Epistemic Semantics in Guarded String Models

Eric Hayden Campbell
Cornell University
ehc86@cornell.edu

Mats Rooth
Cornell University
mr249@cornell.edu

Abstract

Constructive and computable multi-agent epistemic possible worlds models are interpreted as sets of guarded string models in an epistemic extension of Kleene Algebra with Tests (KAT). The account is framed as a formal language EpiKAT (Epistemic KAT) for defining such models. The language is implemented by translation into the finite state calculus, and alternatively by modeling propositions as lazy lists in Haskell. The syntax-semantics interface for a fragment of English is defined by a categorial grammar.

1 Introduction and Related Work

Linguistic semantics in the Montague tradition proceeds by assigning propositional *semantic values* to disambiguated sentences of a natural language. A proposition is a set or class of *possible worlds*, which are often assumed to have the same nature and complexity as the world we occupy (Lewis, 1986). But alternatively, one can work with small idealized models, to illustrate and test ideas. The point of this paper is to extend idealized models to countable sets of worlds, and to constructively and computably model alternatives for epistemic agents.

Enter EpiKAT, which is a systematic way of defining such models, and can be applied to natural language semantics, specifically, epistemic semantics and clausal embedding. The fundamental insight is to identify possible worlds with strings of primitive events, so that propositions are sets of (guarded) strings, whose regular sets have a rich algebraic characterization (Kozen, 1997) and computational model (Kozen and Smith, 1997; Kozen, 2001). EpiKAT leverages these algebras to define a mathematical model (Sections 2 & 3) and computationally interpret using finite state machines (Section 4) and lazy lists of strings (Section 5)).

Related Work. EpiKAT synthesizes five antecedents. John McCarthy's *Situation Calculus* is the source of the idea of constructing possible worlds as event sequences (McCarthy, 1963; Reiter, 2001). The algebraic theory of *Kleene Algebra with Tests* (KAT) characterizes algebras of regular sets of guarded strings (Kozen, 2001), which form the basis for EpiKAT's propositions and event types. *Action models* in dynamic epistemic semantics introduce the technique of constructing epistemic models from primitive alternative relations on events, in order to capture the epistemic consequences of perceptual and communicative events (Baltag et al., 1999). Literature on *finite state methods in linguistic semantics* has used event strings and sets of event strings to theorize about tense and aspect in natural language semantics (Fernando, 2004, 2007; Carlson, 2009) and to express intensions (Fernando, 2017). Work on *finite state intensional semantics* has investigated how to do the semantics of intensional complementation in a setting where compositional semantics is expressed in a finite state calculus (Rooth, 2017; Collard, 2018).

Example. Consider an example event-sequence model called *The Concealed Coin*. Amy and Bob are seated at a table. There is a coin on the table under a cup. The coin could be heads-up (H) or tails-up (T), and neither agent knows which it is. Call this initial possible world w_1. Additional worlds can be constructed via the events in (1).

(1) a_h Amy peeks at H, by tipping the cup. Bob sees she's peeking, but not what she sees.

b_h Bob peeks at H.

Thanks to Tobias Kappé for insightful reactions to the paper. Thanks also to three SCiL reviewers for helpful comments. Earlier versions of the work were presented in the LUSH talk series at Utrecht in December 2019, and in the Workshop on formal/philosophical/computational approaches to natural language meaning at Rochester in December 2020. Thanks to the audiences for their reactions.

a_t	Amy peeks at T.
b_t	Bob peeks at T.
a_{th}	Amy secretly turns the coin from T to H. She knows she turned the coin over, but not which side was face up. Bob thinks nothing happened.
a_{ht}	Amy secretly turns the coin from H to T.
b_{th}	Bob secretly turns the coin from T to H.
b_{ht}	Bob secretly turns the coin from H to T.

The worlds in (2) are examples constructed from the events in (1). Juxtaposition indicates the order of events; for example, w_3 indicates that after the events of w_2, event b_1 occurs, which, starting in the initial world w_1 corresponds to Amy peeking at a heads-up coin, and then Bob doing the same.

(2) $w_2 \triangleq w_1 a_h \quad w_3 \triangleq w_2 b_h \quad w_4 \triangleq w_3 a_{ht} b_{th} b_h$

Now, the truth (1) or falsity (0) of English sentences can be evaluated in each world w_i, i.e. *after* the events have happened. See (3) for examples such as in w_4, where Bob knows it's heads (because he peeked after turning the coin), but Amy does not (because Bob secretly turned the coin over).

(3)

w_1	w_2	w_3	w_4	Sentence
0	1	1	0	Amy knows it's heads.
0	0	1	1	Bob knows it's heads.
0	0	1	0	Bob knows Amy knows it's heads.
0	1	1	0	Bob knows Amy knows whether it's heads or tails.

To create realistic models, contradictory event sequences must be prohibited. For example, $a_h a_{th}$ would require the coin to simultaneously be heads-up and tails-up. To mechanize such reasoning, events in EpiKAT come with Boolean pre-conditions and post-conditions *à la* Hoare Logic (Hoare, 1969). In the running example, the Boolean variables h (the coin is heads-up) and t (the coin is tails-up) represent the coin's state. Then, since Amy can only observe heads when the coin is heads-up, event a_h has the invariant condition h. Similarly, Amy can only turn the coin from tails to heads if it shows tails, so a_{th} has the tails-up precondition t and the heads-up postcondition h.

Pre- and post- conditions are expressed using an operator ":" (read "and next") that pairs Boolean formulas into an *effect formula*. Specifically, the effects for a_h are written $h : h$, and for a_{th} are written $t : h$. Effect formulas are interpreted as relations on boolean valuations, as defined in Figure 1.

However, a coin cannot simultanously show both heads and tails! Currently, the precondition h of a_{ht} only says that heads must be showing, and says nothing about the fact that tails must be face-down, indicated by the formula $\bar{t}$. The effect valuations are restricted to only the feasible ones via a *state formula*, shown and demonstrated for a_h in (4). The state formula says that the coin is either heads-up or tails-up, but not both[1]. Here juxtaposition represents conjuction, $+$ is disjunction, and overbar is negation.

With the state formula in hand, $a_h a_{th}$ is contradictory, since the post-condition h of a_h is incompatible with the pre-condition t of a_{th}. See Figure 1 for more details.

Given a set B of state primitives and φ a state formula over B, define $\mathcal{A}^{\varphi}_{\mathsf{B}}$ to be the set of valuations of B that make formula φ true. The valuations for the coin example are shown in 4, using the sequence notation for valuations, e.g. $\bar{h}t$, wherein every state primitive is listed in fixed order, and left unmarked (indicating true) or marked with the overbar (indicating false). Valuations are called atoms, because they correspond to the atoms of a Boolean algebra of tests (Kozen, 2001).

(4)

B	$\{h, t\}$
state formula φ	$h\bar{t} + \bar{h}t$
effect formula ζ_{a_h} for a_h	$h : h$
$\mathcal{A}^{\varphi}_{\mathsf{B}}$	$\{h\bar{t}, \bar{h}t\}$
$[\![\zeta_{a_h}]\!]^{\varphi}$	$\{\langle h\bar{t}, h\bar{t}\rangle\}$

Thus far, the presentation is closely related to Kleene Algebra with Hypotheses (Cohen, 1994), which permits user-specification of futher equations beyond the axioms of KA. Specifically, the state formula φ can be thought of as a hypothesis $\varphi \equiv 1$, and an event formula ζ_a for event a, is simply the hypothesis $a \equiv \sum_{\varphi,\varphi' \in [\![\zeta_a]\!]} \varphi; a; \varphi'$. Rather than rely on general techniques (Doumane et al., 2019; Hardin, 2002; Kozen and Smith, 1997), we realize our specific hypotheses directly.

EpiKAT is first and foremost a multi-agent epistemic logic, with epistemic operators that cannot naively be represented as hypotheses.[2] Aside from formalizing the metaphysical modality seen so far, Sections 2 and 3 develop the epistemic modality.

[1] Two Booleans constrained by a state formula, is more illustrative than a single boolean h where tails is just $\bar{h}$.

[2] Characterizing this relationship precisely is future work.

state formulas $(a \in \mathsf{B})$

$\rho, \sigma, \varphi \quad ::= \quad a \mid 0 \mid 1 \mid \rho + \sigma \mid \rho\,\sigma \mid \bar{\rho}$

effect formulas

$\zeta, \eta \quad ::= \quad \rho : \sigma \mid \zeta + \eta \mid \zeta \,\&\, \eta \mid \bar{\zeta}$

$[\![\rho : \sigma]\!]^{\varphi} \triangleq \mathcal{A}_{\mathsf{B}}^{\rho\varphi} \times \mathcal{A}_{\mathsf{B}}^{\sigma\varphi}$

$[\![\zeta + \eta]\!]^{\varphi} \triangleq [\![\zeta]\!]^{\varphi} \cup [\![\eta]\!]^{\varphi}$

$[\![\zeta \,\&\, \eta]\!]^{\varphi} \triangleq [\![\zeta]\!]^{\varphi} \cap [\![\eta]\!]^{\varphi}$

$[\![\bar{\zeta}]\!]^{\varphi} \triangleq \mathcal{A}_{\mathsf{B}}^{\varphi} \times \mathcal{A}_{\mathsf{B}}^{\varphi} \setminus [\![\zeta]\!]^{\varphi}$

Figure 1: Syntax of state formulas and syntax and semantics of effect formulas. Effect formulas denote relations between atoms. In a state formula, juxtaposition $\rho\,\sigma$ is conjunction.

2 Epistemic guarded string models

EpiKAT is a specification language for possible worlds models that includes declarations of events and states, state formulas, effect formulas, and additional information. Figure 2 shows an EpiKAT program that describes a possible worlds model for two agents with information about one coin, events of the agents semi-privately looking at the coin, and events of secretly turning the coin. The line beginning with `state` enumerates B. The line beginning with `restrict` gives the state formula. The lines beginning with `event` declare events and their effect formulas.

Finally, the lines beginning with `agent` define *event alternative* relations for the epistemic agents in the model. Each clause with an arrow has a single event symbol on the left, and a disjunction of alternative events on the right of the arrow. The interpretation of Amy's alternatives for b_h (Bob peeks at heads), is that when b_h happens, for Amy either b_h or b_t (Bob peeks at tails) could be happening, indicating that she doesn't know whether Bob saw heads or tails, only that he peeked at the coin.

Similarly, her alternatives for a_{th} (she turns the coin over from heads to tails) are a_{th} and a_{ht}, indicating that she doesn't know, *a priori*, whether she's turning the coin from H to T or from T to H. She has the same alternatives for the event a_{ht}. Conversely, when Bob secretly turns the coin over, in event b_{th} or b_{ht}, she doesn't know anything has happened, so her alternative is the "no-operation" or "no-information" event o. Bob's event relation is symmetric.

The sequel focuses on defining a concrete possible worlds model from an EpiKAT specification. The models are an extension of guarded-string models for Kleene Algebra with Tests (KAT). This algebraic theory has model classes including guarded

```
state h t          agent amy
restrict h!t         o -> o
         + t!h       a1 -> a1
event o h:h + t:t    a0 -> a0
event a1 h:h         b1 -> b1 + b0
event a0 t:t         b0 -> b1 + b0
event b1 h:h         a10 -> a10 + a01
event b0 t:t         a01 -> a10 + a01
event a10 h:t        b10 -> o
event a01 t:h        b01 -> o
event b10 h:t      agent bob
event b01 t:h        <sim. swap a and b>
```

Figure 2: EpiKAT program describing a possible-worlds event sequence model for two agents with information about one coin, and events of the agents semi-privately looking at the coin, and privately turning the coin.

string models, relational models, finite models, and matrix models. Our definitions and notation follow (Kozen, 2001). We add syntax and semantics to cover multi-agent epistemic semantics.

Guarded strings over a finite alphabet P are like ordinary strings, but with atoms over a set B alternating with the symbols from P. In the model described by Figure 2, P is the set of events $\{a_h, a_t, b_h, b_t, a_{th}, a_{ht}, b_{th}, b_{ht}, o\}$, and B is $\{h, t\}$. As we already saw in (4), $\mathcal{A}_{\mathsf{B}}^{\varphi}$ is $\{h\bar{t}, \bar{h}t\}$, for which we use the shorthand $\{H, T\}$. A guarded string over P and B is a string of events from P, alternating with atoms over B, and beginning and ending with atoms. In this construction, $w_1 = H$, $w_2 = Ha_hH$, $w_3 = Ha_hHb_hH$, and $w_4 = Ha_hHb_hHa_{ht}Tb_{th}Hb_hH$.

The discussion of (2) mentioned building worlds by incrementing smaller worlds with events and maintaining the pre- and post- conditions. This is accomplished in guarded string models with the fusion product $(x \diamond y)$, a partial operation that combines two guarded strings x and y, subject to the condition that the atom at the end of the x is identical to the atom at the start of y. (5) gives some examples.

(5) $H\,b_h\,H \;\diamond\; H\,a_h\,H = H\,b_h\,H\,a_h\,H$
$T\,b_{th}\,H \;\diamond\; T\,a_h\,T = \textit{undefined}$

Rather than individual guarded strings, elements of a guarded string model for KAT are sets of guarded strings. In EpiKAT, these elements have the interpretation of propositions, which are (regular) sets of possible worlds. In a free guarded string model for KAT, any event can be adjacent to any atom in a guarded string that is an element

of the underlying set for the algebra. However, this subsumes the valid worlds defined by the state and effect formulas. (6) defines the *well-formed* guarded strings (valid worlds) determined by an EpiKAT specification. Condition (i) says that each atom is consistent with the state constraint, and condition (ii) says that each constituent token event $\alpha_i e_i \alpha_{i+1}$ is consistent with its effects.[3]

(6) Given P, B, a state formula φ, and an effect formula ζ_e for each event e in P, $\alpha_0 e_0 ... e_n \alpha_{n+1}$ is well-formed iff
(i) $\alpha_i \in \mathcal{A}_{\mathsf{B}}^{\varphi}$ $(0 \leq i \leq n)$, and
(ii) $\langle \alpha_i, \alpha_{i+1} \rangle \in [\![\zeta_{e_i}]\!]^{\varphi}$, $(0 \leq i \leq n)$.

Well-formed guarded strings have the interpretation of worlds in the application to natural-language semantics. The set of possible worlds in the modal frame determined by an EpiKAT specification is the set of well-formed guarded strings, and propositions are sets of guarded strings. Certain sets of well-formed guarded strings have the addional interpretation of event types. An event-type is something that can "happen" in different worlds. For example, a_h corresponds to the event type $\{H\, a_h\, H\}$, and o to the event type $\{T\, o\, T, H\, o\, H\}$.

The construction so far defines a set of worlds from an EpiKAT specification. Normally the set is countably infinite, though some choices of effect formulas can result in a finite set of worlds. The next step is to define an alternative relation R_a on worlds for each agent a. This will result in a general modal frame $\langle W, \hat{R}_1, ..., \hat{R}_n, K \rangle$ consisting of a set of worlds, a world-alternative relation for each agent, and a set K of propositions, where each proposition is a subset of W (Chagrov, 1997).[4]

An EpiKAT specification defines an alternative relation on bare events for each agent a, which is notated R_a, and lifted to a relation $\hat{R}_a$ on worlds. The basic idea is that when a world w is incremented with an event e, in the resulting world $w \diamond e$, epistemic alternatives for agent a are of the form $w' \diamond e'$, where w' is an alternative to for a in w, and

events	$e \in \mathsf{P}$		
p, q	$::= e \mid \sigma \mid p + q \mid pq \mid p^* \mid \neg p \mid \Diamond_a p$		
$\Box_a p$	$\triangleq \neg \Diamond_a \neg p$	$\bullet$	$\triangleq \sum_{e \in \mathsf{P}} e$
$p \wedge q$	$\triangleq \neg(\neg p + \neg q)$	$p \rightarrow q$	$\triangleq \neg p + q$

Figure 3: The language of EpiKAT terms and key derived operators.

e' is and event-alternative to e for a.[5] This needs to be implemented in a way that takes account of pre- and post-conditions for events. For this, our approach is to refer the definition of well-formed guarded strings. (7) defines an epistemic alternative relation on worlds from an alternative relation on bare events.

(7) Let W be a set of guarded strings over events P and primitive tests B, and R be a relation on P. The corresponding relation $\hat{R}$ on W holds between a guarded string $\alpha_0 e_0 ... e_n \alpha_{n+1}$ in W and a guarded string q in W iff q is of the form $\alpha'_0 e'_0 ... e'_n \alpha'_{n+1}$,, where for $0 \leq n$, $e_i R e'_i$.

This requires that in an alternative world, each constituent event e'_i is an alternative to the corresponding event e_i in the base world. Compatibilities between events in the alternative world are enforced by the requirement that the alternative world is an element of W, so that state and effect formulas are enforced.

Consider a scenario like the one from Figure 2, augmented with an additional agent Cal. The base world $T\, b_t\, T\, c_t\, T$ is one where the coin is tails, and first Bob looks at tails, and them Cal looks at tails. The first event b_t has the alternatives b_t and b_h for Amy, and the second event b_t has the alternatives c_t and c_h for Amy. This results in four combinations b_t, c_t, b_t, c_h, b_h, c_t, and b_h, c_h. These are filtered by pre- and post-conditions of events in the alternative world, so that the set of alternatives for Amy in $T\, b_t\, T\, c_t\, T$ is $\{T\, b_t\, T\, c_t\, T, H\, b_h\, H\, c_h\, H\}$, with two world-alternatives instead of four.

[3] An alternative is to define equations such as $\bar{\phi} = 0$ (from the state formula ϕ) and $a_h = h a_h h$ (from the effect formula $h : h$ for event a_h), and construct a quotient algebra from the equivalence relation generated by these equations. This results in equating sets of guarded strings in the free algebra that differ by guarded strings that are ill-formed according to the state and effect formulas. In the development in the text, we instead use a set of guarded strings that are well-formed according to the state and effect formulas as the representative of the equivalence class.

[4] As explained in the next section, K will not be the power set of W, rather it will consists of the regular subsets of W.

[5] In this it is important that the event-alternative relation for an agent is constant across worlds. We anticipate that the definition given here produces results equivalent to what is found in literature on event alternatives in dynamic epistemic semantics, though we have not verified this. That literature primarily focuses on mapping an epistemic model for a single time and situation to another, and uses general first-order models, rather than guarded string models. See Baltag et al. (1999), Van Ditmarsch et al. (2007), and articles in Van Ditmarsch et al. (2015). This literature is motivated by epistemic logic and AI planning, rather than computable possible worlds models in natural language semantics.

3 The logical language of Epistemic KAT

The standard language for Kleene algebra with tests has the signature $\langle K, +, \cdot, *, \bar{}, 0, 1\rangle$ (Kozen, 2001). In a guarded string model for KAT, K is a set of sets of guarded strings, $+$ is set union, the operation $\cdot$ is fusion product raised to sets, $*$ is Kleene star, the operation $\bar{}$ is complement for tests, 0 is the empty set, and 1 is the set of atoms.[6] To this we add a unary modal operation $\Diamond_a$ for each agent, and a unary complement operation $\neg$ on elements of K. Intuitively, $\Diamond_a p$ is the set of worlds where proposition p is epistemically possible for agent a. Propositional complement is included because natural languages have sentence negation. In addition, universal box modalities are defined as duals of existential diamond modalities.

With modalities and propositional negation added, the signature of n-agent epistemic KAT is $\langle K, +, \cdot, *, \bar{}, 0, 1, \neg, \Diamond_1, ..., \Diamond_n\rangle$. Figure 3 defines the syntax of the language. Juxtaposition is used for product. Terms in this language are used to represent the propositional semantic values of English sentences. (8) gives examples. The abbreviation $\bullet$ as defined in Figure 3 is the disjunction of the primitive events. Since a world is a well-formed sequence of events, $\bullet^*$ is the set of worlds. Multiplying by the state symbol h in the term $\bullet^* h$ has the effect of conjoining h with the atom at the end of the world. So $\bullet^* h$ is the set of worlds where the coin ends heads-up.

(8)
$\bullet^* t$	It's tails.
$\bullet^* h$	It's heads.
$\bullet^* h \wedge \Box_a \bullet^* h$	Amy knows it's heads.
$\Box_b(\bullet^* t \wedge \Box_a \bullet^* t + (\neg \bullet^* t) \wedge \Box_a \neg \bullet^* t)$	Bob believes Amy knows whether it's tails.

Standard Epistemic Modalities. Using the modal primitive $\Diamond_a$, and the dual encodings of $\Box_a$ and $\wedge$, we can encode the standard modal operators expressing knowledge ($\mathcal{K}_a$) and belief ($\mathcal{B}_a$) as (9).[7]

(9)
BELIEF	$\mathcal{B}_a\, p$	$\triangleq \Box_a\, p$
KNOWLEDGE	$\mathcal{K}_a\, p$	$\triangleq p \wedge \mathcal{B}_a\, p$

Different types of reasoners (e.g. accurate, inaccurate, etc) are modeled using the event-alternatives in an EpiKAT specification.[8] The agents in Figure 1 do not always have reliable beliefs, because of the possibility of secret turning.

$$
\begin{array}{lcl}
[\![0]\!]^{\mathsf{B},\varphi,\zeta} & \triangleq & \emptyset \\
[\![1]\!]^{\mathsf{B},\varphi,\zeta} & \triangleq & \mathcal{A}_{\mathsf{B}}^{\varphi} \\
[\![b]\!]^{\mathsf{B},\varphi,\zeta} & \triangleq & \mathcal{A}_{\mathsf{B}}^{b\varphi} \\
[\![\bar{\sigma}]\!]^{\mathsf{B},\varphi,\zeta} & \triangleq & \mathcal{A}_{\mathsf{B}}^{\varphi} \setminus [\![\sigma]\!]^{\mathsf{B},\varphi,\zeta} \\
[\![e]\!]^{\mathsf{B},\varphi,\zeta} & \triangleq & \{\alpha e \beta | \alpha\ [\![\zeta_e]\!]^{\varphi}\ \beta\} \\
[\![p+q]\!]^{\mathsf{B},\varphi,\zeta} & \triangleq & [\![p]\!]^{\mathsf{B},\varphi,\zeta} \cup [\![q]\!]^{\mathsf{B},\varphi,\zeta} \\
[\![pq]\!]^{\mathsf{B},\varphi,\zeta} & \triangleq & \left\{ x \diamond y \;\middle|\; \begin{array}{l} x \in [\![p]\!]^{\mathsf{B},\varphi,\zeta} \\ y \in [\![q]\!]^{\mathsf{B},\varphi,\zeta} \\ x \diamond y \text{ is defined} \end{array} \right\} \\
[\![p^*]\!]^{\mathsf{B},\varphi,\zeta} & \triangleq & \bigcup_{n \geq 0} [\![p^n]\!]^{\mathsf{B},\varphi,\zeta} \\
[\![\neg p]\!]^{\mathsf{B},\varphi,\zeta} & \triangleq & [\![\bullet^*]\!]^{\mathsf{B},\varphi,\zeta} \setminus [\![p]\!]^{\mathsf{B},\varphi,\zeta} \\
[\![\Diamond_a\, p]\!] & \triangleq & \{x \mid \exists y. x \hat{R}_a y \wedge y \in [\![p]\!]^{\mathsf{B},\varphi,\zeta}\}
\end{array}
$$

Figure 4: Interpretation of EpiKAT terms as sets of guarded strings

Guarded String Interpretation. A term p of the logical language is interpreted as a set of guarded strings $[\![p]\!]^{\mathsf{B},\varphi,\zeta}$, where superscript captures dependence on an EpiKAT specification. Figure 4 defines the interpretation. The interpretation $[\![1]\!]^{\mathsf{B},\varphi,\zeta}$ of the multiplicative identity 1 is the set of atoms that satisfy the state constraint φ. Where b is a primitive Boolean, $[\![b]\!]^{\mathsf{B},\varphi,\zeta}$ is the set of atoms that satisfy the state constraint and where b is true. Where e is a primitive event, $[\![e]\!]^{\mathsf{B},\varphi,\zeta}$ is the set of guarded strings that have the form of e flanked by compatible atoms, as determined by the effect formula ζ_e. The product pq is interpreted with fusion product raised to sets of guarded strings. Kleene star is interpreted as the union of exponents (p^n is the n-times product of p with itself, with $p^0 = 1$). Propositional complement is complement relative to the set of worlds. The epistemic formula $\Diamond_a p$ is interpreted with Kripke semantics for epistemic modality, as the pre-image of p under the world-alternative relation $\hat{R}_a$.

Summing up, given an EpiKAT specification $\langle \mathsf{B}, \varphi, \zeta\rangle$, term p (as defined syntactically in Figure 3) is interpreted as a set of guarded strings $[\![p]\!]^{\mathsf{B},\varphi,\zeta}$. Let $K^{\mathsf{B},\varphi,\zeta}$ be the sets that are interpretations of terms. Then $\langle K^{\mathsf{B},\varphi,\zeta}, +, \cdot, *, \bar{}, 0, 1, \neg, \Diamond_{a_1}, ..., \Diamond_{a_n}\rangle$ is a concrete guarded string interpretation for the signature of EpiKAT, with operations as in Figure 4. This provides a concrete n-agent general modal frame $\langle [\![\bullet^*]\!]^{\mathsf{B},\varphi,\zeta}, \hat{R}_1, ..., \hat{R}_n, K^{\mathsf{B},\varphi,\zeta}\rangle$. The frame

[6] 0 has the dual role the identity for $+$ (union), and as False for operations on tests. 1 has the dual role of the identity for product (fusion product raised to sets), and True for tests.

[7] Deeper analysis of the lexical semantics of *know* requires adding modeling of presupposition (Collard, 2018). The grammar fragment in Section 6 does not model the presupposition of *know*, except as an entailment.

[8] See the discussion of modal axioms **T** and **D** below.

consists of a set of worlds, an epistemic-alternative relation for each agent, and a set of propositions, each of which is a set of worlds. It is used as a target for natural-language interpretation in Section 6. Since $K^{\mathsf{B},\varphi,\zeta}$ is the set of sets of worlds that are interpretations of terms, and there are only countably many terms, not all sets of worlds are propositions in the general modal frame. Rather, propositions are regular sets of worlds.

Axiomatic Classification. To situate our logic as a modal logic, consider the soundness of the standard modal axioms given our semantics (Hughes et al., 1996). The axioms in (10) are valid.

(10) **N** If p is valid, then $\Box_a p$ is valid
K $\Box_a(p \to q) \to \Box_a\, p \to \Box_a\, q$ is valid.

The axioms in (11) are not valid, but do hold when when the relation R_a has a certain shape.

(11) **T** $\Box_a p \to p$ if $g\ \hat{R}_a\ g, \forall g$
D $\Box_a p \to \Diamond_a p$ if $\hat{R}_a$ is a function

The conditions on axioms **4**, **5** and **B** are just restatements in relational terms.

4 Translation into the finite state calculus

The finite state calculus is an algebra of regular sets of strings and regular relations between strings that was designed for use in computational phonology and morphographemics (Kaplan and Kay, 1994; Beesley and Karttunen, 2003). Current implementations allow for the definition of functions on regular sets and relations (Hulden, 2009; Lindén et al., 2009; Karttunen, 2010). Such definitions are used here to construct a model for EpiKAT inside the finite state calculus (Figures 5&6).

The space of worlds is a set of ordinary (as opposed to guarded) strings. Bit sequences (sequences of 0's and 1's) encode atoms, and as before, these alternate with event symbols to encode a world. In this construction, w_3 of the example is the string "1 0 ah 1 0 bh 1 0".[9]

Terms in the finite state calculus are interpreted as sets of strings, or for relational terms, as relations between strings. Computationally, the sets and relations are represented by finite state acceptors. As used here, a program in the Fst language of the finite state calculus is a straight-line program that defines a sequence of constants naming sets,

```
St
  Atoms such as 0110.
UnequalStPair
  Sequence of two unequal atoms such as 0110 0111.
define Wf0 ~[$ UnequalStPair];
  String that doesn't contain a non-matching pair of atoms.
define Squash St -> 0 || St _;
  Rewrite relation deleting the second of two atoms.
define Cn(X,Y)
    [[[X Y] & Wf0] .o. Squash].l;
  KAT product.
define Kpl(X)
    [[[X+] & Wf0] .o. Squash].l;
define Kst(X) St | Kpl(X);
  KAT Kleene plus and Kleene star. The Fst operation | is
  union.
```

Figure 5: EpiKAT product and star defined in Fst.

constants naming relations, and functions (defined as macros) mapping one or more regular sets or relations to a regular set or relation.

Translating the EpiKAT terms 0, 1, b, and e is straightforward: convert the atoms as previously described, and decorate the events e with their compatible atoms. For example, a_1 becomes an Fst term denoting $\{01a_h01\}$. Union $(+)$ and intersection $(\wedge)$ operators are directly translated to Fst's built-in analogous operators ((`|`) and (`&`)). Propositional complement in EpiKAT becomes Fst set difference $(-)$ from the set of all worlds.

Defining KAT product using Fst's set-lifted string concatenation (denoted by juxtaposition `X Y`) requires more care. Naively concatenating strings with atoms (Boolean vectors) at both ends doubles atoms at the juncture, and does not enforce the requisite atom equality. To implement KAT product, we define the binary operation `Cn`, which concatenates strings in the string algebra, removes strings with non-matching atoms, and then deletes the second of two atoms to create a set of well-formed guarded strings (Figure 5). `Wf0` is the set of ordinary strings with only equal pairs of atoms, as defined using Fst's containment operator `$`. The `Squash` relation uses Fst's rewrite notation to delete atoms (elements of `St`) that are preceded by another atom.[10] This relation is applied via the composition (`.o.`) and codomain (`.l`) operators.

Kleene plus is defined in a similar way, using Kleene plus in the string algebra, with checks for equality of atoms and deletion of atoms. Kleene

[9] The Fst programming language of the finite state calculus allows for multicharacter symbols, so that ah is a single symbol in the string.

[10] This is a non-equal length regular relation. The finite state calculus includes such relations, and they can be used with relation composition and relation domain and co-domain. They are restricted in that the complement and set difference for non-equal length relations is not defined. Epistemic alterative relations are equal-length relations.

```
define RelKpl(R)
  Squash.i .o. Wf0 .o. [R+] .o. Wf0 .o. Squash
               c       b     a     b     c
```

a Relational Kleene plus in the string algebra
b Constrain domain and co-domain to contain no unmatched atoms.
c Reduce doubled atoms to a single atom in the domain and co-domain.

```
define RelKst(R) [St.x.St] | Kpl(X);
```

The Fst operation `.x.` is Cartesian product. `R.i` is the inverse of relation `R`.

Figure 6: Definition in Fst of the Kleene concatenation closure of a relation between guarded strings.

star is defined from Kleene plus the well-formed atoms `St`, which implements 1.

Finally, world alternative relations are constructed in Fst. Combining the agent relation with the effect formula gives a relation on events decorated with compatible atoms. Then the corresponding relation on worlds is constructed using the closure of the relational concatenation product operation. The concatenation product $R\ S$ of two relations R and S is the set of pairs of the form $\langle x_1x_2, y_1y_2\rangle$, where $x_1\ R\ y_1$, and $x_2\ S\ y_2$. In Fst, `R+` is the closure of relation `R` with respect to this operation. Figure 6 defines the corresponding operation on sets of guarded strings as encoded in Fst.[11] The epistemic alternative relation on worlds for an agent is then defined as the KAT relation concatenation closure `RelKst` of the decorated-event alternative relation for the agent.

5 Bounded Lazy Interpretation

EpiKAT specifications also target lazy lists in Haskell, rather than the direct interpretation as sets. Using lists sidesteps checking the set invariant (that elements are unique) for large sets, such as $\bullet^*$, and laziness allows us to delay computing these large sets until they are actually needed. To sidestep the infiteness of models, we parameterize the interpretation function on a positive integer n and only produce guarded strings of length n or less.

The bounded interpretation into lists of strings is very similar to the unbounded interpretation into sets of strings, except for the (lazy) bounds checking. The full details are shown in Figure 7. First note that when $n = 0$, the denotation is empty, written []. Terms of the form 0, 1, e, and ψ have the same denotation as before, translated into a list (written $\lfloor S \rfloor$, for a set S). We compute atoms

[11] Relation concatenation in Fst differs from relation composition (`.o.`), and the closure under discussion here is the closure of the former rather than the latter.

$$
\begin{array}{rcl}
(\!|p|\!)^{\mathsf{B},\varphi,\zeta}_0 & \triangleq & [] \\
(\!|0|\!)^{\mathsf{B},\varphi,\zeta}_n & \triangleq & [] \\
(\!|1|\!)^{\mathsf{B},\varphi,\zeta}_n & \triangleq & \lfloor \mathcal{A}^{\varphi}_{\mathsf{B}} \rfloor \\
(\!|e|\!)^{\mathsf{B},\varphi,\zeta}_n & \triangleq & [\alpha e \beta \mid \alpha\ [\![\zeta_e]\!]^{\varphi}\ \beta] \\
(\!|b|\!)^{\mathsf{B},\varphi,\zeta}_n & \triangleq & \lfloor \mathcal{A}^{b\psi}_{\mathsf{B}} \rfloor \\
(\!|p+q|\!)^{\mathsf{B},\varphi,\zeta}_n & \triangleq & (\!|p|\!)^{\mathsf{B},\varphi,\zeta}_n \mathbin{+\!\!+} (\!|q|\!)^{\mathsf{B},\varphi,\zeta}_n \\
(\!|p;q|\!)^{\mathsf{B},\varphi,\zeta}_n & \triangleq & ((\!|p|\!)^{\mathsf{B},\varphi,\zeta}_n \diamond (\!|q|\!)^{\mathsf{B},\varphi,\zeta}_n)|_n \\
(\!|p^*|\!)^{\mathsf{B},\varphi,\zeta}_n & \triangleq & [] + ((\!|p|\!)^{\mathsf{B},\varphi,\zeta}_n \diamond (\!|p^*|\!)^{\mathsf{B},\varphi,\zeta}_{n-i})|_n \\
 & & \text{where } i{=}\max\{1,\min\{|g||g{\in}(\!|p|\!)^{\mathsf{B},\varphi,\zeta}_n\}\} \\
(\!|\neg p|\!)^{\mathsf{B},\varphi,\zeta}_n & \triangleq & (\!|\bullet^*|\!)^{\mathsf{B},\varphi,\zeta}_n \setminus (\!|p|\!)^{\mathsf{B},\varphi,\zeta}_n \\
(\!|\Diamond_a p|\!)^{\mathsf{B},\varphi,\zeta}_n & \triangleq & [g' \mid g'\ \hat{R}_a\ g, \texttt{for}\ g\ \texttt{in}\ (\!|p|\!)^{\mathsf{B},\varphi,\zeta}_n]
\end{array}
$$

Figure 7: Bounded interpretation using lazy lists

using BDDs, which concisely represent boolean functions (Lee, 1959).

We lift the remaining operators (except Kleene star) to their list equivalents: union becomes list append (written $+\!\!+$); fusion product is lifted to lists instead of sets, negation is implemented using list difference ($\setminus$), and the modal operator lifts the alternative relation over lists of strings[12]. The only caveat to these direct interpretations is that we lazily restrict the strings to have size $\leq n$, written as $l|_n$ for a list of guarded strings l.

The denotation of p^* uses the fact that p^* and $1 + p; p^*$ are equivalent, and decrements the size threshold on the recursive denotation of p^* by i, where i is the length of the longest (nonzero) string in the denotation of p, making sure to filter out guarded strings that are too long.

6 Syntax-semantics interface

English sentences are mapped to terms in the logical language via a semantically interpreted multimodal categorial grammar, consisting of a lexicon of words, their categorial types, and interpretations in a logical lambda language. The grammar covers basic statives (*it's heads*), *that-* and *whether-*complements, negation, and predicate and sentence conjunction. Figure 8 gives the lexicon.[13] The grammar and semantics are optimized for a simple fragment of English concerned with clausal

[12] Figure 7 depicts this using the list comprehension notation, which is analogous to set builder notation, except that it is written using square brackets. Element order is evoked by the keyword `for`, rather than using the unordered $\forall$.

[13] Category symbols use Lambek/Bar-Hillel notation for slashes, so that $(d\backslash t)/(d\backslash_D t)$ combines with $d\backslash_D t$ on the right to give a value that combines with d on the left to give t. In the semantics, lambda abstractions with multiple parameters are written $\lambda x\ y.\ e$ rather than $\lambda x.\lambda y.e.\ d$ is the category of expletive *it*.

ITEM	TYPE	SEMANTICS
Statives with expletive subject		
heads	$d\backslash_D t$	$\lambda x.\ \bullet^* h$
tails	$d\backslash_D t$	$\lambda x.\ \bullet^* t$
Agents		
Amy	e	$\hat{R}_a$
Bob	e	$\hat{R}_b$
Auxiliary verbs and expletive subjects		
it	d	*dummy*
is	$(d\backslash t)/(d\backslash_D t)$	$\lambda P\, x.\ P\, x$
it's	$(t/(d\backslash_D t)$	$\lambda P\, x.\ P\, x$
isn't	$t/(d\backslash_D t)$	$\lambda P\, x.\ \neg P\, x$
doesn't	$(e\backslash t)/(d\backslash_V t)$	$\lambda P\, x.\ \neg P\, x$
Tensed attitude verbs		
knows	$(e\backslash t)/_M t$	$\lambda p\, R.\ p + \Diamond_R p$
believes	$(e\backslash t)/_M t$	$\lambda p\, R.\ \Diamond_R p$
Base form attitude verbs		
know	$(e\backslash_V t)/_M t$	$\lambda p\, R.\ p + \Diamond_R p$
believe	$(e\backslash_V t)/_M t$	$\lambda p\, R.\ \Diamond_R p$
Complementizers		
that	$(((e\backslash t)/_M t)\backslash(e\backslash t))/t$	$\begin{pmatrix}\lambda p\, m\, R.\\ \neg(m\,(\neg p)\,R)\end{pmatrix}$
whether	$(((e\backslash t)/_M t)\backslash(e\backslash t))/t$	$\begin{pmatrix}\lambda p\, m\, R.\\ \neg(m\,(\neg p)\,R)\\ +\neg(m\,p\,R)\end{pmatrix}$
Complementizers for base form verbs		
that	$(((e\backslash_V t)/_M t)\backslash(e\backslash_V t))/t$	$\begin{pmatrix}\lambda p\, m\, R.\\ \neg(m\,(\neg p)\,R)\end{pmatrix}$
whether	$(((e\backslash_V t)/_M t)\backslash(e\backslash_V t))/t$	$\begin{pmatrix}\lambda p\, m\, R.\\ \neg(m\,(\neg p)\,R)\\ +\neg(m\,p\,R)\end{pmatrix}$
Conjunction		
and	$(t\backslash t)/t$	$\lambda p\, q.\ p \wedge q$
or	$(t\backslash t)/t$	$\lambda p\, q.\ p + q$
and	$(((e\backslash t)\backslash(e\backslash t))/(e\backslash t)$	$\lambda p\, q\, x.\ p(x) \wedge q(x)$
or	$(((e\backslash t)\backslash(e\backslash t))/(e\backslash t)$	$\lambda p\, q\, x.\ p(x) + q(x)$

Figure 8: Categorial grammar lexicon, showing word form (column 1), a categorial type (column 2), and a semantic translation in EpiKAT extended with lambda abstraction (column 3).

complementation. The agent names *Amy* and *Bob* contribute the epistemic alternative relations for those agents, rather than individuals. The root verb *believe* contributes existential modal force. The complementizers *that* and *whether* are the heads of their dominating clauses, and assemble an alternative relation, modal force, and proposition contributed by the complement. These complementizers introduce the dual via two negations, in order to express universal modal force.

Multimodal categories such as $\backslash_D$ and $\backslash_M$ are used to control the derivation—phrases with these top-level slashes can only combine syntactically as arguments. The semantic translations in the third column of Figure 8 use the logical language, incremented with lambda. The body of $\lambda x.\ \bullet^* h$, which is the semantic lexical entry for *heads*, is a term denoting the set of all worlds where the coin is heads, expressed as the set of all guarded strings that end with a Boolean valuation where the primitive proposition h (it's heads) is true. The body of $\lambda p.\lambda R.\Diamond_R p$, which is the semantic lexical entry of *believes*, is an term denoting the pre-image of the world-alternative relation contributed by the subject. This is not the right semantics for *Amy believes that it's heads*, because it has an existential modality $\Diamond_R p$, rather than an universal modality $\Box_R p$. This is corrected by the complementizer *that*, which introduces the dual.

Sentences are parsed with a chart parser for categorial grammar. The semantics for complex phrases are obtained by application of semantic translations, accompanied by beta reductions that eliminate all lambdas in logical forms for clauses. In consequence, the semantic term translating a sentence is an EpiKAT logical term. Such a term designates a set of possible words (guarded strings). By way of example, (12a) is an English sentence with conjunction and several levels of clausal embedding. Using the grammar and parser, the sentence is mapped to the term in (12b). (12c) shows a simplified logical form constructed from (12b) using syntactic equivalences. Either term is compiled in an implementation of the finite state calculus to a finite state machine with 49 states and 110 edges, which accepts a countably infinite set of worlds.[14] In this way the methodology "directly" represents the set of worlds denoted by (12a).

(12) a. Amy knows that it's tails and doesn't believe that it's heads and believes that Bob believes that it's heads.

b. $\neg(\neg \bullet^* t + \Diamond_a \neg \bullet^* t) \ \wedge$
$\neg\neg\Diamond_a \neg \bullet^* h \ \wedge$
$\neg\Diamond_a \neg\neg\Diamond_b \neg \bullet^* h$

c. $\mathcal{K}_a \bullet^* t \ \wedge\ \neg\mathcal{B}_a \bullet^* h \ \wedge\ \mathcal{B}_a\mathcal{B}_b \bullet^* h$

d. Bob believes that it's heads.

e. $\mathcal{B}_b \bullet^* h$

Sentence (12d) is assigned a logical form that is syntactically equivalent to (12e). Logical relations between propositions are checked in the finite

[14] Machine sizes need not be small, especially as the cardinality of B increases. A certain EpiKAT model with fourteen primitive Booleans has the set of worlds representented by a finite state machine with 184,794 states and 257,881 edges.

state calculus by checking set-theoretic relations between sets of worlds. Propositional entailment from p to q is decided by checking in an interpreter for the finite state calculus whether $p - q$ is non-empty. In the model defined by Figure 2, proposition (12c) entails proposition (12e).

Another way of using the Fst interpreter is to display worlds in a given proposition. (13) shows the three words of length three in proposition (12c). In the first one, Bob looks at heads, then Amy looks at heads, then Amy secretly turns the coin, believing that she is turning it from heads to tails because she believed after two steps that the coin was heads. Intuitively sentence (12a) is true in this scenario.

(13) $10\ b_h\ 10\ a_h\ 10\ a_{ht}\ 01$
$10\ b_h\ 10\ a_{ht}\ 01\ a_t\ 01$
$10\ a_h\ 10\ b_h\ 10\ a_{ht}\ 01$

7 Discussion

EpiKAT is designed for research in linguistic semantics, for computational linguistic research on model-theoretically grounded semantics, and for course modules on intensional formal semantics. Currently, the grammar covers statives, *that*- and *whether*-complements, negation, and conjunction. There are straightforward extensions to additional linguistic phenomena, such as tense and perfective aspect (14a), and the combination of metaphysical modality and prospective aspect (14b).

(14) a. Amy has learned that Bob had learned that it's heads.
b. Amy might learn that it's heads.

The model framework is a constructive branching-time framework with metaphysical and epistemic modalities, which will be applicable in linguistic semantic research on combinations of tense, metaphysical modality, and epistemic complementation (Thomason, 1984; Abusch, 1998; Condoravdi, 2002). Connections with research in a finite state framework on temporal constitution of events remain to be explored (Fernando, 2004, 2007; Carlson, 2009).

The grammar formalism uses the standard approach in categorial grammar to map between strings (or trees or derivations) and logical terms (Steedman, 2000; Bozsahin, 2012), which comes with practically useful computational implementations (Barker and Shan, 2005; Bozsahin, 2021).[15] While this approach is simple and attractive for our applications, it would be possible to use EpiKAT with other grammar formalisms that support lambda extensions of logical languages.

The semantics presented here is exclusively concerned with events and worlds. Semantic type systems for natural language usually include a type for individuals (Montague, 1975; Gallin, 1975). In the coin example, the individuals are hidden in the primitive event symbols such as a_h and b_h. This situation would get worse in a more elaborate model with more agents and multiple coins. The solution to this should be to base the world construction on grounded or parametric event terms such as $\mathbf{peek}(\mathbf{a},\mathbf{k},\mathbf{h})$, for "agent $\mathbf{a}$ peeks at heads-up coin $\mathbf{k}$", rather than atomic event symbols. This approach is found in research on situation calculus (Reiter, 2001). How to incorporate it in the scheme for EpiKAT specifications and into the computational parts of the proposal is a topic for future research. The ramifications of quantification or abstraction over individuals is unknown. Rooth (2017) develops an approach based on introducing markers for witnesses for discourse referents in the construction of worlds.

The development here is concerned with defining concrete computable possible worlds models, and applying them in natural language semantics. Logical and computational characterizations of EpiKAT, such as sound and complete axioms, coalgebras, and relations to other logics, warrant further investigation.

Source code, examples, and instructions for building and running EpiKAT are available under an open-source license at `https://github.com/ericthewry/epikat`.

References

Dorit Abusch. 1998. Generalizing tense semantics for future contexts. In *Events and Grammar*, pages 13–33. Springer.

Alexandru Baltag, Lawrence S Moss, and Slawomir Solecki. 1999. The logic of public announcements, common knowledge, and private suspicions.

[15] Our grammar uses basic categorial grammar, not additional features such as combinators, type raising, continuations, or multimodal slashes that interact with the grammar in non-trivial ways. These features become relevant in more sophisticated grammars. Multimodal categories are used for phrases such as a tenseless verb phrase that semantically are functional, but combine in the grammar only as arguments.

Chris Barker and Chung-chieh Shan. 2005. Reference parser for explaining crossover and superiority as left-to-right evaluation. semanticsarchive.net/Archive/TI1M2UxN.

Kenneth R Beesley and Lauri Karttunen. 2003. *Finite State Morphology*. Chicago and CSLI.

Cem Bozsahin. 2012. *Combinatory Linguistics*. De Gruyter.

Cem Bozsahin. 2021. Ccglab. github.com/bozsahin/ccglab, accessed January 2021.

Lauri Carlson. 2009. *Tense, Mood, Aspect, Diathesis*. Book ms., University of Helsinki.

Alexander Chagrov. 1997. Modal logic.

Ernie Cohen. 1994. Hypotheses in kleene algebra. *Unpublished manuscript*.

Jacob Collard. 2018. Finite state reasoning for presupposition satisfaction. In *Proceedings of the First International Workshop on Language Cognition and Computational Models*, pages 53–62.

Cleo Condoravdi. 2002. Temporal interpretation of modals: Modals for the present and for the past. *The Construction of Meaning*, page 88.

Amina Doumane, Denis Kuperberg, Damien Pous, and Pierre Pradic. 2019. Kleene algebra with hypotheses. In *Foundations of Software Science and Computation Structures*, pages 207–223, Cham. Springer International Publishing.

Tim Fernando. 2004. A finite-state approach to events in natural language semantics. *Journal of Logic and Computation*, 14(1):79–92.

Tim Fernando. 2007. Observing events and situations in time. *Linguistics and Philosophy*, 30(5):527–550.

Tim Fernando. 2017. Intensions, types and finite-state truthmaking. In *Modern Perspectives in Type-Theoretical Semantics*, pages 223–243. Springer.

Daniel Gallin. 1975. *Intensional and Higher-order Modal Logic: With Applications to Montague Semantics*. North-Holland Publishing Company.

Chris Hardin. 2002. On the elimination of hypotheses in kleene algebra with tests. Technical report, Cornell University Ithaca.

Charles Antony Richard Hoare. 1969. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):576–580.

George Edward Hughes, Max J Cresswell, and Mary Meyerhoff Cresswell. 1996. *A New Introduction to Modal Logic*. Psychology Press.

Mans Hulden. 2009. Foma: a finite-state compiler and library. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Demonstrations Session*, pages 29–32. Association for Computational Linguistics.

Ronald M Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331–378.

Lauri Karttunen. 2010. Update on finite state morphology tools. *Ms., Palo Alto Research Center*.

Dexter Kozen. 1997. Kleene algebra with tests. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 19(3):427–443.

Dexter Kozen. 2001. Automata on guarded strings and applications. Technical report, Cornell University.

Dexter Kozen and Frederick Smith. 1997. Kleene algebra with tests: Completeness and decidability. In *Computer Science Logic*, pages 244–259, Berlin, Heidelberg. Springer Berlin Heidelberg.

Chang-Yeong Lee. 1959. Representation of switching circuits by binary-decision programs. *The Bell System Technical Journal*, 38(4):985–999.

David Lewis. 1986. *On the Plurality of Worlds*. Blackwell.

Krister Lindén, Miikka Silfverberg, and Tommi Pirinen. 2009. Hfst tools for morphology–an efficient open-source package for construction of morphological analyzers. In *International Workshop on Systems and Frameworks for Computational Morphology*, pages 28–47. Springer.

John McCarthy. 1963. Situations, actions, and causal laws. Technical report, Stanford CS.

Richard Montague. 1975. *Formal Philosophy. Selected papers of Richard Montague*. Yale. Edited by Richmond Thomason.

Raymond Reiter. 2001. *Knowledge in Action: Logical foundations for specifying and implementing dynamical systems*. MIT press.

Mats Rooth. 2017. Finite state intensional semantics. In *IWCS 2017-12th International Conference on Computational Semantics-Long papers*.

Mark Steedman. 2000. *The Syntactic Process*. MIT press Cambridge, MA.

Richmond H Thomason. 1984. Combinations of tense and modality. In *Handbook of Philosophical Logic*, pages 135–165. Springer.

Hans Van Ditmarsch, Wiebe van Der Hoek, and Barteld Kooi. 2007. *Dynamic Epistemic Logic*. Springer.

Hans Van Ditmarsch, Joseph Y Halpern, Wiebe van der Hoek, and Barteld Pieter Kooi. 2015. *Handbook of Epistemic Logic*. College Publications.