Thesis Proposal
Changing Beliefs in a Changing World

João G. Martins

April 2018

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
André Platzer, Chair, co-advisor (CMU)
João Leite, co-advisor (FCT)
Frank Pfenning,
Stephen Brookes,
Wiebe van der Hoek, external chair

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

Keywords: dynamic epistemic logic, dynamic doxastic logic, cyber-physical systems, differential dynamic logic, modal logic, dynamic logic

Abstract

We find ourselves on the cusp of a revolution in the transportation industry, with an expectation for the impending widespread adoption of self-driving capabilities, promising faster and more efficient deliveries, more comfort and better bottom lines.

Critical to achieving this vision, however, is large scale deployment of vehicles with self-driving capabilities, or *cyber-physical systems* (CPSs), into roadways and airways already teeming with other human- and computer-controlled vehicles. This deployment cannot, and indeed *should not* be acceptable until the safety of the controllers in charge of new CPSs is credibly proven. Already several incidents, some fatal, cast a shadow of doubt onto the public's belief about whether heterogeneously controlled systems can safely function within the same environments.

While significant progress has been made in formally verifying the safety of CPS models, we argue that current models often do not accurately represent real systems because of an implicit assumption of perfect knowledge of the real-world. In reality, sensor noise and incomplete or even counterfactual beliefs create ample possibilities for controller decisions that lead to safety violations. We claim that the notion of *belief* can capture the real life phenomena of noisy observations and faulty human intuition, leading to appropriately informed decisions. A failure to take belief into account results in a critical gap between theoretical and practical safety of CPSs.

One such example, and inspiration for this thesis, is the crash of Air France Flight 447, in which the interaction between beliefs, control and physics led an aircraft with no actuator malfunctions from safety to tragedy in under four minutes. We propose to address this safety-critical intersection by developing the technical underpinnings of a change towards *belief-triggered controllers*, where decisions are grounded in imperfect *perceptions* of the real world rather than the real world itself.

In this proposal, we develop a logic whose semantics are capable of adequately capturing the interleaved changes of both belief and world-state that characterize real-world CPSs: a logic for *changing beliefs in a changing world*. We prove the soundness of a sequent calculus which captures the fundamental behaviors of doxastic change within the context of CPSs, and propose to generalize it to augment its applicability to more complex scenarios.

We further propose to conduct case studies of small scenarios exhibiting safety-violation symptoms similar to those of AF-447, or fragments of the AF-447 incident itself. The expected outcome of such case studies are formal safety proofs of revised *belief-triggered* controllers, built from state-triggered controllers whose flaws that led to safety violations on display in real scenarios. These revisions highlight safety-critical changes that can be adopted by legislative, regulation and training authorities to create better policies leading to a safer world.

Ultimately, this thesis proposal argues for the necessity of a paradigm shift in which belief becomes a first-class citizen of CPS design, verification and legislation processes, and provides the first technical steps on that long road.

Contents

Chapter 1

Introduction

While self-driving transportation, such as autopilots in aircraft, has been around for decades, only recently has it garnered interest for more widespread application. With a universe of 263,610,219 registered road vehicles reported by the USA's Bureau of Transportation in 2015[1], the potential for conversion to self-driving vehicles is significant indeed.

Many prototypes for more mainstream application, like Google's and Uber's self-driving cars, have already been plying roads in relatively small numbers for testing purposes for years. The Tesla Model 3 promotional materials state that it is equipped with many more sensors and computational power, "enabling full self-driving in almost all circumstances, at what we believe will be a probability of safety at least twice as good as the average human driver." Amazon's Prime Air initiative, wherein unmanned drones will deliver smaller packages directly to customers in thirty minutes or less from "drone hives", effectively guarantees autonomous vehicles will soon be pervasive in the air as well as on the ground.

The impending promise of a majority of vehicles on road and air being autonomous raises serious questions about safety, however. These questions are being, and *ought* to be, asked at the consumer level (e.g. "Should I buy a self-driving car?"), at the corporate level (e.g. "Should my company invest in a self-driving fleet?") and at the government level (e.g. "What regulations should be instituted to ensure safe autonomous behavior?"). Ultimately, the success of the pending autonomous revolution hinges on the public and governmental perception of both safety and efficiency achieved by transitioning to self-driving vehicles.

Several years' worth of self-driving car tests have provided some statistical evidence for at least some level of safety: so far, human intervention accounts for most safety incidents, including the few fatalities. This is, however, enough to begin casting a shadow of doubt on a policy of wholesale acceptance of self-driving vehicles. Indeed, the jump from a few hundreds of autonomous vehicles to multiple millions, in a heterogeneous environment with humans of all skill levels and self-driving algorithms from different developers, will result in a number of interactions that is orders of magnitude higher than what we have seen or studied thus far. This begs the question: in a state space this vast, how reliable do we consider statistical methods to be, given their known proclivity towards missing low probability edge cases [**?**]?

It is thus important to advocate not only for efficient statistical methods of verification, but

[1] `https://www.rita.dot.gov/bts/sites/rita.dot.gov.bts/files/publications/national_transportation_statistics/html/table_01_11.html`

also for more comprehensive methods of formal verification for hardware like sensors, low-level software controlling actuators, and high-level algorithms for autonomous vehicles. Logic-based formal methods for verification have the advantage that they exhaustively check the state space, at the expense of requiring a different skill-set and the challenge of developing formal models that conservatively approximate reality.

1.1 Knowledge, Belief and Autonomous Vehicles

One of the critical lacunas in the formal verification of theoretical models of cyber-physical systems is that current modeling languages implicitly encourage the idea that the real state of the world is known.

Thus, in theoretical CPS models, one might expect to find a condition such as $d < 20$, where d represents distance in meters to a traffic light, as a precondition to a self-driving car braking. The elephant in the room is, of course, that the algorithm controlling the car cannot possibly know the car's exact distance to the traffic light. It must instead rely on distance sensors or, e.g. a GPS system, which, like all sensors, are noisy by nature. In the case of a human driver, distance estimation is known to be far from perfect [18]. Assuming that the sensors, including a driver's eyes, are functioning correctly, we may talk of knowledge (epistemics): these are beliefs which may be uncertain, but can never be counterfactual. To represent such uncertainty, we may look at d as having multiple possible values somewhere around, and including, the real distance.

Belief (doxastics) is a weaker notion. It is a conviction that something is true when it may not be. This may occur because of a malfunctioning sensor whose readings are significantly off from reality, or, for example, if the traffic light is occluded by a large truck, and the driver's estimate of their position is erroneous. Thus, one may *believe* falsities, but it is impossible to *know* falsities.

For the purposes of this thesis proposal, there are only minor technical differences between knowledge and belief, and hence the two terms will be used somewhat interchangeably, particularly when addressing related work.

To explain the importance of these notions in the realm of autonomous vehicles, and thus cyber-physical systems, we return to Air France Flight 447, which is a prime, if tragic, example of what the consequences of failing to take beliefs into account.

1.1.1 Use Cases

Air France Flight 447 Air France 447 was a scheduled flight from Rio de Janeiro to Paris on June 1 2009. It crashed into the Atlantic ocean, with no survivors, after a temporary speed sensor failure, as a direct result of pilot inputs due to confusion. If, after sensor failure, the aircraft had retained a level-flying attitude with stable engine thrust, safety would have been maintained.

We now present an abbreviated timeline of events based on the final report from the Bureau d'Enquêtes et d'Analyses pour la Sécurité de l'Aviation Civile (BEA) [5].

1. While flying near its Maximum Permissible Altitude, the aircraft hits a particularly cold spot. As a result, the pitot tubes, sensors responsible for calculating velocity, freeze and begin reporting inconsistent speeds.

2. Because of inconsistent sensors readings, the aircraft disconnects the autopilot, starts blaring stall warnings, and reverts the control mode to *alternate law*, which significantly increases the effect of pilot inputs, which under regular law are damped for passenger comfort.
3. The two pilots, surprised, appear to reach opposing understandings of what is happening. The pilot-in-control believes the airplane is nose-down, low and fast. The pilot-not-in-control believes the aircraft is stalling.

   Neither realize the aircraft is stable, that the cause of autopilot disconnection is a sensor malfunction, or that their inputs have a much larger effect than normal due to alternate control law.
4. The pilot in control pulls the nose up in an attempt to correct what they believe is an overspeed situation. This raises the aircraft above the Maximum Permissible Altitude, causing loss of lift and thus actually initiating a stall which can temporarily be physically felt, increasing confusion about what the real state of the world is.
5. Over the next four minutes, there is significant mode confusion about which pilot is control, which flight control law is active, and what precisely is happening. Throughout, at least one of the pilots holds counterfactual beliefs about the airplane's state. In their differing understandings, the pilots take control away from each other, contradict each other's inputs, and neither are fully aware of the other's actions. The combination of both pilots' inputs keeps airplane in a stall.
6. Seconds before impact with the ocean and total loss of life, the pilots, too late, realize what is in fact happening.

The incredible thing about this incident is that there were no aircraft actuator flaws: performance was normal. The *only direct factor* leading to a safety violation was pilot input. This indicates that the human controllers made the decisions contrary to safety. However, it would be disingenuous and in very poor taste to claim, as many verification approaches do, that one of the pilots was behaving adversarially, i.e. that they wanted to crash the plane. The a reasonable, true-to-life model has both agents making reasonable decisions *given what they believe*, just as in real life.

The issue with the pilots, the controllers of the airplane, was not one of intention or ability. It was instead a failure of *belief*, which underlies and informs every decision of every controller, due to both overwhelming and insufficient information at the same time.

More self-aware controllers, in a cockpit whose instruments are explicitly tailored to the doxastic requirements of surprised pilots, or whose checklists incorporate an understanding of how pilot beliefs evolve under uncertainty, would have 1) more easily understood the state of the system, 2) more readily agreed on such an understanding, and 3) cooperatively decided on a *well-informed* and *safe* course of action.

This particular safety violation occurred because, in suddenness and confusion, the pilots developed conflicting and counterfactual beliefs about the state of the aircraft, then made decisions that were unsafe, but sensible, given those beliefs. In one sentence: the cause of AF-447's accident is fundamentally related to doxastics.

To fully grasp the complexity of incident, one must accurately model intersection of mode

confusion, airplane physics and decisions made based on false beliefs initiated the stall and then maintained it until impact. This intersection must be fully understood if aviation authorities are to prevent accidents *before* they happen.

Flaps malfunction Similar doxastic phenomena show up during flight training[2], resulting in simpler scenarios that are more amenable to first attempts at applying a logic for formal verification.

In this case, a student pilot practicing touch-and-go maneuvers pulled the flaps lever without visually checking their deployment. In reality, the flaps failed to deploy, resulting in a trajectory below the desired glide path. The pilot held beliefs inconsistent with the real world during the initial approach, and kept taking actions inappropriate to the state of the airplane. Pilot intuition kicked in near the runway, when the difference between expected and perceived altitude hit a threshold and became too obvious to ignore. Corrective actions were finally taken, and this particular incident ended safely, if rather bumpily. It can, however, still be used to highlight the important role that belief plays in the control and safety of CPSs.

It is worth spending some moments on how general the notion of belief can be. In the traffic light example, it was used to refer to what a digital controller would infer from its distance sensors, which often come with attached known noise bounds. In the touch-and-go incident, the pilot's eyes were their sensors into their altitude, and, like digital sensors, were seen to be imperfect. Thus, it makes sense for a controller, human or otherwise, to make decisions which incorporate expected errors into its decision making procedures. At the other end of the spectrum, belief was used to represent the absence of sensors at the time of AF-447's speedometer malfunction and the night-time conditions providing zero visibility.

Belief was also used to refer to a pilot's general perception of the world without regard to specific sensors, e.g. the student pilot's beliefs about their flaps, or the pilots' (flawed) beliefs about the state of AF-447.

There are also some more subtle uses: a human operator's intuition. This can go beyond belief about the current state of the world, and into how the world *works and changes*, not *is*. For instance, when the student pilot moved the flaps lever, they fully expected the flaps to come down. A more suspicious or cautious pilot may not have those expectations, and may decide a visual confirmation is always in order. We wouldn't be surprised if this particular student pilot now belongs to that latter group! There are many more phenomena within this category: one pilot may believe an aircraft stalls under a high-G turn, whereas another may think it merely turns tighter; and a pilot has some basic understanding of flight dynamics, such that even without instruments or visibility, we may expect them to hold beliefs about where the airplane has ended up given where it started.

A comprehensive treatment of belief is capable of addressing all of these through a sufficiently expressive language for doxastic change.

[2] `http://airfactsjournal.com/2014/03/flaps-anyone-strange-things-can-happen/`

1.1.2 Formal verification and belief

As a starting point for formal verification of CPSs *without* doxastics, some logics such as the Differential Dynamic Logic $\mathsf{d}\mathcal{L}$ [16] have been successfully applied to the verification of surgical robots [11], aircraft collision avoidance procedures [10], and autonomous car control [12, 14]. By successful application, we mean that verification attempts have uncovered counter-examples to safety in algorithms previously believed to be safe, even by statistical methods, as well as providing a mathematical proof that current or modified algorithms are, in fact, safe.

The Air France 447 incident, however, proves that belief is safety-critical in the real world. Thus, the absence of doxastics is one significant shortcoming to current approaches, both statistical and logical: models for CPS controllers most often fail to include reasoning about what they know, or *think* they know. In short, about their beliefs.

There is significant research in understanding the notions of knowledge and belief from a logical perspective. Dynamic epistemic logics, a summary of which can be found in [6], capture not only such notions, but also provide us with syntax which may be used to update an agent's knowledge. The field of belief revision, described in the seminal paper [1], purports to do the same within a more declarative and arguably less expressive framework.

To the best of our knowledge (no pun intended!), however, there is currently no work that integrates complex changes in beliefs interleaved with complex changes in the physical world within a context appropriate for CPSs, e.g. with continuous domains and unobservable events.

1.2 Proposal

We have brought attention to the need for comprehensive methods of formal verification to be present in CPS design and development, emergency checklists, safety regulations, and above all, for this thesis, for belief to take a central role in those efforts. Failing to address the critical notion of belief, risks not bridging the the gap between theoretical and practical safety evidenced by the crash of AF-447.

To address this challenge, we propose to develop Dynamic Doxastic Differential Dynamic Logic, or $\mathsf{d}^4\mathcal{L}$ for short, as an extension of the differential dynamic logic $\mathsf{d}\mathcal{L}$. Using a logical approach enables reasoning about the entire state space, and by using $\mathsf{d}\mathcal{L}$ as a starting point, we can be confident in the logic's ability to handle the difficult challenges of CPS verification. We also follow in the footsteps of $\mathsf{d}\mathcal{L}$ in developing a sequent calculus to prove important properties.

While the fields of world-change and belief-change are fairly well understood in isolations, their intersection brings up some interesting challenges.

- Because the world change must interleave with belief change, we propose to unify the languages of both. The hybrid programs of $\mathsf{d}\mathcal{L}$, describing CPS behavior, can be augmented with a learning operator $L_p(\alpha)$, whose programs α describe belief change. To represent doxastic properties, we may add to our formulas the traditional doxastic modalities $B_p(\cdot)$ for belief and $P_p(\cdot)$ for possibility.
- Most logics for changing knowledge or belief often require propositional or finite domains, because their axiomatizations work by creating formulas based on underlying semantic structures. However, CPSs require continuous time and uncountably infinite domains.

Thus, the semantics of $\mathsf{d}^4\mathcal{L}$ will be $\mathbb{R}$ valued and consequently, the techniques used to prove a sound calculus will not be able to take direct inspiration from the current state of the art.

- The belief states of agents in dynamic epistemic logics are often inextricably connected. This is in direct conflict with the important requirement that agents are able to learn something new without it affecting the belief state of another agent. This is precisely what happened during AF-447, when one pilot would change their input unobserved by the other. This is a very significant challenge that requires a fundamental rethinking of the underlying semantic structures, and as such, we consider it outside the scope of this thesis. Instead, we will restrict ourselves to a logic where agents cannot hold beliefs about other's beliefs. This is reducible to a single-agent logic, which is what we present in this document, and then easily extendable to multiple agents whose doxastic states are distinct.

In this proposal, we present a sound sequent calculus for $\mathsf{d}^4\mathcal{L}$, which captures the fundamentals, but not the entirety, of doxastic change under infinite domains, as part of CPS models. Our proposal is to augment the generality of our proof rules.

With more general proof rules, and to attest to the usefulness of $\mathsf{d}^4\mathcal{L}$, we intersperse the proposal with running examples and embryonic case-studies, directly taken from or at least inspired by real life scenarios exhibiting phenomena similar to those of AF-447. These will serve as the basis for some proposed case studies, which will establish that $\mathsf{d}^4\mathcal{L}$ is not merely a fun exercise in theory but also a meaningful contribution to solving the problem of modern-day CPS verification.

Chapter 2

Preliminaries

In this chapter, we report on the state of the art in the logical approach to verifying cyber-physical systems, and in the logics of changing knowledge and belief that serve as the inspiration to integrate with the former.

2.1 Differential Dynamic Logic

The logic dL [15, 16, 16, 17] uses hybrid programs to model and describe the operation of cyber-physical systems as entities moving in continuous space over continuous time. The dL sequent calculus enables purely syntactic proofs on formulas representing important features of CPS such as safety and liveness.

The primary mode of establishing CPS safety in dL is by the validity of a formula

$$\mathit{pre} \rightarrow [(\texttt{ctrl; plant})^*]\,\mathit{safe}$$

It states that starting from safe preconditions *pre*, the system performs a controller decision `ctrl` followed by a continuous-time system evolution in `plant`, all of which is repeated any number of times as indicated by *, after which a safety property *safe* holds. For a self-driving car approaching an intersection at which the traffic light may be green (value 1) or red (value 0), in dL, the controller `ctrl`, whether human driver or computer algorithm, will choose to accelerate or brake according to the state of the traffic light. The `plant` updates the car's position with the laws of physics, based on the driver's acceleration/braking input.

$$\big((?light = 0; acc := -A) \cup (?light = 1; acc := A)\big);\ x' = vel, vel' = acc$$

The above dL hybrid program models an "if-then-else" controller based on the traffic light, followed by a differential equation describing linear motion: position x changes according to the velocity vel, which in turn changes depending on the acceleration acc. The variable acc is not updated by `plant`. Instead, it is set by the controller `ctrl`. Physically, this models the positioning of the gas or brake pedals. This is one of the many ways in which the interleaving of discrete dynamics (e.g. instantaneous driver choices and inputs) influence the continuous dynamics (e.g. the car navigating through the physical world) in dL.

2.1.1 Syntax

To faithfully model real cyber-physical systems, dL terms must be $\mathbb{R}$-valued and include real arithmetic. State variables x represent the state of the world and logical variables X are introduced by quantifiers to discharge reasoning about continuous time, finding witnesses for certain modalities, as well as many other uses.

Thus, the syntax of dL terms is that of real arithmetic. It contains sum, subtraction, multiplication, from which integer exponents can be derived, e.g. $x^2 = x \times x$, and division. In what follows, let $\mathbb{V}$ be a countable set of state variable names and Σ a countable set of logical variable names.

Definition 1 (dL Terms)**.** *The terms θ of dL, with $\otimes \in \{+, -, \times, \div\}$, $X \in \Sigma$, and $x \in \mathbb{V}$, are defined as follows.*

θ	$::=$		*(Expression)*
		X	*(Logical variable)*
	$\mid$	x	*(State variable)*
	$\mid$	$\theta \otimes \theta$	*(Function)*

The formulas of dL include propositions such as $\theta_1 < \theta_2$ and $\theta_1 = \theta_2$, and logical quantifiers $\forall X.\phi$ which assist in decomposing, e.g., the infinite nondeterminism of the differential equations of linear motion $x' = vel, vel' = acc$ into equivalent formulas quantifying over continuous time $\forall T \geq 0.\phi$. Most importantly, they also include the dynamic modality $[\alpha]\,\phi$, meaning ϕ is true for all states reachable from with hybrid program α. Its dual $\langle\alpha\rangle\,\phi$ can be defined as $\neg\,[\alpha]\,\neg\phi$, and requires ϕ to be true for at least one execution of α. These two *dynamic modalities* represent CPS execution.

Definition 2 (dL Formulas)**.** *The formulas ϕ of dL, with θ a term, and $X \in \Sigma$, and α a hybrid program, are defined as follows.*

ϕ	$::=$		*(Formula)*
		$\phi \vee \phi$	*(Disjunction)*
	$\mid$	$\neg\phi$	*(Negation)*
	$\mid$	$\forall X.\phi(X)$	*(Quantification)*
	$\mid$	$[\alpha]\,\phi$	*(Dynamic modality)*
	$\mid$	$\theta < \theta$	*(Comparison)*
	$\mid$	$\theta = \theta$	*(Equality)*

In the above, and throughout the rest of this document, the pattern $\phi(X)$ is used whenever it is helpful to explicitly remind ourselves that occurrences of X are possible and likely in ϕ. The formula $\phi(\theta)$, in this context, would be formula ϕ with all occurrences of X syntactically substituted by a term θ.

Finally, the hybrid programs of dL define the discrete and continuous change of CPS. Two atomic operators are used for this purpose. First, there is assignment $x := \theta$, which performs instantaneous physical change, such as a flip-switch taking control away from the autopilot. Second, differential equations $x' = f(x)\ \&\ \chi$ describe continuous motion over time, subject to an evolution domain constraint χ. Since the duration of continuous evolution is nondeterministic, the most common usage of the evolution domain constraint is to limit the amount of time the system may evolve without controller input, e.g. $t' = 1, x' = vel\ \&\ t < 10$ limiting to 10 seconds the amount of time a car may move before the controller gets to reconsider their choices.

The test $?\phi$ transitions iff the condition ϕ is satisfied. Tests are crucial in defining which states a program may reach, and thus which must be considered by the modality $[\alpha]\,\phi$. For example, $[?1 = 0]\,1 = 0$ is vacuously true since there are no reachable states. As previously illustrated, test is often found as "if-then" statements and as preconditions for controller actions, as in $?alt < 100; climb := 1$, in which if an aircraft is low, the controller decides to climb.

Non-atomic programs include sequential composition $\alpha; \beta$, which allows programs to be executed sequentially, without which it would be hard to describe any sort of real system.

The addition of nondeterministic choice $\alpha \cup \beta$ can potentially increase the number of reachable states, and permits "if-then-else" statements, such as with $(?alt < 1000; climb := 1) \cup (?alt \geq 1000; climb := -1)$, where an airplane attempts to remain at around 1000 meters by climbing up or down depending on whether they are above or below the threshold.

Finally, nondeterministic repetition α^* allows a program to be arbitrarily iterated.

Definition 3 (dℒ Hybrid Programs)**.** *Hybrid programs, with $x \in \mathbb{V}$, ϕ, χ being formulas, are defined as follows.*

α	$::=$	*(Hybrid Program)*
	$x := \theta$	*(Discrete physical state change)*
$\mid$	$x' = f(x)\&\chi$	*(Continuous physical state change)*
$\mid$	$?\phi$	*(Test)*
$\mid$	$\alpha; \alpha$	*(Sequential composition)*
$\mid$	$\alpha \cup \alpha$	*(Non-deterministic choice)*
$\mid$	α^*	*(Non-deterministic repetition)*

With well defined syntax and a more solid understanding of the behavior of hybrid programs, we may revisit the general pattern for CPS *pre* $\rightarrow [(\texttt{ctrl; plant})^*]$ *safe*, which is now even clearer. First, the controller makes decisions based on the current state of the world, then the system is allowed to evolve continuously for a bounded amount of time before the controller reevaluates their choices with the new state of the world. Finally, repeating this behavior an unbounded number of times ensures the behavior is fundamentally safe, rather than conservative enough to maintain safety for only a limited amount of time.

2.1.2 Semantics

The semantics of dℒ are given using states ω and variable assignments η, much like in first-order logic for real arithmetic. A state $\omega : \mathbb{V} \rightarrow \mathbb{R}$ is a function from the set of variable names $\mathbb{V}$ to value $v \in \mathbb{R}$, i.e. $\omega(x)$ is the current value of state variable x. A variable assignment $\eta : \Sigma \rightarrow \mathbb{R}$ works similarly for logical variables, with $\eta(X)$ being the value of logical variable X. The interpretation of terms and formulas is as done as usual in first-order logic.

Definition 4 (Term interpretation)**.** *Let ω be a state, and η be a variable assignment, with $\otimes \in \{+, -, \times, \div\}$, $x \in \mathbb{V}$ and $X \in \Sigma$. Then, the interpretation of terms is defined as follows.*

- $val_\eta(\omega, x) = \omega(x)$ *for state variable* x
- $val_\eta(\omega, X) = \eta(X)$ *for logical variable* X
- $val_\eta(\omega, \theta_1 \oplus \theta_2) = val_\eta(\omega, \theta_1) \oplus val_\eta(\omega, \theta_2)$

The meaning of formulas is, again, as usual for first order real arithmetic, with programs given meaning through a reachability semantics used for the dynamic modality. Furthermore, η^v_X

is notation for updating η such that the value of X is v.

Definition 5 (Evaluation of formulas)**.** *Let $\omega = \langle r, W, V, s\rangle$ be a state, η be a variable assignment, θ_1, θ_2 be terms, ϕ_1, ϕ_2 be formulas and α be a hybrid program. Then, the valuation of formulas is given as follows.*

$$
\begin{array}{lll}
val_\eta(\omega, \theta_1 = \theta_2) = \mathtt{T} & \textit{iff} & val_\eta(\omega, \theta_1) = val_\eta(\omega, \theta_2) \\
val_\eta(\omega, \theta_1 < \theta_2) = \mathtt{T} & \textit{iff} & val_\eta(\omega, \theta_1) < val_\eta(\omega, \theta_2) \\
val_\eta(\omega, \phi_1 \vee \phi_2) = \mathtt{T} & \textit{iff} & val_\eta(\omega, \phi_1) = \mathtt{T} \textit{ or } val_\eta(\omega, \phi_2) = \mathtt{T} \\
val_\eta(\omega, \neg\phi) = \mathtt{T} & \textit{iff} & val_\eta(\omega, \phi) = \mathtt{F} \\
val_\eta(\omega, \forall X.\phi) = \mathtt{T} & \textit{iff} & \textit{for all } v \in \mathbb{R}, val_{\eta^v_X}(\omega, \phi) = \mathtt{T} \\
val_\eta(\omega, [\alpha]\,\phi) = \mathtt{T} & \textit{iff} & \textit{for all } (\omega, \omega') \in \rho_\eta(\alpha), val_\eta(\omega', \phi)
\end{array}
$$

Finally, the semantics of programs are given as reachability between states. Perhaps the most complex of these is the differential equation. The semantics finds the solution of the diff. eqs., applies it to the current physical world to obtain an initial value problem, and uses the solution to determine what PD-models are reachable through continuous time evolution (so long as the evolution domain constraint χ is satisfied throughout).

Definition 6 (Transition semantics)**.** *Let ω be a $\mathsf{d}\mathcal{L}$ state, $x \in \mathbb{V}$, θ be terms and ϕ, χ be formulas. The transition relation for hybrid programs is defined as follows.*

- $(\omega, \omega') \in \rho_\eta(x := \theta)$ *iff* $\omega' = \omega[x \mapsto val_\eta(\omega, \theta)]$
- $(\omega, \omega') \in \rho_\eta(x' = \theta \,\&\, \chi)$ *iff there is $y : D \to \mathbb{R}$, with $n = |\mathbb{V}|$, a solution to the diff. eq. in $D = [0, \tau^f]$, s.t. $\omega' = \omega_x^{y(\tau^f)}$ and for all $0 \le \tau \le \tau^f$, $val_\eta\left(\omega_x^{y(\tau)}, H\right) = \mathtt{T}$*
- $(\omega, \omega) \in \rho_\eta(?\phi)$ *iff* $val_\eta(\omega, \phi) = \mathtt{T}$
- $\rho_\eta(\alpha_1; \alpha_2) = \rho_\eta(\alpha_2) \circ \rho_\eta(\alpha_1) = \{\omega_3 :$ *there is* $\omega_2.(\omega_1, \omega_2) \in \rho_\eta(\alpha_1)$ *and* $(\omega_2, \omega_3) \in \rho_\eta(\alpha_2)\}$
- $\rho_\eta(\alpha_1 \cup \alpha_2) = \rho_\eta(\alpha_1) \cup \rho_\eta(\alpha_2)$
- $(\omega, \omega') \in \rho_\eta(\alpha^*)$ *iff there is a $n \in \mathbb{N}$ such that $(\omega, \omega') \in \rho_\eta(\alpha^n)$, with α^n being α sequentially composed n times.*

2.1.3 Calculus

The logic $\mathsf{d}\mathcal{L}$ has a sound calculus that has been successfully applied to many case studies ranging from cars, trains, airplanes and medical robots [16, 17]. Its many iterations, which we will omit in the interest of space, can be found throughout the literature [15, 16, 16, 17].

The process of proving a safety formula *pre* $\to [(\mathtt{ctrl;\ plant})^*]$ *safe* in the $\mathsf{d}\mathcal{L}$ calculus is multi-faceted. To begin with, traditional logical rules can be applied for the $\to$ logical connective. After that, however, it is important to identify invariants of the program `ctrl; plant`, which hold independently of how many times it is repeated. By having such invariants imply safety, we can prove the system to be safe.

The controller `ctrl` is most often handled by successively applying rules that decompose complex programs into progressively smaller, simpler subprograms. The treatment of differential equations in `plant` is done through the use of differential invariants, a continuous analogue to the invariants of nondeterministic repetition. If known, the solution to the differential equation

may be used instead. Eventually, by reaching atomic programs such as assignment, test and differential equations, the dynamic modality can be reduced entirely to formulas of first-order real arithmetic. Checking the validity of such formulas is known to be decidable.

2.1.4 The logic $\mathsf{d}\mathcal{L}$ and belief

The framework of $\mathsf{d}\mathcal{L}$ does not explicitly capture beliefs, and there is no syntax defined for that purpose. In this sense, it is one of those aforementioned languages which encourages the notion that real world state is known.

However, there are ways in which limited aspects of belief, or uncertainty, may be captured. To do that, we introduce the program $x := *$ as syntactic sugar for $x' = 1; x' = -1$. This *nondetermistic assignment* $x := *$ assigns any value in $\mathbb{R}$ to x.

For the car approaching a traffic light, we may follow the assignment with a test, e.g. $d := *; ?d^2 - 100^2 \leq 10^2$, and thus force our controller to consider the case that the distance d is within the interval $[90, 110]$.

This is a way in which uncertainty about the sensor reading may be incorporated into a model. Crucially, however, it relies on subtle modeling tricks which can be unclear or hard to grasp when reading the program. It also only works with the universal quantification of the box modality, e.g., $[d := *; ?d^2 - 100^2 \leq 10^2]\,\phi$, and cannot be applied for liveness proofs which use the existencial quantification of diamond modalities, e.g. $\langle d := *; ?d^2 - 100^2 \leq 10^2\rangle\,\phi$.

2.2 Belief Revision

The pilots of Air France Flight 447 made mutually incompatible inputs because they held counterfactual perceptions of reality. Belief is the philosophical and logical notion that captures such convictions about the truth of something when it need not necessarily be true.

As situations change in ways that are observable by agents, then so must their belief states. The field that studies how belief states may be altered so as not to produce inconsistencies with previous beliefs is called *belief revision*. In this section, we briefly document the declarative AGM approach to belief revision [1], while cautioning that this field does not also provide the mechanisms with which to describe world change - only to revise beliefs once change has already happened.

In belief revision, an agent's belief state is given by a *belief base* K, which is a set of logical formulas closed under logical deduction. A set of AGM *rationality postulates* propose desirable properties for three operators on belief bases.

- *Belief expansion*, $K + p$, which expands belief base K with proposition p, and all consequences therefrom.
- *Belief contraction*, $K - p$, which contracts belief base K with p, such that p can no longer be derived.
- *Belief revision*, $K * p$, which revises K with a potentially inconsistent proposition p, so that the final belief base is consistent.

Under certain conditions, revision can be reduced to expansion and contraction. First, the belief base is first contracted by $\neg p$, eliminating any potential sources of inconsistencies with p. Then, the belief base is expanded with p.

$$K * p = (K - \neg p) + p$$

We are now equipped with an intuitive understanding of belief revision operators. What follows are the rationality postulates originally proposed for any "sensible" revision operator.

1. Closure: $K * p$ is a belief base. The revision operator should produce belief bases.
2. Success: $p \in K * p$. The revision operator's result should include the new information.
3. Inclusion: $K * p \subseteq K + p$. The revision operator should be more conservative than, but still congruent with, expansion, e.g. by "removing" inconsistencies.
4. Vacuity: if $(\neg p) \notin K$, then $K * p = K + p$. If p is consistent with K, then the revision operator should behave just like expansion.
5. Consistency: $K * p$ is inconsistent only if p is inconsistent or K is inconsistent.
6. Extensionality: if p and q are logically equivalent, then $K * p = K * q$.
7. $K * (p \wedge q) \subseteq (K * p) + q$
8. If $(\neg q) \notin K * p$ then $(K * p) + q \subseteq K * (p \wedge q)$

The two final postulates are not generally considered to have the same level of importance as the first six, and are thus left out of some treatments of AGM [4].

There are further postulates for expansion and contraction, and a robust body of work around this framework for belief revision. Among that body of work is some significant controversy regarding yet another postulate, the recovery postulate $K \subseteq (K - p) + p$, including a caveat by one of the original authors [13], as well as more general criticisms of the approach itself [7]. Ultimately, there is no absolute consensus in this mostly declarative field on what to "declare" to be sensible postulates.

Shortcomings of belief revision for CPS verification In this proposal, we are most interested in describing, in minute detail, how belief changes as agents in the system observe (or don't) new information. We need to be able to explicitly model how uncertainty comes to be. For this, it is useful to have complex language of belief-change that is more expressive than a single revision operator. The properties of the language of belief change should arise naturally from its semantics, and be captured by a sequent calculus amenable to computer-aided human use. Of course, it also becomes much more difficult to establish postulates that are considered sensible *a priori* for more complex languages.

This is not to say that the AGM approach should be entirely discarded. Indeed, there are axiomatizations for the AGM postulates in temporal logic [4], as well as in dynamic epistemic logics [19]. This provides strong evidence that we may capture the general principles of belief revision within dynamic logic frameworks that are more algorithmic and procedural in nature, and thus more amenable to sequent calculi.

2.3 Dynamic Epistemic Logic

We thus turn our attention to the field of Dynamic Epistemic Logics (DEL), which provide just such a framework. This section will show that the dynamic modalities of DELs are more easily integrated with the modalities of $\mathsf{d}\mathcal{L}$.

This section is mostly inspired by the literature overview in [6], as well as more specific publications such as [2, 3, 9, 20, 21] which provide additional technical intuitions and inspirations.

Before diving into technical details, it is worth pointing out that *epistemic* logics focus on *knowledge*. However, there are reasons why epistemic logics are relevant even when the notion of interest is belief. Firstly, much of the research available on dynamic modal logics is for knowledge, not belief. We conjecture the reason for this to be that, under the Kripke semantics that are most often used, the notion of knowledge relies on equivalence relations, whose theoretical properties are well studied and well liked.

Whatever the case, the differences between Kripke-style epistemic or doxastic logics are technically minimal: merely one axiom or graph property. This results in relatively minor challenges when adapting existing epistemic work to the notion of belief, and this justifies the relevance of what comes next.

All of the following logics' semantics are Kripke-style semantics, relying upon commonly used Kripke models for a set of agents $\mathcal{A}$. These models establish a *possible world* framework under which to understand knowledge modalities. Something is *known* by agent $a \in \mathcal{A}$, $K_a(\phi)$, when it is true at all possible worlds reachable by a, and *possible*, $P_a(\phi)$, when it is true in at least one reachable possible world.

A *distinguished* world, chosen from among the possible worlds, determines the real state of the world. Finally, accessibility relations between worlds determine which worlds are reachable from which worlds, for each agent. In epistemic logics, these accessibility relations are assumed to be equivalence relations, which, it is argued, provide a philosophically credible formal treatment of the notion of knowledge.

Returning to the traffic light example, if there are two possible worlds (in the same equivalence relation), one in which the light is red, and one in which it is green, then it ought to be impossible to know whether it is green or red, since both colors are possible, $\neg K_a(light = 0) \wedge \neg K_a(light = 1)$ or $P_a(light = 0) \wedge K_a(light = 1)$. If the two worlds were not accessible form each other, then we would know whether the light was green or red depending on which world was distinguished, $K_a(light = 0) \vee K_a(light = 1)$. If the worlds were in the same equivalence relation, but the light were green in both possible worlds, and it was sunny in one and rainy in another, then we would know that it was indeed green, $K_a(light = 1)$, but would not be able to establish any knowledge about the weather.

Let us then introduce Kripke models formally.

Definition 7 (Kripke models)**.** *Let Prop be a set of atomic propositions. A Kripke model M is a tuple $\langle W, \sim, V \rangle$, where*

- *W is a set of worlds, called the possible worlds*
- *$\sim$ is a set of accessibility (equivalence) relations $\sim_a \subseteq W \times W$ for each agent $a \in \mathcal{A}$*
- *V : Prop $\to 2^W$ is a valuation, which tells us the worlds at which each proposition is true.*

A pointed Kripke model is given by M, s, with $s \in W$ being called a distinguished world*.*

In epistemic logics, the distinguished world serves two purposes. Firstly, it defines what is factually true: the real state of the world. Secondly, it allows an inductive definition of the semantics as we shall see below.

The formulas of epistemic logic are those of propositional logic with the modality $K_p(\phi)$, for agent a knows ϕ. In Kripke semantics, this means that ϕ must hold at every world that a considers possible, i.e., that is reachable by the accessibility relation $\sim_a$.

Definition 8 (Modal semantics)**.** *Modal semantics are given as follows.*

- $M, s \models p$ *iff* $s \in V(p)$
- $M, s \models \neg\phi$ *iff* $M, s \not\models \phi$
- $M, s \models \phi \wedge \psi$ *iff* $M, s \models \phi$ *and* $M, s \models \psi$
- $M, s \models K_a(\phi)$ *iff for every* t *such that* $s \sim_a t$, $M, t \models \phi$

This is a description of static knowledge. But when people communicate and trust has been established, new facts are learned, and knowledge changes. We must therefore talk about dynamic epistemics.

Non-well-founded semantics Before moving on to a presentation of logics based on Kripke models, we would be remiss not to acknowledge work on dynamic epistemic logics that rely on *possibilities* [8, 9]. Possibilities require non-well-founded set theory, but are otherwise equivalent to Kripke models. These particular logics also feature revision or learning operators that change an agent's knowledge states, much like the dynamic epistemic logics that follow.

2.3.1 Public Announcement Logic

Public announcements are one of the fundamental operators of epistemic change. They are announcements made by a trusted source simultaneously to all agents. One example would be a traffic light in a day with perfect visibility: the traffic light "announces" it has turned green, and thus every driver at the intersection comes to know the light is green by altering their epistemic state.

The Public Announcement Logic PAL, a review of which can be found in [3, 6], adds public announcement to propositional epistemic logic as a dynamic modality $[\phi]\,\psi$, stating that ψ must hold after ϕ is publicly announced by a trusted source.

Because knowledge implies physical truth - if something is known, then it must be true in the (distinguished) world - the semantics of the public announcement of ϕ become simple: the underlying Kripke model can exclude every world in which ϕ is false, since we know the distinguished world is not among them. Thus, only the worlds where ϕ holds remain, including the distinguished world.

For the example with the light turning green, with g being the proposition that the light is green, then $[g]\,\psi$ results in all possible worlds in which the light was red, i.e. $\neg g$, being eliminated. Slightly more formally, the semantics are defined as follows:

$$M, s \models [\phi]\,\psi \text{ iff } M, s \models \phi \text{ implies } M|_\phi, s \models \psi$$

where $M|_\phi$ is a restriction of M to only the worlds where ϕ holds.

This logic has been axiomatized:

$$[\phi]\, p \leftrightarrow (\phi \rightarrow p)$$
$$[\phi]\, (\psi_1 \wedge \psi_2) \leftrightarrow ([\phi]\, \psi_1 \wedge [\phi]\, \psi_2)$$
$$[\phi]\, (\psi_1 \rightarrow \psi_2) \leftrightarrow ([\phi]\, \psi_1 \rightarrow [\phi]\, \psi_2)$$
$$[\phi]\, \neg\psi \leftrightarrow (\phi \rightarrow \neg\, [\phi]\, \psi)$$
$$[\phi]\, K_a\, (\psi) \leftrightarrow (\phi \rightarrow K_a\, ([\phi]\, \psi))$$
$$[\phi_1]\, [\phi_2]\, \phi_3 \leftrightarrow ([\phi_1 \wedge [\phi_1]\, \phi_2]\, \phi_3)$$

Notice that the axioms of PAL require a case analysis of formula ψ in $[\phi]\, \psi$, rather than of ϕ. This is a significant departure from the dL calculus that decomposes programs α in $[\alpha]\, \phi$, and will come up in our efforts as well.

The logic PAL also does not describe factual change - only epistemic change - and does not allow agents to learn individually, so that certain changes are unobserved. Furthermore, because the basis for public announcement is knowledge, it is impossible to effect knowledge revision as in belief revision[1], like the traffic light turning red again. The formula $[g]\, [\neg g]\, p$, for instance, has no intuitive meaning.

To allow for processes such as revision, it is important that languages of epistemic change also allow us to expand the number of possible worlds, e.g. adding new worlds in which the light is red once more. The next logics extend PAL with such capabilities.

2.3.2 Epistemic Action Logic

The Epistemic Action Logic EAL [20] builds upon public announcement logic by extending the language of learning new information.

The public announcement ϕ is replaced by an explicit test $?\phi$. Different learning events can also be composed sequentially using $\alpha; \beta$. More interestingly, perhaps, it is possible to limit learning to only some agents, e.g. $L_A\, (\alpha)$, where A is the set of agents learning that α occurs. Agents outside the group A still learn that *something* occurred, just not the details, and update their epistemic states accordingly. This is still not quite in line with our requirements of an agent being *completely* unaware of some actions, such as the student pilot failing to realize the flaps did not deploy, or the confusion experienced by the pilots of AF-447.

When it comes to expanding the set of possible worlds, rograms now allow for agents to be unsure about which of two courses of action happened. For example, $L_A\, (?g \cup ?\neg g)$ means that the light is green, but that the agent cannot distinguish whether g or $\neg g$. In a sense, we may say that the agents in A cannot observe the resolution of the nondeterminism $?g \cup ?\neg g$, which looks suspiciously (hint!) like a dL program.

EAL has somewhat involved semantics beyond the scope of this section, although there are a few points of interest. The learning operator, $L_A\, (\alpha)$, creates a new Kripke model in which each possible world is a Kripke model itself. Using dL reachability notation, if $((M, s), (M', s')) \in \rho_\eta\, (L_A\, (\alpha))$, with $M = (W, \sim, V)$ and $M' = (W', \sim', V')$, then each $t' \in M'$ is, itself, a Kripke model (M'_t, s'_t). The model (M'_t, s'_t) effectively represents what happens when program α is "executed" at the possible world $t \in W$.

This is a very valuable insight into the semantic representation of the dynamics of knowledge. In fact, it may appear at first glance that this language is sufficient to, with some minor additions, describe knowledge revision when the world changes, as per our requirements.

Unfortunately, there are some relatively strong restrictions in EAL. Each sequentialized learning operator, as in $L_A(\alpha); L_B(\beta)$ must be written such that $B \subseteq A$. Given that establishing the safety of CPS requires repetition, in a sense, $L_B(\beta)$ comes after $L_A(\alpha)$ i.e. $L_A(\alpha); L_B(\beta)$, as much as $L_A(\alpha)$ comes after $L_B(\beta)$, i.e. $L_B(\alpha); L_A(\beta)$. The net effect, then, is that $B \subseteq A$ and $A \subseteq B$, and thus $A = B$. This attests to the difficulty of communal learning, and more generally the notion of common knowledge. We sidestep these issues by having individual learning only.

While there is no complete axiomatization of EAL, some sound axioms explain the fundamental behavior learning programs. These are very insightful in our endeavors.

$$
\begin{aligned}
[?\phi]\,\psi &\leftrightarrow (\phi \to \psi)\\
[\alpha;\beta]\,\phi &\leftrightarrow [\alpha]\,[\beta]\,\phi\\
[\alpha \cup \beta]\,\phi &\leftrightarrow [\alpha]\,\phi \wedge [\beta]\,\phi\\
[\alpha!\beta]\,\phi &\leftrightarrow [\alpha]\,\phi\\
[\alpha]\,p &\leftrightarrow (\mathbf{pre}(\alpha) \to p)\\
[\alpha]\,\neg\phi &\leftrightarrow (\mathbf{pre}(\alpha) \to \neg\,[\alpha]\,\phi)
\end{aligned}
$$

In the above, $\mathbf{pre}(\alpha)$ is a collection of formulas found in α smartly calculated as preconditions for the program successfully executing. One may think of the implications including $\mathbf{pre}(\alpha)$ as a generalization of the implications found in the axiomatization of PAL.

It is worth pointing out that with more complex programs for epistemic change, we now see case analyses over the programs themselves. Thus, this axiomatization comes closer to the sequent calculus of $\mathsf{d}\mathcal{L}$.

2.3.3 Action Models

Actions models, with the corresponding Action Model Logic AML [3], are an incredibly general approach to describing epistemic change. Action models are structures of the form **M,s**, very similar to Kripke structures where possible worlds represent an action, and have an associated precondition determining when the action can take place.

The fundamental insight of AML is that it is possible to define a product $M, s \otimes$ **M, s** of an action model **M, s** describing change, and a Kripke model M, s describing the current epistemic state. This results in a new Kripke model M', s' representing knowledge after the change described by **M, s**

Action models straddle an ambiguous line between syntax and semantics: while they are inspired by Kripke models, which are most often confined to realm of semantics, action models show up in syntax as programs of change, e.g. [**M, s**] ϕ.

Clearly, action models allow enormous flexibility in the description of epistemic change, especially when it comes to updating each agents' knowledge individually due to a very fine-grained interaction between the accessibility relations of the Kripke model and the action model.

Furthermore, AML comes with a proof system containing axioms very similar to those of EAL. It would thus look very promising as a starting point for integrating with $\mathsf{d}\mathcal{L}$ in an attempt to get a logic of changing knowledge in a changing world, despite the challenge of a somewhat unintuitive and difficult to read modeling language.

There is a fundamental issue with some of the crucial axioms. Consider the following one:

$$[\mathbf{M}, \mathbf{s}]\, K_a\left(\phi\right) \leftrightarrow \left(\mathbf{pre}(s) \rightarrow \bigwedge_{\mathbf{s} \sim_a \mathbf{t}} K_a\left([\mathbf{M}, \mathbf{t}]\, \phi\right)\right)$$

This axiom unpacks the action model into a formula with as many conjuncts as there are possible worlds. Unfortunately, while this wonderful insight works for the finite sets of possible worlds generated by finite propositional universes, it is incapable of tackling the uncountable sets of possible worlds required by beliefs or knowledge over $\mathbb{R}$-valued variables.

Chapter 3

Changing beliefs in a changing world

The first section of this Chapter analyzes several doxastic phenomena integral to aircraft operation safety, drawn from real world scenarios ranging from small elements of the AF-447 incident to airport landing systems and sensor readings. The insights from these analyses highlight the requirements for a logic capable of appropriately describing those phenomena. The collected requirements inform how we use the building blocks set forth in the previous chapter, particularly the logic $\mathsf{d}\mathcal{L}$, and augment them with new operators, also inspired by previous work, to create a new logic called the Dynamic Doxastic Differential Dynamic Logic $\mathsf{d}^4\mathcal{L}$.

Subsequent sections formally develop the syntax and semantics intuitively described in the first section, followed by an in-depth look at how semantics behave through state and belief change.

The final sections will introduce a sound calculus $\mathsf{d}^4\mathcal{L}$, taking the first steps towards verifying the safety of belief-aware cyber-physical systems.

3.1 Approach & framework

We begin by presenting a bird's eye view of our framework for modeling and verifying *belief-aware cyber-physical systems*. We begin by using Air France Flight 447's incident report to justify a significant simplification that will be helpful throughout the proposal.

Afterwards, we will focus on airplane landing approaches to highlight important aspects of aircraft operation that rely on pilot belief. These aspects are then translated into technical requirements for $\mathsf{d}^4\mathcal{L}$, which inform our proposed solution, built from the intuitions introduced in the previous chapter.

3.1.1 Single vs multi agent logics

A study of the Air France Flight 447 incident report [5] makes it clear that a significant factor in the plane remaining in a stall, and ultimately impacting the ocean, was that pilot input actions went unobserved by the copilot and vice-versa. This phenomenon merits careful consideration when designing cockpits and checklists to be followed by pilots during emergencies.

For example, when the pilot pulled the joystick back to climb, the pilot's beliefs were updated consider the plane had in fact started climbing. The same, however, was not true for the copilot, who did not observe the pilot's joystick action. The copilot's beliefs remained entirely unchanged, as if nothing had happened and the plane was still straight and level. In practice, this notion of *unobservable actions* by some but not all agents means that $\mathsf{d}^4\mathcal{L}$ needs to update an observing agent's beliefs without affecting, *in any way*, the beliefs of unobserving agents.

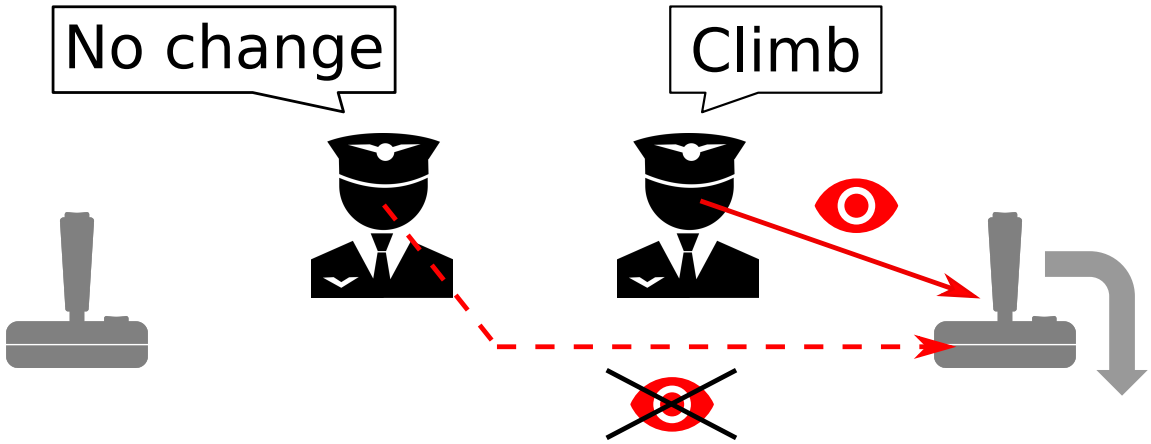


Figure 3.1: Climb input by pilot (right) unobserved by copilot (left)

This requirement of the logic - that the pilot's belief change has no impact on the copilot's beliefs - affects whether $\mathsf{d}^4\mathcal{L}$ can rely on Kripke models for its representation of belief: Kripke models are "flat" in the sense that all agents' beliefs rely on the same interconnected set of possible worlds with the same valuations. Only the agents' accessibility relations are distinct, but there's no hierarchy between them, and they connect the same worlds. Each agent's belief, then, is inextricably linked to other agents' beliefs in a Kripke model.

Updating Kripke models

If we're to use Kripke models $\langle W, \sim, V \rangle$ to represent belief, then, we must check whether it is possible to update $\langle W, \sim, V \rangle$ to change the pilot's beliefs exclusively, without change to the copilots'. To alter beliefs, one changes the (global to all agents) set of possible worlds W, the (also global) valuations of the worlds V, or the (local to each agent) agents' accessibility relations $\sim$. If the copilot believes some formula ϕ, denoted $B_{cp}(\phi)$, then changing the set of possible worlds W or altering their valuation V can obviously affect the truth value of $B_{cp}(\phi)$.

The same is not immediately clear with changes to the pilot's (not copilot's!) accessibility relation $\sim_p$. Specifically, the copilot's beliefs about the world *only* rely only on $\sim_{cp}$ and remain unfazed by a change to $\sim_p$. However, a change in $\sim_p$ may influence formulas of the form $B_{cp}(B_p(\phi))$, referring to the copilot's beliefs about the pilot's beliefs.

Thus, the prospect of effecting change in Kripke models, by changing either W, V or any of the agents' accessibility relations $\sim$ *individually*, is bleak.

Unfortunately, approaches that change the entire Kripke model at once also run into problems. Notions such as knowledge or belief, and the modalities that are derived from them, rely on properties of the accessibility relations, such as reflexivity, transitivity and symmetry. It is those properties that result in some of the axioms of knowledge like $K_p(\phi) \rightarrow K_p(K_p(\phi))$. It is crucial that the properties of the accessibility relations be upheld by any change to the Kripke model - if we start with equivalence relations, after any change, we should end up with equivalence relations - since failure to do so comes at the cost of all the axioms that define the very nature of knowledge and belief.

However, the flatness of Kripke models means that any changes which update one agent's belief while keeping another's intact ultimately result in the loss of at least of the properties of the accessibility, as exemplified in Appendix A. The modalities of such a logic would lose all connection to the notions of belief as soon as the underlying Kripke model is updated, which beats the entire purpose.

Solution

This appears to be a fundamental problem in the combination of 1) Kripke models to represent belief, 2) unobservable doxastic change, and 3) agents having beliefs about other agents' beliefs. Indeed, finding a semantic structure that solves this problem is a whole thesis topic in itself, and therefore out of the purview of the present one.

Instead, we relinquish property 3), and thus declare that agents can no longer hold beliefs about one another's beliefs. In this approach formulas such as $B_{cp}(B_p(\phi))$ are equivalent to $B_p(\phi)$, and thus effectively intuitively meaningless.

Simplification

In Kripke models, agents may have multiple equivalence classes of possible worlds. One such class will be distinguished in that it contains the distinguished world.

Equivalence classes beyond the distinguished one are only relevant because they are reachable using multiple accessibility relations, e.g. with $B_{cp}(B_p(B_{cp}(\phi)))$. This is illustrated in Figure 3.2, with the copilot in blue and the pilot in red: if the left circle is the copilot's distinguished equivalence class, then the formula travels through the pilot's red equivalence class, on to the copilot's second, non-distinguished, equivalence class.

Without the ability to talk about beliefs about beliefs, the formula $B_{cp}(B_p(B_{cp}(\phi)))$ is not possible, and we may thus significantly simplify our underlying Kripke models by having a single equivalence class for each agent.

Thus, in our modified Kripke models, each agent a has their own set of possible worlds W_a with their own valuations V_a, but no accessibility relation since we assume W_a to be an equivalence class. Furthermore, while every agent a has worlds W_a and valuation V_a, whenever dealing with single-agent scenarios we will omit the agent for notational clarity.

With one major decision out of the way, we may now direct our attention to real use cases, which will inform the yet more such decisions.

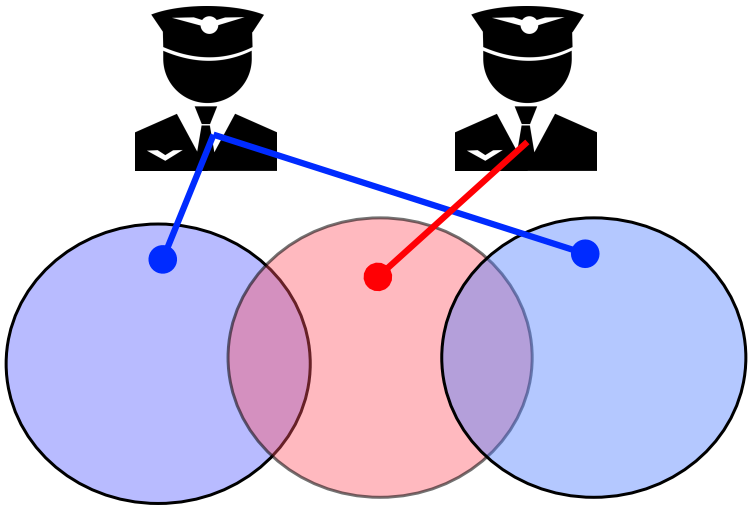


Figure 3.2: Beliefs about beliefs, $B_{cp}\left(B_p\left(B_{cp}\left(\phi\right)\right)\right)$

3.1.2 Landing systems: Precision Approach Path Indicators

To land, aircraft must first align themselves with the runway. Once aligned, aircraft on a landing approach attempt to remain on what is called a glideslope: a line in the vertical plane that indicates the optimal descent path to the runway.

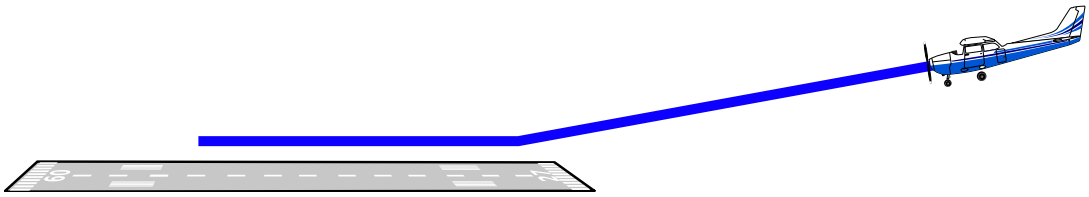


Figure 3.3: Landing approach - glideslope

Many runways are equipped with electronic and visual aids to help pilots remain on the glideslope, which can be otherwise tricky or even impossible depending on training and weather conditions. One such visual system is the Precision Approach Path Indicator, or PAPI. It is composed of long-distance directional lights, usually four, sited at the beginning of the runway, which will look either green[1] or red depending on where the airplane is in relation to the glideslope. There is a helpful mnemonic for pilots:

White on white - check your height (too high)
Red on white - you're alright (on glideslope)
Red on red - you're dead (too low)

With four lights, the PAPI system provides a fair amount of information to the pilot, as can be seen in the following figure.

[1] in real life, the lights are white, but it is easier to represent them as green in this document

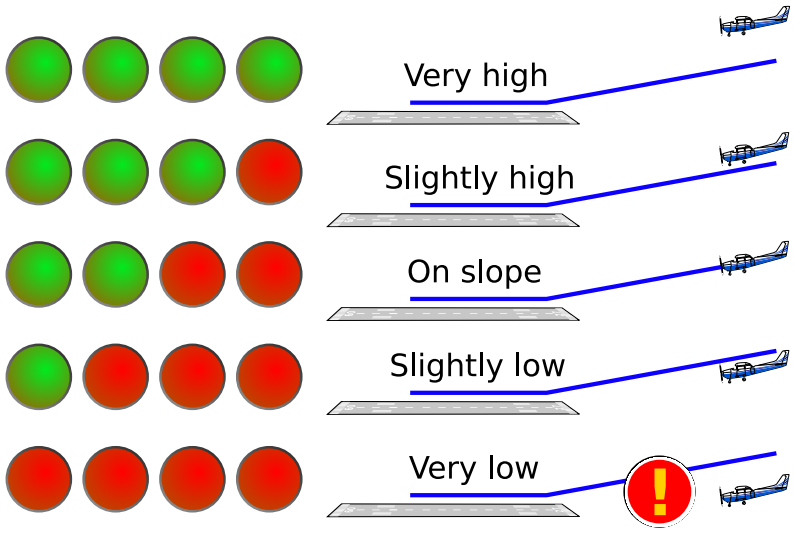


Figure 3.4: Precision Approach Path Indicator (PAPI)

In order to model our airplane and the PAPI lights, we can begin with the logic dℒ. We represent an aircraft's vertical position as a variable alt, and its vertical velocity as a variable vy. The dℒ hybrid program $alt' = vy$ would use the differential equation to represent the aircraft's altitude changing according to its vertical velocity. For the sake of simplicity, we assume a constant horizontal speed vx affecting distance to the runway d.

$$alt' = vy, d' = -vx$$

We also need a model for the PAPI lights. We use variables $\texttt{l1}, \texttt{l2}, \texttt{l3}$ and $\texttt{l4}$ for each of the lights from left to right. The domain of these variables is $\mathbb{R}$, as with all dℒ variables, but to simplify notation, we will use colored letters to represent any two values, 0 and 1 for green and red respectively. Thus, here is the formula for being on glideslope:

$$\texttt{l1} = G \wedge \texttt{l2} = G \wedge \texttt{l3} = R \wedge \texttt{l4} = R$$

And for slightly above the glideslope:

$$\texttt{l1} = G \wedge \texttt{l2} = G \wedge \texttt{l3} = G \wedge \texttt{l4} = R$$

The state of the lights must also change to reflect the position of the plane, and for this we will need to know where the plane is in relation to the glideslope. To avoid unnecessary notation, let us assume that we have dℒ formulas *very-high*(alt, d), *high*(alt, d), *low*(alt, d) and *very-low*(alt, d) when when the aircraft is (far) above, or (far) below the glideslope. When the airplane is on track, we may use $\neg$*low*$(alt, d) \wedge \neg$*high*(alt, d). Recall that variables occurring in the formula show up in parenthesis as a helpful reminder of which variables are relevant.

To update the state of the lights, we can, once again, use a hybrid program. It should check relevant conditions for each light. For example, if the airplane is very low, then the first light

must be red. If not, then it must be green.

$$((?\textit{very-low}; \texttt{l1} := R) \cup (?\neg\textit{very-low}; \texttt{l1} := G));$$
$$((?\textit{low}; \texttt{l2} := R) \cup (?\neg\textit{low}; \texttt{l2} := G));$$
$$((?\textit{high}; \texttt{l3} := G) \cup (?\neg\textit{high}; \texttt{l3} := R));$$
$$((?\textit{very-high}; \texttt{l4} := G) \cup (?\neg\textit{very-high}; \texttt{l4} := R))$$

The program for each light, e.g. $(?\textit{very-low}; \texttt{l1} := R) \cup (?\neg\textit{very-low}; \texttt{l1} := G)$, is a nondeterministic choice. However, because the two tests are mutually exclusive, one of the choices must fail the test, and the other must pass it. Therefore, while the choice is nominally nondeterministic, the program is actually deterministic and performs an "if-else" statement, setting the lights as desired.

For the plane to remain safe, the pilot must make appropriate decisions to increase, decrease or maintain descent at the right time. To model in the pilot as a controller, we assign to a variable yi indicating vertical input, e.g. $yi := 1$ for climb input, $yi := -1$ for descent input, and $yi := 0$ for maintaining descent rate. To incorporate pilot input into the model, we make yi influence the vertical speed vy:

$$alt' = vy, vy' = vi, d' = -vx$$

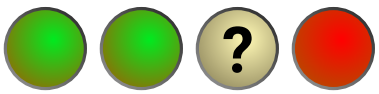
So now the input variable is being taken into account, but we are missing the most crucial aspect of CPS modeling and verification: the control procedure of the pilot, which determines what yi should be set to.

In this case, the pilot is looking to the PAPI lights for guidance: that is the basis for their choices. A simplified, initial decision process may be that if the second light is red (the airplane is low or very low), then the pilot climbs. Analogously, if the third light is green (the airplane is high or very high), the pilot descends. Otherwise, the airplane is on slope and the pilot maintains a neutral input.

$$(?\texttt{l2} = R; yi := 1) \cup$$
$$(?\texttt{l3} = G; yi := -1) \cup$$
$$(?\texttt{l2} = G \wedge \texttt{l3} = R; yi := 0)$$

Since, much like before, the test conditions are mutually exclusive (the airplane is either high, low or on-slope), the program behaves like a deterministic "if, else-if, else" statement, despite the use of nondeterministic choice.

However, and this is the main motivator for this thesis, *the pilot does not have direct, perfect access to the state of the world*. Due to deteriorating weather conditions, sun glare, partial PAPI light malfunction, or any other host of reasons, the pilot may be unable to fully ascertain the state of, for example, the third PAPI light.



This means that while we may be able to prove the theoretical safety of the controller presented above, these guarantees would not translate to the real world, in which there is no such thing as perfect information.

From a syntactical perspective then, using a *state* variable in control decisions, i.e. `l3` being direct access to world-state, is entirely unrealistic: *any control decision which makes use of the exact state of the world can, and ultimately will, be ill-informed in the real world, resulting in possible safety violations.*

Instead, the pilot's control decisions must be explicitly based on the information that they do have access to: their beliefs.

3.1.3 Belief-triggered controllers

We must now to find an adequate representation of belief that can be integrated as seamlessly as possible with the logic of physical change $\mathsf{d}\mathcal{L}$. This will enable us to write control decision conditions based on belief rather than physical state.

In order to do this, we must develop for $\mathsf{d}^4\mathcal{L}$ an appropriate semantic representation for the distinction between the physical state of the world, and the doxastic state of the agent: both are needed.

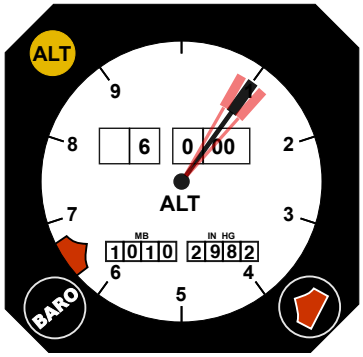
The syntax of $\mathsf{d}^4\mathcal{L}$ must then allow a CPS engineer to easily refer to physical and doxastic state in order to model important phenomena such as sensor readings, which reflect imperfect beliefs about the physical state of the world.

We begin by addressing precisely the phenomenon of sensor readings, in the context of syntax for belief.

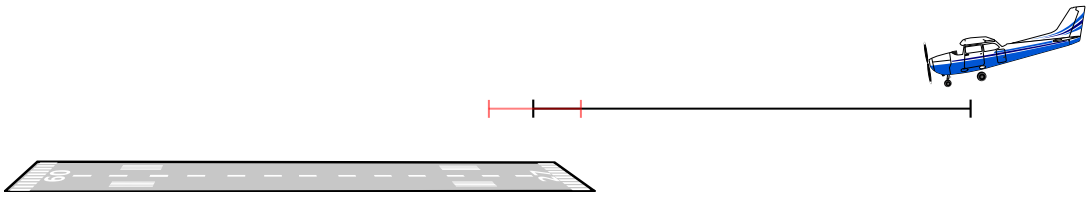
Doxastic variables

Due to the complications with the third PAPI light, the pilot is now unsure of the plane's position in the glideslope. Instead, they must rely on visual estimation of distance to the airfield, and on the altimeter for a reading of altitude.

The black needle of the altimeter below shows that the sensor reading for altitude is at 100 meters. In reality, this sensor is imperfect, and so the pilot believes that the real altitude of the aircraft is somewhere between $[100 - \epsilon, 100 + \epsilon]$, as indicated by the two imaginary red needles in the altimeter. The variable ϵ is an usually known or conservatively estimated error bound for the altimeter.



Similarly, there is significant evidence to support the notion that humans are not particularly good at estimating distances [18]. A pilot model that is safe in practice as well as theory must take into account the distinction between perceived and actual distance to the runway. The picture below shows this situation.



If we think of the human eye as a sensor, then altimeter readings and distance estimation are two instances of the same phenomenon. In simple terms, the phenomenon establishes a relation between some perceived value and its real, physical value.

Let us then define a *doxastic variable* alt_p representing the perceived value of alt for agent p. Then the following formulas represent the relation that the perceived and real value of a variable differ by at most ϵ, the error bounds of the sensor being used.

$$(alt_p - alt)^2 < \epsilon^2$$

$$(d_p - d)^2 < \epsilon^2$$

Crucially, both alt and alt_p must appear in the same context as in these formulas This makes it impossible to use context to determine whether interpretation should be doxastic or physical, e.g. in $alt > 0$, alt would be a state variable, but in $B_p\,(alt > 0)$, alt would be a doxastic variable for agent p, with the agent being inferred from the doxastic modality.

Term interpretation: single- or set-valued

Having established the need to have separate syntax for both state as well as doxastic variables, we must now consider how they may be interpreted.

Indeed, it is important that the distinction in our models is clear enough that doxastic and state variables can change independently of one another. This is a crucial component of being able to represent unobservable change, such as when the AF-447 pilot added climb input (state change) without being noticed by the copilot (doxastic change).

To achieve this, we add to our simplified Kripke models a valuation r that represents the real state of the world, which corresponds to a $\mathsf{d}\mathcal{L}$ state. Thus, a model for $\mathsf{d}^4\mathcal{L}$ would be $\langle r, W, V, s\rangle$, and state variable interpretation would be done in the real world valuation:

$$\text{val}_\eta\,(\langle r, W, V, s\rangle, alt) = r(alt)$$

However, we are particularly interested in doxastic variables, and whether they should have set-valued or single-valued interpretations. Using the semantics of Kripke modal logics, and $\mathsf{d}\mathcal{L}$ notation, these are the two options:

- The single-valued interpretation of a variable is its value at the only possible choice of world: the distinguished world.

$$\text{val}_\eta\,(\langle r, W, V, s\rangle, alt_p) = V(s)(alt_p)$$

- The set-valued interpretation of a variable would follow the common intuition of Kripke modal logics: the values at all possible worlds considered by the agent p:

$$\text{val}_\eta\left(\langle r, W, V, s\rangle, alt_p\right) = \{v \in \mathbb{R} : \text{ there is } t \in W \text{ s.t. } V(t)(alt_p)\}$$

Using set-valued interpretations, we would be able to further simplify models by eliminating the distinguished world, as follows:

$$\text{val}_\eta\left(\langle r, W, V\rangle, alt_p\right) = \{v \in \mathbb{R} : \text{ there is } t \in W \text{ s.t. } V(t)(alt_p)\}$$

The implications of the choice between set- or single-valued interpretations are far reaching, but the most relevant consequence of choosing set-valued term interpretation is that it affects the domain and meaning of formula interpretation. When terms can take multiple values, truth becomes a more nuanced concept.

If we wish the truth domain of formulas to remain two-valued, i.e. $\{\mathtt{T}, \mathtt{F}\}$, then perhaps the most intuitive choice would be for $\mathtt{T}$ to mean the formula is true for all possible term values, and $\mathtt{F}$ to mean that at least one term interpretation falsifies the formula. We could also consider three-valued truth domains, such as true for all term values, for some but not all, and for no term values.

Whatever the case, such a shift in the truth domain of formulas would invalidate the entire $\mathsf{d}\mathcal{L}$ calculus, on which $\mathsf{d}^4\mathcal{L}$ relies for reasoning about the physicality of CPS. Reproving the entire body of work around $\mathsf{d}\mathcal{L}$ for set-valued terms and multi-valued truth domains would be a gargantuan task indeed, and thankless given that belief-free CPS reasoning does not require multi-valued terms in the least.

We thus come to our second major decision: all of our terms will remain single-valued, so as to maintain integration with $\mathsf{d}\mathcal{L}$. It thus becomes the prerogative of doxastic modalities to determine the meaning of formulas whose interpretation depends on multiple possible worlds.

Doxastic modalities

With doxastic terms being single-valued, and interpreted in the distinguished world, it falls to doxastic modalities to "aggregate" the values at all possible worlds and make sense of them.

The notions of truth for formulas with set-valued terms can captured by doxastic modalities such as belief and possibility. Using $\mathsf{d}\mathcal{L}$ notation and Kripke modal semantics:

$$\begin{gathered}\text{val}_\eta\left(\langle r, W, V, s\rangle, B_p\left((alt_p - alt)^2 < \epsilon^2\right)\right) = \mathtt{T} \text{ iff} \\ \text{for every } t \in W, \text{val}_\eta\left(\langle r, W, V, t\rangle, (alt_p - alt)^2 < \epsilon^2\right) = \mathtt{T}\end{gathered}$$

$$\begin{gathered}\text{val}_\eta\left(\langle r, W, V, s\rangle, P_p\left((alt_p - alt)^2 < \epsilon^2\right)\right) = \mathtt{T} \text{ iff} \\ \text{for some } t \in W, \text{val}_\eta\left(\langle r, W, V, t\rangle, (alt_p - alt)^2 < \epsilon^2\right) = \mathtt{T}\end{gathered}$$

We can see that these inductive semantics for doxastic modalities rely on the distinguished world to interpret doxastic variables. Interestingly, while it is still technically necessary for such inductive semantics, the distinguished world no longer represents the real state of world like in epistemic logics - that job now falls to r.

This raises some questions as to how the initial choice of distinguished world affects the truth of formulas. We will see when developing the logic and calculus that when proving a formula,

the choice of distinguished world is handled in the same way that the choice of initial state is handled in dL, and thus not a matter of concern. Since validity of a formula ranges over all models, then it also ranges of all choices of distinguished world.

It is perhaps curious to note that even though we have made a decision to forgo multi-valued truth domains, they can be at least partially captured by the addition more doxastic modalities. A modality S could be added for suspicion, i.e. "for some but not all" possible term interpretations. It turns out that suspicion can be reduced to possibility, which is the dual of belief, and is thus a part of the logic already:

$$S_p((alt_p - alt)^2 < \epsilon^2) \equiv P_p\left((alt_p - alt)^2 < \epsilon^2\right) \wedge P_p\left(\neg(alt_p - alt)^2 < \epsilon^2\right)$$

Thus, d^{4}L will use an inductive semantics that relies on the distinguished world to make sense of doxastic variables. Modalities are then tasked with the aggregation of many truth values into a single one. This follows the principles of modal logics, where modalities define the different modes of being true, but not the fabric of truth itself.

Belief and state variables

It is now possible to write unintuitive formulas that speak of belief about *state* variables, such as

$$B_p\left(\mathtt{l3} = R\right)$$

With what we have defined so far, we can already see that there is a well defined interpretation for this formula:
$\text{val}_\eta\left(\langle r, W, V, s\rangle, B_p\left(\mathtt{l3} = R\right)\right)$ iff
for all $t \in W$, $\text{val}_\eta\left(\langle r, W, V, t\rangle, \mathtt{l3} = R\right)$ iff
for all $t \in W$, $\text{val}_\eta\left(\langle r, W, V, t\rangle, \mathtt{l3}\right) = R$ iff
for all $t \in W$, $r(\mathtt{l3}) = R$

Thus, it becomes apparent that the universal quantification is redundant, and thus that $B_p\left(\mathtt{l3} = R\right)$ is actually equivalent to $\mathtt{l3} = R$. While this may seem surprising at first, consider that beliefs do not have particular meaning for state variables. Indeed, a state variable inside a belief modality is a constant.

Belief only has meaning for *doxastic* variables, and hence, what we have uncovered is merely a semantic idiosyncrasy of a modeling mistake. It happens to result in a formula whose modalities are unnecessary, much like $[\alpha]\, x^2 \geq 0$ in dL, or $[\phi]\left(K_p\left(\phi\right) \vee \neg K_p\left(\phi\right)\right)$ in Public Announcement Logic. In truth, what our CPS engineer meant to write was $B_p\left(\mathtt{l3}_p = R\right)$.

Belief-triggered controller for PAPI

Recall that the problem with the original controller was that the pilot's decisions relied on the state of the world instead of the pilot's beliefs thereof.

$$\begin{aligned}
&(?\mathtt{l2} = R; yi := 1) \cup \\
&(?\mathtt{l3} = G; yi := -1) \cup \\
&(?\mathtt{l2} = G \wedge \mathtt{l3} = R; yi := 0)
\end{aligned}$$

By adding doxastic modalities, we may finally adapt the above controller into a *belief-triggered controller*:

$$(?B_p(\texttt{l2} = R);yi := 1) \cup$$
$$(?B_p(\texttt{l3} = G);yi := -1) \cup$$
$$(?B_p(\texttt{l2} = G \wedge \texttt{l3} = R);yi := 0)$$

The change appears minor on paper, but it represents a major paradigm shift: this new CPS modeling language explicitly indicates that control decisions are based on beliefs, not on world state, addressing one of the crucial problems in bridging the gap between theoretical safety and real safety of CPS.

Our pilot can now make decisions based on their beliefs about the PAPI lights. The next step is turning our attention towards how those beliefs come to be: how they are gained, lost and changed.

3.1.4 Updating beliefs

In order to successfully incorporate belief-change into a logic based on dℒ, we must first understand how CPS are modeled and verified therein. It is only then that we can successfully develop belief-change operators that integrate properly with current modeling and verification techniques.

Recall that CPS safety in dℒ often takes the following form.

$$\mathit{pre} \rightarrow [(\texttt{ctrl; plant})^*]\,\mathit{safe}$$

This pattern does not explicitly mention beliefs, which are, most often, also not implicitly modeled.

In reality, beliefs are often obtained through *observation*, either of the world or of sensors. Beliefs about the PAPI light, obtained by looking out the cockpit window, inform the pilot's decision to alter or maintain the descent rate. It is clear, then, that observation should come before decision. We thus offer an updated belief-aware CPS pattern, which closely resembles the previous one, with the addition of observation for belief acquisition.

$$\mathit{pre} \rightarrow [(\texttt{obs; ctrl; plant})^*]\,\mathit{safe}$$

First, the pilot observes, then makes a decision to, e.g. climb, and finally, the CPS is allowed to evolve, and the plane continues its descent for some time. Then, the pilot makes another observation, followed by yet another decision, and the airplane flies a little more. Whenever the pilot becomes uncertain because of imperfect observations, such as when the third PAPI light fails to resolve to a clear red or green, then more conservative choices would end up being triggered, as fewer things are believed with certainty.

Whichever method we use in `obs` to update an agents' belief state must integrate well with dℒ hybrid programs. We may already discard the declarative approach of AGM because of the simplicity (sometimes a desirable trait, sometimes not) of its revision operator. Let us instead review what inspiration we may draw from each dynamic epistemic logic.

Public Announcements

Recall public announcements from PAL, $[\phi]\,\psi$, which say that after some trusted source announces that ϕ is true, thus updating every agent's knowledge, then the formula ψ is true. It does this by eliminating every possible world in which ϕ does not hold.

In order to add this behavior to our logic, we could add a *learning operator* to hybrid programs, to which we assign the same semantics. Thus, if we wanted our pilot to learn that the first PAPI light was red, we could write $L_p\,(\texttt{l1}_p = R)$.

But of course, public announcements merely contract the set of possible worlds. When the third PAPI light becomes unclear, what is needed is an *expansion* of the set of possible worlds so that both red and green are possibilities. So clearly, public announcements do not have the expressiveness we require, and we need something more.

Action models

At the other end of the expressiveness spectrum we have Action Model Logic, in which Kripke-like structures describe, in minute detail, how knowledge should be updated. These structures, most often considered to be semantic, are written out as syntax inside a dynamic modality defined for updating knowledge. The effect of the update is computed by means of an algebraic product operator between the two Kripke-like structures: the one representing the epistemic state, and the other representing epistemic change.

With this level of expressiveness, it is conceivable we could encode most desired belief-change patterns in action models.

Unfortunately, recall that the calculus relies rather fundamentally on the finiteness of the underlying Kripke models: the central axiom of action model belief creates a conjunction that ranges over each accessible world.

But when the pilot looks at the altimeter, showing the airplane may be anywhere between $[100 - \epsilon, 100 + \epsilon]$, then already our Kripke model contains an uncountably infinite number of worlds, and uncountable infinite formulas are not possible.

We must continue our search.

Epistemic actions

The logic of epistemic actions offers a level of expressiveness somewhere between that of Public Announcement Logic and that of Action Model Logic. Indeed, with extensions to include public and atomic assignment [21], it comes very, very close to the semantics we desire.

However, it is worth stating disadvantages first, so that we may focus on the advantages later. The absence of a complete axiomatization speaks to the challenges of common learning and knowledge. The admission that it was in fact beyond to reach of those trying to prove one, is another strong indicator that a direct adoption of such an approach may be ultimately unfruitful.

Perhaps more relevant to this discussion are the restrictions imposed by the semantics themselves. The groups of agents A and B of subsequent or nested learning operators, such as in $L_A\,(\alpha)\,;L_B\,(\beta)$, must be such that $B \subseteq A$. With nondeterministic repetition in patterns such as $(\texttt{obs; ctrl; plant})^*$, it would mean that A and B both occur after and before each other, which is impossible unless $A = B$.

This points to an intrinsic difficulty with *sets* of agents whose knowledge, or in our case, belief, must be updated simultaneously. But, given our decision to forgo precisely such phenomena of interacting beliefs, we will perhaps not face the same challenges and be able to proceed further.

Indeed, we find that there is much to be inspired by in this logic. To begin with, the language of epistemic actions is uncannily similar to that of hybrid programs. They both feature sequential composition, and there is a strong parallel between the choice operators of both programs, as well as between $\mathsf{d}\mathcal{L}$'s test and learning a formula.

Perhaps, then, we can use the existing semantics of the logics of epistemic actions as inspiration rather than starting point, and because of their similarities, unify the language of doxastic and physical change.

3.1.5 The learning operator

The first step in integrating the language of doxastic change with hybrid programs is to add a learning operator. Thus, $L_p(\alpha)$, meaning that the pilot p learns, or comes to believe, that program α executed. This does not mean that α actually happened: the learning operator affects doxastic state *only*.

Hybrid programs as learning actions

If the languages that describe physical and doxastic change are identical, then this immediately opens up interesting patterns such as $\alpha; L_p(\alpha)$ for some event α that the pilot happens to observe accurately. For example, if, while on approach, the pilot is a little too enthusiastic with their descent, the second PAPI light will visibly turn red, and we may apply the pattern.

$$\texttt{l2} := R; L_p(\texttt{l2} := R)$$

Equipped with this simplest of examples, we may begin to tackle the challenge of the semantics. We have help, however: the semantics of all dynamic epistemic logics described thus far have one recurring property. All of them describe change in some way, but then apply it *to each possible world*. PAL looks at the truth value of the announced formula at each world, and discards it or not. Epistemic Action Logic makes each possible world a distinguished world in an otherwise unchanged Kripke model, and executes actions from that Kripke model. And finally, Action Model Logic's Kripke model algebraic product operator combines each possible world with a possible action.

We follow in those footsteps. The assignment in $L_p(\texttt{l2} := R)$ should behave such that each possible world's `l2` becomes red. The simplest way to do this is to use the system of EAL whereby the assignment is executed in the the same Kripke model, but where the distinguished world is changed to each possible world. The final Kripke model would be an aggregation of all the results: each possible execution from each "original" possible world results in a "new" possible world.

Thus, so long as the assignment affects the distinguished world, then it will affect all possible worlds.

Doxastic assignment

Unfortunately, the change described by $\mathtt{12} := R$ is physical in nature. Its effect is exclusive to the physical state r, and is unaltered by a change in the distinguished world. What we want is an explicitly doxastic assignment which affects the distinguished world: $\mathtt{12}_p := R$.

Using the transition notation of dL, the behavior of this assignment can be described as follows: $(\langle r, W, V, s\rangle, \langle r', W', V', s'\rangle) \in \rho_\eta(\mathtt{12}_p := R)$ iff $r' = r$, $W' = W$, $s' = s$, and $V' = V$ except that $V'(s)(\mathtt{12}_p) = R$.

This is, admittedly, a departure from our claim that we would use *exactly* the same programs to describe physical and doxastic change, but one that allows us to have a fully inductive semantics for the learning operator. The alternative is to explicitly change the semantics of regular assignment when it appears inside the learning operator, which is bothersome, inelegant and unnecessary.

Thus, for assignment, the learning operator would behave as in Figure 3.5: applying assignment at each possible world, by creating a model in which the only difference is that it is distinguished. The blue circle represents the physical state r, and each other circle the possible worlds. The distinguished world is double-circled, and changes in valuations are highlighted by a change in color. Equal valuations retain equal colors.

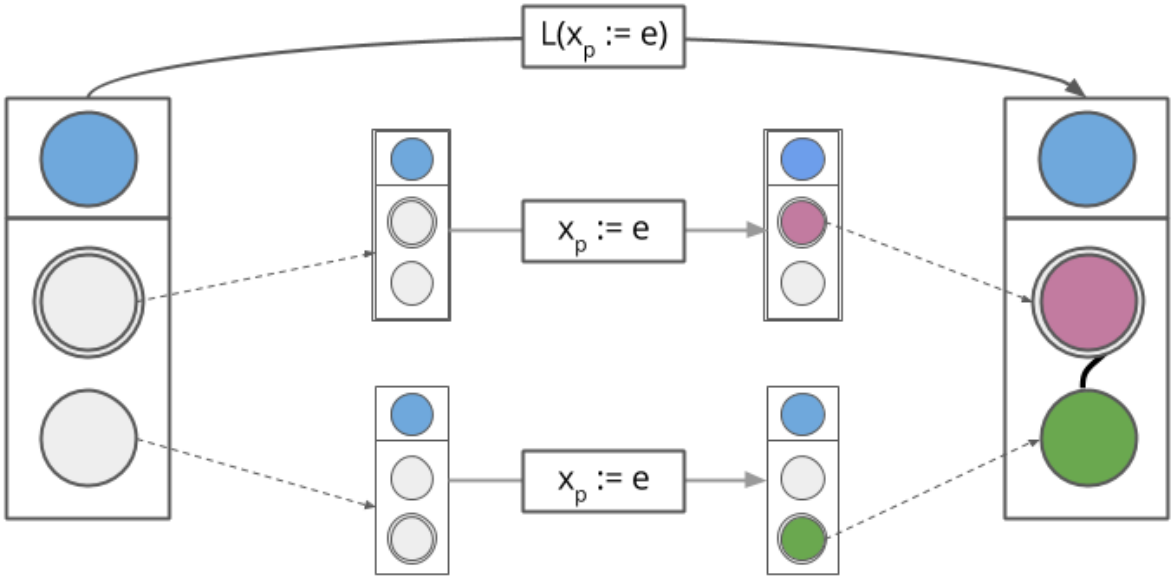


Figure 3.5: The transition of $L_p(x_p := \theta)$

Notice how each possible world results in a "small Kripke model" identical to the initial one, but in which the chosen possible world becomes distinguished. The program $\mathtt{12}_p := R$ then alters the valuation of the distinguished world in each "small model". Finally, all small models are aggregated as the possible worlds of the final model: the result of $L_p(\mathtt{12}_p := R)$. The valuations of the worlds of the final model are those of the distinguished world of their respective "small model".

Effectively, when a pilot learns that the second light became red, what happens behind the scenes is that the learning operator turns the light red at each possible world.

None of the other operators we consider need their own specific doxastic variants, so that we may directly use the same semantics to elegantly incorporate into learning the nondeterminism from nondeterministic choice, the interruption of traces from test, and sequential composition.

As these will be seen in greater detail later, we will merely sketch their intuition here.

Learned test

Since test simply makes it so that certain models don't transition, in these semantics certain "small models" won't transition, and thus won't result in any new possible worlds.

Effectively, then, the semantics of $[L_p(?\phi)]\psi$ behave exactly like those of the public announcement $[\phi]\psi$ of PAL: worlds who do not pass the test for the formula simply don't show up in the resulting model.

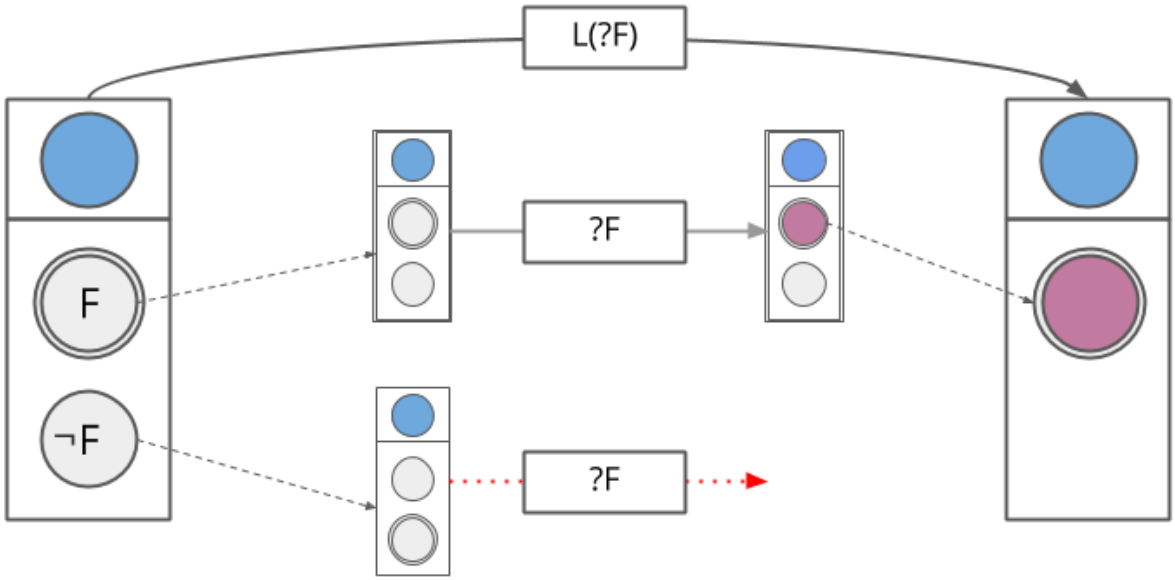


Figure 3.6: The transition of $L_p(?\phi)$

Sequential composition

Sequential composition $L_p(\alpha;\beta)$, despite somewhat difficult technically, behaves intuitively: α generates new "small models", from which yet more "small models" are generated using β. Those are the ones made into the possible worlds of the final model.

Nondeterministic choice and unobservability

As we said, nondeterminism is elegantly incorporated into these semantics. It simply means that "small models" may "multiply" through nondeterminism, resulting in a larger number of possible worlds than one started with.

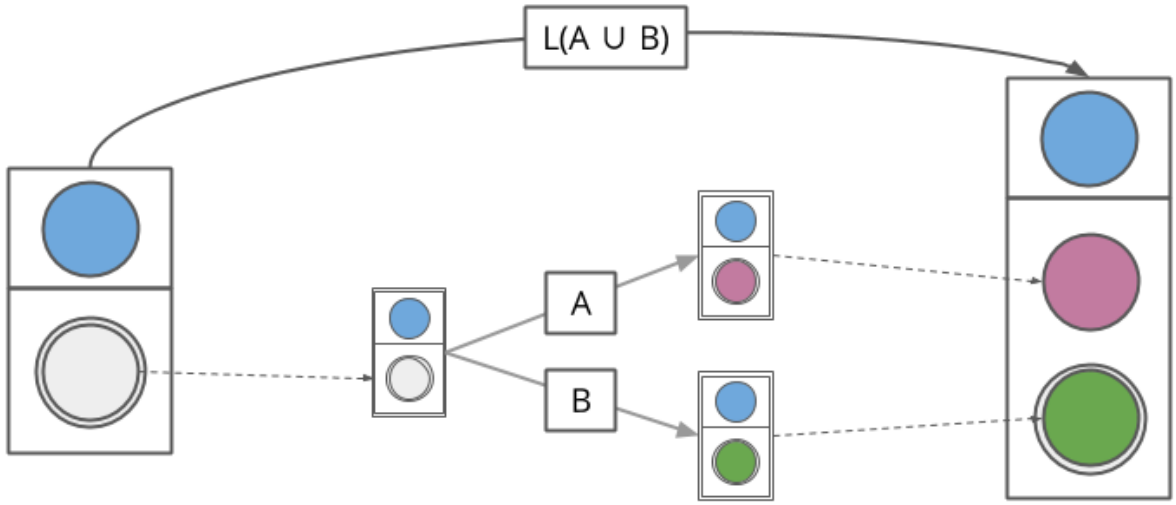


Figure 3.7: The transition of $L_p(\alpha \cup \beta)$

This is similar to what happens in AML, with each possible world of the original model being multiplied by the multiple worlds of the action model itself. This results in potentially larger numbers of worlds in the final model.

It also highlights an important principle: nondeterminism as doxastic indistinguishability. When nondeterminism is learned, such as in EAL, agents cannot distinguish between the different outcomes. This is why all outcomes of nondeterminism end up in an agent's equivalence class.

This is also the reason why $L_p(\mathtt{l3}_p := R \cup \mathtt{l3}_p := G)$ is so conceptually different from $L_p(\mathtt{l3}_p := R) \cup L_p(\mathtt{l3}_p := G)$. In the latter case, the pilot either comes to believe that the third light became red, or they come to believe that it became green. But only one of those outcomes is possible, at the end of such a program, the pilot is either convinced that the light is red, or they are convinced the light is green.

With $L_p(\mathtt{l3}_p := R \cup \mathtt{l3}_p := G)$, however, the pilot cannot distinguish whether the light became red or green. Thus, they do not come to believe it to be green or red - both are possible. With less certainty should come more conservative choices for the pilot. For instance, if they don't believe that the airplane is on slope or above-slope, then climbing is the safest choice.

These are the crucial issues with which a belief-aware cyber-physical system must contend with if they are ever to proven safe in the *real* as well as in the theoretical world.

3.1.6 Learning for belief-triggered controllers

To unify the language of change, programs which specify physical change can also define doxastic change. Whether a program is encapsulated within a learning operator determines which part of the state is being affected, α affecting physical state and $L_p(\alpha)$ affecting doxastic state.

Recall the student pilot performing touch-and-go maneuvers from Section 1.1.1. With the statement $\mathit{flaps} = \mathtt{F}$ indicating retracted flaps and $\mathit{flaps} = \mathtt{T}$ for deployed flaps, the pilot learning

beliefs consistent with reality could be modeled as follows:

$$\mathit{flaps} \coloneqq \mathtt{T}; L_p\left(\mathit{flaps}_p \coloneqq \mathtt{T}\right) \tag{3.1}$$

The first assignment describes the flaps being deployed in the real world, possibly affecting lift, drag, and how the airplane flies. The second assignment, in the learning operator, represents the pilot coming to believe that the flaps are deployed. The pilot is now modeled as a belief-triggered controller, as their doxastic state would then influence their decisions later on, e.g. increasing throttle due to perceived increased drag.

After program (3.1) executes, the state of the world, physical and doxastic, is accurately captured by the following formulas:

$$\mathit{flaps} = \mathtt{T} \ \wedge \ B_p\left(\mathit{flaps}_p = \mathtt{T}\right) \tag{3.2}$$

This is another instance of the $\alpha; L_p\left(\alpha_p\right)$ pattern, where α_p substitutes assignment for doxastic assignment, which is very common for *observable actions*. Whenever an agent performs an action that they can confirm the results of, the agent's doxastic state is updated (almost) simultaneously with the physical state of the world.

This will not, however, always be the case. There are actions and events whose results cannot be confirmed, such as when the weather turns and the color of the third PAPI light is no longer obvious.

The pilot would then correctly know the state of lights 1, 2, and 4, but be uncertain about the state of light 3. The following program models such uncertainty.

$$\mathtt{l1} \coloneqq G; \mathtt{l2} \coloneqq G; \mathtt{l3} \coloneqq R; \mathtt{l4} \coloneqq R;$$

$$L_p\left(\mathtt{l1}_p \coloneqq G; \mathtt{l2}_p \coloneqq G; \overbrace{\left(\mathtt{l3}_p \coloneqq G \cup \mathtt{l3}_p \coloneqq R\right)}^{\text{uncertainty}}; \mathtt{l4}_p \coloneqq R\right)$$

Notice the similarities to the observable action/event pattern $\alpha; L_p\left(\alpha\right)$ from before, excepting the third light's uncertainty. A simple yet intuitively equivalent reshuffling of the program makes this more obvious:

$$\mathtt{l1} \coloneqq G; L_p\left(\mathtt{l1}_p \coloneqq G\right); \mathtt{l2} \coloneqq G; L_p\left(\mathtt{l2}_p \coloneqq G\right);$$

$$\mathtt{l3} \coloneqq R; L_p\left(\overbrace{\left(\mathtt{l3}_p \coloneqq G \cup \mathtt{l3}_p \coloneqq R\right)}^{\text{uncertainty}}\right); \mathtt{l4} \coloneqq R; L_p\left(\mathtt{l4}_p \coloneqq R\right)$$

So the third light, while actually red, cannot be accurately read by the pilot due to the lighting circumstances. The relevant part of the program, $L_p\left(\mathtt{l3}_p \coloneqq G \cup \mathtt{l3}_p \coloneqq R\right)$, updates the pilot's beliefs to a state where it is possible that the third light is red, and also possible that it is green. The pilot would not be able to know which, though there is certainty about the other lights. Thus, the following formula represents the physical and doxastic state of the world for lights 1, 2 and 4.

$$\mathtt{l1} = G \ \wedge \ \mathtt{l2} = G \ \wedge \ \mathtt{l4} = R \ \wedge \ B_p\left(\mathtt{l1}_p = G \ \wedge \ \mathtt{l2}_p = G \ \wedge \ \mathtt{l4}_p = R\right)$$

But more importantly,

$$\texttt{l3} = R \ \wedge \ \neg B_p\left(\texttt{l3}_p = G\right) \ \wedge \ \neg B_p\left(\texttt{l3}_p = R\right) \ \wedge \ P_p\left(\texttt{l3}_p = G\right) \ \wedge \ P_p\left(\texttt{l3}_p = R\right) \quad (3.3)$$

Because of the lighting conditions, we see by formula (3.3) that the pilot cannot infer whether the airplane is on or slightly above the glide path, though it certainly isn't under it. These doxastic statements can now be used by the pilot as the basic building blocks for decision making procedures.

For example, the pilot might decide to continue the approach since they know the airplane is not too far from where it needs to be, indicated by the two outer-most lights being red and green. As a further decision, the pilot might decide not to adjust the approach since the two confirmed green lights indicate the airplane is not lower than the glide path, at a danger of hitting obstacles like buildings or trees.

To illustrate just how far this notion of learning and believe can take us, imagine that this particular pilot considers an approach is *safe* as long as the airplane is not below the glide path:

$$\textit{safe-glideslope} \equiv \texttt{l1}_p = G \wedge \texttt{l2}_p = G$$

An approach is considered *reasonable* as long as the airplane is not very low or very high.

$$\textit{reasonable-glideslope} \equiv \texttt{l1}_p = G \wedge \texttt{l4}_p = R$$

This representation of reasonable glide-path, as belief, i.e. $B_p\left(\texttt{l1}_p = G \wedge \texttt{l4}_p = R\right)$ can also be represented as possibility, or rather, as impossibility: $\neg P_p\left(\texttt{l1}_p = R\right) \wedge \neg P_p\left(\texttt{l4}_p = G\right)$. These two are equivalent and it is up to the designer to choose that which they find to be most intuitive for the understanding of the system.

A simple program that reflects a starting point for sensible pilot decision procedures might look like this:

$$?\left(\neg B_p\left(\textit{safe-glideslope}\right)\right); \textit{yinput} := 1 \cup$$
$$?\left(B_p\left(\textit{reasonable-glideslope}\right)\right); \textit{yinput} := 0$$

Thus, if the pilot isn't certain that the approach is safe, they will decide climb away, and if the pilot is certain the approach is reasonable, then they may decide to maintain a neutral input.

But much can be done with the granularity of belief. For instance, if the pilot believes they are safe, but even considers it possible that the fourth light is green i.e. $B_p\left(\textit{safe-glideslope}\right) \wedge P_p\left(\texttt{l4}_p = G\right)$, meaning the aircraft is too high, then they could initiate a descent, $\textit{yinput} := -1$.

$$?(B_p\left(\textit{safe-glideslope}\right) \wedge P_p\left(\texttt{l4}_p = G\right)); \textit{yinput} := -1$$

This particular pilot is very cautious, and subtle changes in the formula can result in big behavioral changes. Suppose we transform

$$?\left(\neg B_p\left(\textit{safe-glideslope}\right)\right); \textit{yinput} := 1$$

into the much more reckless:

$$?\left(B_p\left(\neg \textit{safe-glideslope}\right)\right); \textit{yinput} := 1$$

This latter pilot will only decide to climb if they are absolutely certain that the aircraft is not safe. That is quite the departure from climbing if they cannot be certain that it is safe! How do these different policies affect the safety of the aircraft in the end? Are there conditions under which both the cautious and reckless pilots remain safe? Under which does the reckless pilot endanger the airplane?

It is these kind of questions and thought processes that we find are currently absent, or at most implicit, in CPS modeling and design. Here, on the other hand, having access to these logical primitives, we are encouraged to think about and explore them.

3.2 Syntax

The syntax of $\mathsf{d}^4\mathcal{L}$ will be that of $\mathsf{d}\mathcal{L}$, with doxastic modalities from doxastic logic and a learning operator that incorporates the intuitions from dynamic epistemic logics.

Terms in $\mathsf{d}^4\mathcal{L}$ are the terms of $\mathsf{d}\mathcal{L}$ with the addition of *doxastic variables* x_p, to be interpreted in the distinguished world. Let p be one agent, our pilot. Like in $\mathsf{d}\mathcal{L}$, let $\mathbb{V}$ be a countable set of state variable names, and Σ a countable set of logical variable names. Unlike $\mathsf{d}\mathcal{L}$, let $\mathbb{V}_p = \{x_p : x \in \mathbb{V}\}$ be the set of doxastic variables for the pilot. These need not be tied so tightly to $\mathbb{V}$, but doing so is a helpful reminder that these doxastic variables are often perceptions of something in the real world.

Definition 9 ($\mathsf{d}^4\mathcal{L}$ Terms)**.** *The terms θ of $\mathsf{d}^4\mathcal{L}$, with $\otimes \in \{+, -, \times, \div\}$, $X \in \Sigma$, $x \in \mathbb{V}$, and $x \in \mathbb{V}_p$ are defined as follows.*

$$\begin{array}{rcll} \theta & ::= & & \textit{(Expression)} \\ & & X & \textit{(Logical variable)} \\ & | & x & \textit{(State variable)} \\ & | & x_p & \textit{(Doxastic variable)} \\ & | & \theta \otimes \theta & \textit{(Function)} \end{array}$$

The formulas of $\mathsf{d}^4\mathcal{L}$ are those of $\mathsf{d}\mathcal{L}$ (c.f. Chapter 2) extended with the doxastic modalities $B(\phi)$ and $P(\phi)$ meaning, respectively, that the agent believes ϕ, and considers it possible that ϕ.

Definition 10 ($\mathsf{d}^4\mathcal{L}$ formulas)**.** *The formulas ϕ of $\mathsf{d}^4\mathcal{L}$, with θ a term, and $X \in \Sigma$, and α a doxastic hybrid program, are defined as follows.*

$$\begin{array}{rcll} \phi & ::= & & \textit{(Formula)} \\ & & \phi \vee \phi & \textit{(Disjunction)} \\ & | & \neg\phi & \textit{(Negation)} \\ & | & \forall X.\phi(X) & \textit{(Quantification)} \\ & | & [\alpha]\,\phi & \textit{(Program execution)} \\ & | & B_p(\phi) & \textit{(Belief)} \\ & | & \theta_1 < \theta_2 & \textit{(Comparison)} \\ & | & \theta_1 = \theta_2 & \textit{(Equality)} \end{array}$$

The most important difference from previous work is found in the *doxastic hybrid programs* that describe change in both belief and physical states. Here, hybrid indicates that they describe both discrete and continuous dynamics, and doxastic that the program can affect the belief state of the agent as well as the physical state.

We extend the hybrid programs of $\mathsf{d}\mathcal{L}$ with a learning operator $L(\gamma)$, where γ is a restricted $\mathsf{d}\mathcal{L}$ hybrid program. The learning operator *only* changes the belief state.

Definition 11 (Doxastic Hybrid Programs)**.** *Let $\mathbb{V}$ be a set of state variable names, let $x \in \mathbb{V}$, ϕ be a formula, and ψ be a formula not containing $L_p(\cdot)$, $B_p(\cdot)$ or $P_p(\cdot)$.*

$$
\begin{array}{rcll}
\alpha & ::= & & \textit{(Hybrid Program)} \\
 & & x := \theta & \textit{(Discrete physical state change)} \\
 & | & x' = f(x) \& \chi & \textit{(Continuous physical state change)} \\
 & | & ?\phi & \textit{(Test)} \\
 & | & \alpha; \alpha & \textit{(Sequential composition)} \\
 & | & \alpha \cup \alpha & \textit{(Nondeterministic choice)} \\
 & | & \alpha^* & \textit{(Nondeterministic repetition)} \\
 & | & L_p(\gamma) & \textit{(Learning)} \\
\\
\gamma & ::= & & \textit{(Learning program)} \\
 & & x_p := \theta & \textit{(Discrete doxastic state change)} \\
 & | & x_p := * & \textit{(Nondeterministic assignment)} \\
 & | & ?\psi & \textit{(Test)} \\
 & | & \gamma; \gamma & \textit{(Sequential composition)} \\
 & | & \gamma \cup \gamma & \textit{(Nondeterministic choice)}
\end{array}
$$

The restriction in which programs may be learned, i.e. the absence of nondeterministic repetition and differential equations, is not a restriction of some fundamental nature. Instead, it merely reflects those programs for which we have made successful progress in developing a calculus.

Because there are currently no learned differential equations, then $x_p := *$ cannot be derived, and is added to the syntax.

3.3 Semantics

The semantics of $\mathsf{d}^4\mathcal{L}$ will be those of $\mathsf{d}\mathcal{L}$, with additions for the doxastic modalities and the learning operator.

3.3.1 Models

The models of $\mathsf{d}^4\mathcal{L}$, called physical/doxastic models, juxtapose the models of $\mathsf{d}\mathcal{L}$ and those of a simplified Kripke modal semantics.

Definition 12 (Physical/doxastic model)**.** *A physical/doxastic or PD-model $\omega = \langle r, W, V, s \rangle$ is composed of*

- *$r : \mathbb{V} \to \mathbb{R}$, the true, physical state of the world*
- *W, a set of worlds, called the possible worlds*
- *$V : W \to (\mathbb{V}_x \to \mathbb{R})$, a valuation function that for each world $t \in W$ returns the agent's perceived value of the doxastic variable x_p associated with each state variable $x \in \mathbb{V}$, $V(t)(x_p)$*
- *$s \in W$ is a distinguished world*

PD-models are sufficient to give meaning to all terms, formulas and programs. They are, however, notationally heavy, so a few shorthands are in order.

We will often use ω, ν, μ as shorthand to denote a PD-model $\langle r, W, V, s\rangle$. Annotations in the shorthand will transfer to the structure, e.g. $\omega' = \langle r', W', V', s'\rangle$. Notation for doxastic state is also lifted, thus allowing statements such as $t \in \omega$ instead of $t \in W$, $\omega(x)$ instead of $r(x)$, and $\omega(t)(x_p)$ instead of $V(t)(x_p)$. The distinguished world is denoted $\text{DW}(\omega)$ and its distinguished valuation $\text{DV}(\omega) = \omega(\text{DW}(\omega)) = \omega(s) = V(s)$. The physical state is $\text{R}(\omega) = r$. Finally, $\langle r, W, V, s\rangle \oplus t = \langle r, W, V, t\rangle$ allows us to alter only the distinguished world.

Semantic equivalence of PD-models

In $\mathsf{d}\mathcal{L}$, it is a trivial thing to check whether two states ω and ν are equivalent. They are either the same valuation, i.e. for all variable x, $\omega(x) = \nu(x)$, or they are not.

PD-models slightly complicate things, because they contain many valuations for differently named worlds, so that the notion of semantic equivalence must be adapted.

Definition 13 (Semantic subsumption)**.** *Let ω and ν be two PD-models. We say that ω semantically subsumes ν, or just subsumes ν, denoted $\nu \sqsubseteq \omega$, if*

1. $\text{R}(\omega) = \text{R}(\nu)$*, as in $\mathsf{d}\mathcal{L}$, i.e. for all $x \in \mathbb{V}$, $\omega(x) = \nu(x)$*
2. *For all $t \in \nu$, there is $u \in \omega$ such that $\nu(t) = \omega(u)$*

Definition 14 (Semantic equivalence)**.** *Let ω and ν be two PD-models. We say that ω and ν are semantically equivalent, denoted $\omega \sim \nu$, if $\nu \sqsubseteq \omega$ and $\omega \sqsubseteq \nu$.*

The reader more familiar with epistemic or doxastic logics may already have an intuition for the use of these notions, and be aware of the following results.

Proposition 1. *Let ω, ν be two PD-models such that $\nu \sqsubseteq \omega$, then*

- $val_\eta(\nu, P_p(\phi)) = \mathtt{T}$ *implies* $val_\eta(\omega, P_p(\phi)) = \mathtt{T}$
- $val_\eta(\omega, B_p(\phi)) = \mathtt{T}$ *implies* $val_\eta(\nu, B_p(\phi)) = \mathtt{T}$

Thus, if a doxastic universe expands, then the witnesses for the possibilities that were there "before" remain. Conversely, if the doxastic universe contracts, elimination of possible worlds cannot create a witness for falsifying belief, and therefore belief is maintained.

3.3.2 Interpretation

The interpretation of terms is done as before, with doxastic variables given through worlds and valuations, and state variables by the physical state.

Definition 15 (Term interpretation)**.** *Let ω be a PD-model, η be a variable assignment, $\otimes \in \{+, -, \times, \div\}$, $x \in \mathbb{V}$, $x_p \in \mathbb{V}_p$ and $X \in \Sigma$. Then, the interpretation of terms is defined as follows.*

- $val_\eta(\omega, x) = \omega(x)$ *for state variable x*
- $val_\eta(\omega, X) = \eta(X)$ *for logical variable X*
- $val_\eta(\omega, x_p) = V(s)(x_p) = \text{DV}(\omega)(x_p)$ *for doxastic variable x_p and the distinguished world $s = \text{DW}(\omega)$*
- $val_\eta(\omega, \theta_1 \oplus \theta_2) = val_\eta(\omega, \theta_1) \oplus val_\eta(\omega, \theta_2)$

The interpretation of formulas is a direct adaptation from $\mathsf{d}\mathcal{L}$ and dynamic epistemic logics.

Definition 16 (Evaluation of formulas)**.** *Let $\omega = \langle r, W, V, s\rangle$ be a PD-model, η be a variable assignment, θ_1, θ_2 be terms, ϕ_1, ϕ_2 be formulas and α be a doxastic hybrid program. Then, the valuation of formulas is given as follows.*

$$
\begin{array}{lll}
val_\eta(\omega, \theta_1 = \theta_2) = \mathtt{T} & \textit{iff} & val_\eta(\omega, \theta_1) = val_\eta(\omega, \theta_2) \\
val_\eta(\omega, \theta_1 < \theta_2) = \mathtt{T} & \textit{iff} & val_\eta(\omega, \theta_1) < val_\eta(\omega, \theta_2) \\
val_\eta(\omega, \phi_1 \vee \phi_2) = \mathtt{T} & \textit{iff} & val_\eta(\omega, \phi_1) = \mathtt{T} \textit{ or } val_\eta(\omega, \phi_2) = \mathtt{T} \\
val_\eta(\omega, \neg\phi) = \mathtt{T} & \textit{iff} & val_\eta(\omega, \phi) = \mathtt{F} \\
val_\eta(\omega, \forall X.\phi) = \mathtt{T} & \textit{iff} & \textit{for all } v \in \mathbb{R}, val_{\eta^v_X}(\omega, \phi) = \mathtt{T} \\
val_\eta(\omega, B_p(\phi)) = \mathtt{T} & \textit{iff} & \textit{for all } t \in \omega, val_\eta(\omega \oplus t, \phi) = \mathtt{T} \\
val_\eta(\omega, [\alpha]\phi) = \mathtt{T} & \textit{iff} & \textit{for all } (\omega, \omega') \in \rho_\eta(\alpha), val_\eta(\omega', \phi)
\end{array}
$$

While there is no particular use for doxastic variables outside of doxastic modalities, there are important use cases for physical variables inside doxastic modalities, as we saw before. With the semantics in place, it is worth revisiting noisy sensors.

The following formula captures semi-accurate beliefs when "guesstimating" altitude on approach, where quick reflexes are important and looking at the altimeter may not be desirable

$$B_p\left(\mathrm{alt}_p^2 - \mathrm{alt}^2 < \epsilon^2\right)$$

The semantics of $B_p(\cdot)$ interpret the formula $\mathrm{alt}_p^2 - \mathrm{alt}^2 < \epsilon^2$ at each possible world. Thus, the formula states that the perceived altitude alt_p and the real altitude alt may not differ by more than ϵ at each world. This is an incomplete but not untruthful belief about the current state of the world.

A similar pattern will emerge in the context of the learning operator, which finally brings us to program semantics.

3.3.3 Intuition: learned nondeterminism as doxastic indistinguishability

We go back to the PAPI light example to illustrate this principle of learned nondeterminism as doxastic indistinguishability. The pilot still cannot recognize whether the third light is green or red. This would be represented by the following program:

$$L_p(\mathtt{l3}_p := G \cup \mathtt{l3}_p := R)$$

In essence, the pilot learns that either the light "turned" green, or "turned" red, but cannot tell which. Whatever the original doxastic state of the pilot, they should now consider it possible that the light is green, and possible that it is red. The nondeterminism of the program $\mathtt{l3}_p = G \cup \mathtt{l3}_p = R$ generates two reachable PD-models, one in which the distinguished world features the green light, and one in which it features the red light. The learning operator transforms each trace into a new possible world whose valuation in the reflects the changes to the distinguished valuation at the end of trace execution. All the resulting worlds are indistinguishable from each other: they are all in the same equivalence relation.

Figures 3.8 and 3.9 show just this phenomenon. Rectangles are PD-models with the physical state r omitted, and a circle for each possible world. We further annotate each world with the airplane's perceived altitude, and draw it in a color matching the third PAPI light's state at that world. A black circle simply indicates that the color is irrelevant for the example.

The thick arrows represent the transition using the program $L_p\left(\mathtt{l3}_p \coloneqq G \cup \mathtt{l3}_p \coloneqq R\right)$, whereas the thin arrows show the origin of each new possible world.

We thus have a visual representation for doxastic states, i.e. for the Kripke models, and of how they are affected by the learning operator.

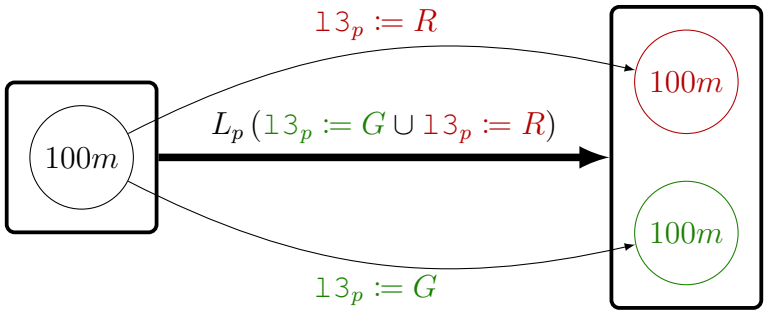


Figure 3.8: Doxastic change for $L_p\left(\mathtt{l3}_p \coloneqq G \cup \mathtt{l3}_p \coloneqq R\right)$

Here we can see a doxastic state where a pilot is sure of the plane's altitude at 100m and of the color of the third PAPI light, irrelevant whether red or green. Upon looking at the confusing light again, i.e. executing $L_p\left(\mathtt{l3}_p \coloneqq G \cup \mathtt{l3}_p \coloneqq R\right)$, the doxastic state is altered. Our confident pilot's very certain single possible world at 100m is replaced by two worlds, each with its own color but both at 100m since the program does not affect altitude.

The doxastic state finds itself expanded to two worlds, and thus uncertainty is born! The pilot must now consider both worlds, $P_p\left(\mathtt{l3}_p = G\right) \wedge P_p\left(\mathtt{l3}_p = R\right)$, and therefore can't be sure of the color, $\neg B_p\left(\mathtt{l3}_p = G\right) \wedge \neg B_p\left(\mathtt{l3}_p = R\right)$. The altitude of 100 meters, being equal among all worlds, remains certain, $B_p\left(alt_p = 100\right)$.

In the examples of Figure 3.9, we omit the program labels for clarity. The pilot's initial belief state already contains some uncertainty. This helps illustrate how the learning operator handles multiple initial possible worlds. The learning operator $L_p\left(\alpha\right)$ behaves similarly to the doxastic modality $B\left(\phi\right)$ in that it interprets (i.e. "executes") program α at each world $t \in \omega$. It then turns each result into a new possible world.

The pilot imagines the effect of α at every possible world: if α has no transitions, the origin world disappears. If α is deterministic, a single new possible world results from each original world. If α is nondeterministic, each reachable state creates a new possible world: learned nondeterminism as doxastic indistinguishability.

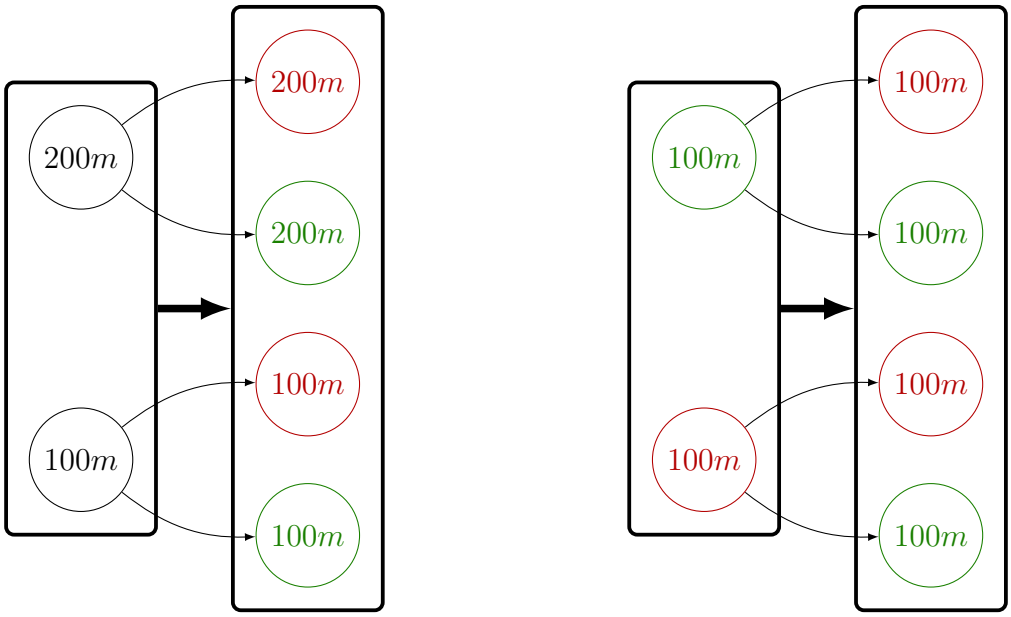


Figure 3.9: More complex doxastic change for $L_p\,(\mathtt{l3}_p \coloneqq G \cup \mathtt{l3}_p \coloneqq R)$

On the left figure, the pilot is unsure of the airplane's altitude since it could be at 100 or 200 meters. Depending on the colors at each of the worlds, the pilot might be certain or uncertain of the third PAPI light. Once the pilot fails to ascertain the light's color, the uncertainty about the altitude at 100m or 200m remains, as it should, but whatever the altitude, the color of the light must now be uncertain as well. Thus, for each case, 100m or 200m, the light may either be green or Red, for a total of 4 possibilities and 4 possible worlds.

On the right figure, the pilot knows that the airplane is at 100 meters, but does not know the color of the light. It might be surprising that this results in two sets of worlds that represent essentially the same state, e.g. two worlds in which the airplane is at 100 meters and where the light is red.

The reason is that the naming scheme for possible worlds after learning includes their origin world. This allows us to easily refer back to original world in proofs and establish properties about the new worlds from properties about the origin worlds.

Having the ability to distinguish between worlds with the same valuations is important because in modal logics with multiple agents (future work!), such worlds may have distinct accessibility relation connections, and thus affect the interpretation of modal formulas in different ways.

3.3.4 Program semantics

We finally introduce formal program semantics. These are exactly the semantics of $\mathsf{d}\mathcal{L}$ with extra cases for doxastic assignment and learning.

Definition 17 (Transition semantics)**.** *Let* $\omega = \langle r, W, V, s\rangle$*. The transition relation for doxastic dynamic programs is given by:*

- $(\omega, \omega') \in \rho_\eta\,(x \coloneqq \theta)$ *iff* $\omega' = \omega$ *except* $\mathrm{R}(\omega') = \mathit{val}_\eta\,(\omega, \theta)$

- $(\omega, \omega') \in \rho_\eta(x_p \coloneqq \theta)$ *iff* $\omega' = \omega$ *except* $\omega'(s)(x_p) = val_\eta(\omega, \theta)$
- $(\omega, \omega') \in \rho_\eta(x_p \coloneqq *)$ *iff* $\omega' = \omega$ *except* $\omega'(s)(x_p) = v$ *for some* $v \in \mathbb{R}$.
- $(\omega, \omega') \in \rho_\eta(x' = \theta \;\&\; H)$ *iff there is* $y : D \to \mathbb{R}$, *with* $n = |\mathbb{V}|$, *a solution to the IVP in* $D = [0, \tau^f]$, *such that* $r' = r_x^{y(\tau^f)}$ *and for all* $0 \leq \tau \leq \tau^f$, $val_\eta\left(\langle r_x^{y(\tau)}, W, V, s\rangle, H\right) = \mathtt{T}$, $M' = M$ *and* $s' = s$.
- $(\omega, \omega) \in \rho_\eta(?\phi)$ *iff* $val_\eta(\omega, \phi) = \mathtt{T}$
- $\rho_\eta(\alpha_1; \alpha_2) = \rho_\eta(\alpha_1) \circ \rho_\eta(\alpha_2) = \{\omega_3 :$ *there is* ω_2 *s.t.* $(\omega_1, \omega_2) \in \rho_\eta(\alpha_1)$ *and* $(\omega_2, \omega_3) \in \rho_\eta(\alpha_2)\}$
- $\rho_\eta(\alpha_1 \cup \alpha_2) = \rho_\eta(\alpha_1) \cup \rho_\eta(\alpha_2)$
- $(\omega, \omega') \in \rho_\eta(\alpha^*)$ *iff there is a* $n \in \mathbb{N}$ *such that* $(\omega, \omega') \in \rho_\eta(\alpha^n)$, *see footnote*[2]
- $(\omega, \omega') \in \rho_\eta(L(\gamma))$ *iff all of the following:*
  - $r' = r$
  - $W' = \{\nu :$ *there is* $t \in \omega$ *s.t.* $(\omega \oplus t, \nu) \in \rho_\eta(\gamma)\}$
  - $\omega'(\nu) = \mathrm{DV}(\nu)$ *for all* $\nu \in \omega'$
  - $\mathrm{DW}(\mathrm{DW}(\omega')) = \mathrm{DW}(\omega)$

In this definition, when we say that $\omega' = \omega$, we mean to say that the physical state is the same, $\mathrm{R}(\omega') = \mathrm{R}(\omega)$; the set of worlds is the same, $W(\omega') = W(\omega)$; the distinguished world is the same, $\mathrm{DW}(\omega') = \mathrm{DW}(\omega)$; and for every world $t \in \omega$, all doxastic variables have the same value, $\omega'(t) = \omega(t)$.

There is a lot to unpack in the definition for the learning operator: what constitutes a possible world in ω', what the valuation of each of those worlds should be, and the choice of the distinguished world.

While analyzing the definition, we will remain attentive to whether each of these components may be a source of nondeterminism.

Possible worlds

$$W' = \{\nu : \text{ there is } t \in \omega \text{ s.t. } (\omega \oplus t, \nu) \in \rho_\eta(\gamma)\}$$

To create a possible world for ω', after the execution of $L_p(\gamma)$, we choose a possible world t of the original PD-model ω, $t \in \omega$. Then, we make that the distinguished world for ω, i.e. $\omega \oplus t$, and execute program γ. This results a new PD-model ν (with an altered distinguished valuation, such as in Figure 3.10), which becomes a world of ω'.

The set of all worlds for ω' is obtained by doing this for each world of ω. Thus, each execution of γ, for each choice of distinguished world of ω, becomes a possible world of ω'. When γ contains nondeterminism, each origin world may result in multiple worlds, which are PD-model with the same distinguished world, e.g. $\nu_1, \nu_2 \in \omega'$ and $\mathrm{DW}(\nu_1) = \mathrm{DW}(\nu_2)$.

The following figure shows a diagram for when α is doxastic assignment.

[2] $\alpha^0 = ?\mathtt{T}, \alpha^1 = \alpha, \alpha^2 = \alpha; \alpha$, etc

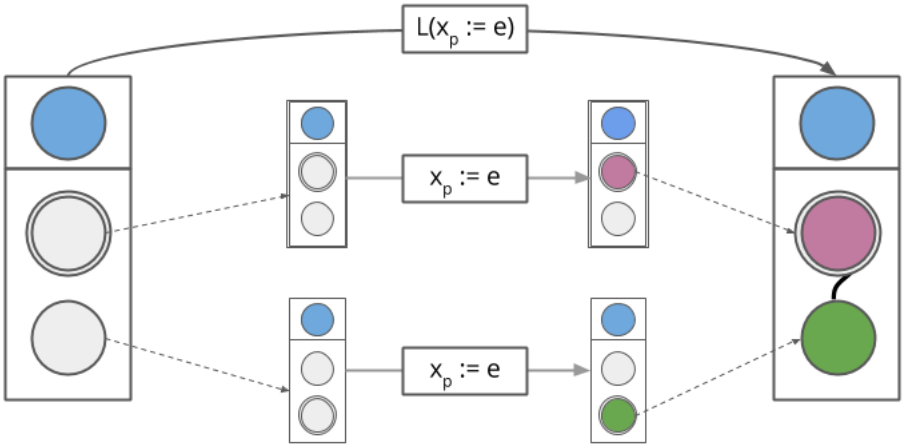


Figure 3.10: The transition of $L_p\,(x_p := \theta)$

So, the *names* of the worlds of $\nu \in \omega'$ are PD-models themselves. Because no atomic doxastic programs change the choice of distinguished world, then the distinguished world of each world/PD-model $\nu \in \omega'$ is the origin world of ν in ω:

$$\text{DW}(\nu) = t \text{ when } (\omega \oplus t, \nu) \in \rho_\eta\,(\gamma)$$

We may refer back to the "origin" world with $\text{DW}(\nu)$, and its original valuation $\omega(\text{DW}(\nu))$. As previously mentioned, this will be very useful for proofs, since we can infer the changes wrought by γ given a starting point of $\omega(\text{DW}(\nu))$. As a corollary and a curiosity, each PD-model contains embedded in its worlds' names the entire history of doxastic change since the starting PD-model.

Computing the set of worlds $W(\omega')$ is deterministic, as each resolution of nondeterministic in γ is "captured" as a world in $W(\omega')$.

Valuation

$$\omega'(\nu) = \text{DV}(\nu) \text{ for all } \nu \in \omega'$$

In the previous subsection, we noted how each trace of program γ creates a new world ν, which is also a PD-model, by changing the valuation of its distinguished world. We wish to see those changes to $\text{DV}(\nu)$ reflected in the final model ω'. Thus, we "lift" them from the possible world ν up to the resulting model ω', by copying them: $\omega'(\nu) = \text{DV}(\nu)$.

Clearly, these valuations are obtained deterministically.

Distinguished world

$$\text{DW}(\text{DW}(\omega')) = \text{DW}(\omega)$$

The intuition here is that the distinguished world of ω' should have originated from the distinguished world of ω. If the distinguished world originally had the plane at 100m, and γ was $\texttt{l3}_p := G \cup \texttt{l3}_p := R$, then we can still expect that the new distinguished world has the plane

at 100m, independently of the new light color. If we use our usual notation of $\text{DW}(\omega) = s$ and $\text{DW}(\omega') = s'$, we may simplify this slightly:

$$\text{DW}(s') = s$$

In the previous section, we showed that the distinguished world of any $\nu \in \omega'$ is the world $t \in \omega$ from which it originated. In particular, s' is one such world, and therefore the statement that $\text{DW}(s') = s$ indicates that the origin of s' was $\text{DW}(\omega) = s$.

It is now clear that $\text{DW}(\omega')$ must be some PD-model $\nu \in \omega'$ that originated from the distinguished world of ω, i.e. $(\omega, \nu) \in \rho_\eta(\gamma)$, and thus $\text{DW}(\nu) = \text{DW}(\omega)$.

This line of reasoning is encapsulated in the following equalities.

$$s = \text{DW}(\omega) = \text{DW}(\nu) = \text{DW}(s') = \text{DW}(\text{DW}(\omega'))$$

Let us now think about nondeterminism for the choice of distinguished world. Consider the simple nondeterministic transition below.

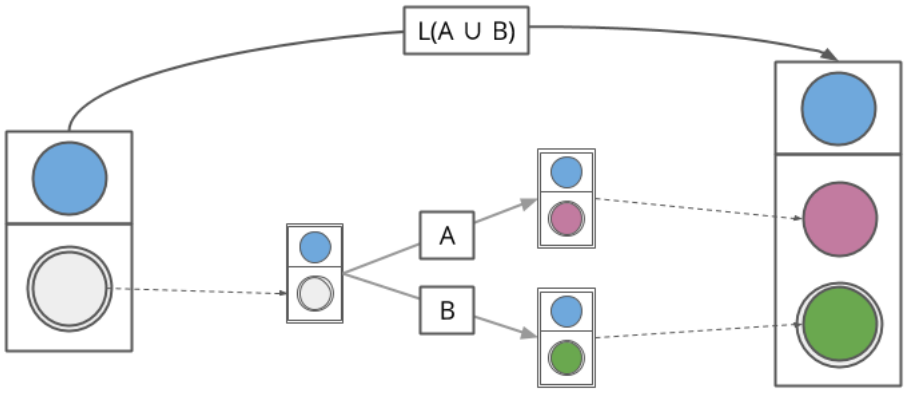


Figure 3.11: The transition of $L_p(\alpha \cup \beta)$

With a single choice of distinguished world for ω, we still see that $\gamma \equiv \alpha \cup \beta$ generates two possible worlds. Since they both originate from $\text{DW}(\omega)$, either could be chosen as $\text{DW}(\omega')$.

Therefore, while all other parts of the learning operator semantics were shown to be deterministic, there is some nondeterminism in the choice of distinguished world.

For $L(?\phi)$, there may be no transition at all if $\text{DW}(\omega)$ does not satisfy ϕ. However, different choices of distinguished world earlier in a program's execution might have satisfied ϕ, and thus allowed the program to transition. Unfortunately, we cannot easily infer which choices are more likely to satisfy later tests. We thus allow nondeterminism to make the choice of distinguished world, and let dynamic modalities handle this nondeterminism like dℒ's nondeterminism.

As an illustration of the above remark, suppose that our pilot estimates that the airplane is either at 100, 150 or 200 meters.

$$L_p(alt_p := 100 \cup alt_p := 150 \cup alt_p := 200)$$

Upon looking at the altimeter, however, the pilot is now convinced the altitude must be above 100 meters.

$$L_p\,(?alt_p > 100)$$

Upon execution of the first program, the nondeterministic choice for distinguished world may have it such that $alt = 100$. When reaching the second program, the program will thus be unable to transition, since the distinguished world falsifies $alt > 100$. Any other choice of distinguished world, however, would have allowed a transition.

With this program within a box dynamic modality, e.g.

$$[L_p\,(alt_p := 100 \cup alt_p := 150 \cup alt_p := 200)\,;L_p\,(?alt_p > 100)]\,\phi$$

we ensure that all such possibilities are considered when evaluating ϕ.

3.3.5 Extended semantics example

To better cement an understanding of these semantics, we now present a more thorough example.

Let us expand the example of Figure 3.8: we now have an initial state in which the airplane is believed to be at 125m, represented in Figure 3.12. Because the airplane is on final approach, the pilot estimates the airplane to have descended to 100m, i.e. $L_p\,(alt_p := 100)$. However, the change in altitude makes the famous third PAPI light's color unrecognizable, i.e. $L_p\,(\mathtt{l3}_p := G \cup \mathtt{l3}_p := R)$. Finally, the plane descends again and the pilot now believes the altitude to be 75m, while the color of the light remains a mystery, i.e. $L_p\,(alt_p := 75)$.

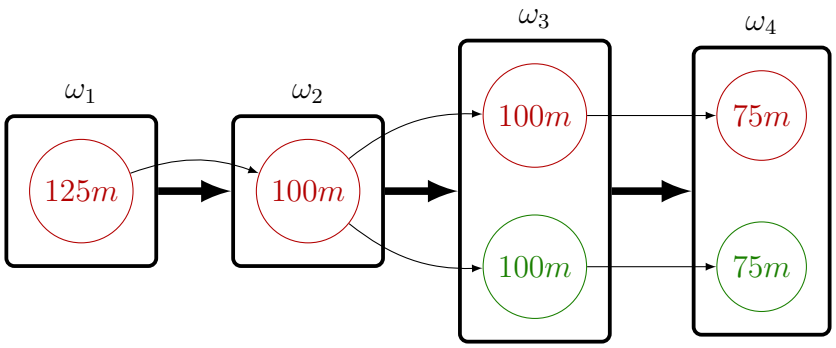


Figure 3.12: Doxastic change for $L_p\,(alt_p := 100)\,;L_p\,(\mathtt{l3}_p := G \cup \mathtt{l3}_p := R)\,;L_p\,(alt_p := 75)$

The first PD-model, ω_1, is easy to describe. The set of possible worlds contains a single world with a unique starting name, in this case arbitrarily chosen to be t_1^a.

$W(\omega_1) = \{t_1^a\}$, with

- $\omega_1(t_1^a)(alt_p) = 125m$
- $\omega_1(t_1^a)(\mathtt{l3}_p) = R$

The second PD-model, ω_2, is a little more complicated. It is obtained by executing program $alt_p := 100$ at the single possible world of ω_1.

$W(\omega_2) = \{t_2^a\}$, with

- $t_2^a = \langle r, W_2^a, V_2^a, t_1^a \rangle$ because $(\omega_1 \oplus t_1^a, \langle r, W_2^a, V_2^a, t_1^a \rangle) \in \rho_\eta\,(alt_p := 100)$
- $\omega_2(t_2^a)(alt_p) = t_2^a(\text{DW}(t_2^a))(alt_p) = t_2^a(t_1^a)(alt_p) = V_2^a(t_1^a)(alt_p) = 100m$
- $\omega_2(t_2^a)(\texttt{l3}_p) = t_2^a(t_1^a)(\texttt{l3}_p) = \omega_1(t_1^a)(\texttt{l3}_p) = R$, since doxastic assignment affects only the assigned variable

The third PD-model, ω_3, is produced using the nondeterministic choice in $L_p\,(\texttt{l3}_p := G \cup \texttt{l3}_p := R)$ and contains two possible worlds with references to the same origin world, but different valuations of their distinguished worlds.

$W(\omega_3) = \{t_3^a, t_3^b\}$, with

- $t_3^a = \langle r, W_3^a, V_3^a, t_2^a \rangle$ because $(\omega_2 \oplus t_2^a, \langle r, W_3^a, V_3^a, t_2^a \rangle) \in \rho_\eta\,(\texttt{l3}_p := G \cup \texttt{l3}_p := R)$ with
  - $\omega_3(t_3^a)(alt_p) = t_3^a(t_2^a)(alt_p) = \omega_2(t_2^a)(alt_p) = 100m$, because alt is not bound in $\texttt{l3}_p := G \cup \texttt{l3}_p := R$.
  - $\omega_3(t_3^a)(\texttt{l3}_p) = t_3^a(\text{DW}(t_3^a))(\texttt{l3}_p) = t_3^a(t_2^a)(\texttt{l3}_p) = V_3^a(t_2^a)(\texttt{l3}_p) = R$, because $\texttt{l3}_p$ is bound in $\texttt{l3}_p := G \cup \texttt{l3}_p := R$ and we "went right".
- $t_3^b = \langle r, W_3^b, V_3^b, t_2^a \rangle$ because $(\omega_2 \oplus t_2^a, \langle r, W_3^b, V_3^b, t_2^a \rangle) \in \rho_\eta\,(\texttt{l3}_p = G \cup \texttt{l3}_p = R)$ with
  - $\omega_3(t_3^b)(alt_p) = \omega_2(t_2^a)(alt_p) = 100m$, for the same reason as in t_3^a.
  - $\omega_3(t_3^b)(\texttt{l3}_p) = \omega_3(\langle r, W_3^b, V_3^b, t_2^a \rangle)(\texttt{l3}_p) = V_3^b(\texttt{l3}_p) = G$, because $\texttt{l3}$ is bound in $\texttt{l3}_p = G \cup \texttt{l3}_p = R$ and we "went left".

Finally, we reach the state ω_4, obtained by executing $L_p\,(alt_p := 75)$, in which the pilot notices another decrease in altitude but is still unable to determine the color of the light. This illustrates multiple possible worlds with differing origin worlds.

$W(\omega_4) = \{t_4^a, t_4^b\}$, with

- $t_4^a = \langle r, W_4^a, V_4^a, t_3^a \rangle$ because $(\omega_3 \oplus t_3^a, \langle r, W_4^a, V_4^a, t_3^a \rangle) \in \rho_\eta\,(alt_p := 75)$ with
  - $\omega_4(t_4^a)(alt_p) = \omega_4(\langle r, W_4^a, V_4^a, t_3^a \rangle)(alt_p) = V_4^a(t_3^a)(alt_p) = 75m$, because alt_p is bound.
  - $\omega_4(t_4^a)(\texttt{l3}_p) = \omega_4(\langle r, W_4^a, V_4^a, t_3^a \rangle)(\texttt{l3}_p) = \omega_3(t_3^a)(\texttt{l3}_p) = R$, because $\texttt{l3}_p$ is not bound.
- $t_4^b = \langle r, W_4^b, V_4^b, t_3^b \rangle$ because $(\omega_3 \oplus t_3^b, \langle r, W_4^b, V_4^b, t_3^b \rangle) \in \rho_\eta\,(alt_p := 75)$ with
  - $\omega_4(t_4^b)(alt_p) = \omega_4(\langle r, W_4^b, V_4^b, t_3^b \rangle)(alt_p) = V_4^b(t_3^b)(alt_p) = 75m$, because alt_p is bound.
  - $\omega_4(t_4^b)(\texttt{l3}_p) = \omega_4(\langle r, W_4^b, V_4^b, t_3^b \rangle)(\texttt{l3}_p) = \omega_3(t_3^b)(\texttt{l3}_p) = G$, because $\texttt{l3}_p$ is not bound.

As far as distinguished worlds are concerned, it is clear that both t_1^a and t_2^a are the only options for ω_1 and ω_2 respectively. Because both t_3^a and t_3^b originate from t_2^a, i.e. $\text{DW}(t_3^a) = \text{DW}(t_3^b) = t_2^a$, either may be chosen as $\text{DW}(\omega_3)$. That choice, however, entirely determines $\text{DW}(\omega_4)$. If $\text{DW}(\omega_3) = t_3^a$, then $\text{DW}(\omega_4) = t_4^a$ whereas if $\text{DW}(\omega_3) = t_3^b$, then $\text{DW}(\omega_4) = t_4^b$ since $\text{DW}(t_4^a) = t_3^a \neq t_3^b = \text{DW}(t_4^b)$.

3.4 Sound proof calculus

A sound proof calculus is a set of proof rules that have been shown to be sound. Proof rules will look like the following template.

$$\frac{\Gamma_1 \vdash \phi_1}{\Gamma_2 \vdash \phi_2}\,(PR)$$

This introduces us to the notation for a sequent $\Gamma \vdash \phi$. Intuitively, a sequent $\Gamma \vdash \phi$ is syntax that represents the semantic notion that ϕ, also called the consequent, is implied by Γ, a set of formulas also called the antecedent.

Definition 18 (Sequent Evaluation)**.** *Let ω, η, and let $\Gamma \vdash \phi$ be a sequent, with $\Gamma = \{\psi_1, ..., \psi_n\}$ a set of formulas. Then, $val_\eta(\omega, \Gamma \vdash \phi) = \mathtt{T}$ under the following condition:*

If for all $\psi \in \Gamma$, $val_\eta(\omega, \psi) = \mathtt{T}$, then $val_\eta(\omega, \phi) = \mathtt{T}$.

Semantically, then $\Gamma \vdash \phi \equiv \psi_1 \wedge ... \wedge \psi_n \rightarrow \phi$.

3.4.1 Soundness

We will define two notions of soundness for a proof rule PR, one stronger than another. What type of soundness is possible for each rule can give us a better understanding of the properties of the program being addressed by the rule.

We begin with global soundness, which requires the conclusion of the rule to be true *always*.

Definition 19 (Global Soundness)**.** *We say that the proof rule PR is globally sound when:*

if for all ω and η, $val_\eta(\omega, \Gamma_1 \vdash \phi_1) = \mathtt{T}$
then for all ω and η, $val_\eta(\omega, \Gamma_2 \vdash \phi_2) = \mathtt{T}$.

Local soundness, on the other hand, does not require the conclusion of the rule to be true in every combination of PD-model and variable assignment. It only needs to be true under the same conditions as the top of the rule is.

Definition 20 (Local Soundness)**.** *We say that the proof rule PR is locally sound when for all ω and η,*

if $val_\eta(\omega, \Gamma_1 \vdash \phi_1) = \mathtt{T}$ then $val_\eta(\omega, \Gamma_2 \vdash \phi_2) = \mathtt{T}$

The two notions of soundness are related, with local soundness being the stronger condition.

Proposition 2. *If PR is locally sound, then PR is globally sound.*

Proof. Let PR be locally sound. Then, for every ω and η, if $\text{val}_\eta(\omega, \Gamma_1 \vdash \phi_1) = \mathtt{T}$, then also $\text{val}_\eta(\omega, \Gamma_2 \vdash \phi_2) = \mathtt{T}$.

Let us now distinguish two cases:

1. It is *not* the case that for all ω and η, $\text{val}_\eta(\omega, \Gamma_1 \vdash \phi_1) = \mathtt{T}$. Then global soundness is trivially satisfied.
2. It *is* the case that for all ω and η, $\text{val}_\eta(\omega, \Gamma_1 \vdash \phi_1) = \mathtt{T}$. Then, using local soundness at each of those choices of ω and η, we can conclude that $\text{val}_\eta(\omega, \Gamma_2 \vdash \phi_2) = \mathtt{T}$.
   Thus, for all ω and η, $\text{val}_\eta(\omega, \Gamma_2 \vdash \phi_2) = \mathtt{T}$, and the rule is globally sound.

□

3.4.2 Preliminary Calculus

We adopt the sound calculus of $\mathsf{d}\mathcal{L}$ to deal with all programs excluding the learning operator, and formulas excluding belief and possibility. Instead of full doxastic assignment, the current calculus handles a limited form of assignment that we call non self-referential assignment.

The axioms of S5 logic can be used to deal with doxastic modalities which do not contain dynamic modalities within them such as $B_p([\alpha]\phi)$, and with state variables considered as constants.

$$\frac{\Gamma\backslash_{x_p} \vdash B_p(x_p = \theta) \to \phi}{\Gamma \vdash [L_p(x_p := \theta)]\phi}([] :=^{\text{NSR}}) \qquad \frac{\Gamma\backslash_{x_p} \vdash B_p(x_p = \theta) \to \phi}{\Gamma \vdash \langle L_p(x_p := \theta)\rangle \phi}(\langle\rangle :=^{\text{NSR}})$$

$$\frac{\Gamma_R;\Gamma_B\backslash_{x_p};\Gamma_P;\Gamma_O\backslash_{x_p} \vdash \phi}{\Gamma \vdash [L_p(x_p := *)]\phi}([]L := *) \qquad \frac{\Gamma_R;\Gamma_B\backslash_{x_p};\Gamma_P;\Gamma_O \vdash \phi}{\Gamma \vdash \langle L_p(x_p := *)\rangle \phi}(\langle\rangle L := *)$$

$$\frac{\Gamma_R;\Gamma_B;\Gamma_O \vdash B_p(\xi) \to \psi}{\Gamma_R;\Gamma_B;\Gamma_P;\Gamma_O \vdash [L_p(?\phi)]\psi}([]L?) \qquad \frac{\Gamma_R;\Gamma_B;\Gamma_O \vdash B_p(\xi) \wedge \psi}{\Gamma_R;\Gamma_B;\Gamma_P;\Gamma_O \vdash \langle L_p(?\phi)\rangle \psi}(\langle\rangle L?)$$

$$\frac{\Gamma \vdash [L_p(\alpha);L_p(\beta)]\phi}{\Gamma \vdash [L_p(\alpha;\beta)]\phi}([]L;) \qquad \frac{\Gamma \vdash \langle L_p(\alpha);L_p(\beta)\rangle \phi}{\Gamma \vdash \langle L_p(\alpha;\beta)\rangle \phi}(\langle\rangle L;)$$

$$\frac{\Gamma \vdash [L_p(\alpha)]B_p(\xi) \wedge [L_p(\beta)]B_p(\xi)}{\Gamma \vdash [L_p(\alpha\cup\beta)]B_p(\xi)}([]LB\cup) \qquad \frac{\Gamma \vdash \langle L_p(\alpha)\rangle B_p(\xi) \wedge \langle L_p(\beta)\rangle B_p(\xi)}{\Gamma \vdash \langle L_p(\alpha\cup\beta)\rangle B_p(\xi)}(\langle\rangle LB\cup)$$

$$\frac{\Gamma \vdash [L_p(\alpha)]P_p(\phi) \wedge [L_p(\beta)]P_p(\phi)}{\Gamma \vdash [L_p(\alpha\cup\beta)]P_p(\phi)}([]LP\cup) \qquad \frac{\Gamma \vdash \langle L_p(\alpha)\rangle P_p(\phi) \vee \langle L_p(\beta)\rangle P_p(\phi)}{\Gamma \vdash \langle L_p(\alpha\cup\beta)\rangle P_p(\phi)}(\langle\rangle LP\cup)$$

$$\frac{\Gamma \vdash [L_p(\alpha)]\phi \wedge [L_p(\beta)]\phi}{\Gamma \vdash [L_p(\alpha\cup\beta)]\phi}([]L\cup)^{3} \qquad \frac{\Gamma \vdash \langle L_p(\alpha)\rangle \phi \vee \langle L_p(\beta)\rangle \phi}{\Gamma \vdash \langle L_p(\alpha\cup\beta)\rangle \phi}(\langle\rangle L\cup)^{3}$$

Figure 3.13: Dynamic doxastic fragment of the $\mathsf{d}^4\mathcal{L}$ calculus

This preliminary calculus has already been shown to be sound. We now provide a few example proofs, starting with a limited, but still useful, form of assignment.

Non self-referential (NSR) assignment

The rules for regular assignment require a substitution lemma which states that syntactic substitution is equivalent to semantic substitution. Substitution lemmas are known to be difficult pieces of mathematics, and that is certainly the case here. This justifies why a proof of the substitution lemma is proposed, partially done, but not yet finished. See Section 3.4.3.

However, there are less general forms of assignment which are still interesting and can be obtained with much simpler proofs. This is the case for non self-referential assignments, by which we mean assignments $x_p := \theta$, where x_p cannot appear in θ. The semantics of this assignment are the same as those for regular assignment from Definition 17.

While incrementing a variable with $x_p := x_p + 1$ is no longer possible, we argue that a great many scenarios can make use of NSR assignment exclusively. For instance, any time that a pilot

[3] formula ϕ does not contain doxastic modalities or variables, or learning operators

is relying on instruments, they do not need to remember the previous value of a sensor in order to compute its new value: what they see is what it is.

The reason that NSR assignment makes for a proof rule that is more easily provable to be sound is because we need not worry about how syntactically substituting the occurrences of x_p by θ will affect the interpretation of formulas that rely on x_p or on the variables that occur in θ.

Instead, because the assignment is NSR, we may simply contract our contexts to "forget" everything we know about the variable being assigned to.

$$\frac{\Gamma\backslash_{x_p} \vdash B_p\left(x_p = \theta\right) \to \phi}{\Gamma \vdash \left[L_p\left(x_p := \theta\right)\right]\phi}\left([] :=^{\text{NSR}}\right)$$

The set $\Gamma\backslash_{x_p}$ is the set Γ restricted only to formulas that do *NOT* contain an occurrence of x_p.

Global soundness for $([] :=^{NSR})$. Let η, ω. Assume $\text{val}_\eta\left(\omega, \Gamma\right) = \mathtt{T}$, shortened to $\text{val}_\eta\left(\omega, \Gamma\right)$. We must show $\text{val}_\eta\left(\omega, \left[L_p\left(x_p := \theta\right)\right]\phi\right)$. Since doxastic assignment is deterministic, there is a single transition $\left(\omega, \omega'\right) \in \rho_\eta\left(L_p\left(x_p := \theta\right)\right)$, for which we must show that $\text{val}_\eta\left(\omega', \phi\right)$. By Definition 17,

$$W(\omega') = \{t' : \text{ there is } t \in \omega \text{ s.t. } \left(\omega \oplus t, t'\right) \in \rho_\eta\left(x_p := \theta\right)\} \tag{3.4}$$

$$\omega'(t') = \omega(\text{DW}(t')) \text{ for all } t' \in \omega', \text{ except } \omega'(t')(x_p) = \text{val}_\eta\left(\omega \oplus \text{DW}(t'), \theta\right) \tag{3.5}$$

From (3.4) and the semantics of NSR assignment, for all $t' \in \omega'$, $\text{DW}(t') = t$, meaning t is the "origin" world of t'. It will be useful to relate the two.

We will show that in $\text{val}_\eta\left(\omega', B_p\left(x_p = \theta\right)\right)$ so that we may apply the hypothesis. By (3.4) there is a bijection between the set of worlds of ω and ω', and by (3.5), only the interpretation of x_p changes from ω to ω'.

Because this is NSR assignment, θ cannot contain x_p, and thus for all $t' \in \omega'$, $\omega'(t')(x_p) = \text{val}_\eta\left(\omega' \oplus t', x_p\right) = \text{val}_\eta\left(\omega' \oplus t', \theta\right) = \text{val}_\eta\left(\omega \oplus \text{DW}(t'), \theta\right) = \text{val}_\eta\left(\omega \oplus t, \theta\right)$. It follows that $\text{val}_\eta\left(\omega', B_p\left(x_p = \theta\right)\right)$. By the assumption $\text{val}_\eta\left(\omega, \Gamma\right)$ and $\Gamma\backslash_{x_p} \subseteq \Gamma$, trivially $\text{val}_\eta\left(\omega, \Gamma\backslash_{x_p}\right)$. Because only x_p changed from ω to ω', and because x_p cannot occur in $\Gamma\backslash_{x_p}$, then $\text{val}_\eta\left(\omega', \Gamma\backslash_{x_p}\right)$ by coincidence lemma.

A direct application of the proof rule's hypothesis to $\text{val}_\eta\left(\omega', \Gamma\backslash_{x_p}\right)$ and $\text{val}_\eta\left(\omega', B_p\left(x_p = \theta\right)\right)$ gives us $\text{val}_\eta\left(\omega', \phi\right)$. □

Nondeterministic doxastic assignment

Nondeterministic doxastic assignment $x_p := *$ allows programs to completely "reset" the state of a variable such that it can take any value whatsoever. This is useful since tests reduce doxastic universes, and while nondeterministic choice can expand them, it does only finitely.

When learned tests are found within a nondeterministic repetition, e.g. $\left(L_p\left(?\phi\right)\right)^*$, it is possible for the doxastic universe to progressively contract until there are no more transitions, and for a formula to thus become trivially true. Adding nondeterministic assignment short-circuits such phenomena, with programs such as $L_p\left(x_p := *; ?\phi(x_p)\right)^*$ always having transitions so long as $\phi(x_p)$ is a sensible formula.

The proof rule for nondeterministic assignment $[]L \coloneqq *$ is based on the insight that when a doxastic variable x_p may take any possible value, the doxastic universe expands, and thus any *beliefs* about that variable are voided, and must be removed from the context. Possibilities, however, remain unaffected: any witnesses prior to nondeterministic doxastic assignment can be found after the assignment. This proof sketch follows the same lines as that for $[] \coloneqq^{\text{NSR}}$.

Proof sketch for $[]L \coloneqq *$. We must show $\text{val}_\eta\left(\omega', \Gamma_R; \Gamma_B\backslash_{x_p}; \Gamma_P; \Gamma_O\backslash_{x_p}\right)$. This is trivial for Γ_R, $\Gamma_B\backslash_{x_p}$ and $\Gamma_O\backslash_{x_p}$ since x_p does not occur therein, and it was the only change from ω to ω'. We know by assumption that $\text{val}_\eta(\omega, \Gamma_P)$, and thus for each $P_p(\phi) \in \Gamma_P$, there is $t \in \omega$ such that $\text{val}_\eta(\omega \oplus t, \phi)$. Let $v = \omega(t)(x_p)$ be the value that contributes to satisfying ϕ.

By the semantics of the learning operator and nondeterministic doxastic assignment, there will be $t' \in \omega'$, with $\text{DW}(t') = t$, such that $\omega'(t')(x_p) = v'$ for all $v' \in \mathbb{R}$, and in particular, for $v' = v$. Since only x_p has changed from ω to ω', the world $t' \in \omega'$ serves as the witness for $\text{val}_\eta(\omega', P_p(\phi))$. □

It is interesting that in the soundness proof for $\langle\rangle L \coloneqq *$, we need not remove any formulas from Γ_O: the $\langle\rangle$ modality allows us to pick the distinguished world that satisfies the formulas in Γ_O, not unlike what we did for $[]L \coloneqq *$.

Test

The formula $[L_p(?\phi)]\,\psi$ states that after learning that ϕ is true, ψ must hold. Given the semantics of the learning operator, this is intuitively equivalent to assuming that ϕ holds in order to prove ψ. This is exactly what the following proof rules represent.

$$\frac{\vdash B_p(\phi) \to \psi}{\vdash [L_p(?\phi)]\,\psi}\,([]L?) \qquad \frac{\vdash B_p(\phi) \wedge \psi}{\vdash \langle L_p(?\phi)\rangle\,\psi}\,(\langle\rangle L?)$$

Before we prove the soundness of the test rules, we will show that it is meaningless to allow doxastic modalities inside learned tests, and therefore that we may assume such a thing does not happen.

Proposition 3. *There is* $(\omega, \omega_1') \in \rho_\eta(L_p(?B_p(\phi)))$ *if and only if there is* $(\omega, \omega_2') \in \rho_\eta(?B_p(\phi))$*, and furthermore, if both exist,* $\omega_1' \sim \omega_2'$*, i.e.* ω_1' *and* ω_2' *are semantically equivalent. Equivalently for* $L_p(?P_p(\phi))$ *and* $?P_p(\phi)$*.*

Proof sketch. The semantics of the learning operator will execute the test at every possible world. But, since the formula being tested for is $B_p(\phi)$ (respectively $P_p(\phi)$), then its interpretation will be true only so long as every world (resp. at least one) passes the test ϕ, and false if at least one does not (resp. none do). Thus, either all worlds or no worlds will pass the test $?B_p(\phi)$ (resp. $?P_p(\phi)$), so there either is a single transition or there is no transition, as no world can be chosen as distinguished. This behavior is independent of the choice of distinguished world. In fact, it is the definition of $?B_p(\phi)$, which is therefore equivalent to $L_p(?B_p(\phi))$. □

One very interesting consequence of this is that programs and formulas inside a learning operator can only ever rely on the physical state and on the distinguished valuation for their interpretation.

Proposition 4. *Let ω_1 and ω_2 be PD-models, with $\mathrm{R}(\omega_1) = \mathrm{R}(\omega_2)$ and $\mathrm{DV}(\omega_1) = \mathrm{DV}(\omega_2)$. Then, for any program γ without doxastic modalities or learning operators,*

$$(\omega_1, \nu) \in \rho_\eta(\gamma) \text{ iff } (\omega_2, \nu) \in \rho_\eta(\gamma)$$

Proof sketch. By assumption, γ does not contain learning operators or doxastic modalities $B_p(\cdot)$ or $P_p(\cdot)$. These are actually the only constructs that are able to observe worlds other than the distinguished one, e.g. $u \in \omega$ such that $u \neq \mathrm{DW}(\omega)$. Therefore, any terms interpreted in γ are outside of those modalities, and thus are either state variables interpreted in $\mathrm{R}(\omega_1) = \mathrm{R}(\omega_2)$ or doxastic variables interpreted in $\mathrm{DV}(\omega_1) = \mathrm{DV}(\omega_2)$. □

These propositions simplify the soundness proof that follows.

Global soundness of $([]L?)$ and $(\langle\rangle L?)$. Let ω be a PD-model. Assume $\mathrm{val}_\eta(\omega, \Gamma_R; \Gamma_B; \Gamma_P; \Gamma_O)$. We must show $\mathrm{val}_\eta(\omega, [L_p(?\phi)]\psi)$, which is true iff for all ω' such that $(\omega, \omega') \in \rho_\eta(L_p(?\phi))$, $\mathrm{val}_\eta(\omega', \psi)$.

If there are no transitions the formula is trivially true, so assume there is $(\omega, \omega') \in \rho_\eta(L_p(?\phi))$. By the semantics of the learning operator

$$W(\omega') = \{t' : \text{ there is } t \in \omega \text{ s.t. } (\omega \oplus t, t') \in \rho_\eta(?\phi)\} \tag{3.6}$$
$$\omega'(t') = \mathrm{DV}(t') \text{ for all } t' \in \omega' \tag{3.7}$$
$$\mathrm{DW}(\mathrm{DW}(\omega')) = \mathrm{DW}(\omega) \tag{3.8}$$

Thus, for each $t' \in \omega'$, there is some $t \in \omega$ such that $(\omega \oplus t, t') \in \rho_\eta(?\phi)$, $t' = (\omega \oplus t)$, and $\omega'(t') = \mathrm{DV}(t')$. We know that test does not alter valuations, and therefore, $\omega'(t') = \mathrm{DV}(t') = \mathrm{DV}(\omega \oplus t) = \omega \oplus t(t) = \omega(t)$.

The formula ϕ, being inside a learning operator, cannot contain other learning operators or doxastic modalities. It follows from this and $\omega'(t') = \omega(t)$ that for all $t' \in \omega'$, $\mathrm{val}_\eta(\omega' \oplus t', \phi) = \mathrm{val}_\eta(\omega \oplus t, \phi) = \mathtt{T}$. We have thus established $\mathrm{val}_\eta(\omega', B_p(\phi))$.

To apply the hypothesis to finally show that $\mathrm{val}_\eta(\omega', \psi)$, we need only show that $\mathrm{val}_\eta(\omega', \Gamma_R; \Gamma_B; \Gamma_O)$. Since $\mathrm{val}_\eta(\omega', \Gamma_R; \Gamma_P; \Gamma_B; \Gamma_O)$ and $\mathrm{R}(\omega') = \mathrm{R}(\omega)$, $\mathrm{val}_\eta(\omega', \Gamma_R)$. Since $\omega \sqsubseteq \omega'$, by Proposition 1, $\mathrm{val}_\eta(\omega', \Gamma_B)$. Finally, since $\mathrm{DV}(\omega') = \mathrm{DV}(\omega)$, $\mathrm{val}_\eta(\omega', \Gamma_O)$.

The proof for $(\langle\rangle L?)$ differs only in that the hypothesis guarantees $\mathrm{val}_\eta(\omega, B_p(\phi))$, and thus that there is a transition for $L_p(?\phi)$. The rest of the proof follows the same lines as $[]L?$. □

Sequential Composition

Ideally, sequential composition within and without the learning operator behave similarly, and thus we can reduce "doxastic sequential composition" to dL sequential composition, as in the following proof rules.

$$\frac{[L_p(\alpha); L_p(\beta)]\phi}{[L_p(\alpha;\beta)]\phi}([]L;) \qquad \frac{\langle L_p(\alpha); L_p(\beta)\rangle\phi}{\langle L_p(\alpha;\beta)\rangle\phi}(\langle\rangle L;)$$

We are going to see that there is a very tight connection between between the worlds we arrive at after α but before executing β in $L_p(\alpha;\beta)$, and the worlds we arrive in after the execution of $L_p(\alpha)$, before executing $L_p(\beta)$.

This connection is the fundamental insight of the following proof.

Local soundness of ($[]L$;). Let ω be a PD-model. We must show that for all $(\omega,\omega'') \in \rho_\eta(L_p(\alpha;\beta))$, $\text{val}_\eta(\omega'',\phi)$. The first step is to examine the transitions for $L_p(\alpha;\beta)$ and $L_p(\alpha);L_p(\beta)$, as illustrated in Figure 3.14. Let us now begin with $(\omega,\omega'') \in \rho_\eta(L(\alpha;\beta))$.

$$\begin{aligned} W(\omega'') &= \{\nu'' : \text{ there is } t \in \omega.(\omega \oplus t, \nu'') \in \rho_\eta(\alpha;\beta)\} \\ &= \{\nu'' : \text{ there is } t \in \omega \text{ and } \mu.(\omega \oplus t, \mu) \in \rho_\eta(\alpha) \text{ and } (\mu,\nu'') \in \rho_\eta(\beta)\} \end{aligned}$$

We draw attention to the PD-models μ, which we will call the intermediate PD-models of ω''. For every $\nu'' \in \omega''$, there is a $t \in \omega$ and an intermediate PD-model μ such that $(\omega \oplus t, \mu) \in \rho_\eta(\alpha)$ and $(\mu,\nu'') \in \rho_\eta(\beta)$.

Let us now look at $(\omega,\omega_2'') \in \rho_\eta(L(\alpha);L(\beta))$. By the dL semantics of sequential composition, there exists ω_2' such that $(\omega,\omega_2') \in \rho_\eta(L_p(\alpha))$ and $(\omega_2',\omega_2'') \in \rho_\eta(L_p(\beta))$.

$$W(\omega_2') = \{\nu_2' : \text{ there is } t \in \omega.(\omega \oplus t, \nu_2') \in \rho_\eta(\alpha)\} \tag{3.9}$$

$$W(\omega_2'') = \{\nu_2'' : \text{ there is } \nu_2' \in \omega_2'.(\omega_2' \oplus \nu_2', \nu_2'') \in \rho_\eta(\beta)\} \tag{3.10}$$

We will now establish a semantic correspondence between the intermediate worlds μ of $L_p(\alpha;\beta)$ and the worlds $\nu_2' \in \omega_2'$. This will allow us to prove that ω'' and ω_2'' are semantically equivalent, and therefore that $\text{val}_\eta(\omega'',\phi) = \text{val}_\eta(\omega_2'',\phi) = \mathtt{T}$ by the rule's hypothesis.

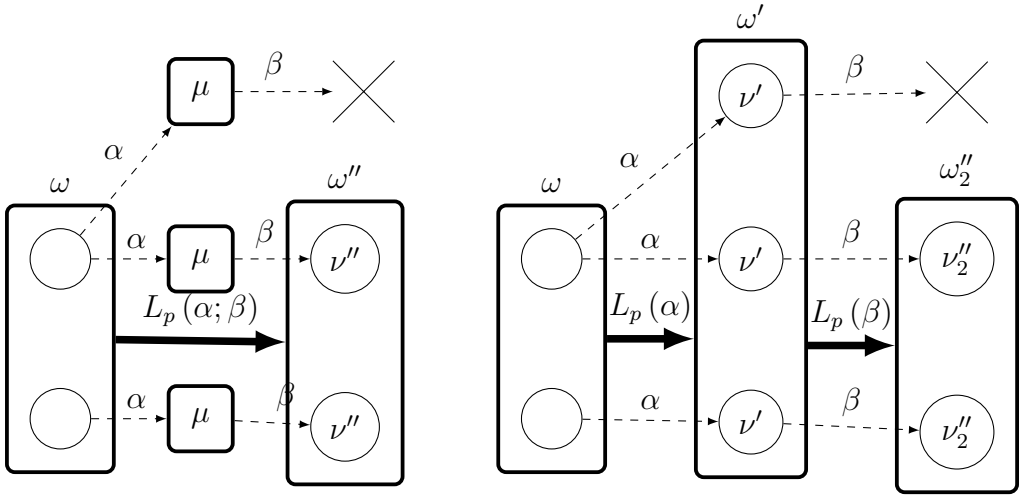


Figure 3.14: The transitions of $L_p(\alpha;\beta)$ and $L_p(\alpha);L_p(\beta)$

Claim 1: $\mathrm{R}(\omega'') = \mathrm{R}(\omega_2'')$. Trivial.

Claim 2: For every intermediate world μ of $\nu'' \in \omega''$, there is $\nu_2' \in \omega_2'$ such that $\omega_2'(\nu_2') = \text{DV}(\mu)$.

Let $\nu'' \in \omega''$. We already know there is $t \in \omega$ and μ such that $(\omega \oplus t, \mu) \in \rho_\eta(\alpha)$. But these are exactly the conditions of belonging to $W(\omega_2')$, and thus $\mu \in \omega_2'$. This is an equivalent statement to there being $\nu_2' \in \omega_2'$ such that $\mu = \nu_2'$. We will therefore use μ and ν_2' interchangeably from now on: $\omega_2'(\mu) = \omega_2'(\nu_2')$ and $\text{DV}(\nu_2') = \text{DV}(\mu)$.

Claim 3: ω_2'' subsumes ω'', or $\omega'' \sqsubseteq \omega_2''$.

For every $\nu'' \in \omega''$ we must show there is $\nu_2'' \in \omega_2''$ such that $\omega''(\nu'') = \omega_2''(\nu_2'')$. We already know there are transitions from ω using $t \in \omega$ and an intermediate PD-model μ.

But Claim 2 shows that there is $\nu_2' \in \omega_2'$ with exactly the same relevant valuations as μ: $\text{R}(\mu) = \text{R}(\nu_2') = \text{R}(\omega_2')$ and $\omega_2'(\nu_2') = \text{DV}(\nu_2') = \text{DV}(\mu)$. These are precisely those (and *only* those) that β can use in the transitions of $(\omega_2' \oplus \nu_2', \nu_2'') \in \rho_\eta(\beta)$.

Thus, for each transition $(\mu, \nu'') \in \rho_\eta(\beta)$, there is an equivalent transition $(\omega_2' \oplus \nu_2', \nu_2'') \in \rho_\eta(\beta)$ in where $\mu = \nu_2'$, $\text{DV}(\mu) = \text{DV}(\nu_2')$ and, more importantly, $\text{DV}(\nu'') = \text{DV}(\nu_2'')$. By the semantics of the learning operator, $\omega''(\nu'') = \omega_2''(\nu_2'')$.

Claim 4: ω'' subsumes ω_2'', or $\omega_2'' \sqsubseteq \omega''$

Claim 2 already established a correspondence between each μ and some world $\nu_2' \in \omega_2'$. However, there may be some $\nu_2' \in \omega_2'$ that does not correspond to any μ, since the existence μ requires a successful transition $(\mu, \nu'') \in \rho_\eta(\beta)$, with $\nu'' \in \omega''$. In contrast, there is no such restriction on $\nu_2' \in \omega_2'$.

But we know there exists some transition $(\omega_2' \oplus \nu_2', \nu_2'') \in \rho_\eta(\beta)$. Let us create μ such that $\mu = \nu_2'$ with $\text{DV}(\mu) = \text{DV}(\nu_2')$. Then, $(\omega \oplus \text{DW}(\nu_2'), \mu) \in \rho_\eta(\alpha)$ and furthermore, because $(\omega_2' \oplus \nu_2', \nu_2'') \in \rho_\eta(\beta)$, there must also be a transition $(\mu, \nu'') \in \rho_\eta(\beta)$. Therefore, $\nu'' \in \omega''$ and $\omega''(\nu'') = \omega_2''(\nu_2'')$.

Claims 3 and 4 together state $\omega'' \sim \omega_2''$, and we may conclude from the hypothesis that $\text{val}_\eta(\omega'', \phi)$.

The nondeterministic choice of distinguished world and its interaction with the box modality is left to the reader. □

Nondeterministic Choice

Nondeterministic choice is interesting because learned nondeterminism interacts with doxastic modalities, and the choice of distinguished world interacts with dynamic modalities. Thus, simply reducing $L_p(\alpha \cup \beta)$ to $L_p(\alpha) \cup L_p(\beta)$ is insufficient. If we removed the side condition from $\langle\rangle L\cup$, which prevents such a thing, it is not hard to find a counter-example:

$$\frac{\vdash \langle L_p(?\phi)\rangle B_p(\phi) \vee \langle L_p(?\top)\rangle B_p(\phi)}{\vdash \langle L_p(?\phi \cup ?\top)\rangle B_p(\phi)} \, (\langle\rangle L\cup)$$

Now let ω be a PD-model, $t \in \omega$ with $t \neq \text{DW}(\omega)$, $\text{val}_\eta(\omega, \phi)$, and $\text{val}_\eta(\omega \oplus t, \phi) = \texttt{F}$. Because $\text{val}_\eta(\omega, \phi)$, then $\text{val}_\eta(\omega, \langle L_p(?\phi)\rangle B_p(\phi))$, and since only worlds that satisfy ϕ exist after the test, we trivially satisfy the left-hand side of the hypothesis' disjunction.

Unless the conclusion is true as well, the rule is unsound. However, $L_p(?\phi \cup ?\texttt{T})$ aggregates all worlds that both tests, which due to $?\texttt{T}$ are all of them. Particularly, t will remain, and we know $\text{val}_\eta(\omega \oplus t, \phi) = \texttt{F}$. It will thus serve as a witness to the falsity of $B_p(\phi)$, and thus the rule is unsound.

The reason for this phenomenon is that although the disjunction is in line with the behavior of the $\diamond$ dynamic modality, it conflicts with the $\Box$ doxastic modality $B_p(\cdot)$. The logical operator used in these rules must satisfy the restrictions imposed by both types of modalities.

Thus, it can be seen in Figure 3.13 that it is only possible to apply disjunction when using both the diamond $\diamond$ dynamic modality and the diamond doxastic modality $P_p(\cdot)$.

We therefore have an understanding of the fundamentals of learned nondeterminism as it relates to doxastic modalities. However, the fact that $L_p(\alpha \cup \beta)$ cannot be reduced to $L_p(\alpha) \cup L_p(\beta)$ also speaks to the challenges of understanding what precisely can be learned from $[L_p(\alpha \cup \beta)]\phi$ for arbitrary ϕ, so that it may be stored in sequent contexts for use in the proof of ϕ.

On to the proof. First, let us define three sets of worlds which are not necessarily connected to any transitions.

$$W_\alpha = \{\nu_\alpha : \text{ there is } t \in \omega \text{ s.t. } (\omega \oplus t, \nu_\alpha) \in \rho_\eta(\alpha)\}$$
$$W_\beta = \{\nu_\beta : \text{ there is } t \in \omega \text{ s.t. } (\omega \oplus t, \nu_\beta) \in \rho_\eta(\beta)\}$$
$$W_\cup = \{\nu_\cup : \text{ there is } t \in \omega \text{ s.t. } (\omega \oplus t, \nu_\cup) \in \rho_\eta(\alpha \cup \beta)\} = W_\alpha \cup W_\beta$$

We make a distinction between the sets W_α and $W(\omega_\alpha)$ from $(\omega, \omega_\alpha) \in \rho_\eta(L_p(\alpha))$ because the latter does not exist if $L_p(\alpha)$ has no transitions.

Global soundness ($[]LB\cup$). The rule: Let ω be an arbitrary PD-model. We must show that $\text{val}_\eta(\omega, [L_p(\alpha \cup \beta)] B_p(\phi))$. We will split the proof into multiple claims that cover all possible cases.

Claim 1: If there are no transitions $(\omega, \nu_\alpha) \in \rho_\eta(\alpha)$ or $(\omega, \nu_\beta) \in \rho_\eta(\beta)$, the rule is sound.

If there are no such transitions, there can be no transition $(\omega, \omega_\cup) \in \rho_\eta(L_p(\alpha \cup \beta))$, and therefore the rule is trivially satisfied.

Claim 2: If there are transitions $(\omega, \nu_\alpha) \in \rho_\eta(\alpha)$ and $(\omega, \nu_\beta) \in \rho_\eta(\beta)$, the rule is sound.

If there is a transition $(\omega, \nu_\alpha) \in \rho_\eta(\alpha)$, then there must be $(\omega, \omega_\alpha) \in \rho_\eta(L_p(\alpha))$. Therefore, $W(\omega_\alpha) = W_\alpha$ and $W(\omega_\alpha) \subseteq W(\omega_\cup)$.

By the left conjunct of the hypothesis, $\text{val}_\eta(\omega_\alpha, B_p(\phi))$, and so for every $\nu_\alpha \in \omega_\alpha$, $\text{val}_\eta(\omega_\alpha \oplus \nu_\alpha, \phi)$. Using the same argument, $W(\omega_\beta) \subseteq W(\omega_\cup)$ and for every $\nu_\beta \in \omega_\beta$, $\text{val}_\eta(\omega_\alpha \oplus \nu_\alpha, \phi)$.

But $W(\omega_\cup) = W(\omega_\alpha) \cup W(\omega_\beta)$, and therefore for every $\nu_\cup \in \omega_\cup$, $\text{val}_\eta(\omega_\cup \oplus \nu_\cup, \phi)$. Thus, $\text{val}_\eta(\omega_\cup, B_p(\phi))$. This is exactly the statement for the conclusion of the proof rule.

The specific nondeterministic choice of the distinguished world for $\omega_\cup$ is irrelevant since $B_p(\phi)$ effectively overwrites it with every world of $\omega_\cup$.

Claim 3: If there is only one of the following two transitions, $(\omega, \nu_\alpha) \in \rho_\eta(\alpha)$ and $(\omega, \nu_\beta) \in \rho_\eta(\beta)$, the rule is sound.

Without loss of generality, let $(\omega, \nu_\alpha) \in \rho_\eta(\alpha)$ but not $(\omega, \nu_\beta) \in \rho_\eta(\beta)$.

If $W_\beta = \emptyset$, then β contributes no worlds to $W(\omega_\cup)$, thus $W_\cup = W_\alpha$. Because $(\omega, \nu_\alpha) \in \rho_\eta(\alpha)$, then $(\omega, \omega_\alpha) \in \rho_\eta(L_p(\alpha))$, and $\omega_\cup = \omega_\alpha$. Thus, the program in the conclusion of the rule is reduced to the program in the left conjunct of the hypothesis, $[L_p(\alpha)]\, B_p(\phi)$, and the rule is sound.

However, it is possible that $W_\beta \neq \emptyset$. If there are no transitions $(\omega, \omega_\beta) \in \rho_\eta(L_p(\beta))$, that only means the *distinguished world* failed to transition using β. There may be are other worlds $t \in \omega$ such that there are transitions $(\omega \oplus t, \nu_\beta) \in \rho_\eta(\beta)$, and thus $\nu_\beta \in \omega_\cup$. Thus, $W(\omega_\cup) = W(\omega_\alpha) \cup W_\beta$.

Claim 2 already shows that for all $\nu_\alpha \in \omega_\alpha$, $\text{val}_\eta(\omega_\cup \oplus \nu_\alpha, \phi)$. Unfortunately, we cannot use the hypothesis to show the same for $\nu_\beta \in \omega_\beta$, since $[L_p(\beta)]\, B_p(\phi)$ is trivially satisfied due to there being no transitions for $L_p(\beta)$.

Our notion of soundness helps here. We will construct a PD-model that is semantically equivalent to ω, whose choice of distinguished world allows a transition, and thus the hypothesis to be applied.

Since $W_\beta \neq \emptyset$, let $\mu \in W_\beta$ be one of those worlds which transitioned, and let $u = \text{DW}(\mu)$. Now consider the PD-model $\omega_\mu = \omega \oplus u$, which is equivalent to ω except we now know $(\omega \oplus u, \mu) \in \rho_\eta(\beta)$. Therefore, there is now $(\omega_\mu, \omega_\beta) \in \rho_\eta(L_p(\beta))$. We may thus conclude that for all $\nu_\beta \in \omega_\beta$, $\text{val}_\eta(\omega_\cup \oplus \nu_\beta, \phi)$.

Thus, since for all $\nu_\alpha \in \omega_\alpha$, $\text{val}_\eta(\omega_\cup \oplus \nu_\alpha, \phi)$, $\nu_\beta \in \omega_\beta$, $\text{val}_\eta(\omega_\cup \oplus \nu_\beta, \phi)$, and $W(\omega_\cup) = W(\omega_\alpha) \cup W(\omega_\beta)$, then for all $\nu_\cup \in \omega_\cup$, $\text{val}_\eta(\omega_\cup \oplus \nu_\cup, \phi)$.

Therefore, $\text{val}_\eta(\omega_\cup, B_p(\phi))$. □

3.4.3 Doxastic Assignment and Admissibility

Full generality doxastic assignment is one of the main challenges in this thesis, and the reason why deserves some attention. Does physical $\mathsf{d}\mathcal{L}$ assignment and its syntactic substitution proof rule directly apply to $\mathsf{d}^4\mathcal{L}$?

$$\frac{\vdash \phi(\theta)}{\vdash [x := \theta]\, \phi(x)}\ ([:=])$$

We investigate with the following rule instance depicting beliefs about an altitude increase.

$$\frac{\vdash B_p(alt_p < alt + 1)}{\vdash [alt := alt + 1]\, B_p(alt_p < alt)}\ ([:=])$$

Comparing perceived altitude to the current altitude after it has increased is equivalent to comparing it to an already-increase altitude. We do not allow doxastic expressions in physical assignments since it that is a meaningless, and potentially problematic, assignment. Despite the existence of doxastic expressions in these formulas, then, it still appears as though, in general, the $\mathsf{d}\mathcal{L}$ assignment rule is sound.

The big question is whether the substitution principle also applies to doxastic assignments and doxastic variables:

$$\frac{\vdash \phi(\theta)}{\vdash [L_p\,(x_p := \theta)]\,\phi(x_p)}\,([]L :=)$$

Admissibility

We will study the notion of admissibility according to the following insight.

"If you bind a free variable, you go to logic jail."
André Platzer, PhD

The substitution σ of x_p for term θ above, also denoted by $\sigma\,(x_p) = \theta$, is regulated by the notion of admissibility, since not all substitutions are semantically acceptable.

For instance, suppose we wanted to hardcode a change in our pilof's belief about altitude.

$$[alt_p := 2]\,[alt_p := 3]\,B_p\,(alt_p = 3)$$

It is easy to see that this formula ought to be true. Without care with admissibility, however, we run into trouble. Here's what happens when we carelessly apply our a substitution-based rule which replaces all occurrences of alt_p with 2.

$$\frac{\vdash [alt_p := 3]\,B_p\,(2 = 3)}{\vdash [alt_p := 2]\,[alt_p := 3]\,B_p\,(alt_p = 3)}\,([]L :=)$$

Clearly, the top formula is no longer true, which it ought to be. This issue arises because the occurrence of alt_p inside $B_p\,(alt_p = 3)$ refers to the *second* assignment, not the first one. To use the technical term, it is *bound* by the second assignment.

Sequential assignments can often be dealt with through α-renaming, e.g. using instead the program

$$\big[alt_{p_1} := 2\big]\,\big[alt_{p_2} := 3\big]\,B_p\,\big(alt_{p_2} = 3\big)$$

This way, it becomes clear that the first assignment does not refer to the variable inside the belief modality. However, there are many situations alpha-renaming does not ensure admissibility, such as when assignments happen within nondeterministic repetition.

We now formalize what we mean by admissibility.

Definition 21 (Admissibility)**.** *A substitution $\sigma\,(x) = \theta$ of state, logical or doxastic variable x is admissible for formula ϕ if no occurrence of x in ϕ appears within the scope of a binding quantifier or modality and no variable in the expression θ becomes bounded.*

Thus far, the definition does not differ from what we find in dL, though now we must address doxastic variables. Intuitively, the learning operator works on doxastic state, and therefore it will bind doxastic instead of physical variables.

Definition 22 (Bound variables)**.** *The following constructs bind variables.*

- *Quantifiers $\forall X$ and $\exists X$ bind logical variable X.*
- *$x := \theta$ and $x' = f(x)$ bind state variable x.*
- *$x_p := \theta$, inside or outside a learning operator, binds doxastic variable x_p.*
- *$?\phi$ inside a learning operator binds any doxastic variable that appears within ϕ.*

We note that doxastic assignment binds the variable both inside and outside the learning operator. This is interesting because doxastic variables, when not within the scope of doxastic modalities such as $B_p(\cdot)$ and $P_p(a)$, behave as state variables: there is only one possible interpretation for them, and it comes from the distinguished valuation $\text{DV}(\omega)$, much like the one interpretation of state variables comes from the physical state r.

Much more interesting and surprising, however, is that learned tests bind doxastic variables! This is a direct consequence of our fundamental principle of nondeterminism as doxastic indistinguishability. In hybrid programs, a test is a statement about the number of transitions a program has. Our learning operator, however, takes statements about transitions, and turns them into statements about possible worlds. Therefore, because the test within a learning operator effectively reduces the number of possible worlds, it affects the possible interpretations of the doxastic variables: it alters *doxastic state*.

Substitution lemma

The substitution lemmas are the pillars upon which full generality assignment rules stand. They state that syntactic substitution, as seen in the the assignment and doxastic assignment proof rules, is equivalent to semantic substitution.

The first lemma is directly copied from dL.

Lemma 1 (Substitution Lemma)**.** *Let σ be an admissible substitution for the formula ϕ, and let σ replace only logical or state variables. Then, for each η and $\omega = \langle r, W, V, s\rangle$*

$$val_\eta(\omega, \sigma(\phi)) = val_{\sigma(\eta)}(\langle \sigma(r), W, V, s\rangle, \phi)$$

where $\sigma(\eta)$ concurs with η except $\sigma(\eta)(X) = val_\eta(\omega, \sigma(X))$ for substituted variable logical X and $\sigma(r)$ concurs with r except $\sigma(r)(x) = val_\eta(\omega, \sigma(x))$ for substituted state variable x.

The lemma of interest to us is, in some sense, an application of the lemma for state and logical variables at each possible world $t \in \omega$.

Lemma 2 (Doxastic Substitution Lemma)**.** *Let ϕ be a formula. Let σ be an admissible substitution for ϕ which replaces only doxastic variables. Then, for every η and $\omega = \langle r, W, V, s\rangle$,*

$$val_\eta(\omega, \sigma(\phi)) = val_\eta(\sigma(\omega), \phi)$$

where $\sigma(\omega) = \langle r, W, \sigma(V), s\rangle$, and

- *$\sigma(\omega)(t)(x_p) = \omega(t)(x_p)$ for all non-substituted doxastic variables x_p and all $t \in \omega$*
- *$\sigma(\omega)(t)(x_p) = val_\eta(\omega \oplus t, \sigma(x_p))$ for all substituted doxastic variables x_p and all $t \in \omega$*

This lemma will be proven by structural induction on ϕ, meaning that we will need the lemma for terms, formulas and programs. The proof is currently incomplete, but for each sub-lemma below we will at least provide a case or two that illustrates how the structure of those proofs.

Lemma 3. *Let θ be a term. Let σ be an admissible substitution for ϕ which replaces only doxastic variables. Then, for every η, and $\omega = \langle r, W, V, s\rangle$,*

$$val_\eta(\omega, \sigma(\theta)) = val_\eta(\sigma(\omega), \theta)$$

Proof. By structural induction on θ. We will use an interpretation I that gives meaning to the arithmetic operators like addition and multiplication, and later to propositions such as $<$ and $=$.

- $\theta \equiv X$. For logical variable X, $\text{val}_\eta(\omega, \sigma(X)) = \text{val}_\eta(\omega, X) = \eta(X) = \text{val}_\eta(\sigma(\omega), X)$.
- $\theta \equiv x$. For state variable x, $\text{val}_\eta(\omega, \sigma(x)) = \text{val}_\eta(\omega, x) = r(x) = \text{val}_\eta(\sigma(\omega), x)$.
- $\theta \equiv x_p$. For $x_p \notin \sigma$, $\text{val}_\eta(\omega, \sigma(x_p)) = \text{val}_\eta(\omega, x_p) = \omega(s)(x_p) = \sigma(\omega)(s)(x_p) = \text{val}_\eta(\sigma(\omega), x_p)$
- $\theta \equiv x_p$. For $x_p \in \sigma$, $\text{val}_\eta(\omega, \sigma(x_p)) \overset{*}{=} \sigma(\omega)(s)(x_p) = \text{val}_\eta(\sigma(\omega), x_p)$. The annotated step is by definition of σ (c.f. Lemma 2) applied to s.
- $\theta \equiv f(\theta)$. For unary function f, $\text{val}_\eta(\omega, \sigma(f(\theta))) = \text{val}_\eta(\omega, f(\sigma(\theta))) = I(f)(\text{val}_\eta(\omega, \sigma(\theta))) \overset{\text{IH}}{=} I(f)(\text{val}_\eta(\sigma(\omega), \theta)) = \text{val}_\eta(\sigma(\omega), f(\theta))$.
  This proof is easily generalisable to functions of any arity, particularly those of first order real arithmetic such as addition, multiplication, etc.

□

Lemma 4. *Let ϕ be a formula. Let σ be an admissible substitution for ϕ which replaces only doxastic variables. Then, for every η, and $\omega = \langle r, W, V, s\rangle$,*

$$\mathit{val}_\eta(\omega, \sigma(\theta)) = \mathit{val}_\eta(\sigma(\omega), \theta)$$

Proof. By structural induction on ϕ.

- $p(\theta)$. For a unary proposition (e.g. $\theta = 0$), $\text{val}_\eta(\omega, \sigma(p(\theta))) = \text{val}_\eta(\omega, p(\sigma(\theta))) = I(p)(\text{val}_\eta(\omega, \sigma(\theta))) \overset{L.3}{=} I(p)(\text{val}_\eta(\sigma(\omega), \theta)) = \text{val}_\eta(\sigma(\omega), p(\theta))$. This proof is easily generalisable to other arities.
- $\neg\phi$. For negation, $\text{val}_\eta(\omega, \sigma(\neg\phi)) = \text{val}_\eta(\omega, \neg\sigma(\phi)) = 1 - \text{val}_\eta(\omega, \sigma(\phi)) \overset{\text{IH}}{=} 1 - \text{val}_\eta(\sigma(\omega), \phi)) = \text{val}_\eta(\sigma(\omega), \neg\phi))$
- $\phi_1 \wedge \phi_2$. For conjunction, $\text{val}_\eta(\omega, \sigma(\phi_1 \wedge \phi_2)) = \text{val}_\eta(\omega, \sigma(\phi_1) \wedge \sigma(\phi_2)) = \min(\text{val}_\eta(\omega, \sigma(\phi_1)), \text{val}_\eta(\omega, \min(\text{val}_\eta(\sigma(\omega), \phi_1)), \text{val}_\eta(\sigma(\omega), \phi_2))) = \text{val}_\eta(\sigma(\omega), \phi_1 \wedge \phi_2))$
- $\forall X.\phi$. For logical quantifiers, $\text{val}_\eta(\omega, \sigma(\forall X.\phi)) = \text{val}_\eta(\omega, \forall X.\sigma(\phi))$, which is only possible due to admissibility.
  $\text{val}_\eta(\omega, \forall X.\sigma(\phi)) = \mathtt{T}$ iff for all $v \in \mathbb{R}$, $\text{val}_{\eta_X^v}(\omega, \sigma(\phi)) = \mathtt{T}$ iff, by induction hypothesis applied for each v, for all $v \in \mathbb{R}$, $\text{val}_{\eta_X^v}(\sigma(\omega), \phi) = \mathtt{T}$, iff $\text{val}_\eta(\sigma(\omega), \forall X.\phi) = \mathtt{T}$. A couple of notes about this proof:
  - The induction hypothesis is applicable despite the change in variable assignment (η_X^v) because the I.H. includes universal quantification over η.
  - Admissibility plays a crucial role in what we may substitute here. If X appeared in $\sigma(x_p)$ for some x_p, then its value would be overriden by the quantifier, and we would not get the expected interpretation of $\sigma(x_p)$.
- $B(\phi)$. For the doxastic belief modality, $\text{val}_\eta(\omega, \sigma(B(\phi))) = \text{val}_\eta(\omega, B(\sigma(\phi))) = \mathtt{T}$ iff for all $t \in \omega$, $\text{val}_\eta(\omega \oplus t, \sigma(\phi)) = \mathtt{T}$ iff, by induction hypothesis for each t, for all $t \in \omega$, $\text{val}_\eta(\sigma(\omega) \oplus t, \phi) = \mathtt{T}$, iff $\text{val}_\eta(\sigma(\omega), B(\phi)) = \mathtt{T}$.

  - Once again, the induction hypothesis is applicable despite the change in distinguished world t because it includes universal quantification over ω.

- $[\alpha]\,\phi$. For the dynamic box modality,
  $\text{val}_\eta(\omega, \sigma([\alpha]\,\phi)) = \text{val}_\eta(\omega, [\sigma(\alpha)]\,\sigma(\phi))$ iff
  for all ω', if $(\omega, \omega') \in \rho_\eta(\sigma(\alpha))$ then $\text{val}_\eta(\omega', \sigma(\phi))$ iff by I.H. at each ω'
  for all ω', if $(\omega, \omega') \in \rho_\eta(\sigma(\alpha))$ then $\text{val}_\eta(\sigma(\omega'), \phi)$ iff by Lemma 5
  for all ν, if $(\sigma(\omega), \nu) \in \rho_\eta(\alpha)$ then $\text{val}_\eta(\nu, \phi)$ iff
  $\text{val}_\eta(\sigma(\omega), [\alpha]\,\phi)$ for all ω', if $(\sigma(\omega), \sigma(\omega')) \in \rho_\eta(\alpha)$ then $\text{val}_\eta(\sigma(\omega'), \phi)$ iff
  $\text{val}_\eta(\sigma(\omega), [\alpha]\,\phi)$

□

Now of course, we need the lemma for programs, which is where the learning operator will show up.

Lemma 5. *Let α be a program. Let σ be an admissible substitution for α which replaces only doxastic variables. Then, for every η, ω,*

$$(\omega, \omega') \in \rho_\eta(\sigma(\alpha)) \textit{ iff } (\sigma(\omega), \sigma(\omega')) \in \rho_\eta(\alpha)$$

The statement of Lemma 5 is more explicitly written as the following two implications.

1. If $(\omega, \omega') \in \rho_\eta(\sigma(\alpha))$ then $(\sigma(\omega), \sigma(\omega')) \in \rho_\eta(\alpha)$
2. If $(\sigma(\omega), \nu) \in \rho_\eta(\alpha)$ then there is ω' such that $(\omega, \omega') \in \rho_\eta(\sigma(\alpha))$ and $\nu = \sigma(\omega')$.

Proof. By structural induction on α.

- $y := \theta$. For assignment,
  1. Let $(\omega, \omega') \in \rho_\eta(y := \sigma(\theta))$. We must show $(\sigma(\omega), \sigma(\omega')) \in \rho_\eta(\alpha)$.
     By the semantics of assignment,
     $\omega' = \langle r_y^{\text{val}_\eta(\omega, \sigma(\theta))}, W, V, s\rangle$, which, by Lemma 3 for expressions,
     $\omega' = \langle r_y^{\text{val}_\eta(\sigma(\omega), \theta)}, W, V, s\rangle$.
     From the above, we get that $\sigma(\omega') = \langle r_y^{\text{val}_\eta(\sigma(\omega), \theta)}, \sigma(W), V, s\rangle$.
     It now also easy to see that the transition $(\sigma(\omega), \sigma(\omega')) \in \rho_\eta(\alpha)$, with $\sigma(\omega)$ and $\sigma(\omega')$ as defined above, concur with the semantics of assignment, thus concluding the proof.
  2. Let $\omega = \langle r, W, V, s\rangle$, $\sigma(\omega) = \langle r, \sigma(W), V, s\rangle$, and $(\sigma(\omega), \nu) \in \rho_\eta(y := \theta)$. We must show there is ω' such that $(\omega, \omega') \in \rho_\eta(y := \sigma(\theta))$ and $\nu = \sigma(\omega')$.
     By the semantics of assignment, $\nu = \langle r_y^{\text{val}_\eta(\sigma(\omega), \theta)}, \sigma(W), V, s\rangle$.
     Now consider $(\omega, \omega') \in \rho_\eta(\sigma(\alpha))$. Then, again by the semantics of assignment,
     $\omega' = \langle r_y^{\text{val}_\eta(\omega, \sigma(\theta))}, W, V, s\rangle$, which, by Lemma 3 for expressions,
     $\omega' = \langle r_y^{\text{val}_\eta(\sigma(\omega), \theta)}, W, V, s\rangle$.
     To conclude, we need only observe that $\sigma(\omega') = \langle r_y^{\text{val}_\eta(\sigma(\omega), \theta)}, \sigma(W), V, s\rangle = \nu$.
- $\alpha \cup \beta$. For nondeterministic choice,

1. Let $(\omega, \omega') \in \rho_\eta(\sigma(\alpha) \cup \sigma(\beta))$. We must show $(\sigma(\omega), \sigma(\omega')) \in \rho_\eta(\alpha \cup \beta)$.
   Either $(\omega, \omega') \in \rho_\eta(\sigma(\alpha))$ or $(\omega, \omega') \in \rho_\eta(\sigma(\beta))$.
   Without loss of generality, assume $(\omega, \omega') \in \rho_\eta(\sigma(\alpha))$. By I.H. $(\sigma(\omega), \sigma(\omega')) \in \rho_\eta(\alpha)$.
   Therefore, $(\sigma(\omega), \sigma(\omega')) \in \rho_\eta(\alpha \cup \beta)$.
2. Let $(\sigma(\omega), \nu) \in \rho_\eta(\alpha \cup \beta)$. We must show there is ω' such that $(\omega, \omega') \in \rho_\eta(\sigma(\alpha) \cup \sigma(\beta))$ and $\nu = \sigma(\omega')$.
   Either $(\sigma(\omega), \nu) \in \rho_\eta(\alpha)$ or $(\sigma(\omega), \nu) \in \rho_\eta(\beta)$. Without loss of generality, assume $(\sigma(\omega), \nu) \in \rho_\eta(\alpha)$. Then by I.H., there is ω' such that $(\omega, \omega') \in \rho_\eta(\sigma(\alpha))$, and $\nu = \sigma(\omega')$.
   But then, $(\omega, \omega') \in \rho_\eta(\sigma(\alpha) \cup \sigma(\beta))$, concluding the proof.

- $\alpha; \beta$. For sequential composition,
  1. Let $(\omega, \omega'') \in \rho_\eta(\sigma(\alpha); \sigma(\beta))$. We must show $(\sigma(\omega), \sigma(\omega'')) \in \rho_\eta(\alpha; \beta)$.
     By the semantics of sequential composition, there is ω' such that $(\omega, \omega') \in \rho_\eta(\sigma(\alpha))$ and $(\omega', \omega'') \in \rho_\eta(\sigma(\beta))$. Applying the i.h. twice, there is ω', and therefore $\sigma(\omega')$, such that $(\sigma(\omega), \sigma(\omega')) \in \rho_\eta(\alpha)$ and $(\sigma(\omega'), \sigma(\omega'')) \in \rho_\eta(\beta)$. Thus, $(\sigma(\omega), \sigma(\omega'')) \in \rho_\eta(\alpha; \beta)$.
  2. Let $(\sigma(\omega), \nu'') \in \rho_\eta(\alpha; \beta)$. We must show there is ω'' such that $(\omega, \omega'') \in \rho_\eta(\sigma(\alpha); \sigma(\beta))$ and $\nu'' = \sigma(\omega'')$.
     By the semantics of sequential composition, there is ν' such that $(\sigma(\omega), \nu') \in \rho_\eta(\alpha)$ and $(\nu', \nu'') \in \rho_\eta(\beta)$. By i.h., there is ω' such that $(\omega, \omega') \in \rho_\eta(\sigma(\alpha))$ and $\nu' = \sigma(\omega')$.
     Because $\nu' = \sigma(\omega')$, we now also have $(\sigma(\omega'), \nu'') \in \rho_\eta(\beta)$, so that we may apply the i.h. again. Thus, there is ω'' such that $(\omega', \omega'') \in \rho_\eta(\sigma(\beta))$, and $\nu'' = \sigma(\omega'')$.
     Because $(\omega, \omega') \in \rho_\eta(\sigma(\alpha))$ and $(\omega', \omega'') \in \rho_\eta(\sigma(\beta))$, by the semantics of sequential composition, $(\omega, \omega'') \in \rho_\eta(\sigma(\alpha); \sigma(\beta))$.
- $L(\alpha)$. For the learning operator, we use an auxiliary lemma.

□

Lemma 6. *Let α be a program. Let σ be an admissible substitution for α which replaces only doxastic variables. Then, for every η, ω,*

$$(\omega, \omega') \in \rho_\eta(L_p(\sigma(\alpha))) \text{ iff } (\sigma(\omega), \sigma(\omega')) \in \rho_\eta(L_p(\alpha))$$

As before (c.f. Lemma 5), we will be proving the two more technically precise statements that follow.

1. If $(\omega, \omega') \in \rho_\eta(L_p(\sigma(\alpha)))$ then $(\sigma(\omega), \sigma(\omega')) \in \rho_\eta(L_p(\alpha))$
2. If $(\sigma(\omega), \nu) \in \rho_\eta(L_p(\alpha))$ then there is ω' such that $(\omega, \omega') \in \rho_\eta(L_p(\sigma(\alpha)))$ and $\nu = \sigma(\omega')$.

There are some proof sketches and incomplete proofs for previous versions of the semantics, but they still need to be adapted to the semantics presented here.

Once this is proven, we will be able to use substitution for doxastic variables, and thus use the learned doxastic assignment rule as presented above.

Chapter 4

Proposal

Here, we propose a set of tasks and loose timeline for the completion of this thesis.

1. Proving a substitution lemma **1 month**
2. Generalizing the sequent calculus **1.5 months**
3. Case studies, concurrent with generalization **1.5 months**

Thus, we expect the completion of this thesis to take around **4 months**.

4.1 Phase 1: Substitution Lemma

A substitution lemma will be one of the main contributions of this thesis. Without it, we cannot have full generality doxastic assignment.

A proof for this lemma has been extremely close to existing for the many iterations of our semantics. The relative simplicity of the latest semantics, along with experience proving the various iterations of this lemma, will make this an attainable first step in ensuring the validity of our logic.

One important detail that previously went unnoticed was that tests within learning operators bind doxastic variables. This will affect the applicability of the substitution lemma, and that restriction may have significant consequences in what we may be able to prove with the logic. It will thus be important to investigate how test becoming a binding operator affects the calculus, and how we may alleviate such restrictions with more proof rules.

Our experience tells us that the proof rule of the substitution lemma is extensive proof and contains many corner cases that are easy to miss, so we allow a larger timeframe to ensure our proof is, in fact, a proof!

Timeframe: 1 month.

4.2 Phase 2: Generalizing the sequent calculus

Once we have a deeper understanding of soundness for our basic, specific rules, we will be better prepared to tackle full-generality rules, if they are possible at all.

In this phase, we will investigate whether our rules can be generalized, and discover why or why not.

It is crucial that they be as general as possible, lest we be unable to apply the calculus to any real-world scenario.

Timeframe: 1.5 months.

4.3 Phase 3: Case Study

It is important to continually test the practical applicability of our sequent calculus. To this end, we hope to develop several case studies using the examples in this proposal as a stepping stone.

By case study, we mean the following:

1. A study of the documentation related to several safety incidents of real world cyber-physical systems. We will focus on incidents that contain doxastic phenomena.
2. Modeling the incident through one or multiple $\mathsf{d}^4\mathcal{L}$ formulas, formalizing it and providing insight into its distinct moving parts. Successfully doing so is proof that $\mathsf{d}^4\mathcal{L}$ is expressive enough to capture relevant real scenarios.
3. Using the calculus to obtain 1) a proof of validity of those formulas, i.e. a mathematical guarantee of safety for the system, or 2) indications of counter-examples to the formulas, i.e. identifying situations where safety cannot be guaranteed.
4. Revision of non-valid formulas with different belief-triggered controllers or slightly modified CPS designs. This process should continue until some combination of controller and CPS design becomes valid, and thus, safe.

It is useful to highlight the difference between the two types of proofs we have mentioned thus far.

In this case, the deliverable of a case study is a sequent proof: a sound logical argument for why the chosen belief-triggered controllers keep the modeled CPS safe. Regulation authorities can then use them to create law, policy, checklists and training programs to increase the safety of CPS.

It has been our experience with preliminary case study proofs that:

- We identify proof patterns that are used very often. This gives us insight into new proof rules, which while not strictly necessary, provide helpful shortcuts and intuitions.
- We find that there are further proof rules, at the intersection of belief and hybrid dynamics, without which certain case studies cannot be proven. These may result in new syntax, semantics and proof rules being added.

Phase 3 will thus be running concurrently with Phase 2 as a feedback loop, and may unexpectedly extend the timeline of Phase 2. The rules available inform case study proofs, which in turn inform what proof rules are needed.

Timeframe: running concurrently with Phase 2, plus a dedicated 1.5 months.

4.3.1 PAPI Stepping Stone Case Study

A more robust case study of the PAPI light scenario will serve a first step ensuring that $\mathsf{d}^4\mathcal{L}$ and its calculus can indeed meet the modeling and proving challenges of safe CPS design.

To write down the case study more comprehensively, we will a more fleshed out model that approximates the dynamics of flight in the vertical plane. Such model may start simple and become gradually more complicated as we gain experience and confidence with this logic.

We expect to also gradually improve ours understanding of what a good belief-triggered controller should look like. The answer to this is precisely objective of this (or any!) case study, and indeed, it is what the CPS safety engineers of the future will be tasked to address.

A first attempt at a very granular belief-triggered controller may look as follows.

Example 1 (Granular belief-triggered control)**.**

$$
\begin{aligned}
&?\left(\neg B_p\left(\mathtt{l1}_p = G \wedge \mathtt{l2}_p = G\right)\right); \mathit{yinput} := 1 \cup \\
&?\left(B_p\left(\mathtt{l1}_p = G \wedge \mathtt{l2}_p = G\right) \wedge \neg P_p\left(\mathtt{l4}_p = G\right)\right); \mathit{yinput} := 0 \cup \\
&?\left(B_p\left(\mathtt{l1}_p = G \wedge \mathtt{l2}_p = G\right) \wedge P_p\left(\mathtt{l4}_p = G\right)\right); \mathit{yinput} := -0.5 \cup \\
&?\left(B_p\left(\mathtt{l1}_p = G \wedge \mathtt{l2}_p = G\right) \wedge B_p\left(\mathtt{l4}_p = G\right)\right); \mathit{yinput} := -1
\end{aligned}
$$

We'll say that the pilot believes the airplane is safe if it is on the glide path or above, i.e. $B_p\left(\mathtt{l1}_p = G \wedge \mathtt{l2}_p = G\right)$. Then,

1. If the pilot does not believe that the airplane is safe, they will climb.
2. If the pilot believes it is safe, and but doesn't think it's possible for the aircraft to be "very high", the glide path remains unchanged.
3. If the pilot believes it is safe, and it is possible that the aircraft is "very high", they will begin a *cautious* descent.
4. If the pilot is certain that the aircraft is "very high", they will choose an assertive descent.

It is exciting to see the level of granularity achievable with just two doxastic modalities. Even this initial example showcases the extent to which we may now model this phenomenon.

4.3.2 Touch-and-go flap failure

A second, previously mentioned case study[1] highlights different aspects of belief. Recall that in this case, a student pilot is practicing touch-and-go maneuvers when the aircraft suffers an *unobserved* flap failure. Thus, we will explore the phenomenon where doxastic and physical state are entirely inconsistent with one another, and integrate slightly more complex flight dynamics as required by the flaps' effect on lift.

At some point, the pilot almost unconsciously felt that *something* was wrong, because the altitude lower than what their gut told them it should be. This triggered a series of sensorial actions to determine the cause of the perceived symptoms. Thus, the greatest doxastic challenge will be to discover and develop $\mathsf{d}^4\mathcal{L}$ patterns that emulate such human intuition, and learn how to encode them into a belief-triggered controller.

[1] `http://airfactsjournal.com/2014/03/flaps-anyone-strange-things-can-happen/`

We suspect a good starting point will be a threshold definition of human intuition: whenever the real value and believed value are too far, e.g. ϵ distant, from one another, this would trigger sensorial actions on the part of the controller.

$$?B_p\left(alt^2 - alt_p^2 > \epsilon^2\right); \texttt{obs}$$

The result of this case study will be a belief-triggered controller that is resilient to unobserved flap failures during landings. By using this general approach for different types of failures (e.g. engine, different control surfaces, etc), CPS designers can build a good understanding of what sensory actions work best under which circumstances, e.g. low altitude. Equipped with that knowledge, designers will be able to develop efficient controllers for more general failure models, first triggering the sensory actions that are most likely find the cause for the observed symptoms.

4.3.3 Air France 447

The Air France 447 disaster is a more complex case study, and to a large extent the motivation for this entire thesis. It would be very exciting to provide a comprehensive case study, but such an endeavor goes beyond the scope, and timeline, of this thesis. Instead, if time allows, we will attempt to break the complexities of this case study down into smaller component parts, which should be more tractable in isolation.

To do so, we are likely to still will need the ability to reason about different agents. More specifically, we must encode different belief states and actions for both the pilot and co-pilot of an Airbus 330. As mentioned early, this should not be a significant technical challenge since beliefs can only be held about the state of the world, and not about other agents' beliefs.

The Airbus 330 allows multiple modes of control, which include full control to the pilot, full control to the copilot, and combined inputs from both pilot and copilot. This contributed to mode confusion, and was undoubtedly a factor in the ultimate demise of the airplane. This project will thus require a careful examination not only of pilot belief dynamics, but also of how the hardware itself is designed.

By exploring several smaller "what-if" scenarios, we will attempt to determine whether a change in cockpit design, policy or checklists may have effectively addressed the situation that led to the crash. We thus hope to show that $\mathsf{d}^4\mathcal{L}$ should become one more tool in the arsenal used to prevent, and determine the cause of, safety violations.

4.4 Phase 4: Write-up

All theses need some time dedicated to write-up and proof-reading once all the desired results have been achieved ☺

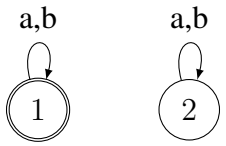
Timeframe: 4-6 weeks

Appendix A

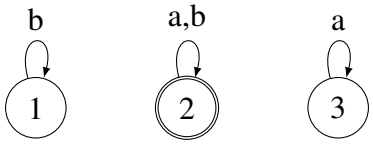
Brief on multi-agent logic

Multi-agent doxastic logics add an extra layer of complexity to an already complex logic, as the *de facto* standard is for agents in those logics to be able to reason about each other's beliefs.

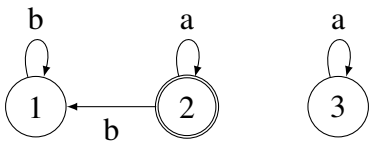
The semantics for these logics are often built on top of Kripke structures, which use a "flat" graph structure to represent belief, thus intricately linking different agents' beliefs. However, one of the core needs of $\mathsf{d}^4\mathcal{L}$ is that one agent learning something new should not affect another's beliefs, requiring at least some decoupling between agents beliefs. The use of Kripke structures together with the need for belief decoupling is one of the greatest challenges, of even impossibilities. Consider this incredibly simple Kripke structure with two agents.



In this structure, both a and b know that $x = 1$, and can distinguish that from the possibility that $x = 2$. Now suppose that we want agent a to learn that x_p was incremented, i.e. $L_a\left(x_a := x_a + 1\right) \equiv L_a\left(x_a{+}{+}\right)$. We would now need to find a Kripke structure in which the beliefs of agent a have been updated, but b's remain unchanged. Let's give it a few tries.



Now, b continues to distinguish between 1 and 2, a distinguished between 2 and 3, as expected, but we may only choose a single distinguished world. If we pick 2, then b is going to believe that $x = 2$, which is a change from previously believing $x = 1$. Only a learnt something though, so b's beliefs changed from $x = 1$ to $x = 1$. Conversely, if the distinguished world was 1 to satisfy the need for b's beliefs not to change, then a would not have learnt that $x = 2$, as per the program. *Next try!*



Now b believes that $x = 1$, and a believes that $x = 2$, as expected! Unfortunately, b can no longer consider $x = 2$ since we had to remove the loop on that world. Most importantly, however, the Kripke structure is no longer symmetric or reflexive, so that it loses many of the properties that make it into a doxastic logic.

In fact, we would argue that there is no easy way to update an agent's a's beliefs while leaving b's entirely unchanged. There is also the additional challenge, heretofore unmentioned, that if a learns something new, then a's beliefs about b's beliefs should remain unchanged as well.

We believe (pun intended!) that a multi-agent extension to the logic proposed here would have to rely on entirely different semantic structures, and while we have put some thought into this, no ideas have been forthcoming.

Bibliography

[1] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *J. Symb. Log.*, 50(2):510–530, 1985. doi: 10.2307/2274239. 1.1.2, 2.2, 2.3.1

[2] Alexandru Baltag and Lawrence S. Moss. Logics for epistemic programs. *Synthese*, 139 (2):165–224, 2004. doi: 10.1023/B:SYNT.0000024912.56773.5e. 2.3

[3] Alexandru Baltag, Lawrence S. Moss, and Slawomir Solecki. The logic of public announcements and common knowledge and private suspicions. In *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-98), Evanston, IL, USA, July 22-24, 1998*, pages 43–56, 1998. 2.3, 2.3.1, 2.3.3

[4] Giacomo Bonanno. Axiomatic characterization of the AGM theory of belief revision in a temporal logic. *Artif. Intell.*, 171(2-3):144–160, 2007. doi: 10.1016/j.artint.2006.12.001. URL `https://doi.org/10.1016/j.artint.2006.12.001`. 2.2, 2.2

[5] Bureau d'Enquêtes et d'Analyses (BEA). Final report on the accident on 1st june 2009 to the airbus A330-203 registered F-GZCP operated by Air France flight AF 447 from Rio de Janeiro to Paris. 2012. 1.1.1, 3.1.1

[6] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic Epistemic Logic*. Springer Publishing Company, Incorporated, 1st edition, 2007. ISBN 1402058381, 9781402058387. 1.1.2, 2.3, 2.3.1

[7] Nir Friedman and Joseph Y. Halpern. Belief revision: A critique. *CoRR*, cs.AI/0103020, 2001. 2.2

[8] Jelle Gerbrandy. Bisimulations on planet kripke, ph.d thesis. 01 1999. 2.3

[9] Jelle Gerbrandy and Willem Groeneveld. Reasoning about information change. *Journal of Logic, Language and Information*, 6(2):147–169, 1997. doi: 10.1023/A:1008222603071. 2.3, 2.3

[10] Jean-Baptiste Jeannin, Khalil Ghorbal, Yanni Kouskoulas, Aurora Schmidt, Ryan Gardner, Stefan Mitsch, and André Platzer. A formally verified hybrid system for safe advisories in the next-generation airborne collision avoidance system. *STTT*, 19(6):717–741, 2017. doi: 10.1007/s10009-016-0434-1. URL `http://www.cs.cmu.edu/~aplatzer/pub/acasx-zones-long.pdf`. 1.1.2

[11] Yanni Kouskoulas, David W. Renshaw, André Platzer, and Peter Kazanzides. Certifying the safe design of a virtual fixture control algorithm for a surgical robot. In Calin Belta

and Franjo Ivancic, editors, *Hybrid Systems: Computation and Control (part of CPS Week 2013), HSCC'13, Philadelphia, PA, USA, April 8-13, 2013*, pages 263–272. ACM, 2013. doi: 10.1145/2461328.2461369. 1.1.2

[12] Sarah M. Loos, David Witmer, Peter Steenkiste, and André Platzer. Efficiency analysis of formally verified adaptive cruise controllers. In Andreas Hegyi and Bart De Schutter, editors, *ITSC*, pages 1565–1570, 2013. ISBN 978-1-4799-2914-613. doi: 10.1109/ITSC.2013.6728453. 1.1.2

[13] David Makinson. On the status of the postulate of recovery in the logic of theory change. *J. Philosophical Logic*, 16(4):383–394, 1987. doi: 10.1007/BF00431184. URL `https://doi.org/10.1007/BF00431184`. 2.2

[14] Stefan Mitsch, Khalil Ghorbal, David Vogelbacher, and André Platzer. Formal verification of obstacle avoidance and navigation of ground robots. *I. J. Robotics Res.*, 36(12):1312–1340, 2017. doi: 10.1177/0278364917733549. 1.1.2

[15] André Platzer. Differential dynamic logic for hybrid systems. *J. Autom. Reas.*, 41(2):143–189, 2008. ISSN 0168-7433. doi: 10.1007/s10817-008-9103-8. 2.1, 2.1.3

[16] André Platzer. Logics of dynamical systems. In *LICS*, pages 13–24. IEEE, 2012. ISBN 978-1-4673-2263-8. doi: 10.1109/LICS.2012.13. 1.1.2, 2.1, 2.1.3

[17] André Platzer. Logic & proofs for cyber-physical systems. In Nicola Olivetti and Ashish Tiwari, editors, *IJCAR*, volume 9706 of *LNCS*, pages 15–21. Springer, 2016. doi: 10.1007/978-3-319-40229-1_3. 2.1, 2.1.3

[18] Mark Strauss and James Carnahan. Distance estimation error in a roadway setting. *The Police Journal*, 82(3):247–264, 2009. doi: 10.1350/pojo.2009.82.3.458. URL `https://doi.org/10.1350/pojo.2009.82.3.458`. 1.1, 3.1.3

[19] Johan van Benthem. Dynamic logic for belief revision. *Journal of Applied Non-Classical Logics*, 17(2):129–155, 2007. 2.2

[20] Hans P. van Ditmarsch. Descriptions of game actions. *Journal of Logic, Language and Information*, 11(3):349–365, 2002. doi: 10.1023/A:1015590229647. 2.3, 2.3.2

[21] Hans P. van Ditmarsch, Wiebe van der Hoek, and Barteld P. Kooi. Dynamic epistemic logic with assignment. In *4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), July 25-29, 2005, Utrecht, The Netherlands*, pages 141–148, 2005. doi: 10.1145/1082473.1082495. 2.3, 3.1.4