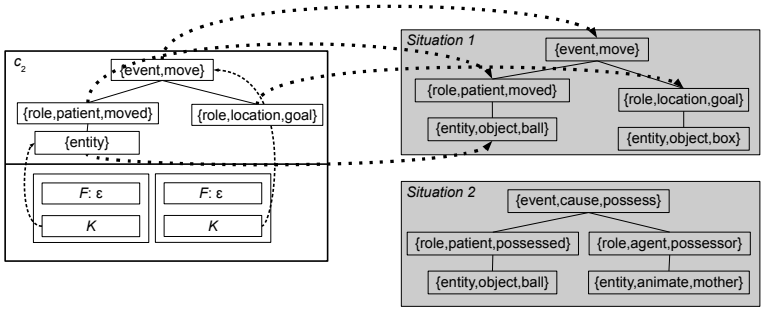
3.4.6 An example of the space of possible derivations

To illustrate the space of possible analyses of an utterance, let us take a look at an example. First, assume the constructicon in figure 3.5. This constructicon consists of five constructions. Let us further assume that the model is trying to create derivations over the utterance $U =$ *ball go there*. The situations S co-present are given in figure 3.6.

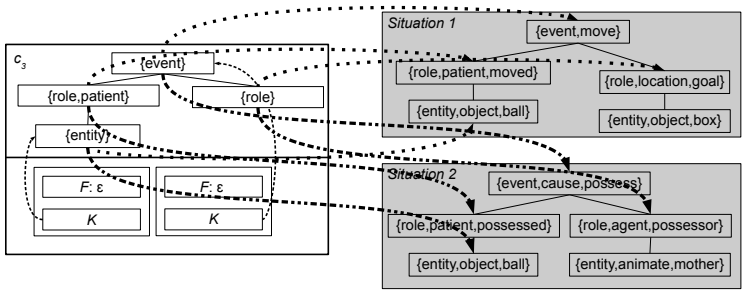
First, all subset mappings between the constructions and subgraphs of the situations are retrieved. Figure 3.7 gives all subset mappings for the five constructions and the two situations. Constructions c_1 and c_2 each have one mapping to situation s_1. Construction c_3, being more abstract, has two mappings: one to s_1 (let us call it $\mathbf{map}_1(c_3)$), and one to s_2 ($\mathbf{map}_2(c_3)$), as has construction c_4 ($\mathbf{map}_1(c_4)$ and $\mathbf{map}_2(c_4)$). The lexical construction c_5, finally, just has one



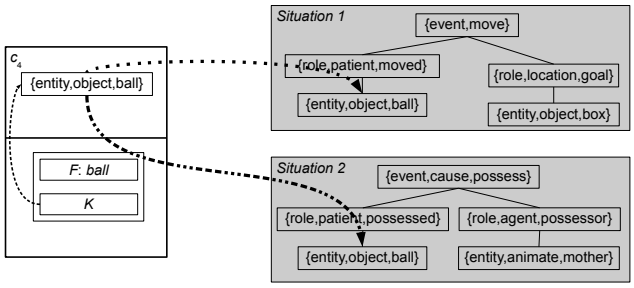
(a) The mapping between c_1 and the situations.



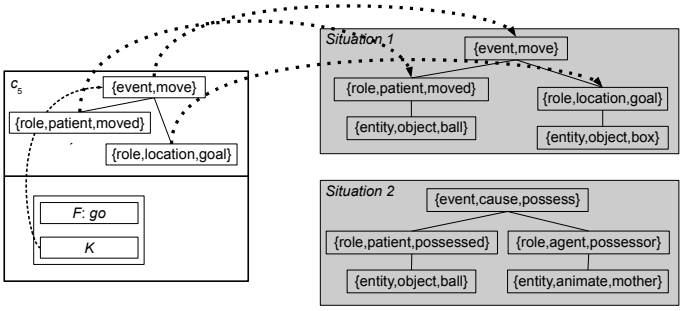
(b) The mapping between c_2 and the situations.



(c) The mapping between c_3 and the situations.



(d) The mapping between c_4 and the situations.



(e) The mapping between c_5 and the situations.

Figure 3.7: The mappings between the constructions in the constructicon and the situations.

subset mapping.

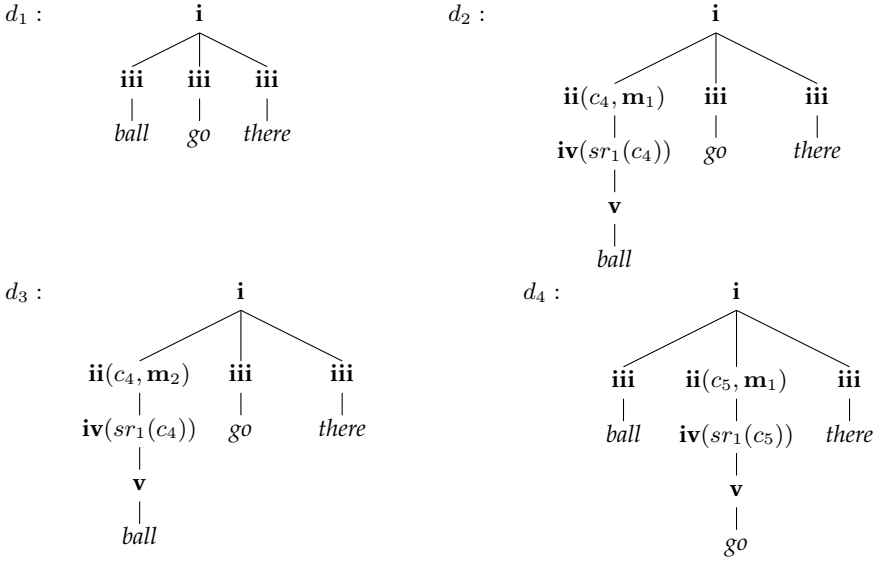


Figure 3.8: Derivations $d_1 - d_4$ for *ball go there*.

Which derivations are possible given this set of construction-situation mappings and the eleven processing mechanisms? Firstly, in the most trivial case, d_1, we ignore all words by applying rule **i** with an arity of three, followed by three times rule **iii**, with which we ignore a word. In the next three derivations, d_2-d_4, we apply one grammatical construction with rule **ii** and ignore all other words. Rule **ii** applies a c, **map** pair (represented as $(c, \mathbf{m}_i)$), which then splits into the constituents of c with rule **iv**. Because the single constituent of constructions c_4 and c_5 is phonologically specified and can be retrieved from U with rule **vi**, the derivation is valid. Note that in the case of d_2 and d_4, the construction is mapped to elements of situation s_1, and in the case of d_3 to s_2.

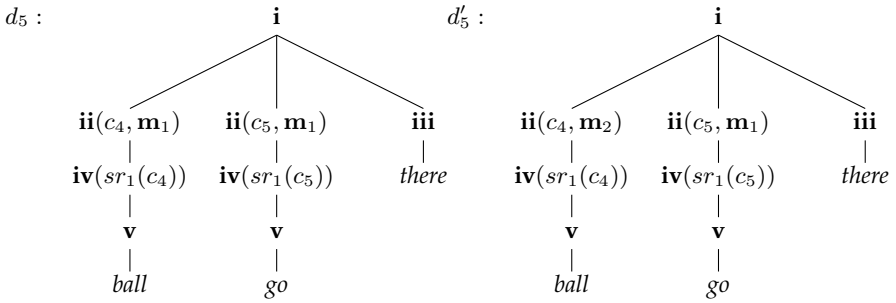


Figure 3.9: Derivations d_5 and d_5' for *ball go there*.

Concatenating the constructions c_4 and c_5 is also possible. Derivation d_5 exemplifies this: rule **i** is applied with an arity of three, after which constructions c_4 and c_5 are inserted with rule **ii**, and the final word is ignored with rule **iii**. Note that the derivation in d_5' is illegal: as c_4 is mapped to situation s_2 via $\mathbf{map}_2$ and c_5 to situation s_1 via $\mathbf{map}_1$, the **coherence** constraint is violated, rendering this derivation invalid.

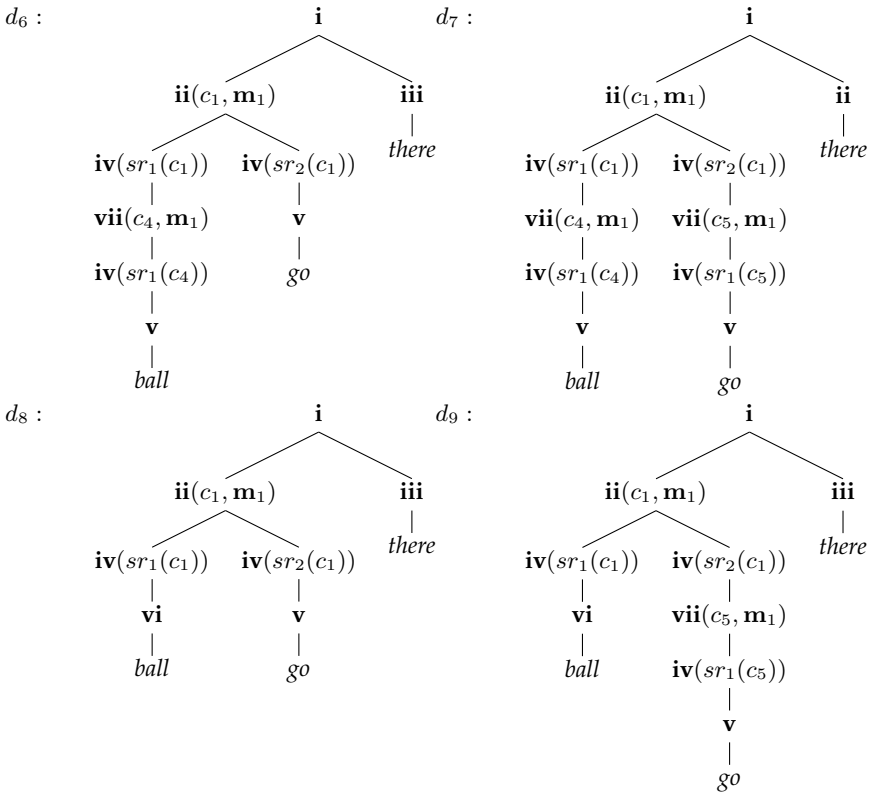


Figure 3.10: Derivations $d_6 - d_9$ for *ball go there*.

Then there are four derivations in which construction c_1 is applied and *there* is ignored. In the first two cases, d_6 and d_7, the open constituent of c_1 is filled with the pairing $c_4, \mathbf{m}_1$. In the latter two, d_8 and d_9, the word *ball* is bootstrapped by directly terminating the phonological open constituent $sr_1(c_1)$ with rule **vi**. Secondly, in d_6 and d_8 rule **vi** is applied to the recognition of the word *go*, whereas in d_7 and d_9 the second constituent of c_1 is filled with $c_5, \mathbf{m}_1$ via rule **vii**, which then terminates in the word *go*.

Construction c_2, with two open constituents, allows for more derivations. Derivation d_{10} gives the case in which c_2 is combined with c_4 and c_5 and the word *there* is ignored. However, we can also bootstrap either ($d_{11} - d_{14}$) or both

($d_{15} - d_{19}$) constituents.

Note that, although in principle construction c_1 could be combined with the second constituent of c_2, $sr_2(c_2)$, the **isomorphy** constraint precludes this. The combination of c_5 with $sr_2(c_2)$ *is* legal: although both $c_2, \mathbf{m}_1$ and $c_5, \mathbf{m}_1$ have root nodes mapped to the {EVENT,MOVE} vertex of situation s_1, $sr_2(c_2)$ is the head constituent of c_2, and hence this situation is exempt to **isomorphy**.

Finally, construction c_3 allows for even more derivations. As it has two mappings and is fully phonologically unspecified, many derivations involving bootstrapped constituents can be made. Below all 19 derivations can be found ($d_{20} - d_{37}$) in figures 3.13 and 3.14.

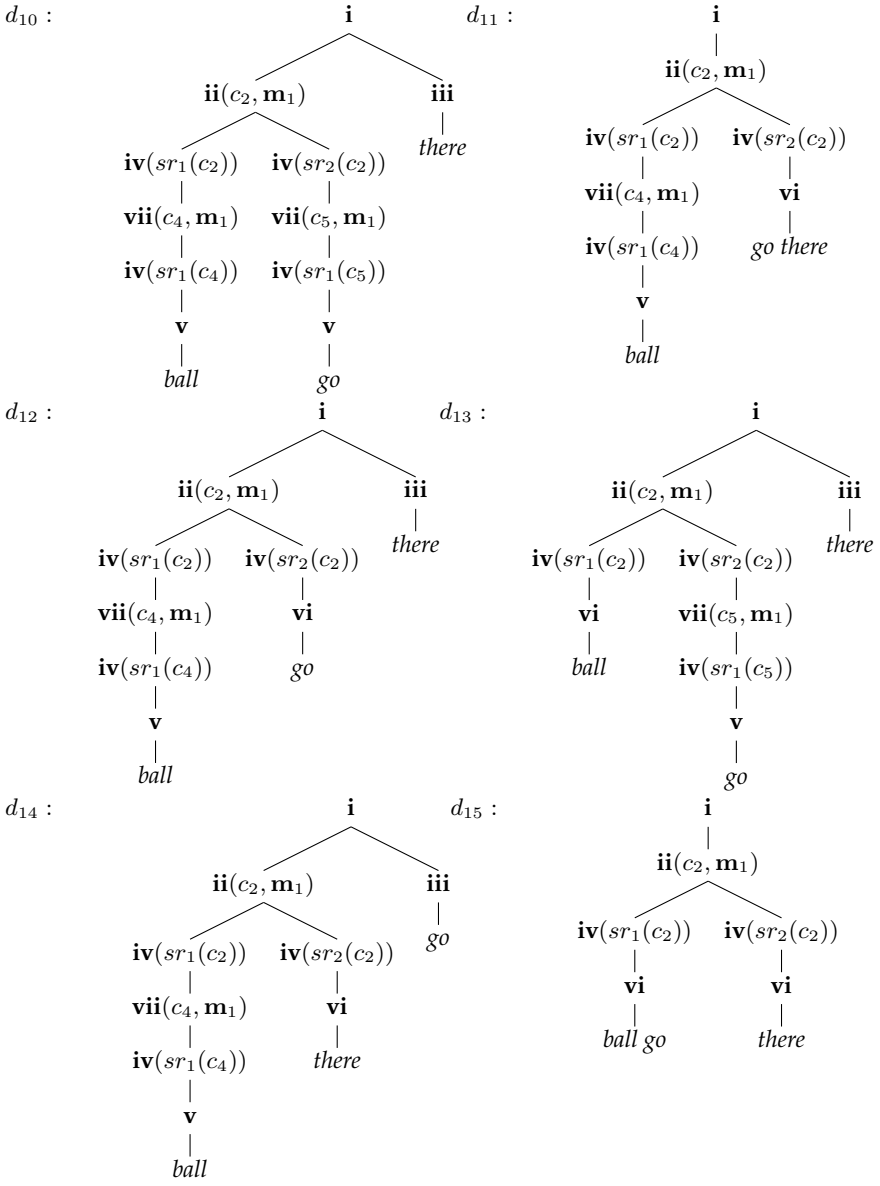


Figure 3.11: Derivations $d_{10} - d_{15}$ for *ball go there*.

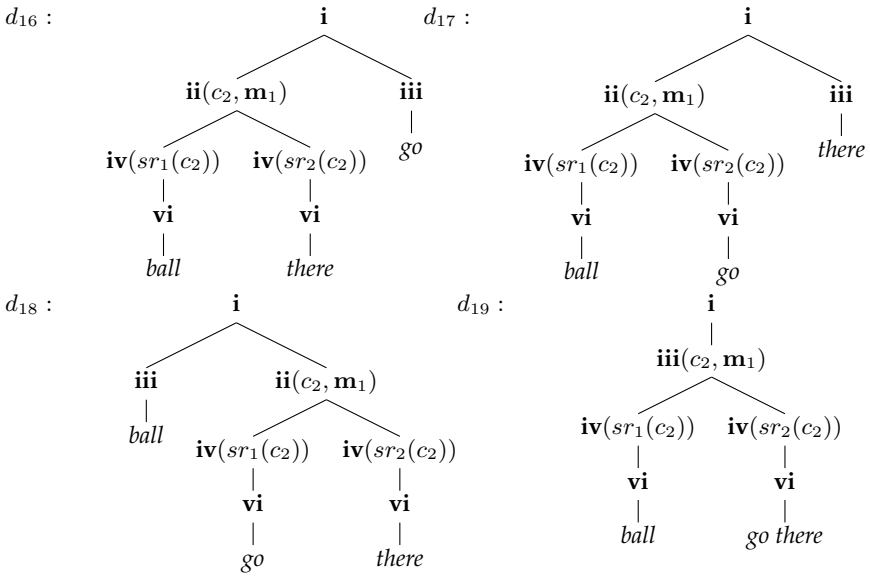


Figure 3.12: Derivations $d_{16} - d_{19}$ for *ball go there*.

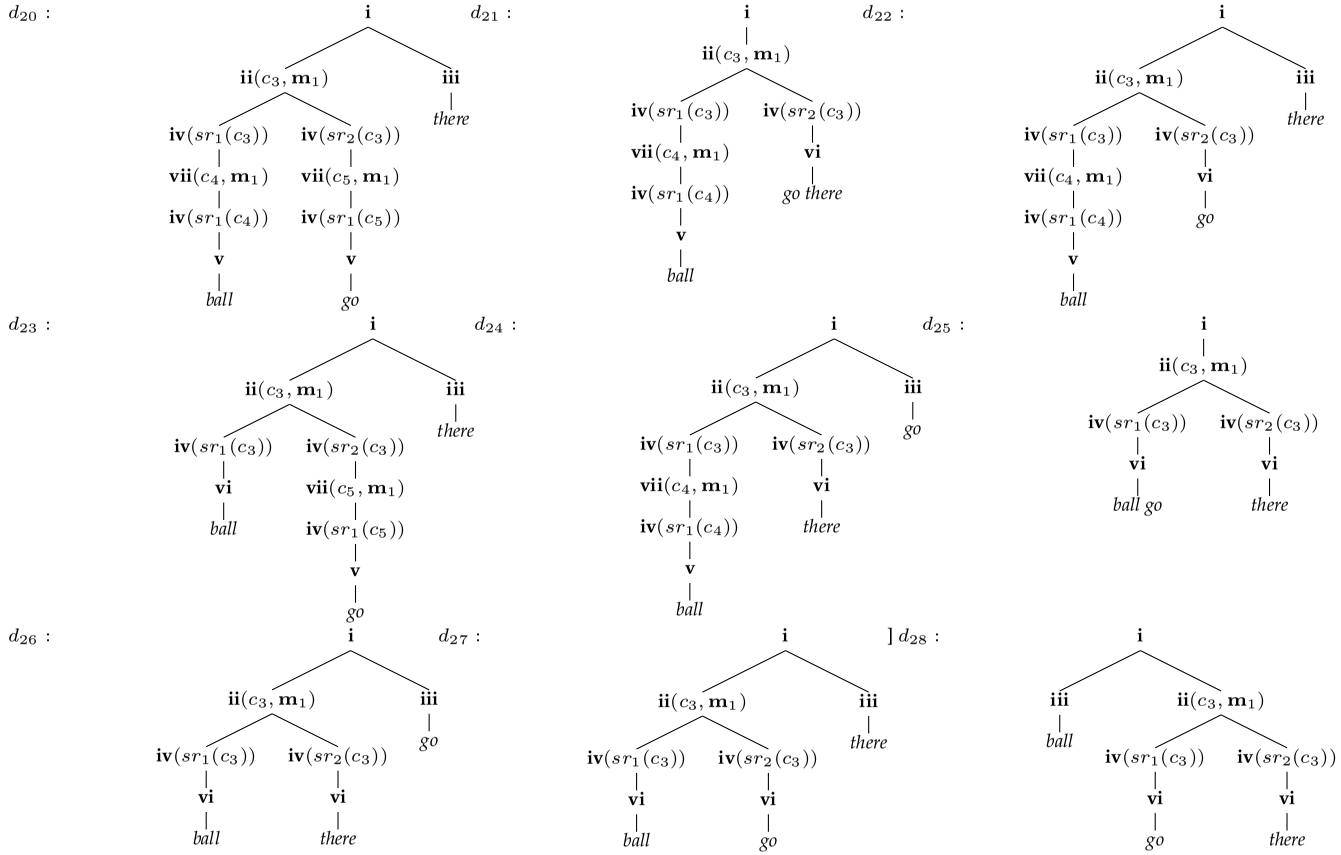


Figure 3.13: Derivations $d_{20} - d_{28}$ for *ball go there*.

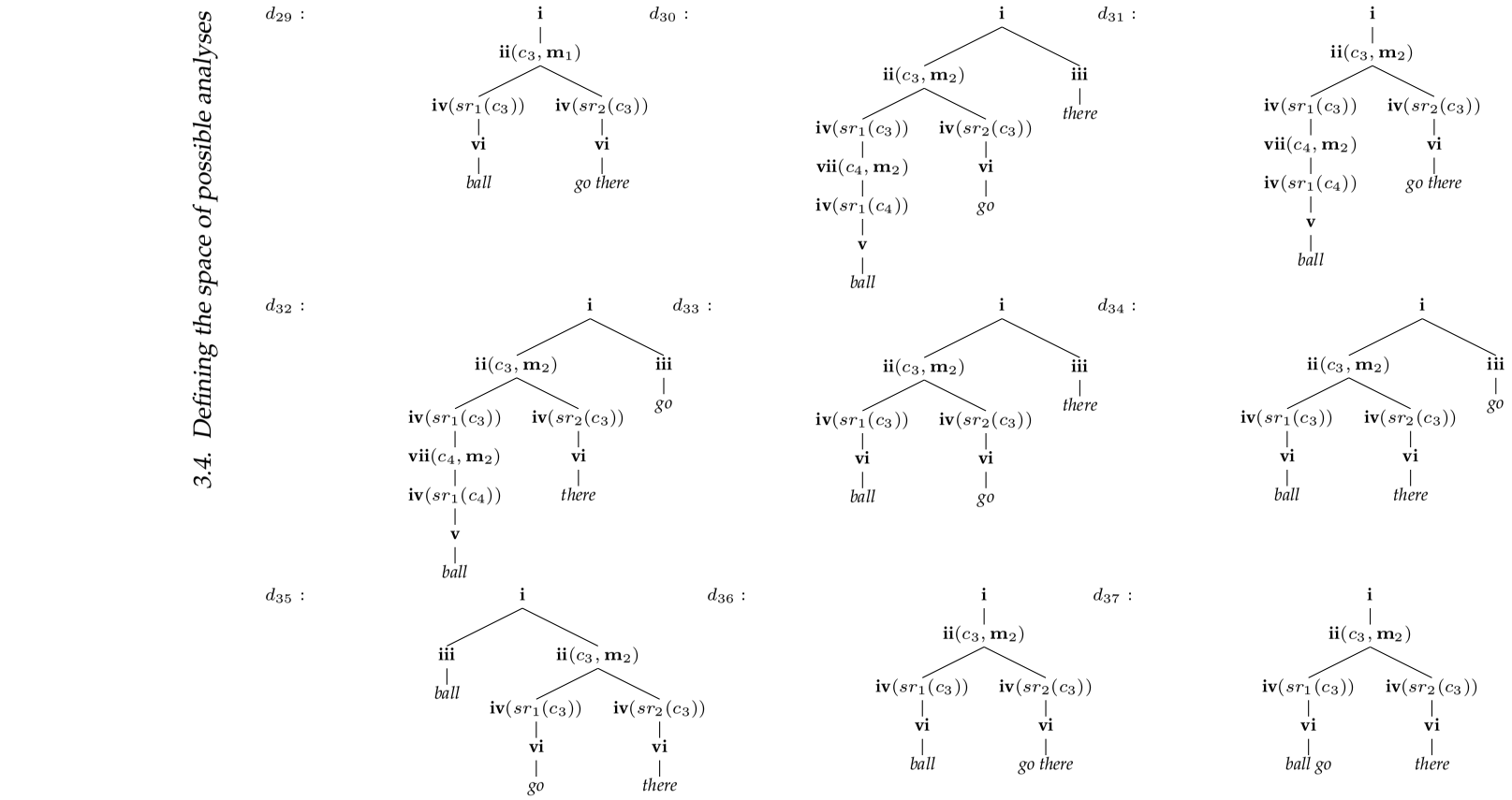


Figure 3.14: Derivations $d_{29} - d_{37}$ for *ball go there*.

	rule	probability
i	TOP $\rightarrow$ START$^+$	$\frac{1}{2}^{\lvert\text{START}^+\rvert}$
ii	START $\rightarrow (c, \mathbf{map})$	$P(c, \mathbf{map} \lvert CS_{\text{START}})$
iii	START $\rightarrow \alpha$	$P(\mathbf{u} \lvert CS_{\text{START}})$
iv	$(c, \mathbf{map}) \rightarrow sr_1(c), \ldots, sr_n(c)$	1
v	$sr_c^i \rightarrow \alpha^+$ (if $F(sr_c^i) \neq \epsilon$)	1
vi	$sr_c^i \rightarrow \alpha^+$ (if $F(sr_c^i) = \epsilon$)	$P(\mathbf{u} \lvert CS_{sr_c^i}) \cdot \frac{1}{2}^{\lvert\alpha^+\rvert} \cdot P(\mathbf{u} \lvert CS_{\text{START}})^{\lvert\alpha^+\rvert}$
vii	$sr_c^i \rightarrow (c', \mathbf{map}')$	$P(c', \mathbf{map}' \lvert CS_{sr_c^i})$

Table 3.1: Probabilities of the processing mechanisms in the analysis procedure.

3.5 Selecting the best analysis

The six processing mechanisms, along with the construction-mapping pairings used in them, will typically lead to a situation in which many derivations are possible, as we have seen in the example above. I assume that the learner selects a single best analysis among those different analyses. In this section, I describe the process for doing so, and the actual implementation, which makes the model more realistic in its processing of the utterance.

3.5.1 The probability model for derivations

We can consider the branching process defined by the seven processing mechanisms to be a probabilistic process, where the application of a rule at a point in the derivation has a certain probability of occurring given that point in the derivation. Some of these probabilities are fixed, whereas others change as the state of linguistic knowledge of the learner progresses. Table 3.1 gives the probabilities of the processing mechanisms, which will be explained below.

$P(c, \mathbf{map}|CS)$ in mechanisms **ii**, **vi**, and **vii** is defined as the smoothed relative frequency of the construction c out of all $c, \mathbf{map}$ pairings that can be applied at that point in the derivation, i.e., that compete with $c, \mathbf{map}$ for being applied. We call the set of all applicable $c, \mathbf{map}$ pairings at some point x the **competition set** given x, or CS_x. Formally, a competition set is defined as follows:

$$CS_{\text{START}} = \forall(c_{c\in\Gamma}, \mathbf{map}).\mathbf{map}(sd_c) = s \in S \tag{3.1}$$

$$CS_{sr^i_c} = \forall(c'_{c'\in\Gamma}, \mathbf{map}').\mathbf{map}'(v_{\text{root}}(sd_{c'})) = \mathbf{map}(K(sr^i_c)) \tag{3.2}$$

That is to say: given the START symbol, all pairings of a construction in the constructicon and a legal subset mapping compete with each other. Next, given a constituent of a construction sr^i_c, all pairings of a construction c' and a mapping $\mathbf{map}'$ for which the root vertex of the meaning of c' refers to the same vertex in a situation as the conceptual constraint of sr^i_c.

The smoothed relative frequency of the construction-mapping pairing is then defined as:

$$P(c, \mathbf{map}|CS) = \frac{count_c + 1}{\sum_{c', \mathbf{map}' \in CS} (count_{c'} + 1) + 1} \tag{3.3}$$

The probability of an unseen event **u**, applied in rules **iii** and **v**, is given by the remaining probability mass given a competition set CS, i.e.:

$$P(\mathbf{u}|CS) = \frac{1}{\sum_{c, \mathbf{map} \in CS} (count_c + 1) + 1} \tag{3.4}$$

Motivating the probability of rule i In the concatenation process of rule **i**, we set the probability of concatenating n derivations to $\frac{1}{2}^n$, that is: the more derivations are concatenated, the lower the probability of the overall derivation. This probability can be seen as a prior on the length of the concatenation, while, at the same time, it ensures that the probabilities of all generations given the constructicon and all possible situations sum to 1.

Motivating the probabilities of rules ii and iii Rule **ii** involves the application of a $c, \mathbf{map}$ pairing given the START symbol. This means that any construction, with any possible mapping to a situation in S can be applied. The competition set (as given in equation (3.1)) thus consists of all these construction-mapping pairings. The probability of selecting the pairing $c, \mathbf{map}$ out of all possible pairings is given by the smoothed relative frequency of c out of all applicable pairings. The fact that the probabilities are based on the counts of the constructions reflects desideratum D2-3, viz. the idea that the representational strength of the representations or their ease of retrieval should be grounded in their frequency of use.

Ignoring a word with rule **iii**, then, involves *not* selecting any construction-mapping pairing. That is: the model considers ignoring a word to be an unseen event **u**, and the remainder of the probability mass given CS_{START}, as defined in equation (3.4) is applied. Importantly, the probability of ignoring a word goes down as the size of the part of the constructicon that can be applied to

the current input item grows. This means that the more the model has learned, the smaller the probability of ignoring a word becomes.

Motivating the probability of rule iv Rule **iv** can be considered a dummy rule that expands the c of some $c, \mathbf{map}$ pairing into its signifying constituents sr_c. Because it can be trivially applied after rules **ii** and **vii**, I assign it a probability mass of 1

Motivating the probabilities of rules v, vi, and vii When substituting a signifying constituent sr_c^i of a construction for another element, several things can happen. Firstly, if sr_c^i is phonologically specified (i.e., if $F(sr_c^i) \neq \epsilon$), we can terminate the derivation directly into the phonological structure given by $F(sr_c^i)$ with rule **v**. In that case, the termination has a probability of 1.

Regardless of whether the phonological constraint on sr_c^i is specified, we can combine it with other $c, \mathbf{map}$ pairings with rule **vii**. Again, any $c, \mathbf{map}$ pairing applied at this point in the derivation stands in competition with all $c, \mathbf{map}$ pairings that can be applied at that point, that is: all $c, \mathbf{map}$ pairings that satisfy the phonological and semantic constraints on sr_c^i. The probability of applying a construction-mapping pairing $c', \mathbf{map}'$ thus is the smoothed relative frequency of c' out of all $c, \mathbf{map}$ pairings that can be used to fill sr_c^i (i.e., $CS_{sr_c^i}$), as given in equation (3.3). Again, this aspect of the probability model is grounded in desideratum D2-3, the idea that representational strength is grounded in the frequency of use.

Finally, if the phonological constraint on sr_c^i is empty, we may nonetheless terminate the derivation into a string of phonological elements α^+ with rule **vi**. This is the bootstrapping operation described earlier. The bootstrapping operation competes with all construction-mapping pairings that are applicable given sr_c^i, and, as with ignoring words, we assign it the remainder of the probability mass of $c', \mathbf{map}'$ pairings given sr_c^i. However, as the bootstrapped string α^+ can be of any length, it is undesirable if bootstrapped phonological strings of any length are equiprobable. This would lead the model to bootstrapping very long phonological strings too eagerly. Therefore, we apply the same principle as in the concatenation process of rule **i** to assign a quadratically decreasing probability over the length of the phonological string. Finally, we consider all elements α in α^+ to be ignored elements, and therefore multiply $P(\mathbf{u}|CS_{sr_c^i}) \cdot \frac{1}{2}^{|\alpha^+|}$ with the number of times rule **iii** would be applied if it was a regular 'ignore' operation, that is: with $P(\mathbf{u}|CS_{\text{START}})^{|\alpha^+|}$.

The probability of a derivation

The probability of a derivation can now be defined as the joint probability of all applications of the mechanisms in the derivation process. That is, $P(d|\Gamma, S)$ is the product of the probabilities of all rules r applied in it, as defined in table 3.1:

$$P(d|\Gamma, S) = \prod_{r \in d} P(r) \tag{3.5}$$

3.5.2 Equivalent derivations: parses

The 38 derivations we saw in section 3.4.6 give rise to different interpretations: d_3 and $d_{30} - d_{37}$ refer to situation s_2, d_1 to no situation, and the remaining derivations to s_1. Also within the groups of parses referring to the same situation, there is variation as to which parts of the utterance and the inferred linguistic structure point to which parts of the situation.

Under the usage-based assumption that linguistic knowledge can be redundantly stored at several levels of abstraction (Beekhuizen, Bod & Zuidema 2013), the model will apply constructions at varying levels of abstraction when analyzing an utterance. Several of these, however, have an identical derivational structure and refer in the same way to the same aspects of a situation. Therefore, for the purposes of analyzing an utterance they can be considered identical. We define DERIVATIONAL IDENTITY as follows:

Definition of DERIVATIONAL IDENTITY

Derivations d_1 and d_2 are derivationally identical iff

- $\text{rules}(d_1) = \text{rules}(d_2)$
- $\forall r_{i:r_i \in \text{rules}(d_1)}, r_{j:r_j \in \text{rules}(d_2)}.\mathbf{map}_i(c_i) = \mathbf{map}_j(c_j)$

Given this definition, d_1 and d_2 first have to satisfy the constraint that the strings of rules $\text{rules}(d_1)$ and $\text{rules}(d_2)$ be equal, that is: the same rules are applied in the same order. The $c, \mathbf{map}$ pairings applied in these strings of rules may, however, differ. At the same point in a derivation, the model can sometimes apply different construction-mapping pairings. Now, if for two strings of rules each mapping $\mathbf{map}$ applied at a certain point in d_1 has the same set of vertices in its codomain (the subgraph of a situation $s \in S$) as the mapping $\mathbf{map}'$ applied at the parallel point in derivation d_2, we consider the two derivations to be equal.

We define a parse or analysis a as the set of derivations that are derivationally identical to each other, and the set A as all parses given the utterance, the situations, and the constructicon. The probability of a parse can then be defined as the probability of either of the derivations subsumed by that parse being generated by the constructicon given the situation:

$$P(a|\Gamma, S) = \sum_{d \in a} P(d) \tag{3.6}$$

d_{30} :

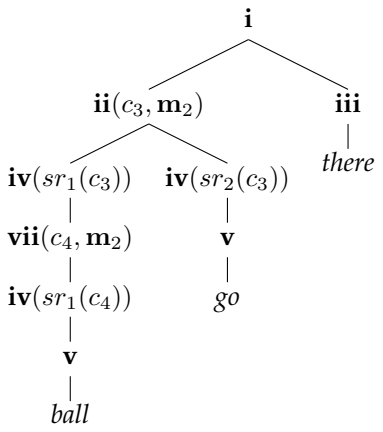


Figure 3.15: Derivation d_{30} for *Ball go there*.

Parallel c, **map** pairings in various derivations of a parse stand in a parent-child relationship to each other. As the derivational structure of the various derivations is identical, the constructions used should have the same number and types of constituents (otherwise the tree structure and choice of processing mechanisms would be different), and given the mapping equivalence, they should have meanings that are supersets or subsets of each other. We can relate this to Langacker's notion of immanence: the various derivations are not distinct events, but are all activation patterns over the same traces of linguistic usage events. The advantage of using both the parents and child constructions in the same parse is that we allow abstract constructions to back-up more concrete ones. This can be seen as a form of multiple licensing, albeit a very simple one (cf. Kay 2002)

The best parse a_{best} then, is taken to be the most-probable one.

$$a_{\text{best}} = \arg\max_{a \in A} P(a|\Gamma, S) \tag{3.7}$$

The situation mapped to by a_{best} is the identified situation $s_{\text{identified}}$, that is: the situation SPL thinks the speaker refers to. If the best parse has no mapping to any situation $s \in S$, for instance in the case when all words are ignored, one situation is selected at random to be the interpretation of the utterance. If multiple analyses are equally likely, one is selected at random to be a_{best}.

3.5.3 An example of the probability model

A single derivation

With the counts of the constructions, as given in figure 3.5, we can calculate the probabilities of all derivations. Let us look at derivation d_{30}, repeated here as figure 3.15. By substituting the left-most open symbol every time, we can order the rules as follows:

- **i**, **ii**$(c_3, \mathbf{m}_2)$, **iv**$(sr_1(c_3))$, **vii**$(c_4, \mathbf{m_2})$, **iv**$(sr^i_{c_4})$, **v**, **iv**$(sr_2(c_3))$, **vi**, **iii**

Rule **i**, applied with an arity of 2, has a probability of $\frac{1}{2}^2 = \frac{1}{4}$. Applying rule **ii** to the pairing $c_3, \mathbf{map}_2$ requires us to consider the competing construction-mapping pairings. Given the START symbol, this means we consider the competition set CS_{START}. As CS_{START} contains all possible construction-mapping pairings, it consists of $\{(c_1, \mathbf{map}_1), (c_2, \mathbf{map}_1), (c_3, \mathbf{map}_1), (c_3, \mathbf{map}_2), (c_4, \mathbf{map}_1), (c_4, \mathbf{map}_2), (c_5, \mathbf{map}_1)\}$. The probability of selecting $(c_3, \mathbf{map}_2)$ out of this competition set, or its smoothed relative frequency, is 3 (the count of c_3 plus one) over the sum of all smoothed frequencies of the elements in the competition set, plus one, or $(1+1)+(2+1)+(2+1)+(2+1)+(5+1)+(5+1)+(2+1)+1$:

$$P(\mathbf{ii}) = \frac{3}{27} \tag{3.8}$$

Next, the application of rule **iv** has a probability of 1. Applying rule **vii** afterwards again requires us to consider the competition set of the selected $c, \mathbf{map}$ pairing. In this case $c_4, \mathbf{map}_2$ is selected. The set of $c, \mathbf{map}$ pairings referring to the vertex {ENTITY, OBJECT, BALL} in situation 2 consists only of $c_4, \mathbf{map}_2$ itself, and the probability of selecting this pairing is

$$P(\mathbf{vii}) = \frac{5+1}{(5+1)+1} = \frac{6}{7} \tag{3.9}$$

The subsequent applications of rule **iv** and **v** each have a probability of 1, in the case of rule **v** because the phonological constituent of the first constituent of c_4 is specified.

After having terminated the first constituent of c_3, we look at the second constituent. Again, rule **iv** is applied with a probability of 1. After this application, *go* is bootstrapped into the constituent slot. The second constituent of c_3 has no phonological specification, and hence we take the second equation for rule **v**. This requires us to get the competition set for the second constituent, which consists of only the pairing $c_3, \mathbf{map}_2$ itself,[3] as well as for CS_{START}, which we saw in the application of rule **ii** before. The probability of an unseen event given $CS_{sr_2(c_3)}$ is 1 over 4 ($2+1$ for c_3, and 1 to smooth). The

[3]Note that the application of this pairing is ruled out by the **single-dependent-distribution** constraint, which in this case specifies that whatever fills the second constituent must be a lexical construction.

probability of an unseen event given CS_{START} is $\frac{1}{27}$, given that the denominator given this competition set is 27, as we have seen in the application of rule **ii** before. The $P(\mathbf{u}|CS_{\text{START}})$ is applied once, as the length of the phonological string is $|\alpha^+| = 1$. Similarly, the probability constraining the length of the concatenation $\frac{1}{2}$ is also raised to the power $|\alpha^+| = 1$, giving us the following probability

$$P(\mathbf{vi}) = \frac{1}{(2+1)+1} \cdot \frac{1}{2}^{1} \cdot \frac{1}{27}^{1} = \frac{1}{216} \tag{3.10}$$

Finally, we apply rule **iii** in order to ignore the word *there*. This amounts to an instance of an unseen event given the START symbol, which we have seen before, viz. $P(\mathbf{iii}) = \frac{1}{27}$. Table 3.2 below gives the probabilities of all derivations. I leave the calculation of the individual probabilities of the mechanisms as an exercise to the reader.

Several things can be learned from this example. First of all, because of the probability model, bootstrapping two adult words or bootstrapping one and ignoring one are equiprobable. Derivations d_{30} and d_{31} illustrate this.

Second, not every bootstrapping operation is equally likely. The higher the frequencies of the items in the competition set, the lower the probability of bootstrapping an element into it. Derivations d_{22} and d_{23} show this effect: because a highly frequent construction (c_4) can be fit into the first signifier, bootstrapping it becomes less likely. c_5 has a lower count, and hence bootstrapping the second constituent is relatively more likely, resulting in a probability of d_{22} that is twice as high as that of d_{23}. This effect can be seen as a pragmatic line of reasoning: I know some construction to be applicable given the constituent and the situation, so if that's a very likely construction, it is unlikely that the speaker would use a novel element to express it.

Third, we can see that the most likely derivations are those in which c_1 is used. This is a semi-open schema, with the phonological element *go* specified on the second constituent. As such, less rules have to be applied in order to arrive at a full derivations, and because of this, derivations with c_1 are globally more likely than those with c_2 and c_3, despite c_1 having a lower count than either c_2 or c_3. The most likely derivation is d_6 ($P(d_6) = \frac{1}{1701}$), in which c_1 is combined with c_4, and the last word is ignored. Here we see an effect akin to statistical pre-emption, which I will explore later in this thesis, namely that derivations with more concrete constructions use fewer rules and are thereby often more likely. It follows, however, from the general probability model and the rules, and is as such not a special built-in feature of the model.

Getting equivalent derivations

As discussed in section 3.5.2, we first look for all parses, that is: sets of derivations that are created by the same processing mechanisms, and for which every node in one derivation has the same subset mapping to a subgraph of a

derivation		P
d_1	$\frac{1}{2}^3 \cdot \frac{1}{27} \cdot \frac{1}{27} \cdot \frac{1}{27} =$	$\frac{1}{157{,}464}$
d_2	$\frac{1}{2}^3 \cdot \frac{6}{27} \cdot 1 \cdot \frac{1}{27} \cdot \frac{1}{27} = \frac{6}{157{,}464} =$	$\frac{1}{26{,}244}$
d_3	$\frac{1}{2}^3 \cdot \frac{6}{27} \cdot 1 \cdot \frac{1}{27} \cdot \frac{1}{27} = \frac{6}{157{,}464} =$	$\frac{1}{26{,}244}$
d_4	$\frac{1}{2}^3 \cdot \frac{1}{27} \cdot \frac{3}{27} \cdot 1 \cdot \frac{1}{27} = \frac{3}{157{,}464} =$	$\frac{1}{52{,}488}$
d_5	$\frac{1}{2}^3 \cdot \frac{6}{27} \cdot 1 \cdot \frac{3}{27} \cdot 1 \cdot \frac{1}{27} = \frac{18}{157{,}464} =$	$\frac{1}{8748}$
d_6	$\frac{1}{2}^2 \cdot \frac{2}{27} \cdot 1 \cdot \frac{6}{7} \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot \frac{1}{27} = \frac{12}{20{,}412} =$	$\frac{1}{1701}$
d_7	$\frac{1}{2}^2 \cdot \frac{2}{27} \cdot 1 \cdot \frac{6}{7} \cdot 1 \cdot 1 \cdot 1 \cdot \frac{2}{12} \cdot 1 \cdot 1 \cdot \frac{1}{27} = \frac{24}{489{,}888} =$	$\frac{1}{20{,}412}$
d_8	$\frac{1}{2}^2 \cdot \frac{2}{27} \cdot 1 \cdot \frac{1}{7} \cdot 1 \cdot 1 \cdot \frac{1}{27} = \frac{2}{40{,}824} =$	$\frac{1}{20{,}412}$
d_9	$\frac{1}{2}^2 \cdot \frac{2}{27} \cdot 1 \cdot \frac{1}{7} \cdot 1 \cdot \frac{2}{12} \cdot 1 \cdot 1 \cdot \frac{1}{27} = \frac{4}{489{,}888} =$	$\frac{1}{122{,}472}$
d_{10}	$\frac{1}{2}^2 \cdot \frac{3}{27} \cdot 1 \cdot \frac{6}{7} \cdot 1 \cdot 1 \cdot 1 \cdot \frac{3}{12} \cdot 1 \cdot 1 \cdot \frac{1}{27} = \frac{54}{244{,}944} =$	$\frac{1}{4536}$
d_{11}	$\frac{1}{2}^1 \cdot \frac{3}{27} \cdot 1 \cdot \frac{6}{7} \cdot 1 \cdot 1 \cdot 1 \cdot \frac{1}{34{,}992} = \frac{18}{13{,}226{,}976} =$	$\frac{1}{734{,}832}$
d_{12}	$\frac{1}{2}^2 \cdot \frac{3}{27} \cdot 1 \cdot \frac{6}{7} \cdot 1 \cdot 1 \cdot 1 \cdot \frac{1}{648} \cdot \frac{1}{27} = \frac{18}{13{,}226{,}976} =$	$\frac{1}{734{,}832}$
d_{13}	$\frac{1}{2}^2 \cdot \frac{3}{27} \cdot 1 \cdot \frac{1}{378} \cdot 1 \cdot \frac{3}{12} \cdot 1 \cdot 1 \cdot \frac{1}{27} = \frac{9}{13{,}226{,}976} =$	$\frac{1}{1{,}469{,}664}$
d_{14}	$\frac{1}{2}^2 \cdot \frac{3}{27} \cdot 1 \cdot \frac{1}{378} \cdot 1 \cdot \frac{3}{12} \cdot 1 \cdot 1 \cdot \frac{1}{27} = \frac{9}{13{,}226{,}976} =$	$\frac{1}{1{,}469{,}664}$
d_{15}	$\frac{1}{2}^1 \cdot \frac{3}{27} \cdot 1 \cdot \frac{1}{20{,}412} \cdot 1 \cdot \frac{1}{648} = \frac{3}{714{,}256{,}704} =$	$\frac{1}{238{,}085{,}568}$
d_{16}	$\frac{1}{2}^2 \cdot \frac{3}{27} \cdot 1 \cdot \frac{1}{378} \cdot 1 \cdot \frac{1}{648} \cdot \frac{1}{27} = \frac{3}{714{,}256{,}704} =$	$\frac{1}{238{,}085{,}568}$
d_{17}	$\frac{1}{2}^2 \cdot \frac{3}{27} \cdot 1 \cdot \frac{1}{378} \cdot 1 \cdot \frac{1}{648} \cdot \frac{1}{27} = \frac{3}{714{,}256{,}704} =$	$\frac{1}{238{,}085{,}568}$
d_{18}	$\frac{1}{2}^2 \cdot \frac{1}{27} \cdot \frac{3}{27} \cdot 1 \cdot \frac{1}{378} \cdot 1 \cdot \frac{1}{648} = \frac{3}{714{,}256{,}704} =$	$\frac{1}{238{,}085{,}568}$
d_{19}	$\frac{1}{2}^1 \cdot \frac{3}{27} \cdot 1 \cdot \frac{1}{20{,}412} \cdot 1 \cdot \frac{1}{648} = \frac{3}{714{,}256{,}704} =$	$\frac{1}{238{,}085{,}568}$
d_{20}	$\frac{1}{2}^2 \cdot \frac{3}{27} \cdot 1 \cdot \frac{6}{7} \cdot 1 \cdot 1 \cdot 1 \cdot \frac{3}{12} \cdot 1 \cdot 1 \cdot \frac{1}{27} = \frac{54}{244{,}944} =$	$\frac{1}{4536}$
d_{21}	$\frac{1}{2}^1 \cdot \frac{3}{27} \cdot 1 \cdot \frac{6}{7} \cdot 1 \cdot 1 \cdot 1 \cdot \frac{1}{34{,}992} = \frac{18}{13{,}226{,}976} =$	$\frac{1}{734{,}832}$
d_{22}	$\frac{1}{2}^2 \cdot \frac{3}{27} \cdot 1 \cdot \frac{6}{7} \cdot 1 \cdot 1 \cdot 1 \cdot \frac{1}{648} \cdot \frac{1}{27} = \frac{18}{13{,}226{,}976} =$	$\frac{1}{734{,}832}$
d_{23}	$\frac{1}{2}^2 \cdot \frac{3}{27} \cdot 1 \cdot \frac{1}{378} \cdot 1 \cdot \frac{3}{12} \cdot 1 \cdot 1 \cdot \frac{1}{27} = \frac{9}{13{,}226{,}976} =$	$\frac{1}{1{,}469{,}664}$
d_{24}	$\frac{1}{2}^2 \cdot \frac{3}{27} \cdot 1 \cdot \frac{1}{378} \cdot 1 \cdot \frac{3}{12} \cdot 1 \cdot 1 \cdot \frac{1}{27} = \frac{9}{13{,}226{,}976} =$	$\frac{1}{1{,}469{,}664}$
d_{25}	$\frac{1}{2}^1 \cdot \frac{3}{27} \cdot 1 \cdot \frac{1}{20{,}412} \cdot 1 \cdot \frac{1}{648} = \frac{3}{714{,}256{,}704} =$	$\frac{1}{238{,}085{,}568}$
d_{26}	$\frac{1}{2}^2 \cdot \frac{3}{27} \cdot 1 \cdot \frac{1}{378} \cdot 1 \cdot \frac{1}{648} \cdot \frac{1}{27} = \frac{3}{714{,}256{,}704} =$	$\frac{1}{238{,}085{,}568}$
d_{27}	$\frac{1}{2}^2 \cdot \frac{3}{27} \cdot 1 \cdot \frac{1}{378} \cdot 1 \cdot \frac{1}{648} \cdot \frac{1}{27} = \frac{3}{714{,}256{,}704} =$	$\frac{1}{238{,}085{,}568}$
d_{28}	$\frac{1}{2}^2 \cdot \frac{1}{27} \cdot \frac{3}{27} \cdot 1 \cdot \frac{1}{378} \cdot 1 \cdot \frac{1}{648} = \frac{3}{714{,}256{,}704} =$	$\frac{1}{238{,}085{,}568}$
d_{29}	$\frac{1}{2}^1 \cdot \frac{3}{27} \cdot 1 \cdot \frac{1}{20{,}412} \cdot 1 \cdot \frac{1}{648} = \frac{3}{714{,}256{,}704} =$	$\frac{1}{238{,}085{,}568}$
d_{30}	$\frac{1}{2}^2 \cdot \frac{3}{27} \cdot 1 \cdot \frac{6}{7} \cdot 1 \cdot 1 \cdot \frac{1}{162} \cdot \frac{1}{27} = \frac{18}{3{,}306{,}744} =$	$\frac{1}{183{,}708}$
d_{31}	$\frac{1}{2} \cdot \frac{3}{27} \cdot 1 \cdot \frac{6}{7} \cdot 1 \cdot 1 \cdot \frac{1}{8748} = \frac{18}{3{,}306{,}744} =$	$\frac{1}{183{,}708}$
d_{32}	$\frac{1}{2}^2 \cdot \frac{3}{27} \cdot 1 \cdot \frac{6}{7} \cdot 1 \cdot 1 \cdot \frac{1}{162} \cdot \frac{1}{27} = \frac{18}{3{,}306{,}744} =$	$\frac{1}{183{,}708}$
d_{33}	$\frac{1}{2}^2 \cdot \frac{3}{27} \cdot 1 \cdot \frac{1}{378} \cdot 1 \cdot \frac{1}{162} \cdot \frac{1}{27} = \frac{18}{178{,}564{,}176} =$	$\frac{1}{9{,}920{,}232}$
d_{34}	$\frac{1}{2}^2 \cdot \frac{3}{27} \cdot 1 \cdot \frac{1}{378} \cdot 1 \cdot \frac{1}{162} \cdot \frac{1}{27} = \frac{18}{178{,}564{,}176} =$	$\frac{1}{9{,}920{,}232}$
d_{35}	$\frac{1}{2}^2 \cdot \frac{1}{27} \cdot \frac{3}{27} \cdot 1 \cdot \frac{1}{378} \cdot 1 \cdot \frac{1}{162} = \frac{18}{178{,}564{,}176} =$	$\frac{1}{9{,}920{,}232}$
d_{36}	$\frac{1}{2} \cdot \frac{3}{27} \cdot 1 \cdot \frac{1}{378} \cdot 1 \cdot \frac{1}{8748} = \frac{3}{178{,}564{,}176} =$	$\frac{1}{59{,}521{,}392}$
d_{37}	$\frac{1}{2} \cdot \frac{3}{27} \cdot 1 \cdot \frac{1}{20{,}412} \cdot 1 \cdot \frac{1}{162} = \frac{3}{178{,}564{,}176} =$	$\frac{1}{59{,}521{,}392}$

Table 3.2: Probabilities of derivations $d_1 - d_{37}$.

situation as the parallel node in the other derivations. In our set of 37 derivations, we find several that are derivationally equivalent. There are ten cases in which in one derivation the pairing $c_2, \mathbf{map}_1$ is applied, and in the other $c_3, \mathbf{map}_1$. As the mappings of these pairings point to exactly the same vertices in situation 1, it is possible that two derivations, when otherwise using the same rules in the same order, are equivalent. Furthermore, there are two cases (a_7 and a_9) in which three derivations are equivalent, namely when c_1, c_2 and c_3 are applied at the same point in the derivation. In these cases, the first constituent of the three constructions is combined with c_4 and the second constituents with c_5.

What this table tells us is that parse a_6, consisting of derivations d_6 is the most likely analysis or a_{best}. In parse a_6, c_1 is combined with c_4 as its first constituent, and the second constituent is phonologically specified and can hence be terminated. This analysis is only minimally different from the second-best parse, the slightly more compositional a_7. In this parse, the second constituent is combined with another construction (c_5). The third-best analysis is a_5, consisting of just the derivation d_5. In this derivation, the two lexical constructions c_4 and c_5 are concatenated and no overarching construction is used. The three best analyses, in the bracket notation, are given below:

(26) a_6: [[ENTITY]→[BALL / *ball*] [MOVE / *go*]]

(27) a_7: [[ENTITY]→[BALL / *ball*] [MOVE]→[MOVE(MOVED,GOAL) / *go*]]

(28) a_5: ([BALL / *ball*] [MOVE(MOVED,GOAL) / *go*])

3.5.4 Implementation: lineair processing and pruning

The use of a probability model to find the best analysis is inspired by the statistical parsing tradition (Jurafsky & Martin 2009), where the disambiguation between multiple possible analyses is a massive practical problem. We may, however, doubt its cognitive reality. The most elementary of these concerns, that human beings do not actually perform such calculations, can be considered well-addressed by Jurafsky's (2003) discussion of the use of probability models in language comprehension and production. Jurafsky acknowledges that it is unlikely that people actually perform these calculations, but argues that probability models constitute a well-understood tool to model aspects of frequency and the competition between units (words, constructions).

Nonetheless, it remains unlikely that, even if the probability model is but an analytical tool, language users 'consider' all of the possible derivations that a model like SPL allows for. Starting from the insight that processing takes place linearly, and that language users do not keep track of all possible analyses (as evidenced by studies on garden-path sentences, see for instance Ferreira, Bailey & Ferraro (2002)), SPL performs the actual analysis in a bottom-up way, pruning away all but the most likely analyses (similar to the model developed by Jurafsky (1996)). As this aspect of the model was not at the heart

parse	derivations	probabilities derivations	probability parse
a_1	d_1	$\frac{1}{157,464}$	$\frac{1512}{238,085,568}$
a_2	d_2	$\frac{1}{26,244}$	$\frac{9072}{238,085,568}$
a_3	d_3	$\frac{1}{26,244}$	$\frac{9072}{238,085,568}$
a_4	d_4	$\frac{1}{52,488}$	$\frac{4536}{238,085,568}$
a_5	d_5	$\frac{1}{8748}$	$\frac{27,216}{238,085,568}$
a_6	d_6	$\frac{1}{1701}$	$\frac{139,967}{238,085,568}$
a_7	d_7, d_{10}, d_{20}	$\frac{1}{20,412} + \frac{1}{4536} + \frac{1}{4536}$	$\frac{116,640}{238,085,568}$
a_8	d_8	$\frac{1}{20,412}$	$\frac{11,664}{238,085,568}$
a_9	d_9, d_{13}, d_{23}	$\frac{1}{122,472} + \frac{162}{238,085,568} + \frac{1}{1,469,664}$	$\frac{2268}{238,085,568}$
a_{10}	d_{11}, d_{21}	$\frac{1}{734,832} + \frac{1}{734,832}$	$\frac{648}{238,085,568}$
a_{11}	d_{12}, d_{22}	$\frac{1}{734,832} + \frac{1}{734,832}$	$\frac{648}{238,085,568}$
a_{12}	d_{14}, d_{24}	$\frac{1}{1,469,664} + \frac{162}{238,085,568}$	$\frac{324}{238,085,568}$
a_{13}	d_{15}, d_{25}	$\frac{1}{238,085,568} + \frac{1}{238,085,568}$	$\frac{2}{238,085,568}$
a_{14}	d_{16}, d_{26}	$\frac{1}{238,085,568} + \frac{1}{238,085,568}$	$\frac{2}{238,085,568}$
a_{15}	d_{17}, d_{27}	$\frac{1}{238,085,568} + \frac{1}{238,085,568}$	$\frac{2}{238,085,568}$
a_{16}	d_{18}, d_{28}	$\frac{1}{238,085,568} + \frac{1}{238,085,568}$	$\frac{2}{238,085,568}$
a_{17}	d_{19}, d_{29}	$\frac{1}{238,085,568} + \frac{1}{238,085,568}$	$\frac{2}{238,085,568}$
a_{18}	d_{30}	$\frac{1}{183,708}$	$\frac{1296}{238,085,568}$
a_{19}	d_{31}	$\frac{1}{183,708}$	$\frac{1296}{238,085,568}$
a_{20}	d_{32}	$\frac{1}{9,920,232}$	$\frac{24}{238,085,568}$
a_{21}	d_{33}	$\frac{1}{9,920,232}$	$\frac{24}{238,085,568}$
a_{22}	d_{34}	$\frac{1}{9,920,232}$	$\frac{24}{238,085,568}$
a_{22}	d_{35}	$\frac{1}{9,920,232}$	$\frac{24}{238,085,568}$
a_{23}	d_{36}	$\frac{1}{59,521,392}$	$\frac{4}{238,085,568}$
a_{24}	d_{37}	$\frac{1}{59,521,392}$	$\frac{4}{238,085,568}$

Table 3.3: All parses A for *Ball go there*.

of my research, the implementation of the parser simply satisfies these constraints, but more realistic processing models can be thought of. In line with desideratum D5-2, I implemented the parser of SPL as follows.

SPL processes the words one by one. Over a span of words up to a certain word, a (possibly empty) set of derivations can be formed, which can be derivationally equivalent, and hence form a set of parses. From among this set of parses over a span, SPL only keeps the most likely one (or ones if there are multiple equiprobable parses) and discards the rest. When processing the next word, only the parses that are still active can be used to be combined into larger parses. Technically, the model employs an adaptation of the Cocke-Younger-Kasami algorithm that allows for words to be ignored, and prunes every cell in the matrix to the most likely analysis.

The motivation for this way of implementing the model not only comes from processing studies, but also from Langacker's (1988) discussion of processing, where he argues that when multiple units are in competition, a language user only selects a single one as the active unit. The implementation therefore not only constitutes an attempt to adhere to processing studies, but is also faithful to the description of processing within the theoretical framework.

3.5.5 SPL as a usage-based processing model

The derivation process described in this section allows the model to do comprehension on the basis of an utterance and a set of situations. As we will see later in this chapter, the production of an utterance on the basis of a situation is also among the model's possibilities, and therefore the model satisfies desideratum D2 (comprehensiveness). The model furthermore satisfies desiderata D5-1 (heterogeneous structure building) and D6-4 (developmental continuity) by having a set of diverse processing mechanisms that remain available over time. In the actual implementation of the way SPL performs its analyses, the model can be said to satisfy desideratum D5-2, although this aspect is not at the center stage of this research, and likely more realistic models of processing can be developed.

3.6 Learning

The resulting best parse a_{best} from every input item constitutes the input for the learning procedure. The constructions used in the best parse are reinforced (**reinforcement**), the result of any concatenative process is stored (**syntagmatization**) and any new possible abstractions that can be made are added to the constructicon (**paradigmatization**). Finally, the learner stores a limited number of recent best parses and the situations they were assumed to refer to, and employs a simple form of **cross-situational learning** to extract initial representations.

The learning model presented here aims to make three contributions to usage-based theory. First, SPL uses the syntagmatization operation to gradually build up longer constructions. The build-up of increasingly long constructions is a feature needed by the model to satisfy the law of cumulative complexity (D6-1). Second, the paradigmatization operation extracts any overlaps between the more concrete constructions and involves no grammar-wide evaluation of how useful the abstraction is. As I will argue, these features make the model conceptually congruent with the learning-by-processing and immanence view (cf. desiderata D4-3 and D6-2). Third, the model is the first usage-based model of language acquisition that is shown to acquire both lexical and grammatical constructions at the same time (cf. desideratum D2-8).

3.6.1 Reinforcement

The simplest form of learning is the reinforcement of the constructions employed in the derivations of the best parse.

Maximally concrete constructions

Langacker (2009) argues that the maximally concrete representations of the situation and the linguistic units used leave a trace in memory when they are processed. I operationalize this idea as follows. Recall that in an analysis a, there are several parallel derivations (i.e., every step in the derivation is the same, although different constructions may be used). For every such parallel step $\mathbf{s}$ in the derivation where a construction-mapping pairing $c, \mathbf{map}$ is applied, a maximally concrete construction $\mathbf{mcc}$ is extracted. We assume $\mathbf{mcc}$ to have as its meaning $sd_{\mathbf{mcc}}$ the subgraph of the situation to which the meaning of c maps via $\mathbf{map}$. The meaning of this novel construction thus directly reflects the conceptualization of the usage event in full detail. The signifiers of $\mathbf{mcc}$ consist of the signifiers of c, where the phonological constraints will be specified with whatever substring of the utterance is filling them, thus reflecting the utterance of the usage event in full detail.

In the case of a string that has been bootstrapped into a signifier of a construction c, we assume a novel construction $\mathbf{mcc}$ with as its signified meaning $sd_{\mathbf{mcc}}$ the vertex in the situation to which sr_c^i maps via the mapping $\mathbf{map}$ paired with c. The signifier of this bootstrapped construction $\mathbf{mcc}$ is then a pair of the string of words bootstrapped and a semantic constraint pointing to the vertex that constitutes $sd_{\mathbf{mcc}}$.

All maximally concrete constructions ($\mathbf{mcc}$s) are then added to the constructicon Γ (if they are not already present in it) with a count of 0. Example 3.16 illustrates the extraction of the maximally concrete constructions out of parse a_7. Importantly, the $\mathbf{mcc}$ for the second step of the derivation (the application of rule **ii**) is a phonologically specified grammatical construction. This construction, as opposed to the second and third $\mathbf{mcc}$s, is not present yet in the constructicon and added with a count of zero if a_7 were the best parse.

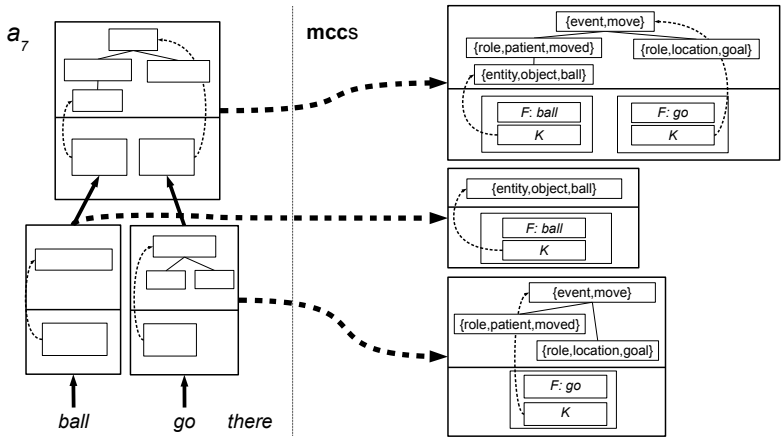


Figure 3.16: Extracted maximally concrete constructions from parse a_7.

The storage of maximally concrete representations is needed to account for prototype effects. A group of constructions in the constructicon whose meaning stands in superset-subset relations to each other may regularly map to the same subgraph of a situation in the analysis. However, if some more concrete constructions (i.e., constructions having more conceptual features specified in the constructional meaning sd) are used more frequently, we expect them to be more readily applicable than equally concrete constructions that are not as frequent. Now, if we only reinforce the more abstract constructions used, we cannot keep track of the frequency of the more concrete ones. Therefore, adding the maximally concrete ones, and generalizing over them in the abstraction step of the learning procedure (cf. section 3.6.3) allows us to keep track of this information.

This approach is similar to Alishahi & Stevenson's (2010) clustering approach, where frequently occurring conceptual features have more weight in the recognition of a construction. However, because in SPL the cluster can be said to be stored in a distributed fashion (a cluster in Alishahi & Stevenson's (2010) approach would correspond to a number of constructions in the constructicon in my approach), a 'cloud' of constructions may have multiple prototypes. That is to say: there may be two distinct sets of features being prototypical for a construction and both would be seperately stored in my approach, whereas in Alishahi & Stevenson's (2010) approach the association strength of the features is averaged over when they are clustered together.[4]

[4]However, if they are too distinct, they will form different cluster. The point is that with a hard

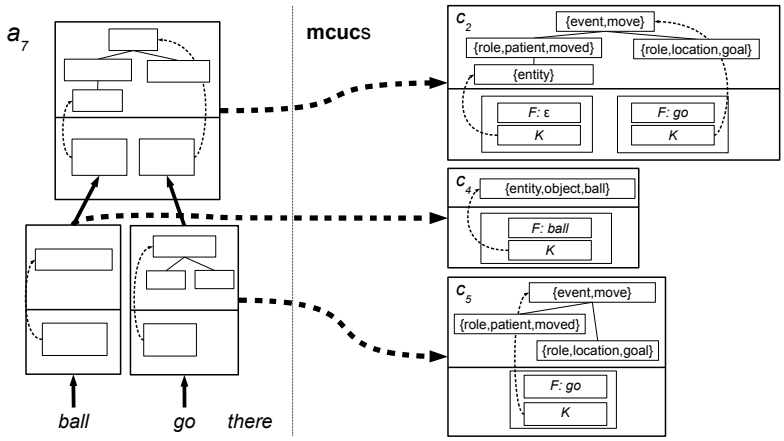


Figure 3.17: Extracted maximally concrete used constructions from parse a_7.

Reinforcement for maximally concrete used constructions

Secondly, not only the most concrete constructions are added to the constructicon, the constructions that are actually used are reinforced as well. The model does not reinforce all constructions used in all derivations of the best analyses, but only the most concrete *used* constructions per parallel step $\mathbf{s}$ in the derivation or: $\mathbf{mcuc}(\mathbf{s})$. Note that these are not necessarily the $\mathbf{mcc}(\mathbf{s})$ as defined in the previous paragraph. A construction c in any derivation in a_{best} is a maximally-concrete used construction if there is no other construction c' at the same step $\mathbf{s}$ in another derivation in a_{best} whose meaning is a superset of the meaning of c.

For simplicity's sake, I assume that after every input item, one 'count' can be distributed over the various most concrete constructions per step. That is: if there is a single derivation in the best parse, all constructions in that derivation are updated with 1. However, we will sometimes have multiple maximally-concrete used constructions for a certain step of the derivation. In that case, we distribute the count of 1 uniformly over the maximally-concrete used constructions at that step of the derivation. The update function thus is defined as follows:

$$count_{c^t} = \begin{array}{ll} count_{c^{t-1}} + \frac{1}{|\mathbf{mcuc}(\mathbf{s})|} & \text{if } c \in \mathbf{mcuc}(\mathbf{s}) \\ count_{c^{t-1}} & \text{otherwise} \end{array} \tag{3.11}$$

Figure 3.17 gives an example of the extraction and update of the maximally-concrete used constructions for parse a_7. The main difference with

the **mccs** in figure 3.16 is that the reinforced construction is not fully phonologically specified and has a more abstract signified conceptual representation.

The reason for using maximally-concrete used constructions, is that only those constructions are reinforced that are used productively. Their 'parents' in the constructional network that may be used in parallel steps in other derivations of a_{best} do not get reinforced, as there is a more concrete construction 'blocking' their update. This could be regarded as a form of pre-emption in the reinforcement procedure. Alternatively, we could say that the more abstract constructions are only motivating the use of the use of the maximally concrete constructions, backing them up with their probability mass.

A desirable effect of this procedure is that more abstract constructions are only reinforced when they are used productively, that is: in novel situations where no more concrete daughter constructions of those constructions can be used. This reflects Bybee's (2006) ideas about type and token frequency: the more *novel* instances of an abstract pattern are found, the more distinct types it can be said to have, and the more it will be reinforced.

3.6.2 Syntagmatization

Syntagmatization allows for the gradual build-up of the valency of constructions (i.e., the number of slots they have). Postponing the formal definition of the process for now, syntagmatization as a learning process is derived from the same general gradualist starting points many usage-based developmental theorists start off from (Tomasello 2003, Goldberg 2006). Despite being a gradualist take on the growth of grammar, this notion has not been worked out in detail by either of these theorists. If we want to adhere to Brown's law of cumulative complexity (desideratum D6-1), we have to assume that at least something akin to this learning process has to take place in the language-learning child

The fact that early productions often have fewer arguments expressed can, to my mind, be explained most readily if we assume that the constructions underlying these productions have more restricted valency patterns than later constructions. Most developmental approaches assume a combination of richer linguistic structure plus the deletion of some elements (Bloom et al. 1975). I believe this to be (1) a less parsimonious explanation, and (2) not in line with findings such as those presented by Theakston et al. (2012), who show that productions with transitive verbs and a single argument (SV and VO-utterances) have a different profile than productions at the same age with transitive verbs and two arguments (SVO-utterances). I interpret this fact as suggesting that the child uses different representations to generate SV, VO, and SVO-utterances respectively.

Similarly to the hypothesis that the various paradigms of a construction are gradually learned (which is typically called 'abstraction'), I assume that the

clustering operation, the model needs to decide and the cluster takes on a centroid representation.

syntagms constituting adult constructions are also acquired in an item-based, piecemeal way. Most developmental theorists, and many usage-based computational models discussed in the previous chapter assume that the learner is able to process the complete utterance and understand the valency relations between several elements of the utterance. Over these maximally concrete valency relations, then, more abstract constructions are learned. To my mind, this approach overlooks two other steps which we should expect to take place simultaneously, viz. the acquisition of lexical constructions and the acquisition of the linguistic realization of the semantic valency relations of these words. Syntagmatization takes care of this latter process. The gradual build-up of grammatical syntagms is reminiscent of Freudenthal et al.'s (2010) approach. Their MOSAIC model gradually builds up an inventory of strings of words to process utterances. The SPL model takes a similar approach, but combines it with a semantic parsing approach.

Implementation

Recall that a derivation can contain a number of concatenated constructions by the application of rule **i**. These constructions are understood by the model as being part of the same communicative intent. What syntagmatization does, then, is to take these concatenated constructions, look for constructions whose meanings stand in a semantic head-dependent relation to each other, and extend the 'head' constructions expressing that semantic head with the 'dependent' constructions.

Formally, the set of concatenated derivations consists of all applications of construction-mapping pairings that are directly governed by rule **i**. We use the maximally-concrete construction **mcc** for every construction-mapping pairing. For every construction c in this set with the meaning sd_c, we take all other constructions c' in this set whose meaning $sd_{c'}$ refers to a child or grandchild of the root vertex of sd_c. If the root vertex expresses an event, this involves the semantic roles it projects and the referents filling these roles. The reason we include grandchildren is that event roles are specified on a separate vertex in the meaning representation, and we want to capture events and their participants. This particular design choice thus depends on the semantic formalism used, and has to be modified to accomodate different representational formats.

Next, we take the constituents of c and the head constituents of any other construction c' that refers to semantic dependents of sd_c, and linearly consider those to be the signifiers of a novel construction c_{syn}. The meaning of c_{syn}, viz. $sd_{c_{\text{syn}}}$, consists of the meaning of c, the root vertices of the meanings of all dependent c' and any vertices from the situation needed to make $sd_{c_{\text{syn}}}$ connected. c_{syn} is then added to the constructicon with a count of 0.

To give an example, let us assume a_5 was the best parse. In this parse, three partial derivations are concatenated with rule **i**. The third, however, is the ignoring of *there*, and hence is not considered. The set of concatenated constructions thus consists of the **mcc**s for c_4 and c_5. For the former, there

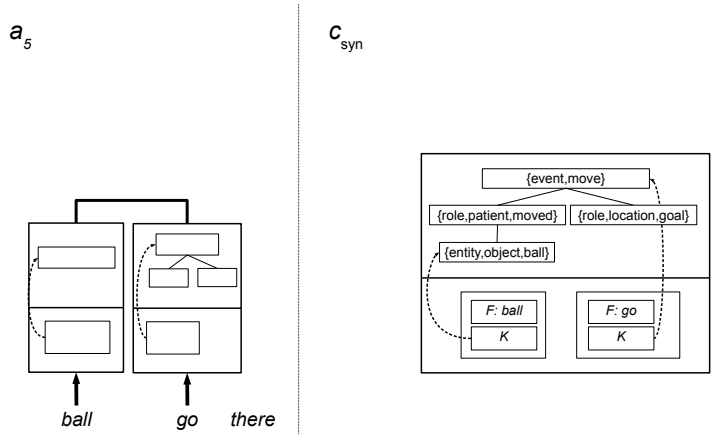


Figure 3.18: An example of the syntagmatization process applied to parse a_5.

are no other constructions in the set that express semantic dependents of it and hence no novel syntagmatizations can be made. c_5, on the other hand, expresses the {EVENT,MOVE} vertex of the situation, and c_4 expresses {ENTITY,OBJECT,BALL}, which is a grandchild of the {EVENT,MOVE} vertex. We therefore take all constituents of the **mcc** of c_5 and the head constituent of the **mcc** of c_4 and consider those to be a novel construction. The meaning of this novel construction consists of the meaning of the **mcc** of c_4 and the root vertex of the **mcc** of c_5 and any vertices needed to connect them (i.c., none). This novel construction is then added to the constructicon as c_6 with a count of 0. Figure 3.18 illustrates this process.

3.6.3 Paradigmatization

As we saw before, the model is able to parse utterances using a mixture of concrete and abstract constructions. How does it obtain these more abstract constructions? We can consider abstraction as the formation of paradigms of linguistic elements that can be substituted for each other, and hence call the process **paradigmatization**. In my implementation of the notion of abstraction, I again follow Langacker (2009), who argues that abstraction is not so much the creation of a novel hypothesis about the constructicon, but rather a by-product of processing several more concrete instantiations of a pattern. The overlap between these more concrete instantiations then becomes a potential to generalize. Whether one describes this in terms of abstract schemas or as a set of exemplars plus a rule to analogize over these, does not matter ac-

cording to Langacker (2009), as long as one is aware that these abstractions are not new cognitive 'entities' created from other 'entities' but rather a potential that is 'immanent' in these exemplars. In my implementation, however, the abstractions are separate entities. This should be seen as reflecting an implementational rather than an ontological issue, and as such it does not conflict with desideratum D4-3.

It is the idea of acquiring a grammar as a hypothesis testing procedure that underlies Bayesian Model Merging. What I propose, for abstraction, is to take the view seriously that there is no such thing as selection between levels of abstraction, i.e., that the organization of the abstraction in the constructicon is not governed by a selection mechanism deciding which level or clustering is the most appropriate one given the data and some prior conception on what the constructicon, or grammars in general, should look like (e.g., compact, or uniform). Rather, all possible abstractions over reinforced constructions (i.e., constructions with non-zero counts) are made, and the reinforcement of some of these abstractions, but not others (as discussed in section 3.6.1) leads to a constructicon that is highly general, but probabilistically constrained (i.e., utterances analyzed with both abstract and concrete constructions will have higher probabilities than utterances analyzed with only abstract constructions). This way, the abstraction in the SPL model differs from that of Chang (2008) and Beekhuizen, Zuidema & Bod (2013), who apply a Minimum Description Length criterion (Rissanen 1978) to the selection of abstractions, as well as Alishahi & Stevenson (2010), who cluster maximally concrete frames, thereby forcing the model to categorize an input item discretely with one or the other centroid cluster. Incorporating an element of 'selection' in one's model fits better with a deductionist view on language acquisition than an inductionist. For that reason, I think having a model that does allow for a gradual, bottom-up search through the hypothesis space, but without selection, is the preferable computational approach for a usage-based account of grammar acquisition (cf. desideratum D6-2).

Implementation

Whenever a construction c obtains its first reinforcement, it is compared to all constructions $c' \in \Gamma$ at that point in time that have also been reinforced (i.e., have a $count_{c'} > 0$). If from the overlap between c' and c a new construction c_{para} can be formed, and if c_{para} is not in Γ yet, c_{para} is added to the grammar.

The formation of an abstraction requires a comparison between c and c'. Not all comparisons lead to novel abstractions. Crucially, the model has to be able to find parallels between c and c' in both their signifiers and signifieds. This does not constitute a selection process, but rather reflects what comparisons the learner can and cannot make. A novel construction c_{para} can be formed from c and $c' \in \Gamma$ under the following conditions:

Conditions for creating an abstraction over two constructions

---

- Let $\mathbf{map}_{\text{overlap}}$ be a bijective structure-preserving mapping $f : sd_c \rightarrow sd_{c'}$ between the signified meanings sd_c and $sd_{c'}$ of two constructions c and c' such that
  - $\forall v_{v \in sd_c}.(\mathbf{map}_{\text{overlap}}(v) \cup v) \neq \emptyset$
- Let $\mathbf{M}_{\text{overlap}}(c, c')$ be the set of all possible $\mathbf{map}_{\text{overlap}}$ between c and c'.
- For each $\mathbf{map}_{\text{overlap}} \in \mathbf{M}_{\text{overlap}}(c, c')$, a novel construction c_{para} is created iff
  - $|sr_c| = |sr_{c'}|$
  - $\forall i_{i \in [1,\ldots,|sr_c|]}.\mathbf{map}_{\text{overlap}}(K(sr_c^i)) = K(sr_{c'}^i)$
  - $\neg((|sr_c| = 1) \wedge (F(sr_c^1) \neq F(sr_{c'}^1)))$
- where c_{para} consists of
  - sd_c o contains the intersection between all elements in $\mathbf{map}_{\text{overlap}}$ as well as their edge structure.
  - $sr_{c_{\text{para}}}$, where for each $i \in [1, \ldots, |sr_c|]$, $sr_{c_{\text{new}}}^i$ consists of
    - $F(sr_{c_{\text{new}}}^i = F(sr_c^i)$ if $F(sr_c^i) = F(sr_{c'}^i)$ else ϵ
    - $K(sr^i(c_{\text{new}})) = \mathbf{map}_{\text{overlap}}(K(sr_c^i)) \cup K(sr_c^i)$

The starting point for the abstraction over two constructions c and c' is an intersection mapping $\mathbf{map}_{\text{overlap}}$ between their meanings sd_c and $sd_{c'}$. For every possible intersection mapping between two constructions, we create a novel construction if all signifiers in both c and c' have conceptual constraints mapped to each other per $\mathbf{map}_{\text{overlap}}$.

Two further constraints are that the number of signifying constituents must be equal for both constructions, and that if one of the constructions is a lexical construction, the two constructions must have the same phonological constraint on that signifier. This last constraint is intended to obviate the possibility of having phonologically-unspecified single-constituent constructions. These constructions add little to the potential for analyzing an utterance, and even though they could be extracted, using them would always result in derivations of a lower probability than derivations mapping to the same part of the same situation without them. Chang (2008), however, does allow for them.

The constraints proposed above are motivated, but not cast in stone. One

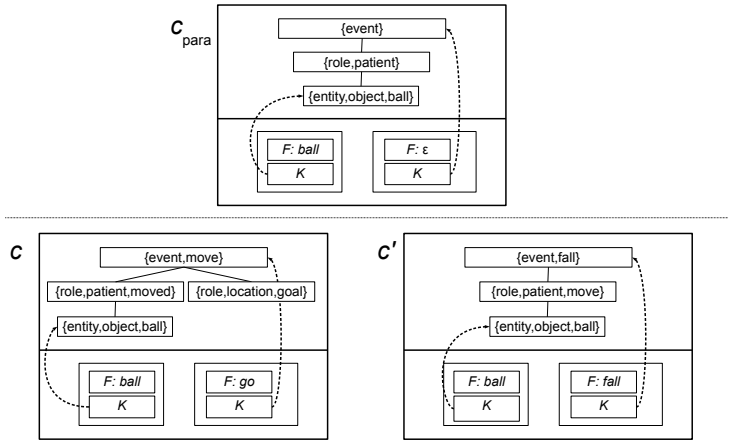


Figure 3.19: An example of succesful paradigmatization.

may drop the latter two constraints. The final constraint (identical phonological signifiers in the case of lexical constructions) only keeps the model from making spurious abstractions that, due to the probability model, will not be used.[5] The equal-number-of-signifiers constraint is more interesting. Loosening or dropping this constraint may result in different kinds of constructions being abstracted (in a similar way to Chang (2008)), but for the current purposes, this would needlessly complicate the model.

To give an example of the paradigmatization procedure, assume that a_5 was the best parse, and that the syntagmatized construction in figure 3.18 was added to the grammar. Through some subsequent input item, that construction becomes reinforced. Assume furthermore that another construction, represented as c' in figure 3.19, was present in the grammar as well. The meanings of the two constructions can be mapped with an overlap mapping, and the signifying constituents of both constructions point to each other via this mapping, so an abstraction can be made. This abstraction, c_{para} in figure 3.19, contains the intersection between c and c' as its meaning. The first constituent is phonologically specified and points to the {OBJECT,ENTITY,BALL} vertex in the meaning of c_{para}. The second, on the other hand, is not phonologically specified (as the phonological constraints on the second constituents of c and c' differ) and points to the root vertex of the meaning of c_{para}. With the paradigmatization operation, the model has now extracted a semi-open [[BALL / *ball*]

[5] An 'abstract' lexical construction is simply useless in creating derivations as for every use of an abstract lexical construction plus a concrete one, the competing analysis involving only the concrete lexical construction is more probable.

[EVENT]] construction, which is then added to the grammar with a count of 0.

Paradigmatization, I claim, is congenial with the view that abstractions are immanent in the more concrete patterns that instantiate them. Recall that Langacker argues that abstraction is essentially the co-activation pattern of several more concrete patterns. I believe the overlaps correspond to these co-activation patterns. Because SPL 'extracts' any and all of these patterns, the set of paradigmatized constructions can be seen as the potential for abstraction immanent in the more concrete ones. If a selection between paradigmatizations were made, on the basis of some criterion, the immanence would be, at least, harder to defend, as it would require a bridging hypothesis between the selection of certain paradigmatizations but not others. The view that abstractions are immanent, but discretely represented in a model is not new: we can find a similar idea in Skousen's (1989) Analogical Modeling, where all abstractions over a feature set are abstracted, and the model performs analogical reasoning over these.[6]

Note that paradigmatized constructions can be reinforced without the more concrete constructions instantiating them receiving further reinforcement. If an abstract construction is frequently used as the maximally-concrete used construction in many cases, it will receive much reinforcement, and hence be established as a unit, without the more concrete constructions achieving unit status. If an abstraction, however, is hardly used, for instance, because it generalizes over only two more concrete patterns that are themselves often used as **mcuc**s, the abstraction will stay rather weakly reinforce. As we will see in the following chapters, this dynamic leads to interesting insights in the development of constructional networks.

3.6.4 Cross-situational learning

When we assume a usage-based perspective on language acquisition, the model starts with an empty inventory of signs. Therefore, it should have learning operations at its disposal to get an initial inventory of constructions off the ground. A prime candidate for such learning operations is cross-situational learning.

Broadly speaking, cross-situational learning is the process whereby a learner observes multiple situations in which utterances are produced and extracts or reinforces recurring matching pairs of parts of the utterances and parts of the situations. An intuitive example would be the case in which the learner first hears the utterance *you grab the ball!* and sees a ball on the table and understands the intention that the caregiver want her to grab it, and next the utterance *oh, now the ball is on the floor!*, paired with a situation where the child just threw the ball off of the table. The phonological substring *the ball* is

[6] The difference being that the abstractions themselves can receive reinforcement and thus obtain a degree of representational autonomy, whereas this in not possible in lazy learners such as Analogical Modeling or Memory-Based Learning.

shared between the two utterance-situation pairs, as is the semantic element of an entity 'ball' being present in both understood communicative intentions.

The acquisition of form-meaning pairings via cross-situational learning can be interpreted in several ways. Many word learning models (Xu & Tenenbaum 2000, Frank, Goodman & Tenenbaum 2009, Fazly et al. 2010) assume a probabilistic model, where the connections between phonological strings and their referents get reinforced every time a pair of a word and a semantic element co-occur in an utterance-situation pairing. Recently, this view has been contested by researchers who argue that this places too much of a burden on the learner, as she has to maintain and update an $m \times n$ matrix for all m seen words and all n seen semantic elements (Stevens 2011). Instead, Stevens proposes, the learner forms hypotheses at random, and validates these in next rounds, either reinforcing them if the new utterance-situation pairing corroborates it, or discarding them if not.

There are things to be said for both views. The probabilistic view, as opponents of this view argue, creates behavior that is too gradient in nature. Learner's behavior seems more categorical than would be expected on the basis of a probabilistic view. On the other hand, one could argue that a probabilistic system interacts with more discrete decision-making systems which are addressed (possibly in different ways) in experiments and natural processing and production behavior. This could resolve the issue of apparent discreteness in behavior, but until it is worked out, it remains hand-waving. On the other hand, the creation of hypotheses at random seems like a strange starting point. It is not clear to me why a learner would form a hypothesis about a form-meaning pairing on the basis of no evidence.

The instantiation of cross-situational learning I assume here can be seen as taking a halfway position between the two. Because I do not think the metaphor 'language acquisition as hypothesis testing' is the right one (see section 3.6.3 as well), I consider these initial form-meaning pairings to be reflections of the processing of utterance-situation pairs, despite the learner not having any contentive linguistic knowledge yet. The cross-situationally extracted patterns are not random guesses, but reflect a simple form of analogical reasoning. On the other hand, I do not want to assume too much keeping track of every contingency between possible forms and possible meanings.

An exemplar is a structured representation of an experience. Importantly, it is *structured* by linguistic processing. That is: whatever linguistic structure is found in the input item (the U, S pairing) is stored alongside the U, S pair. For the current purposes, I assume that a linguistic exemplar is a pair of the selected situation $s \in S$, and the best parse a_{best}. Let us furthermore assume that the learner keeps track of the most recent n exemplars, where $n = [1, \infty]$.

Implementation

The form of cross-situational learning I assume extracts overlaps in form and meaning between a new exemplar and the most recent n exemplars. It only

extracts those overlaps about which it is sure, that is: only if a single maximal overlap in the ignored parts of the utterance strings and a single maximal shared subgraph between the analyses can be found, a novel construction containing exactly this overlap is extracted and added to the grammar with a count of 0. More formally:

Cross-situational learning

For every new exemplar s^t, a^t_{best} and every exemplar $s^{t-i}, a^{t-i}_{\text{best}}$ in the range $i = [1, \ldots, n]$:

- Let U_I be the yield of a parse a that is governed by rule **iii** (i.e., ignored words).
- Let G_I be the subgraph of s to which no root node of any construction used in a has a mapping, or:
  $G_I = \forall v_{v \in V_s}.\ \nexists c, \mathbf{map}_{c,\mathbf{map} \in a}.\mathbf{map}(sd_c) = v' \rightarrow v' \neq v$
- Extract a novel construction c_{xsl} iff:
  - $U^t_I \cup U^{t-i}_I \neq \emptyset$
  - $U^t_I \cup U^{t-i}_I$ is a contiguous substring of the yield of a^t_{best} (i.e., the original utterance U^t).
  - $U^t_I \cup U^{t-i}_I$ is a contiguous substring of the yield of a^{t-i}_{best} (i.e., the original utterance U^{t-i}).
  - Assuming the set $\mathbf{M}(G^t_I)$,
    - which consists of all possible structure-preserving bijective functions $\mathbf{map}_{\text{identical}}$ between a connected subgraph of G^t_I and a connected subgraph of G^{t-i}_I containing identical feature sets on the mapped vertices,

    there is exactly one most-encompassing mapping $\mathbf{map}_{\text{mem}} \in M(G^t_I)$ such that all other functions $\mathbf{map}'_{\text{identical}} \in M(G^t_I)$ specify a domain and a codomain that are subsets of the domain and codomain of $\mathbf{map}_{\text{mem}}$.
- where the new construction c_{xsl} consists of
  - $sd_{c_{\text{xsl}}}$, which is the subgraph of G^t_I being the domain of $\mathbf{map}_{\text{mem}}$
  - $sr_{c_{\text{xsl}}}$, being a single constituent sr^1,
    - where $K(sr^1_c) = v_{\text{root}}(sd_{c_{\text{new}}})$, and
    - $F(sr^1_c) = U^t_I \cup U^{t-i}_I$

Whenever a new exemplar s^t, a^t is added, it is compared to all exemplars in this 'memory buffer'. In this comparison, for s^t, a^t and some s^{t-i}, a^{t-i}, the part of the utterance that is ignored in a^t is compared to the part of the utterance that is ignored in a^{t-i}. If the maximal overlapping substring is not a contiguous substring of the full yield of a^t and of a^{t-i}, or if the maximal overlapping substring is empty, no new construction is extracted.

On the side of the meaning, a similar process takes place. The model compares the part of the situations s^t and of s^{t-i} that are not analyzed with a^t and a^{t-i} respectively (i.e., G_I^t and G_I^{t-i}). If, out of all functions mapping a subgraph of G_I^t to an identical subgraph of G_I^{t-i}, there is exactly one function $\mathbf{map}_{\text{mem}}$ such that all other mapping functions specify subgraphs of the two graphs in $\mathbf{map}_{\text{mem}}$, a construction can be extracted. Otherwise, no construction can be extracted.

This precludes the situation in which more than one most-emcompassing mapping can be made, i.e., the situation in which one mapping points to one part of the situation graph, and another mapping to a (at most partially) overlapping other part of the situation. In those cases the learner cannot be fully sure which analogy to make, and hence extracts no novel construction. Admittedly, this strict rule for extracting initial constructions is relatively brittle and simplistic, but given the complexity of the rest of the model I believe an overly constrained learning mechanism is preferable over an underconstrained one. Furthermore, the constraints all derive from more general principles of making analogies and reasoning with uncertainty (beit in a very simple form: if the learner encounters any uncertainty, it will do nothing).

Importantly, substrings of more than a single word in the adult representation can be extracted with the cross-situational learning procedure. These holistic chunks correspond to undersegmentation in Peters's (1983) sense.

To given an example of the cross-situational learning procedure, assume that a_6 was the best parse of the utterance, and that there is a previous exemplar consisting of a^{t-1}, s^{t-1}, depicted in figure 3.20 below. As the maximal overlap in unanalyzed parts of the situations consists of the {ROLE,LOCATION, GOAL} vertex, and the overlap in the utterance of *there*, a novel construction, linking these two can be extracted.

3.6.5 SPL as a usage-based learner

The mechanisms described in this section jointly embody many of the insights discussed in the previous chapter. Despite being operationalized as learning operations applying to the best analyses, they can be conceptualized as neural effects of the processing of the best analysis itself. That is to say: there is no decision-making process outside the processing of the usage events (D6-2). Novel constructions are learned without evaluating their value: if they are of use to the model in analyzing utterances, they will receive subsequent reinforcement, if not, they will remain one-off patterns with a zero count. The model embodies developmental continuity (D6-4), with all learning mecha-

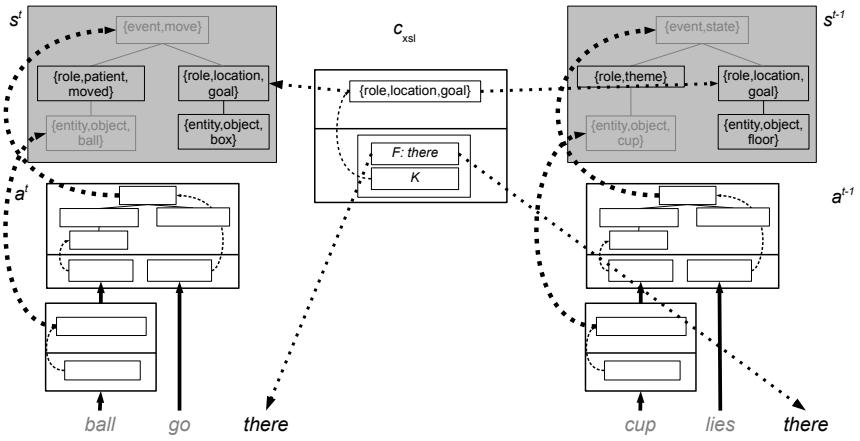


Figure 3.20: An example of succesful cross-situational learning. Analyzed parts of the utterances and the situations are marked in grey.

nisms being available throughout developmental time and simultaneity (D3) because both lexical and grammatical constructions can be learned with the same set of mechanisms. Although in practice, **cross-situational learning** operations will precede **syntagmatization** operations, which in turn precede the **paradigmatization** operations on them, all operations are used all the time.

Interestingly, the model has several ways of acquiring novel lexical constructions, viz. cross-situational learning, bootstrapping, and the use of maximally-concrete constructions. As bootstrapping will only take place after slightly more abstract constructions have been acquired, it typically takes place later in development than the cross-situational learning. These two ways of acquiring lexical constructions can be seen as reflecting Gleitman et al.'s (2005) ideas of the various ways in which lexical mappings can be learned, but from a usage-based perspective.

The two core mechanisms for the acquisition of grammatical constructions, syntagmatization and paradigmatization, further instantiate a few other desiderata. With syntagmatization, the acquisition of longer grammatical rules is qualitatively grounded in the usage-events: there is no preconception that longer rules will be part of the language, but the joint processing of several shorter rules leaves a trace in the mind of the learner (D5-1 and D6-2). At the same time, this means that longer constructions can only emerge if their parts are known, which satisfies D6-1, but having this as the only mechanism of the acquisition of rules conflicts with D6-3, the idea that we want a learner that does parts-to-whole and whole-to-parts learning. The latter is not

instantiated in this model, and, as I expressed before, I am skeptical about the necessity of such an operation and whether it fits in with ideas about learning-as-processing (D6-2).

The notion of abstraction engendered in the model is similar to Chang's, but differs in that it contains no post-hoc decision mechanism, thus being closer to the idea of learning-as-processing, to my mind. By extracting any and all abstractions, we can easily dereify the (reified) discrete representations as the potential for abstraction immanent in the most concrete constructions. Furthermore, the reinforcement mechanism only 'rewards' the most concrete *used* constructions, thereby boosting the potential of abstract representations (giving them more of a 'unit' status in Langacker's (2000) terminology) only if they are productively used.

3.7 Generation

An important property of SPL, as a generative model, is that it is bidirectional: we can analyze given utterances with it as well as generate new ones given a situation. Doing so, the model can simulate both processes of language comprehension and production (desideratum D2). Generation works largely by the same processes as analyzing an utterance, in that we generate a derivation that corresponds to the situation and take the phonological symbols at the leaf nodes of the derivation to be the utterance the model produces. Again, many analyses are possible and the model has to select the best one.

3.7.1 Differences with the analysis procedure

One aspect of the model differs from the comprehension procedure in the generation procedure. Several processing mechanisms defined in section 3.4 are geared towards processing input of which a part is not understood. In particular, the concatenation, ignoring and bootstrapping operations, as defined by rules **i**, **iii**, and **vi**, are operations allowing the model to interpret utterances despite having a limited inventory of linguistic signs. Generation works on the basis of known signs, and hence these three rules are not used. We assume that any derivation starts at rule **ii**, that is: with the application of a construction-mapping pairing. That is, the set of rules applicable in generation consists of rules **ii**, **iv**, **v**, and **vi**.

The probability of the derivation is again the product of the rules that are applied in it, as in equation (3.5). The probabilities of the $c, \mathbf{map}$ pairings are the same as for the comprehension procedure, and are repeated here as equations (3.12) and (3.13).

$$P(c, \mathbf{map}|CS) = \frac{count_c + 1}{\sum\limits_{c', \mathbf{map'} \in CS} (count_{c'} + 1) + 1} \tag{3.12}$$

	rule	probability
ii	START $\rightarrow (c, \mathbf{map})$	$P(c, \mathbf{map}\|CS_{\text{START}})$
iv	$(c, \mathbf{map}) \rightarrow sr_c^1, \ldots, sr_c^n$	1
v	$sr_c^i \rightarrow \alpha^+$ (if $F(sr_c^i) \neq \epsilon$)	1
vii	$sr_c^i \rightarrow (c', \mathbf{map}')$	$P(c', \mathbf{map}'\|CS_{sr_c^i})$

Table 3.4: Probabilities of the processing mechanisms in the generation procedure.

$$P(\mathbf{u}|CS) = \frac{1}{\sum\limits_{c,\mathbf{map}\in CS}(count_c + 1) + 1} \tag{3.13}$$

The probability of a derivation is, as in comprehension, given by:

$$P(d|\Gamma, S) = \prod_{r\in d} P(r) \tag{3.14}$$

3.7.2 Expressivity

In producing an utterance, a language user wants to be as expressive as possible (with as little effort as possible). I operationalize this idea as follows. Assume that the model creates analyses consisting of equivalent derivations, as in the comprehension procedure. The model penalizes an analysis for every feature of the situation that it does not express. It does so by taking the summed proportion of features in the situation s not expressed by the analysis a ($\mathbf{unexpressed}(\mathbf{a}, \mathbf{s})$) and raises the probability of an unseen event to the power of $\mathbf{unexpressed}(\mathbf{a}, \mathbf{s})$.

The calculation of $\mathbf{unexpressed}(\mathbf{a}, \mathbf{s})$ is defined as follows. For every vertex, the model checks which features are expressed by any construction in the derivation and takes the proportion of unexpressed features per vertex, after which the proportions are summed. Vertices that are completely unexpressed will thus contribute a proportion of 1 to $\mathbf{unexpressed}(\mathbf{a}, \mathbf{s})$, whereas partially expressed vertices add a value between 0 and 1 (exclusive) to the score. The referential penalty over a, or referential penalty(a) can thus be defined as:

$$\text{referential penalty}(a) = P(\mathbf{u}|CS_{\text{start}})^{\mathbf{unexpressed}(\mathbf{a},\mathbf{s})} \tag{3.15}$$

Obviously, the notion of referential expressivity employed here is a rather naïve one. The speaker, while interacting with the hearer, has a shared common ground with the hearer in the speech situation often allowing for the correct identification of the situation referred to with minimal means (see, e.g., Clark 1996). Furthermore, framing communicative success as the correct identification of a situation is rather simplistic as well. I leave it to future research to operationalize and implement more complex notions, involving for instance construal (Croft & Cruse 2004, ch. 3), argumentativity (Verhagen 2005), and the many other dimensions of communication.

3.7.3 Selecting the best analysis and utterance

The full probability of an analysis can now be given as follows:

$$P(a|s,\Gamma) = \sum_{d \in a} P(d|s,\Gamma) \cdot \text{referential penalty}(a) \tag{3.16}$$

Again, we can select the best analysis a_{best} given the situation, using the same definitions as in section 3.5.

Equally interesting is the selection of the best utterance. Multiple analyses may have the same yield (i.e., the same string of phonological structures) as the leaf nodes of the derivations. I take the probability of an utterance U given a situation s and a grammar Γ to be the disjoint probability of all analyses that have U as their yield.

$$P(U|s,\Gamma) = \sum_{a:\mathbf{yield}(a)=U} P(a|s,\Gamma) \tag{3.17}$$

3.7.4 An example of the generation procedure

Assume the grammar with the five constructions in figure 3.5, and the first situation (s_1) from figure 3.6. All of these constructions have subset mappings to the situation and hence all can be applied as the first rule. Constructions $c_1 - c_3$ are grammatical constructions and can therefore be combined with other constructions. Figure 3.21 gives all possible derivations given the situation and the grammar in the box-diagrammatic notation.

Calculating the derivational probabilities and the referential penalties, we get the probabilities of the six derivations in table 3.5. As we can glean from the table, derivations d_2, d_3 and d_4 are derivationally equivalent, while the other derivations are not equivalent with any other derivation. We thus arrive at the four analyses in table 3.6.

The penalty is calculated by looking at the summed proportion of the features on the vertices of the situation. For analysis a_1, the entire vertex {ENTITY, OBJECT,BOX} is left out. This means that the penalty is the probability of an

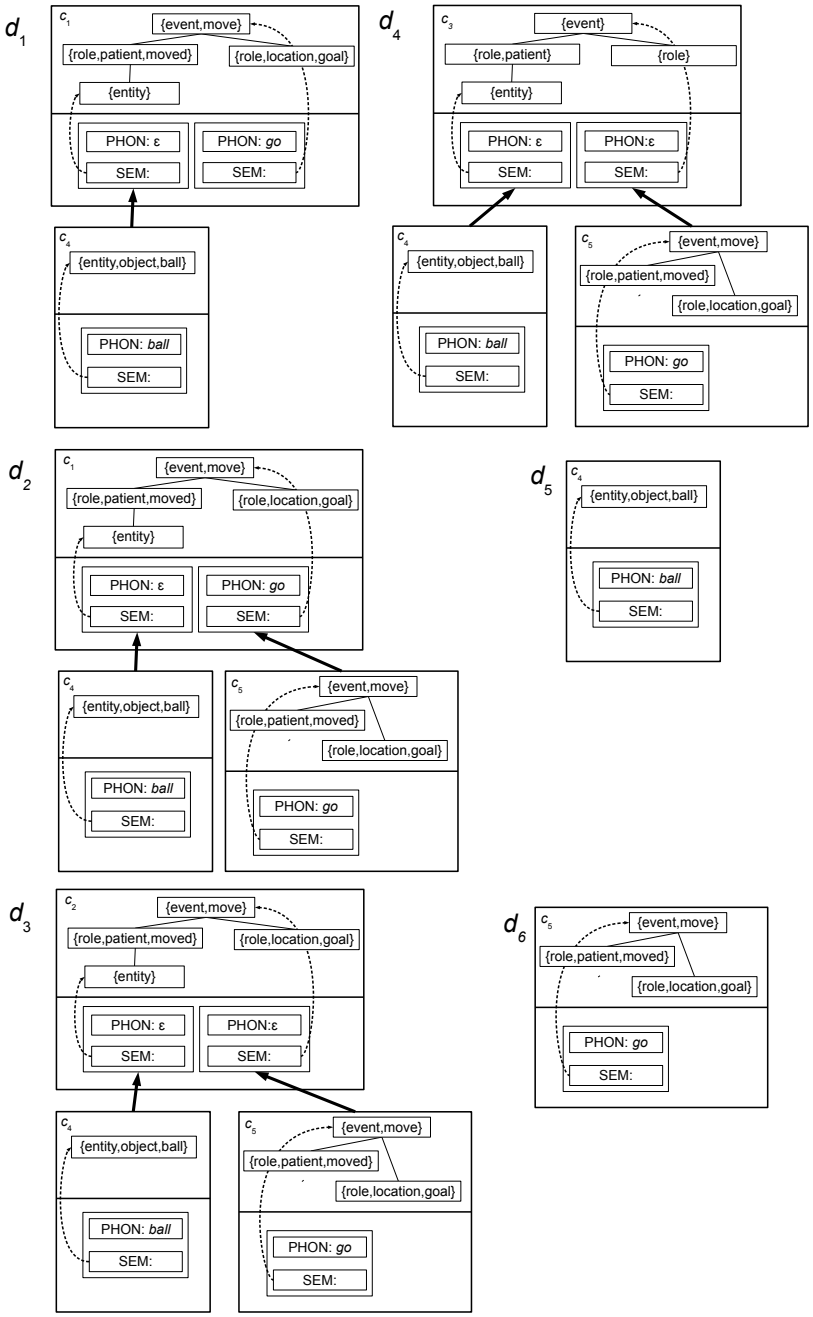


Figure 3.21: All derivations for the generation of utterances given s_1.

d	rules	yield	$\prod_{r \in d} P(r)$	$P(d\|s,\Gamma)$
d_1	**ii**, **iv**, **vi**, **iv**, **v**, **v**	*ball go*	$\frac{1}{13} \cdot 1 \cdot \frac{5}{6} \cdot 1 \cdot 1 \cdot 1$	$\frac{5}{78}$
d_2	**ii**, **iv**, **vi**, **iv**, **v**, **vi**, **iv**, **v**	*ball go*	$\frac{1}{13} \cdot 1 \cdot \frac{5}{6} \cdot 1 \cdot 1 \cdot \frac{2}{8}$	$\frac{10}{624}$
d_3	**ii**, **iv**, **vi**, **iv**, **v**, **vi**, **iv**, **v**	*ball go*	$\frac{2}{13} \cdot 1 \cdot \frac{5}{6} \cdot 1 \cdot 1 \cdot \frac{2}{8}$	$\frac{10}{624}$
d_4	**ii**, **iv**, **vi**, **iv**, **v**, **vi**, **iv**, **v**	*ball go*	$\frac{2}{13} \cdot 1 \cdot \frac{5}{6} \cdot 1 \cdot 1 \cdot \frac{2}{8}$	$\frac{10}{624}$
d_5	**ii**, **iv**, **v**	*ball*	$\frac{5}{13} \cdot 1 \cdot 1$	$\frac{5}{13}$
d_6	**ii**, **iv**, **v**	*go*	$\frac{2}{13} \cdot 1 \cdot 1$	$\frac{5}{13}$

Table 3.5: The probabilities of the six derivations in figure 3.21.

analysis	derivations	P derivations	$\sum_{d \in a}$	penalty	$P(a)$
a_1	d_1	$\frac{5}{78}$	$\frac{5}{78}$	$\frac{1}{13}^1$	$\frac{5}{1014} \approx 4.93e-3$
a_2	d_2, d_3, d_4	$\frac{10}{624} + \frac{10}{624} + \frac{10}{624}$	$\frac{30}{624}$	$\frac{1}{13}^1$	$\frac{30}{8112} \approx 4.93e-3$
a_3	d_5	$\frac{5}{13}$	$\frac{5}{13}$	$\frac{1}{13}^4$	$\frac{5}{371{,}293} \approx 1.35e-5$
a_4	d_6	$\frac{5}{13}$	$\frac{5}{13}$	$\frac{1}{13}^2$	$\frac{5}{2197} \approx 2.28e-3$

Table 3.6: The probabilities of the parses given the six derivation in figure 3.21.

unseen event to the power 1, or $\frac{1}{13}^{(1)}$. Analysis a_3, on the other hand, leaves out four full vertices, and thus has a penalty of $\frac{1}{13}^4$.

Two analyses are equally likely, viz. a_1 and a_2. If we look at the yields, we find that the utterance *ball go* is the most likely utterance given the situation and the grammar, with a probability of $P(U = \textit{ball go}|s,\Gamma) = 9.86e-3$. At about a quarter of that probability is the utterance *go*, supported only by analysis a_4 ($P(U = \textit{go}|s,\Gamma) = 2.28e-3$). The utterance *ball* is least likely to be generated by the model, with a probability of $1.35e-5$.

3.8 Meeting desiderata with SPL

SPL was developed with the theoretical discussion about the mechanisms necessary to account for language acquisition in mind. The close adherence to linguistic theorizing is therefore an aspect of this research that warrants its own section. In this section I evaluate whether SPL meets the various desiderata we set out in chapter 2. Throughout this chapter, I have discussed why the several aspects of SPL do so. Here, I briefly summarize them.

desideratum	evaluation
D1 (explicitness)	+
D2 (comprehensiveness)	+
D3 (simultaneity)	+
D4 (representational realism)	
D4-1 (qualitative grounding)	+
D4-2 (quantitative grounding)	+
D4-3 (immanence)	+
D5 (processing realism)	
D5-1 (heterogeneous structure building)	+
D5-2 (linear processing)	+
D6 (ontogenetic realism)	
D6-1 (cumulative complexity)	+
D6-2 (learning-by-processing)	+
D6-3 (parts-to-whole and whole-to-parts)	+/−
D6-3 (developmental continuity)	+
D7 (explanatory insight)	
D7-1 (unification)	+

Table 3.7: Evaluating SPL against the desiderata.

Concerning the explicitness of the model's simplifying assumptions (D1), I believe SPL to be relatively clear. Several aspects of the model were more at center stage than others, and, for instance, the implementation of cross-situational learning and the linear parser constitute highly simplified versions of obviously much richer cognitive processes.

Second, the model is (in principle) able to comprehend and produce utterances using the process of forming derivations and selecting the best one from among those (D2). It can, however, only be gradually expected to acquire this skill, as the model starts with an empty set of constructions. Over time, the model learns lexical and grammatical constructions, where the processes for the acquisition of both apply at the same time and are available to the model throughout developmental time (D3, D6-3).

The representations used by SPL, constructions, consist solely of conceptual and phonological structure, as well as a symbolic link, and can thus be said to be qualitatively grounded in the linguistic usage events (D4-1). By reinforcing the used constructions (more specifically, the maximally-concrete used construction), the model is sensitive to the frequencies of aspects of usage events (D4-2).

SPL processes utterances by using a derivation process, and selecting the most likely set of equivalent derivations from among all possibilities. Because such a global optimization process can be considered cognitively unrealistic, the actual analysis is done with a parser that goes over the utterance linearly and prunes all but the most likely analyses as it goes (D5-2). In building up analyses, the model has several processing mechanisms at its disposal: simple slot-filling, but also the creation of non-hierarchical analyses by means of concatenation, the top-down interpretation of a word by means of bootstrapping, and the possibility to ignore words, and as such the model has a robust toolkit of processing mechanism (D5-1).

Learning in SPL can be considered to be a by-product of processing (D6-2): the model processes an utterance, and the resulting best analysis as it is mapped to a situation leaves a trace by adding syntagmatized constructions and maximally-concrete constructions. Syntagmatization can be said to instantiate a cognitive take on Brown's law of cumulative complexity (D6-1) and part-to-whole learning: more complex constructions can only be learned on the basis of concatenations of simpler structures that leave traces in the mind of the speaker in the form of syntagmatized constructions.

Multiple constructions may share structure, in which case the model extracts a more abstract construction by means of paradigmatization. The paradigmatization operation involves no selection process whereby it is decided whether an abstraction is useful or not. As such any and all abstraction are extracted, and, as I argued, this makes this implementation of a notion of abstraction congruent with the idea of abstractions being immanent (D4-3).

Whole-to-parts learning was not the focus of this model, and, for instance, the model does not break down acquired unanalyzed chunks any further, and hence I evaluated D6-3 as $+/-$. There are some aspects of whole-to-parts

learning available to the model. One is the bootstrapping operator, whereby the meaning of an unknown part is assigned to it on the basis of a whole. This, however, does not constitute a case of the decomposition of a thitherto unanalyzed whole, and as such not all whole-to-parts learning operations conceivable are done by the model. Discussing this, I argued that the decomposition of chunks is perhaps an unlikely kind of cognitive operation from the perspective of D6-2: it requires the learner to engage in a post-hoc adjustment of the acquired chunks, which seems to be a kind of off-line reasoning for which more evidence would be needed. I leave it to proponents of the starting-big perspective to reconcile the decomposition of chunks with the learning-as-processing perspective, or to find evidence for off-line operations that break down chunks.

The issue of explanatory insight (D7) can of course only be discussed sensibly after we have seen some results. Before we turn to these, there is one aspect of the realism of the model that I would like to consider, viz. the nature of the input items, especially the situational contexts, given to the model. The next chapter deals with this issue.