19

APPLICATIONS OF MODAL LOGIC IN LINGUISTICS

Lawrence S. Moss and Hans-Jörg Tiede

1 INTRODUCTION 1
2 SEMANTICS 3
2.1 Possible Worlds in Semantics 4
2.2 Specific contributions: an overview 6
2.3 Intensionality 7
2.4 Propositional attitudes 8
2.5 Conditionals 9
2.6 Time and tense 10
2.7 The reference time 13
2.8 Temporal reference and hybrid logic 15
2.9 A note on multidimensionality 16
2.10 Problems and prospects 16
2.11 Montague Semantics 17
3 SYNTAX 23
3.1 Mathematical Linguistics 23
3.2 Preliminary: Logics of Strings 25
3.3 Logics of Trees 30
3.4 Assessment: Why Modal Logic for Syntax and Which One? . 41
4 CONCLUSION AND OPEN PROBLEMS 42

1 INTRODUCTION

That logic and language are closely related is almost true by definition. Logic is concerned with the study of valid inferences in arguments, and these are most commonly defined in terms of truth in models. Symbolic logic studies formal languages (logics) as models of certain aspects of natural languages, such as quantification, while abstracting away from certain other aspects of natural languages, such as ambiguity, as models typically do. Linguistics studies the structure of natural languages as well as the relation of language

to other areas of cognitive science. The roles that logic in general, and modal logic in particular, play in linguistics are quite varied, as we shall see.

In linguistic semantics, logic is used to formalize, or interpret, an *object language*. We take as given that we want to study the semantics of some natural language, and in this chapter the language that we shall deal with is English. Above all else, we would like to directly interpret English sentences in some formally specified model. So even at this point we can see some connection to the Kripke semantics of modal operators: just as all of the other phenomena in this "applied" section of the handbook have been modeled with mathematical structures involving possible worlds, so too have these been used in semantic applications. For example, all manner of linguistic phenomena involving time have led to proposals for using the models from temporal logic. More generally, the main models of all types of *intensional* phenomena are closely related to the models in modal logic.

But so far we have only considered the matter of interpreting natural language directly. Usually, this is difficult or even impossible. (For example, consider the famous quantifier scope ambiguities in sentences like *every handbook has a famous editor*. The ambiguity is neatly expressed in logical notation as $\exists\forall$ vs $\forall\exists$: is there one person, let's call him Dov, who edits all the handbooks, or is it merely that every handbook has some editor or other? One lesson to take from such ambiguities is that it is impossible to associate a function from (English $\times$ models) to truth values in a way that respects our intuitions.) So one way or another, we translate natural language to some artificial language and then interpret that other language, in such a way that ambiguous sentences will be translated into multiple logical formulas. And here is a second place modal logic comes in: the language of higher-order modal logic has been used extensively to drive this translation process, as we shall see when we discuss Montague semantics.

We next turn to syntax, a field in which one finds several different uses of logic. There are syntactic frameworks which are heavily proof-theoretic, so the question of whether a given string is a sentence or not boils down to whether a related (formal) sentence is a theorem in some logical system. This proof-theoretic move is especially prominent in categorial grammar. Another quite different use of logic is as a *meta language* in which one formalizes a linguistic formalism declaratively. This is the move of *model theoretic syntax*, a research program we consider in depth in the second half our chapter. This application relates logic to linguistics in the same way that logic can be applied to formalize theories of other sciences, like set theory. However, the aims of this formalization are somewhat different from those of other areas, since model theoretic syntax is particularly interested in using *decidable* logics for this formalization so that matters can be implemented. This is of course one of the reasons why modal logics are attractive in this context, although much of the focus has been on monadic second-order logic of trees, which is decidable as well.

Applications of logic in linguistics have traditionally not been too concerned with meta-results. The main uses of modal logic in semantics are independent from the main concerns of modal logicians: completeness and correspondence. We are not aware of any serious application of the basic theory of modal logic in semantics, let alone the advanced theory that is showcased in various chapters of this handbook. The only exception is definability theory, interest in which is motivated by trying to find a logic for linguistic applications that has the right kind of "expressiveness." For example, the fact that *most A are B* is not first-order definable is of some importance for semantics. On the

other side, the application of logic in syntax has led to more applications of sophisticated meta-results, for example proof theoretical results like cut-elimination or normalization in categorial grammar. It is interesting to note that definability is also of importance in model theoretic syntax, due to its relation to descriptive complexity theory. A related point: because so many current syntactic frameworks are designed with a hope of implementation, sharper theoretical results about them are called for.

In this chapter, we only survey applications of modal logic to the syntax and semantics of natural languages. We concentrate on these two applications because of the historical importance of modal logic in the development of natural language semantics and because of the significance of model theoretic syntax in current research in mathematical linguistics. There are many areas of applications of logic in linguistics that we do not mention, some of which are surveyed in the *Handbook of Logic & Language* [4].

2 SEMANTICS

Linguistic semantics studies meaning in natural languages. The central assumption of current semantic theory is that meaning should be studied model theoretically, in the same way that semantics of logics are studied. Thus, the study of *meaning* is tied to the concept of *truth*. Of course, there are other ways to pursue the project of understanding meaning, most notably to tie it to *action* in some way. As it happens, for some purposes possible worlds semantics is even better for this second purpose than for the first; see, for example, [69].

The interpretation of logical formulas usually involves the interpretation of subformulas in some systematic fashion. For instance, in propositional logic we have interpretational clauses like

$$[\![\varphi \wedge \psi]\!] = [\![\wedge]\!]([\![\varphi]\!], [\![\psi]\!])$$

where $[\![\wedge]\!]$ is the boolean *and* function. The methodological principle that stipulates that all interpretations of complex expressions should involve the interpretations of its parts is called the "principle of compositionality," and it plays a central role in linguistic semantics. Whether that principle is in fact a meaningful restriction on semantic theory or whether it is vacuous is a point of ongoing debate. For one source that discusses the matter at length, see Janssen [41].

Since natural language semantics applies model theoretic methods, the role of modal logic in this context involves the application of possible worlds semantics to natural languages, mainly to model *intensional* phenomena. However, in order to follow the principle of compositionality uniformly, the meanings of some expressions are modeled using higher-order logic. Thus, the most influential, systematic application of modal logic to linguistic semantics, usually referred to as *Montague semantics* after its founder, involved *higher-order intensional logic*. Although Montague's application of higher-order intensional logic to natural language semantics yielded many important results, almost all of the contemporary research is concerned with finding suitable alternatives to this framework. Many of these are surveyed in the *Handbook of Logic and Language* [4]. Another handbook in this area, with a more empirical and linguistic, as opposed to theoretical and logical, slant is the *Handbook of Contemporary Semantic Theory* [57]. There are many introductory textbooks in linguistic semantics, including [15, 22, 36].

2.1 Possible Worlds in Semantics

The major use of modal logic in semantics stems from possible worlds semantics. Indeed, this is the only kind of application we are considering in this chapter.

This is a good point to make a comment which relates to the place of this chapter in the overall handbook, and also one relating to current practice in the field of modal logic. One of the main subtexts in this volume is that research on modal logic has much to contribute to other areas. So the volume downplays the problematic points of possible worlds semantics by emphasizing topics in modal logic which are interest in areas far removed from those problems. In other words, one can put aside ontological worries (as one would like to do in any mathematical study) because in the kind of transition-system models emphasized in the study, these worries are not relevant. This kind of move is not appropriate for semantics: on the one side, problems about the status of possible worlds come up quicker and they persist; we shall shortly see an example. On the other side, there are few, if any, technical matters of interest in semantics. It is essentially all a matter of studying data from language, proposing treatments that use possible worlds, describing informal models related to the phenomena or the treatments, and occasionally working out the semantics of one or another formal logical language. In this section, we are mainly trying to provide a reader who is conversant with modal logic a feeling for what goes on in semantics.

Here is an example that motivates the use of possible worlds in semantics, taken from McCawley [64]. In a normative English sentence, two uses of a first person pronoun (*I/me/myself*) must be coreferential. And if the sentence has both a first person subject and object, the object must be the reflexive pronoun *myself*. So one cannot say *I kissed me* but instead must have *I kissed myself*. The only exception to this, and this is heart of the matter, is that "Multiple references for first-person pronouns arise when the sentence alludes to an alternative world in which the speaker ... is presented as experiencing something from someone else's vantage point" (McCawley [64]). For example,

(1) I dreamed that I was Brigitte Bardot and that I kissed me.

This would not mean the same thing as *I dreamed that I was Brigitte Bardot and that I kissed myself*. Getting back to (1), it shows what appears to be a hard-and-fast syntactic rule has to be understood in essentially semantic terms. (This is not as surprising as it might at first be: try formulating a principle of reflexive pronouns without using the semantic concepts of *subject* and *object*.)

(1) also shows that in some implicit sense speakers refer to "dream worlds" and more generally to alternative worlds of other kinds, or alternative ways this world could be. Note that the status of who *I* and *me* are in the dream world is problematic, but we shall not delve into this. The point is that if one wants to construct a formal semantics for (1) using Kripke models, then *prima facie* one would want to use worlds: a world where the speaker has a dream, and a world that represents what is happening in that dream. Note as well that what happens in dreams might be logically inconsistent, so Kripke models as one standardly finds them in modal logic are not going to be sufficient for representations of this kind. But they are a useful first step. Indeed, practicing semanticists have found the informal talk about possible worlds to be convenient and motivating. Like contemporary modal logicians, they are usually not interested in, or bothered by, worries about whether possible worlds are real. But again, the difference is that for modal logicians, the worries go away precisely because they tend to avoid

modeling anything like an imaginary world, something which is evidently of linguistic interest.

For another example of why semanticists want to think in terms of possible worlds, consider the following contrasting sentences:

(2) 1. It's certain that you'll find a job, and it's conceivable that it will be a good-paying one.

2. ??It's conceivable that you'll find a job, and it's certain that it will be a good-paying one.

This example is from McCawley [64] in a section entitled "'World-Creating' Predicates." (Incidentally, the quotes here are his, betraying already a certain discomfort with either the notion or the terminology. In any case, we shall expand on just this point below. But the terminology again shows an embrace of possible worlds as well.) The question marks in (2ii) indicates a semantic anomaly. That is, what appears strange in (2ii) is not due to syntactic ill-formedness: from (2i) and the fact that *certain* and *conceivable* both take sentence complements, one would expect (2ii) to be grammatical. So one of the goals of any analysis would be a principled explanation of the different acceptability judgments between (2i) and (2ii).

We encourage readers who are not familiar with semantics to attempt a translation into any logical language of (2i), and also to draw pictures of Kripke models to explain their intuitions. Incidentally, in both (2i) and (2ii), we are concerned with the "non-referential" interpretations: there is no specific job that Gladys is looking for.

Since the sentences in (2) are in the future, a representation should have at least two worlds: the present world, in which (presumably) you do not have a job but are seeking one, and at least one future world. Thus we are inclined to model (2i), say, by having one actual world, w and *many* alternative worlds for the relatively-near future, each with the property that you find some job or other in it. This is quite typical for semantic analysis: if one is going to use worlds to represent alternatives which are only partially specified, then very quickly one must consider many alternatives. Frequently there will be infinitely many. It is the second half of the sentence which gives more trouble; it seems to require that in some (or perhaps some significant proportion of) the successors of w, your job is a good-paying one. And even with this sketch of a representation, we have not learned the lesson that (2) teaches. The point is that a use of *it* in the second part of (2i) is dependent on the existence of a job. If the existence of a job is in doubt, as it is in the first part of (2ii), then it is infelicitous to use *it* to refer to one later. This kind of reasoning could be fleshed out in a fuller analysis in several ways. One would be to use a theory of presupposition. Another would be to ground the whole discussion in Discourse Representation Theory (DRT) [43] or some "dynamic" theory which has enough theoretical apparatus around to talk about different occurrences of pronouns like *it*. In any case, a DRT analysis of the sentences in (2) would most likely use possible worlds at the very least.

One last comment: from more sophisticated examples, such as ... *it is more likely than not that it will be a good-paying one*, we see that complex relations between worlds are going to be the norm rather than the exception. These relations can involve additional structure, as in this probabilistic setting, or various notions of nearness (as we find in treatments of conditionals).

2.2 Specific contributions: an overview

In an assessment of the importance of possible worlds semantics for linguistics, Partee [74] highlights the following six areas:

1. The identification of propositions with sets of possible worlds.
2. The analysis of intensional phenomena with functions from possible worlds to their extensions.
3. The semantics of propositional attitudes.
4. The semantics of conditionals.
5. The semantics of questions and the pragmatics of the question-answer relations.
6. Pragmatics in general, and presuppositions in particular.

Beginning with this subsection, we shall explore several of these contribution areas in detail. Some of these topics are not treated in our chapter: we won't have much to say about questions and answers or pragmatics. The prevalent view of the role of pragmatics in linguistics is that it is the part of semantics that is concerned with the *context-dependent* meaning of linguistic expressions. In the narrowest sense, pragmatics is concerned with the interpretation of indexical or deictic expressions, like personal pronouns, and that is the sense in which the term was used by Montague [99]. Since then, the subject matter of pragmatics has been extended to include many of the topics discussed by ordinary language philosophers like Austin, Grice, Strawson, and Searle. Thus, pragmatics now includes topics such as implicature, presupposition, and speech acts. There is also some overlap between pragmatics and sociolinguistics, though that overlap has little to do with modal logic.

Of the subjects of pragmatics mentioned above, primarily indexicals and presuppositions have been analyzed using possible worlds semantics, although Posner [78] reconstructs communicative actions and ultimately speech act theory in terms of suitable iterations of modal operators for believing, causing, and intending.

Concerning work related to questions and answers, we only mention one quite recent reference, the dissertation Murakami [68]. This proposes an analysis of notions like *complete and just complete answer to a question* based on the modal logic of *partitions*.

With respect to the first two items listed by Partee, one set of notions worth keeping in mind goes back at least to Carnap [14]. He emphasized the distinction between extension and intension, and linked these to particular syntactic items as follows:

Expression	Intension	Extension
sentence	proposition	truth-value
predicate	property	set
individual term	individual concept	individual

Note that the identification of propositions with sets of possible worlds is a special case of this analysis, since sets of possible worlds are essentially the same as functions from possible worlds to the set of truth values. In essence, this identification of propositions with sets amounts to the imposition of a boolean algebra structure on the set of propositions. This kind of structure needs to be supplemented with accessibility relations or other more typically modal structures to be relevant to our discussion.

2.3 Intensionality

We have yet to discuss intensionality in a general way, and now is the time to do just that. It is relatively uncontroversial that some linguistic expressions refer to objects. For instance, names refer to the object that they name. Frege [29] postulated that there is another dimension to "meaning" other than reference in order to explain the apparent difference between statements of the form $a = b$ and $a = a$. While the latter is true *a priori*, the former, when true, typically requires some kind of observation. The difference between these two statements, according to Frege, is that, if $a = b$ is true, a and b have the same *reference* (*Bedeutung*), but different *sense* (*Sinn*). However, Frege did not give a formal definition of "sense." Carnap [14] used the concepts of *extension* and *intension* as a model of reference and sense, which is one of the first and most influential applications of modal logic to natural language semantics. Carnap, however, used *state descriptions*, maximally consistent sets of literals, instead of Kripke models for the semantics of modal logic. Montague is credited with bringing together Carnap's analysis of sense and reference with Kripke's possible world semantics. (Incidentally, a different analysis of *sense* along algorithmic lines has been suggested in recent years by Moschovakis; see [67].)

David Lewis [59] expressed the motivation for this approach to natural language semantics particularly well:

> In order to say what meaning *is*, we may first ask what a meaning *does* and then find something that does that. [...] It is the meaning which determines how the extension depends on the combination of relevant factors. What sort of things determine how something depends on something else? *Functions*, of course [...]

Thus, *intensions* are defined by Lewis to be functions from possible worlds (and possibly other indices, which are the *relevant factors*, above) to extensions. This is the central idea behind the possible world analysis of intensionality. So far, we have only mentioned the extensions of names: individuals. What about the extensions of other kinds of expressions? It has been part of the Fregean orthodoxy to consider the extension of a sentence to be its truth value. Since intensions are functions from possible worlds to extensions, the intensions of sentences are functions from possible worlds to truth values, or simply sets of possible worlds; i.e., those possible worlds in which the sentence is true. The intensions of sentences are also called "propositions." As Partee [74] points out, this analysis of the meaning of sentences gives a good approximation to the notion of "synonymity," since two sentences that have the same intensions are true in exactly the same possible worlds.

Other, classical, examples of intensional phenomena include intensional transitive verbs, like *seek*, and propositional attitude verbs, like *believe*. Interacting with both of these are the *de re* and *de dicto* distinctions. To illustrate this distinction, consider the following sentence, and note the intensional transitive verb: *Barney wants to drive the fastest car in town.* One reading of this sentence is where there is a specific car, say c, and Barney wants to drive c. (But the fact that c is the fastest car in town is not germane to Barney's wish: he just wants to drive c.) This is the *de re* reading. The *de dicto* reading is where Barney wishes to drive whatever car is the fastest in town; so if that description were to change referents over time, then Barney's desire would also change.

Before moving on, we should mention that the entire treatment of intensionality via possible worlds semantics is not universally accepted in semantics by any means. A good

source for some criticisms is John Perry's side [76] of a discussion with Barbara Partee. In other areas as well, one has the feeling that the whole application of possible worlds in semantics is, as one prominent semanticist privately told one of us, a "counterfactual exercise": even though possible worlds semantics are the community's standard and the best thing known, many researchers believe that in the long run they cannot succeed at everything they are being applied to.

2.4 Propositional attitudes

The phenomena of interest in the context of propositional attitudes are belief and knowledge, and also the *root modals* like *can*, *may*, and *should*. The main lines of the standard treatments are probably closest to the heart of a semantically-oriented modal logician: one takes a space of worlds which is equipped with a relation corresponding to each attitude or modality of interest. Then one defines semantics for the attitudes themselves as modal operators in the expected way, by quantifying using the accessibility relations. Modal logicians will also recognize the parallel to the algebraic semantics of modal logic; see Chapter 6. The point is that by moving to the power set algebra of the set of worlds of a model is like moving to the space of *intensions*. So the propositional attitudes turn into operators on the intensional rather than extensional level. This is a two-edged sword: on the one hand, it allows us to explain why statements of identity are not preserved in modal contexts. But the down side is the problem of logical omniscience: logically necessary propositions wind up as being known by everyone at every point. So exactly the same advantages and disadvantages come up as in the theory of knowledge.

Here is one textbook treatment of the basics of propositional attitudes, following the final chapter of Heim and Kratzer [36]. This chapter is called a "first step" on the way to an intensional semantics, and the authors emphasize, and close with, the limitations of their work. The point for us concerns the treatment of the attitude verbs such as *know* and *believe*. The way things work syntactically, attitude verbs take sentences as their arguments; a verb phrase then results. So their categorial type (see Section 2.11 below) would be VP/S, and so their semantics is a function from propositions (i.e., functions from worlds to truth values) to VP meanings (here functions from individuals x to truth values). The semantics is then given by

(3) $[\![believes]\!] =$
$\lambda w.\lambda p.[\lambda x.(\forall w')$(if w' is belief-consistent with w for x, then $p(w') = 1)]$

"Belief consistent" here a relation on worlds defined as follows: we say that w' is belief consistent with w for (person) x if all of x's beliefs in w are true in w'. (So in effect this treatment does not work on a set Kripke model whose accessibility relation is up to the semanticist to specify, but rather that the accessibility relation is *given* in terms of what we are calling belief consistency here.)

To illustrate this, we consider a sentence of the form *Mary believes S*, where S is another sentence. Overall principles of compositionality insure that for all worlds w,

$$[\![Mary\ believes\ S]\!]_w = [\![believes\ S]\!]_w(Mary)$$

The definitions in the semantics are set up so that the following holds:

$$[\![believes\ S]\!]_w(Mary) = [\![believes]\!]_w(\lambda w'.[\![S]\!]_w(Mary)). \tag{4}$$

At this point we apply the general definition of $[\![believes]\!]$ from (3). We see that *Mary believes S* is true in a world w just in case for all worlds w' which are belief consistent with w for Mary, S is true in w'. So in this way we reconstruct the semantics of the attitudes which would be expected from the Kripke semantics of modal logic. We shall discuss how the calculation in (4) works when we turn to Montague grammar in Section 2.11.

Before returning to a discussion of the modalities in language, here is another point on the treatment of belief in linguistics. McCawley [64] suggests a departure from what modal logicians might expect concerning belief when he writes, "Belief worlds may even conform to a different version of logic than the real world is taken to be subjected to; such worlds would be appropriate devices for analyzing such sentences as those in which an adherent of standard logic attributes beliefs to an intuitionist." But he also holds also that "one has a single set of beliefs at a time (possibly inconsistent beliefs, but a single set nonetheless)." So this seems to suggest that belief worlds might be *paraconsistent* in some sense. But later, in connection with wishes, he is of the opinion that "It will probably be clearest if one simply avoids terms such as 'wish world,' which misleadingly suggest that there is a single system of wishes whose simultaneous fulfillment is at issue, and instead use circumlocutions to say that a particular world corresponds to a particular wish. ... These worlds may serve as reference worlds for other worlds that correspond to, say, the fulfillment of wishes, hopes, and so forth, that are contingent on the fulfillment of a given wish."

In any case, much of the linguistic discussion is not about these kinds of points, but rather questions of reference and presupposition. For example, here are sentences from McCawley [64], page 426:

(5) 1. Arthur thinks that a unicorn has been eating his roses. He hopes he can catch it.

2. Arthur denies that a unicorn has been eating his roses. ??He hopes he can catch it.

The underlined *it* in the first sentence refers to the unicorn in the preceeding sentence. Actually, *a unicorn* is best understood non-referentially here; there is no particular unicorn which Arthur is thinking about, just some-unicorn-or-other. The point is that it is a property of *think* that it allows nonreferential NP's in its complement to be the antecedents of later pronouns (the subsequent *it*). In contrast, *deny* does not have this property. This is why the second sentence in (2 ii) is anomalous. This last example is intended to be more typical of the linguist's concerns than the previous paragraph on ontological points.

2.5 Conditionals

Modal-type notions are of central interest in work on conditionals, following Lewis [60] and Stalnaker [90]. The idea here is to analyze counterfactual conditionals (those whose antecedents are known or taken to be false) using a semantics that comes with some extra apparatus or other. One should see Chapter 18 for an extensive treatment of the logical systems that come from the natural semantics of counterfactual conditionals. But counterfactuals are only one type of conditionals, and another important type are indicative conditionals (where the antecedent is true). See von Kutschera [106] for a

proposal on indicative conditionals related to, but different from, the standard treatment of counterfactuals.

As it happens, most contemporary work in semantics does not use the Lewis-Stalnaker semantics but instead works with elaborations based on it. Probably the main proposal in the area is due to Kratzer [55]. Her work allows one to work with the combination of conditionals and modals, as in *If this is an article on linguistics, there must be examples from many languages*. Kratzer's semantics makes use of a "modal base"; this is basically a spelled out version of an accessibility relation. It also uses a three-placed similarity relation on worlds. It was later observed by Frank [28] and independently Zvolensky [107] that sentences of the form *If X, then it must be the case that X* came out automatically true in Kratzer's semantics. Modal logicians might find it interesting to note that a similar debate about the adequacy of modal semantics crops up in areas like deontic logic. Indeed, one of the lessons we learned in writing this chapter is exactly that similar questions about the adequacy of various semantic proposals coming from modal logic come up independently in different forms. From the point of view of *applied modal logic* it would clearly be of value for people to pay close attention to points like this, in order to make theoretical contributions that could be appreciated by people in different fields.

2.6 Time and tense

Another semantic area where ideas of possible worlds semantics are put to use concerns time and tense in natural language. Our discussion of these issues is once again intended only as an invitation to this fascinating field. It is based largely on the survey of the area in Mark Steedman's draft textbook [91] and also on Dick Crouch's ESSLLI notes [21]. We have also again found McCawley's book [64] full of insightful examples and proposals; see Section 12.2 on Tense Logic.[1] An essential resource for researchers in this area is Robert I. Binnick's web site [5] entitled "The Project on Annotated Bibliography of Contemporary Research in Tense, Grammatical Aspect, Aktionsart, and Related Areas":

`http://www.scar.utoronto.ca/~binnick/TENSE/`

In particular, the logic part of the site lists a large collection of papers relevant to the subject of this handbook chapter.

For readers of this handbook, perhaps the primary observation concerning the analysis of time in natural language is that the whole matter of *temporal ontology* is highly complicated and problematic. First of all, there are words, endings, and expressions which are *usually* used to indicate past, present, and future time references. But even these have exceptions. For example, *ing* usually indicates a present tense, but in examples like *the editors are calling Larry tomorrow to complain that his paper contained a lot of misleading remarks*, the word *tomorrow* changes this to a future time reference. Yet another point concerns *embedded tenses*; as our last example shows, it is not always

[1] Indeed, at various places in this chapter we have marveled at McCawley's use of the best logical tools available to him. He wrote "I teach courses on logic from a linguists point of view, taking a broad view of the subject matter of logic (logic has suffered 23 centuries of myopia, which I try to make up for) and giving full weight to linguistic considerations in revising (or replacing) existing systems of logic to maximize their contact with natural language syntax and linguistic semantics." [65] We therefore wonder what he as a linguist would have found useful in the exploding logical literature. Although we never met him personally, we would like to think that some of our comments about various connections and possible applications would have inspired him (and those who follow in his footsteps).

straightforward even to interpret these constructions, let alone represent and analyze them. For another example along the same lines, *Sonia said that Rajiv liked to dance* should have the same meaning as *Sonia said "Rajiv likes to dance"*; the problem then is to account for this sameness. Finally, important temporal information is often absent from the surface forms. Consider

(6) John went to kindergarten with a bank president.

The intended meaning is that at some past time John went to kindergarten with some individual who would later become a president of a bank.

The first, and most basic, proposal for the representation of temporal phenomena is to add an explicit time parameter to propositional functions. So instead of a predicate like $alive(x)$ which indicates whether an object is alive or not, we might have $alive(x, t)$. Then one might want to translate various tense constructions into, say, a two-sorted first-order logic; the point is that one then has quantification over times and also a symbol $<$ for the relation of *preceding* on times. Then one can translate a future sentence like *Sonia will go* as

$$(\exists t > t_0)(go(Sonia, t))$$

Note that there is a "now" time t_0; this can be taken to be either a constant or a variable.

However, this is usually not what is done. There are logical and also linguistic reasons for making other moves. In a comment directly related to this, Thomason [100] writes:

> Physics should have helped us realize that a temporal theory of a phenomenon X is, in general, more than a simple combination of two components: the statics of X and the ordered set of temporal instants. The case in which all functions from times to world-states are allowed is uninteresting; there are too many such functions, and the theory has not begun until we have begun to restrict them. And often the principles that emerge from the interaction of time with the phenomena seem new and surprising.

The new and surprising principles here are the interactions of tense and modality that Thomason discusses in his handbook article [100]. But mention of physics also raises the question of the structure of time. In the linguistic literature, the emphasis nearly always is on what might be called *linguistic time*, the common-sense notion that we want to tease out and model from "people on the street". It is not the notion that we would get from physics.

An alternative way to go is to take the basic sentences of language to be tenseless and then to add temporal modal operators P (for the past) and F (for the future). This is the basic move of Tense Logic, usually mentioned in connection with its main developer, Arthur Prior.

These are interpreted on linear orders $(L, <)$. The semantics is the standard one from temporal logic

$$\begin{array}{lll} l \models P\phi & \text{iff} & m \models \phi \text{ for some } m < l \\ l \models F\phi & \text{iff} & m \models \phi \text{ for some } m > l \end{array} \tag{7}$$

So P and F are *past* and *future* modalities.

Let us see how this idea fares with some examples. We should think of an atomic proposition as representing an untensed assertion. After a moment's thought, one can see that the very question of whether "untensed assertions" are possible will be a source

The oracle speaks	p	The oracle spoke/has spoken	Pp
The oracle will speak	Fp	The oracle had spoken	PPp
The oracle will have spoken	FPp	The oracle never spoke	$\neg Pp$
There will be a time after which the oracle will not speak			$F\neg Fp$
There was a first time the oracle spoke			$P(p \wedge \neg Pp)$

Figure 1. Sentences and Priorean Translations

of debate in this area. But let us ignore this and think of stative present tense assertions like *the oracle speaks* as an untensed assertion. Suppose that we take its semantics to be an atomic proposition p of the logic above. Then we can translate some English sentences as in Figure 1.

It is important to make a few comments about the contents of the figure. As with all translations from natural language into a formal language, one has to be clear on what has been achieved and what some of the problems are.

We also mention some ways that the system can be fruitfully extended. For example, it is straightforward to add binary modalities S and U for *since* and *until*. With a little more work, we can also add *now*. The simplest way to do this is to work on models $(L, <)$ with a distinguished l^* for the "present moment". Then we add to the clauses in (7) the following

$$l \models N\phi \quad \text{iff} \quad l^* \models \phi.$$

This proposal is due to Kamp [42], and it is discussed further in Burgess [11], Section 4B. Among the facts shown in these references is the fact that N is actually eliminable in this language. However, if one moves from a purely propositional setting one with more linguistically interesting phenomena, this reduction is rightfully lost. For example, consider

(8) 1. The oracle predicted that there will be an earthquake.

2. The oracle predicted that there would be an earthquake.

A natural representation of (1) is $Pr(o, NFe)$; the important point is that the future operator F is evaluated from the vantage point of "now". This contrasts with (2). Here a representation might be $Pr(o, Fe)$. The difference is that the prediction in (2) is that there will be an earthquake at some point later than the prediciton, not the moment of utterance of the sentence.

There are some linguistic problems with any treatment of time as an extra parameter. One problem again concerns embedded tenses; these are especially interesting for modal logic since all of the important problems in modal logic arise precisely because modalities in formal systems may be iterated, and because accessibility relations in models can be deep. The natural symbolization of (6) in a modal approach comes out as something like

$$P \quad ((\exists x)(\exists y)(\text{kindergarten}(x) \wedge \text{go}(J, x) \wedge \text{go}(y, x) \wedge F((\exists z)(\text{bank}(z) \wedge \text{president}(y, z)))))$$

But then consider *John went to kindergarten with someone who has become a bank president*. Here the intended reading is that the person became a bank president before the

utterance time. So having F in the scope of P in the representation would be a mistake. Another problem is that many temporal phenomena pertain more to events that distributed in time and hence do not admit a nice formulation: *Boris took piano lessons for six months*.

Sentences like Pp may be rendered either in the *simple past* as *The oracle spoke* or in the *present perfect* as *The oracle has spoken*. This means that whatever differences we ascribe to the two English forms will not be representable in the Priorean formalism.

Further, the logic contains forms like $PFPPFp$ which cannot be rendered into English except by transcribing the formal semantics into mathematical English. This is a problem not just for this work, but also for practially all accounts of any phenomenon which use recursion: the formalism will quickly contain forms not naturally renderable without heavy uses of devices like numbered or named pronouns.

For other natural English sentences that cannot be translated adequately in the Priorean formalism, consider *The oracle did not speak*. What we have here is an implicit reference to a particular time or set of times. So our sentence is not captured by $\neg Pp$, since that sentence amounts to a universal quantification over past times.

Furthermore, one would suspect that since we can add operators corresponding to *Since* and *Until*, we might also add an operator Y for *Yesterday*. Suppose our semantics makes use of a function $l \mapsto l - 1$ and works by $l \models Y\phi$ iff $l - 1 \models \phi$. However, here the a sentence like *Yesterday the oracle spoke* would correspond to Yp rather than YPp. So we are left with a puzzle about why the natural language sentence uses the past tense marker in the first place.

Hinrichs [37] noted that the sentence *Vincent left yesterday* has two natural renderings:

$$\textit{Y(P(leave(Vincent)))} \qquad \text{and} \qquad \textit{P(Y(leave(Vincent)))}$$

However, these both fail to have the intended meaning: the Y operator shifts the evaluation point to the previous day, but then the P operator takes the past from this point. A similar problem, noted by Partee [73] and reiterated in Hinrichs' paper is that tense and negation do not work well in Prior's approach. Translating "Vincent did not leave" by either $P(\neg leave(V))$ or the other alternative do not work.

Even though the rest of our discussion has dwelled the shortcomings of the Priorean approach, some aspects of the temporal system of language clearly are captured in it. Further discussion of tense logic and standard logic may be found in van Benthem [103, 104] and also Chapter 11 of this handbook. We also discuss an extension of Prior's approach due to Patrick Blackburn in Section 2.8 below.

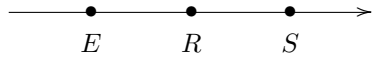
2.7 The reference time

One of the key contributions to this area comes in Reichenbach's textbook (on logic (!)) [82]. He points out that linguistic tense does not involve only "now" and "then" but also a third time, the *reference time*. So he described the tense system in terms of three times, S (the speech point), E (the event point), and R (the reference point). R is the time in a sentence "that we are really talking about".

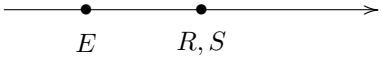
For example, consider the difference between the *simple past* and *past perfect* in English. The simple past is exemplified by *I saw John*; the past perfect by *I had seen John*. The difference between these is that in the past perfect, E is prior to R (and both come before S): the speaker is describing an event from a vantage point (R) which is later

than the time (E) of the event itself. In contrast, in the simple past, the event time and reference time are the same.

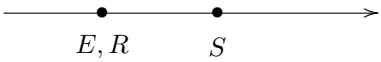
Incidentally, although Reichenbach seems to have preferred to think in terms of R, E, and S as *points*, there is good reason to prefer to take them to be *intervals*. The use of intervals in tense logic is a natural move, and many semantics papers do in fact make it. Here are some examples of the way various tense and aspect combinations in Reichenbach's system come out when we take R, E, and S to be intervals, writing $<$ for the subinterval relation:



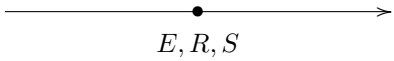
the past perfect: I had seen John



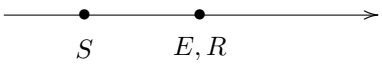
the present perfect: John has been to Boston



the simple past: I gave a lecture



the present: I name this ship *Chips*



the simple future: I lecture (next week)

We have completed a quick tour of some of the central proposals concerning temporal ontology. It goes without saying that we have barely scratched the surface, that further work is in large measure concerned with corrections and criticisms of the classical ideas, etc. But we would be remiss in ending without metioning that much of the current work is concerned not with points of time but rather with *events*; see for example, the book *The Proper Treatment of Events* [105]. Figure 2 contains a chart of some of the Reichenbach examples, worked in terms of events for S, E and R. The relation $<$ is that of *subevent*.

	Past	Present	Future
Simple	$E = R$, $R < S$ Mary saw John	$E = R = S$ Mary sees John	$E = R$, $S < R$ Mary will see John
Perfect	$E < R < S$ Mary had seen John	$E < R = S$ Mary has seen John	$E < R$, $S < R$ Mary will have seen John
Progressive	$E = R$, $R < S$ Mary was seing John	$E = R = S$ Mary is seeing John	$E = R$, $S < R$ Mary will be seeing John

Figure 2. Tense and Aspect in Terms of E, R, and S

Structure	**Name**	**English example**	**Representation**
E–R–S	Pluperfect	I had seen	$P(i \wedge P\varphi)$
E,R–S	Past	I saw	$P(i \wedge \varphi)$
R–E–S	Future-in-the-past	I would see	$P(i \wedge F\varphi)$
R–S,E	Future-in-the-past	I would see	$P(i \wedge F\varphi)$
R–S–E	Future-in-the-past	I would see	$P(i \wedge F\varphi)$
E–S,R	Perfect	I have seen	$i \wedge P\varphi$
S,R,E	Present	I see	$i \wedge \varphi$
S,R–E	Prospective	I am going to see	$i \wedge F\varphi$
S–E–R	Future perfect	I will have seen	$F(i \wedge P\varphi)$
S,E–R	Future perfect	I will have seen	$F(i \wedge P\varphi)$
E–S–R	Future perfect	I will have seen	$F(i \wedge P\varphi)$
S–R,E	Future	I will see	$F(i \wedge \varphi)$
S–R–E	Future-in-the-future	(Latin: abiturus ero)	$F(i \wedge F\varphi)$

Figure 3. Reichenbach's analysis in hybrid logic

2.8 Temporal reference and hybrid logic

Reference to specific times can be incorporated into a Prior-style formalism by using ideas from hybrid logic (see Chapter 14 of this handbook). The basic idea of hybrid logic is to add a new sort of propositional symbol to the underlying modal language; these symbols are called *nominals*, and they are typically written i, j, and k. When working with nominals, one adds a semantic constraint that they be true at exactly one point. In this way, nominals 'name' the unique point they are true at.

This gives us a way of coping with some of the difficulties noted earlier. For example, we saw that *The oracle could not speak* could not be translated into the standard Priorean formalism; the simple representation $\neg Pp$ amounts to universal quantification over past times. But with the aid of nominals we have a better representation:

$$P(i \wedge \textit{the oracle not speak}).$$

This anchors the silence of the oracle at a particular time in the past, namely the time named by the nominal i.

Moreover, we now have a way handling reference times. Consider the sentence *The oracle had not spoken*. This picks out some past time (the reference time) and locates the silence of the oracle before that. This cannot be handled in the standard Priorean formalism, but once again with the aid of nominals, we can capture its meaning:

$$P(i \wedge P(\textit{the oracle not speak})).$$

This formula says that there is some time in the past (namely the one named by i) and that *before that* the oracle did not speak. In fact, as Blackburn [6] observes, all of Reichenbach's analyses can be handled in this way; the required hybrid representations are given in Figure 3.

It's also worth remarking that the ideas of hybrid logic combine naturally with multi-dimensional modal semantics of the type mentioned above. For example, Blackburn [6] uses this style of semantics to interpret propositional symbols like *yesterday*. The use

of such symbols avoids the problems associated with the yesterday operator Y. For example, the hybrid representation of *Vincent left yesterday* would be

$$P(\mathit{yesterday} \wedge \mathit{Vincent\ leave}),$$

and this has the required interpretation. Moreover, the hybrid approach also correctly classifies sentences such as *Vincent will leave yesterday* as semantically anomalous. This sentence would be represented by the hybrid formula

$$F(\mathit{yesterday} \wedge \mathit{Vincent\ leave}).$$

This formula is false at all points in all models, hence the anomaly.

2.9 A note on multidimensionality

One very interesting application of modal logic to semantic analysis is the use of multidimensional modal logic in connection with cross-world comparatives. Consider, for example, sentences like *This article is shorter than it might have been.* One approach to its semantics is to use not just a single world in the semantics, but to move to two or even more worlds. We might have an "actual" world and an "evaluation" world. For applications in semantics related to comparatives, see Cantwell [13]. By now there are also quite sophisticated modelings of tense and aspect; see, for example ter Meulen [94].

2.10 Problems and prospects

Even with the move to a Reichenbachian treatment of tense and aspect, there are remaining stubborn problems. Many of these are especially pertinent to the discussion of the application of possible world semantics; they indeed cause one to either re-think the use of possible worlds, or to propose modifications or extensions of it. Consider, for example, the *present relevance* of the perfect. For example, *Jimmy has lost his mind* intuitively entails that Jimmy has not gotten it back.

It is also important to note that a lot of real-world knowledge goes into judgments about sentences using time and tense. For example

(9) 1. ??James McCawley has written many books.
2. James McCawley wrote many books.

The first is anomalous, but only to one who knows that McCawley died in 1999. For someone who didn't know this, (9i) carries the implicature that McCawley is still alive. The point here is that (9i) and (9ii) are not equivalent, but the difference is due to background knowledge. So the entire system of time/tense/aspect interacts with the knowledge background of speakers and hearers. With this in mind, consider also, and note the difference between

(10) 1. The authors have regretted that they never met McCawley.
2. The authors had regretted that they never met McCawley.

Many recent papers and books in the area emphasize the presence of causality and real-world knowledge in discourse about time; see Steedman [91]. Another book on this topic which emphasizes connections to logic programming and even robotics is van Lambalgen and Hamm [105].

Category	Description	Examples
S	Sentence	*John seek a unicorn*
CN	Common nouns	*man, woman, unicorn*
IV	Intransitive verb phrases	*walk*
S/IV	Noun phrases	*the man, every unicorn, John*
(S/IV)/CN	Determiners	*every, a, one, the*
IV/(S/IV)	Transitive verb phrases	*love, seek*
IV/S	Sentential complement verbs	*believe, hope, doubt*
CN/CN	Adjectives	*red, fake*
S/S	Sentential adverbs	*frequently, necessarily*

Figure 4. Categories and Sample Expressions in a Categorial Grammar

2.11 Montague Semantics

As was mentioned above, Montague's application of higher-order intensional logic marked the starting point for applications of modal logic in natural language semantics. Montague developed his theory of natural language semantics over the course of three papers (collected in [99]), each of which differs from the others in some respect. In the following, we give a survey of a "streamlined" approach, taken from Gamut [32].

Montague semantics consists of three parts: syntactic categories, semantic types, and operations on the members of each of these, where each operation on members of syntactic categories has a corresponding operation on the members of the corresponding semantic types. This correspondence is Montague's formalization of the principle of compositionality.

The theory of syntactic categories assumed in Montague semantics is loosely based on categorial grammar. The categories of categorial grammar are either basic categories or derived categories, which are formed by closing the basic categories under two operators: / and \. An expression that has a category of the form A/B is "looking to its right for" an expression of the category B to make an expression of the category A. And an expression of category $B\backslash A$ is looking to its left for an expression of category B to again make one of category A. Thus, derived categories have a functional behavior.

The full set of categories, CAT, is obtained by closing the basic categories, S (for *sentence*), CN (*common noun*), IV (*intransitive verb*), under the / operator. Thus, only one of the two operators of categorial grammar is used by Montague. Figure 4 has examples of the most common categories and some of their expressions.

At this point, we have described the set of syntactic categories and given examples. One forms the full set of expressions of the various categories by juxtaposition following the categorial rules. For example, *John walks* is an S because *John* is S/IV and *walks* is an IV. From this, *believes John walks* is an IV. And then *Mary believes John walks* is again an S. This is as it should be, since we have a grammatical sentence.

We return to the syntactic categories and expressions below, after a digression concerning the formal semantics. Let e and t be any distinct objects, and define the set T of *semantic types* by the following inductive definition:

1. $e, t \in T$,

2. if $a, b \in T$, then $\langle a, b\rangle \in T$,
3. if $a \in T$, then $\langle s, a\rangle \in T$.

The idea is that e stands for *entity* and t for *truth value*, s for a set of possible worlds, and $\langle a, b\rangle$ for the set of all functions from a to b (or rather for the set of functions from the set that a stands for to the set b stands for). The difference between syntactic categories and semantic types is that syntactic categories have a notion of "order" built-in. In the following, we will use upper-case letters for syntactic categories and lower-case letters for semantic types.

Given infinite sets of variables for each type a, denoted by VAR_a, and, possibly empty, sets of constants for each type a, denoted by CON_a, we define the well-formed expressions of type a, denoted by WE_a, as follows:

1. $\mathsf{VAR}_a \subseteq \mathsf{WE}_a$ and $\mathsf{CON}_a \subseteq \mathsf{WE}_a$,
2. if $\alpha \in \mathsf{WE}_{\langle a,b\rangle}$ and $\beta \in \mathsf{WE}_a$, then $\alpha(\beta) \in \mathsf{WE}_b$,
3. if $\varphi, \psi \in \mathsf{WE}_t$, then $\neg\varphi \in \mathsf{WE}_t$ and $(\varphi \wedge \psi) \in \mathsf{WE}_t$,
4. if $\varphi \in \mathsf{WE}_t$ and $v \in \mathsf{VAR}_a$, then $\forall v\varphi \in \mathsf{WE}_t$,
5. if $\alpha, \beta \in \mathsf{WE}_a$, then $\alpha = \beta \in \mathsf{WE}_t$,
6. if $\alpha \in \mathsf{WE}_a$ and $v \in \mathsf{VAR}_b$, then $\lambda v\alpha \in \mathsf{WE}_{\langle b,a\rangle}$,
7. if $\varphi \in \mathsf{WE}_t$, then $\Box\varphi \in \mathsf{WE}_t$,
8. if $\alpha \in \mathsf{WE}_a$, then ${}^{\wedge}\alpha \in \mathsf{WE}_{\langle s,a\rangle}$,
9. if $\alpha \in \mathsf{WE}_{\langle s,a\rangle}$, then ${}^{\vee}\alpha \in \mathsf{WE}_a$.

We will use other connectives: $\Diamond, \vee, \rightarrow, \leftrightarrow, \exists$, to abbreviate their usual definitions in terms of the connectives above. The reason that these types are referred to as *semantic* types is that each type has a corresponding domain. Given a set of individuals, D, and a set of worlds, W, we define the domain of a type t, denoted by $\mathcal{D}_{t,D,W}$ as follows:

1. $\mathcal{D}_{e,D,W} = D$
2. $\mathcal{D}_{t,D,W} = \{0, 1\}$
3. $\mathcal{D}_{\langle a,b\rangle,D,W} = \mathcal{D}_{b,D,W}^{\mathcal{D}_{a,D,W}}$
4. $\mathcal{D}_{\langle s,a\rangle,D,W} = \mathcal{D}_{a,D,W}^{W}$

where A^B denotes the set of functions from B to A. We now define the interpretation of expressions. A *model*, $\mathcal{M}$, is a triple, (D, W, I), where D is a non-empty set of individuals, W is a non-empty set of worlds, and I is an interpretation of the constants at a world. We define $[\![\alpha]\!]_{\mathcal{M},w,g}$, where $\mathcal{M}$ is a model, $w \in W$ is a world, and g is a variable assignment. As usual, we denote the variable assignment that differs from g at most in that it assigns d to v by $g[v/d]$.

1. if $\alpha \in \mathsf{CON}_a$, then $[\![\alpha]\!]_{\mathcal{M},w,g} = I(w, \alpha)$; if $\alpha \in \mathsf{VAR}_a$, then $[\![\alpha]\!]_{\mathcal{M},w,g} = g(\alpha)$,

2. if $\alpha \in \mathsf{WE}_{\langle a,b\rangle}$ and $\beta \in \mathsf{WE}_a$, then $[\![\alpha(\beta)]\!]_{\mathcal{M},w,g} = [\![\alpha]\!]_{\mathcal{M},w,g}([\![\beta]\!]_{\mathcal{M},w,g})$,
3. if $\varphi, \psi \in \mathsf{WE}_t$, then $[\![\neg\varphi]\!]_{\mathcal{M},w,g} = 1$ iff $[\![\varphi]\!]_{\mathcal{M},w,g} = 0$, and $[\![(\varphi \wedge \psi)]\!]_{\mathcal{M},w,g} = 1$ iff $[\![\varphi]\!]_{\mathcal{M},w,g} = 1$ and $[\![\psi]\!]_{\mathcal{M},w,g} = 1$,
4. if $\varphi \in \mathsf{WE}_t$ and $v \in \mathsf{VAR}_a$, then $[\![\forall v\varphi]\!]_{\mathcal{M},w,g} = 1$ iff for all $d \in \mathcal{D}_a$, $[\![\varphi]\!]_{\mathcal{M},w,g[v/d]} = 1$,
5. if $\alpha, \beta \in \mathsf{WE}_a$, then $[\![\alpha = \beta]\!]_{\mathcal{M},w,g} = 1$ iff $[\![\alpha]\!]_{\mathcal{M},w,g} = [\![\beta]\!]_{\mathcal{M},w,g}$,
6. if $\alpha \in \mathsf{WE}_a$ and $v \in \mathsf{VAR}_b$, then $[\![\lambda v\alpha]\!]_{\mathcal{M},w,g}$ is the function $h \in \mathcal{D}_{\langle b,a\rangle}$, such that for all $d \in \mathcal{D}_b$, $h(d) = [\![\alpha]\!]_{\mathcal{M},w,g[v/d]}$,
7. if $\varphi \in \mathsf{WE}_t$, then $[\![\Box\varphi]\!]_{\mathcal{M},w,g} = 1$ iff for all $w' \in W$, $[\![\varphi]\!]_{\mathcal{M},w',g} = 1$,
8. if $\alpha \in \mathsf{WE}_a$, then $[\![^\wedge\alpha]\!]_{\mathcal{M},w,g}$ is the function $h \in \mathcal{D}_{\langle s,a\rangle}$, such that for all $w' \in W, h(w') = [\![\alpha]\!]_{\mathcal{M},w',g}$,
9. if $\alpha \in \mathsf{WE}_{\langle s,a\rangle}$, then $[\![^\vee\alpha]\!]_{\mathcal{M},w,g} = [\![\alpha]\!]_{\mathcal{M},w,g}(w)$.

This allows us to have formal terms for what we informally wrote above in (3).

There is a symmetry between λ-abstraction and application (i.e. β conversion), and $^\wedge$ abstraction and $^\vee$ application. However, while the following form of β-conversion can only be applied in the *extensional* fragment of this system, it only holds in restricted cases in the intensional system.

THEOREM 1. *In the extensional fragment of $\lambda x\beta(\gamma)$ is equivalent to $\beta[x \mapsto \gamma]$ if all free variables in γ are free for x in γ.*

However, in the extensional system this equivalence fails. It is possible to extend this equivalence to a restricted set of expressions of the intensional system: the intensionally closed expressions, whose extension does not vary from world to world. The *intensionally closed expressions* in L, denoted by ICE^L, is the minimal subset of WE^L such that

1. If $x \in \mathsf{VAR}_a$, then $v \in \mathsf{ICE}^L$,
2. If $\alpha \in \mathsf{WE}^L_a$, then $^\wedge\alpha \in \mathsf{ICE}^L$
3. If $\varphi \in \mathsf{WE}^L_t$, then $\Box\varphi \in \mathsf{ICE}^L$
4. ICE^L is closed under boolean connectives, quantifiers, and λ-abstraction.

The above mentioned symmetry is summarized in the following two theorems:

THEOREM 2. $^\vee(^\wedge\alpha)$ *is equivalent to* α.

THEOREM 3. $\lambda x\beta(\gamma)$ *is equivalent to* $\beta[x \mapsto \gamma]$ *if*

1. *all free variables in γ are free for x in γ; and*
2. *either $\gamma \in \mathsf{ICE}^L$, or no free occurrence of x in β lies within the scope of $\Box$,$^\wedge$.*

Now, we associate semantic types with syntactic categories as follows using the following function f:

$$f(\mathsf{S}) = t$$
$$f(\mathsf{CN}) = f(\mathsf{IV}) = \langle e, t\rangle$$
$$f(\mathsf{A/B}) = \langle\langle s, f(\mathsf{B})\rangle, f(\mathsf{A})\rangle$$

With each syntactic category, A, we associate a set of *basic expression* of that category, denoted by B_{A}, and a set of expressions of that category, denoted by P_{A}.

The next step involves the definition of syntactic operations that create complex expressions. In the following, we will use the same rule numbers as [32]. Here are the first three rules:

$$B_{\mathsf{A}} \subseteq P_{\mathsf{A}} \tag{S1}$$

$$\text{If } \alpha \in P_{\mathsf{S/IV}} \text{ and } \beta \in P_{\mathsf{IV}}, \text{ then } F_1(\alpha,\beta) \in P_{\mathsf{S}}, \text{ and } F_1(\alpha,\beta) = \alpha\beta', \text{where } \beta' \tag{S2}$$

is the result of replacing the main verb in β by its third-person singular present form.

$$\text{If } \alpha \in P_{(\mathsf{S/IV})/\mathsf{CN}} \text{ and } \beta \in P_{\mathsf{CN}}, \text{ then } F_2(\alpha,\beta) \in P_{\mathsf{S/IV}}, \text{ and } F_2(\alpha,\beta) = \alpha\beta. \tag{S3}$$

Rule S1 simply makes the basic expressions of category A expressions of category A. Rule S2 combines noun phrases with verb phrases to make sentences, the side condition enforcing subject-verb agreement. Rule S3 combines determiners with common nouns to form noun phrases.

EXAMPLE 4. Here is a derivation for "every man walks":

$$F_2(every, man) = every\ man$$
$$F_1(every\ man, walk) = every\ man\ walks$$

Since the syntactic derivations in Montague grammar are very straightforward, we will dispense with them in the rest of this article.

We now define a function $\mapsto$ that associates with each expression of category A an expression of type $f(\mathsf{A})$, its *meaning*. First, the translation of most basic expressions will simply be a constant. We will denote the constants corresponding to basic expressions using CAPS. Thus, the constant corresponding to *walk* is WALK. The only exceptions to this rule are noun phrases, determiners, the verb *be*, and *necessarily*.

$T1$:

$$John \mapsto \lambda X({}^{\vee}X(j))$$
$$Mary \mapsto \lambda X({}^{\vee}X(m))$$
$$he_n \mapsto \lambda X({}^{\vee}X(x_n))$$
$$every \mapsto \lambda Y \lambda X \forall x({}^{\vee}Y(x) \rightarrow^{\vee} X(x))$$

$$\text{If } \alpha \in P_{\mathsf{S/IV}}, \beta \in P_{\mathsf{IV}}, \alpha \mapsto \alpha', \text{ and } \beta \mapsto \beta', \text{ then } F_1(\alpha,\beta) \mapsto \alpha'({}^{\wedge}\beta') \tag{T2}$$

$$\text{If } \alpha \in P_{(\mathsf{S/IV})/\mathsf{CN}}, \beta \in P_{\mathsf{CN}}, \alpha \mapsto \alpha', \text{ and } \beta \mapsto \beta', \text{ then } F_2(\alpha,\beta) \mapsto \alpha'({}^{\wedge}\beta') \tag{T3}$$

EXAMPLE 5. Here is the translation for "every man walks."

$$\dfrac{\dfrac{\dfrac{\dfrac{\lambda Y \lambda X \forall x({}^{\vee}Y(x) \rightarrow^{\vee} X(x)) \quad \text{MAN}}{\lambda X \forall x(\text{MAN}(x) \rightarrow^{\vee} X(x))}\, T2 \quad \text{WALK}}{\lambda X \forall x(\text{MAN}(x) \rightarrow^{\vee} X(x))({}^{\wedge}\text{WALK})}\, T3}{\forall x(\text{MAN}(x) \rightarrow^{\vee} ({}^{\wedge}\text{WALK}(x)))}\, \beta\text{-conversion}}{\forall x(\text{MAN}(x) \rightarrow \text{WALK}(x))}\, {}^{\vee}{}^{\wedge}\text{-cancellation}$$

Now we will consider transitive verbs. Again, we first give a syntactic rule, followed by a semantic rule:

$$\text{If } \alpha \in P_{\mathsf{IV}/(\mathsf{S}/\mathsf{IV})} \text{ and } \beta \in P_{\mathsf{S}/\mathsf{IV}}, \text{ then } F_6(\alpha,\beta) \in P_{\mathsf{IV}}, \text{ and } F_6(\alpha,\beta) = \alpha\beta' \text{ where } \beta' \tag{S7}$$

is the accusative form of β if β is a syntactic variable; otherwise $\beta' = \beta$.

$$\text{If } \alpha \in P_{\mathsf{S}/\mathsf{IV}}, \beta \in P_{\mathsf{IV}}, \alpha \mapsto \alpha', \text{ and } \beta \mapsto \beta', \text{ then } F_6(\alpha,\beta) \mapsto \alpha'(^\wedge\beta'). \tag{T7}$$

We will use the following two notational conventions from Gamut [32]. The first is just an instance of what computer scientists call *uncurrying*.

$$\text{If } \gamma \text{ is an expression of type } \langle a,\langle b,t\rangle\rangle, \alpha \text{ and expression of type } a, \text{ and } \beta \text{ an expression of type } b, \text{ then we may write } \gamma(\beta,\alpha) \text{ instead of } (\gamma(\alpha))(\beta). \tag{NC1}$$

Before discussing the second notational convention ($NC2$), we need to review how Montague proposed to treat transitive verbs. In Montague's system, the meanings transitive verbs are relations between individuals and second-order properties, i.e. they are of type $\langle\langle s,\langle\langle s,\langle e,t\rangle\rangle,t\rangle\rangle,\langle e,t\rangle\rangle$. Thus, since they are not relations between individuals, we can have a statement such as "John seeks a unicorn" be true, without it entailing the existence of unicorns. However, there are certain transitive verbs, so-called *extensional transitive verbs* which entail the existence of their arguments. For these expressions NC2 will allow us to move from Montague's higher-order interpretation of transitive verbs to relations between individuals.

$$\text{If } \delta \text{ is an expression of type } \langle\langle s,\langle\langle s,\langle e,t\rangle\rangle,t\rangle\rangle,\langle e,t\rangle\rangle, \text{ then we may write } \delta_* \text{ instead of } \lambda y\lambda x\delta(x,^\wedge\lambda X^\vee X(y)). \tag{NC2}$$

The expression δ_* refers to the relation that holds between x and y iff the relation δ holds between x and the intension of the set of all properties of y, i.e. $^\wedge\lambda X^\vee X(y)$. For further details and discussion, see Gamut [32].

EXAMPLE 6. Here is a derivation for "every man loves a woman."

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\lambda Y\lambda X\forall y(^\vee Y(y)\rightarrow^\vee X(y)) \quad \textsc{man} \qquad \dfrac{\textsc{love} \qquad \dfrac{\lambda Y\lambda X\exists x(^\vee Y(x)\rightarrow^\vee X(x)) \quad \textsc{woman}}{\lambda X\exists x(\textsc{woman}(x)\wedge^\vee X(x))}}{\textsc{love}(^\wedge\lambda X\exists x(\textsc{woman}(x)\wedge^\vee X(x)))}\,T7}{\lambda Y\forall y(\textsc{man}(y)\rightarrow^\vee Y(y))}}{\lambda Y\forall y(\textsc{man}(y)\rightarrow^\vee Y(y))(^\wedge\textsc{love}(^\wedge\lambda X\exists x(\textsc{woman}(x)\wedge^\vee X(x))))}\,T2}{\forall y(\textsc{man}(y)\rightarrow^\vee(^\wedge\textsc{love}(^\wedge\lambda X\exists x(\textsc{woman}(x)\wedge^\vee X(x))))(y))}\,\beta\text{-conversion}}{\forall y(\textsc{man}(y)\rightarrow(\textsc{love}(^\wedge\lambda X\exists x(\textsc{woman}(x)\wedge^\vee X(x))))(y))}\,{}^\vee{}^\wedge\text{-cancellation}}{\forall y(\textsc{man}(y)\rightarrow\textsc{love}(y,^\wedge\lambda X\exists x(\textsc{woman}(x)\wedge^\vee X(x)))}\,NC1}{\forall y(\textsc{man}(y)\rightarrow\exists x(\textsc{woman}(x)\wedge\textsc{love}_*(y,x)))}\,NC2$$

The following rule schema is used to create different derivations for ambiguous sen-

tences.

If $\alpha \in P_{\mathsf{S/IV}}$ and $\varphi \in P_{\mathsf{S}}$, then $F_{7,n}(\alpha, \varphi) = \varphi'$, where φ' is the result of the following substitution in φ : (i) if α is not a syntactic variable he_k, then replace the first occurrence of he_n or him_n with α, and the other occurrences of he_n or him_n with the appropriate anaphoric pronoun.;
(ii) if $\alpha = he_k$, then replace every occurrence of he_n and of him_n with him_k. (S8,n)

And the corresponding semantic rule:

$$\text{If } \alpha \in P_{\mathsf{S/IV}}, \varphi \in P_{\mathsf{S}}, \alpha \mapsto \alpha', \text{ and } \varphi \mapsto \varphi', \text{ then } F_{7,n}(\alpha, \varphi) \mapsto \alpha'(^\wedge \lambda x_n \varphi') \qquad \text{(T8,n)}$$

EXAMPLE 7. Here is another derivation of "every man loves a woman." This time, "everybody loves him$_1$" is derived first, and "a woman" is *quantified in*.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\lambda Y \lambda X \exists x(^\vee Y(x) \rightarrow^\vee X(x)) \quad \textsc{woman}}{\vdots} \qquad \textsc{love} \quad \dfrac{\lambda X^\vee X(x_1)}{\vdots}}{\lambda X \exists x(\textsc{woman}(x) \wedge^\vee X(x)) \qquad \forall y(\textsc{man}(y) \rightarrow \textsc{love}_*(y, x_1))}\, T8,1}{\lambda X \exists x(\textsc{woman}(x) \wedge^\vee X(x))(^\wedge \lambda x_1 \forall y(\textsc{man}(y) \rightarrow \textsc{love}_*(y, x_1)))}\, \beta\text{-conversion}}{\exists x(\textsc{woman}(x) \wedge^\vee (^\wedge \lambda x_1 \forall y(\textsc{man}(y) \rightarrow \textsc{love}_*(y, x_1))(x)))}\, {}^\vee{}^\wedge\text{-cancellation}}{\exists x(\textsc{woman}(x) \wedge (\lambda x_1 \forall y(\textsc{man}(y) \rightarrow \textsc{love}_*(y, x_1))(x)))}\, \beta\text{-conversion}}{\exists x(\textsc{woman}(x) \wedge (\forall y(\textsc{man}(y) \rightarrow \textsc{love}_*(y, x))))}$$

Now we can use quantifying-in to derive multiple interpretations of *believe*-sentences, corresponding to the above-mentioned *de dicto-de re* distinction. Assuming that the syntactic category of *believe* is $\mathsf{IV/S}$, we need new syntactic and semantic rules for such expressions:

$$\text{If } \alpha \in P_{\mathsf{IV/S}} \text{ and } \beta \in P_{\mathsf{S}}, \text{ then } F_{11}(\alpha, \beta) \in P_{\mathsf{IV}}, \text{ and } F_{11}(\alpha, \beta) = \alpha\beta \qquad \text{(S15)}$$
$$\text{If } \alpha \in P_{\mathsf{IV/S}}, \varphi \in P_{\mathsf{S}}, \alpha \mapsto \alpha', \text{ and } \varphi \mapsto \varphi', \text{ then } F_{11}(\alpha, \varphi) \mapsto \alpha'(^\wedge \varphi') \qquad \text{(T15)}$$

EXAMPLE 8. Here are derivations for *John believes that a man walks*. The first gives us the *de dicto* reading:

$$\dfrac{\lambda X(^\vee X(j)) \qquad \dfrac{\dfrac{\dfrac{\lambda Y \lambda X \exists x(^\vee Y(x) \wedge^\vee X(x)) \quad \textsc{man}}{\vdots}}{\lambda X \exists x(\textsc{man}(x) \wedge^\vee X(x))} \quad \textsc{walk}}{\vdots} \qquad \dfrac{\textsc{believe} \qquad \exists x(\textsc{man}(x) \wedge \textsc{walk}(x))}{\textsc{believe}(^\wedge \exists x(\textsc{man}(x) \wedge \textsc{walk}(x))}\, T15}{\vdots}}{\textsc{believe}(j, (^\wedge \exists x(\textsc{man}(x) \wedge \textsc{walk}(x))))}$$

While the quantifying-in version gives the *de re* reading:

$$\dfrac{\dfrac{\lambda Y\lambda X\exists x({}^{\vee}Y(x) \rightarrow^{\vee} X(x)) \quad \textsc{man} \qquad \lambda X^{\vee}X(x_1) \quad \textsc{walk}}{\lambda X\exists x(\textsc{man}(x) \wedge^{\vee} X(x)) \qquad \textsc{believe}(j, ({}^{\wedge}\textsc{walk}(x_1))}\, T8,1}{\lambda X\exists x(\textsc{man}(x) \wedge^{\vee} X(x))({}^{\wedge}\lambda x_1 \textsc{believe}(j, ({}^{\wedge}\textsc{walk}(x_1)))}$$

$$\exists x(\textsc{man}(x) \wedge \textsc{believe}(j, ({}^{\wedge}\textsc{walk}(x))))$$

In addition to epistemic modalities, we can give a treatment of of sentential adverbs, like *necessarily*, using the following two rules:

$$\text{If } \alpha \in P_{\mathsf{S}/\mathsf{S}} \text{ and } \beta \in P_{\mathsf{S}}, \text{ then } F_{11}(\alpha,\beta) \in P_{\mathsf{S}}, \text{ and } F_{11}(\alpha,\beta) = \alpha\beta \tag{S20}$$

$$\text{If } \alpha \in P_{\mathsf{S}/\mathsf{S}}, \varphi \in P_{\mathsf{S}}, \alpha \mapsto \alpha', \text{ and } \varphi \mapsto \varphi', \text{ then } F_{11}(\alpha,\varphi) \mapsto \alpha'({}^{\wedge}\varphi') \tag{T20}$$

However, we need to distinguish non-logical adverbs, like *rarely*, from logical ones, like *necessarily*, because we want the meaning of the former, but not the latter, to vary across models. This is accomplished through *meaning postulates*, an invention of Carnap's [14], which can be used to relate logical constants with expressions and also to relate expressions with each other, e.g. $\mathsf{bachelor}$ with $\neg\mathsf{married}$. The relevant meaning postulate for *necessarily* would be:

$$\forall p\Box(\textsc{necessarily}(p) \leftrightarrow \Box^{\vee}p)$$

We conclude this section with the example: *Necessarily, John walks.*

EXAMPLE 9.

$$\dfrac{\dfrac{\dfrac{\textsc{necessarily} \qquad \dfrac{\lambda X({}^{\vee}X(j)) \quad \textsc{walk}}{\textsc{walk}(j)}}{\textsc{necessarily}({}^{\wedge}\textsc{walk}(j))}\, MP}{\Box^{\vee}({}^{\wedge}\textsc{walk}(j))}\, T20}{\Box(\textsc{walk}(j))}\, {}^{\vee\wedge}\text{-cancellation}$$

The preceding examples are meant to be illustrative of the way that higher-order intensional logic is used in Montague grammar to model the meanings of natural language expressions. There are many shortcomings of Montague's framework, some formal and some empirical, the rectification of which can be seen as the motivation for the majority of the current approaches in semantic theory. See [24] and [75] for additional details of Montague grammar and for discussions of its shortcomings.

3 SYNTAX

3.1 Mathematical Linguistics

Mathematical linguistics is concerned with models of natural languages and linguistic theories and their formal properties, especially those theories about syntax. The properties of interest are typically those of theoretical computer science, particularly from

formal language theory, complexity theory, and learnability theory. Applications of logic to syntax combine the first two by giving tools to assess the complexity of formal languages descriptively. This line of research is related to finite model theory [25] and descriptive complexity theory [40].

We will review the basics of formal language theory as it applies to this setting. For a more detailed introduction see Partee, ter Meulen, and Wall [72], or, at a more advanced level and including many applications of modal logic, Kracht [54]. In order to model natural language syntax mathematically, we use strings. An *alphabet* is a finite set Σ of *symbols*, and a *string* over Σ is a finite sequence of elements of Σ. This includes the empty sequence, denoted by ε. A fundamental operation on strings is *string concatenation* which we will denote by juxtaposition. We denote the set of all strings over Σ by Σ^*, and the set of all non-empty strings by Σ^+. A *(formal) language* is a set of strings; i.e., a subset of Σ^*. In the intended applications, Σ is the set of words (or even morphemes) of a natural language (rather than just letters of some alphabet), however, for the purpose of examples we will frequently just use letters. Mathematical linguistics uses formal languages as models of natural languages: we identify English with the set of English sentences.

The purpose of grammatical theories is to distinguish the well-formed (grammatical) strings from the ill-formed (ungrammatical) strings. This can be achieved in a number of ways, using automata, grammars, algebras, or logic. However, it is an important assumption of linguistic theory, dating back at least to American Structuralism, that sentences of natural languages are not just linear sequences, but that they contain hierarchical structures: constituents. Furthermore, the division of a string into constituents plays an important part in semantics, since the principle of compositionality stipulates that the meaning of a string depends on the meaning of the words and the way they are put together, the latter of which can be captured by the constituent structure. While automata, algebras, and logics can be used to define languages, formal grammars play a central role in mathematical linguistics because they can associate a hierarchical structure with the strings that they generate: the derivation tree. Note that not all formal grammars can associate derivation trees with the strings they generate, but for linguistic applications, grammars that do are typically more interesting. Thus, we distinguish the *weak generative capacity*, the set of strings generated by a grammar, from the *strong generative capacity*, the set of structural descriptions or trees assigned by the grammar to the strings that it generates.

We will be referring frequently to a particular class of grammars and the languages they generate: the *context-free grammars* (CFGs) and *context-free languages* (CFLs). CFGs are specified in terms of two alphabets, Σ and Γ, which are called the *terminal* and *non-terminal* alphabets, respectively. The terminal alphabet consists of the symbols that make up the strings that the grammar generates; the non-terminal alphabet can be thought of as corresponding to the syntactic categories of traditional grammar. In addition, CFGs are specified in terms of *a finite set of rules*, P, and a distinguished member of Γ, *the start symbol*, denoted by S. The rules in P are of the form $A \to w$, where $w \in (\Sigma \cup \Gamma)^*$. CFGs derive strings of terminal symbols by successively rewriting non-terminal symbols. Let $x, y, z \in (\Sigma \cup \Gamma)^*$, and $A \in \Gamma$. We write $xAz \Rightarrow_G xyz$ to indicate that xyz can be obtained from xAz by using the rule $A \to y$ of G. We use $\Rightarrow_G^*$ to denote the reflexive, transitive closure of $\Rightarrow_G$. The language generated by a CFG, G,

denoted by $L(G)$, is defined as

$$L(G) = \{w \mid w \in \Sigma^*, S \Rightarrow_G^* w\}$$

A language, L, is called a CFL if there is a CFG, G, such that $L = L(G)$.

The CFLs play a central role in mathematical linguistics; they are in some sense a yardstick, because they approximate many natural language languages reasonably well [33] and they can be processed efficiently [39]. On the other hand, many formalisms are defined for the explicit purpose of extending the weak generative capacity of CFGs, that is to obtain non-CF languages. The reason for this is that there are natural language phenomena that are not context-free [89]. There are also some proposals which only go beyond CFGs in terms of strong generative capacity, although they are weakly equivalent to CFGs. That is, they are interested in obtaining sets of structures which go beyond the sets of parse trees of CF languages, but which generate CF string languages. Both the Lambek calculus [101] and regular tree languages (see below) are examples of such proposals.

3.2 Preliminary: Logics of Strings

The logic of strings was first studied by logicians interested in decidability [10]. It was continued within formal language theory, and had an algebraic slant [77, 92]. The case of strings has not found many applications to linguistics, as the more involved settings below have. However, this simpler case is useful in getting an intuition about the more complex cases. For a more detailed introductions to this area see Khoussainov and Nerode [45] and Thomas [98].

First of all, our intended models are what we shall call *string structures*. These are Kripke frames of the form

$$1 \longrightarrow 2 \longrightarrow \quad \cdots \quad \longrightarrow n$$

The idea is that a *string* on some alphabet gives rise to a frame as above. Adding a valuation amounts to specifying subsets of the model. Since we intend the atomic propositions to be the alphabet symbols, the subsets of the frame corresponding to these symbols correspond to the positions in the given word with the given symbols. Thus there is an extra condition that each world in the frame satisfy exactly one atomic sentence. A string model is a pair (W, v) consisting of a string structure W together with a valuation v that meets this extra condition. We usually omit the valuation from our notation. Let Σ be an alphabet, considered also as a set of atomic propositions for our modal language. A set of string models over Σ corresponds to a subset of Σ^+ of non-empty words over Σ. The correspondence associates to the string model (W, v) the string $W_1 \cdots W_n$, where n is the length of W and each W_i is the unique element of Σ satisfied by i in the model W.

Modal Logic of Strings

Figure 5 contains the basic modal logic of strings which we will call $\mathfrak{L}(\rightarrow, \rightarrow^*)$. The semantics in Figure 5 defines the relation $W, i \models \varphi$. We say that a string W satisfies a formula φ if $W, 1 \models \varphi$. A language L is *definable* in this (or another language) if there is a sentence φ so that L is exactly the set of strings satisfying a sentence φ.

Syntax	**Sentences** φ	$p_i \mid \neg\varphi \mid \varphi \wedge \psi \mid [\to]\varphi \mid [\to^*]\varphi$
Semantics	**Main Clauses**	$W, i \models [\to]\varphi$ iff $W, i+1 \models \varphi$
		$W, i \models [\to^*]\varphi$ iff for all $j \geq i, W, j \models \varphi$

Figure 5. Modal logic of strings: $\mathcal{L}(\to, \to^*)$

In order to study the languages definable in $\mathcal{L}(\to, \to^*)$, we introduce a class of languages, called the *star-free* languages (the reason for the name will become apparent later). The star-free languages, which are defined in Figure 6, were introduced by McNaughton and Pappert [66] to study first-order (FO) definable languages. It should be noted that star-free languages are sets of strings, and as such might well contain the empty string ε. Since we are going to be interested in classes of models which correspond to strings, and since the carrier sets our models must be non-empty, we are going to be interested in ε-*free* star languages as defined in Figure 7.

Syntax	**Expressions** r	$0 \mid 1 \mid a \mid rs \mid r+s \mid -r$
Semantics		$[\![0]\!] = \varnothing$
		$[\![1]\!] = \{\varepsilon\}$
		$[\![a]\!] = \{a\}$
		$[\![rs]\!] = \{st \mid s \in [\![r]\!], t \in [\![s]\!]\}$
		$[\![r+s]\!] = [\![r]\!] \cup [\![s]\!]$
		$[\![-r]\!] = \Sigma^* - [\![r]\!]$

Figure 6. The syntax and semantics of star-free expressions

Syntax	**Expressions** r	$0 \mid a \mid rs \mid r+s \mid -r$
Semantics	**Main Clause**	$[\![-r]\!] = \Sigma^+ - [\![r]\!]$

Figure 7. The syntax and semantics of ε-free star-free expressions

Here are some examples of formulas in $\mathcal{L}(\to, \to^*)$ and the star-free languages that correspond to them (taken from [19]). Notice that we are using *regular expressions* (see below) to describe these star-free languages because they are shorter; these languages can also be described with star-free expressions.

EXAMPLE 10. The language $(ab)^+$ is defined by the formula

$$a \wedge \langle\to^*\rangle(b \wedge \neg\langle\to\rangle a \wedge \neg\langle\to\rangle b) \wedge \neg\langle\to^*\rangle(a \wedge \langle\to\rangle a) \wedge \neg\langle\to^*\rangle(b \wedge \langle\to\rangle b)$$

Here and in the following, $\langle\cdot\rangle\varphi$ abbreviates $\neg[\cdot]\neg\varphi$. This formula says that the first letter is an a, the last letter is a b, and there are no consecutive a's or b's. Notice that $(aa)^+$ is not $\mathcal{L}(\to, \to^*)$ definable; in fact it is not even FO definable.

EXAMPLE 11. Let $A = \{a, b, c\}$. The language $A^*a(a+c)^*$ is defined by the formula

$$\langle\to^*\rangle(a \wedge \langle\to\rangle\neg\langle\to^*\rangle b)$$

Syntax	**Formulas** φ	$p_i \mid \neg\varphi \mid \varphi \wedge \psi \mid [\rightarrow]\varphi \mid [\rightarrow^*]\varphi \mid \mathcal{U}(\varphi,\psi)$
Semantics	**Main Clause**	$W,i \models \mathcal{U}(\varphi,\psi)$ iff there exists a $j \geq i$, such that $W,j \models \varphi$, and for all $n, i \leq n \leq j, W,n \models \psi$

Figure 8. Temporal logic of strings: PTL

The following proposition states the relationship between $\mathcal{L}(\rightarrow,\rightarrow^*)$ definability and star-freeness. There is also an algebraic characterization in Cohen *et al.* [19] which is omitted here.

PROPOSITION 12. *If $L \subseteq \Sigma^+$ is definable in $\mathcal{L}(\rightarrow,\rightarrow^*)$, then L is star-free.*

However, there are star-free languages that are not definable in $\mathcal{L}(\rightarrow,\rightarrow^*)$.

PROPOSITION 13. *The language $a^*b(a+b+c)^*$ is not $\mathcal{L}(\rightarrow,\rightarrow^*)$ definable.*

Proof. We use a version of Ehrenfeucht games for $\mathcal{L}(\rightarrow,\rightarrow^*)$ between word models W and V. The r-round game works exactly as in the standard game for modal logic on Kripke models. There are distinguished points in the two models, and they are updated in each round of a play. The difference from the standard games is that player I may decide at each round to play a standard move or else a $*$-*move*. In the $*$-move, I picks one of the two structures and moves the distinguished point (say w_i) to some w_j with $j \geq i$. Then II does the same in the other structure. If the distinguished points are labelled differently at any round, then I wins the play; otherwise II wins.

To show that $L = a^*b(a+b+c)^*$ is not definable, we show that for each r there are words w and v such that $w \in L$ and $v \notin L$, but player II has a winning-strategy in the r-round game on the string models corresponding to w and v. We take

$$\begin{array}{lcl} w & = & a^{r+1}b(a^rca^rb)^r \\ v & = & a(a^rca^rb)^r \end{array}$$

Let $n_i(b,W)$ be the number of b points strictly greater than the distinguished point in W at the end of round i; similarly for $n_i(b,V)$. The winning strategy for player II is to match a b with a b and a c with a c, and to maintain the assertion that either $n_i(b,W) = n_i(b,V)$, or else both numbers are at least $r-i$ (and similarly for c). ■

Temporal Logic of Strings

We can define all star-free languages if we add the temporal operator $\mathcal{U}$, called *until*. This logic, which we will call PTL, is defined in Figure 8. We can define the language from Proposition 13 in PTL.

EXAMPLE 14. The language $a^*b(a+b+c)^*$ is defined by the following formula

$$\mathcal{U}(b,a)$$

Thus, adding $\mathcal{U}$ gives us a more expressive language. In fact, Etessami and Wilke [26] have shown that there is an "until hierarchy," based on the nesting depth of $\mathcal{U}$. The following theorem characterizes the expressive power of PTL extending the classical characterization of temporal logic by Kamp [44].

THEOREM 15. *The following are equivalent for a language $L \subseteq \Sigma^+$:*

Syntax	**Expressions** r	$0 \mid a \mid rs \mid r + s \mid r^+$
Semantics	**Main Clauses**	$[\![r^+]\!] = \bigcup_{n>0} [\![r]\!]^n$, where $[\![r]\!]^1 = [\![r]\!]$ and $[\![r]\!]^{n+1} = [\![r]\!]^n [\![r]\!]$

Figure 9. The syntax and semantics of ε-free regular expressions

1. *L is FO definable (over the signature $<$ and monadic predicates corresponding to the alphabet letters).*
2. *L is definable in PTL.*
3. *L is star-free.*

Proof. The equivalence (1) iff (3) is due to McNaughton [66]. For an accessible proof, see [25]. The equivalence (1) iff (2) uses Gabbay's [31] separation method, but can also be proved algebraically [19]. See also [30]. ■

EXAMPLE 16. The language $(aa)^+$ is not PTL definable.

Proof. See [92]. ■

We will now consider an extension of the star-free languages, called the *regular languages*, defined in Figure 9.

Propositional Dynamic Logic of Strings

The regular languages were first logically characterized by Büchi who showed that they correspond to the languages definable in the monadic second order logic of strings (MSO). We will use propositional dynamic logic (PDL), defined in Figure 10, to characterize them. First, notice that $(aa)^+$ is definable:

EXAMPLE 17. The language $(aa)^+$ is defined by

$$a \wedge \langle \rightarrow \rangle a \wedge [\rightarrow; a?; \rightarrow; a?]^* \neg \langle \rightarrow \rangle \top$$

We will now define the automata theoretic model of the regular languages: finite automata. A *finite automaton* (FA) M is a structure $(\Sigma, Q, F, q_0, \Delta)$ where Σ is a alphabet, Q is a finite set of states, $F \subseteq Q$ is the set of final states, q_0 is the initial state, and Δ is a finite set of transition rules of the form $(q, a) \rightarrow p$ with $a \in \Sigma$ and $p, q \in Q$. We define the transition relation $\Rightarrow_M \subseteq (Q \times \Sigma^*) \times (Q \times \Sigma^*)$ inductively as follows:

$$(q, \varepsilon) \Rightarrow_M (q, \varepsilon)$$
$$(q, aw) \Rightarrow_M (p, w)$$

where $(q, a) \rightarrow p$ is a transition rule in Δ. We say that M *accepts* a string w if $(q_0, w) \Rightarrow_M^* (p, \varepsilon)$ where $p \in F$ and $\Rightarrow_M^*$ is the reflexive, transitive closure of $\Rightarrow_M$. Given an FA M, the language accepted by M, denoted by $L(M)$, is defined as

$$L(G) = \{w \mid (q_0, w) \Rightarrow_M^* (p, \varepsilon), p \in F\}$$

Syntax	**Formulas** φ	$p_i \mid \neg\varphi \mid \varphi \wedge \psi \mid [\pi]\varphi$
	Programs π	$\rightarrow \mid ?\varphi \mid \pi;\sigma \mid \pi \cup \sigma \mid \pi^*$
Semantics	**Main Clauses**	$W, i \models [\pi]\varphi$ iff for all j such that $(i,j) \in [\![\pi]\!]_W, W, j \models \varphi$
		$[\![?\varphi]\!]_W = \{(i,i) : i \in [\![\varphi]\!]_W\}$
		$[\![\pi;\sigma]\!]_W = [\![\pi]\!]_W; [\![\sigma]\!]_W$
		$[\![\pi \cup \sigma]\!]_W = [\![\pi]\!]_W \cup [\![\sigma]\!]_W$
		$[\![\pi^*]\!]_W = ([\![\pi]\!]_W)^*$

Figure 10. Propositional Dynamic Logic (PDL) on string models

One interesting observation, using the automata theoretic characterization of regular languages, is that they are closed under complementation (notice that "$-$" is not included in the definition of regular expressions).

THEOREM 18. *The following are equivalent for a language $L \subseteq \Sigma^+$:*

1. *L is definable in PDL.*
2. *L is definable in MSO.*
3. *L is regular.*
4. *L is accepted by a FA.*

Proof. The equivalence (2) iff (3) is Büchi's theorem. Again, see [25] for an accessible proof. The equivalence (3) iff (4) is known as Kleene's theorem, see e.g. [39]. For a proof of the equivalence (1) iff (3), see Kracht [53]. ■

Two interesting results that use algebraic proofs show that it is decidable whether a regular language is definable in $\mathcal{L}(\rightarrow, \rightarrow^*)$ or in PTL [19]. Since it is decidable for any regular language L whether $L = \Sigma^+$, it is decidable whether a formula in MSO is valid over string models [10].

Variations

Other non-modal logics that have been studied in the context of strings include monadic transitive closure (MTC) [3] and least fixed point (MLFP) logic [79], as well as logics with modular counting quantifiers [93]. The latter are of interest because they allow to extend FO logic so that $(aa)^+$ becomes definable without going to the full power of MSO. The logics MTC and MLFP define precisely the regular languages, since it is easy to see that PDL $\leq$ MTC $\leq$ LFP $\leq$ MSO. The equivalence then follows from theorem 18.

Extensions

The first proof that a natural language is not regular was given by Chomsky index-Chomsky, Noam [16, 17]. Thus, we would have to find stronger logics to describe natural languages within this framework. This is however not the line of research pursued (Rounds [85] being a notable exception), for two reasons. First, decidability of the logical

formalism employed is of some importance, as this line of research ultimately aims to contribute to computational linguistics. However, trying to find *decidable* extensions of PDL or MSO is quite challenging (a point we will revisit later). There exists a characterization of the CFLs in terms of an extension of MSO [58], allowing quantification over special kinds of binary relations, so-called "matchings". Even though the question whether this logic is decidable is not addressed there, validity is undecidable for any logic characterizing CFLs, since the questions whether $L(G) = \Sigma^+$ is undecidable for CFGs. Such logics are also bound to be odd, since CFLs are not closed under complementation. The second problem is that using a logics for *strings* does not give a notion of strong generative capacity: the process of verifying that a formula is true of a string does not assign a structure to that string, as the process of deriving a string using a grammar does.

Digression

One point which should be of interests to modal logicians concerns the sense in which *words* are like *(modal) sentences*. The analogy is neatly captured in coalgebra (see Chapter 6), especially in studies pertaining to *coalgebraic logic*. We are considering several functors on the category of sets. Here $\mathcal{P}$ is the power set functor, and $\mathcal{P}_f$ is the finite power set functor. $AtProp$ is a set of atomic propositions, and A is an alphabet. We have the following analogies,

	Kripke semantics	automata
Functor $F(x)$ on sets	$\mathcal{P}(x) \times \mathcal{P}(AtProp)$	deterministic : $x^A \times \{0,1\}$ non-deterministic : $\mathcal{P}_f(x)^A \times \{0,1\}$
coalgebra	Kripke model	deterministic automaton
final coalgebra	canonical model	regular languages
notion of equivalence	bisimulation	bisimulation

Kripke models, deterministic, and non-deterministic automata are described as coalgebras of the given functors. In both cases, the elements of the carrier of the coalgebra may be thought of as *states*. (However, coalgebras do not include specified "real worlds" or "start states".) In the case of the automata, the state sets might be infinite, but our use of $\mathcal{P}_f$ insures that they will be finitely branching. The set $\{0,1\}$ in the automata functors is there to equip the state set with *accepting* and *non-accepting* states. The final coalgebra in each case turns out to be an important mathematical object, and the reader can see the sense in which the canonical model is the analog of the regular languages. Indeed, modal sentences might be thought of as the record of "possible observations" on Kripke models in the same way that words are on automata. Finally, from coalgebra we have a very general notion of equivalence, the coalgebraic bisimulation. The special cases of this are bisimulation of Kripke models and also the bisimulation of automata (the largest such is the Myhill-Nerode equivalence relation). See Rutten [86] for more information on the connection between automata and coalgebra.

3.3 Logics of Trees

Since extending the logic of strings to capture more complex string languages than the regular languages often leads to undecidability, one approach to extending the coverage

of our logic is to describe more complex structures: move from strings to trees. Thus, the Kripke structures we will be considering are trees, and the logics will contain more complicated modalities to describe trees. One immediate advantage of this approach for linguistic purposes is that these logics will automatically be connected to strong generative capacity, since they describe sets of trees. One disadvantage is that the *recognition* or *parsing* problem, which in the string case just amounts to model checking, now involves satisfiability checking (see below).

The extension of the descriptive approach to trees was originally also motivated by decidability questions [97]. Even though the connections to CFLs were pointed out by Thatcher [95], this line of research did not find applications in linguistics until the development of constraint based grammar formalisms which replaced the derivational approach to natural language syntax. The work of Rogers [83], Kracht [54], and others provided formal models for these constraint based grammar formalisms and established formal language theoretic results for them at the same time.

As mentioned above our Kripke structures will now be trees. We will use the concept of tree domains [35] to define such Kripke structures. A (finite, binary) *tree domain*, T, is a finite subset of $\{0,1\}^*$, such that for all $u, v \in \{0,1\}^*$

1. if $uv \in T$, then $u \in T$, and
2. if $u1 \in T$, then $u0 \in T$.

A string in T describes a path from the root to a node, where 0 means "go left" and 1 means "go right". We identify nodes with the path leading to them. Thus, ε is the root. The first condition above says that if there is a path to a node, then there is a path to any node above it (this is called prefix closure). The second condition says that if a node has a right daughter, then it has a left daughter (called left sibling closure).

The main relations between nodes in a tree that are of interest in linguistics are domination and linear precedence. We say that that a node $u \in T$ *dominates* a node $v \in T$ if for some $w \in \{0,1\}^*$, $v = uw$. A special case of domination is the parent-of relation, defined by: u is the parent of v if $v = u0$ or $v = u1$. We say that u *linearly precedes* v if for some $x, y, z \in \{0,1\}^*$, $u = x0y$ and $v = x1z$. Following Rogers [83], we will denote the domination relation by $\lhd^*$, the parent-of relation by $\lhd$, and linear precedence by $\prec$. Thus, our Kripke frames will be variations of the form $(T, \lhd, \lhd^*, \prec)$, where T is a tree domain.

Regular tree languages

In order to generalize from strings to labeled trees, we will now consider *ranked alphabets* in which each symbol has an arity or rank. For surveys of tree languages see Gécseg and Steinby [34] or Thatcher [96]. Let Σ be a ranked alphabet. We will denote the set of n-ary symbols in Σ by Σ_n. The set of terms over Σ is denoted by T_Σ. A subset of T_Σ is called a *tree language*.

In a number of settings, trees are considered to be labeled with boolean features, rather than with ranked symbols. We note that these two approaches commensurable using the following representation. Given a finite set of boolean features $F = \{f_1, \ldots, f_n\}$, the *binary ranked alphabet based on* F, Σ^F, is defined as

$$\Sigma^F = \{f_1, \neg f_1\} \times \cdots \times \{f_n, \neg f_n\} \times \{0, 2\}$$

where each $f_i, \neg f_i$ represents whether or not a feature holds at a given node and 0 or 1 represent the aritiy of the symbol. Thus, $(f_1, \neg f_2, 0)$ would be a leaf symbol, and $(f_1, \neg f_2, 2)$ would be an internal node symbol. The previous definition can be easily generalized to trees of any arity.

The *yield* of a tree, t, is the string over Σ_0 which is obtained by concatenating the symbols at the leaves of t from left to right, or more formally:

$$yield(c) = c, \text{ for } c \in \Sigma_0$$
$$yield(f(t_1, \ldots, t_n)) = yield(t_1) \ldots yield(t_n), \text{ for } f \in \Sigma_n$$

A (bottom-up, non-deterministic) *finite tree automaton* (FTA) M is a structure of the form (Σ, Q, F, Δ) where Σ is a ranked alphabet, Q is a finite set of states, $F \subseteq Q$ is the set of final states, and Δ is a finite set of transition rules of the form $f(q_1, \ldots, q_n) \to q$ with $f \in \Sigma_n$ and $q, q_1, \ldots, q_n \in Q$. An FTA is *deterministic* if there are no two transition rules with the same left-hand-side. It can be shown that the bottom-up variety of finite tree automata can be determinized, while the top-down variety cannot.

A *context* s is a term over $\Sigma \cup \{x\}$ containing the zero-ary term x exactly once. We write $s[x \mapsto t]$ for the term that results from substituting x in s with t. Given a finite tree automaton $M = (\Sigma, Q, F, \Delta)$ the derivation relation $\Rightarrow_M \subseteq T_{Q \cup \Sigma} \times T_{Q \cup \Sigma}$ is defined by $t \Rightarrow_M t'$ if for some context $s \in T_{\Sigma \cup Q \cup \{x\}}$ there is a rule $f(q_1, \ldots, q_n) \to q$ in Δ, and

$$t = s[x \mapsto f(q_1, \ldots, q_n)]$$
$$t' = s[x \mapsto q]$$

We use $\Rightarrow_M^*$ to denote the reflexive, transitive closure of $\Rightarrow_M$. A finite automaton M *accepts* a term $t \in T_\Sigma$ if $t \Rightarrow_M^* q$ for some $q \in F$. The *tree language accepted* by a finite tree automaton M, $L(M)$, is

$$L(M) = \{t \in T_\Sigma \mid t \Rightarrow_M^* q, \text{ for some } q \in F\}.$$

A tree language, L, is *regular* if $L = L(M)$ for some FTA M.

The following example is concerned with the Circuit Value Problem (CVP), in which the trees labeled with boolean functions are evaluated. It is interesting to note that a number of separation results of logically defined tree languages use trees labeled with boolean functions [79].

EXAMPLE 19. Let $\Sigma = \{\wedge, \vee, 0, 1\}$. The tree language $CVP \subseteq T_\Sigma$ such that each tree in CVP evaluates to true can be accepted by the following FTA, $M = (\Sigma, Q, F, \Delta)$, where

$$Q = \{t, f\}$$
$$F = \{t\}$$

and

$$\Delta = \left\{ \begin{array}{ll} 0 \to f, & 1 \to t, \\ \wedge(t,t) \to t, & \wedge(t,f) \to f, \\ \wedge(f,t) \to f, & \wedge(f,f) \to f, \\ \vee(t,t) \to t, & \vee(t,f) \to t, \\ \vee(f,t) \to t, & \vee(f,f) \to f \end{array} \right\}$$

Given a finite sets of feature $F = \{f_1, \ldots, f_n\}$ and a feature $f_i \in F$, we define the *projection*, π, that eliminates f_i in the natural way:

$$\pi : \Sigma^F \rightarrow \Sigma^{F-\{f_i\}}$$

This definition can be extended to arbitrary subsets $G \subseteq F$, where

$$\pi : \Sigma^F \rightarrow \Sigma^{F-G}$$

Given a projection $\pi : \Sigma^F \rightarrow \Sigma^{F-G}$, we extend π to a tree homomorphism $\hat{\pi} : T_{\Sigma^F} \rightarrow T_{\Sigma^{F-G}}$ as follows:

$$\hat{\pi}(c) = \pi(c)$$
$$\hat{\pi}(f(t_1, \ldots, t_n)) = \pi(f)(\hat{\pi}(t_1), \ldots, \hat{\pi}(t_n))$$

with $c \in \Sigma_0$ and $f \in \Sigma_n, n > 0$. For a tree language L, we define $\hat{\pi}(L) = \{\hat{\pi}(t) \mid t \in L\}$.

We will consider the relationship between regular tree languages and the derivation trees of CFGs.

PROPOSITION 20. *(Thatcher [95]) If $L \subseteq T_\Sigma$ is a regular tree language, then*

$$\{yield(t) \mid t \in L\}$$

is a CFL.

While the yields of regular tree languages are CFLs, regular tree languages are more complex than the derivation trees of CFG. In order to compare the regular tree languages to the derivation trees of CFGs, we formalize the latter using the local tree languages.

The *fork* of a tree t, $fork(t)$, is defined by

$$fork(c) = \varnothing$$
$$fork(f(t_1, \cdots, t_n)) = \{(f, root(t_1), \ldots, root(t_n))\} \cup \bigcup_{i=1}^{n} fork(t_i)$$

with $c \in \Sigma_0$, $f \in \Sigma_n, n > 0$, and *root* being the function that returns the symbol at the root of its argument. For a tree language L, we define

$$fork(L) = \bigcup_{t \in L} fork(t)$$

The intuition behind the definition of *fork* is that an element of $fork(T_\Sigma)$ corresponds to a rewrite rule of a CFG. Note that $fork(T_\Sigma)$ is always finite, since Σ is finite.

A tree language $L \subseteq T_\Sigma$ is *local* if there are sets $R \subseteq \Sigma$ and $E \subseteq fork(T_\Sigma)$, such that, for all $t \in T_\Sigma, t \in L$ iff $root(t) \in R$ and $fork(t) \subseteq E$.

We quote without proof the following two theorems by Thatcher [95].

THEOREM 21. *[95] A tree language is a set of derivation trees of some CFG iff it is local.*

THEOREM 22. *[95] Every local tree language is regular.*

While there are regular tree languages that are not local, the following theorem, also due to [95], demonstrates that we can obtain the regular tree languages from the local

Syntax	**Formulas** φ	$p_i \mid \neg\varphi \mid \varphi \wedge \psi \mid [\pi]\varphi$
	Programs π	$\rightarrow \mid \leftarrow \mid \uparrow \mid \downarrow \mid \pi^*$
Semantics	**Main Clauses**	$[\![\rightarrow]\!]_T = \{(u0, u1) \mid u1 \in T\}$
		$[\![\leftarrow]\!]_T = \{(u1, u0) \mid u1 \in T\}$
		$[\![\downarrow]\!]_T = \{(u, ui) \mid i \in \{0,1\}, ui \in T\}$
		$[\![\uparrow]\!]_T = \{(ui, u) \mid i \in \{0,1\}, ui \in T\}$

Figure 11. Modal logic of trees: $\mathcal{L}_{core}$

tree languages via projections. We will review the main points of the proof, because we will use some of its details later on.

THEOREM 23. *[95] For every regular tree language L, there is a local tree language L' and a one-to-one projection π, such that $L = \hat{\pi}(L')$.*

Proof. Let L be a regular tree language. Assume that L is accepted by the deterministic FTA $M = (\Sigma, Q, F, \Delta)$. We define L' terms of R and E as follows: $R = \Sigma \times F$ and

$$E = \{((f, q), (f_1, q_1), \ldots, (f_n, q_n)) \mid f(q_1, \ldots, q_n) \rightarrow q \in \Delta, f_1, \ldots, f_n \in \Sigma\}$$

We then define $L' = \{t \in T_{\Sigma \times Q} \mid root(t) \in R, fork(t) \subseteq E\}$. Notice that the trees in L' encode runs of M. The tree homomorphisms $\hat{\pi}$ based on the projection $\pi : \Sigma \times Q \rightarrow \Sigma$ maps L' to L as can be easily verified.

It should be noted that, since M is deterministic, there exists exactly one accepting run for each tree in $L(M)$ and thus the homomorphism $\hat{\pi} : L' \rightarrow L$ is one-to-one. ■

This rather technical result is of some importance in the context of linguistic application, for it implies that we can use frameworks of lower complexity to describe the same structures as a more complex framework *if we use more complex categories or features.* Since we can also add new categories as names for the more complex ones, we can use a less complex framework to describe the same structures as a more complex framework by adding more categories. Thus, parsimony would seem to imply that we should always use the simpler framework. However from the point of linguistics, the use of complex or additional features needs to be justified. To further elaborate on the previous point, we will have to keep in mind that all of the logics we will consider can define the local tree languages and all the languages they can define are regular. Thus undefinability will always mean undefinability over a fixed finite set of propositional variables, since we can always define a regular, undefinable tree language by using more features.

The basic modal logic of trees: $\mathcal{L}_{core}$

To the best of our knowledge, the first explicit use of modal logic to define tree languages can be found in [7]. Two variations of this logic were considered in [8, 9], of which we will consider the latter. The basic modal logic of trees, $\mathcal{L}_{core}$, is defined in Figure 11. Again, we say that a tree T satisfies a formula φ if $T, \varepsilon \models \varphi$. A language L is *definable* in this (or another language) if there is a sentence φ so that L is exactly the set of trees satisfying a sentence φ.

The following proposition establishes that $\mathcal{L}_{core}$ is expressive enough to define any binary branching, local tree language. The restriction to binary branching is only due to the fact that we defined our tree domains to be binary branching.

PROPOSITION 24. *Let $L \subseteq T_\Sigma$ be a local tree language. There is a sentence φ_G in $\mathcal{L}_{core}$ that defines L.*

Proof. By Theorem 21, there is a CFG G such that L is equal to the derivation trees of G. Let $G = (\Sigma, \Gamma, P, S)$. Since we are only considering binary branching trees, every rule in P is of the form $A \to BC$ or $A \to a$ with $A, B, C \in \Gamma$ and $a \in \Sigma$. We can simply encode the rules directly in our logic:

$$A \to \bigvee_{A \to BC \in P} \langle\downarrow\rangle(B \wedge \langle\rightarrow\rangle C)$$

and

$$A \to \bigvee_{A \to a \in P} (\langle\downarrow\rangle a)$$

This ensures that the models of φ_G are parse trees of G. However, we further need to ensure only the parse trees of G model φ_G. So, we need to express that each node makes exactly one symbol true:

$$[\downarrow^*](\bigvee_{a \in (\Sigma \cup \Gamma)} a \wedge \bigwedge_{a \neq b} (\neg a \vee \neg b))$$

that the start symbol of the grammar is true at the root: S, that the terminal symbols are true at the leaves:

$$[\downarrow^*](\bigvee_{a \in \Sigma} a \to \neg\langle\downarrow\rangle\top)$$

and that the non-terminal symbols are true at the internal nodes

$$[\downarrow^*](\bigvee_{A \in \Gamma} A \to \langle\downarrow\rangle\top)$$

■

As is observed by Blackburn and Meyer-Viol, this translation of a CFG into logical formulas brings with it a change in perspective. Instead of a *procedural* or *derivational* perspective that considers CFG rules to be rewrite rules, we move to a *declarative* or *descriptive* perspective that considers CFG rules to be *constraints*. This change in perspective is the main motivation for the application of logic in syntax, because of a similar change in perspective that occurred in a number of grammar formalisms proposed by linguists in the 1980s, most notably Chomsky's "Government and Binding" (GB) [18] and Gazdar, Klein, Pullum, and Sag's "Generalized Phrase Structure Grammar" (GPSG) [33].

ID/LP Grammars

The rules of a CFG encode two kinds of information: the categories of a node and its children, and the order in which the categories of the children occur. Thus, a rule of the form $A \to BC$ tells us that a node labeled A can have two children, one labeled

Syntax	$p_i \mid \neg\varphi \mid \varphi \wedge \psi \mid \mathcal{U}_{\rightarrow}(\varphi,\psi) \mid \mathcal{U}_{\leftarrow}(\varphi,\psi) \mid \mathcal{U}_{\uparrow}(\varphi,\psi) \mid \mathcal{U}_{\downarrow}(\varphi,\psi)$
Semantics	$T,u \models \mathcal{U}_{\downarrow}(\varphi,\psi)$ iff there exists a v such that $u \lhd^* v$, $T,v \models \varphi$, and for all w such that $u \lhd^* w \lhd^* v$, $T,w \models \psi$

Figure 12. Temporal logic of trees: $\mathfrak{X}_{until}$ (only one clause in the semantics)

B, the other C, and that the node labeled B precedes the node labeled C. Linguists have observed that separating these two notions can lead to more compact grammars. Thus, ID/LP grammars have been proposed that consist of *unordered* rewrite (immediate dominance or ID) rules, $A \rightarrow B, C$, and *linear precedence* (LP) rules, $B < C$. Linear precedence rules only apply to sisters, which is why we used $<$ rather than $\prec$ which applies to arbitrary nodes.

ID/LP grammars can be very naturally expressed in $\mathfrak{L}_{core}$; in fact ID/LP grammars are, in some sense, a very limited logic for trees. See Gazdar et al. [33] or Shieber [88] for applications and detailed examinations of ID/LP grammars.

Variations of $\mathfrak{L}_{core}$

Two additional basic modal logics of trees have been considered by Blackburn and associates [7, 8]. The first includes the connectives $\varphi \Rightarrow \psi$ and $\bullet(\varphi_1, \ldots \varphi_n)$. The latter is used in the context of trees with n children, so we will only consider the case where n is 2. Their semantics are given by $T,v \models \varphi \Rightarrow \psi$ iff for all u, $T,u \models \varphi \rightarrow \psi$, and $T,v \models \bullet(\varphi,\psi)$ iff $T,u0 \models \varphi$ and $T,u1 \models \psi$. Notice that the purpose of $\bullet$ is to combine immediate dominance and linear precedence into one connective.

Blackburn and Meyer-Viol [8] define a modal logic of trees that differs from $\mathfrak{L}_{core}$ in that it contains modalities for the left and right daughter: $\downarrow_1, \downarrow_2$.

Temporal Logic of Trees

We now move on to an extension of $\mathfrak{L}_{core}$, temporal logic. The syntax and semantics of propositional tense logic on trees, $\mathfrak{X}_{until}$, is defined in Figure 12. The main application of $\mathfrak{X}_{until}$ was given by Palm [70], though with a different formulation which we will consider below. We follow here the formulation of Marx [63], because it lends itself to a more direct proof of equivalence with FO.

THEOREM 25. *[63] The following are equivalent for a tree language $L \subseteq T_\Sigma$:*

1. *L is FO definable.*

2. *L is definable in $\mathfrak{X}_{until}$.*

While the notion of regular expressions can be generalized to trees, the correspondence between star-free expressions and FO (or $\mathfrak{X}_{until}$) definability breaks down at this level. In fact, Thomas and Potthoff [81] showed that every regular language that does not contain unary branching symbols is star-free. The question whether FO definability of regular tree language is decidable is still open.

Syntax	**Formulas** φ	$p_i \mid \neg\varphi \mid \varphi \wedge \psi \mid [\pi]\varphi$
	Programs π	$\rightarrow \mid \leftarrow \mid \uparrow \mid \downarrow \mid \pi_\varphi \mid \pi^*$
Semantics	**Main Clauses**	$[\![\pi_\varphi]\!]_T = \{(u,v) \mid (u,v) \in [\![\pi]\!]_T, T, u \models \varphi\}$

Figure 13. Conditional path logic of trees: $\mathcal{L}_{cp}$

Variations of $\mathcal{X}_{until}$

As was mentioned above, Palm's [70] application of $\mathcal{X}_{until}$ was carried out using a different formulation which he called propositional tense logic and which Afanasiev et al. [1] called conditional path logic, $\mathcal{L}_{cp}$. The syntax and semantics of $\mathcal{L}_{cp}$ are defined in Figure 13.

X-bar theory

As was mentioned above, which non-terminals are used in a natural language grammar matters to linguists. The point again is that the label assigned to a node in a tree signifies the grammatical category of the constituent it dominates. One theory of the organization of non-terminals and their rules is X-bar theory, which provides the foundation for a variety of grammar formalisms, including GB and GPSG. There are many variations of X-bar theory, so the particular formulation discussed here may not agree with those found in other places.

In terms of the organization of the non-terminals of a grammar, X-bar theory stipulates that there is a finite set of *lexical categories*, like $N(oun)$, $V(erb)$, $P(reposition)$, $A(djective)$, $Adv(erb)$, corresponding to the parts of speech, and that all other non-terminals are *projections* of the lexical categories. The idea of a projection is best motivated by the following example. The constituent *tall man* consists of two words, a noun and an adjective. When considering what the category of the constituent should be, we should take into account that *tall man* behaves more like a noun than like an adjective, which can be verified by substituting *tall man* for a noun in a sentence, preserving grammaticality, and substituting it for an adjective in a sentence, not preserving grammaticality. Thus, the category of *tall man* should be derived from the category of *man*. The category that X-bar theory assigns to the phrase is called N' (pronounced N-bar). N' is a projection of N. While X-bar theory within GB considered N and N' as atomic categories, the idea that the bar-level of a node is a syntactic feature is due to GPSG.

While there are various proposal for X-bar theory, we will assume that all rules of an X-bar grammar should be of the form

$$X'' \rightarrow X', Y'' \tag{11}$$

$$X' \rightarrow X', Y'' \tag{12}$$

$$X' \rightarrow X, Y'' \tag{13}$$

The non-terminal Y'' has different roles in the three rule schemata, each of which has a name in X-bar theory. In rule schema 11, Y'' is called the *specifier*; in rule schema 12, it is called the *adjunct*, and in rule schema 13, it is called the *complement*. In each of the rules, the X or X' on the right hand side is called the *head*.

It has been observed in a variety of contexts [48, 51, 70] that it is desirable to dispense with the bar-feature and to define the constraints posed by the X-bar schemata in terms

Syntax	**Formulas** φ	$p_i \mid \neg\varphi \mid \varphi \wedge \psi \mid [\pi]\varphi$
	Programs π	$\rightarrow \mid \leftarrow \mid \uparrow \mid \downarrow \mid ?\varphi \mid \pi;\sigma \mid \pi \cup \sigma \mid \pi^*$

Figure 14. Dynamic logic of trees

of projections. Thus, we would like to define a constraint that states that every node has a path to a leaf such that the node, the leaf, and all the nodes on the path have the same lexical features. This can expressed in $\mathcal{L}_{cp}$ as follows. First, we state that a feature φ belongs to a head:

$$hd\ \varphi \equiv \varphi \wedge head$$

Then, we state that a feature φ is projected from a leaf:

$$proj\ \varphi \equiv \langle \downarrow^*_{hd\ \varphi} \rangle (hd\ \varphi \wedge leaf)$$

Finally, we can restate the X-bar convention by requiring every node to be a projection, given a finite set of lexical features Lex:

$$[\downarrow^*](\bigvee_{\varphi \in Lex} proj\ \varphi)$$

Notice that we would need a feature to indicate that a node is the head in case two siblings share the same lexical feature. Furthermore, there are certain regularities that this head feature has to observe, such as that no two sisters may both be heads:

$$[\downarrow^*](head \rightarrow \neg(\langle\leftarrow\rangle head \vee \langle\rightarrow\rangle head))$$

Dynamic Logic of Trees

The first descriptive characterization of the regular tree languages was obtained by Doner [23], and Thatcher and Wright [97]. They generalized Büchi's theorem to trees.

THEOREM 26. *The following are equivalent for a tree language $L \subseteq T_\Sigma$:*

1. *L is regular.*
2. *L is definable in MSO.*

Kracht [49] introduced PDL on trees in the context of model theoretic syntax.

While the correspondence between $\mathfrak{X}_{until}$ and FO continues to hold in the generalization from strings to trees, the same is not true for the correspondence between PDL and MSO on strings, as was shown by Kracht, a topic we shall investigate in detail in the next section.

Undefinability: Inessential Features

The relationships between the three logics discussed above are well-understood, in that $\mathcal{L}_{core}$ is properly included in $\mathfrak{X}_{until}$, which is properly included in PDL, which in turn is properly included in MSO. There is a central property that can be used to describe the

languages that can be defined in one logic, but not in another. This property was first introduced by Kracht [50] and it is defined in terms of *inessential features*.

Let F be a finite set of features, $G \subseteq F$, $L \subseteq T_{\Sigma^F}$, and $\pi : \Sigma^F \to \Sigma^{F-G}$ be a projection. We call the features in G *inessential for* L if the homomorphism $\hat{\pi} : L \to T_{\Sigma^{F-G}}$ based on π is one-to-one. The intuition for this definition of inessential features is that no two trees in L can be distinguished using features in G. Thus, given a tree t in $\hat{\pi}(L)$, we can recover the features from G in t using $\hat{\pi}^{-1}$, since $\hat{\pi}$ is one-to-one.

EXAMPLE 27. The bar feature of the version of X-bar theory sketched above is inessential. To see that, notice that there is only one head (bar-level 0) which has a maximal projection (bar-level 2) and all projections in between are of bar-level 1.

While being an inessential feature is defined with respect to a language, being eliminable is defined with respect to a logic and a language. Let F be a finite set of features, $G \subseteq F$, $L \subseteq T_{\Sigma^F}$, $\pi : \Sigma^F \to \Sigma^{F-G}$ be a projection, and $\mathfrak{L}$ be a logic. Suppose that L is definable in $\mathfrak{L}^F$. We say that G is *eliminable in* $\mathfrak{L}$ *for* L if $\hat{\pi}(L)$ is definable in $\mathfrak{L}^{F-G}$.

It should be noted that this definition of eliminability does not coincide with Kracht's [50], who defines eliminable as being globally explicitly definable. Kracht's definition implies the definition used here, and thus is stronger. However, since we are interested in *ineliminability*, by contraposition, the definition employed here implies Kracht's definition of ineliminability.

The following, well-known, inclusions follow primarily from the definition of the three modal logics.

THEOREM 28. $\mathfrak{L}_{core} \leq \mathfrak{L}_{cp} \leq PDL_{tree} \leq MSO$

Proof. The first two inclusions follow from the definitions of these logics. The third inclusion follows from the fact that transitive closure is MSO-definable. ■

Next, we consider strictness of these inclusions.

PROPOSITION 29. *[87] Let $F = \{a, b\}$. The tree language $L_1 \subseteq T_{\Sigma^F}$ such that each tree in L_1 contains a path from the root to a leaf at which exactly one a holds is not $\mathfrak{L}_{core}$-definable, but is $\mathfrak{L}_{cp}$-definable.*

PROPOSITION 30. *Let $\Sigma = \{\wedge, \vee, 0, 1\}$. The tree language $CVP \subseteq T_\Sigma$ such that each tree in CVP evaluates to true is not $\mathfrak{L}_{cp}$-definable, but is PDL_{tree}-definable.*

Proof. Potthoff [80] showed that CVP is not definable in an extension of first-order logic with modular counting quantifiers, and since $\mathfrak{L}_{cp}$ is equivalent to first-order logic on trees [1], the undefinability follows. That CVP is definable in PDL_{tree} is shown in [1]. ■

PROPOSITION 31. *[52, 53] Let $F = \{p, q\}$. Let $L_2 \subseteq T_{\Sigma^F}$ where each tree in L is a ternary branching tree such that p is true along a binary branching subtree and q is true at all leaves at which p is true. The language $L_3 \subseteq T_{\Sigma^{\{q\}}}$ obtained from the projection that eliminates p is not PDL_{tree}-definable, but is MSO-definable.*

These three propositions demonstrate the strictness of the inclusion of the three modal logics and MSO. Next, we will consider how languages that are undefinable in one of these logics can be defined with additional features.

THEOREM 32. *[102] There exists a set of features F, a tree language $L \subseteq T_{\Sigma^F}$, and a subset $G \subseteq F$, such that G is ineliminable in $\mathfrak{L}_{core}$ (resp. $\mathfrak{L}_{cp}$) but eliminable in $\mathfrak{L}_{cp}$ (resp. PDL_{tree}).*

Proof. Both of these constructions work the same way. Given two of our logics $\mathfrak{L}_1, \mathfrak{L}_2$, with $\mathfrak{L}_1 < \mathfrak{L}_2$, pick a tree language, L, that is not definable in $\mathfrak{L}_1$ but is definable in $\mathfrak{L}_2$, which exists by propositions 29 and 30.

By Theorem 28, we know that L is regular, and by Theorem 24, we know that any local tree language is definable in $\mathfrak{L}_1$. Given a deterministic FTA $M = (\Sigma, Q, F, \Delta)$, with $L = L(M)$, we can use theorem 23 to construct a local tree language $L' \subseteq T_{\Sigma \times Q}$ such that $\hat{\pi}(L') = L$. Now, the features in Q are inessential, since M is deterministic, but ineliminable, since L is undefinable in $\mathfrak{L}_1$. However, since L is definable in $\mathfrak{L}_2$, the features in Q are eliminable in $\mathfrak{L}_2$. ■

The previous theorem can be strengthened in that it can be used to *characterize* the tree languages that are undefinable in some logic $\mathfrak{L}_1$ but definable in some other logic $\mathfrak{L}_2$, with $\mathfrak{L}_1 \leq \mathfrak{L}_2$.

THEOREM 33. *[102] Any tree language that is not definable in $\mathfrak{L}_{core}$ (resp. $\mathfrak{L}_{cp}$) but is definable in $\mathfrak{L}_{cp}$ (resp. PDL_{tree}) can be defined with additional, inessential features in $\mathfrak{L}_{core}$ (resp. $\mathfrak{L}_{cp}$) that are not eliminable in $\mathfrak{L}_{core}$ (resp. $\mathfrak{L}_{cp}$).*

Model Theoretic Syntax and Parsing

Recall that we generalized from strings to trees because we wanted to retain decidability and because we wanted to have a formalism that associates grammatical structure to an unstructured string. While decidability has been retained by this move, we need to say a little bit about how model theoretic syntax associates structures with strings. It should be noted that CFGs are formalisms that generate strings and that the structures that they assign to the strings arise in the process of generating the string, i.e. that trees are not a primary but a derived notion for formal grammars, Tree Adjoining Grammars being a notable exception. It should also be noted that, in our move to logics of trees, strings are no longer a primary notion because we are talking about trees directly. However, when we are interested in, say, checking whether a particular sentence is grammatical, we are given a string. So, while parsing, the process of determining whether a given grammar generates a given string, for CFG amounts to checking whether the grammar generates the string, this is not quite as straightforward here. The following quote from [2] gives an outline of how parsing in the logical framework might look like:

> The intent here is to translate a given grammar G into a formula φ_G such that the set of trees generated by the grammar is exactly the set of trees that satisfy φ_G. Parsing, then, is just identifying the set of models of φ_G that yield a given string.

Following an idea proposed by Cornell [20] in the context of parsing with finite tree automata, we can improve on the above parsing procedure by observing that we can describe the set of all trees that yield a given string w, φ_w, and then simply check whether $\varphi_w \wedge \varphi_G$ is satisfiable. Notice, though, that having moved from logics of strings to logic of trees entails that the complexity of parsing, which in the string case is that of *model checking*, now is that of *satisfiability checking*. For all of the modal logics considered here, satisfiability checking is EXPTIME-complete. This is still significantly better than MSO or even FO with $\triangleleft^*$ both of which are non-elementary. However, model checking for the modal logics considered here is linear. For another approach to parsing and model theoretic syntax, see Palm [71].

Variations

Just as in the case of strings, monadic transitive closure (MTC) and least fixed point (MLFP) logic and logics with modular counting quantifiers have been considered on trees [79], as well as Thomas' chain and anti-chain logics [98]. While, over trees, MLFP is equally expressive as MSO, the question whether this equivalence also holds for MTC is currently open.

Kracht [54] also considers a modal logic with quantifiers ranging over propositions which is equivalent to MSO over trees.

Extensions

While the fact that natural languages are not regular has been known since the 1950s, examples of non-context-free phenomena in natural languages were only found in the 1980s; see Shieber [89]. Thus, we again need to consider how to strengthen the logics employed here if we want this approach to be applicable to all natural languages.

One approach, a generalization of the logical characterization of CFLs to trees, is Langholm's [56] characterization of the indexed languages by an extension of MSO which generalizes the logical characterization of CFLs mentioned above to trees. The indexed languages are located strictly between the CFLs and the context-sensitive languages. However, as was pointed out above, since parsing with tree logics involves testing for satisfiability rather than model checking, using an undecidable logic makes this approach uninteresting to computational linguistics.

Other approaches to extending model theoretic syntax to non-regular tree languages include Rogers' [84] extension of MSO to n-dimensional trees and the approach by Mönnich and colleagues [47] that encodes non-regular tree language in regular tree languages. Both approaches have in common that they introduce a new level of abstraction, since the direct connection between a logical formula and the tree it encodes is only available via a translation, which is explicit only in the latter approach. While this move from trees to more complex structures is analogous to the move from strings to trees, the latter move still corresponds to structures employed by linguists (derivation trees) while the former does not. However, both approaches retain decidability. Whether decidable, non-regular extensions of PDL can be used to define interesting classes of tree languages is, at present, an open problem.

3.4 Assessment: Why Modal Logic for Syntax and Which One?

The foregoing multitude of tree logics raises two questions: what are the advantages and disadvantages of modal logics over classical logics for the description of trees, and similarly between the different modal logics? With respect to classical logic, the advantage is not, as in the general case, that modal logics are decidable while classical logic is not, since even MSO over trees is decidable. However, there is an advantage in complexity: all the modal logics considered are EXPTIME-complete [1], while MSO and FO with $\triangleleft^*$ are not elementary. One exception is FO with two successors, S_1, S_2 which is elementary [27], but not very expressive, since not even $\triangleleft^*$ is FO definable from S_1, S_2. For further discussions of complexity theoretic aspects of MSO, see [61].

Another more general question: why should logic be used at all to formalize grammatical theories? The first advantage that the approach outlined in this chapter has is that

it connects a descriptive approach to grammars to a procedural approach: grammars formalized in these logics can be translated into tree automata which can be implemented. Another issue has to do with methodology in linguistics. While some linguists have become downright hostile towards formalization, the methodological paradigm of Government and Binding theory was to formulate more and more "principles;" i.e., general statements about the structure of sentences that were supposed to be true for all languages. However, it was quite unclear how one would check whether or not any new principle was consistent with all the previously stated principles. Formalizing principles from GB in one of these logics would allow to check whether an adding a given principle would make a particular theory contradictory. For further discussions of methodological issues in GB, see Hintikka and Sandu [38].

4 CONCLUSION AND OPEN PROBLEMS

Like other areas of applied mathematics which use formal tools to model phenomena under consideration, logic in general, and modal logic in particular, is one of the main tools for modeling in mathematical linguistics. As we have seen, modal logic is used in semantics to give a formal model of the meanings of the object language, while it is used in syntax to formalize grammatical theories; i.e., the meta-language. While the use of logic in semantics has considerable history with many significant successes, the logical approach to syntax outlined here is relatively new, although its foundations date back further.

There are many applications of logic in linguistics that we have not discussed here, however, two stand out because they contain applications of modal logic: categorial grammar and feature structures. However, both of these topics have already received authoritative surveys in the *Handbook of Logic & Language* [4].

One area of research in mathematical linguistics that has had considerable success in recent years has been the study of learnability of grammar formalisms, particularly of variations of categorial grammars; see Buszkowski [12]. Similar results for model theoretic syntax have not been obtained yet. While there exist interesting approaches to learning logical theories [62] which would seem to be relevant to extending learnability theory to model theoretic syntax, these approaches depend heavily on properties of their main tool, first-order logic. Thus, a significant amount of groundwork would have to be done before one could extend this approach to model theoretic syntax.

Further open problems in model theoretic syntax include computational implementations, for which some progress has already been made by the existing implementations of monadic second order logic [46]. However similar implementations of modal logics of trees or applications of the existing applications to linguistic problems do not seem to exist. The relationship between the different approaches to extending model theoretic syntax to non-regular tree languages outlined above is also currently open. For example, is there an easy way to translate between Rogers' extension in [84] of MSO to n-dimensional trees and the approach by Mönnich and colleagues [47] that encodes non-regular tree language in regular tree languages? Finally, while the different modal logics in this chapter were separated using the tree languages in Propositions 29, 30 and 31, it would be interesting to find linguistically motivated tree languages that can also separate these logics. Until such examples are found, very little motivation seems to exist to use the more expressive logics.

One interesting property that the logical approaches to both syntax and semantics outlined here have in common is that extending their empirical scope to different natural language phenomena depends on corresponding coverage of these phenomena in some syntactic theory. Since it is the main aim of model theoretic syntax to formalize linguistic theories, instead of *being* a linguistic theory, this dependence is clear here. In the case of semantic theory, coverage of linguistic phenomena depends, because of the principle of compositionality, on syntactic representations from which the semantic representations are built.

ACKNOWLEDGEMENTS

We thank Uwe Mönnich, Patrick Blackburn, Reinhard Muskens, Lawrence Stout, and Johan van Benthem for useful comments and encouragement at various points. Hans-Jörg Tiede would like to acknowledge support from Illinois Wesleyan University in the form of an Artitistic/Scholarly Development Grant and a Junior Faculty Leave.

INDEX

coalgebra, 30
conditional, 9

Frege, Gottlob, 7

hybrid logic, 15

intensionality, 7

language
    context free, 24
    regular, 28
    regular tree, 31
Lewis, David, 7
linguistics, 1–43
    mathematical, 23
logic
    of strings, 25
    of trees, 31

McCawley, James, 10
Montague, 3
multidimensional modal logic, 16

PDL, 28
Prior, Arthur, 11

Reichenbach, Hans, 13
root modal, 8

syntax
    model theoretic, 40
    natural language, 23–42

tense
    logic of, 11, 15
    natural language, 10
    reference time, 13

BIBLIOGRAPHY

[1] Loredana Afanasiev, Patrick Blackburn, Ioanna Dimitriou, Bertrand Gaiffe, Evan Goris, Maarten Marx, and Maarten de Rijke. PDL for ordered trees. *Journal of Applied Non-Classical Logic*, 15(2):115–135, 2005.

[2] Rolf Backofen, James Rogers, and K. Vijay-Shanker. A first-order axiomatization of the theory of finite trees. *Journal of Logic, Language and Information*, 4(1):5–39, 1995.

[3] Yaniv Bargury and Johann Makowsky. The expressive power of transitive closure and 2-way multihead automata. In Egon Börger, Gerhard Jäger, and Hans Kleine Büning, editors, *Computer science logic (Berne, 1991)*. Springer, Berlin, 1992.

[4] Johan van Benthem and Alice ter Meulen, editors. *Handbook of logic and language*. Elsevier, Amsterdam, 1997.

[5] Robert I. Binnick. The project on annotated bibliography of contemporary research in tense, grammatical aspect, *aktionsart*, and related areas.
`http://www.scar.utoronto.ca/~binnick/TENSE/logic.html`, 2005.

[6] Patrick Blackburn. Tense, temporal reference and tense logic. *Journal of Semantics*, 11:83–101, 1994.

[7] Patrick Blackburn, Claire Gardent, and Wilfried Meyer Viol. Talking about trees. In Steven Krauwer, Michael Moortgat, and Louis des Tombe, editors, *Sixth Conference of the European Chapter of the Association for Computational Linguistics — Proceedings of the Conference*. ACL, 1993.

[8] Patrick Blackburn and Wilfried Meyer-Viol. Linguistics, logic and finite trees. *Bulletin of the Interest Group in Pure and Applied Logics*, 2(1):3–29, 1994.

[9] Patrick Blackburn, Wilfried Meyer-Viol, and Maarten de Rijke. A proof system for finite trees. In Dirk van Dalen and Marc Bezem, editors, *Computer science logic (Paderborn, 1995)*. Springer, Berlin, 1996.

[10] J. Richard Büchi. Weak second-order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 6:66–92, 1960.

[11] John P. Burgess. Basic tense logic. In Dov Gabbay and Franz Guenther, editors, *Handbook of Philosophical Logic*, volume 2. Reidel, Dordrecht, 1984.

[12] Wojciech Buszkowski. Type logics in grammar. In Vincent F. Hendricks and Jacek Malinowski, editors, *Trends in logic*. Kluwer, Dordrecht, 2003.

[13] John Cantwell. Comparatives. *Theoretical Linguistics*, 21(2-3):145–158, 1995.

[14] Rudolf Carnap. *Meaning and Necessity*. University of Chicago Press, Chicago, second edition, 1956.

[15] Gennaro Chierchia and Sally McConnell-Ginet. *Meaning and Grammar: An Introduction to Semantics*. The MIT Press, Cambridge, second edition, 2000.

[16] Noam Chomsky. Three models for the description of language. *IRE Transactions of information theory*, 2-3:113–124, 1956.

[17] Noam Chomsky. *Syntactic Structures*. Mouton, The Hague, 1957.

[18] Noam Chomsky. *Lectures on Government and Binding*. Foris Publications, Dordrecht, 1981.

[19] Joëlle Cohen, Dominique Perrin, and Jean-Eric Pin. On the expressive power of temporal logic. *Journal of Computer and System Sciences*, 46(3):271–294, June 1993.

[20] Thomas Cornell. Parsing and grammar engineering with tree automata. In Anton Nijholt Dirk Heylen and Giuseppe Scollo, editors, *Algebraic Methods in Language Processing AMiLP 2000, Iowa City, Iowa*, 2000.

[21] Dick Crouch. Temporality in natural language. ESSLLI class notes, 1998.

[22] Henriette de Swart. *Introduction to Natural Language Semantics*. CSLI Publications, Stanford, CA, 1998.

[23] John Doner. Tree acceptors and some of their applications. *Journal of Computer and System Sciences*, 4:406–451, 1970.

[24] David R. Dowty, Robert E. Wall, and Stanley Peters. *Introduction to Montague Semantics*. Reidel, Dordrecht, 1989.

[25] Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite Model Theory*. Springer, Berlin, 1995.

[26] Kousha Etessami and Thomas Wilke. An until hierarchy and other applications of an Ehrenfeucht-Fraïssé game for temporal logic. *Information and Computation*, 160(1/2):88–108, July 2000.

[27] Jeanne Ferrante and Charles W. Rackoff. *The computational complexity of logical theories*. Springer, Berlin, 1979.

[28] Anette Frank. *Context Dependence in Modal Constructions*. PhD thesis, University of Stuttgart, 1997.

[29] Gottlob Frege. On sense and reference. In Peter Geach and Max Black, editors, *Translations from the Philosophical Writings of Gottlob Frege*. Blackwell, Oxford, 1960.

[30] Dov Gabbay. Functional completeness in tense logic. In Uwe Mönnich, editor, *Aspects of Philosophical Logic*. Reidel, Dordrecht, 1981.

[31] Dov Gabbay, Amir Pnueli, Saharon Shelah, and Jonathan Stavi. On the temporal analysis of fairness. In *Proc. 7th Annual ACM Symposium on Principles of Programming Languages*, 1980.

[32] L. T. F. Gamut. *Logic, Language and Meaning*, volume 2. The University of Chicago Press, Chicago, 1991.
[33] Gerald Gazdar, Ewan Klein, Geoffrey Pullum, and Ivan Sag. *Generalized Phrase Structure Grammar*. Harvard University Press, Cambridge, 1985.
[34] Ferenc Gécseg and Magnus Steinby. Tree languages. In *Handbook of formal languages, Vol. 3*. Springer, Berlin, 1997.
[35] Saul Gorn. Explicit definitions and linguistic dominoes. In *Systems and Computer Science (Proc. Conf., London, Ont., 1965)*. Univ. Toronto Press, Toronto, Ont., 1967.
[36] Irene Heim and Angelika Kratzer. *Semantics in Generative Grammar*. Blackwell, Malden, MA, 1998.
[37] Erhard W. Hinrichs. Tense, quantifiers, and contexts. *Computational Linguistics*, 14(2):3–14, 1988.
[38] Jaakko Hintikka and Gabriel Sandu. *On the Methodology of Linguistics*. Basil Blackwell, Oxford, 1991.
[39] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Reading, 1979.
[40] Neil Immerman. *Descriptive Complexity*. Springer, Berlin, 1999.
[41] Theo M. V. Janssen. Compositionality. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*. Elsevier, Amsterdam, 1997.
[42] Hans Kamp. Formal properties of "now". *Theoria*, 37:227–273, 1971.
[43] Hans Kamp and Uwe Reyle. *From Discourse to Logic*. Kluwer, Dordrecht, 1993.
[44] Johan Anthony Willem Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, University of California, Los Angeles, 1968.
[45] Bakhadyr Khoussainov and Anil Nerode. *Automata theory and its applications*. Birkhäuser, Boston, 2001.
[46] Nils Klarlund. Mona & fido: The logic-automaton connection in practice. In Georg Gottlob, Etienne Grandjean, and Katrin Seyr, editors, *Computer Science Logic (Brno, 1998)*, Berlin, 1998. Springer.
[47] Hans-Peter Kolb, Jens Michaelis, Uwe Mönnich, and Frank Morawietz. An operational and denotational approach to non-context-freeness. *Theoretical Computer Science*, 293(2):261–289, 2003.
[48] Andràs Kornai and Geoffrey K. Pullum. The X-bar theory of phrase structure. *Language*, 66:24–50, 1990.
[49] Marcus Kracht. Syntactic codes and grammar refinement. *Journal of Logic, Language and Information*, 4(1):41–60, 1995.
[50] Marcus Kracht. Inessential features. In Alain Lecomte, François Lamarche, and Guy Perrier, editors, *Logical aspects of computational linguistics*. Springer, Berlin, 1997.
[51] Marcus Kracht. On reducing principles to rules. In Patrick Blackburn and Maarten de Rijke, editors, *Specifying Syntactic Structures*. CSLI Publications, Stanford, CA, 1997.
[52] Marcus Kracht. *Tools and techniques in modal logic*. North-Holland, Amsterdam, 1999.
[53] Marcus Kracht. Logic and syntax—a personal perspective. In Michael Zakharyaschev, Krister Segerberg, Maarten de Rijke, and Heinrich Wansing, editors, *Advances in modal logic, Vol. 2*. CSLI Publications, Stanford, CA, 2001.
[54] Marcus Kracht. *The Mathematics of Language*. de Gruyter, Berlin, 2003.
[55] Angelika Kratzer. The notational category of modality. In Hans-Jürgen Eikmeyer and Hannes Rieser, editors, *Words, Worlds, and Contexts*, pages 38–74. de Gruyter, BERLIN, 1981.
[56] Tore Langholm. A descriptive characterisation of indexed grammars. *Grammars*, 4(3):205–262, 2001.
[57] Shalom Lappin, editor. *The Handbook of Contemporary Semantic Theory*. Blackwell Publishers, Oxford, 1996.
[58] Clemens Lautemann, Thomas Schwentick, and Denis Thérien. Logics for context-free languages. In Leszek Pacholski and Jerzy Tiuryn, editors, *Computer science logic (Kazimierz, 1994)*. Springer, Berlin, 1995.
[59] David Lewis. General semantics. In Donald Davidson and Gilbert H. Harman, editors, *Semantics of Natural Language*. Reidel, Dordrecht, 1972.
[60] David Lewis. *Counterfactuals*. Harvard University Press, Cambridge, 1973.
[61] Leonid Libkin. *Elements of finite model theory*. Springer, Berlin, 2004.
[62] Eric Martin and Daniel N. Osherson. *Elements of Scientific Inquiry*. MIT Press, Cambridge, MA, 1998.
[63] Maarten Marx. Conditional XPath, the first order complete XPath dialect. In *Proceedings of PODS '04*, 2004.
[64] James D. McCawley. *Everything that Linguists Have Always Wanted to Know about Logic (But Were Ashamed to Ask)*. University of Chicago Press, Chicago, second edition, 1993.
[65] James D. McCawley. Departmental web site. `http://humanities.uchicago.edu/depts/linguistics/faculty/mccawley.html`, 1998.
[66] Robert McNaughton and Seymour Papert. *Counter-Free Automata*. MIT Press, Cambridge, MA, 1971.

[67] Yiannis N. Moschovakis. Sense and denotation as algorithm and value. In Juha Oikkonen and Jouko Väänänen, editors, *Logic Colloquium '90 (Helsinki, 1990)*. Springer, Berlin, 1993.
[68] Yuko Murakami. *Modal Logic of Partitions*. PhD thesis, Indiana University, Bloomington, IN, 2005.
[69] Reinhard Muskens, Johan van Benthem, and Albert Visser. Dynamics. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*. Elsevier, Amsterdam, 1997.
[70] Adi Palm. Propositional tense logic for finite trees. In *Proceedings of Mathematics of Language (MOL 6)*, 1999.
[71] Adi Palm. Model theoretic syntax and parsing: An application to temporal logic. In *Proceedings of Formal Grammar and Mathematics of Language (FGMOL)*, 2001.
[72] Barbara Partee, Alice ter Meulen, and Robert E. Wall. *Mathematical Methods in Linguistics*. Kluwer, Dordrecht, 1990.
[73] Barbara H. Partee. Some structural analogies between tenses and pronouns. *The Journal of Philosophy*, 70(18):601–609, 1973.
[74] Barbara H. Partee. Possible worlds in model-theoretic semantics: A linguistic perspective. In Sture Allén, editor, *Possible Worlds in Humanities, Arts and Sciences: Proceedings of Nobel Symposium 65*. de Gruyter, Berlin, 1988.
[75] Barbara H. Partee. Montague grammar. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*. Elsevier, Amsterdam, 1997.
[76] John Perry. Possible worlds and subject matter. In Sture Allén, editor, *Possible Worlds in the Humanities, Arts and Sciences, Proceedings of the Nobel Symposium 65*. de Gruyter, Berlin, 1989.
[77] Jean-Eric Pin. Logic, semigroups and automata on words. *Annals of Mathematics and Artificial Intelligence*, 16:343–384, 1996.
[78] Roland Posner. Believing, causing, intending. In René Jorna, Barend van Heusden, and Roland Posner, editors, *Signs, Search, and Communication*. de Gruyter, Berlin, 1992.
[79] Andreas Potthoff. *Logische Klassifizierung regulärer Baumsprachen*. PhD thesis, Christian-Albrechts-Universität zu Kiel, 1994.
[80] Andreas Potthoff. Modulo-counting quantifiers over finite trees. *Theoretical Computer Science*, 126(1):97–112, 1994.
[81] Andreas Potthoff and Wolfgang Thomas. Regular tree languages without unary symbols are star-free. In *Fundamentals of computation theory (Szeged, 1993)*. Springer, Berlin, 1993.
[82] Hans Reichenbach. *Elements of Symbolic Logic*. University of California Press, Berkeley, 1947.
[83] James Rogers. *A descriptive approach to language-theoretic complexity*. Studies in Logic, Language and Information. CSLI Publications, Stanford, CA, 1998.
[84] James Rogers. Syntactic structures as multi-dimensional trees. *Research on Language and Computation*, 1(3-4):265–305, 2003.
[85] William C. Rounds. LFP: A logic for linguistics descriptions and an analysis of its complexity. *Computational Linguistics*, 14(4):1–9, 1988.
[86] J. J. M. M. Rutten. Automata and coinduction (an exercise in coalgebra). In Davide Sangiorgi and Robert de Simone, editors, *CONCUR'98: concurrency theory*. Springer, Berlin, 1998.
[87] Bernd-Holger Schlingloff. On the expressive power of modal logics on trees. In Anil Nerode and Michael A. Taitslin, editors, *Logical Foundations of Computer Science - Tver '92*. Springer, Berlin, 1992.
[88] Stuart Shieber. Direct parsing of ID/LP grammars. *Linguistics and Philosophy*, 7:135–154, 1984.
[89] Stuart Shieber. Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8:333–343, 1985.
[90] Robert C. Stalnaker. A theory of conditionals. In William L. Harper, Robert Stalnaker, and Glenn Pearce, editors, *Ifs*. Reidel, Dordrecht, 1981.
[91] Mark Steedman. *The Productions of Time: Temporality and Causality in Linguistic Semantics*. ms., 2002.
[92] Howard Straubing. *Finite automata, formal logic, and circuit complexity*. Birkhäuser, Boston, 1994.
[93] Howard Straubing, Denis Thérien, and Wolfgang Thomas. Regular languages defined with generalized quantifiers. *Information and Computation*, 118(2):289–301, 1995.
[94] Alice G. B. ter Meulen, editor. *Representing Time in Natural Language: The Dynamic Interpretation of Tense and Aspect*. MIT Press, Cambridge, MA, 1997.
[95] James W. Thatcher. Characterizing derivation trees of context-free grammars through a generalization of finite automata theory. *Journal of Computer and System Sciences*, 1:317–322, 1967.
[96] James W. Thatcher. Tree automata: an informal survey. In *Currents in the theory of computing*. Prentice-Hall, Englewood Cliffs, N. J., 1973.
[97] James W. Thatcher and Jesse B. Wright. Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical Systems Theory*, 2:57–81, 1968.
[98] Wolfgang Thomas. Languages, automata, and logic. In *Handbook of formal languages, Vol. 3*. Springer, Berlin, 1997.

[99] Richmond H. Thomason, editor. *Formal Philosophy: Selected Papers of Richard Montague*. Yale University Press, New Haven, 1974.
[100] Richmond H. Thomason. Combinations of tense and modality. In Dov Gabbay and Franz Guenthner, editors, *Handbook of Philosophical Logic*, volume 7. Kluwer, Dordrecht, second edition, 2002.
[101] Hans-Jörg Tiede. Proof theory and formal grammars: applications of normalization. In Benedikt Löwe, Wolfgang Malzkom, and Thoralf Räsch, editors, *Foundations of the formal sciences II*. Kluwer, Dordrecht, 2003.
[102] Hans-Jörg Tiede. Inessential features, ineliminable features, and modal logics for model theoretic sytax. In *Proceedings of Formal Grammar and Mathematics of Language (FGMOL)*, 2005.
[103] Johan van Benthem. *The Logic of Time*. Reidel, Dordrecht, 1983.
[104] Johan van Benthem. Temporal logic. In Dov Gabbay, Chris Hogger, and J. Alan Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 4. Oxford University Press, Oxford, 1995.
[105] Michiel van Lambalgen and Fritz Hamm. *The Proper Treatment of Events*. Blackwell, Malden, MA, 2005.
[106] Franz von Kutschera. Indicative conditionals. *Theoretical Linguistics*, 1(3):257–269, 1974.
[107] Zsófia Zvolensky. Is a possible-worlds semantics of modality possible? A problem for Kratzer's semantics. In Brendan Jackson, editor, *Proceedings of SALT XII*. Cornell University Press, Ithaca, 2002.