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 defined, where possible worlds models are guarded string models in an epistemic extension of Kleene Algebra with Tests. The account is framed as a formal language Epik (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*. These are often assumed to be things with the same nature and complexity as the world we occupy (Lewis, 1986). But alternatively, one can work with small idealized models, in order to illustrate and test ideas. The point of this paper is to scale up toy or idealized models to countable sets of worlds, and to constructive and computable modeling of epistemic alternatives for agents. We describe a systematic way of defining such models, and illustrate how to apply them in natural language semantics. The focus is on epistemic semantics and clausal embedding. The fundamental move is to identify possible worlds with strings of primitive events, so that propositions are sets of strings. An advantage in this is that it allows for a mathematical description of an algebra of propositions, coupled with a

computational representation using either lazy lists of strings, or finite state machines that describe sets of strings.

The approach taken here 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* characterizes algebras with elements corresponding to propositions and event types in our application (Kozen, 2001). *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).

A running example of an event-sequence model is *The Concealed Coin*. Amy and Bob are seated at a table. There is a coin on the table under a cup, heads up. The coin could be heads-up H or tails-up T, and neither agent knows which it is. This initial situation is possible world w_1. We can construct additional worlds by sequencing events like those shown in (1).

(1) a_1 Amy peeks at H, by tipping the cup. Bob sees she's peeking, but not what she sees.

b_1 Bob peeks at H.

a_0 Amy peeks at T.

b_0 Bob peeks at T.

a_{01} 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_{10} Amy secretly turns the coin from H to T.

b_{01} Bob secretly turns the coin from T to H.

b_{10} Bob secretly turns the coin from H to T.

The worlds $w_1, \ldots, w_4$ in (2) are examples constructed from the events in (1). Juxtaposition indicates the order and occurence of events, for example w_3 indicates that after the events of w_2, event b_1 occurs, fully, this indicates beginning in world w_1, Amy surreptitiously peeking at a heads-up coin, and then Bob doing the same.

$$(2) \quad w_2 \triangleq w_1 a_1 \quad w_3 \triangleq w_2 b_1 \quad w_4 \triangleq w_3 a_{10} b_{01} b_1$$

The truth values for English sentences are observed at the end of the event sequences specified by $w_1, \ldots, w_4$, where 0 stands for falsity and 1 for truth.

(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.

The events come with pre-conditions and post-conditions. Amy can turn the coin from heads to tails only if the coin is heads-up, so a_{10} has the pre-condition of the coin being heads up. Once she turns the coin over, tails must be face-up, so a_{10} has the post-condition of the coin being tails-up. Let h be the Boolean proposition that the coin is heads up and t be the Boolean proposition that the coin is tails-up. Then pre- and post-conditions can be described by Boolean formulas, with h being the pre-condition of a_{10} and t being the post-condition. This is expressed using an operator ":" (read "and next") that pairs Boolean formulas. The formula $h{:}t$ describes a_{10} (Amy turning the coin from heads to tails) as happening only in an h state, and concluding in a t state. Events don't have to change state: the event a_0 (Amy peeking at tails) can happen only in a t state, and does not change the state $(t : t)$.

However a coin cannot be showing both heads and tails! Currently the precondition h of a_{10} only says that heads must be showing, and says nothing

---

$$
\begin{aligned}
&\text{state formulas} \quad\quad (a \in \mathsf{B}) \\
&\rho, \sigma, \varphi \quad ::= \quad a \mid 0 \mid 1 \mid \rho + \sigma \mid \rho\,\sigma \mid \bar{\rho} \\
&\text{effect formulas} \\
&\zeta, \eta \quad\quad ::= \quad \rho : \sigma \mid \zeta + \eta \mid \zeta \,\&\, \eta \mid \bar{\zeta} \\
&[\![\rho : \sigma]\!]^{\varphi} \quad \triangleq \quad \mathcal{A}_{\mathsf{B}}^{\rho\varphi} \times \mathcal{A}_{\mathsf{B}}^{\sigma\varphi} \\
&[\![\zeta + \eta]\!]^{\varphi} \quad \triangleq \quad [\![\zeta]\!]^{\varphi} \cup [\![\eta]\!]^{\varphi} \\
&[\![\zeta \,\&\, \eta]\!]^{\varphi} \quad \triangleq \quad [\![\zeta]\!]^{\varphi} \cap [\![\eta]\!]^{\varphi} \\
&[\![\bar{\zeta}]\!]^{\varphi} \quad \triangleq \quad \mathcal{A}_{\mathsf{B}}^{\varphi} \times \mathcal{A}_{\mathsf{B}}^{\varphi} \setminus [\![\zeta]\!]^{\varphi}
\end{aligned}
$$

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.

about the fact that tails must be face-down, indicated by the formula $\bar{t}$. We will further restrict the feasible conditions for our actions by restricting the space of valuations for our formulas.

A sequence such as $\bar{h}t$ can be viewed both as a formula and as a valuation of primitive propositions, which we use to describe world state. The primitives are listed in fixed order, and left unmarked (indicating true) or marked with the overbar (indicating false). Since a coin is heads or tails but not both, we want to allow the valuations $h\bar{t}$ and $\bar{h}t$, and disallow ht and $\bar{h}\bar{t}$.[1] This is enforced by a *state formula*, which is a Boolean formula, in this case the one given on the second line of (4). Where B is a set of state primitives and φ is a constraint over B, $\mathcal{A}_{\mathsf{B}}^{\varphi}$ is the set of valuations of B that make formula φ true. Valuations are called atoms, because they correspond to the atoms of a Boolean algebra of tests (Kozen, 2001).

Formulas that describe pre- and post-conditions are *effect formulas*. They are interpreted as defining relations between atoms, as defined in Figure 1. The atoms they relate are constrained by the state formula as well. For the heads-tails example, let the state formula and the effect formula for a_1 (Amy peeking at heads) be as specified in (4). Then $\mathcal{A}_{\mathsf{B}}^{\varphi}$ and the relation on atoms for the event a_1 are as given at the bottom in (4).

(4)	B	$\{h, t\}$
	state formula φ	$h\bar{t} + \bar{h}t$
	effect formula ζ for a_1	$h : h$
	$\mathcal{A}_{\mathsf{B}}^{\varphi}$	$\{h\bar{t}, \bar{h}t\}$
	$[\![\zeta]\!]^{\varphi}$	$\{\langle h\bar{t}, h\bar{t}\rangle\}$

---

[1]Two Booleans constrained by a state formula, rather than one Boolean, are stipulated here to illustrate state formulas.

```
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 -> o0 + o1
event a01 t:h        b01 -> o0 + o1
event b10 h:t      agent bob
event b01 t:h        <sim. swap a and b>
```

Figure 2: Epik 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.

2 Epistemic guarded string models

Epik 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 Epik 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_1 (Bob peeks at heads), is that when b_1 happens, for Amy either b_1 or b_0 (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_{01} (she turns the coin over from heads to tails) are a_{01} and a_{10}, 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_{10}. Conversely, When Bob secretly turns the coin over, in event b_{01} or b_{10}, she doesn't know anything has happened, so her alternative is the "no-operation" or "no-information" event event o. Bob's event relation is symmetric.

This paper focuses on defining a concrete possible worlds model from an Epik specification. The models are an extension of guarded-string models

for Kleene Algebra with Tests (KAT). This algebraic theory has model classes including guarded 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 algebra described by Figure 2, P is the set of events $\{a_1, a_0, b_1, b_0, a_{01}, a_{10}, b_{01}, b_{10}, o\}$, and B is $\{h, t\}$. As we already saw in (4), $\mathcal{A}_B^{\varphi}$ 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, $w_1 = H$, $w_2 = Ha_1H$, $w_3 = Ha_1Hb_1H$, and $w_4 = Ha_1Hb_1Ha_{10}Tb_{01}Hb_1H$.

The discussion of (2) mentioned building worlds by incrementing worlds with events. This is accomplished in guarded string models with fusion product $\diamond$, a partial operation that combines two guarded strings, subject to the condition that the atom at the end of the the first argument is identical to the atom at the start of the second one. (5) gives some examples.

(5) $\quad H\,b_1\,H \;\diamond\; H\,a_1\,H = H\,b_1\,H\,a_1\,H$
$\quad T\,b_{01}\,H \;\diamond\; T\,a_1\,T = undefined$

Rather than individual guarded strings, elements of a guarded string model for KAT are sets of guarded strings. In our application, these elements have the interpretation of propositions, which are 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. We instead impose the constraints coming from the state and effect formulas. (6) defines the well-formed guarded strings determined by an Epik 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 the effect constraint on e_i.[2]

---

[2]An alternative is to define equations such as $\bar{\phi} = 0$ (from the state formula ϕ) and $a_1 = ha_1h$ (from the effect formula $h : h$ for event a_1), 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

(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) $\alpha_i \in \mathcal{A}_B^\varphi$ $(0 \leq i \leq n)$, 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 Kripke frame determined by an Epik specification is the set of well-formed guarded strings. At this point, we could say that any set of worlds is a proposition, so that the set of propositions is the power set of the set of worlds (Montague, 1975; Gallin, 1975). We will instead define a more restrictive set of propositions corresponding to the regular sets of strings. This is deferred to the next section. Certain sets of well-formed guarded strings have the addional interpretation of event types. An event-type is something that can "happen" is different worlds. For example, a_1 corresponds to the event type $\{H\,a_1\,H\}$, and a_0 corresponds to the event type $\{T\,a_0\,T\}$. Event types need not be singletons.

The construction so far defines a set of worlds from an Epik 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 $\langle W, R_1, ..., R_n, K \rangle$ 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).[3]

An Epik 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 e' is and event-alternative to e for a.[4] This needs

---

[3] 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.

[4] 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

---

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

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

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 1, but with an additional agent Cal. The base world $Tb_0 Tc_0 T$ is one where the coin is tails, and first Bob looks at tails, and them Cal looks at tails. The first event b_0 has the alternatives b_0 and b_1 for Amy, and the second event c_0 has the alternatives c_0 and c_1 for Amy. This results in four combinations $b_0 c_0$, $b_0 c_1$, $b_1 c_0$, and $b_1 c_1$. But some of these worlds are bogus, such as $b_0 c_1$, which says that Bob peeked at the coin and saw tails, and Cal peeked at the same coin and impossibly saw heads. These contradictions are filtered by post- and pre-conditions of events in the alternative world, so that the set of alernatives for Amy in $Tb_0 Tc_0 T$ is $\{Tb_0 Tc_0 T, Hb_1 Hc_1 H\}$, with two world-alternatives instead of four.

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 opera-

---

of the equivalence class.

---

models in natural language semantics.

tion $\cdot$ 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.[5] 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 proposositional semantic values of English sentences. (8) gives some examples. To explain the first one, $\bullet$ 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 our existential 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 in (9).[6]

(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 Epik specification.[7] The agents in Figure 1 do not always have reliable beliefs, because of the possibility of secret turning.

---

[5]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.

[6]Deeper analyisis 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.

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

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

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

Guarded String Interpretation. A term p of the logical language is interpreted as a set of guarded strings $[\![p]\!]^{\mathsf{B,P},\varphi,\zeta}$, where superscript captures dependence on an Epik specification. Figure 4 defines the interpretation. The interpretation $[\![1]\!]^{\mathsf{B,P},\varphi,\zeta}$ of the multiplicative identity 1 is the set of atoms that satisfy the state constraint φ. Where b is a primitive Boolean, $[\![b]\!]^{\mathsf{B,P},\varphi,\zeta}$ is the set of atoms that satisfy the state constraint and where b is true. Where e is a primitive event, $[\![e]\!]^{\mathsf{B,P},\varphi,\zeta}$ is the set of guarded strings that have the form of e flanked by compatible atoms, as determined by the event 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 Epik specification $\mathsf{B}, \mathsf{P}, \varphi, \zeta$, term p (as defined syntactically in Figure 3) is interpreted as a set of guarded strings $[\![p]\!]^{\mathsf{B,P},\varphi,\zeta}$. Let $K^{\mathsf{B,P},\varphi,\zeta}$ be the sets that are interpretations of terms. Then $\langle K^{\mathsf{B,P},\varphi,\zeta}, +, \cdot, *, \bar{}, 0, 1, \neg, \Diamond_{a_1}, ..., \Diamond_{a_n} \rangle$ is a concrete guarded string interpretation for the signature of epistemic KAT, with operations as in Figure 4 (e.g. the binary operation $+$ is union, and the unary operation $\Diamond_a$ is pre-image relative to $\hat{R}_a$). This provides a concrete n-agent general modal frame $\langle [\![\bullet^*]\!]^{\mathsf{B,P},\varphi,\zeta}, \hat{R}_1, ..., \hat{R}_n, K^{\mathsf{B,P},\varphi,\zeta} \rangle$. The frame 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.

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). Some of these standard axioms (see (10)) hold all the time, i.e. they are *valid*, and the remaining axioms are valid when $\hat{R}_a$ has a certain shape. The axioms in (11) have a nontrivial condition validity condition.

(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.

(11) **T** $\Box_a p \to p$ if $g\ \hat{R}_a\ g, \forall g$

      D $\Box_a p \to \Diamond_a p$ if $g \in \mathrm{dom}(\hat{R}_a), \forall g$

      4 $\Box_a p \to \Box_a\Box_a p$ if $\hat{R}_a$ idempotent

The condition on **4** is essentially just a restatement of the theorem in relational terms. The conditions on the remaining standard modal axioms, **B** and **5**, are of a similarly trivial flavor.

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 computationonal 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 of a model for epistemic KAT inside the finite state calculus. We describe our translation from Epik logical terms to Fst terms here.

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 `10 a1 10 b1 10`.

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, constants naming relations, and functions (defined as macros) mapping one or more regular sets or

```
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: Definition in Fst of KAT product and star.

relations to a regular set or relation.

Translating the Epik terms 0, 1, b, and e are straightforward: we simply convert the atoms as previously described, and decorate the events e with their compatible atoms. For example a_1 becomes an Fst term denoting $\{01a_101\}$. Fst has built-in operations of union (`|`), and intersection (`&`), which are the sum and intersection operations in the guarded string algebra as represented in Fst. Fst set difference (`-`) is used to define propositional complement as the difference between the set of all worlds and the argument.

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. See Figure 5. `Wf0` is the set of ordinary strings that does not contain unequal 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. [8] This relation is applied via the relational composition (`.o.`) and codomain (`.l`) operators.

KAT Kleene plus is defined in a similar way using Kleene plus in the string algebra, with checks

---

[8]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.

for equality of atoms and deletion of atoms. KAT
Kleene star is defined from KAT Kleene plus and
the multiplicative identity, which is the set of well-
formed atoms `St`.

It remains to define an epistemic alternative rela-
tion on worlds for each agent. The relevant infor-
mation in Figure 2 is a relation between bare events
for each agent. This determines a relation between
bare events decorated with compatible atoms. In
Fst, we use the closure of the concatenation product
operation on relations to lift a relation on decorated
events for an agent to the corresponding relation
on worlds. The concatenation product $R\,S$ of two
relations R and S is the set of pairs of the form
$\langle x_1 x_2, y_1 y_2 \rangle$, 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.[9] The epistemic alternative relation on worlds
for an agent is then defined as the KAT relation con-
catenation closure `RelKst` of the decorated-event
alternative relation for the agent.

5 Bounded Lazy Interpretation

We also implement the semantics of Epik terms
using lazy lists in Haskell, rather than the direct
interpretation as sets. Using lists sidesteps check-
ing the set invariant (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 posi-
tive 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 bounds checking. The

---

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

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

Figure 7: Bounded interpretation using lazy lists

full details are shown in Figure 7. First note that
when $n = 0$, the denotation is empty, denoted $[]$.
Terms of the form 0, 1, e, and ψ have the same
denotation as before, translated into a list (denoted
$\lfloor S \rfloor$, for a set S). We compute atoms 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 al-
ternative relation over lists of strings[10]. The only
caveat to these direct interpretations is that we re-
strict the operators to have size $\le n$, denoted 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 logi-
cal language via a semantically interpreted multi-
modal categorial grammar, consisting of a lexicon
of words, their categorial types, and interpretations
in a logical lambda language. The grammar cov-
ers basic statives (*it's heads*), *that-* and *whether-*
complements, negation, and predicate and sentence
conjunction. Figure 8 gives the lexicon.[11] The on-

[Conjunction missing!]

---

[10] Figure 7 depicts this using the list comprehension nota-
tion, 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 $\forall$.

[11] 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

line supplement includes a version of this grammar that can be run using Barker and Shan's chart parser for categorial grammar (Barker and Shan, 2005). The grammar and semantics are optimized for a simple fragment of English concerned with clausal 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 $\square_R p$. 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 Epik 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 from constructed form (12b) using logical equivalences. Either term is compiled in

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 Epik extended with lambda abstraction (column 3).

an implementation of the finite state calculus to a finite state machine with 10 states and 23 edges, which accepts a countably infinite set of worlds.[12] In this way the methodology "directly" represents the set of worlds denoted by (12a).

(12) a. It's tails and Amy knows that Bob knows that Amy knows whether it's heads.

b. $\bullet^* t \wedge \neg(\neg(\neg(\neg(\neg(\neg(\neg \bullet^* h \vee \diamond_a \neg \bullet^* h) + \neg(\bullet^* h \wedge \diamond_a \bullet^* h)) + \diamond_b \neg(\neg(\neg \bullet^* h \vee$

---

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*.

$\diamondsuit_a \neg \bullet^* h) + \neg (\bullet^* h \wedge \diamondsuit_a \bullet^* h)))) + \diamondsuit_a \neg (\neg (\neg (\neg (\neg \bullet^* h \vee \diamondsuit_a \neg \bullet^* h) + \neg (\bullet^* h \wedge \diamondsuit_a \bullet^* h)) + \diamondsuit_b \neg (\neg (\neg \bullet^* h \vee \diamondsuit_a \neg \bullet^* h) + \neg (\bullet^* h \wedge \diamondsuit_a \bullet^* h)))))$

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

 d. Amy knows that it's tails.

 e. $\bullet^* t \wedge \neg \diamondsuit_a \neg \bullet^* t$ $(\equiv \mathcal{K}_a \bullet^* t)$

Sentence (12d) is assigned the logical form (12e) by the grammar. Logical relations between propositions are checked in the finite state calculus by checking set-theoretic relations between sets of worlds. For instance entailment $p \rightarrow q$ 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, the propositions (12b) and (12e) are independent. They are equivalent in a version without secret flipping.

7 Discussion

Epistemic KAT is designed for use in research in linguistic semantics, for research in computational linguistics on model-theoretically grounded semantics, and for education at the level of a second graduate course in formal semantics, covering intensionality. There are straightforward extensions to additional linguistic phenomena, such as tense and perfective aspect (13a), and the combination of metaphysical modality and prospective aspect (13b).

(13) 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 modality and epistemic modality, 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 on temporal constitution of events in a related formal setting remain to be explored (Fernando, 2004, 2007; Carlson, 2009).

The syntactic part of the grammar formalism uses categorial grammar to map between strings (or trees or derivations) and logical translations. This way of using a categorial grammar is standard (Steedman, 2000; Bozsahin, 2012), and there are computational implementations for it (Barker and Shan, 2005; Bozsahin, 2021).[13] While this way of proceeding is simple and attractive for the applications we are interested in, it would be possible to use our semantics with other frameworks that have a semantics with a lambda extension of a logical language.

The semantics presented here is exclusively concerned with events and worlds. Semantic type systems for natural language usually include a type for individuals (Gallin, 1975). In the coin example, the individuals are hidden in the primitive event symbols such as a_1 and b_1. 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 event terms such as **look(a,k,1)**, for "agent **a** see that coin **k** is heads", rather than atomic event symbols. This way of proceeding is found in research on situation calculus (Reiter, 2001). How to incorporate it in the scheme for Epik specifications and into the computational parts of the proposal is a topic for future research. In this it is not clear whether it is possible to introduce quantification over individuals, together with individual constants. Rooth (2017) develops an approach to a positive answer, based on introducing markers for witnesses for discourse referents in the construction of worlds.

Epik is closely related to Kleene Algebra with Hypotheses KA+H (Cohen, 1994), which permits user-specification of futher equations beyond the axioms of KA. In practice, these hypotheses are used to express domain-specific properties of the events, such as the commutativity. We can model both the state formulae and the event formulae as KA+H hypotheses. 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 $a \equiv \sum_{\varphi, \varphi' \in [\![\zeta_a]\!]} \varphi; a; \varphi'$. In general, the Horn theory of KA+H is undecidable, however many special cases are (Doumane et al., 2019; Hardin, 2002; Kozen and Smith, 1997).

The key facet that differentiates Epik from KA+H is the modality. Each alternative relation represents a set of uni-directional rewrites that must be applied for that agent, rather than the universal, optional rewrites represented by the KA+H

---

[13]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.

hypotheses. It may be possible to encode Epik's alternative relations into KA+H hypotheses, but such an encoding is future work.

The development here is concerned with defining concrete computable possible worlds models, and applying them in natural language semantics. Issues for further investigation are mathematical characterizations of epistemic KATs, e.g. sound and complete axioms and coalgebrae.

Source code, examples, and instructions for Epik are available at https://github.com/ericthewry/epik.

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.

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.

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. 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.