#### Policy-Based Design and Verification for Mission Assurance

Shiu-Kai Chin<sup>1</sup>, Sarah Muccio<sup>2</sup>, Susan Older<sup>1</sup>, and Thomas Vestal<sup>1,2</sup> <sup>1</sup>Syracuse University, Syracuse, New York <sup>2</sup>Air Force Research Laboratory, Rome, New York

#### Why Trust and Assurance Matter

"That is, the pilot can trust information that a target is the foe, not innocent inhabitants of a school building or hospital or embassy. ... This new way of war is data dependent. So we need to think in terms of trust and securing trust."

Michael Wynne, Former SECAF

"No operator should ever have to ask ... 'Will my weapon work?' ... Cyberspace warfare creates just this possibility."

General John A. Shaud, USAF

#### Why Trust and Assurance Matter

"That is, the pilot can trust information that a target is the foe, not innocent inhabitants of a school building or hospital or embassy. ... This new way of war is data dependent. So we need to think in terms of trust and securing trust."

Michael Wynne, Former SECAF

"No operator should ever have to ask ... 'Will my weapon work?' ... Cyberspace warfare creates just this possibility."

General John A. Shaud, USAF

#### Why Trust and Assurance Matter

"That is, the pilot can trust information that a target is the foe, not innocent inhabitants of a school building or hospital or embassy. ... This new way of war is data dependent. So we need to think in terms of trust and securing trust."

#### Michael Wynne, Former SECAF

"No operator should ever have to ask ... 'Will my weapon work?' ... Cyberspace warfare creates just this possibility."

General John A. Shaud, USAF

#### Intended audience

• Designers, builders, specifiers, buyers, and evaluators of secure and trustworthy computer and information systems

Focus: access policies and concepts of operation

- hardware, virtual machines, networks
- credentials, authority, delegation
- confidentiality & integrity policies
- Logic is a means to an end
  - means of description
  - inference rules
  - theorem-based design & verification (proofs)

Designers who sleep well combine experience with math & logic

#### Intended audience

• Designers, builders, specifiers, buyers, and evaluators of secure and trustworthy computer and information systems

Focus: access policies and concepts of operation

- hardware, virtual machines, networks
- credentials, authority, delegation
- confidentiality & integrity policies
- Logic is a means to an end
  - means of description
  - inference rules
  - theorem-based design & verification (proofs)

Intended audience

 Designers, builders, specifiers, buyers, and evaluators of secure and trustworthy computer and information systems

#### Focus: access policies and concepts of operation

- hardware, virtual machines, networks
- credentials, authority, delegation
- confidentiality & integrity policies
- Logic is a means to an end
  - means of description
  - inference rules
  - theorem-based design & verification (proofs)

Intended audience

 Designers, builders, specifiers, buyers, and evaluators of secure and trustworthy computer and information systems

#### Focus: access policies and concepts of operation

- hardware, virtual machines, networks
- credentials, authority, delegation
- confidentiality & integrity policies
- Logic is a means to an end
  - means of description
  - inference rules
  - theorem-based design & verification (proofs)

Intended audience

 Designers, builders, specifiers, buyers, and evaluators of secure and trustworthy computer and information systems

#### Focus: access policies and concepts of operation

- hardware, virtual machines, networks
- credentials, authority, delegation
- confidentiality & integrity policies
- Logic is a means to an end
  - means of description
  - inference rules
  - theorem-based design & verification (proofs)

Intended audience

 Designers, builders, specifiers, buyers, and evaluators of secure and trustworthy computer and information systems

#### Focus: access policies and concepts of operation

- hardware, virtual machines, networks
- credentials, authority, delegation
- confidentiality & integrity policies
- Logic is a means to an end
  - means of description
  - inference rules
  - theorem-based design & verification (proofs)

Intended audience

• Designers, builders, specifiers, buyers, and evaluators of secure and trustworthy computer and information systems

Focus: access policies and concepts of operation

- hardware, virtual machines, networks
- credentials, authority, delegation
- confidentiality & integrity policies

#### Logic is a means to an end

- means of description
- inference rules
- theorem-based design & verification (proofs)

Intended audience

• Designers, builders, specifiers, buyers, and evaluators of secure and trustworthy computer and information systems

Focus: access policies and concepts of operation

- hardware, virtual machines, networks
- credentials, authority, delegation
- confidentiality & integrity policies

#### Logic is a means to an end

- means of description
- inference rules
- theorem-based design & verification (proofs)

Intended audience

• Designers, builders, specifiers, buyers, and evaluators of secure and trustworthy computer and information systems

Focus: access policies and concepts of operation

- hardware, virtual machines, networks
- credentials, authority, delegation
- confidentiality & integrity policies

#### Logic is a means to an end

- means of description
- inference rules
- theorem-based design & verification (proofs)

Intended audience

• Designers, builders, specifiers, buyers, and evaluators of secure and trustworthy computer and information systems

Focus: access policies and concepts of operation

- hardware, virtual machines, networks
- credentials, authority, delegation
- confidentiality & integrity policies

#### Logic is a means to an end

- means of description
- inference rules
- theorem-based design & verification (proofs)

Designers who sleep well combine experience with math & logic

Intended audience

• Designers, builders, specifiers, buyers, and evaluators of secure and trustworthy computer and information systems

Focus: access policies and concepts of operation

- hardware, virtual machines, networks
- credentials, authority, delegation
- confidentiality & integrity policies
- Logic is a means to an end
  - means of description
  - inference rules
  - theorem-based design & verification (proofs)



When given a command/request, trust assumptions, credentials, jurisdiction, authority, and policy

- Logically justify whether the command/request is honored or not
- Anything less is regarded as a don't know, don't care, or incompetence

No different for hardware designers and verifiers

4/15



When given a command/request, trust assumptions, credentials, jurisdiction, authority, and policy

- Logically justify whether the command/request is honored or not
- Anything less is regarded as a don't know, don't care, or incompetence

No different for hardware designers and verifiers



When given a command/request, trust assumptions, credentials, jurisdiction, authority, and policy

- Logically justify whether the command/request is honored or not
- Anything less is regarded as a don't know, don't care, or incompetence

No different for hardware designers and verifiers



When given a command/request, trust assumptions, credentials, jurisdiction, authority, and policy

- Logically justify whether the command/request is honored or not
- Anything less is regarded as a don't know, don't care, or incompetence

No different for hardware designers and verifiers

4/15

Access-control logic is used in the same way hardware engineers use propositional logic to specify, design, and verify hardware

- Modification of multi-agent propositional modal logic created by Abadi, Burrows, Lampson, and Plotkin
- Implemented as a conservative extension to the Cambridge Higher Order Logic (HOL-4) Kananaskis 5 theorem prover (joint work done with Lockwood Morris)
- Routinely taught to SU graduate students in *Principles of Distributed Access Control* course
- Used since 2003 by over 226 ROTC cadets from over 40 universities as part of Air Force Research Lab's *Advanced Course in Engineering for Cybersecurity Bootcamp*

Methods usable by practicing engineers and provide assurance

Access-control logic is used in the same way hardware engineers use propositional logic to specify, design, and verify hardware

- Modification of multi-agent propositional modal logic created by Abadi, Burrows, Lampson, and Plotkin
- Implemented as a conservative extension to the Cambridge Higher Order Logic (HOL-4) Kananaskis 5 theorem prover (joint work done with Lockwood Morris)
- Routinely taught to SU graduate students in *Principles of Distributed Access Control* course
- Used since 2003 by over 226 ROTC cadets from over 40 universities as part of Air Force Research Lab's Advanced Course in Engineering for Cybersecurity Bootcamp

Methods usable by practicing engineers and provide assurance

Access-control logic is used in the same way hardware engineers use propositional logic to specify, design, and verify hardware

- Modification of multi-agent propositional modal logic created by Abadi, Burrows, Lampson, and Plotkin
- Implemented as a conservative extension to the Cambridge Higher Order Logic (HOL-4) Kananaskis 5 theorem prover (joint work done with Lockwood Morris)
- Routinely taught to SU graduate students in *Principles of Distributed Access Control* course
- Used since 2003 by over 226 ROTC cadets from over 40 universities as part of Air Force Research Lab's Advanced Course in Engineering for Cybersecurity Bootcamp

Methods usable by practicing engineers and provide assurance

Access-control logic is used in the same way hardware engineers use propositional logic to specify, design, and verify hardware

- Modification of multi-agent propositional modal logic created by Abadi, Burrows, Lampson, and Plotkin
- Implemented as a conservative extension to the Cambridge Higher Order Logic (HOL-4) Kananaskis 5 theorem prover (joint work done with Lockwood Morris)
- Routinely taught to SU graduate students in *Principles of Distributed Access Control* course
- Used since 2003 by over 226 ROTC cadets from over 40 universities as part of Air Force Research Lab's Advanced Course in Engineering for Cybersecurity Bootcamp

Methods usable by practicing engineers and provide assurance

Access-control logic is used in the same way hardware engineers use propositional logic to specify, design, and verify hardware

- Modification of multi-agent propositional modal logic created by Abadi, Burrows, Lampson, and Plotkin
- Implemented as a conservative extension to the Cambridge Higher Order Logic (HOL-4) Kananaskis 5 theorem prover (joint work done with Lockwood Morris)
- Routinely taught to SU graduate students in *Principles of Distributed Access Control* course
- Used since 2003 by over 226 ROTC cadets from over 40 universities as part of Air Force Research Lab's Advanced Course in Engineering for Cybersecurity Bootcamp

Methods usable by practicing engineers and provide assurance

Access-control logic is used in the same way hardware engineers use propositional logic to specify, design, and verify hardware

- Modification of multi-agent propositional modal logic created by Abadi, Burrows, Lampson, and Plotkin
- Implemented as a conservative extension to the Cambridge Higher Order Logic (HOL-4) Kananaskis 5 theorem prover (joint work done with Lockwood Morris)
- Routinely taught to SU graduate students in *Principles of Distributed Access Control* course
- Used since 2003 by over 226 ROTC cadets from over 40 universities as part of Air Force Research Lab's Advanced Course in Engineering for Cybersecurity Bootcamp

Methods usable by practicing engineers and provide assurance

#### **CONOPS** definition

"The CONOPS clearly and concisely expresses what [is to be] accomplish[ed] and how it will be done using available resources. It describes how the actions of ... components and supporting organizations will be integrated, synchronized, and phased to accomplish the mission ..."

JP 5-0, Joint Operation Planning

- Reveals the thinking of commanders in terms of mission requirements, critical capabilities, policies, jurisdiction, and trust assumptions
- Mission assurance requires commanders and implementers precisely and accurately agree on the CONOPS

#### **CONOPS** definition

"The CONOPS clearly and concisely expresses what [is to be] accomplish[ed] and how it will be done using available resources. It describes how the actions of ... components and supporting organizations will be integrated, synchronized, and phased to accomplish the mission ..."

JP 5-0, Joint Operation Planning

- Reveals the thinking of commanders in terms of mission requirements, critical capabilities, policies, jurisdiction, and trust assumptions
- Mission assurance requires commanders and implementers precisely and accurately agree on the CONOPS

#### **CONOPS** definition

"The CONOPS clearly and concisely expresses what [is to be] accomplish[ed] and how it will be done using available resources. It describes how the actions of ... components and supporting organizations will be integrated, synchronized, and phased to accomplish the mission ..."

JP 5-0, Joint Operation Planning

- Reveals the thinking of commanders in terms of mission requirements, critical capabilities, policies, jurisdiction, and trust assumptions
- Mission assurance requires commanders and implementers precisely and accurately agree on the CONOPS

#### **CONOPS** definition

"The CONOPS clearly and concisely expresses what [is to be] accomplish[ed] and how it will be done using available resources. It describes how the actions of ... components and supporting organizations will be integrated, synchronized, and phased to accomplish the mission ..."

JP 5-0, Joint Operation Planning

- Reveals the thinking of commanders in terms of mission requirements, critical capabilities, policies, jurisdiction, and trust assumptions
- Mission assurance requires commanders and implementers precisely and accurately agree on the CONOPS

#### **CONOPS** definition

"The CONOPS clearly and concisely expresses what [is to be] accomplish[ed] and how it will be done using available resources. It describes how the actions of ... components and supporting organizations will be integrated, synchronized, and phased to accomplish the mission ..."

JP 5-0, Joint Operation Planning

- Reveals the thinking of commanders in terms of mission requirements, critical capabilities, policies, jurisdiction, and trust assumptions
- Mission assurance requires commanders and implementers precisely and accurately agree on the CONOPS

#### Purpose

- Show how we describe and verify the logical consistency of CONOPS
- Show that the logic, proofs, and methods are well within the capabilities of practicing engineers
- Formally reasoning about CONOPS provides insight and precision into what is being relied upon, trust assumptions, policies, delegations, and flow of control

- Overview of the logic
- General representation of CONOPS
- A specific example
- Conclusions

#### Purpose

- Show how we describe and verify the logical consistency of CONOPS
- Show that the logic, proofs, and methods are well within the capabilities of practicing engineers
- Formally reasoning about CONOPS provides insight and precision into what is being relied upon, trust assumptions, policies, delegations, and flow of control

- Overview of the logic
- General representation of CONOPS
- A specific example
- Conclusions

#### Purpose

- Show how we describe and verify the logical consistency of CONOPS
- Show that the logic, proofs, and methods are well within the capabilities of practicing engineers
- Formally reasoning about CONOPS provides insight and precision into what is being relied upon, trust assumptions, policies, delegations, and flow of control

- Overview of the logic
- General representation of CONOPS
- A specific example
- Conclusions

#### Purpose

- Show how we describe and verify the logical consistency of CONOPS
- Show that the logic, proofs, and methods are well within the capabilities of practicing engineers
- Formally reasoning about CONOPS provides insight and precision into what is being relied upon, trust assumptions, policies, delegations, and flow of control

- Overview of the logic
- General representation of CONOPS
- A specific example
- Conclusions

Purpose

- Show how we describe and verify the logical consistency of CONOPS
- Show that the logic, proofs, and methods are well within the capabilities of practicing engineers
- Formally reasoning about CONOPS provides insight and precision into what is being relied upon, trust assumptions, policies, delegations, and flow of control

- Overview of the logic
- General representation of CONOPS
- A specific example
- Conclusions

Purpose

- Show how we describe and verify the logical consistency of CONOPS
- Show that the logic, proofs, and methods are well within the capabilities of practicing engineers
- Formally reasoning about CONOPS provides insight and precision into what is being relied upon, trust assumptions, policies, delegations, and flow of control

- Overview of the logic
- General representation of CONOPS
- A specific example
- Conclusions
# Purpose & Preview

Purpose

- Show how we describe and verify the logical consistency of CONOPS
- Show that the logic, proofs, and methods are well within the capabilities of practicing engineers
- Formally reasoning about CONOPS provides insight and precision into what is being relied upon, trust assumptions, policies, delegations, and flow of control

# Preview

- Overview of the logic
- General representation of CONOPS
- A specific example
- Conclusions

# Purpose & Preview

Purpose

- Show how we describe and verify the logical consistency of CONOPS
- Show that the logic, proofs, and methods are well within the capabilities of practicing engineers
- Formally reasoning about CONOPS provides insight and precision into what is being relied upon, trust assumptions, policies, delegations, and flow of control

# Preview

- Overview of the logic
- General representation of CONOPS
- A specific example
- Conclusions

# Purpose & Preview

Purpose

- Show how we describe and verify the logical consistency of CONOPS
- Show that the logic, proofs, and methods are well within the capabilities of practicing engineers
- Formally reasoning about CONOPS provides insight and precision into what is being relied upon, trust assumptions, policies, delegations, and flow of control

# Preview

- Overview of the logic
- General representation of CONOPS
- A specific example
- Conclusions

#### Syntax

### BNF

- Principals (actors) P ::= A / P&Q / P | Q
- Statements they  $\varphi ::= P / \neg \varphi / \varphi_1 \land \varphi_2 / \varphi_1 \lor \varphi_2 / \varphi_1 \supset \varphi_2 / \varphi_1 \equiv \varphi_2 /$ make  $P \Rightarrow Q / P$  says  $\varphi / P$  controls  $\varphi / P$  reps Q on  $\varphi$

#### Kripke structures

		€ <sub>M</sub> [[p]]		
	$PropVar \to \mathcal{P}(W)$	$\mathcal{E}_{\mathcal{M}}[\![ \neg \varphi ]\!]$		$W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi \rrbracket$
	$PName \to \mathcal{P}(W \times W)$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \wedge \varphi_2  rbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1  rbracket \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2  rbracket$
		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \vee \varphi_2 \rrbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2 \rrbracket$		$(W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket) \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \equiv \varphi_2 \rrbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2  rbracket \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \supset \varphi_1  rbracket$
		$\mathcal{E}_{\mathcal{M}}\llbracket P \Rightarrow Q \rrbracket$		$\begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases}$
		$\mathcal{E}_{\mathcal{M}}$ [P says $\varphi$ ]		$\{w J(P)(w)\subseteq \mathcal{E}_{\mathcal{M}}\llbracket\varphi\rrbracket\}$
		$\mathcal{E}_{\mathcal{M}}$ [P controls $\varphi$ ]		$\mathcal{E}_{\mathcal{M}}[(P \text{ says } \varphi) \supset \varphi]$
		$\mathcal{E}_{\mathcal{M}}[P \text{ reps } Q \text{ on } \varphi]$		$\mathcal{E}_{\mathcal{M}}\llbracket P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi \rrbracket$
			•	□ → <i>◆ 園 → ◆</i> ミ → ◆ ミ → りゃ

# Syntax

# BNF

- Principals (actors) P ::= A / P&Q
- Statements they make

- $\begin{array}{rcl} := & A \ / \ P \& Q \ / \ P \ | \ Q \\ \\ := & p \ / \ \neg \varphi \ / \ \varphi_1 \land \varphi_2 \ / \ \varphi_1 \lor \varphi_2 \ / \ \varphi_1 \supset \varphi \end{array}$ 
  - $P \Rightarrow Q / P$  says  $\varphi / P$  controls  $\varphi / P$  reps Q on  $\varphi$

### Kripke structures

#### Semantics

		€ <sub>M</sub> [[p]]	
	$PropVar \to \mathcal{P}(W)$	$\mathcal{E}_{\mathcal{M}}[\![\neg \varphi]\!]$	$W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi \rrbracket$
	$PName \to \mathcal{P}(W \times W)$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \wedge \varphi_2  rbracket$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1  rbracket \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2  rbracket$
		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \vee \varphi_2 \rrbracket$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2 \rrbracket$	$(W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket) \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \equiv \varphi_2 \rrbracket$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2  rbrace \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \supset \varphi_1  rbrace$
		$\mathcal{E}_{\mathcal{M}}\llbracket P \Rightarrow Q \rrbracket$	$\begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases}$
		$\mathcal{E}_{\mathcal{M}}$ [P says $\varphi$ ]	$\{w J(P)(w)\subseteq \mathcal{E}_{\mathcal{M}}\llbracket\varphi\rrbracket\}$
		$\mathcal{E}_{\mathcal{M}}[P \text{ controls } \varphi]$	$\mathcal{E}_{\mathcal{M}}[(P \text{ says } \varphi) \supset \varphi]$
		$\mathcal{E}_{\mathcal{M}}[P \text{ reps } Q \text{ on } \varphi]$	$\mathcal{E}_{\mathcal{M}}\llbracket P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi \rrbracket$
			□ ▶ < @ ▶ < 볼 ▶ < 볼 ▶ 월 - ∽ Q

### Syntax

# BNF

- Principals (actors)
- ::= A / P & Q / P |
- Statements they make

 $P \rightarrow Q / P \text{ savs } \varphi / \varphi_1 \lor \varphi_2 / \varphi_1 \lor \varphi_2 / \varphi_1 \supset \varphi_2 / \varphi_1 \equiv \varphi_2 / P \Rightarrow Q / P \text{ savs } \varphi / P \text{ controls } \varphi / P \text{ reps } Q \text{ on } \varphi$ 

### Kripke structures

		$\mathcal{E}_{\mathcal{M}}\llbracket p  rbracket$		
	$PropVar \to \mathcal{P}(W)$	$\mathcal{E}_{\mathcal{M}}[\![\neg \varphi]\!]$		$W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi  rbracket$
	$PName \to \mathcal{P}(W \times W)$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \wedge \varphi_2 \rrbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1  brace \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2  brace$
		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \lor \varphi_2 \rrbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1  brace \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2  brace$
		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2 \rrbracket$		$(W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket) \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \equiv \varphi_2 \rrbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2  brace \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \supset \varphi_1  brace$
		$\mathcal{E}_{\mathcal{M}}\llbracket P \Rightarrow Q \rrbracket$		$\begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases}$
		$\mathcal{E}_{\mathcal{M}}$ [P says $\varphi$ ]		$\{w J(P)(w)\subseteq \mathcal{E}_{\mathcal{M}}\llbracket\varphi\rrbracket\}$
		$\mathcal{E}_{\mathcal{M}}$ [P controls $\varphi$ ]		$\mathcal{E}_{\mathcal{M}}\llbracket(P \text{ says } \varphi) \supset \varphi rbrace$
		$\mathcal{E}_{\mathcal{M}}[P \text{ reps } Q \text{ on } \varphi]$		${\mathcal E}_{\mathcal M}\llbracket P   {\mathcal Q}  \operatorname{says}  arphi \supset {\mathcal Q}  \operatorname{says}  arphi  rbracket$
			٩	□ › < @ › < 볼 › < 볼 › 볼 · ♡ < () 8/15

### Syntax

### BNF

- Principals (actors) P ::= A / P & Q / P | Q
- Statements they make

 $\begin{array}{l} p \ / \ \neg \varphi \ / \ \varphi_1 \land \varphi_2 \ / \ \varphi_1 \lor \varphi_2 \ / \ \varphi_1 \supset \varphi_2 \ / \ \varphi_1 \equiv \varphi_2 \ / \\ P \Rightarrow Q \ / \ P \text{ says } \varphi \ / \ P \text{ controls } \varphi \ / \ P \text{ reps } Q \text{ on } \varphi \end{array}$ 

#### Kripke structures

		$\mathcal{E}_{\mathcal{M}}\llbracket p  rbracket$		
	$PropVar \to \mathcal{P}(W)$	$\mathcal{E}_{\mathcal{M}}[\![\neg \varphi]\!]$		$W = \mathcal{E}_{\mathcal{M}}\llbracket \varphi \rrbracket$
	$PName \to \mathcal{P}(W \times W)$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \wedge \varphi_2  rbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1  brace \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2  brace$
		$\mathcal{E}_{\mathcal{M}}[\![\varphi_1 \lor \varphi_2]\!]$		$\mathcal{E}_{\mathcal{M}}[\varphi_1] \cup \mathcal{E}_{\mathcal{M}}[\varphi_2]$
		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2 \rrbracket$		$(W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket) \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \equiv \varphi_2 \rrbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2  brace \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \supset \varphi_1  brace$
		$\mathcal{E}_{\mathcal{M}}\llbracket P \Rightarrow Q\rrbracket$		$\begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases}$
		$\mathcal{E}_{\mathcal{M}}$ [P says $\varphi$ ]		$\{w J(P)(w)\subseteq \mathcal{E}_{\mathcal{M}}\llbracket\varphi\rrbracket\}$
		$\mathcal{E}_{\mathcal{M}}$ [P controls $\varphi$ ]		$\mathcal{E}_{\mathcal{M}}[(P \text{ says } \varphi) \supset \varphi]$
		$\mathcal{E}_{\mathcal{M}}[P \text{ reps } Q \text{ on } \varphi]$		$\mathcal{E}_{\mathcal{M}}\llbracket P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi \rrbracket$
			۲	□ ▶ 《圖 ▶ 《볼 ▶ 《볼 ▶ 볼 · 옛익() 8/15

### Syntax

### BNF

- Principals (actors) P ::=
- Statements they make

- = A / P & Q / P | Q  $= p / \neg \varphi / \varphi_1 \land \varphi_2 / \varphi_1 \lor \varphi_2 / \varphi_1 \supset \varphi_2 / \varphi_1 \equiv \varphi$ 
  - $P \Rightarrow Q / P$  says  $\varphi / P$  controls  $\varphi / P$  reps Q on  $\varphi$

### Kripke structures

		$\mathcal{E}_{\mathcal{M}}\llbracket p  rbracket$		
	$PropVar \to \mathcal{P}(W)$	$\mathcal{E}_{\mathcal{M}}[\![\neg \varphi]\!]$		$W = \mathcal{E}_{\mathcal{M}}\llbracket \varphi \rrbracket$
	$PName \to \mathcal{P}(W \times W)$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \wedge \varphi_2  rbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1  brace \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2  brace$
		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \lor \varphi_2 \rrbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2 \rrbracket$		$(W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket) \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \equiv \varphi_2 \rrbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2  brace \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \supset \varphi_1  brace$
		$\mathcal{E}_{\mathcal{M}}\llbracket P \Rightarrow Q\rrbracket$		$egin{cases} W, &  ext{if } J(\mathcal{Q}) \subseteq J(\mathcal{P}) \ \emptyset, &  ext{otherwise} \end{cases}$
		$\mathcal{E}_{\mathcal{M}}$ [P says $\varphi$ ]		$\{w J(P)(w)\subseteq \mathcal{E}_{\mathcal{M}}\llbracket\varphi\rrbracket\}$
		$\mathcal{E}_{\mathcal{M}}[P \text{ controls } \varphi]$		$\mathcal{E}_{\mathcal{M}}\llbracket(P \text{ says } \varphi) \supset \varphi\rrbracket$
		$\mathcal{E}_{\mathcal{M}}[P \text{ reps } Q \text{ on } \varphi]$		$\mathcal{E}_{\mathcal{M}}\llbracket P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi \rrbracket$
			•	□ ▶ 《圖 ▶ 《 볼 ▶ 《 볼 ▶ ~ 볼 ~ 옛 � ( 8/15

# SyntaxBNF• Principals (actors)P ::= A / P & Q / P | Q• Statements they<br/>make $\varphi ::= P / \neg \varphi / \varphi_1 \land \varphi_2 / \varphi_1 \lor \varphi_2 / \varphi_1 \supseteq \varphi_2 / \varphi_1 \equiv \varphi_2 / P \Rightarrow Q / P \text{ says } \varphi / P \text{ controls } \varphi / P \text{ reps } Q \text{ on } \varphi$

### Kripke structures

		$\mathcal{E}_{\mathcal{M}}\llbracket p \rrbracket$		
	$PropVar \to \mathcal{P}(W)$	$\mathcal{E}_{\mathcal{M}}[\![\neg \varphi]\!]$		$W = \mathcal{E}_{\mathcal{M}}\llbracket \varphi  rbracket$
	$PName \to \mathcal{P}(W \times W)$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \wedge \varphi_2  rbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1  brace \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2  brace$
		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \lor \varphi_2 \rrbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2 \rrbracket$		$(W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket) \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \equiv \varphi_2 \rrbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2  brace \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \supset \varphi_1  brace$
		$\mathcal{E}_{\mathcal{M}}\llbracket P \Rightarrow Q \rrbracket$		$\begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases}$
		$\mathcal{E}_{\mathcal{M}}$ [P says $\varphi$ ]		$\{w J(P)(w)\subseteq \mathcal{E}_{\mathcal{M}}\llbracket\varphi\rrbracket\}$
		$\mathcal{E}_{\mathcal{M}}$ [P controls $\varphi$ ]		$\mathcal{E}_{\mathcal{M}}[(P \text{ says } \varphi) \supset \varphi]$
		$\mathcal{E}_{\mathcal{M}}[P \text{ reps } Q \text{ on } \varphi]$		$\mathcal{E}_{\mathcal{M}}\llbracket P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi \rrbracket$
			٩	□ ▶ < ⓓ ▶ < 볼 ▶ < 볼 ▶ 볼 → 오.0 8/15



Kripke structures			semantics	
			$\mathcal{E}_{\mathcal{M}}\llbracket p  rbracket$	
		$PropVar \to \mathcal{P}(W)$	$\mathcal{E}_{\mathcal{M}}[\![ \neg \varphi ]\!]$	$W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi \rrbracket$
		$PName \to \mathcal{P}(W \times W)$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \wedge \varphi_2  rbracket$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1  rbracket \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2  rbracket$
			$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \vee \varphi_2 \rrbracket$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
			$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2 \rrbracket$	$(W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket) \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
			$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \equiv \varphi_2 \rrbracket$	$\mathcal{E}_{\mathcal{M}}\llbracket arphi_1 \supset arphi_2  brace \cap \mathcal{E}_{\mathcal{M}}\llbracket arphi_2 \supset arphi_1  brace$
			$\mathcal{E}_{\mathcal{M}}\llbracket P \Rightarrow Q \rrbracket$	$\begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases}$
			$\mathcal{E}_{\mathcal{M}}$ [P says $\varphi$ ]	$\{w J(P)(w)\subseteq \mathcal{E}_{\mathcal{M}}\llbracket\varphi\rrbracket\}$
			$\mathcal{E}_{\mathcal{M}}$ [P controls $\varphi$ ]	$\mathcal{E}_{\mathcal{M}}\llbracket(P \text{ says } \varphi) \supset \varphi rbrace$
			$\mathcal{E}_{\mathcal{M}}[P \text{ reps } Q \text{ on } \varphi]$	$\mathcal{E}_{\mathcal{M}}\llbracket P \mid Q$ says $\varphi \supset Q$ says $\varphi  rbracket$
				□ › < @ › < 분 › < 분 ·   분 · ∽ <

#### Syntax

### BNF

- Principals (actors) P ::= A / P &
- Statements they make

 $= A / P \otimes Q / P | Q$  $= p / \neg \varphi / \varphi_1 \land \varphi_2 / \varphi_1 \lor \varphi_2 / \varphi_1 \supset \varphi_2 / \varphi_1 \equiv \varphi_2 / P \Rightarrow Q / P \text{ says } \omega / P \text{ controls } \omega / P \text{ reps } Q \text{ on } \omega$ 

### Kripke structures

### Semantics

		E <sub>M</sub> [[p]]	
	$PropVar \to \mathcal{P}(W)$	$\mathcal{E}_{\mathcal{M}}\llbracket \neg \varphi \rrbracket$	$W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi \rrbracket$
	$PName \to \mathcal{P}(W \times W)$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \wedge \varphi_2 \rrbracket$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1  rbracket \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2  rbracket$
		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \vee \varphi_2 \rrbracket$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2 \rrbracket$	$(W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket) \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \equiv \varphi_2 \rrbracket$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2  rbrace \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \supset \varphi_1  rbrace$
		$\mathcal{E}_{\mathcal{M}}\llbracket P \Rightarrow Q\rrbracket$	$egin{cases} W, &  ext{if } J(Q) \subseteq J(P) \ \emptyset, &  ext{otherwise} \end{cases}$
		$\mathcal{E}_{\mathcal{M}}$ [P says $\varphi$ ]	$\{w J(P)(w)\subseteq \mathcal{E}_{\mathcal{M}}\llbracket\varphi\rrbracket\}$
		$\mathcal{E}_{\mathcal{M}}$ [P controls $\varphi$ ]	$\mathcal{E}_{\mathcal{M}}\llbracket(P \text{ says } \varphi) \supset \varphi rbrace$
		$\mathcal{E}_{\mathcal{M}}[P \text{ reps } Q \text{ on } \varphi]$	$\mathcal{E}_{\mathcal{M}}\llbracket P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi  brace$

#### Syntax

### BNF

- Principals (actors) P ::= A / P & C
- Statements they make

 $= A / F \otimes Q / F | Q$ =  $P / \neg \varphi / \varphi_1 \land \varphi_2 / \varphi_1 \lor \varphi_2 / \varphi_1 \supset \varphi_2 / \varphi_1 \equiv \varphi_2 / P \Rightarrow Q / P \text{ says } \varphi / P \text{ controls } \varphi / P \text{ reps } Q \text{ on } \varphi$ 

### Kripke structures

### Semantics

W	=	non-empty {worlds}
		$PropVar \to \mathcal{P}(W)$
		$PName \to \mathcal{P}(W \times W)$

$\mathcal{E}_{\mathcal{M}}\llbracket p\rrbracket$	
$\mathcal{E}_{\mathcal{M}}\llbracket \neg \varphi \rrbracket$	$W = \mathcal{E}_{\mathcal{M}}\llbracket \varphi \rrbracket$
$\mathcal{A}\llbracket \varphi_1 \wedge \varphi_2 \rrbracket$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1  rbrace \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2  rbrace$
$\mathcal{A}\llbracket \varphi_1 \vee \varphi_2 \rrbracket$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
${}_{1}\llbracket \varphi_{1} \supset \varphi_{2}\rrbracket$	$(W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket) \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
${}_{1}\llbracket\varphi_{1}\equiv\varphi_{2}\rrbracket$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2  rbrace \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \supset \varphi_1  rbrace$
$\mathcal{M}\llbracket P \Rightarrow Q \rrbracket$	$\begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases}$
P says φ]	$\{w J(P)(w)\subseteq \mathcal{E}_{\mathcal{M}}\llbracket\varphi\rrbracket\}$
ontrols $\varphi$ ]	$\mathcal{E}_{\mathcal{M}}\llbracket(P \text{ says } \varphi) \supset \varphi rbrace$
s Q on φ]	$\mathcal{E}_{\mathcal{M}}\llbracket P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi  brace$

#### Syntax

### BNF

- Principals (actors) P ::= A / P & C
- Statements they make

 $= A / P \otimes Q / P | Q$  $= p / \neg \varphi / \varphi_1 \land \varphi_2 / \varphi_1 \lor \varphi_2 / \varphi_1 \supset \varphi_2 / \varphi_1 \equiv \varphi_2 / P \Rightarrow Q / P \text{ says } \varphi / P \text{ controls } \varphi / P \text{ reps } Q \text{ on } \varphi$ 

### Kripke structures

### Semantics

W	=	non-empty {worlds}	E _ [[p]]		
1	=	$PropVar \to \mathcal{P}(W)$	$\mathcal{E}_{\mathcal{M}}[\![\neg \varphi]\!]$		$W = \mathcal{E}_{\mathcal{M}}\llbracket \varphi \rrbracket$
		$PName \to \mathcal{P}(W \times W)$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \wedge \varphi_2  rbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1  rbracket \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2  rbracket$
			$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \lor \varphi_2 \rrbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1  rbrace \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2  rbrace$
			$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2 \rrbracket$		$(W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket) \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
			$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \equiv \varphi_2  rbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2  rbrace \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \supset \varphi_1  rbrace$
			$\mathcal{E}_{\mathcal{M}}\llbracket P \Rightarrow Q\rrbracket$		$\begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases}$
			$\mathcal{E}_{\mathcal{M}}$ [P says $\varphi$ ]		$\{w J(P)(w)\subseteq \mathcal{E}_{\mathcal{M}}\llbracket\varphi\rrbracket\}$
			$\mathcal{E}_{\mathcal{M}}$ [P controls $\varphi$ ]		$\mathcal{E}_{\mathcal{M}}\llbracket(P \text{ says } \varphi) \supset \varphi rbrace$
			$\mathcal{E}_{\mathcal{M}}$ [P reps Q on $\varphi$ ]		$\mathcal{E}_{\mathcal{M}}\llbracket P \mid Q$ says $\varphi \supset Q$ says $\varphi  brace$
				4	

#### Syntax

### BNF

- Principals (actors) P ::= A / P&Q
- Statements they make

 $= A / P \otimes Q / P | Q$  $= p / \neg \varphi / \varphi_1 \land \varphi_2 / \varphi_1 \lor \varphi_2 / \varphi_1 \supset \varphi_2 / \varphi_1 \equiv \varphi_2 / P \Rightarrow Q / P \text{ savs } \varphi / P \text{ controls } \varphi / P \text{ reps } Q \text{ on } \varphi$ 

### Kripke structures

W	=	non-empty {worlds}	<i>Е</i> <sub>М</sub> [[р]		
1	=	$PropVar \to \mathcal{P}(W)$	$\mathcal{E}_{\mathcal{M}}[\![\neg \varphi]\!]$		$W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi  rbracket$
J	=	$PName \to \mathcal{P}(W \times W)$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \wedge \varphi_2 \rrbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1  rbracket \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2  rbracket$
			$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \lor \varphi_2 \rrbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
			$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2 \rrbracket$		$(W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket) \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
			$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \equiv \varphi_2 \rrbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2  rbrace \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \supset \varphi_1  rbrace$
			$\mathcal{E}_{\mathcal{M}}\llbracket P \Rightarrow Q  rbracket$		$\begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases}$
			$\mathcal{E}_{\mathcal{M}}$ [P says $\varphi$ ]		$\{w J(P)(w)\subseteq \mathcal{E}_{\mathcal{M}}\llbracket\varphi\rrbracket\}$
			$\mathcal{E}_{\mathcal{M}}$ [P controls $\varphi$ ]		$\mathcal{E}_{\mathcal{M}}\llbracket(P \text{ says } \varphi) \supset \varphi rbrace$
			$\mathcal{E}_{\mathcal{M}}[P \text{ reps } Q \text{ on } \varphi]$		${\mathcal E}_{\mathcal M}\llbracket {\mathcal P}   {\mathcal Q}  { extsf{says}}  arphi \supset {\mathcal Q}  { extsf{says}}  arphi  rbrace$
				•	□ ▶ < ⓓ ▶ < 볼 ▶ < 볼 ▶ 볼 ∽ ♀@ 8/15

#### Syntax

#### BNF

- Principals (actors) P ::= A / P&Q /
- Statements they make

 $= A / F \otimes Q / F | Q$ =  $P / \neg \varphi / \varphi_1 \land \varphi_2 / \varphi_1 \lor \varphi_2 / \varphi_1 \supset \varphi_2 / \varphi_1 \equiv \varphi_2 / P \Rightarrow Q / P \text{ savs } \varphi / P \text{ controls } \varphi / P \text{ reps } Q \text{ on } \varphi$ 

### Kripke structures

W	=	non-empty {worlds}	$\mathcal{E}_{\mathcal{M}}\llbracket p  rbracket$		
1	=	$PropVar \to \mathcal{P}(W)$	$\mathcal{E}_{\mathcal{M}}[\![\neg \varphi]\!]$		$W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi  rbracket$
J	=	$PName \to \mathcal{P}(W \times W)$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \wedge \varphi_2  rbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1  rbrace \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2  rbrace$
$\mathcal{M}$	=	$\langle W, I, J \rangle$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \vee \varphi_2 \rrbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1  rbrace \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2  rbrace$
			$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2 \rrbracket$		$(W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket) \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
			$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \equiv \varphi_2 \rrbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2  rbrace \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \supset \varphi_1  rbrace$
			$\mathcal{E}_{\mathcal{M}}\llbracket P \Rightarrow Q \rrbracket$		$\begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases}$
			$\mathcal{E}_{\mathcal{M}}$ [P says $\varphi$ ]		$\{w J(P)(w)\subseteq \mathcal{E}_{\mathcal{M}}\llbracket\varphi\rrbracket\}$
			$\mathcal{E}_{\mathcal{M}}$ [P controls $\varphi$ ]		$\mathcal{E}_{\mathcal{M}}[(P \text{ says } \varphi) \supset \varphi]$
			$\mathcal{E}_{\mathcal{M}}[P \text{ reps } Q \text{ on } \varphi]$		$\mathcal{E}_{\mathcal{M}}\llbracket P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi \rrbracket$
				•	□ ▶ < ⓓ ▶ < ≧ ▶ < ≧ ▶ ≧ ♪ ♀ ≧ ∽ ♀< 8/15

#### Syntax

### BNF

- Principals (actors) P ::= A / P&Q
- Statements they make

 $= A / P \otimes Q / P | Q$  $= P / \neg \varphi / \varphi_1 \land \varphi_2 / \varphi_1 \lor \varphi_2 / \varphi_1 \supset \varphi_2 / \varphi_1 \equiv \varphi_2 / P \Rightarrow Q / P \text{ says } \varphi / P \text{ controls } \varphi / P \text{ reps } Q \text{ on } \varphi$ 

### Kripke structures

			$\mathcal{E}_{\mathcal{M}}\llbracket p \rrbracket$	=	<i>I</i> ( <i>p</i> )
		$PropVar \to \mathcal{P}(W)$	$\mathcal{E}_{\mathcal{M}}[\![\neg \varphi]\!]$		$W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi \rrbracket$
		$PName \to \mathcal{P}(W \times W)$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \wedge \varphi_2  rbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1  brace \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2  brace$
$\mathcal{M}$	=	$\langle W, I, J \rangle$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \lor \varphi_2 \rrbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
			$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2 \rrbracket$		$(W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket) \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
			$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \equiv \varphi_2 \rrbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2  brace \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \supset \varphi_1  brace$
			$\mathcal{E}_{\mathcal{M}}\llbracket P \Rightarrow Q\rrbracket$		$\begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases}$
			$\mathcal{E}_{\mathcal{M}}$ [P says $\varphi$ ]		$\{w J(P)(w)\subseteq \mathcal{E}_{\mathcal{M}}\llbracket\varphi\rrbracket\}$
			$\mathcal{E}_{\mathcal{M}}$ [P controls $\varphi$ ]		$\mathcal{E}_{\mathcal{M}}\llbracket(P \text{ says } \varphi) \supset \varphi\rrbracket$
			$\mathcal{E}_{\mathcal{M}}[P \text{ reps } Q \text{ on } \varphi]$		$\mathcal{E}_{\mathcal{M}}\llbracket P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi \rrbracket$
				۲	□ ▶ < @ ▶ < 볼 ▶ < 볼 ▶ 월 - 의 Q (0 8/15

#### Syntax

 $\mathcal{M}$ 

### BNF

- Principals (actors) P ::= A / P&G
- Statements they make

 $= A / P \otimes Q / P | Q$  $= p / \neg \varphi / \varphi_1 \land \varphi_2 / \varphi_1 \lor \varphi_2 / \varphi_1 \supset \varphi_2 / \varphi_1 \equiv \varphi_2 / P \Rightarrow Q / P \text{ says } \varphi / P \text{ controls } \varphi / P \text{ reps } Q \text{ on } \varphi$ 

### Kripke structures

### Semantics

		E _ [[p]]		
	$PropVar \to \mathcal{P}(W)$	$\mathcal{E}_{\mathcal{M}}[\![\neg \varphi]\!]$	=	$W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi  rbracket$
	$PName \to \mathcal{P}(W \times W)$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \wedge \varphi_2  rbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1  brace \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2  brace$
=	$\langle W, I, J \rangle$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \lor \varphi_2 \rrbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2 \rrbracket$		$(W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket) \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \equiv \varphi_2 \rrbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2  rbrace \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \supset \varphi_1  rbrace$
		$\mathcal{E}_{\mathcal{M}}\llbracket P \Rightarrow Q \rrbracket$		$\begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases}$
		$\mathcal{E}_{\mathcal{M}}$ [P says $\varphi$ ]		$\{w J(P)(w)\subseteq \mathcal{E}_{\mathcal{M}}\llbracket\varphi\rrbracket\}$
		$\mathcal{E}_{\mathcal{M}}$ [P controls $\varphi$ ]		$\mathcal{E}_{\mathcal{M}}\llbracket(P \text{ says } \varphi) \supset \varphi rbrace$
		$\mathcal{E}_{\mathcal{M}}[P \text{ reps } Q \text{ on } \varphi]$		$\mathcal{E}_{\mathcal{M}}\llbracket P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi  brace$

#### Syntax

### BNF

- Principals (actors) P ::= A / P
- Statements they make

= A / P & Q / P | Q $= p / \neg \varphi / \varphi_1 \land \varphi_2 / \varphi_1 \lor \varphi_2 / \varphi_1 \supset \varphi_2 / \varphi_1 \equiv \varphi_2 / P \Rightarrow Q / P \text{ says } \varphi / P \text{ controls } \varphi / P \text{ reps } Q \text{ on } \varphi$ 

# Kripke structures

### Semantics

- W = non-empty {worlds}
- $I = \operatorname{PropVar} \rightarrow \mathcal{P}(W)$
- $J = \mathsf{PName} \to \mathcal{P}(W \times W)$
- $\mathcal{M} = \langle W, I, J \rangle$

$$\begin{split} \mathcal{E}_{\mathcal{M}}[\rho] &= I(\rho) \\ \mathcal{E}_{\mathcal{M}}[\neg\varphi] &= W - \mathcal{E}_{\mathcal{M}}[\varphi] \\ \mathcal{E}_{\mathcal{M}}[\varphi_{1} \land \varphi_{2}] &= \mathcal{E}_{\mathcal{M}}[\varphi_{1}] \cap \mathcal{E}_{\mathcal{M}}[\varphi_{2}] \\ \mathcal{E}_{\mathcal{M}}[\varphi_{1} \lor \varphi_{2}] &= \mathcal{E}_{\mathcal{M}}[\varphi_{1}] \cup \mathcal{E}_{\mathcal{M}}[\varphi_{2}] \\ \mathcal{E}_{\mathcal{M}}[\varphi_{1} \supset \varphi_{2}] &= (W - \mathcal{E}_{\mathcal{M}}[\varphi_{1}]) \cup \mathcal{E}_{\mathcal{M}}[\varphi_{2}] \\ \mathcal{E}_{\mathcal{M}}[\varphi_{1} \supseteq \varphi_{2}] &= \mathcal{E}_{\mathcal{M}}[\varphi_{1} \supset \varphi_{2}] \cap \mathcal{E}_{\mathcal{M}}[\varphi_{2} \supset \varphi_{1}] \\ \mathcal{E}_{\mathcal{M}}[P \Rightarrow Q] &= \begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases} \\ \mathcal{M}[P \text{ says } \varphi] &= \{w|J(P)(w) \subseteq \mathcal{E}_{\mathcal{M}}[\varphi]\} \\ \text{controls } \varphi] &= \mathcal{E}_{\mathcal{M}}[P \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathbb{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi] \\ \mathcal{E}_{\mathcal{M}}[P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi]$$

#### Syntax

### BNF

- Principals (actors) P ::= A / P &
- Statements they make

= A / P & Q / P | Q $= p / \neg \varphi / \varphi_1 \land \varphi_2 / \varphi_1 \lor \varphi_2 / \varphi_1 \supset \varphi_2 / \varphi_1 \equiv \varphi_2 / P \Rightarrow Q / P \text{ says } \varphi / P \text{ controls } \varphi / P \text{ reps } Q \text{ on } \varphi$ 

# Kripke structures

- 8/15

#### Syntax

### BNF

- Principals (actors) P ::= A / P &
- Statements they make

= A / P & Q / P | Q $= p / \neg \varphi / \varphi_1 \land \varphi_2 / \varphi_1 \lor \varphi_2 / \varphi_1 \supset \varphi_2 / \varphi_1 \equiv \varphi_2 / P \Rightarrow Q / P \text{ says } \varphi / P \text{ controls } \varphi / P \text{ reps } Q \text{ on } \varphi$ 

# Kripke structures

- - 8/15

### Syntax

### BNF

- Principals (actors) P ::= A / P &
- Statements they make

= A / P & Q / P | Q  $= p / \neg \varphi / \varphi_1 \land \varphi_2 / \varphi_1 \lor \varphi_2 / \varphi_1 \supset \varphi_2 / \varphi_1 \equiv \varphi_2 / P \Rightarrow Q / P \text{ says } \varphi / P \text{ controls } \varphi / P \text{ reps } Q \text{ on } \varphi$ 

# Kripke structures

### Semantics

			$\mathcal{E}_{\mathcal{M}}\llbracket p  rbracket$		
		$PropVar \to \mathcal{P}(W)$	$\mathcal{E}_{\mathcal{M}}[\![\neg \varphi]\!]$		$W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi \rrbracket$
		$PName \to \mathcal{P}(W \times W)$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \wedge \varphi_2  rbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1  rbrace \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2  rbrace$
$\mathcal{M}$	=	$\langle W, I, J \rangle$	$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \lor \varphi_2 \rrbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
			$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2 \rrbracket$		$(W - \mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \rrbracket) \cup \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \rrbracket$
			$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \equiv \varphi_2 \rrbracket$		$\mathcal{E}_{\mathcal{M}}\llbracket \varphi_1 \supset \varphi_2  rbrace \cap \mathcal{E}_{\mathcal{M}}\llbracket \varphi_2 \supset \varphi_1  rbrace$
			$\mathcal{E}_{\mathcal{M}}\llbracket P \Rightarrow Q\rrbracket$	=	$\begin{cases} W, & \text{if } J(Q) \subseteq J(P) \\ \emptyset, & \text{otherwise} \end{cases}$
			$\mathcal{E}_{\mathcal{M}}$ [P says $\varphi$ ]	=	$\{w J(P)(w)\subseteq \mathcal{E}_{\mathcal{M}}\llbracket\varphi\rrbracket\}$
			$\mathcal{E}_{\mathcal{M}}$ [P controls $\varphi$ ]	=	$\mathcal{E}_{\mathcal{M}}\llbracket(P \text{ says } \varphi) \supset \varphi rbrace$
			$\mathcal{E}_{\mathcal{M}}$ [P reps Q on $\varphi$ ]	=	$\mathcal{E}_{\mathcal{M}}\llbracket P \mid Q \text{ says } \varphi \supset Q \text{ says } \varphi \rrbracket$

#### CORE INFERENCE RULES

### RULES

- Inconvenient to use Kripke semantics
- Use inference rules  $\frac{H_1 \cdots H_n}{C}$  instead

# Soundness

 $\frac{H_1 \cdots H_n}{C} \text{ is sound if } for$ all Kripke structures  $\mathcal{M}$  and each  $i \in \{1, \dots, n\}$ :

> If all  $\mathcal{E}_{\mathcal{M}}\llbracket H_i \rrbracket = W$ then  $\mathcal{E}_{\mathcal{M}}\llbracket C \rrbracket = W$

- All rules are sound
- All verified in HOL-4 K-5 theorem prover

	$P Q \supset (P Say)$	$S \ arphi \supset Q \ S$	
Quoting $P \mid Q$		says Q s	
& Says P & Q S			
		$ \begin{array}{c} \overline{P \Rightarrow P} \\ P  Q' \Rightarrow \\ Q' \Rightarrow P \mid Q \end{array} $	$\frac{Q}{2}$
P controls	$\varphi \stackrel{\text{def}}{=} (P$		
P reps Q on	φ <sup>def</sup> ₽  <b>@ ∰</b> a	IVS• ⁄≣∋ Q	SBI¥S φ≣

#### Core inference rules

# RULES

- Inconvenient to use Kripke semantics
- Use inference rules  $\frac{H_1 \cdots H_n}{C}$  instead

# Soundness

 $\frac{H_1 \cdots H_n}{C} \text{ is sound if } for$ all Kripke structures  $\mathcal{M}$  and each  $i \in \{1, \dots, n\}$ :

> If all  $\mathcal{E}_{\mathcal{M}}\llbracket H_i \rrbracket = W$ then  $\mathcal{E}_{\mathcal{M}}\llbracket C \rrbracket = W$

- All rules are sound
- All verified in HOL-4 K-5 theorem prover

	$Q \supset (P \text{ says})$	$S \varphi \supset Q S$		
Quoting P   Q		says Q s		
& Says P & Q Sa		ays $\varphi \wedge G$		
		$\frac{P  Q' \Rightarrow}{Q' \Rightarrow P \mid Q}$	$\frac{Q}{2}$	
	y of $  \frac{P   (Q)}{(P   Q)}$			
P controls				
Prens 0 on /	def plama			c

#### Core inference rules

### Rules

- Inconvenient to use Kripke semantics
- Use inference rules  $\frac{H_1 \cdots H_n}{C}$  instead

# Soundness

 $\frac{H_1 \cdots H_n}{C} \text{ is sound if } for$ all Kripke structures  $\mathcal{M}$  and each  $i \in \{1, \dots, n\}$ :

> If all  $\mathcal{E}_{\mathcal{M}}\llbracket H_i \rrbracket = W$ then  $\mathcal{E}_{\mathcal{M}}\llbracket C \rrbracket = W$

- All rules are sound
- All verified in HOL-4 K-5 theorem prover

	$Q \supset (P \text{ says})$	$\varphi \supset Q$ S	
Quoting $P \mid Q$		says q s	
& Says P & Q Sa		$ys \ \varphi \land q$	
		$\begin{array}{c} r \Rightarrow r \\ P  Q' \Rightarrow \\ Q' \Rightarrow P \mid Q \end{array}$	$\frac{Q}{2}$
	y of $  = \frac{P   (Q   Q)}{(P   Q)}$		
P controls			
Preps 0 on //	def BIO ARA	/≤4/≅⊃⊨ ()	Gavs ⁄≣ ✓

#### CORE INFERENCE RULES

### RULES

- Inconvenient to use Kripke semantics
- Use inference rules  $\frac{H_1 \cdots H_n}{C}$  instead

### Soundness

 $\frac{H_1 \cdots H_n}{C} \text{ is sound if } for$ all Kripke structures  $\mathcal{M}$  and each  $i \in \{1, \dots, n\}$ :

> If all  $\mathcal{E}_{\mathcal{M}}\llbracket H_i \rrbracket = W$ then  $\mathcal{E}_{\mathcal{M}}\llbracket C \rrbracket = W$

- All rules are sound
- All verified in HOL-4 K-5 theorem prover

Speaks For $P \Rightarrow Q \supset (P \text{ says } \varphi \supset Q \text{ says } \varphi)$
Quoting $P \mid Q$ says $\varphi \equiv P$ says Q says $\varphi$
& Says $P \& Q$ Says $\varphi \equiv P$ Says $\varphi \wedge Q$ Says $\varphi$
$Monotonicity of \mid \frac{P' \Rightarrow P Q' \Rightarrow Q}{P' \mid Q' \Rightarrow P \mid Q}$
Associativity of $  \frac{P   (Q   R) \text{ says } \varphi}{(P   Q)   R \text{ says } \varphi}$
$P \ {\sf controls} \ \varphi \ \stackrel{{\rm def}}{=} \ (P \ {\sf says} \ \varphi) \supset \varphi$
Preps 0 on a defail a mansa and more a more

#### Core inference rules

### RULES

- Inconvenient to use Kripke semantics
- Use inference rules  $\frac{H_1 \cdots H_n}{C}$  instead

# Soundness

 $\frac{H_1 \cdots H_n}{C} \text{ is sound if } for$ all Kripke structures  $\mathcal{M}$  and each  $i \in \{1, \dots, n\}$ :

> If all  $\mathcal{E}_{\mathcal{M}}\llbracket H_i \rrbracket = W$ then  $\mathcal{E}_{\mathcal{M}}\llbracket C \rrbracket = W$

- All rules are sound
- All verified in HOL-4 K-5 theorem prover

	$\Rightarrow Q \supset (P \text{ say})$	$(S \ arphi \supset Q \ S$	
Quoting $P \mid Q$		says q s	
& Says P & Q S			
		$ \begin{array}{c} \overline{P \Rightarrow P} \\ \overline{P  Q' \Rightarrow} \\ Q' \Rightarrow P \mid Q \end{array} $	$\frac{Q}{2}$
		Q R) says Q) R says	
P controls			
P reps Q on a	ω def Φ₽Ι@∰	avs∢≣⇒ 0	⊊avs ∞≣

#### CORE INFERENCE RULES

### RULES

- Inconvenient to use Kripke semantics
- Use inference rules  $\frac{H_1 \cdots H_n}{C}$  instead

# Soundness

 $\frac{H_1 \cdots H_n}{C} \text{ is sound if } for \\ all Kripke structures <math>\mathcal{M}$  and each  $i \in \{1, \dots, n\}$ :

> If all  $\mathcal{E}_{\mathcal{M}}\llbracket H_i \rrbracket = W$ then  $\mathcal{E}_{\mathcal{M}}\llbracket C \rrbracket = W$

- All rules are sound
- All verified in HOL-4 K-5 theorem prover

	$Q \supset (P \text{ says})$	$S \varphi \supset Q$		
Quoting $P \mid Q$		says Q s		
& Says P & Q S				
		$ \begin{array}{c} \overline{P \Rightarrow P} \\ P  Q' \Rightarrow \\ \overline{Q' \Rightarrow P \mid 0} \end{array} $	<u>Q</u>	
	y of $\left  \frac{P \left  \left( Q \right) \right }{\left( P \right  Q} \right $			
P controls				
P reps Q on Q	o def PI O Ba	NS• æ⊃ G	Seevs ⊘≣	¢

#### Core inference rules

### RULES

- Inconvenient to use Kripke semantics
- Use inference rules  $\frac{H_1 \cdots H_n}{C}$  instead

# Soundness

 $\frac{H_1 \cdots H_n}{C} \text{ is sound if } for \\ all Kripke structures <math>\mathcal{M}$  and each  $i \in \{1, \dots, n\}$ :

If all  $\mathcal{E}_{\mathcal{M}}\llbracket H_i \rrbracket = W$ then  $\mathcal{E}_{\mathcal{M}}\llbracket C \rrbracket = W$ 

- All rules are sound
- All verified in HOL-4 K-5 theorem prover

	$Q \supset (P \text{ says })$	$\varphi \supset Q$ S		
Quoting $P \mid Q$		ays q s		
& Says P & Q Sa		'S $\varphi \wedge Q$		
	$f \text{ of }    \frac{P' \Rightarrow P}{P' \mid Q'}$	$\begin{array}{c} Q' \Rightarrow \\ \hline \Rightarrow P \mid G \end{array}$	$\frac{Q}{2}$	
	of $\left  -\frac{P \left  \left( Q \right  \right.}{\left( P \left  \right. Q \right) \right } \right $			
P controls				
P reps Q on $\varphi$	def PI @ Bavs	54 æ⊃ 0	SEIVS ⊘≣	¢

#### CORE INFERENCE RULES

### RULES

- Inconvenient to use Kripke semantics
- Use inference rules  $\frac{H_1 \cdots H_n}{C}$  instead

# Soundness

 $\frac{H_1 \cdots H_n}{C} \text{ is sound if } for \\ all Kripke structures <math>\mathcal{M}$  and each  $i \in \{1, \dots, n\}$ :

> If all  $\mathcal{E}_{\mathcal{M}}\llbracket H_i \rrbracket = W$ then  $\mathcal{E}_{\mathcal{M}}\llbracket C \rrbracket = W$

- All rules are sound
- All verified in HOL-4 K-5 theorem prover

Speaks For $P \Rightarrow Q$		$Q$ says $\varphi$ )
Quoting P   Q SAY		Q says $\varphi$
& Says P & Q SAYS		$\sim \wedge Q$ says $\varphi$
	$\begin{array}{c} \text{acy of} \Rightarrow & \hline P \Rightarrow \\ P' \Rightarrow P & Q \\ \hline P' \mid Q' \Rightarrow \end{array}$	$\frac{P}{P} \Rightarrow Q$ $P \mid Q$
	$  \qquad \frac{P \mid (Q \mid R) \leq}{(P \mid Q) \mid R \leq}$	
P controls $\varphi$		
P reps Q on $\varphi \stackrel{\mathrm{de}}{\leftarrow}$	£ □ P•   @ @Pavs• @	:∋Q:523¥S ¢≣ - 4

#### CORE INFERENCE RULES

### RULES

- Inconvenient to use Kripke semantics
- Use inference rules  $\frac{H_1 \cdots H_n}{C}$  instead

# Soundness

 $\frac{H_1 \cdots H_n}{C} \text{ is sound if } for$ all Kripke structures  $\mathcal{M}$  and each  $i \in \{1, \dots, n\}$ :

> If all  $\mathcal{E}_{\mathcal{M}}\llbracket H_i \rrbracket = W$ then  $\mathcal{E}_{\mathcal{M}}\llbracket C \rrbracket = W$

- All rules are sound
- All verified in HOL-4 K-5 theorem prover

	$\Rightarrow Q \supset (P \text{ say})$	$(S \ arphi \supset Q \ S)$		
Quoting P	Q says $\varphi \equiv P$	says q s		
& Says P & Q				
		$ \begin{array}{c} \overline{P \Rightarrow P} \\ P & Q' \Rightarrow \\ Q' \Rightarrow P & Q' \end{array} $	$\frac{Q}{2}$	
	$ity of   \frac{P   (G)}{(P   G)}$			
P contro				
P reps Q on	a def 🗗 🗗 🖉 🖓	ans a an a	S≣WS ⊿≣ 🗸	

#### CORE INFERENCE RULES

### RULES

- Inconvenient to use Kripke semantics
- Use inference rules  $\frac{H_1 \cdots H_n}{C}$  instead

# Soundness

 $\frac{H_1 \cdots H_n}{C} \text{ is sound if } for$ all Kripke structures  $\mathcal{M}$  and each  $i \in \{1, \dots, n\}$ :

> If all  $\mathcal{E}_{\mathcal{M}}\llbracket H_i \rrbracket = W$ then  $\mathcal{E}_{\mathcal{M}}\llbracket C \rrbracket = W$

- All rules are sound
- All verified in HOL-4 K-5 theorem prover

	$\triangleright Q \supset (P Say)$	$S \varphi \supset Q S$		
Quoting $P \mid Q$		says q s		
& Says P & Q S				
		$ \begin{array}{c} \overline{P \Rightarrow P} \\ P & Q' \Rightarrow \\ Q' \Rightarrow P \mid Q \end{array} $	$\frac{Q}{2}$	
	$P \mid (G )$			
P controls	$\varphi \stackrel{\mathrm{def}}{=} (F)$			
P reps Q on a	o def PI @ 🗗	avs∢æ⊃ 0	⊊avs ∞≣	c

### Core inference rules

### RULES

- Inconvenient to use Kripke semantics
- Use inference rules  $\frac{H_1 \cdots H_n}{C}$  instead

# Soundness

 $\frac{H_1 \cdots H_n}{C} \text{ is sound if } for$ all Kripke structures  $\mathcal{M}$  and each  $i \in \{1, \dots, n\}$ :

> If all  $\mathcal{E}_{\mathcal{M}}\llbracket H_i \rrbracket = W$ then  $\mathcal{E}_{\mathcal{M}}\llbracket C \rrbracket = W$

- All rules are sound
- All verified in HOL-4 K-5 theorem prover

$\begin{array}{cc} \text{ Taut } & \underset{\varphi}{ & } & \text{ if } \varphi \text{ is an instance of a prop-logic tau-} \\ & \text{ tology } \end{array}$
$\begin{array}{ccc} \textit{Modus Ponens} & \frac{\varphi & \varphi \supset \varphi'}{\varphi'} & \textit{Says} & \frac{\varphi}{\textit{P says } \varphi} \end{array}$
$\frac{MP \text{ Says }}{(P \text{ says } (\varphi \supset \varphi')) \supset (P \text{ says } \varphi \supset P \text{ says } \varphi')}$
Speaks For $\overline{P \Rightarrow Q \supset (P \text{ says } \varphi \supset Q \text{ says } \varphi)}$
Quoting $P \mid Q$ says $\varphi \equiv P$ says $Q$ says $\varphi$
$\& Says  \hline P \& Q \text{ says } \varphi \equiv P \text{ says } \varphi \land Q \text{ says } \varphi$
Idempotency of $\Rightarrow {P \rightarrow P}$
Monotonicity of $  \frac{P' \Rightarrow P  Q' \Rightarrow Q}{P' \mid Q' \Rightarrow P \mid Q}$
Associativity of $  \frac{P   (Q   R) \text{ says } \varphi}{(P   Q)   R \text{ says } \varphi}$
$P \text{ controls } \varphi \stackrel{\text{def}}{=} (P \text{ says } \varphi) \supset \varphi$
P reps Q on $\varphi \stackrel{\mathrm{def}}{=} P   Q $ says $\varphi \supset Q$ says $\varphi \ge -$



Derived inference rule Controls  $\frac{P \text{ controls } \varphi - P \text{ says } \varphi}{\varphi}$ 

All derived rules are sound

- $1. \quad P \text{ controls } \varphi$
- 2. P says  $\varphi$
- 3. *P* says  $\varphi \supset \varphi$

**4**. φ

# Assumption Assumption

def'n controls
 3 Modus Ponens

Derived inference rule Controls  $\frac{P \text{ controls } \varphi - P \text{ says } \varphi}{\varphi}$ 

All derived rules are sound

・ロ ・ ・ 日 ・ ・ 言 ・ く 言 ・ こ う へ で 10 / 15

- 1. P controls  $\varphi$
- 2. P says  $\varphi$
- 3. *P* says  $\varphi \supset \varphi$

4. φ

Assumption Assumption 1 def'n controls 2, 3 Modus Ponens

Derived inference rule Controls  $rac{P ext{ controls } arphi ext{ P says } arphi}{arphi}$ 

All derived rules are sound

1.  $P \text{ controls } \varphi$ 2.  $P \text{ says } \varphi$ 3.  $P \text{ says } \varphi \supset \varphi$ 4.  $\varphi$  Assumption Assumption 1 def'n controls 2, 3 Modus Ponens

Derived inference rule Controls  $rac{P ext{ controls } arphi ext{ P says } arphi}{arphi}$ 

All derived rules are sound
#### A Simple Proof



# $\begin{array}{l} \text{Derived inference rule} \\ \text{Controls} \quad \frac{P \text{ controls } \varphi \quad P \text{ says } \varphi}{\varphi} \end{array}$

All derived rules are sound

#### A Simple Proof



# $\begin{array}{l} \text{Derived inference rule} \\ \text{Controls} \quad \frac{P \text{ controls } \varphi \quad P \text{ says } \varphi}{\varphi} \end{array}$

All derived rules are sound



- Principals are actors
- Assumptions about jurisdiction, policy, and trust are explicit
- Each step in CONOPS is a derived inference rule



- Principals are actors
- Assumptions about jurisdiction, policy, and trust are explicit
- Each step in CONOPS is a derived inference rule



- Principals are actors
- Assumptions about jurisdiction, policy, and trust are explicit
- Each step in CONOPS is a derived inference rule



- Principals are actors
- Assumptions about jurisdiction, policy, and trust are explicit
- Each step in CONOPS is a derived inference rule



Joint Terminal Air Controller



Remotely Piloted Vehicle



Airborne Early Warning & Control



Air Operations Center



Joint Terminal Air Controller





Remotely Piloted Vehicle



Airborne Early Warning & Control



Air Operations Center

12/15





Remotely Piloted Vehicle



Airborne Early Warning & Control



Air Operations Center

3

12/15



Joint Terminal Air Controller



-

Remotely Piloted Vehicle



Airborne Early Warning & Control



Air Operations Center



Joint Terminal Air Controller



-





Airborne Early Warning & Control



Air Operations Center



Statement	Formal Representation
request 1	(Token <sub>Alice</sub>   JTAC) SAYS <i>(strike, target)</i>
relay 1	(K <sub>JTAC-MVA</sub>   JTAC) SAYS (strike, target)
authenticated request 1	JTAC SAYS (strike, target)
request 2	(Token <sub>Bob</sub>   Controller) <b>SAYS</b> (JTAC <b>SAYS</b> (strike, target))
relay 2	(K <sub>Controller-MVA</sub>   Controller) SAYS (JTAC SAYS (strike, target))
authenticated request 2	Controller SAYS (JTAC SAYS <strike, target="">)</strike,>



Statement	Formal Representation
request 1	(Token <sub>Alice</sub>   JTAC) SAYS (strike, target)
relay 1	(K <sub>JTAC-MVA</sub>   JTAC) SAYS (strike, target)
authenticated request 1	JTAC SAYS (strike, target)
request 2	(Token <sub>Bob</sub>   Controller) SAYS (JTAC SAYS (strike, target))
relay 2	(K <sub>Controller-MVA</sub>   Controller) SAYS (JTAC SAYS (strike, target))
authenticated request 2	Controller SAYS (JTAC SAYS <strike, target="">)</strike,>



Statement	Formal Representation
request 1	(Token <sub>Alice</sub>   JTAC) SAYS (strike, target)
relay 1	(K <sub>JTAC-MVA</sub>   JTAC) SAYS (strike, target)
authenticated request 1	JTAC SAYS (strike, target)
request 2	(Token <sub>Bob</sub>   Controller) SAYS (JTAC SAYS (strike, target))
relay 2	(K <sub>Controller-MVA</sub>   Controller) SAYS (JTAC SAYS (strike, target))
authenticated request 2	Controller SAYS (JTAC SAYS <strike, target="">)</strike,>



Statement	Formal Representation
request 1	(Token <sub>Alice</sub>   JTAC) SAYS (strike, target)
relay 1	(K <sub>JTAC-MVA</sub>   JTAC) SAYS (strike, target)
authenticated request 1	JTAC SAYS (strike, target)
request 2	(Token <sub>Bob</sub>   Controller) SAYS (JTAC SAYS (strike, target))
relay 2	(K <sub>Controller-MVA</sub>   Controller) SAYS (JTAC SAYS (strike, target))
authenticated request 2	Controller SAYS (JTAC SAYS (strike, target))



Statement	Formal Representation
request 1	(Token <sub>Alice</sub>   JTAC) <b>SAYS</b> <i>(strike, target)</i>
relay 1	(K <sub>JTAC-MVA</sub>   JTAC) SAYS (strike, target)
authenticated request 1	JTAC SAYS (strike, target)
request 2	(Token <sub>Bob</sub>   Controller) SAYS (JTAC SAYS (strike, target))
relay 2	(K <sub>Controller-MVA</sub>   Controller) SAYS (JTAC SAYS (strike, target))
authenticated request 2	Controller SAYS (JTAC SAYS (strike, target))



Statement	Formal Representation
request 1	(Token <sub>Alice</sub>   JTAC) <b>SAYS</b> <i>(strike, target)</i>
relay 1	(K <sub>JTAC-MVA</sub>   JTAC) SAYS (strike, target)
authenticated request 1	JTAC SAYS (strike, target)
request 2	(Token <sub>Bob</sub>   Controller) <b>SAYS</b> (JTAC <b>SAYS</b> (strike, target))
relay 2	(K <sub>Controller-MVA</sub>   Controller) SAYS (JTAC SAYS (strike, target))
authenticated request 2	Controller SAYS (JTAC SAYS <strike, target="">)</strike,>



Statement	Formal Representation
request 1	(Token <sub>Alice</sub>   JTAC) SAYS (strike, target)
relay 1	(K <sub>JTAC-MVA</sub>   JTAC) SAYS (strike, target)
authenticated request 1	JTAC SAYS (strike, target)
request 2	(Token <sub>Bob</sub>   Controller) <b>SAYS</b> (JTAC <b>SAYS</b> (strike, target))
relay 2	(K <sub>Controller-MVA</sub>   Controller) SAYS (JTAC SAYS (strike, target))
authenticated request 2	Controller SAYS (JTAC SAYS (strike, target))

# Common *form* of requests, delegations, key certificates, jurisdiction, and trust assumptions

Transmitting MVA:

Receiving MVA

 $(K_{MVA_1} | Role) \operatorname{Says} \varphi$   $K_{Auth} \operatorname{Says} (MVA_1 \operatorname{reps} Role \ On \ \varphi)$   $K_{Auth} \operatorname{Says} (K_{MVA_1} \Rightarrow MVA_1)$ Auth controls ( $MVA_1 \operatorname{reps} Role \ On \ \varphi$ )  $Auth \ controls (K_{MVA_1} \Rightarrow MVA_1)$   $K_{Auth} \Rightarrow Auth$   $Role \ \operatorname{Says} \varphi$   $(\Box \triangleright \langle \Box \rangle \land \langle \Box \rangle \land \langle \Xi \rangle \land \langle \Xi \rangle \land \langle \Xi \rangle \land \langle \Xi \rangle$  14/15

# Common *form* of requests, delegations, key certificates, jurisdiction, and trust assumptions

Input (Token or Key | Role) says  $\varphi$ 

Transmitting MVA:

Receiving MVA

 $(K_{MVA_1} | Role) \operatorname{Says} \varphi$   $K_{Auth} \operatorname{Says} (MVA_1 \operatorname{reps} Role \ On \ \varphi)$   $K_{Auth} \operatorname{Says} (K_{MVA_1} \Rightarrow MVA_1)$ Auth controls (MVA\_1 reps Role \ On \ \varphi) Auth controls (K\_{MVA\_1} \Rightarrow MVA\_1)  $K_{Auth} \Rightarrow Auth$   $Role \operatorname{Says} \varphi$   $(\Box \triangleright \langle \Box \rangle \land \langle \Xi \rangle \land \langle \Xi \rangle \land \langle \Xi \rangle \land \langle \Xi \rangle$  14/15

# Common *form* of requests, delegations, key certificates, jurisdiction, and trust assumptions

nput (Token or Key | Role) says  $\varphi$ 

#### Delegation Cert $K_{Auth}$ says (*Person or Object* reps *Role* on $\varphi$ )

Key Certificate $K_{Auth}$  says (Token or Key  $\Rightarrow$  Person or Object)JurisdictionAuth controls (Person or Object reps Role on  $\varphi$ )JurisdictionAuth controls (Token or Key  $\Rightarrow$  Person or Object)Frust Assumption $K_{Auth} \Rightarrow Auth$ 

Transmitting MVA:

Receiving MVA

 $\begin{array}{c} (\mbox{Token} \mid \mbox{Role}) \ \mbox{Says} \ \varphi \\ K_{Auth} \ \mbox{Says} \ (\mbox{Person reps} \ \mbox{Role} \ \mbox{On} \ \varphi) \\ K_{Auth} \ \mbox{Says} \ (\mbox{Token} \Rightarrow \mbox{Person}) \\ Auth \ \mbox{Controls} \ (\mbox{Person reps} \ \mbox{Role} \ \mbox{On} \ \varphi) \\ Auth \ \mbox{Controls} \ (\mbox{Token} \Rightarrow \mbox{Person}) \\ K_{Auth} \ \mbox{Controls} \ (\mbox{Token} \Rightarrow \mbox{Person}) \\ K_{Auth} \ \mbox{Auth} \ \mbox{Controls} \ \mbox{Token} \Rightarrow \mbox{Person}) \\ K_{Auth} \ \mbox{Auth} \ \mbox{Controls} \ \mbox{Token} \Rightarrow \mbox{Person}) \\ K_{Auth} \ \mbox{Auth} \ \mbox{Controls} \ \mbox{Token} \Rightarrow \mbox{Person}) \\ K_{Auth} \ \mbox{Auth} \ \mbox{Controls} \ \mbox{Token} \Rightarrow \mbox{Person}) \\ K_{Auth} \ \mbox{Role} \ \mbox{Controls} \$ 

 $(K_{MVA_{1}} | Role) \text{ says } \varphi$   $K_{Auth} \text{ says } (MVA_{1} \text{ reps } Role \text{ on } \varphi)$   $K_{Auth} \text{ says } (K_{MVA_{1}} \Rightarrow MVA_{1})$   $Auth \text{ controls } (MVA_{1} \text{ reps } Role \text{ on } \varphi)$   $Auth \text{ controls } (K_{MVA_{1}} \Rightarrow MVA_{1})$   $K_{Auth} \Rightarrow Auth$   $Role \text{ says } \varphi$   $(\Box \triangleright \langle \Box \rangle \land \langle \Box \rangle \land \langle \Xi \rangle \land \langle \Xi \rangle \land \langle \Xi \rangle \land \langle \Xi \rangle$  14/15

# Common *form* of requests, delegations, key certificates, jurisdiction, and trust assumptions

nput (Token or Key | Role) says  $\varphi$ 

Delegation Cert  $K_{Auth}$  says (*Person or Object* reps *Role* on  $\varphi$ )

Key Certificate  $K_{Auth}$  says (Token or Key  $\Rightarrow$  Person or Object)

JurisdictionAuth controls (Person or Object reps Role on  $\varphi$ )JurisdictionAuth controls (Token or Key  $\Rightarrow$  Person or Object)ust Assumption $K_{Auth} \Rightarrow Auth$ 

Transmitting MVA:

Receiving MVA

 $(K_{MVA_{1}} | Role) \operatorname{Says} \varphi$   $K_{Auth} \operatorname{Says} (MVA_{1} \operatorname{reps} Role \operatorname{On} \varphi)$   $K_{Auth} \operatorname{Says} (K_{MVA_{1}} \Rightarrow MVA_{1})$   $Auth \operatorname{controls} (MVA_{1} \operatorname{reps} Role \operatorname{On} \varphi)$   $Auth \operatorname{controls} (K_{MVA_{1}} \Rightarrow MVA_{1})$   $K_{Auth} \Rightarrow Auth$   $Role \operatorname{Says} \varphi$   $\Box \triangleright \langle \Box \triangleright \langle \Box \triangleright \langle \Xi \triangleright \langle \Xi \land \Box E \rangle \langle \Xi \rangle \langle \Xi \rangle$  14/15

# Common *form* of requests, delegations, key certificates, jurisdiction, and trust assumptions

Input(Token or Key | Role) says  $\varphi$ Delegation Cert $K_{Auth}$  says (Person or Object reps Role on  $\varphi$ )Key Certificate $K_{Auth}$  says (Token or Key  $\Rightarrow$  Person or Object)JurisdictionAuth controls (Person or Object reps Role on  $\varphi$ )JurisdictionAuth controls (Token or Key  $\Rightarrow$  Person or Object)rust Assumption $K_{Auth} \Rightarrow Auth$ 

Transmitting MVA:

Receiving MVA

 $(K_{MVA_1} | Role) \operatorname{Says} \varphi$   $K_{Auth} \operatorname{Says} (MVA_1 \operatorname{reps} Role \ On \ \varphi)$   $K_{Auth} \operatorname{Says} (K_{MVA_1} \Rightarrow MVA_1)$ Auth controls ( $MVA_1 \operatorname{reps} Role \ On \ \varphi$ )  $Auth \operatorname{controls} (K_{MVA_1} \Rightarrow MVA_1)$   $K_{Auth} \Rightarrow Auth$   $Role \operatorname{Says} \varphi$   $(\Box \triangleright \langle \Box \rangle \land \langle \Box \rangle \land \langle \Xi \rangle \land \langle \Xi \rangle \land \langle \Xi \rangle \land \langle \Xi \rangle$  14/15

# Common *form* of requests, delegations, key certificates, jurisdiction, and trust assumptions

Input (Token or Key | Role) says  $\varphi$ Delegation Cert  $K_{Auth}$  says (Person or Object reps Role on  $\varphi$ ) Key Certificate  $K_{Auth}$  says (Token or Key  $\Rightarrow$  Person or Object) Jurisdiction Auth controls (Person or Object reps Role on  $\varphi$ ) Jurisdiction Auth controls (Token or Key  $\Rightarrow$  Person or Object) ust Assumption  $K_{Auth} \Rightarrow Auth$ 

Transmitting MVA:

Receiving MVA

 $\begin{array}{c} (\textit{Token} \mid \textit{Role}) \; \texttt{Says} \; \varphi \\ K_{Auth} \; \texttt{Says} \; (\textit{Person reps Role ON } \varphi) \\ K_{Auth} \; \texttt{Says} \; (\textit{Token} \Rightarrow \textit{Person}) \\ Auth \; \texttt{Controls} \; (\textit{Person reps Role On } \varphi) \\ Auth \; \texttt{Controls} \; (\textit{Token} \Rightarrow \textit{Person}) \\ K_{Auth} \; \Rightarrow \textit{Auth} \\ \hline \\ MVA \; 1 \; \hline \\ \hline \\ K_{MVA_1} \mid \textit{Role Says} \; \varphi \end{array}$ 

# Common *form* of requests, delegations, key certificates, jurisdiction, and trust assumptions

Input(Token or Key | Role) says  $\varphi$ Delegation Cert $K_{Auth}$  says (Person or Object reps Role on  $\varphi$ )Key Certificate $K_{Auth}$  says (Token or Key  $\Rightarrow$  Person or Object)JurisdictionAuth controls (Person or Object reps Role on  $\varphi$ )JurisdictionAuth controls (Token or Key  $\Rightarrow$  Person or Object)

Trust Assumption

 $K_{Auth} \Rightarrow Auth$ 

```
Transmitting MVA:
```

Receiving MVA

 $(Token | Role) Says \varphi$   $K_{Auth} Says (Person reps Role On \varphi)$   $K_{Auth} Says (Token \Rightarrow Person)$   $Auth controls (Person reps Role On \varphi)$   $Auth controls (Token \Rightarrow Person)$   $K_{Auth} \Rightarrow Auth$   $MVA 1 \qquad K_{Auth} \Rightarrow Auth$   $K_{MVA_1} | Role Says \varphi \qquad M$ 

 $(K_{MVA_{1}} | Role) \operatorname{Says} \varphi$   $K_{Auth} \operatorname{Says} (MVA_{1} \operatorname{reps} Role \operatorname{On} \varphi)$   $K_{Auth} \operatorname{Says} (K_{MVA_{1}} \Rightarrow MVA_{1})$   $Auth \operatorname{controls} (MVA_{1} \operatorname{reps} Role \operatorname{On} \varphi)$   $Auth \operatorname{controls} (K_{MVA_{1}} \Rightarrow MVA_{1})$   $K_{Auth} \Rightarrow Auth$   $Role \operatorname{Says} \varphi$   $\Box \triangleright \langle \Box \triangleright \langle \Box \triangleright \langle \Xi \triangleright \langle \Xi \triangleright \langle \Xi \rangle \langle \Xi \rangle \rangle \langle \Xi \rangle$  14/15

# Common *form* of requests, delegations, key certificates, jurisdiction, and trust assumptions

#### Transmitting MVA:

Receiving MVA

14/15

# Common *form* of requests, delegations, key certificates, jurisdiction, and trust assumptions

#### Transmitting MVA:

Receiving MVA

 $(K_{MVA_{1}} | Role) \text{ Says } \varphi$   $K_{Auth} \text{ Says } (MVA_{1} \text{ reps } Role \text{ On } \varphi)$   $K_{Auth} \text{ Says } (K_{MVA_{1}} \Rightarrow MVA_{1})$   $Auth \text{ controls } (MVA_{1} \text{ reps } Role \text{ On } \varphi)$   $Auth \text{ controls } (K_{MVA_{1}} \Rightarrow MVA_{1})$   $K_{Auth} \Rightarrow Auth$   $Role \text{ Says } \varphi$   $(\Box \Rightarrow \langle \Box \Rightarrow \langle \Xi \Rightarrow \langle \Xi \Rightarrow \langle \Xi \Rightarrow \rangle \langle C \rangle$  14/15

# Common *form* of requests, delegations, key certificates, jurisdiction, and trust assumptions

Input(Token or Key | Role) says  $\varphi$ Delegation Cert $K_{Auth}$  says (Person or Object reps Role on  $\varphi$ )Key Certificate $K_{Auth}$  says (Token or Key  $\Rightarrow$  Person or Object)JurisdictionAuth controls (Person or Object reps Role on  $\varphi$ )JurisdictionAuth controls (Token or Key  $\Rightarrow$  Person or Object)Frust Assumption $K_{Auth} \Rightarrow Auth$ 

#### Transmitting MVA:

#### Receiving MVA

# Common *form* of requests, delegations, key certificates, jurisdiction, and trust assumptions

#### Transmitting MVA:

#### Receiving MVA

(Token | Role) Says  $\varphi$  $(K_{MVA_1} \mid Role)$  Says  $\varphi$  $K_{Auth}$  says (Person reps Role On  $\varphi$ )  $K_{Auth}$  says ( $\hat{M}VA_1$  reps Role On  $\varphi$ )  $K_{Auth}$  Says (Token  $\Rightarrow$  Person)  $K_{Auth}$  says  $(K_{MVA_1} \Rightarrow MVA_1)$ Auth controls (Person reps Role on  $\varphi$ ) Auth controls (MVA<sub>1</sub> reps Role on  $\varphi$ ) Auth controls (Token  $\Rightarrow$  Person) Auth controls  $(K_{MVA_1} \Rightarrow MVA_1)$  $K_{Auth} \Rightarrow Auth$  $K_{Auth} \Rightarrow Auth$ MVA 1 MVA 2  $K_{MVA_1} \mid Role \text{ says } \varphi$ Role Savs  $\varphi$ 

14/15

226+ ACE cadets, captains, & lieutenants from 40+ universities



Formal approach to access control and CONOPS is feasible (with adequate education)

- 21 hours of instruction
- Kripke semantics, basic & distributed access control, delegation, hardware, and confidentiality/integrity policies

Textbook based on accesscontrol logic taught in ACE



Shiu-Kai Chin Susan Older

226+ ACE cadets, captains, & lieutenants from 40+ universities



Formal approach to access control and CONOPS is feasible (with adequate education)

- 21 hours of instruction
- Kripke semantics, basic & distributed access control, delegation, hardware, and confidentiality/integrity policies

Textbook based on accesscontrol logic taught in ACE



Shiu-Kai Chin Susan Older

226+ ACE cadets, captains, & lieutenants from 40+ universities



Formal approach to access control and CONOPS is feasible (with adequate education)

- 21 hours of instruction
- Kripke semantics, basic & distributed access control, delegation, hardware, and confidentiality/integrity policies

Textbook based on accesscontrol logic taught in ACE



Shiu-Kai Chin Susan Older

226+ ACE cadets, captains, & lieutenants from 40+ universities



Formal approach to access control and CONOPS is feasible (with adequate education)

- 21 hours of instruction
- Kripke semantics, basic & distributed access control, delegation, hardware, and confidentiality/integrity policies

Textbook based on accesscontrol logic taught in ACE



Shiu-Kai Chin Susan Older

A CHAPMAN & HALL BOO

226+ ACE cadets, captains, & lieutenants from 40+ universities



Formal approach to access control and CONOPS is feasible (with adequate education)

- 21 hours of instruction
- Kripke semantics, basic & distributed access control, delegation, hardware, and confidentiality/integrity policies

Textbook based on accesscontrol logic taught in ACE



Shiu-Kai Chin Susan Older

Increased their capabilities to design, specify, evaluate, and procure critical systems

> クへで 15/15