

Chapter 3

Reasoning about Access Control

In the previous chapter, we introduced a language for describing access-control scenarios. We presented the syntax for well-formed statements in the language (i.e., formulas). We also specified their semantics through the use of Kripke structures.

Although Kripke structures provide precise meanings for these statements, it is not practical to analyze access-control situations at such a low level. First of all, Kripke structures are cumbersome to use for even simple situations. Second, in any given situation, it is unclear which structures accurately capture the state of the universe. How could we possibly know which Kripke structure to use for any given analysis?

In this chapter, we introduce a collection of *inference rules* that will provide the basis for reasoning rigorously about access control. These rules describe a system for manipulating well-formed formulas as a way of calculating the consequences of various assumptions. A crucial property for these rules is that they must be *sound* with respect to the Kripke-structure semantics. That is, the rules should ensure that, in any situation where a given Kripke structure satisfies all of a rule's premises, the Kripke structure also satisfies the rule's consequent. Informally, soundness means that it is impossible to deduce a “false” statement from “true” ones. In this chapter, we demonstrate how to verify that the rules we introduce are sound.

3.1 Logical Rules

The logical rules are summarized in Figure 3.1. Each rule of the logic has the form

$$\frac{H_1 \quad \cdots \quad H_k}{C},$$

where the items written above the line (e.g., H_1, \dots, H_k) correspond to *hypotheses* (or *premises*) and the item below the line (e.g., C) corresponds to a *consequence* (or *conclusion*). A special case occurs when there are no hypotheses on the top of the rule (i.e., when $k = 0$): an inference rule with an “empty top” is called an *axiom*.

Informally, we often read such rules as “If all the assertions on the top are true, then the consequence below the line will also be true.” More accurately, however, the logical rules describe a system for manipulating well-formed formulas of the logic. In fact, one can think of the logical rules as defining a game, in which a player writes

FIGURE 3.1 Logical rules for the access-control logic

<i>Taut</i>	$\frac{}{\varphi}$	if φ is an instance of a prop-logic tautology
<i>Modus Ponens</i>	$\frac{\varphi \quad \varphi \supset \varphi'}{\varphi'}$	<i>Says</i> $\frac{\varphi}{P \text{ says } \varphi}$
<i>MP Says</i>	$\frac{}{(P \text{ says } (\varphi \supset \varphi')) \supset (P \text{ says } \varphi \supset P \text{ says } \varphi')}$	
<i>Speaks For</i>	$\frac{}{P \Rightarrow Q \supset (P \text{ says } \varphi \supset Q \text{ says } \varphi)}$	
<i>& Says</i>	$\frac{}{(P \ \& \ Q \text{ says } \varphi) \equiv ((P \text{ says } \varphi) \wedge (Q \text{ says } \varphi))}$	
<i>Quoting</i>	$\frac{}{(P \mid Q \text{ says } \varphi) \equiv (P \text{ says } Q \text{ says } \varphi)}$	
	<i>Idempotency of \Rightarrow</i> $\frac{}{P \Rightarrow P}$	
<i>Transitivity of \Rightarrow</i>	$\frac{P \Rightarrow Q \quad Q \Rightarrow R}{P \Rightarrow R}$	<i>Monotonicity of \Rightarrow</i> $\frac{P \Rightarrow P' \quad Q \Rightarrow Q'}{P \mid Q \Rightarrow P' \mid Q'}$
<i>Equivalence</i>	$\frac{\varphi_1 \equiv \varphi_2 \quad \Psi[\varphi_1/q]}{\Psi[\varphi_2/q]}$	
	$P \text{ controls } \varphi \stackrel{\text{def}}{=} (P \text{ says } \varphi) \supset \varphi$	

FIGURE 3.2 Common propositional-logic tautologies

$p \vee \neg p$	$p \supset (q \supset (p \wedge q))$
$p \equiv (\neg \neg p)$	$(p \wedge q) \supset (p \supset q)$
$p \supset (q \vee p)$	$(p \wedge q) \supset (q \wedge p)$
$p \supset (q \supset p)$	$(p \equiv q) \supset (p \supset q)$
$(p \wedge q) \supset p$	$((p \vee q) \wedge \neg p) \supset q$
$\neg(\neg p \wedge p)$	$((p \supset q) \wedge (q \supset r)) \supset (p \supset r)$

(i.e., *derives*) various formulas on a piece of paper. Each rule states that, if all the premises of an inference rule have already been written down (derived), then the conclusion can also be written down (derived). Axioms can always be written down.

We return to this notion of derivations in Section 3.2, where we introduce *formal proofs*. For now, however, we discuss each of the logical rules in turn.

3.1.1 The *Taut* Rule

The simplest rule is the axiom *Taut*:

$$\text{\textit{Taut}} \quad \frac{}{\varphi} \quad \text{if } \varphi \text{ is an instance of a prop-logic tautology}$$

This axiom states that any instance of a *tautology* from propositional logic can be introduced at any time as a derivable statement in the access-control logic. To understand what this rule means, first recall that a propositional-logic tautology is a formula that evaluates to *true* under *all* possible interpretations of its propositional variables. For example, the propositional formula $p \vee \neg p$ always evaluates to *true*, independent of whether the propositional variable p is assigned the value *true* or the value *false*. In contrast, the formula $p \supset \neg p$ is not a tautology, because it evaluates to *false* whenever p is assigned the value *true*. Although it does not constitute a complete listing, Figure 3.2 summarizes some common propositional-logic tautologies.

A formula φ is an *instance* of the formula ψ if there exist propositional variables p_1, \dots, p_k (for some $k \geq 0$) and modal formulas ϕ_1, \dots, ϕ_k such that φ is obtained by replacing each p_i in ψ by ϕ_i . For example, the formula

$$(Alice \text{ says } go) \vee ((sit \wedge read) \supset (Alice \text{ says } go))$$

is an instance of the formula $q \vee (r \supset q)$: it can be obtained by replacing every q by $(Alice \text{ says } go)$ and every r by $(sit \wedge read)$. In contrast, the formula

$$(Alice \text{ says } go) \vee ((sit \wedge read) \supset stay)$$

is *not* an instance of the formula $q \vee (r \supset q)$, because the two separate occurrences of q were not replaced by the same formula.

Thus, the *Taut* rule allows us to introduce (i.e., derive) any formula that can be obtained from a propositional-logic tautology by a uniform substitution as described above. For example, since $p \vee \neg p$ is a tautology, we can derive the following formula:

$$(Paul \text{ controls } (read \wedge write)) \vee \neg(Paul \text{ controls } (read \wedge write))$$

3.1.2 The *Modus Ponens* Rule

Another common rule is *Modus Ponens*:

$$\text{Modus Ponens} \quad \frac{\varphi \quad \varphi \supset \varphi'}{\varphi'}$$

This rule states that, if both the implication $\varphi_1 \supset \varphi_2$ and the formula φ_1 have been previously introduced, then we can also introduce the formula φ_2 .

For example, if we have previously derived the two formulas

$$(Bill \text{ says } sell) \supset buy \quad \text{and} \quad Bill \text{ says } sell,$$

then we can use the *Modus Ponens* rule to also derive *buy*.

3.1.3 The *Says* Rule

The *Says* rule is defined as follows:

$$\text{Says} \quad \frac{\varphi}{P \text{ says } \varphi}$$

Informally, this rule states that any principal can make any statement (or safely be assumed to have made any statement) that has already been derived. Thus, for example, if we have previously derived $(read \wedge copy)$, then we can derive *Cara says* $(read \wedge copy)$.

3.1.4 The *MP Says* Rule

The *MP Says* axiom serves as a version of modus ponens for statements made by principals:

$$\text{MP Says} \quad \frac{}{(P \text{ says } (\varphi \supset \varphi')) \supset (P \text{ says } \varphi \supset P \text{ says } \varphi')}$$

In effect, this rule allows us to distribute the *says* operator over implications. For example, this axiom allows us to derive the following formula:

$$(Graham \text{ says } (sit \supset eat)) \supset ((Graham \text{ says } sit) \supset (Graham \text{ says } eat)).$$

3.1.5 The *Speaks For* Rule

The *Speaks For* axiom is defined as follows:

$$\text{Speaks For} \quad \frac{}{P \Rightarrow Q \supset (P \text{ says } \varphi \supset Q \text{ says } \varphi)}$$

In effect, this rule captures our intuition about the speaks-for relation. It states that, if P speaks for Q , then any statements P makes should also be attributable to Q . For example, this axiom allows us to derive the following statement:

$$Del \Rightarrow Ed \supset ((Del \text{ says } buy) \supset (Ed \text{ says } buy)).$$

Were we to subsequently to derive the formula $Del \Rightarrow Ed$, the *Modus Ponens* rule would let us also derive the formula

$$(Del \text{ says } buy) \supset (Ed \text{ says } buy).$$

3.1.6 The *& Says* and *Quoting* Rules

There are two rules that relate statements made by compound principals to those made by simple principals. The first of these is the *& Says* rule:

$$\& \text{ Says} \quad \frac{}{(P \& Q \text{ says } \varphi) \equiv ((P \text{ says } \varphi) \wedge (Q \text{ says } \varphi))}$$

This rule reflects the conjunctive nature of a principal $P \& Q$: the statements made by the compound principal $P \& Q$ (i.e., P in conjunction with Q) are precisely those statements that both P and Q are willing to make individually. For example, the *& Says* rule allows us to derive the following formula:

$$(Faith \& Gale \text{ says } sing) \equiv ((Faith \text{ says } sing) \wedge (Gale \text{ says } sing)).$$

The second such rule is the *Quoting* rule:

$$\text{Quoting} \quad \frac{}{(P \mid Q \text{ says } \varphi) \equiv (P \text{ says } Q \text{ says } \varphi)}$$

This rule captures the underlying intuition behind the compound principal $P \mid Q$: the statements made by $P \mid Q$ (i.e., P quoting Q) are precisely those statements that P claims Q has made. As an example, here is an instance of the *Quoting* rule:

$$(Iona \mid Jürken \text{ says } vote \text{ for } Kory) \equiv (Iona \text{ says } Jürken \text{ says } vote \text{ for } Kory).$$

3.1.7 Properties of \Rightarrow

There are three rules that relate to properties of the \Rightarrow relation, namely idempotency, transitivity, and monotonicity. These rules are all quite simple, but they are

useful for analyzing situations that involve chains of principals speaking for one another.

The *Idempotency of \Rightarrow* rule states that every principal speaks for itself:

$$\text{Idempotency of } \Rightarrow \quad \frac{}{P \Rightarrow P}$$

Thus, for example, with this rule we can derive the following formula:

$$\text{Wallace} \Rightarrow \text{Wallace}.$$

Although this rule may seem both obvious and unnecessary, it can be useful in conjunction with monotonicity to reason about quoting principals, as illustrated later in this subsection.

The *Transitivity of \Rightarrow* rule supports reasoning about chains of principals that represent one another:

$$\text{Transitivity of } \Rightarrow \quad \frac{P \Rightarrow Q \quad Q \Rightarrow R}{P \Rightarrow R}$$

This rule states that, if one principal speaks for a second and the second also speaks for a third, then it is also safe to view the first principal as speaking for the third principal. For example, if we have previously derived the two formulas

$$\text{Kanda} \Rightarrow \text{Theo}, \quad \text{Theo} \Rightarrow \text{Vance},$$

then the *Transitivity of \Rightarrow* rule allows us to also derive

$$\text{Kanda} \Rightarrow \text{Vance}.$$

Finally, the *Monotonicity of \Rightarrow* rule states that quoting principals preserve the speaks-for relationship:

$$\text{Monotonicity of } \Rightarrow \quad \frac{P \Rightarrow P' \quad Q \Rightarrow Q'}{P \mid Q \Rightarrow P' \mid Q'}$$

As an example, suppose that we have already derived the following two formulas:

$$\text{Lowell} \Rightarrow \text{Minnie}, \quad \text{Norma} \Rightarrow \text{Orson}.$$

The *Monotonicity of \Rightarrow* rule allows us to also derive the formula

$$\text{Lowell} \mid \text{Norma} \Rightarrow \text{Minnie} \mid \text{Orson}.$$

To see the utility of idempotency, consider the case where we have already derived the formula $\text{Penny} \Rightarrow \text{Ronald}$, and we would like to derive that

$$\text{Penny} \mid \text{Sylvester} \Rightarrow \text{Ronald} \mid \text{Sylvester}.$$

By first using idempotency to derive $\text{Sylvester} \Rightarrow \text{Sylvester}$, we can use monotonicity to derive the desired formula.

3.1.8 The Equivalence Rule

The *Equivalence* rule allows one to replace subformulas in a formula with equivalent subformulas:

$$\text{Equivalence} \quad \frac{\varphi_1 \equiv \varphi_2 \quad \psi[\varphi_1/q]}{\psi[\varphi_2/q]}$$

To understand this rule, one must first understand the meta-notation $\psi[\varphi/q]$, which denotes the result of replacing every occurrence of the propositional variable q within the formula ψ by the formula φ . For example,

$$(t \supset (P \text{ says } (r \wedge t))) [Q \text{ says } s/t]$$

is simply the formula

$$(Q \text{ says } s) \supset (P \text{ says } (r \wedge Q \text{ says } s)).$$

Thus, the equivalence rule states that, if formulas φ_1 and φ_2 are equivalent, then every occurrence of φ_1 in a formula $\psi[\varphi_1/q]$ can be replaced by φ_2 , resulting in the formula $\psi[\varphi_2/q]$. For example, suppose that we have already derived the following two formulas:

$$s \wedge t \equiv \text{Tom says } r, \quad \text{Arnie} \mid \text{May controls } (s \wedge t).$$

Because the latter formula is equivalent to $(\text{Arnie} \mid \text{May controls } p)[(s \wedge t)/p]$, the *Equivalence* rule allows us to derive the formula

$$\text{Arnie} \mid \text{May controls } (\text{Tom says } r).$$

In fact, by choosing ψ and the propositional variable q judiciously, one can also use the *Equivalence* rule to replace only some of the occurrences of φ_1 with φ_2 . As an example, suppose that we have previously derived the following two formulas:


$$(t \supset r) \equiv P \text{ says } w, \quad (((R \text{ says } (t \supset r)) \wedge (t \supset r)) \supset (T \mid R \text{ controls } (t \supset r))).$$

The latter formula can be obtained by any of the following substitutions (among others):


$$\begin{aligned} & (((R \text{ says } (t \supset r)) \wedge q) \supset (T \mid R \text{ controls } (t \supset r))) [t \supset r/q] \\ & (((R \text{ says } (t \supset r)) \wedge q) \supset (T \mid R \text{ controls } q)) [t \supset r/q] \\ & (((R \text{ says } q) \wedge q) \supset (T \mid R \text{ controls } q)) [t \supset r/q]. \end{aligned}$$

Consequently, the *Equivalence* rule would allow us to derive any of the following formulas:

$$\begin{aligned} & (((R \text{ says } (t \supset r)) \wedge q) \supset (T \mid R \text{ controls } (t \supset r))) [P \text{ says } w/q] \\ & (((R \text{ says } (t \supset r)) \wedge q) \supset (T \mid R \text{ controls } q)) [P \text{ says } w/q] \\ & (((R \text{ says } q) \wedge q) \supset (T \mid R \text{ controls } q)) [P \text{ says } w/q]. \end{aligned}$$

 **Exercise 3.1.1** Calculate the results of each of the following substitutions.

- a. $((t \supset r) \vee (P \text{ says } r))[Q \text{ controls } t/r]$
- b. $w \supset P \text{ says } t[P \mid Q \text{ says } r/w]$
- c. $((s \wedge (Q \text{ says } s)) \supset Q \text{ says } p)[P \text{ says } q/s]$
- d. $((s \wedge (Q \text{ says } t)) \supset Q \text{ says } p)[P \text{ says } q/s]$

 **Exercise 3.1.2** For each pair of formulas (ϕ, ϕ') given below, give formulas ψ, ϕ_1, ϕ_2 such that $\psi[\phi_1/q]$ is ϕ and $\psi[\phi_2/q]$ is ϕ' . The formula ψ should represent as much shared structure as possible between ϕ and ϕ' , allowing ϕ_1 and ϕ_2 to be as simple as possible.

- a. $\phi = (P \ \& \ Q \text{ says } (s \supset t)) \supset t$
 $\phi' = (P \mid Q \text{ says } w) \supset t$
- b. $\phi = (Q \text{ controls } (P \text{ controls } t)) \wedge (P \mid Q \text{ says } t)$
 $\phi' = (Q \text{ controls } (P \text{ controls } t)) \wedge ((P \text{ controls } (Q \text{ controls } t)) \supset t)$
- c. $\phi = ((R \text{ says } s) \wedge (Q \text{ says } R \text{ says } s)) \supset (Q \text{ controls } (R \text{ says } s))$
 $\phi' = ((R \text{ says } s) \wedge (Q \text{ says } R \text{ says } t)) \supset (Q \text{ controls } (R \text{ says } t))$
- d. $\phi = ((R \text{ says } s) \wedge (Q \text{ says } R \text{ says } s)) \supset (Q \text{ controls } (R \text{ says } s))$
 $\phi' = ((T \text{ controls } w) \wedge (Q \text{ says } T \text{ controls } w)) \supset (Q \text{ controls } (T \text{ controls } w))$

3.1.9 The Controls Definition

Finally, the following definition¹ governs the use of controls in our logic:

$$P \text{ controls } \phi \stackrel{\text{def}}{=} (P \text{ says } \phi) \supset \phi$$

This definition states that a formula of the form $P \text{ controls } \phi$ is *syntactic sugar* for the longer expression $(P \text{ says } \phi) \supset \phi$. That is, controls doesn't give our logic any additional expressiveness, but it provides a useful way to make more explicit what will turn out to be a common idiom. This definition means that, any time we see an expression of form $P \text{ controls } \phi$, we can replace it by $(P \text{ says } \phi) \supset \phi$, and vice versa. This definition matches the semantics we defined for our logic in Section 2.3.2: we defined the meaning of $P \text{ controls } \phi$ to be the meaning of $(P \text{ says } \phi) \supset \phi$.

As an example of the use of this definition, we can replace any occurrence of the formula *Lily controls read*—even within the context of a larger formula—by the

¹Note that $\stackrel{\text{def}}{=}$ is meta-notation, rather than a part of the logic's syntax. The net effect, however, of this definition is the same as if we introduced an axiom stating $P \text{ controls } \phi \equiv ((P \text{ says } \phi) \supset \phi)$, in that either side of the definition may safely be replaced by the other in any formula.

FIGURE 3.3 A simple formal proof

1. $Al \text{ says } (r \supset s)$	Assumption
2. r	Assumption
3. $(Al \text{ says } (r \supset s)) \supset (Al \text{ says } r \supset Al \text{ says } s)$	MP Says
4. $Al \text{ says } r \supset Al \text{ says } s$	1,3 Modus Ponens
5. $Al \text{ says } r$	2 Says
6. $Al \text{ says } s$	4,5 Modus Ponens

FIGURE 3.4 A formal proof of the *Controls* rule

1. $P \text{ controls } \phi$	Assumption
2. $P \text{ says } \phi$	Assumption
3. $(P \text{ says } \phi) \supset \phi$	1 Def ^m controls
4. ϕ	2,3 Modus Ponens

formula $(Lily \text{ says } read) \supset read$, and vice versa. Thus, for example, the *Controls* definition allows us to replace the formula

$$Manny \text{ says } (Lily \text{ controls } read)$$

by the formula

$$Manny \text{ says } ((Lily \text{ says } read) \supset read).$$

3.2 Formal Proofs and Theorems

A *formal proof* is a sequence of statements of the logic, where each statement is either an assumption or a statement that can be derived by applying one of the inference rules (or definitions) to previous statements in that sequence. It is customary to sequentially number each of these statements, and to label them either with “Assumption” or with the statement numbers and inference-rule name by which it was deduced. In this book, we will always place all assumptions at the beginning of the proof, so that it is quick and easy to determine the premises upon which a conclusion depends.

For example, Figure 3.3 presents a simple formal proof. In this case, only the first two statements in the proof are assumptions; every other statement is either an instance of axiom (e.g., Step 3) or a consequence of applying one of the inference rules. As another simple example, Figure 3.4 demonstrates how the definition of the controls operator can be used in a formal proof.

Every formal proof represents a *theorem*, which is really just a derived inference rule. Specifically, if the only assumptions of the formal proof are statements


H_1, \dots, H_k , and if the final statement of the proof is C , then the proof corresponds to a derived inference rule with the following form:


$$\frac{H_1 \quad \dots \quad H_k}{C}.$$

Thus, the formal proofs in Figure 3.3 and Figure 3.4 correspond respectively to the following two derived theorems²:

$$\frac{Al \text{ says } (r \supset s) \quad r}{Al \text{ says } s} \quad \frac{P \text{ controls } \phi \quad P \text{ says } \phi}{\phi}.$$


These theorems can now be used as additional inference rules in any future proof, without affecting that proof's validity. We shall find the latter derived rule very useful in subsequent chapters, and hence we give it a specific name (*Controls*). In fact, there are several derived rules—some from propositional logic and others related to access control—that we will find convenient to have in our arsenal. For easy reference, we summarize those rules (along with *Controls*) in Figure 3.5, leaving most of their proofs as exercises for the reader. However, Figure 3.6 gives a proof for the *Conjunction* rule, which also demonstrates the general form for many of the other proofs.


 **Exercise 3.2.1** Give a formal proof of the *Hypothetical Syllogism* derived rule from Figure 3.5.


 **Exercise 3.2.2** Technically speaking, the *Equivalence* rule given in Figure 3.1 permits replacements in only one direction: having deduced $\phi_1 \equiv \phi_2$, one can replace occurrences of ϕ_1 in a formula by ϕ_2 , but not vice versa.

Give a formal proof of the following derived rule which permits replacements in the opposite direction³:

$$\frac{\phi_1 \equiv \phi_2 \quad \Psi[\phi_2/q]}{\Psi[\phi_1/q]}.$$

 **Exercise 3.2.3** Give a formal proof for the *Derived Speaks For* rule given in Figure 3.5.

 **Exercise 3.2.4** Give a formal proof for the *Derived Controls* rule given in Figure 3.5.

 **Exercise 3.2.5** Give a formal proof for the *Says Simplification* derived rules given in Figure 3.5.

 **Exercise 3.2.6** Give a formal proof for the following derivable inference rule:

²Technically, the proof of Figure 3.4 is a *proof schema* that represents a *theorem schema*, in that P and ϕ are *meta-variables* that range over (respectively) all possible principal expressions and formulas. Thus, it provides a recipe for any particular instances of P and ϕ . In this book, we shall blur the distinction between proofs and proof schemas, as well as theorems and theorem schemas.

³Henceforth in this book, we shall not distinguish between these two versions of *Equivalence*.


FIGURE 3.5 Some useful derived rules

<i>Conjunction</i>		$\frac{\varphi_1 \quad \varphi_2}{\varphi_1 \wedge \varphi_2}$	
<i>Simplification (1)</i>	$\frac{\varphi_1 \wedge \varphi_2}{\varphi_1}$	<i>Simplification (2)</i>	$\frac{\varphi_1 \wedge \varphi_2}{\varphi_2}$
<i>Disjunction (1)</i>	$\frac{\varphi_1}{\varphi_1 \vee \varphi_2}$	<i>Disjunction (2)</i>	$\frac{\varphi_2}{\varphi_1 \vee \varphi_2}$
<i>Modus Tollens</i>	$\frac{\varphi_1 \supset \varphi_2 \quad \neg \varphi_2}{\neg \varphi_1}$	<i>Double negation</i>	$\frac{\neg \neg \varphi}{\varphi}$
<i>Disjunctive Syllogism</i>	$\frac{\varphi_1 \vee \varphi_2 \quad \neg \varphi_1}{\varphi_2}$	<i>Hypothetical Syllogism</i>	$\frac{\varphi_1 \supset \varphi_2 \quad \varphi_2 \supset \varphi_3}{\varphi_1 \supset \varphi_3}$
<i>Controls</i>		$\frac{P \text{ controls } \varphi \quad P \text{ says } \varphi}{\varphi}$	
<i>Derived Speaks For</i>	$\frac{P \Rightarrow Q \quad P \text{ says } \varphi}{Q \text{ says } \varphi}$	<i>Derived Controls</i>	$\frac{P \Rightarrow Q \quad Q \text{ controls } \varphi}{P \text{ controls } \varphi}$
<i>Says Simplification (1)</i>	$\frac{P \text{ says } (\varphi_1 \wedge \varphi_2)}{P \text{ says } \varphi_1}$	<i>Says Simplification (2)</i>	$\frac{P \text{ says } (\varphi_1 \wedge \varphi_2)}{P \text{ says } \varphi_2}$


FIGURE 3.6 A formal proof of *Conjunction*

1. φ_1	Assumption
2. φ_2	Assumption
3. $\varphi_1 \supset (\varphi_2 \supset (\varphi_1 \wedge \varphi_2))$	Taut
4. $\varphi_2 \supset (\varphi_1 \wedge \varphi_2)$	1,3 Modus Ponens
5. $\varphi_1 \wedge \varphi_2$	2,4 Modus Ponens

$$\frac{P \text{ says } \varphi_1 \quad P \Rightarrow Q}{Q \text{ says } (\varphi_2 \supset \varphi_1)}.$$

 **Exercise 3.2.7** Give a formal proof for the following derivable inference rule:

$$\frac{P \text{ says } (Q \text{ controls } \varphi) \quad P \mid Q \text{ says } \varphi}{P \text{ says } \varphi}.$$

 **Exercise 3.2.8** Give a formal proof for the following derivable inference rule:

$$\frac{(P \mid Q) \text{ controls } \varphi \quad R \Rightarrow Q \quad P \text{ says } R \text{ says } \varphi}{\varphi}.$$

3.3 Soundness of Logical Rules

The rules presented in Section 3.1 are merely rules for manipulating formulas. Up to this point, we have no reason to believe in their *logical consistency*, which makes using them a risky prospect. To show that the inference rules are consistent, we must first use the Kripke-structure semantics to prove their *soundness*.

Specifically, an inference rule

$$\frac{H_1 \quad \dots \quad H_k}{C}$$

is *sound* with respect to the Kripke semantics provided that, whenever a Kripke structure \mathcal{M} satisfies all of the rule's hypotheses (i.e., H_1, \dots, H_k), it also satisfies the rule's consequent. It follows that an inference rule is *not sound* if there exists even just a single Kripke structure $\mathcal{M} = \langle W, I, J \rangle$ such that, for each H_i , $\mathcal{M} \models H_i$ and yet $\mathcal{M} \not\models C$.

We now use the Kripke semantics given in Figure 2.1 to prove the soundness of the inference rules *Taut*, *Modus Ponens*, and *Says*; we also provide a proof sketch for the soundness of *Equivalence*. The proofs of soundness for the remaining rules are left as exercises for the reader.

Example 3.1 (Soundness of *Taut*)

Let φ be an instance of a propositional-logic tautology ψ . We need to show that, for every Kripke structure $\mathcal{M} = \langle W, I, J \rangle$, φ is true in every world in W (i.e., $\mathcal{E}_{\mathcal{M}}[\varphi] = W$).

Let $\mathcal{M}_a = \langle W_a, I_a, J_a \rangle$ be an arbitrary Kripke structure. Because φ is an instance of ψ , there exist propositional variables p_1, \dots, p_k and modal formulas $\varphi_1, \dots, \varphi_k$ such that φ is obtained from ψ by replacing each p_i by φ_i . Without loss of generality, we assume that the variables p_1, \dots, p_k do not appear in φ .

We now construct a new model $\mathcal{M}' = \langle W_a, I', J_a \rangle$, where I' is defined as follows: for each p_i ($1 \leq i \leq k$), $I'(p_i) = \mathcal{E}_{\mathcal{M}_a}[\varphi_i]$; for all other propositional variables q , $I'(q) = I(q)$. Thus, for each $1 \leq i \leq k$,

$$\mathcal{E}_{\mathcal{M}'}[p_i] = \mathcal{E}_{\mathcal{M}_a}[\varphi_i];$$

likewise, for all other propositional variables q ,

$$\mathcal{E}_{\mathcal{M}'}[[q]] = \mathcal{E}_{\mathcal{M}_a}[[q]].$$

From these facts, it is straightforward to show by induction on the structure of ψ that

$$\mathcal{E}_{\mathcal{M}'}[[\psi]] = \mathcal{E}_{\mathcal{M}_a}[[\psi]].$$

Because ψ is a propositional-logic tautology, ψ is true in all worlds, independent of the interpretation of its propositional variables. Therefore, $\mathcal{E}_{\mathcal{M}'}[[\psi]] = W_a$, and thus $\mathcal{E}_{\mathcal{M}_a}[[\psi]] = W_a$ as well.

Because \mathcal{M}_a was arbitrary, we have shown that the *Taut* rule is sound. \diamond

Example 3.2 (Soundness of Modus Ponens)

To prove that the *Modus Ponens* inference rule is sound, we must prove the following, for all formulas ϕ and ϕ' , and for all Kripke structures $\mathcal{M} = \langle W, I, J \rangle$:

If $\mathcal{M} \models \phi$ and $\mathcal{M} \models \phi \supset \phi'$, then $\mathcal{M} \models \phi'$.

That is, we need to show that, whenever $\mathcal{E}_{\mathcal{M}}[[\phi]] = W$ and $\mathcal{E}_{\mathcal{M}}[[\phi \supset \phi']] = W$, it is also the case that $\mathcal{E}_{\mathcal{M}}[[\phi']] = W$.

Therefore, we consider an arbitrary model $\mathcal{M}_a = \langle W_a, I_a, J_a \rangle$ for which $\mathcal{E}_{\mathcal{M}_a}[[\phi]] = W_a$ and $\mathcal{E}_{\mathcal{M}_a}[[\phi \supset \phi']] = W_a$, and we will show that $\mathcal{E}_{\mathcal{M}_a}[[\phi']] = W_a$ necessarily follows. Working straight from the definition of the evaluation functions $\mathcal{E}_{\mathcal{M}}[[_]]$ given in Figure 2.1, we see that

$$\begin{aligned} \mathcal{E}_{\mathcal{M}_a}[[\phi \supset \phi']] &= (W_a - \mathcal{E}_{\mathcal{M}_a}[[\phi]]) \cup \mathcal{E}_{\mathcal{M}_a}[[\phi']] \\ &= (W_a - W_a) \cup \mathcal{E}_{\mathcal{M}_a}[[\phi']] \\ &= \mathcal{E}_{\mathcal{M}_a}[[\phi']]. \end{aligned}$$

Thus, $\mathcal{E}_{\mathcal{M}_a}[[\phi']] = \mathcal{E}_{\mathcal{M}_a}[[\phi \supset \phi']] = W_a$. Because \mathcal{M}_a was arbitrary, we have shown that the *Modus Ponens* rule is sound. \diamond

Example 3.3 (Soundness of Says)

To prove that the inference rule *Says* is sound, we must prove the following, for all Kripke structures $\mathcal{M} = \langle W, I, J \rangle$:

If $\mathcal{M} \models \phi$, then (for all principals P) $\mathcal{M} \models P \text{ says } \phi$.

That is, we need to show that whenever $\mathcal{E}_{\mathcal{M}}[[\phi]] = W$, it is also the case that (for every principal P) $\mathcal{E}_{\mathcal{M}}[[P \text{ says } \phi]] = W$.

As before, we start by considering an arbitrary Kripke structure $\mathcal{M}_a = \langle W_a, I_a, J_a \rangle$ that satisfies the formula ϕ (i.e., for which $\mathcal{E}_{\mathcal{M}_a}[\phi] = W_a$). We also let Q be an arbitrary principal. From the semantic definitions of Figure 2.1, we see that

$$\mathcal{E}_{\mathcal{M}_a}[Q \text{ says } \phi] = \{w \mid J_a(Q)(w) \subseteq \mathcal{E}_{\mathcal{M}_a}[\phi]\} = \{w \mid J_a(Q)(w) \subseteq W_a\}.$$

Because $J_a(Q)$ is by definition a subset of $W_a \times W_a$, every $w' \in W_a$ satisfies the constraint that $J_a(Q)(w') \subseteq W_a$, and thus $\mathcal{E}_{\mathcal{M}_a}[Q \text{ says } \phi] = W_a$. Because both Q and \mathcal{M}_a were arbitrary, we have shown that the *Says* inference rule is sound. \diamond

Example 3.4 (Soundness of Equivalence)

To prove that the inference rule *Equivalence* is sound, we must prove the following, for all formulas ϕ_1, ϕ_2, ψ , propositional variables q , and Kripke structures $\mathcal{M} = \langle W, I, J \rangle$:

$$\text{If } \mathcal{M} \models \phi_1 \equiv \phi_2 \text{ and } \mathcal{M} \models \psi[\phi_1/q], \text{ then } \mathcal{M} \models \psi[\phi_2/q].$$

That is, we need to show that whenever $\mathcal{E}_{\mathcal{M}}[\phi_1 \equiv \phi_2] = W$ and $\mathcal{E}_{\mathcal{M}}[\psi[\phi_1/q]] = W$, it is also the case that $\mathcal{E}_{\mathcal{M}}[\psi[\phi_2/q]] = W$.

The proof proceeds by a straightforward induction on the structure of ψ , using the fact from Exercise 2.3.9 that $\mathcal{E}_{\mathcal{M}}[\phi_1] = \mathcal{E}_{\mathcal{M}}[\phi_2]$ whenever $\mathcal{E}_{\mathcal{M}}[\phi_1 \equiv \phi_2] = W$. \diamond



Exercise 3.3.1 Prove the soundness of the *Speaks For* inference rule.



Exercise 3.3.2 Prove the soundness of the *& Says* inference rule. (Hint: Exercise 2.3.9 is useful here.)



Exercise 3.3.3 Prove the soundness of the *Quoting* inference rule. (Hint: Exercise 2.3.9 is useful here.)



Exercise 3.3.4 Prove the soundness of the *Transitivity of \Rightarrow* inference rule.



Exercise 3.3.5 Prove the soundness of the *Monotonicity of \Rightarrow* inference rule.



Exercise 3.3.6 Consider the following formula, which intuitively states that every principal has the jurisdiction to select its own proxies:

$$(P \text{ says } (Q \Rightarrow P)) \supset (Q \Rightarrow P).$$

Prove that this formula would not make for a sound axiom. That is, find a particular Kripke structure $\mathcal{M} = \langle W, I, J \rangle$ such that

$$\mathcal{M} \not\models (P \text{ says } (Q \Rightarrow P)) \supset (Q \Rightarrow P).$$



Exercise 3.3.7 Consider the following formula, which intuitively states that *controls* distributes over conjunction:

$$(P \text{ controls } (\varphi_1 \wedge \varphi_2)) \equiv ((P \text{ controls } \varphi_1) \wedge (P \text{ controls } \varphi_2)).$$

Prove that this formula would not make for a sound axiom. That is, find particular formulas φ_1, φ_2 and a particular Kripke structure $\mathcal{M} = \langle W, I, J \rangle$ such that

$$\mathcal{M} \not\models (P \text{ controls } (\varphi_1 \wedge \varphi_2)) \equiv ((P \text{ controls } \varphi_1) \wedge (P \text{ controls } \varphi_2)).$$



Exercise 3.3.8 Consider the following plausible inference rule:

$$\frac{P \ \& \ Q \text{ controls } \varphi \quad P \text{ says } \varphi}{Q \text{ controls } \varphi}.$$

Determine whether or not this rule is sound, and justify your answer:

- If you determine that the rule is sound, prove its soundness.
- If you determine that the rule is not sound, give (and explain) a particular Kripke structure, principals P and Q , and formula φ that demonstrate the lack of soundness.



Exercise 3.3.9 Consider the following plausible inference rule:

$$\frac{P \Rightarrow R \quad R \Rightarrow Q}{R \text{ says } \varphi \equiv P \ \& \ Q \text{ says } \varphi}.$$

Determine whether or not this rule is sound, and justify your answer:

- If you determine that the rule is sound, prove its soundness.
- If you determine that the rule is not sound, give (and explain) a particular Kripke structure, principals P, Q, R and formula φ that demonstrate the lack of soundness.



Exercise 3.3.10 Consider the following plausible inference rule:

$$\frac{P \Rightarrow Q \quad Q \Rightarrow P}{P \text{ says } \varphi \equiv Q \text{ says } \varphi}.$$

Determine whether or not this rule is sound, and justify your answer:

- If you determine that the rule is sound, prove its soundness.
- If you determine that the rule is not sound, give (and explain) a particular Kripke structure, principals P, Q, R and formula φ that demonstrate the lack of soundness.



Exercise 3.3.11 Consider the following plausible inference rule:

$$\frac{P \text{ controls } (\varphi_1 \wedge \varphi_2)}{P \text{ controls } \varphi_1}.$$

Determine whether or not this rule is sound, and justify your answer:

- If you determine that the rule is sound, prove its soundness.
- If you determine that the rule is not sound, give (and explain) a particular Kripke structure, principals P, Q, R and formula φ that demonstrate the lack of soundness.

3.4 Summary

As seen in the previous chapter, Kripke structures provide precise meanings for the statements of the access-control logic. However, reasoning at the level of Kripke structures is extremely cumbersome. Furthermore, it is not always possible to know which Kripke structures accurately represent a particular situation.

To avoid the necessity of reasoning about particular structures, we introduced a collection of inference rules that are *sound*. The Kripke-structure semantics provide the basis for guaranteeing this soundness. The benefits of reasoning with sound inference rules include the convenience they provide as well as the guarantee they provide: any situation that satisfies the initial assumptions is guaranteed to satisfy the deduced consequences. We therefore have a basis for answering the question “Should this access-control request be granted?”, as well as an explicit accounting for all assumptions on which the analysis depends.

The learning outcomes associated with this chapter appear in Figure 3.7.

3.5 Further Reading

The inference rules of the access-control logic are based on the semantics presented in the previous chapter. Our semantics is similar to the logic of Abadi, Lampson, and colleagues (Lampson et al., 1992; Abadi et al., 1993). The notion of soundness based on Kripke models is part of standard modal logic. The reader can consult Hughes and Cresswell (Hughes and Cresswell, 1996) for more on this subject.

FIGURE 3.7 Learning outcomes for Chapter 3

After completing this chapter, you should be able to achieve the following learning outcomes at several levels of knowledge:

Analysis

- When given a set of assumptions and a goal that logically follows from those assumptions, you should be able to give a formal proof of that goal.

Synthesis

- When given a sound axiom or inference rule for the access-control logic, you should be able to prove its soundness in the underlying Kripke model.
- When given a potential inference rule for the access-control logic that is *not sound*, you should be able to construct a Kripke structure that demonstrates its lack of soundness.

Evaluation

- When given a theory, inference rules, and a proof, you should be able to judge if the proof is correct.
 - When given a proposed inference rule, you should be able to judge whether or not it is sound.
-

