A note on the representation and learning of quantificational determiners[1]

Roni KATZIR — *Tel Aviv University*
Nur LAN — *Tel Aviv University and ENS*
Noa PELED — *Tel Aviv University*

Abstract. There is a tight, bidirectional connection between the formalism that defines how linguistic knowledge is stored and how this knowledge can be learned. In one direction, the formalism can be mapped onto an evaluation metric that allows the child to compare competing hypotheses given the input data. In the other direction, an evaluation metric can help the linguist to compare competing hypotheses about the formalism in which linguistic knowledge is written. In this preliminary note we explore this bidirectional connection in the domain of quantificational determiners (e.g., 'every' and 'some'). We show how fixing an explicit format for representing the semantics of such elements – specifically, a variant of semantic automata – yields an evaluation metric, based on the principle of Minimum Description Length (MDL), that can serve as the basis for an unsupervised learner for such denotations. We then show how the MDL metric may provide a handle on the comparison of semantic automata with a competing representational format.

Keywords: quantificational determiners, learning, minimum description length, semantic automata.

1. Introduction

Early work in generative linguistics noted a tight, bidirectional connection between (a) the precise formalism that defines how linguistic knowledge is stored and (b) how this knowledge can be learned. The formalism can be mapped onto an evaluation metric – as in the simplicity metric of Chomsky and Halle 1968 (though other mappings exist, and below we will advocate a mapping that is somewhat different from that of early generative grammar) – that allows the child to compare competing hypotheses given the input data. This evaluation metric can then serve as part of a language acquisition device. And, in the opposite direction, a general evaluation metric can serve the linguist as a tool to compare competing hypotheses about the formalism in which linguistic knowledge is written (a point made in Halle 1978 and pursued further in works such as Baker 1979 and Dell 1981): two theories that are comparable in their ability to capture adult judgments might still make divergent predictions about learning when combined with a general evaluation metric. Early work on the bidirectional connection between representations and learning focused primarily on phonology (see especially Halle 1978 and Dell 1981), and to a lesser extent on syntax (see Baker 1979). But of course both the format for representations and the learning mechanism are important in semantics as well. In this paper we discuss the bidirectional connection between representation format and learning in semantics, focusing on the empirical domain of *quantificational determiners* (Q-dets): determiners of type $<et,<et,t>>$ such as 'every' and 'some'. Since our focus is on learning, we will consider only Q-dets that might need to be acquired and stored and set aside expressions that may serve as Q-dets but are syntactically complex and therefore do not need to be lexically stored.

[1]We wish to thank Asaf Bachrach, Adi Behar Medrano, Johan van Benthem, Emmanuel Chemla, Danny Fox, Michael Franke, Yuval Ishay, Fred Landman, Tim O'Donnell, Jonathan Palucci, Ezer Rasin, Raj Singh, Shane Steinert-Threlkeld, Jakub Szymanik, and the audiences at CNRS, MIT, and *SuB 24*.

Proceedings of Sinn und Bedeutung 24, vol. 1, pp.392–410. Osnabrück University.

We start, in section 2, by considering the mapping from representations to evaluation metrics, and ultimately to learners. We show how by fixing a representational format – for concreteness, a variant of *semantic automata* (SA; van Benthem 1986) – we obtain a learner from positive evidence alone. We do not wish to argue for SA (or for any other particular formalism for that matter) in this paper. Rather, our goal is to present and motivate the learning approach. Specifically, the evaluation metric will be that of *Minimum Description Length* (MDL; Rissanen 1978), which balances two competing factors: (a) the complexity of the grammar; and (b) its fit to the data. (By doing so, MDL combines the perspectives of two other evaluation metrics that have been used within the generative tradition: the simplicity metric of Chomsky and Halle 1968, which minimizes the complexity of the grammar, and the subset principle advocated in later work such as Dell 1981 and Berwick 1985, which maximizes the fit of the grammar to the data. See Rasin and Katzir 2016 for discussion.) The resulting learner – very much in line with Piantadosi et al. (2012), who use MDL for an unsupervised learner for a different representation – will be an unsupervised learner for the variant of SA that we use.[2]

In section 3 we then discuss the opposite direction, going from the MDL metric back to representations. Here we will attempt to compare SA to a different approach, which we will refer to as *building blocks* (BB). While the two approaches are hard to tease apart as the format for representing Q-dets, they are clearly different from one another, and finding a way to choose between them empirically can be significant. We will show how MDL might help in this task by outlining two kinds of cases in which the relative MDL scores assigned to certain Q-dets are different under SA and under BB, a difference that may translate into divergent learning-based predictions. Making the actual choice will be difficult, however, and in the present, preliminary (and highly programmatic) work we will have to content ourselves with a sketch of how a future comparison might be made.

2. From representations to learning

The present section shows how having an explicit format for storing knowledge provides a way to acquire such knowledge. As mentioned, we will illustrate this general point using lexical Q-dets (that is, Q-dets that need to be acquired and stored, rather than constructed compositionally): in all our examples, the learner will see a given scenario – for example, one where some boxes are on the shelf and others are not – and hear a sentence such as 'gleeb boxes are on the shelf', where 'gleeb' is the target Q-det and can be assumed to be lexical

[2]See Clark 1996 for an earlier discussion of learning in the context of SA. Differently from our learner, Clark's proposal relies on instruction (through the notion of a *minimally adequate teacher* from Angluin 1987), which includes negative evidence. See Steinert-Threlkeld and Szymanik 2019 for a different framework for representing and learning Q-dets, based on artificial neural networks (and which the authors suggest might be similar to SAs in certain ways). Like Clark (1996)'s learner and differently from ours, Steinert-Threlkeld and Szymanik's learner relies on negative evidence.

Like Piantadosi et al. (2012) and Steinert-Threlkeld and Szymanik (2019), the present paper presents an implemented learner that can be run on various input data. There is also a body of work on theoretical learnability results in various paradigms. In addition to Clark 1996, work of this kind includes Tiede 1999 and Paperno 2011, both of whom discuss classes of Q-dets that are identifiable in the limit in the sense of Gold 1967. Paperno 2011 also considers learnability within the framework of PAC-learning (Valiant 1984) and reaches mostly negative conclusions about learnability within this paradigm, and Magri 2015 uses PAC-learning to motivate certain restrictions on possible Q-dets. See also Schafer 2019 for a PAC-learning analysis of the learning approach of Piantadosi et al. 2016, which, while not directly about Q-dets, is very close to this domain.

and in need of being acquired and stored.[3] (The denotations of 'box' and 'shelf' are taken to be known.) Throughout, we will assume that the input available to the child consists entirely of positive examples, with no corrections or other forms of instruction. This is a conservative assumption, which makes the learning task hard, and we note that the input to actual children might be richer and possibly involve additional information, including certain forms of negative evidence. In particular, Rasin and Aravind (2020) have examined the input to the child in the context of the acquisition of the quantifier 'every' and found that, while the input included no systematic corrections or other forms of semantic negative evidence that could directly inform the child that 'every' denotes a universal and not the more inclusive existential quantifier, it did include pragmatic negative evidence that the child could potentially use.

Given familiar observations in the literature, we will focus only on Q-dets Q for which the following two conditions hold.[4]

(1) Assumptions about target Q-dets for the purposes of this paper:
   a. $Q(A)(B)$ can be determined based on $|A \cap B|$ and $|A \setminus B|$
   b. Q is first-order

We make the assumptions in (1) for convenience. While these assumptions do correspond to various generalizations about attested lexical Q-dets across languages, it is far from clear that they adequately characterize the kinds of Q-dets that children *can* acquire: as far as we can tell there is not enough evidence currently available to evaluate such a claim.[5]

In order to appreciate the significance to learning of using a concrete, explicit format for lexical storage, we start, in section 2.1, by probing intuitions about which denotations should be ac-

[3]We state the discussion simplistically in terms of lexical storage of Q-dets. We hope that the discussion can be restated within a proper morphological theory but will not attempt to do so within the present paper.

[4]See van Benthem 1986 for discussion of (1a). This restriction is often presented in terms of three familiar generalizations about monomorphemic Q-dets: (a) that they (and also complex Q-dets) are *conservative* (that is, that $Q(A)(B) = 1$ exactly when $Q(A)(A \cap B) = 1$; see Barwise and Cooper 1981 and Keenan and Stavi 1986); (b) that they satisfy *extension* (that is, adding individuals to the domain beyond those already in A and B makes no difference to $Q(A)(B)$); and (c) that they are *isomorphism invariant* (if we map the domain isomorphically to another domain, $Q(A)(B)$ does not change). Assumption (1b) might seem less obviously justified typologically given the existence of the second-order Q-det 'most'. See Hackl 2009 and Gajewski 2010, however, for arguments that 'most' is morpho-syntactically complex, composed of 'many' and the superlative morpheme '-est' (a decomposition going back to Bresnan 1973).

[5]Hunter and Lidz 2013 present an experiment in which children acquired a novel conservative Q-det but not a non-conservative one (which is in line with (1a)), but see Spenader and de Villiers 2019 for experimental results (both with children and with adults) that do not show a learning preference for conservative Q-dets. Either way, for both assumptions in (1) further work is required in order to determine whether children can acquire generalization-violating lexical Q-dets. If they can, then the assumptions are too restrictive to adequately characterize the learner's space of possible denotations.

It is also possible that the assumptions in (1) are *not restrictive enough*. In particular, while 'many' and numerals such as 'two' have quantificational uses, it has been argued that they are fundamentally adjectives and achieve their quantificational force compositionally, via type-shifting operations or composition with silent operators. See Landman 2004 for detailed discussion and defense of this position, as well as relevant references. If true, this would be compatible with a view on which the space of possible Q-dets is considerably smaller than the one assumed here – possibly to the point where the learning problem for Q-dets becomes trivial.

Given the above, the characterization of the space of lexical Q-dets in (1) should be taken as highly tentative, assumed as a concrete starting point but in need of much further work to clarify what Q-dets can actually be represented and acquired by children.

quired for a novel Q-det given a set of scenarios in the absence of such a format. We note that intuitions are sometimes reasonably clear about when a given hypothesis is better or worse than a competing hypothesis given a set of data, based on considerations of simplicity. However, using these considerations within an actual evaluation metric requires being explicit about the format of representations. This we do in section 2.2, where we fix as our representational format a variant of SA. (As mentioned above, it is not our goal to argue for SA. Rather, we wish to present a particular approach to learning and to show how any reasonable representational format can serve as the basis for such learning.) With an explicit format for representations fixed, we return, in section 2.3, to the data and hypotheses from our initial discussion and see how to talk about two notions of simplicity – simplicity of the grammar and simplicity of the encoding of the data given the grammar – that jointly contribute to the appropriateness of a grammar given the data. One way to combine the two notions of simplicity is through the MDL evaluation metric, which we discuss in section 2.4. Finally, section 2.5 presents our MDL learner.

2.1. Choosing between hypotheses

Suppose we hear the sentence 'gleeb boxes are on the shelf', with a novel monomorphemic Q-det 'gleeb', in the context of various scenarios in which there are various boxes whose location (on the shelf or not on the shelf) can directly be determined. In order to encode such scenarios, let us assume that we have agreed on a way to enumerate the boxes in each case and that we use 1 to mark that a given box is on the shelf, 0 to mark that it is not, and # to mark that we have reached the end of the encoding of the current scene. For example, $<1,0,1,\#>$ will encode a scenario with exactly three boxes, where the first and third ones are on the shelf and the second one is not. For our present illustration, suppose that the scenarios under consideration are the following:

(2) Different scenarios exemplifying 'gleeb boxes are on the shelf'
   a. $<1,1,1,1,1,\#>$
   b. $<1,0,1,1,0,0,\#>$
   c. $<0,1,1,1,1,1,1,\#>$
   d. $<1,0,0,1,1,1,1,0,1,1,1,0,0,\#>$
   e. $<0,1,0,0,0,1,0,1,1,1,1,\#>$

What does 'gleeb' mean? Assuming a rich enough space of potential denotations such as the one defined by (1), there will always be infinitely many hypotheses that are compatible with any finite input data – here, with the scenarios in (2). However, some hypotheses will be better than others. Let us look at a few examples.

First, 'gleeb' might mean 'any number of' (zero or more). Informally speaking, this seems like a simple, natural kind of hypothesis. If we were to care only about the complexity of the hypothesis (as was done under the simplicity metric of early generative grammar), this hypothesis might be chosen. On the other hand, 'any number of' does not fit the observed data very tightly: it is overly inclusive and would be true of any scene, while our (small) corpus in (2) does not seem entirely random. For example, in each of the examples there are always at least some boxes on the shelf, a fact that becomes an accident under the hypothesis that 'gleeb' means 'any number of'.

Consider next the following hypothesis: 'gleeb' might mean 'exactly 3 or 5 or 6 or 8'. This is a considerably more restrictive hypothesis than 'any number of', and it fits the data very well. If we were to care only about fitting the data (as was done under the subset principle), we might choose this hypothesis. On the other hand, 'exactly 3 or 5 or 6 or 8' feels like a very complex, unnatural hypothesis. If we were to see many further examples like those in (2), we might eventually want to adopt this restrictive but unnatural hypothesis. But given the small input sequence above, this seems unwarranted.

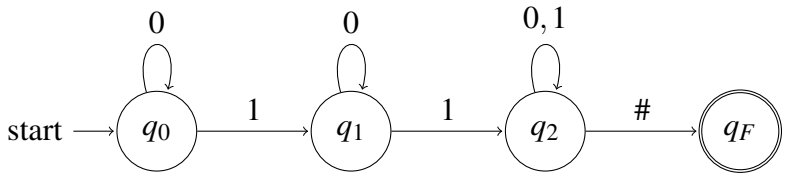
Finally, here is a better hypothesis than either of the above: 'gleeb' might mean 'at least 3'. The hypothesis is still quite simple and natural, and yet it also fits the data quite well. It is a bit of a compromise on each front: it is somewhat less natural and simple than 'any number of', and it fits the data less tightly than 'exactly 3 or 5 or 6 or 8'. But while less simple than 'any number of' it still seems reasonably simple, and while less restrictive than 'exactly 3 or 5 or 6 or 8' it is still reasonably restrictive, and of the three hypotheses under consideration it seems the most suitable overall given the data in (2).

The brief and informal discussion above suggests that balancing simplicity of hypothesis against restrictiveness (or goodness of fit to the data) might match our intuitions about the evaluation of hypotheses. It would be reasonable to consider it as a guiding principle for learning. But how do we measure simplicity and restrictiveness? To do that we will need to be more explicit about our representations than we have been so far.

2.2. Representing Q-dets using semantic automata

Here is one way to be explicit about representations, due to van Benthem (1986) and discussed further by Clark (1996), Steinert-Threlkeld and Icard (2013), and Szymanik (2016) among others. Recall from (1a) that we assume that for any lexical Q-det Q, the value of $Q(A)(B)$ can be determined based on $|A \cap B|$ and $|A \setminus B|$. For example, 'some' checks that the number of A's that are B's (represented as 1's in the input sequence, as discussed above) is at least 1. And 'every' checks that the number of A's that are not B's (represented as 0's in the input sequence) is 0. So we can represent monomorphemic Q-dets with a counting device that checks the cardinalities of the two sets. One kind of counting device that works in many cases – and that will suffice given our assumption in (1b) – is a finite-state automaton like the following:[6]

(3) An automaton for 'at least 2'



The automaton in (3) has four *states*, marked with circles and labeled q_0, q_1, q_2, and q_F. One

[6]See Hopcroft and Ullman 1979 and Sipser 2012 for introductions to the theory of automata. With van Benthem (1986) we take SA to be deterministic. However, as described below, we allow transitions that lead to an implicit sink state not to be encoded and not to count toward the costs associated with an automaton. We will omit the sink state and its transitions from our diagrams as well.

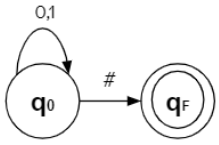
SA respect (1a) and therefore do not discriminate between input sequences based on the order of the 0's and 1's. This means that not every (deterministic) finite-state automaton is an SA. For example, an automaton that accepts the sequence $< 1, 0, \# >$ but rejects $< 0, 1, \# >$ is not a valid SA.

of these states, q_0, is a *start state*, and another, q_F (marked with an extra circle), is an *accept state*. The automaton also has *transitions* between states: a 0 edge tells us where to go when we have a 0 in the input sequence (corresponding to an *A* that is not a *B*), a 1 edge tells us where to go when we see a 1 in the input sequence (an *A* that is a *B*), and a # edge tells us where to go at the end of an input sequence.[7] When we are at a given state and see an input symbol for which there is no written edge – for example, if we are at q_1 and observe # as the next input symbol – this should be thought of as an implicit transition to a sink state, not illustrated in the diagram, from which we never recover. The automaton accepts an input sequence if it can parse that sequence starting from the initial state, following the transitions according to the symbols in the input sequence, and ending at an accept state. In the present case, for example, we accept an input sequence consisting of a box that is not on the shelf, two boxes on the shelf, another box not on the shelf, and end of sequence. More generally, the automaton in (3) accepts any sequence with at least two boxes on the shelf.

2.3. Repeating hypothesis evaluation using SA

Let us repeat the hypothesis evaluation from section 2.1 but with the explicit representational framework of SA. As we will see, having such a framework will allow us to make concrete the notions of simplicity and restrictiveness, which we relied on informally above. First, consider again the hypothesis that 'gleeb' for (2) means 'any number of'. The SA corresponding to this hypothesis is the following:

(4) SA for 'any number of'



We can now see what 'very simple' means: the automaton is very small. If we write this hypothesis as a computer program, using a programming language for SA, we will need very little storage space. In such a language, we will write each hypothesis as an encoding of the states and their transitions in a way that allows a reader to recover the original SA from the description.[8] The length of this encoding, written as $|G|$ and measured in *bits*, grows with the number of states and transitions. In (4), given how few states and transitions the SA has, it will be very short and cost only very few bits. So the grammar, *G*, is small.

What about restrictiveness? Here we should check how well *G* describes the input data *D*. Such a description is a sequence of instructions to *G* that generate *D*. These instructions, like the encoding of *G*, are provided in bits, and they depend on the optional choices in *G*. The unique transition from a unary-branching state in the SA is cost-free: no instruction is needed to tell the SA to move along a given transition from a given state if this transition is the only one leaving that state.[9] A binary transition costs one bit, specifying which of the two transitions is

[7]The # symbol is not part of the alphabet in van Benthem 1986.

[8]The sink state and the transitions leading to it are recoverable from the rest of the SA and therefore do not need to be specified explicitly and do not contribute to the costs of encoding *G*.

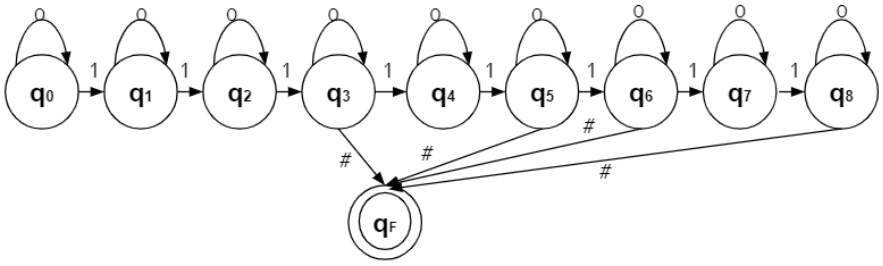
[9]Implicit transitions to a sink state are ignored for the purpose of determining optionality and therefore do not contribute to the costs of describing *D* given *G*.

to be chosen. (We assume that there is a key for making such choices. For example, perhaps the key specifies that if there is one 1-transition and one #-transition, then 0 encodes the former and 1 the latter.) If there are three transitions, additional bits are needed, and we will assume that in such cases two bits encode each choice. (Perhaps the key specifies that for ternary branching states, 00 encodes the 0-transition, 01 the 1-transition, and 10 the #-transition.) The sequence of instructions for a given input sequence – starting from the initial state, progressing through the relevant states along the way while generating the symbols in the input sequence in order, and ending in a final state – is the concatenation of the instructions at each state. In the case of (2a) ($=< 1,1,1,1,1,\# >$), for example, this will amount to five repetitions of the two-bit code for a 1-transition from q_0 followed by the two-bit code for a #-transition from q_0. For multiple input sequences, as in the whole of (2), the codes for the individual scenes are concatenated. We write $D : G$ for the encoding of the input data D given the grammar G, and we write $|D : G|$ for its length in bits.

In the case of (4), $|D : G|$ is quite high, since all symbols are produced through choices from the ternary-branching q_0 and therefore cost two bits each. This is the consequence of G in (4) being overly inclusive, capable of capturing any input scene and therefore not telling the story of the particular D in (2) very well. We may expect that a more specialized SA, which is more selective in the sequences that it accepts, will allow at least some symbols in its accepted sequences to result from unary- or binary-branching transitions and therefore cost fewer bits. In this way, restrictiveness becomes a kind of simplicity: not of the grammar but of the description of the data given the grammar.

Next, consider again 'gleeb' as '3, 5, 6, or 8', but now with the following representation:

(5) SA for 'exactly 3, 5, 6, or 8'

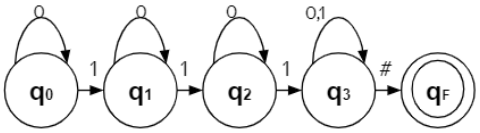


Using (5), $D : G$ is much smaller than with (4). To encode (2a) ($=< 1,1,1,1,1,\# >$), for example, we just need 1 bit for each of the first three boxes (since the branching at the relevant states–the first three ones–is all binary), and then 2 bits for the fourth box (since the branching at q_3 is ternary), and then 1 bit for the fifth box (binary branching), and then 2 bits for the termination of the sequence (ternary branching). So for four of the boxes we would pay just 1 bit instead of 2. On the other hand, $|G|$ is big, as can be seen by considering the encoding of the automaton, which would need to specify a relatively large number of states and transitions.[10]

Finally, 'gleeb' as 'at least 3' offers a good compromise, balancing between $|G|$ and $|D : G|$:

[10](5) illustrates a potentially worrisome property of SA with respect to $D : G$: differently from acceptance, which SA guarantee to be invariant to the order in which individuals are enumerated, $|D : G|$ does in general depend on the order of individuals. For example, the sequence $< 0,1,1,1,\# >$ will cost 6 bits using (5), while $< 1,1,1,0,\# >$ will cost 7 bits using the same SA.

(6) SA for 'at least 3'



2.4. Minimum Description Length

The balancing of $|G|$ and $|D:G|$ that we just discussed is at the heart of the principle of Minimum Description Length (MDL; Rissanen 1978, with roots in Solomonoff 1964), where we balance the two quantities by minimizing their sum, $|G|+|D:G|$. The approach, which is also very closely related to Bayesian induction, has been used for various learning and prediction tasks, including for natural language, in the works of Horning (1969), Berwick (1982), Stolcke (1994), Brent and Cartwright (1996), Grünwald (1996), and de Marcken (1996), among others. It formalizes the intuition we discussed regarding good hypotheses for Q-dets, and in section 2.5 we will show what needs to be added in order to turn it into an implemented unsupervised learner. Before that, however, we wish to briefly review several motivations for MDL as a reasonable null hypothesis for how we learn.

One reason to consider MDL seriously as the child's learning criterion is that it comes almost for free, as discussed in Katzir 2014. Grammars are real cognitive objects. They need to be stored in memory, storage that follows the specifications provided by our innate programming language, and this storage takes up space. The amount of space taken up by the grammar is $|G|$. Moreover, we use G to parse inputs, and an encoding of this parse – an understanding of the input data D according to G – is $D:G$. If we store this information, the amount of storage for this part is $|D:G|$. The MDL quantity $|G|+|D:G|$, then, is simply the overall amount of storage for the grammar and for the data as understood by the grammar. This, in turn, makes the MDL quantity available to the learner with what seems like very minimal stipulation. In order to use MDL to learn, all that is needed is the ability to compare this quantity for a current hypothesis and for a neighboring one and gradually move toward grammars that minimize the overall storage space, as we will do in the next section. We are not aware of competing proposals for learning that require fewer stipulations.

A second reason to consider MDL seriously is that it appears to work well in practice, supporting unsupervised learners for various linguistic frameworks. Rasin and Katzir (2016), for example, use MDL to learn whole phonological grammars within Optimality Theory (Prince and Smolensky 1993), including underlying representations, markedness and faithfulness constraints, and the ranking of those constraints. Rasin et al. (2018, 2019) provide a similar MDL learner for phonological grammars within rule-based phonology. In semantics, Piantadosi et al. (2012) use MDL for an unsupervised learner in a representational framework that is similar to one that we will consider below.[11]

[11]One way to think about the first two reasons above – an admittedly very speculative perspective, but one that combines the two considerations in a natural way – is in terms of evolution. If MDL is indeed as non-stipulative as suggested, the burden that it imposes on evolution is minimal. Evolution would need to provide a representational format that allows grammars to be stored and to be used for parsing inputs, but this much is presumably shared by most theories. Beyond that, only very little additional machinery would need to evolve to support MDL learning. Moreover, since MDL is such a general metric, it could have evolved at a stage in which the format for representing knowledge was different – perhaps much simpler – than in modern humans. Note that the same cannot be said for

A third reason for considering MDL seriously is that its balancing of $|G|$ and $|D:G|$ seems to be in accord with the behavior of human participants in lab experiments. In particular, MDL – like the closely related Bayesian approach, which similarly balances the naturalness of hypotheses and their fit to the data – matches empirical results about generalization in lexical acquisition (Xu and Tenenbaum 2007), causation (Sobel et al. 2004), visual chunk detection (Orbán et al. 2008), and elsewhere.

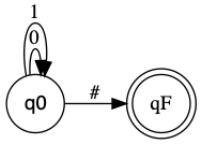
2.5. An MDL learner

We now turn to an implemented unsupervised learner for Q-dets based on the MDL evaluation metric. All that is needed in order to turn the metric into a learner is the ability to read the input data D and search the space of possible Q-dets for the grammar G that minimizes the sum of $|G|$ and $|D:G|$. In general we cannot do this by brute force – the search space is too big. However, there are good, general optimization procedures that can handle complex search spaces. The learner that we discuss here uses *simulated annealing* (Kirkpatrick et al. 1983). We start from an initial grammar – the one for 'any number of' – and proceed by comparing the current grammar G to a neighboring grammar G' (derived from G by certain simple operations) at each step.[12] If G' is better than G in terms of description length, we switch to G'. If it is not, we might still switch to G', depending on a random draw and on how much worse G' is and how far along in the search we are – the worse G' is and the further along we are, the less likely we are to switch.[13]

Here are some snapshots from a simulation in which D consists of 100 sequences conforming to 'between 3 and 6'.

(7) Snapshots from a sample run for 'between 3 and 6'

a. Initial hypothesis



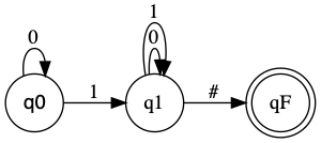
$|G_0| = 18, |D:G_0| = 3{,}954, |G_0| + |D:G_0| = 3{,}972$

b. Step 8

alternative learning approaches that involve various detailed procedural instructions and are therefore stipulative or that are specific to particular representational formats (e.g., constraint re-ranking in Optimality Theory). The second consideration, of MDL working well, might explain why, once MDL had evolved it would be conserved.

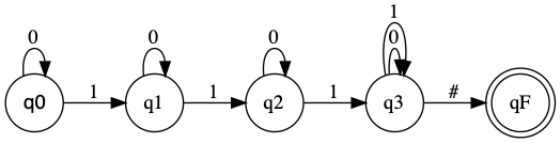
[12]To simplify the search, we only consider SA in which transitions are of the following kinds: (a) loops from some state q_i to itself; (b) transitions from q_i to q_{i+1}; and transitions from q_i to q_F. Transitions of the first two kinds are always labeled with either 0 or 1, and those of the third kind are always labeled with #. This choice rules out certain potential Q-det denotations but as far as we can tell does not affect the general discussion.

[13]The probability of switching to a worse G' is based on a temperature parameter, which decreases as the search progresses. In the examples mentioned below, the initial temperature is 100, the cooling factor by which the temperature is multiplied after each step is 0.99, and the threshold temperature for stopping the search is 1. For each simulation 64 annealing processes were run, and the result with the lowest MDL score of all runs was taken as the final SA. The code for the learner is available at https://github.com/taucompling/semantic$_automatahttps://github.com/taucompling/semantic_automata$.



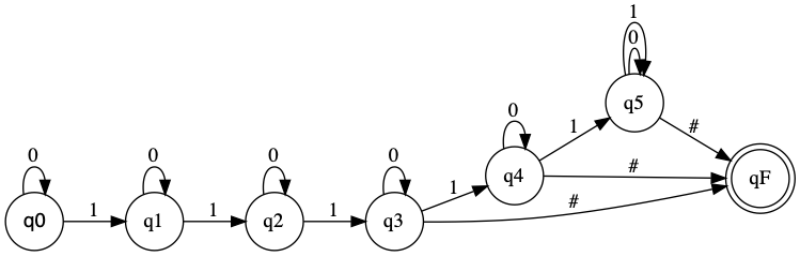
$|G_8| = 31, |D : G_8| = 3{,}543, |G_8| + |D : G_8| = 3{,}574$

c. Step 62



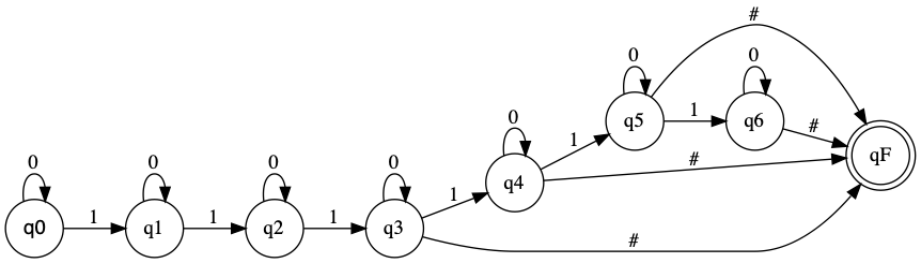
$|G_{62}| = 57, |D : G_{62}| = 2{,}826, |G_{62}| + |D : G_{62}| = 2{,}883$

d. Step 75



$|G_{75}| = 93, |D : G_{75}| = 2{,}826, |G_{75}| + |D : G_{75}| = 2{,}919$

e. Step 86



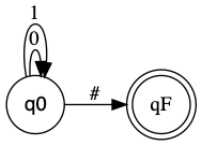
$|G_{86}| = 106, |D : G_{86}| = 2{,}757, |G_{86}| + |D : G_{86}| = 2{,}863$

The initial hypothesis, as mentioned above and shown in (7a), is the one for 'any number of', which is simple, overly inclusive, and very different from the target grammar. Gradually, hypotheses start moving toward the target grammar. G_8, shown in (7b) and representing 'at least 1', is already a small improvement: while it has more states and transitions than G_0 (so $|G_8| > |G_0|$), $|D : G_8| < |D : G_0|$ because the branching at q_0 of G_8 is only binary and not ternary. This results in a shorter encoding length for the first 1 and every 0 before it, each of which will now cost just one bit rather than two. Iteration 62, shown in (7c), is a further improvement, with G_{62} representing 'at least 3'. For a small increase in $|G|$, $|D : G|$ decreases significantly: the branching in the first three states is now binary, which allows much more of each input sequence that conforms to the Q-det to be encoded using one bit rather than two per input element. G_{75}, shown in (7d), also corresponds to 'at least 3', though with a worse MDL score, illustrating how simulated annealing can sometimes move from better hypotheses to worse ones. This particular sub-optimal hypothesis is not maintained for long, however, and at iteration 86, shown in (7e), the search has already reached the correct automaton.

Here is a similar run with 'none', again with D consisting of 100 sequences conforming to the Q-det (which, in the present case, means sequences of zeros followed by #).

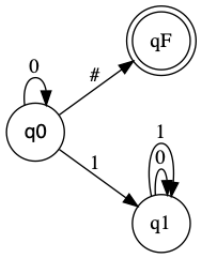
(8) Snapshots from a sample run for 'none'

a. Initial hypothesis



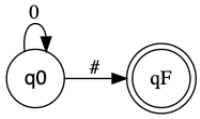
$|G_0| = 18, |D : G_0| = 6,928, |G_0| + |D : G_0| = 6,946$

b. Step 3



$|G_3| = 31, |D : G_3| = 6,928, |G_3| + |D : G_3| = 6,959$

c. Step 6 (Final hypothesis)



$|G_6| = 13, |D : G_6| = 3,464, |G_6| + |D : G_6| = 3,477$

This concludes our sketch of an unsupervised MDL learner for SA. We showed how fixing an explicit format for stored representations immediately yields an evaluation metric, based on the principle of MDL, that can serve as a central component of an unsupervised learner. Our goal was not to argue for the specific format that we used here (or any other particular format) but rather to highlight the mapping from explicit formats to an MDL learning criterion, a criterion that we suggested has certain appealing properties and that we believe makes sense as a starting point for modeling learning in humans.

3. From learning to representations

In the previous section we saw how fixing an explicit format for stored representations yields an evaluation metric, using the principle of MDL, that can then be used for an unsupervised learner. The present section shows the opposite direction: how, with a general criterion such as MDL, one can reason about formats for stored representations.

3.1. The idea

An observation due to Halle (1978) is that with a general approach to learning we can take two competing theories of representation and compare them using their learning-based predictions. This idea was explored further in work by Baker (1979) and Dell (1981), but that work relied on the simplicity metric of early generative grammar that minimized $|G|$ alone. That metric did not work – by focusing only on $|G|$ it led to overly general grammars, as noted by Dell (1981)

– and some of the conclusions of that work do not carry over to balanced learning criteria such as MDL. But the idea was valuable, and we can revisit it now using MDL: given two theories of representation we can consider the MDL predictions of both and see whether the predictions of one of the theories are better than those of the other in how they match the learning behavior of humans.[14] Theory comparison using MDL has been discussed in Katzir 2014 and Rasin and Katzir 2015, 2020, as well as Piantadosi et al. 2016. Here we will show a schematic outline of how such a comparison might be made in the empirical domain of Q-det denotations. Our goal is to show how different representational formats make different Q-dets costly or cheap in terms of MDL. We will not, however, be able to perform an actual comparison and make a choice between formats in the present paper.

3.2. Building blocks

In section 2 we discussed one representational format, namely our variant of SA. In what follows we will try to compare that framework with the following alternative, which we will call *building blocks* (BB), an approach loosely inspired by Keenan and Stavi (1986) and assumed in various later works on Q-dets (see, e.g., Hackl 2009, Piantadosi et al. 2012, and Katzir and Singh 2013). In BB, the representational framework provides various primitive operators and a grammar – in the toy example in (9), which we will assume for the discussion below, a context-free grammar – that determines how these primitives may combine.[15]

(9) Sample BB grammar

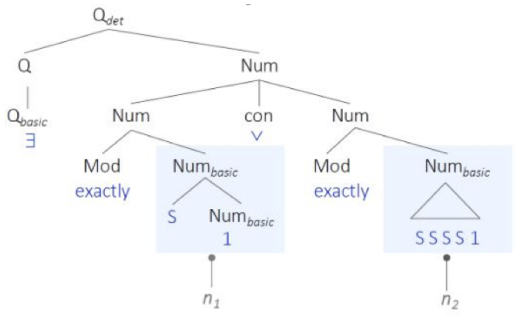
$$\begin{aligned}
Q_{det} &\rightarrow Q\ Num \\
Q &\rightarrow \neg Q \mid Q\ Con\ Q \mid Q_{basic} \\
Num &\rightarrow Num\ Con\ Num \mid Mod\ Num_{basic} \mid \varepsilon \\
Q_{basic} &\rightarrow \forall \mid \exists \\
Con &\rightarrow \wedge \mid \vee \\
Mod &\rightarrow \text{exactly} \mid \text{at least} \mid \text{at most} \\
Num_{basic} &\rightarrow 0 \mid 1 \mid S\ Num_{basic}
\end{aligned}$$

In the – highly simplistic – BB grammar in (9), Q-dets are built out of a quantificational subtree (*Q*, which is built as a boolean combination of existential and universal quantifiers) and a numerical subtree (*Num*). In the numerical subtree, multiple numerals can be represented, each with its own subtree. All numerals other than 0 and 1 must be generated through applications of the successor function *S* (for example, 2 can be represented as *S* 1 or as *S S* 0). Here for example is a derivation tree for a Q-det for 'exactly 2 or exactly 5' using (9).

(10) Derivation tree for 'exactly 2 or exactly 5'

[14] Emphatically, the goal is *not* to see which of the two theories yields better compression of the input data, a measure that as far as we can see is irrelevant to the evaluation of the theories.

[15] A question that arises but that we will not be able to discuss here in any detail is how a BB grammar might relate to the grammar of morpho-syntax. For the present discussion we will assume that BB allows a complex structure to be written also for elements that appear to be morpho-syntactically simplex.



Note how we build this tree by multiple applications of grammar rules, including multiple applications of the successor function in the two relevant subtrees to get the numerals 2 and 5. These multiple and separate applications of the successor function in the two subtrees will be significant for the comparison of BB with SA.

As with SA, we will focus on BB as a format in which lexical Q-dets, which need to be learned and stored, can be written. The example above, for example, will interest us only if the Q-det that it expresses appears on the surface as simplex – e.g., 'gleeb' – and does not reveal its internal structure to the learner.

3.3. Outline of a possible comparison

Semantic automata and building blocks are very different representational formats. The choice between them seems meaningful and should be an empirical matter rather than one of theoretical taste. However, it is not easy to choose between them based on adult judgments alone. The present subsection outlines how MDL-learning might help, though as mentioned in the introduction we will only be able to provide a sketch of what a future comparison might look like.

The key to our comparison of the two frameworks will be the following observation, which we will make more concrete shortly. With SA, G grows with the highest cardinality that the Q-det cares about. Distinctions below that cardinality matter very little. With our grammar for BB, on the other hand, G may grow based also on distinctions below that highest cardinality.[16] We can therefore look for a pair of Q-dets, Q and Q', that care about the same highest cardinality and have the same $|G|$ according to SA but not according to BB. And we can look for an input D that is ambiguous between the two Q-dets and results in the same $|D : G|$ with both.[17] SA will predict that subjects exposed to a D of this kind will show no preference between Q and Q' (since both result in the same $|G| + |D : G|$) while BB predicts that such subjects will show a preference (specifically, for the Q-det that has a smaller $|G|$ under BB). Below are sketches

[16] What matters for the present outlined comparison is the ability to share parts of the representation of numerals. SA have this ability, and the specific grammar for BB that we use does not. One can devise different BB grammars in which sharing is possible, for example by using the kind of multi-dominance structures that have sometimes been used in the syntactic literature (see McCawley 1982, Wilder 1999, and Bachrach and Katzir 2009, among others). The present comparison, then, concerns structure sharing rather than SA vs. BB per se. For presentational convenience we will keep discussing it in terms of the two representational frameworks.

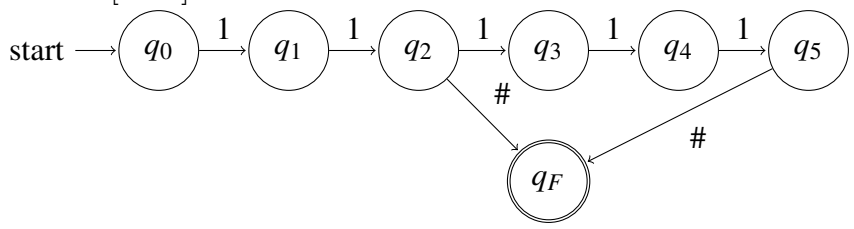
[17] To complete the discussion one needs to specify how $|D : G|$ is computed within the two frameworks. For SA this seems straightforward, as we discussed above, since each SA specifies a way of parsing inputs. BB does not tell us how to parse inputs and thus makes it less clear how to compute $|D : G|$. Here we will assume that $|D : G|$ for BB is computed by translating a given Q-det into an SA for purposes of processing.

of two cases where such comparisons might be made.[18]

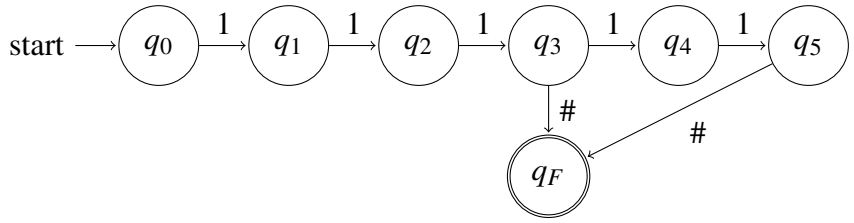
3.3.1. Q-dets for 'all of exactly n_1 or exactly n_2'

First, consider a Q-det 'gleeb' denoting 'all of exactly n_1 or exactly n_2' for some $n_1 < n_2$, which we will write as $\forall[n_1 \vee n_2]$. For 'gleeb' meaning $\forall[2 \vee 5]$ (that is, 'all of exactly 2 or exactly 5'), for example, 'gleeb boxes are on the shelf' will be true if there are exactly two boxes or exactly five boxes and if all those boxes are on the shelf; the sentence will be false otherwise.[19] With BB, Q-dets of this kind will be represented using a tree with two numerical subtrees, similarly to the two subtrees in the representation of the slightly different kind of Q-det in (10) above. If n_1 grows or n_2 grows (or both), so will G. With SA, on the other hand, only n_2 affects the size of G. Different n_1's change the shape of G but not its size. For example, the SA for $\forall[2 \vee 5]$ in (11) has a different shape from the SA in (12) for $\forall[3 \vee 5]$, but the size of the two automata is the same. As to $|D : G|$, note that any individual input sequence (a single sequence of 0's and 1's followed by #) that conforms to a Q-det of this kind will always be the same: exactly 1 bit. This is so since, as can be seen from the SAs in (11) and (12), producing a licit sequence with such SAs requires exactly one, binary choice (exiting q_2 in (11) and exiting q_3 in (12)), while all other transitions are from unary-branching states and cost nothing.

(11) SA for $\forall[2 \vee 5]$



(12) SA for $\forall[3 \vee 5]$



[18]Other, more roundabout comparisons are also possible. For example, instead of comparing Q and Q' directly (through D that is ambiguous between both), it might be more convenient to look at how each of the two Q-dets compares to a third one, for example the simple but non-restrictive Q_0='any number of'. Assuming that Q and Q' are complex but restrictive, MDL will prefer them to Q_0 if the input – here, not necessarily the same D for the two Q-dets – is sufficiently large (in which case the benefits of the restrictive Q-dets in terms of $|D : G|$ will outweigh their disadvantage in terms of $|G|$) but not if it is very small (in which case $|G|$ will play a bigger role than $|D : G|$). The amount of data that warrants moving from Q_0 to either Q or Q' depends, among other things, on the relative size of the two Q-dets, and if they have the same size under SA but different sizes under BB, we may again obtain divergent predictions for the two frameworks. The precise predictions in such cases, however, are somewhat more involved than in the case of a direct comparison of Q and Q' (in part because of the need to factor in $|D : G|$ under Q_0 for inputs corresponding to Q and to Q'), and we will set such comparisons aside in what follows.

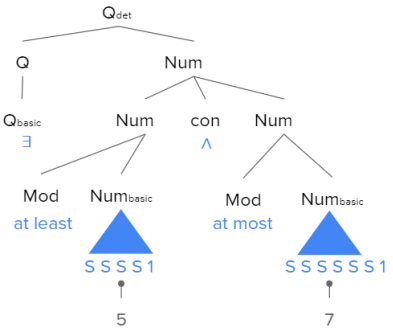
[19]Such Q-dets are of course strange from a typological perspective. We know of no language that lexicalizes a Q-det of this kind. However, the restrictions in (1) make such Q-dets possible, and both SA and BB can represent them. If these are indeed possible lexical Q-dets – as might perhaps be tested in artificial grammar learning experiments – a separate account would be needed to explain why they are typologically unattested.

The above suggests a way to use Q-dets for $\forall[n_1 \vee n_2]$ to compare SA and BB. Consider two Q-dets, $Q = \forall[n_1 \vee n_2]$ and $Q' = \forall[n'_1 \vee n_2]$, with $n_1 < n'_1 < n_2$, and consider an input D consisting of zero or more sequences of n_2 1's followed by #. In the case of $\forall[2 \vee 5]$ and $\forall[3 \vee 5]$, for example, one possible D is $< 1,1,1,1,1,\# >$. Such a D is ambiguous between Q and Q' (among other hypotheses). According to SA, Q and Q' are equally good hypotheses given the data: as just discussed, the two automata are of equal size, while $D : G$ consists of exactly one bit per sequence in both cases, so $|G| + |D : G|$ is the same for both. According to BB, on the other hand, Q is better than Q' given D: $|D : G|$ is still the same for both, but now $|G|$ is smaller for Q' since $n_1 < n'_1$. So all things being equal, SA predicts that Subjects who are exposed to D will show no preference between Q and Q' while BB predicts that such subjects would prefer Q to Q'. Probing such a preference experimentally may of course be difficult – the distance between the comparison just sketched and an actual experiment is big. Still, the expected difference in preferences illustrates the ability of MDL to yield divergent empirical predictions from competing theories of representation.

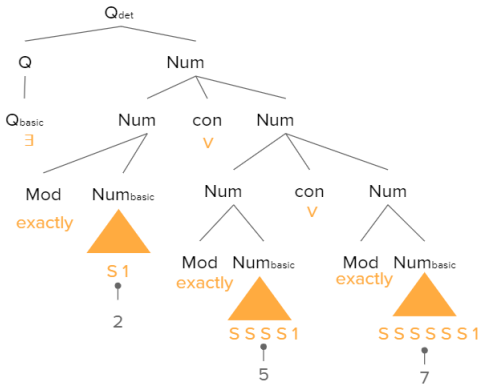
3.3.2. Connected vs. non-connected Q-dets

We now turn to a sketch of a second comparison, one that concerns the difference between connected Q-dets – that is, Q-dets that refer to a single, contiguous sequence of integers (e.g., '5, 6, or 7') – and non-connected Q-dets (e.g., '2, 5, or 7'). With BB, connected Q-dets can have a smaller G than non-connected ones. This is so since, depending on the primitives available on the specific theory of BB, it might be possible to avoid explicitly listing intermediate values in the connected case, while those in the non-connected case need to be explicitly enumerated. For example, the connected '5, 6, or 7' can be represented as 'at least 5 and at most 7', without referring to 6. For the non-connected '2, 5, or 7' no similar compact representations are available. The two BB structures are shown in (13) and (14).

(13) BB representation for 'between 5 and 7' (connected)

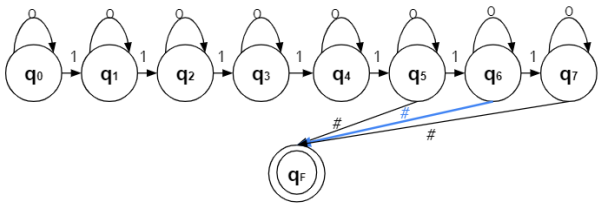


(14) BB representation for '2, 5, or 7' (non-connected)

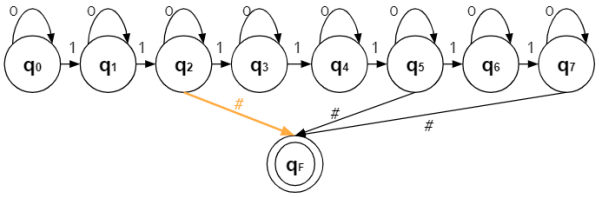


Under the variant of SA presented earlier, on the other hand, no similar shortcuts exist for connected Q-dets. A connected Q-det such as '5, 6, or 7' and a non-connected one such as '2, 5, or 7' have automata of different shapes but of the same size, as illustrated in (15) and (16).

(15) SA representation for 'between 5 and 7' (connected)



(16) SA representation for '2, 5, or 7' (non-connected)



Analogously to what we suggested for $\forall[n_1 \vee n_2]$, we may consider two Q-dets, a connected Q and a non-connected Q' that have the same $|G|$ under SA but not under BB (where $|G|$ is smaller for Q than for Q'). And we can construct an input D that is ambiguous between the two Q-dets (among other hypotheses) and has the same $|D : G|$ for both. All things being equal, SA predicts that subjects exposed to D will show no preference for either of the two Q-dets over the other, while BB predicts that such subjects will prefer Q over Q'. (Again, probing this prediction experimentally may turn out to be non-trivial.)[20]

4. Summary

How we write our stored representations and how we learn them are two important questions in any linguistic domain. Moreover, the two questions are intimately connected, as was noted in

[20]Some suggestive evidence in this domain is provided by Chemla, Buccola, and Dautriche (2019), who show that connected quantificational denotations are easier to learn than non-connected ones (and see Chemla, Dautriche, Buccola, and Fagot 2019 for evidence for a similar connectedness bias in non-humans). However, their experiments were done within a paradigm that provides the learner with negative information, while the comparison that we outlined above relies on the learner encountering positive evidence alone. We can therefore not draw conclusions about the choice between SA and BB based on these results.

early work in generative linguistics. In this paper we outlined the connection in the domain of Q-det semantics. We showed how fixing an explicit format for stored representations – in our example, a variant of SA – yields an evaluation metric based on MDL, which we then turned into an unsupervised learner for Q-dets. We then considered the opposite direction, going from learning to representations, and outlined a possible comparison between SA and BB – two formats that differ from each other in substantial ways but that are both capable of representing Q-dets – based on MDL. We based the comparison on the observation that the two frameworks assign different MDL profiles to Q-dets in certain families. In particular, both the value of n_1 in 'all of exactly n_1 or n_2' (where $n_1 < n_2$) and the connectedness of Q-dets of a particular kind affect the G part of the MDL metric $|G| + |D : G|$ under BB but not under SA.

In both directions, our outline is clearly very preliminary. For the first direction, from representations to learning, much further work is required to establish the actual input data that are available to the child, improve our understanding of the space of Q-dets that children can represent and learn, and compare children's inferences to the predictions of MDL and other learning models. It also remains to be seen if and how MDL scales up to larger, more realistic corpora. In the other direction, from representations to learning, our sketch pointed at what we think is a promising direction for comparing competing frameworks, but we were not able to reach an actual comparison. Still, we hope that our outline is helpful in making explicit various issues involved in both directions and can be of use in future work in this domain.

References

Angluin, D. (1987). Learning regular sets from queries and counterexamples. *Information and computation 75*(2), 87–106.

Bachrach, A. and R. Katzir (2009). Right-node raising and delayed spellout. In K. Grohmann (Ed.), *InterPhases: Phase-Theoretic Investigations of Linguistic Interfaces*, pp. 283–316. Oxford: OUP.

Baker, C. L. (1979). Syntactic theory and the projection problem. *Linguistic Inquiry 10*(4), 533–581.

Barwise, J. and R. Cooper (1981). Generalized quantifiers and natural language. *Linguistics and Philosophy 4*, 159–219.

Berwick, R. C. (1982). *Locality Principles and the Acquisition of Syntactic Knowledge*. Ph. D. thesis, MIT, Cambridge, MA.

Berwick, R. C. (1985). *The Acquisition of Syntactic Knowledge*. Cambridge, Massachusetts: MIT Press.

Brent, M. and T. Cartwright (1996). Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition 61*, 93–125.

Bresnan, J. (1973). Syntax of the comparative clause construction in English. *Linguistic Inquiry 4*, 275–343.

Chemla, E., B. Buccola, and I. Dautriche (2019). Connecting content and logical words. *Journal of Semantics*.

Chemla, E., I. Dautriche, B. Buccola, and J. Fagot (2019). Constraints on the lexicons of human languages have cognitive roots present in baboons *Papio papio*. *Proceedings of the National Academy of Sciences 116*(30), 14926.

Chomsky, N. and M. Halle (1968). *The Sound Pattern of English*. New York: Harper and Row Publishers.

Clark, R. (1996). Learning first order quantifier denotations an essay in semantic learnability. Technical Report IRCS-96-19, University of Pennsylvania.

de Marcken, C. (1996). *Unsupervised Language Acquisition*. Ph. D. thesis, MIT.

Dell, F. (1981). On the learnability of optional phonological rules. *Linguistic Inquiry 12*(1), 31–37.

Gajewski, J. (2010). Superlatives, NPIs and *most*. *Journal of Semantics 27*(1), 125 – 137.

Gold, E. M. (1967). Language identification in the limit. *Information and Control 10*, 447–474.

Grünwald, P. (1996). A minimum description length approach to grammar inference. In G. S. S. Wermter and E. Riloff (Eds.), *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, pp. 203–216. Springer.

Hackl, M. (2009). On the grammar and processing of proportional quantifiers: Most versus more than half. *Natural Language Semantics 17*, 63–98.

Halle, M. (1978). Knowledge unlearned and untaught: What speakers know about the sounds of their language. In M. Halle, J. Bresnan, and G. A. Miller (Eds.), *Linguistic Theory and Psychological Reality*, pp. 294–303. MIT Press.

Hopcroft, J. E. and J. D. Ullman (1979). *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley Publishing Company.

Horning, J. (1969). *A Study of Grammatical Inference*. Ph. D. thesis, Stanford.

Hunter, T. and J. Lidz (2013). Conservativity and learnability of determiners. *Journal of Semantics 30*(3), 315–334.

Katzir, R. (2014). A cognitively plausible model for grammar induction. *Journal of Language Modelling 2*(2), 213–248.

Katzir, R. and R. Singh (2013). Constraints on the lexicalization of logical operators. *Linguistics and Philosophy 36*(1), 1–29.

Keenan, E. L. and J. Stavi (1986). A semantic characterization of natural language determiners. *Linguistics and Philosophy 9*, 253–326.

Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi (1983). Optimization by simulated annealing. *Science 220*(4598), 671–680.

Landman, F. (2004). *Indefinites and the Type of Sets*. Blackwell.

Magri, G. (2015). Universal restrictions on natural language determiners from a PAC-learnability perspective. In D. C. e. a. Noelle (Ed.), *Proceedings of the 37th annual meeting of the Cognitive Science Society*, pp. 1494–1499.

McCawley, J. (1982). Parentheticals and discontinuous constituent structure. *Linguistic Inquiry 13*(1), 91–106.

Orbán, G., J. Fiser, R. N. Aslin, and M. Lengyel (2008). Bayesian learning of visual chunks by human observers. *Proceedings of the National Academy of Sciences 105*(7), 2745–2750.

Paperno, D. (2011, December). Learnable classes of natural language quantifiers: Two perspectives. Ms., UCLA.

Piantadosi, S. T., N. Goodman, and J. B. Tenenbaum (2012). Modeling the acquisition of quantifier semantics: a case study in function word learnability. Under review.

Piantadosi, S. T., J. B. Tenenbaum, and N. D. Goodman (2016, 07). The logical primitives of thought: Empirical foundations for compositional cognitive models. *Psychological review 123*(4), 392–424.

Prince, A. and P. Smolensky (1993). Optimality theory: Constraint interaction in generative grammar. Technical report, Rutgers University, Center for Cognitive Science.

Rasin, E. and A. Aravind (2020). The nature of the semantic stimulus: the acquisition of *every* as a case study. To appear in *Natural Language Semantics*.

Rasin, E., I. Berger, N. Lan, and R. Katzir (2018). Learning phonological optionality and opacity from distributional evidence. In S. Hucklebridge and M. Nelson (Eds.), *Proceedings of NELS 48*, pp. 269–282.

Rasin, E., I. Berger, N. Lan, I. Shefi, and R. Katzir (2019, September). Learning rule-based morpho-phonology. Ms., Leipzig University and Tel Aviv University.

Rasin, E. and R. Katzir (2015). Compression-based learning for OT is incompatible with Richness of the Base. In T. Bui and D. Özyıldız (Eds.), *Proceedings of NELS 45*, Volume 2, pp. 267–274.

Rasin, E. and R. Katzir (2016). On evaluation metrics in Optimality Theory. *Linguistic Inquiry 47*(2), 235–282.

Rasin, E. and R. Katzir (2020). A conditional learnability argument for constraints on underlying representations. To appear in *Journal of Linguistics*.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica 14*, 465–471.

Schafer, J. (2019, March). Some formal aspects of the pLOT theory of learning. Ms., UC Berkeley.

Sipser, M. (2012). *Introduction to the Theory of Computation* (3rd ed.). Course Technology.

Sobel, D. M., J. B. Tenenbaum, and A. Gopnik (2004). Children's causal inferences from indirect evidence: Backwards blocking and Bayesian reasoning in preschoolers. *Cognitive Science 28*(3), 303–333.

Solomonoff, R. J. (1964). A formal theory of inductive inference, parts I and II. *Information and Control 7*(1 & 2), 1–22, 224–254.

Spenader, J. and J. de Villiers (2019). Are conservative quantifiers easier to learn? evidence from novel quantifier experiments. In J. J. Schlöder, D. McHugh, and F. Roelofsen (Eds.), *Proceedings of the 22nd Amsterdam Colloquium*.

Steinert-Threlkeld, S. and I. Icard, Thomas F. (2013). Iterating semantic automata. *Linguistics and Philosophy 36*(2), 151–173.

Steinert-Threlkeld, S. and J. Szymanik (2019). Learnability and semantic universals. *Semantics and Pragmatics 12*(4).

Stolcke, A. (1994). *Bayesian Learning of Probabilistic Language Models*. Ph. D. thesis, University of California at Berkeley, Berkeley, California.

Szymanik, J. (2016). *Quantifiers and cognition: Logical and computational perspectives*, Volume 96. Springer.

Tiede, H.-J. (1999). Identifiability in the limit of context-free generalized quantifiers. *Journal of Language and Computation 1*(1), 93–102.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM 27*(11), 1134–1142.

van Benthem, J. (1986). Semantic automata. In *Essays in Logical Semantics*, pp. 151–176. Dordrecht: Springer Netherlands.

Wilder, C. (1999). Right-Node Raising and the LCA. In S. Bird, A. Carnie, J. D. Haugen, and P. Norquest (Eds.), *Proceedings of WCCFL 18*, pp. 586–598.

Xu, F. and J. Tenenbaum (2007). Word learning as Bayesian inference. *Psychological review 114*(2), 245–272.