KI2023, Berlin, 29.09

Flexible Automation of Quantified Multi-Modal Logics with Interactions

Melanie Taproge and Alexander Steen

University of Greifswald

{s-metapr,alexander.steen}@uni-greifswald.de

Special thanks to:

GI FACHBEREICH KÜNSTLICHE INTELLIGENZ

AKADEMIE DER WISSENSCHAFTEN IN HAMBURG

Agenda

- First-Order Multi-Modal Logics
  - Introduction
  - Characterisation

FOML

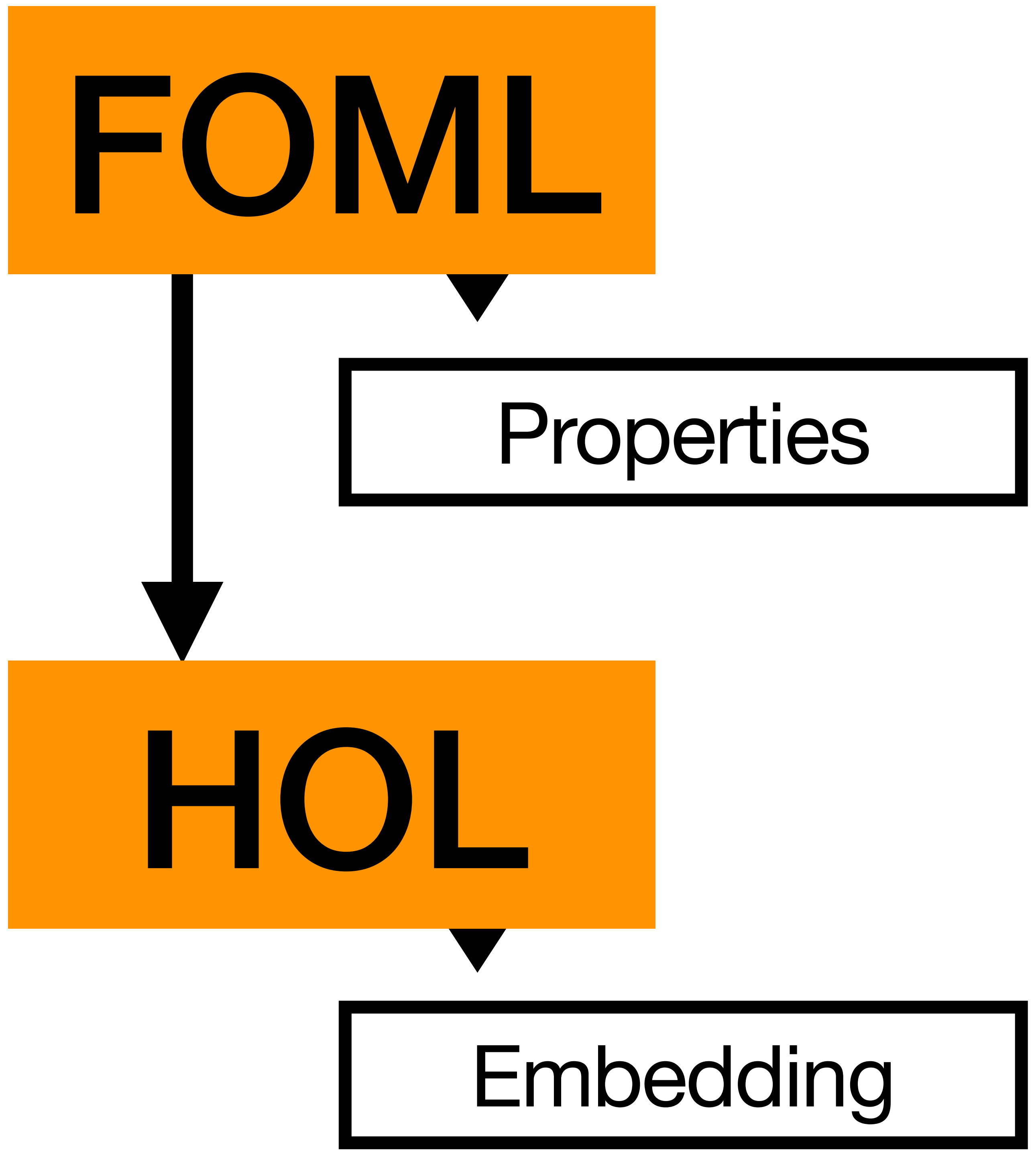
Properties

ATP Systems

Agenda

- First-Order Multi-Modal Logics
  - Introduction
  - Characterisation

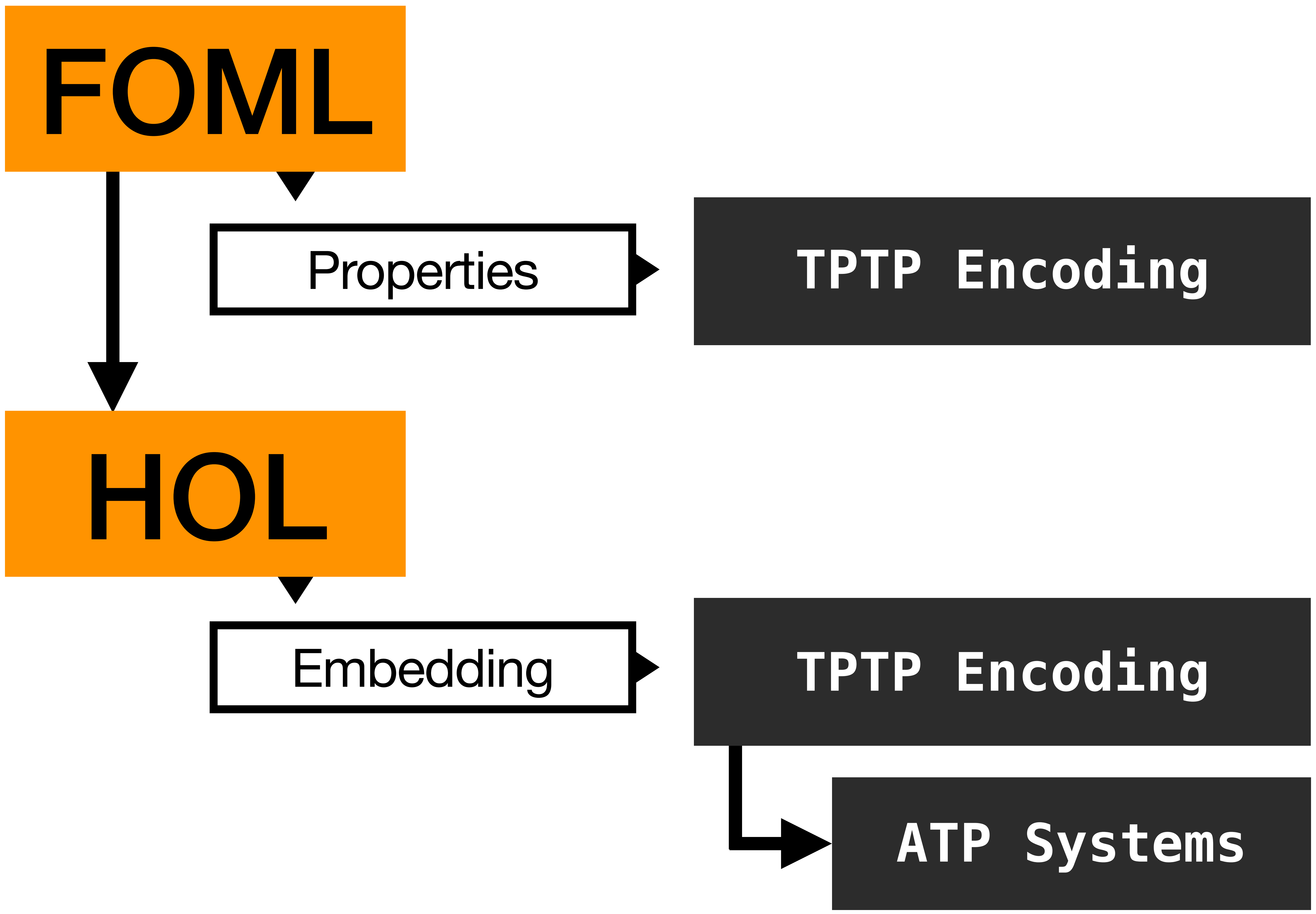
- Higher-Order Logic
  - Introduction
  - Embedding of FOML



ATP Systems

Agenda

- First-Order Multi-Modal Logics
  - Introduction
  - Characterisation
- Higher-Order Logic
  - Introduction
  - Embedding of FOML
- Automation via TPTP Encoding
  - Encoding of HOL and FOML
  - Encoding of FOML Characterisation
- Summary



First-Order Multi-Modal Logics (FOML)

Multi-Modal Logic [5]

Bottom

Syntax:

Propositions

$$A, B := P \in PS \mid \perp$$

First-Order Multi-Modal Logics (FOML)

Multi-Modal Logic [5]

Bottom

PL Operators

Syntax:

Propositions

$$A, B := P \in PS \mid \bot \mid \neg A \mid (A \wedge B) \mid (A \vee B) \mid (A \supset B) \mid (A \equiv B)$$

Sample formula:

$$A \wedge B$$

First-Order Multi-Modal Logics (FOML)

Multi-Modal Logic [5]

Propositions — *Bottom* — *PL Operators* — *„Box"-Operator* — *„Diamond"-Operator*

Syntax:

$$A, B := P \in PS \mid \bot \mid \neg A \mid (A \wedge B) \mid (A \vee B) \mid (A \supset B) \mid (A \equiv B) \mid \Box_i A \mid \Diamond_i A$$

With index set I, e.g. $I = \{alice, bob\}$

Sample formula:

$$\Box_{alice}(A \wedge B)$$

First-Order Multi-Modal Logics (FOML)

Multi-Modal Logic [5]

Syntax:

Bottom

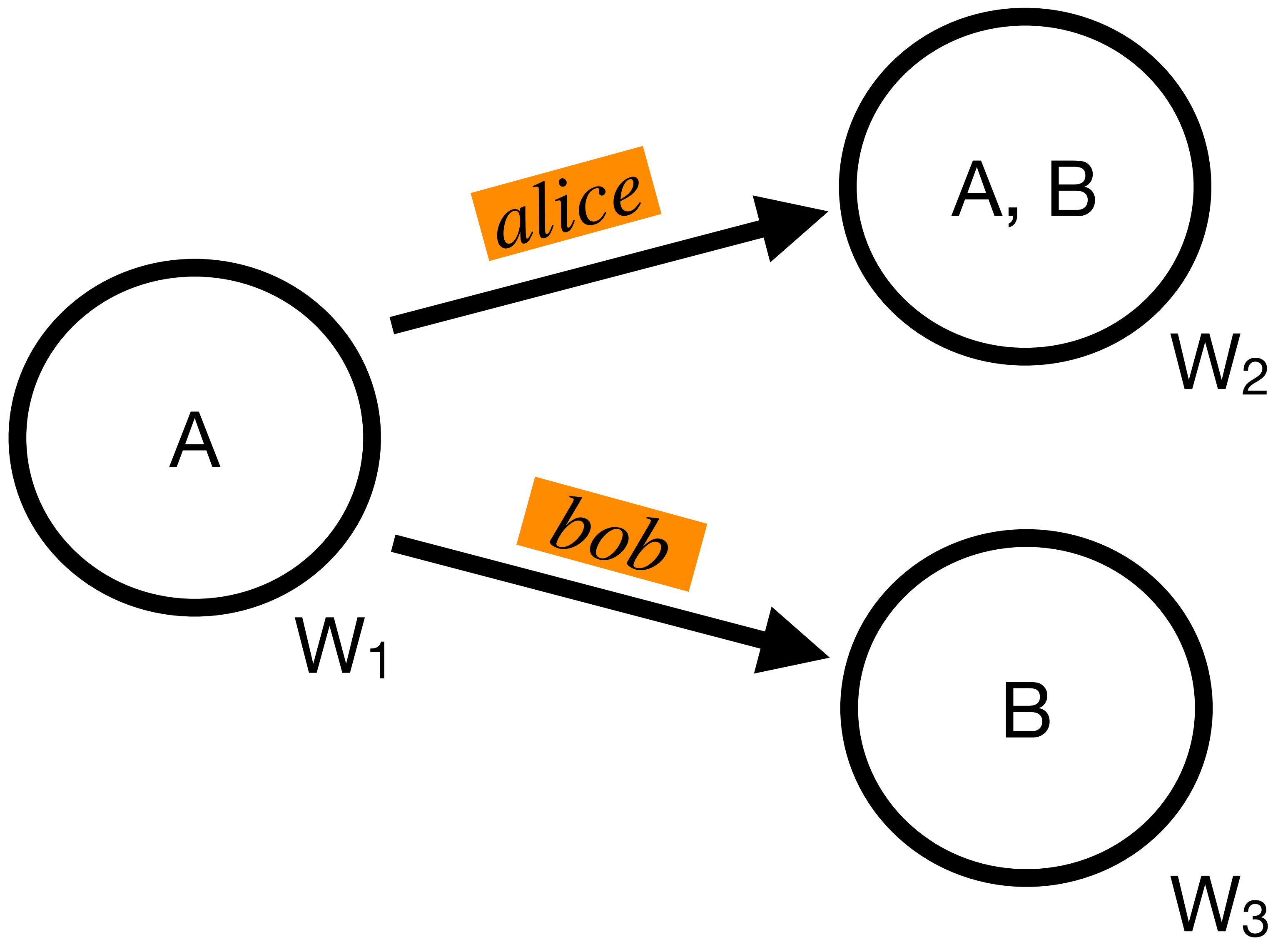
Propositions *PL Operators* *„Box"-Operator* *„Diamond"-Operator*

$$A, B := P \in PS \mid \bot \mid \neg A \mid (A \wedge B) \mid (A \vee B) \mid (A \supset B) \mid (A \equiv B) \mid \Box_i A \mid \Diamond_i A$$

With index set I, e.g. $I = \{alice, bob\}$

Sample formula: $M, W_1 \vDash \Box_{alice}(A \wedge B)$

Semantics: Evaluation relative to possible worlds using Kripke-Structures: $M = (W, \{R_i\}_{i \in I}, V)$



First-Order Multi-Modal Logics (FOML)

Multi-Modal Logic [5]

Syntax:

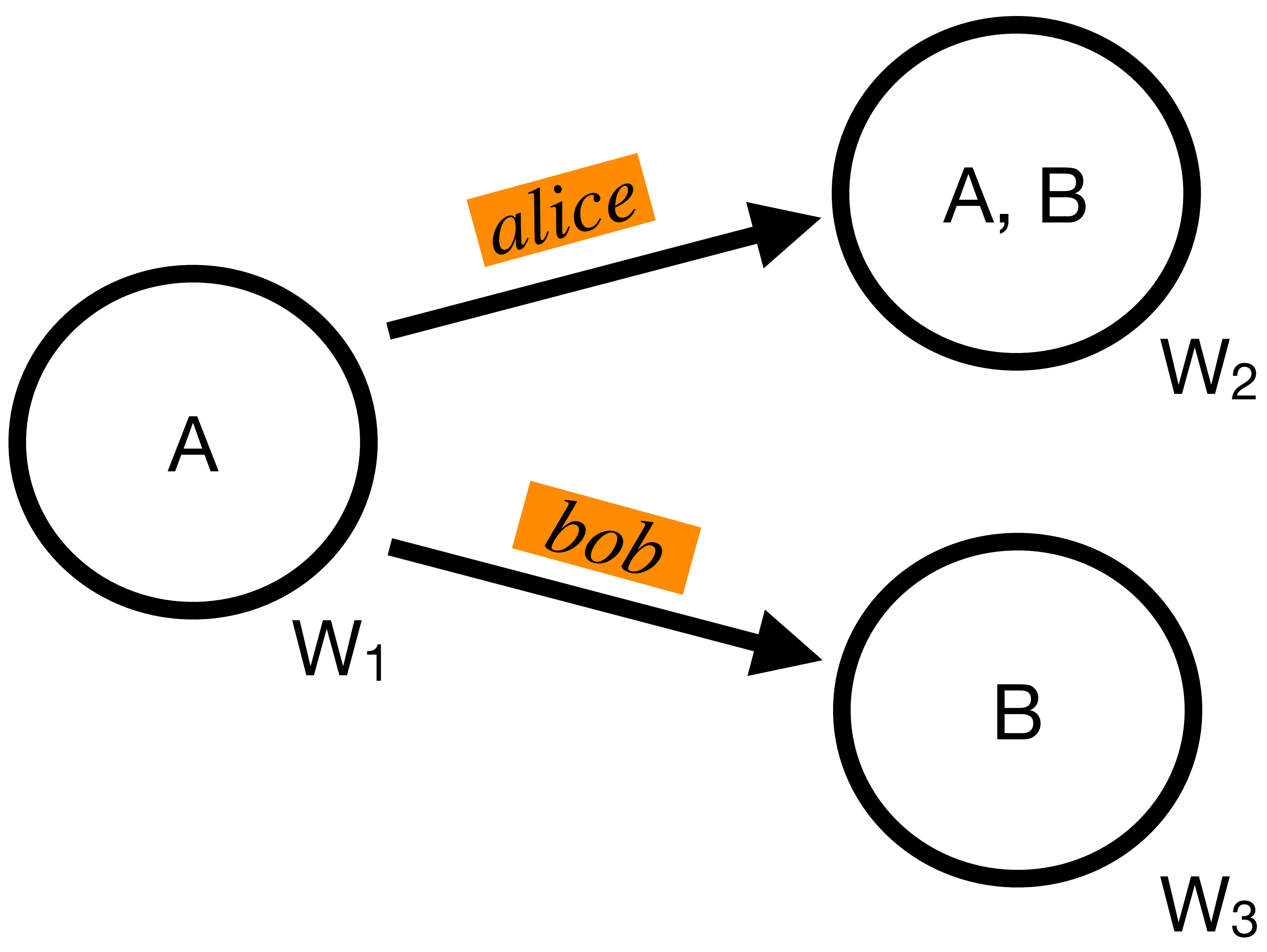
Propositions | Bottom | PL Operators | „Box"-Operator | „Diamond"-Operator

$$A, B := P \in PS \mid \bot \mid \neg A \mid (A \wedge B) \mid (A \vee B) \mid (A \supset B) \mid (A \equiv B) \mid \Box_i A \mid \Diamond_i A$$

With index set I, e.g. $I = \{alice, bob\}$

Sample formula: $M, W_1 \vDash \Box_{alice}(A \wedge B)$ ✓

Semantics: Evaluation relative to possible worlds using Kripke-Structures: $M = (W, \{R_i\}_{i \in I}, V)$



First-Order Multi-Modal Logics (FOML)

First-Order Multi-Modal Logic [5,17]

Syntax:

Predicates

Terms

Quantifier

Variables

$$A, B := P(t_1, \ldots, t_n) \,|\, \neg A \,|\, (A \wedge B) \,|\, (A \vee B) \,|\, (A \supset B) \,|\, (A \equiv B) \,|\, \Box_i A \,|\, \Diamond_i A \,|\, \forall X\, A \,|\, \exists X\, A$$

With index set I, e.g. $I = \{alice, bob\}$

Sample formula: $M, W_1 \vDash \Box_{alice} (P(c) \wedge \forall X\, Q(c, X))$

Semantics: Evaluation relative to possible worlds using Kripke-Structures:
$M = (W, \{R_i\}_{i \in I}, \mathscr{D}, \mathscr{I})$
with $\mathscr{D} = \{D_w\}_{w \in W}$
and $\mathscr{I} = \{I_w\}_{w \in W}$

First-Order Multi-Modal Logics (FOML)

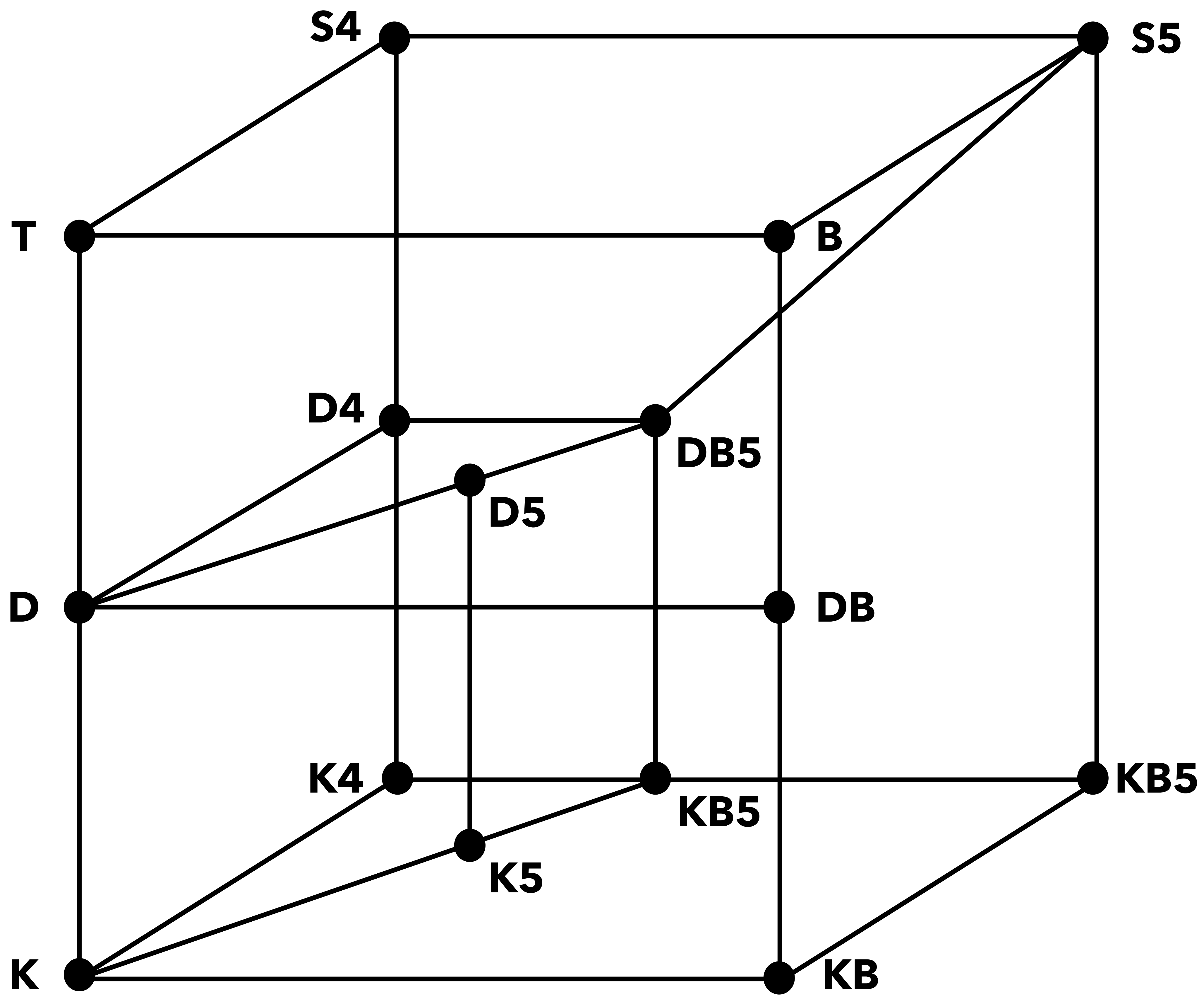
Characterisation [9]

Axiom schemes	
(D)	$\Box A \supset \Diamond A$
(T)	$\Box A \supset A$
(4)	$\Box A \supset \Box\Box A$
(5)	$\Diamond A \supset \Box\Diamond A$
(B)	$A \supset \Box\Diamond A$

First-Order Multi-Modal Logics (FOML)

Characterisation [9]

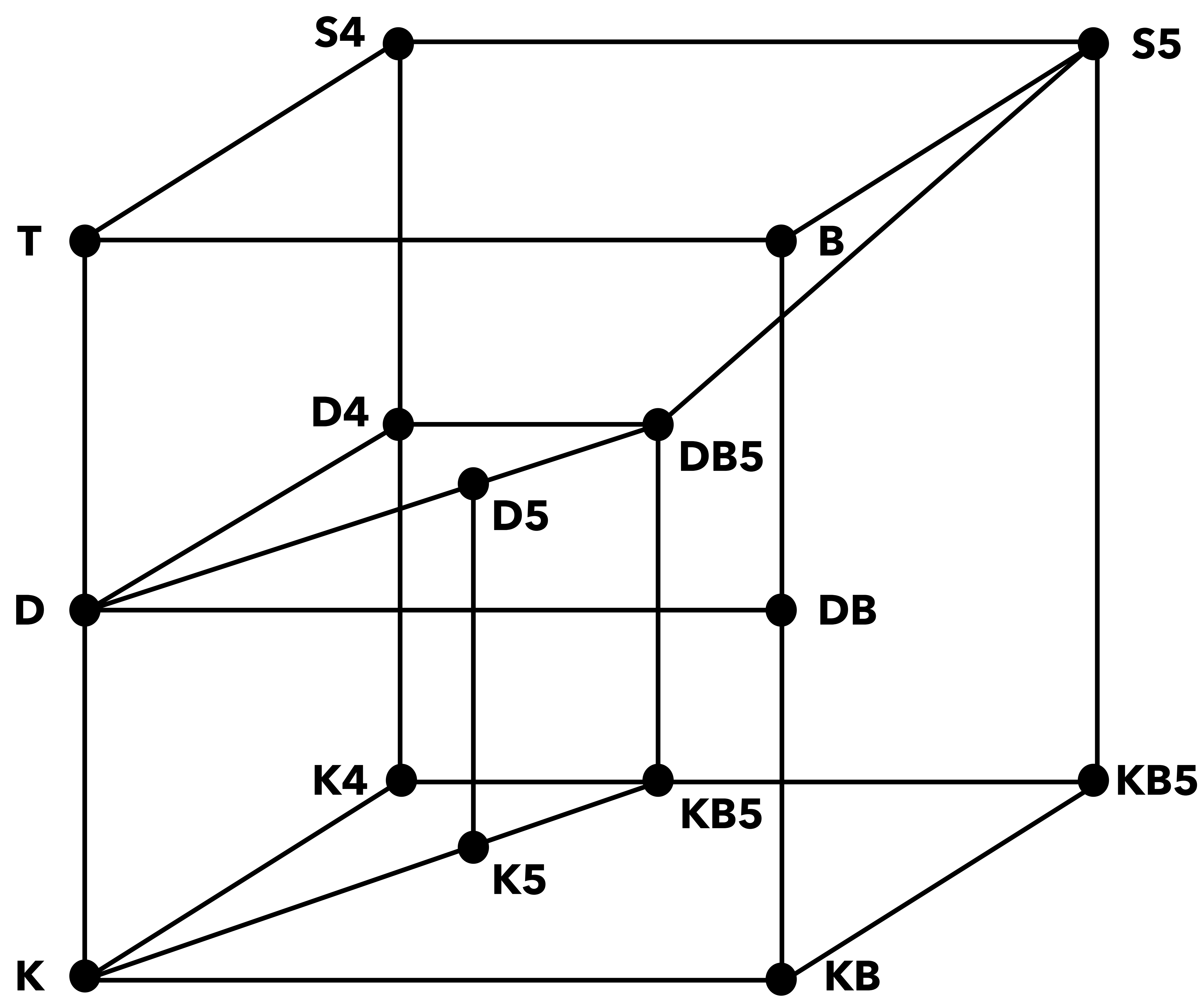
Axiom schemes	
(D)	$\Box A \supset \Diamond A$
(T)	$\Box A \supset A$
(4)	$\Box A \supset \Box\Box A$
(5)	$\Diamond A \supset \Box\Diamond A$
(B)	$A \supset \Box\Diamond A$



First-Order Multi-Modal Logics (FOML)

Characterisation [9]

Frame Properties	Axiom schemes
Serial	$(D)\quad \Box A \supset \Diamond A$
Reflexive	$(T)\quad \Box A \supset A$
Transitive	$(4)\quad \Box A \supset \Box\Box A$
Euclidean	$(5)\quad \Diamond A \supset \Box\Diamond A$
Symmetric	$(B)\quad A \supset \Box\Diamond A$



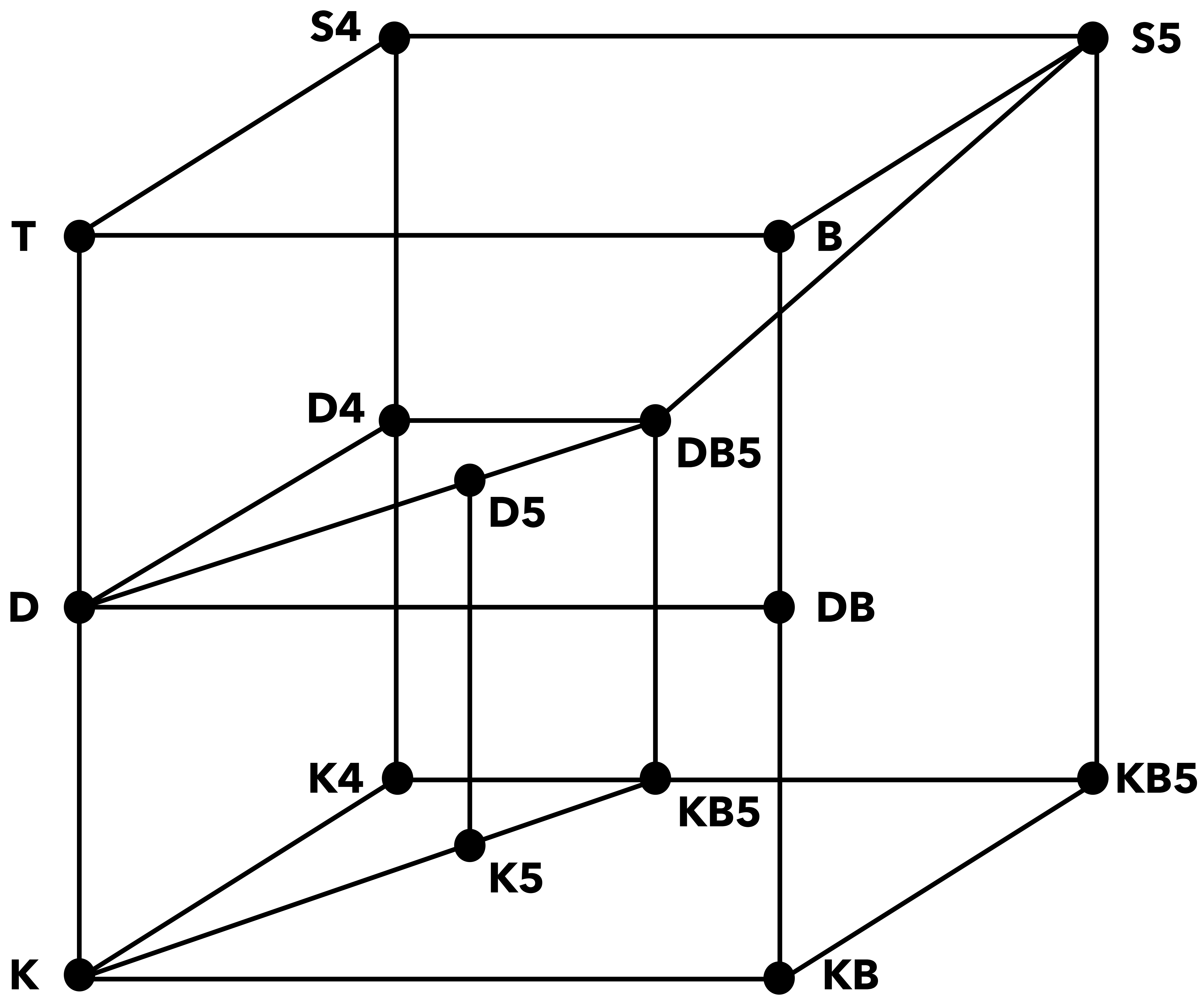
First-Order Multi-Modal Logics (FOML)

Characterisation [9]

Frame Properties	Axiom schemes
Serial	$(D)\quad \Box A \supset \Diamond A$
Reflexive	$(T)\quad \Box A \supset A$
Transitive	$(4)\quad \Box A \supset \Box\Box A$
Euclidean	$(5)\quad \Diamond A \supset \Box\Diamond A$
Symmetric	$(B)\quad A \supset \Box\Diamond A$

Irreflexive
$\neg xRx$

Frame Properties



First-Order Multi-Modal Logics (FOML)

Characterisation [9]

Frame Properties	Axiom schemes	
Serial	(D)	$\Box A \supset \Diamond A$
Reflexive	(T)	$\Box A \supset A$
Transitive	(4)	$\Box A \supset \Box\Box A$
Euclidean	(5)	$\Diamond A \supset \Box\Diamond A$
Symmetric	(B)	$A \supset \Box\Diamond A$

Irreflexive

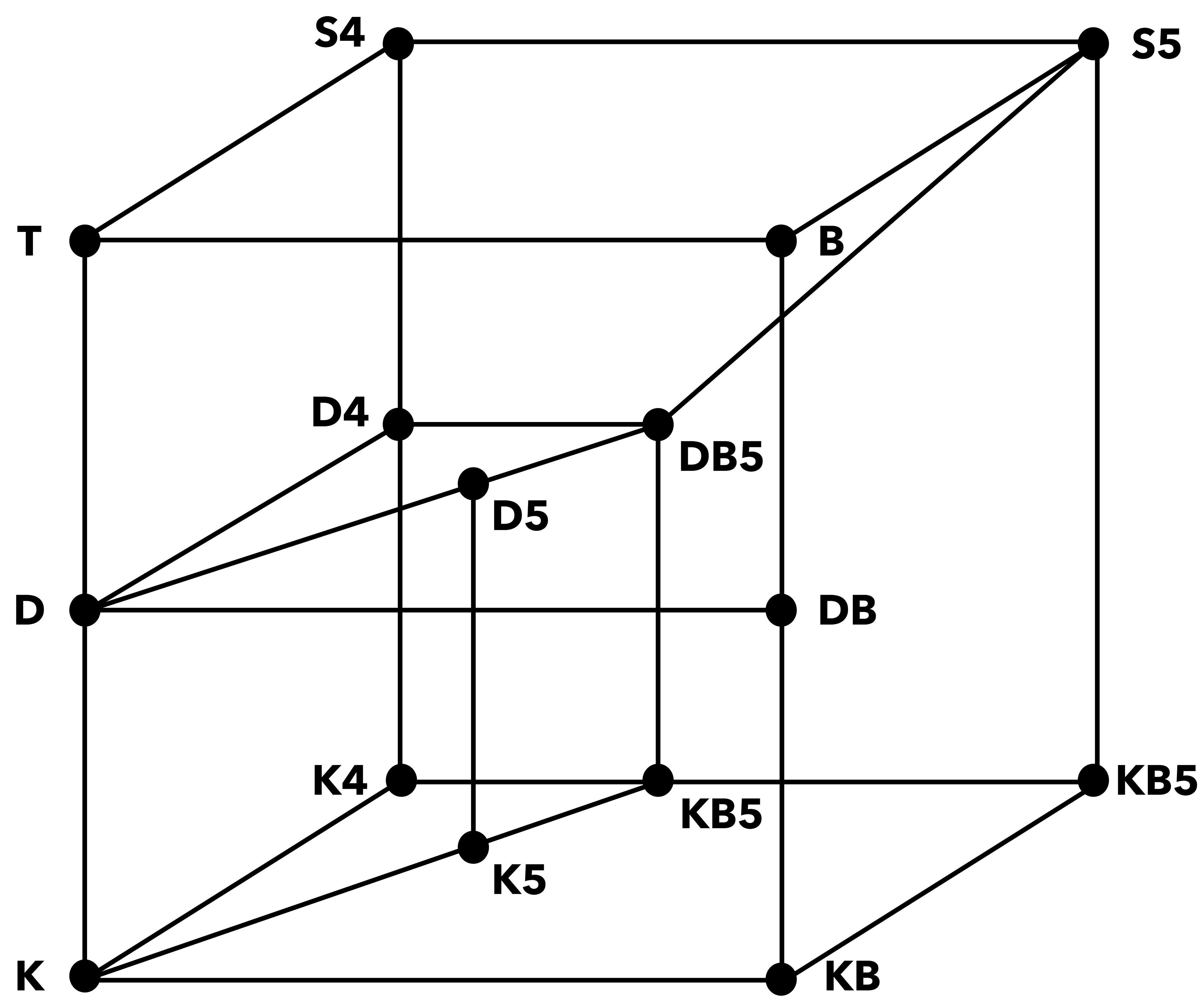
$\neg xRx$

Frame Properties

[11] Axiom Schemes

McKinsey Axiom

$\Box\Diamond A \supset \Diamond\Box A$



First-Order Multi-Modal Logics (FOML)

Characterisation [9]

Frame Properties	Axiom schemes
Serial	$(D)\quad \Box A \supset \Diamond A$
Reflexive	$(T)\quad \Box A \supset A$
Transitive	$(4)\quad \Box A \supset \Box\Box A$
Euclidean	$(5)\quad \Diamond A \supset \Box\Diamond A$
Symmetric	$(B)\quad A \supset \Box\Diamond A$

Irreflexive

$\neg xRx$

Frame Properties

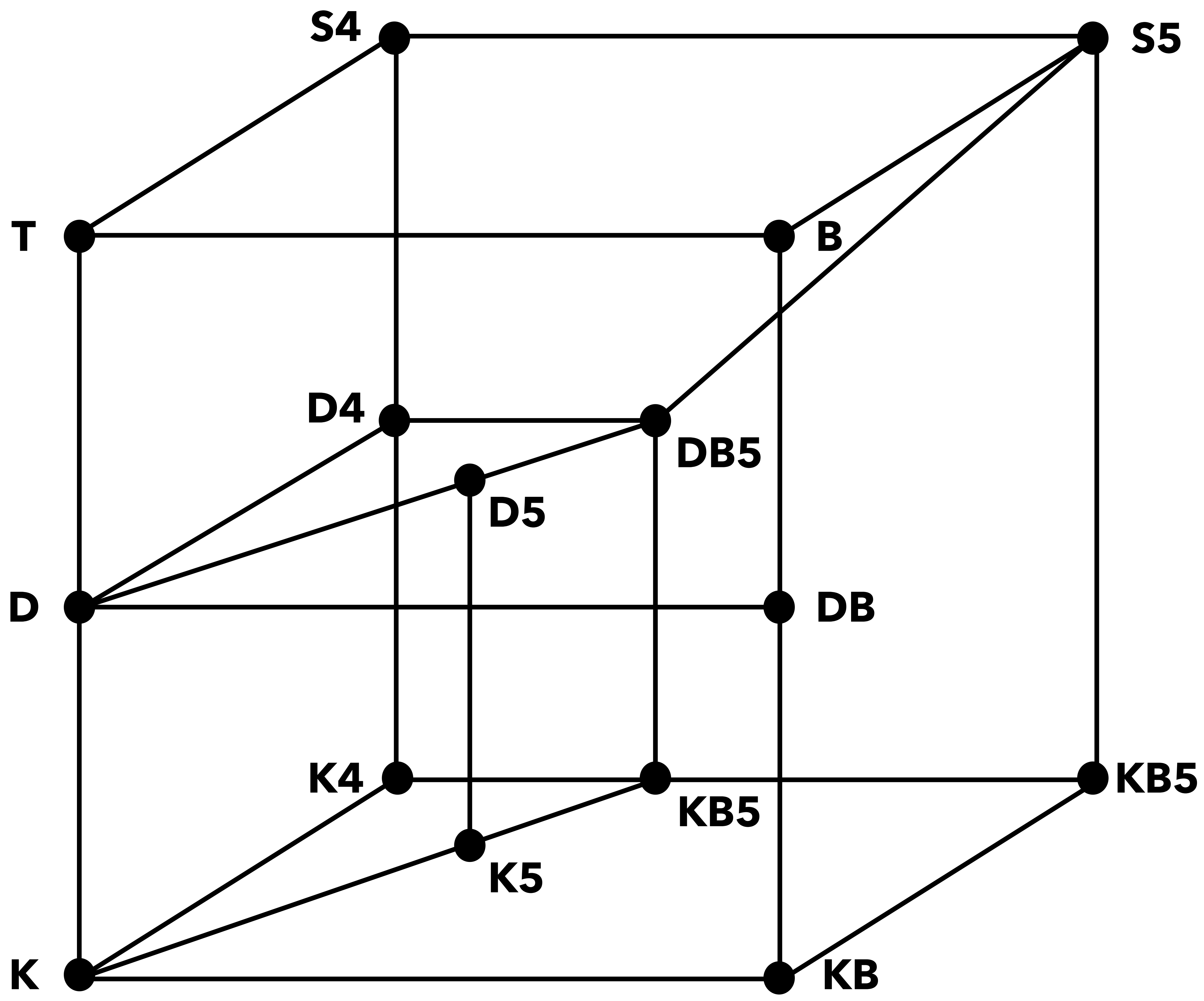
[11] Axiom Schemes

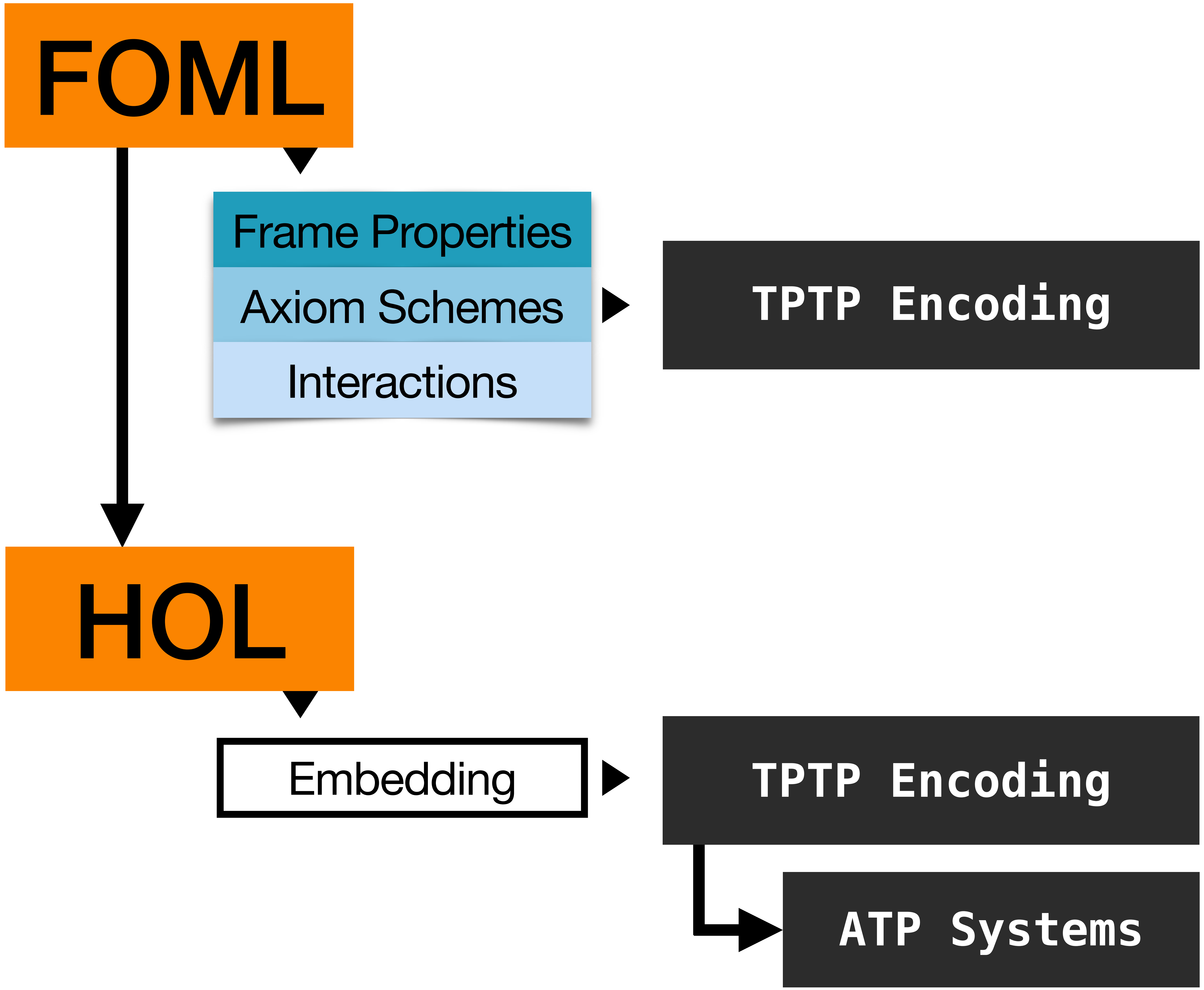
McKinsey Axiom

$\Box\Diamond A \supset \Diamond\Box A$

[8] Interactions

$\Box_1 A \supset \Box_2 A$




FOML
Frame Properties
Axiom Schemes
Interactions
TPTP Encoding
HOL
Embedding
TPTP Encoding
ATP Systems

Higher Order Logic (HOL)
Introduction [4, 6, 12]

Constants

Syntax: *Variables*

$s, t := x_\alpha \in V \,|\, c_\alpha \in \Sigma$ with base types $\mathscr{T} = \{\iota, o\}$

Higher Order Logic (HOL)

Introduction [4, 6, 12]

Constants

Syntax: Variables Abstraction

$$s, t := x_\alpha \in V \,|\, c_\alpha \in \Sigma \,|\, (\lambda x_\alpha \,.\, s_\beta)_{\alpha \to \beta}$$ with base types $\mathcal{T} = \{\iota, o\}$

- Abstractions are unnamed functions that return s_β with each free occurrence of x_α replaced with the argument (ß-reduction)

Higher Order Logic (HOL)
Introduction [4, 6, 12]

Syntax:

Constants

Variables — Abstraction — Application

$$s, t := x_\alpha \in V \,|\, c_\alpha \in \Sigma \,|\, (\lambda x_\alpha \,.\, s_\beta)_{\alpha\to\beta} \,|\, (s_{\alpha\to\beta} t_\alpha)_\beta \qquad \text{with base types } \mathscr{T} = \{\iota, o\}$$

- Abstractions are unnamed functions that return s_β with each free occurrence of x_α replaced with the argument (ß-reduction)

Higher Order Logic (HOL)
Introduction [4, 6, 12]

Syntax:

Constants

Abstraction

Variables

Application

$$s, t := x_\alpha \in V \,|\, c_\alpha \in \Sigma \,|\, (\lambda x_\alpha \,.\, s_\beta)_{\alpha \to \beta} \,|\, (s_{\alpha \to \beta} t_\alpha)_\beta \qquad \text{with base types } \mathcal{T} = \{\iota, o\}$$

- Abstractions are unnamed functions that return s_β with each free occurrence of x_α replaced with the argument (ß-reduction)
- Function types can be constructed using „→“

Higher Order Logic (HOL)

Introduction [4, 6, 12]

Syntax:

Variables *Constants* *Abstraction* *Application*

$$s, t := x_\alpha \in V \,|\, c_\alpha \in \Sigma \,|\, (\lambda x_\alpha \,.\, s_\beta)_{\alpha\to\beta} \,|\, (s_{\alpha\to\beta} t_\alpha)_\beta \qquad \text{with base types } \mathscr{T} = \{\iota, o\}$$

- Abstractions are unnamed functions that return s_β with each free occurrence of x_α replaced with the argument (ß-reduction)
- Function types can be constructed using „→"

Semantics: General semantics of HOL [1] is assumed

Higher Order Logic (HOL)

Introduction

Constants

Syntax: Variables Abstraction Application

$$s, t := x_\alpha \in V \,|\, c_\alpha \in \Sigma \,|\, (\lambda x_\alpha \,.\, s_\beta)_{\alpha\to\beta} \,|\, (s_{\alpha\to\beta} t_\alpha)_\beta$$

with base types $\mathscr{T} = \{\iota, o\}$

Embedding of FOML into HOL [2,3]

- Introduction of new type μ for worlds

Higher Order Logic (HOL)
Introduction

Syntax:

Constants

Variables *Abstraction* *Application*

$s, t := x_\alpha \in V \,|\, c_\alpha \in \Sigma \,|\, (\lambda x_\alpha \,.\, s_\beta)_{\alpha\to\beta} \,|\, (s_{\alpha\to\beta} t_\alpha)_\beta$ with base types $\mathcal{T} = \{\iota, o\}$

Embedding of FOML into HOL [2,3]

- Introduction of new type μ for worlds

- Definition of auxiliary structures representing elements from Kripke semantics, eg. :

$$wR_i v \longrightarrow r^i_{\mu\to\mu\to o} W_\mu V_\mu$$

Higher Order Logic (HOL)
Introduction

Syntax:

Constants *Variables* *Abstraction* *Application*

$s, t := x_\alpha \in V \,|\, c_\alpha \in \Sigma \,|\, (\lambda x_\alpha \,.\, s_\beta)_{\alpha \to \beta} \,|\, (s_{\alpha \to \beta} t_\alpha)_\beta$ with base types $\mathscr{T} = \{\iota, o\}$

Embedding of FOML into HOL [2,3]

- Introduction of new type μ for worlds
- Definition of auxiliary structures representing elements from Kripke semantics, eg. :

$$wR_i v \longrightarrow r^i_{\mu \to \mu \to o} W_\mu V_\mu$$
$$\Box_i A \longrightarrow \big(\lambda X_{\mu \to o} \,.\, \lambda W_\mu \,.\, \forall V_\mu \,.\, \neg (r^i \; W \; V) \vee (X \; V)\big) \; \lceil A \rceil$$

Higher Order Logic (HOL)
Introduction

Syntax:

Constants

Variables

Abstraction

Application

$s, t := x_\alpha \in V \,|\, c_\alpha \in \Sigma \,|\, (\lambda x_\alpha \,.\, s_\beta)_{\alpha\to\beta} \,|\, (s_{\alpha\to\beta} t_\alpha)_\beta$ with base types $\mathscr{T} = \{\iota, o\}$

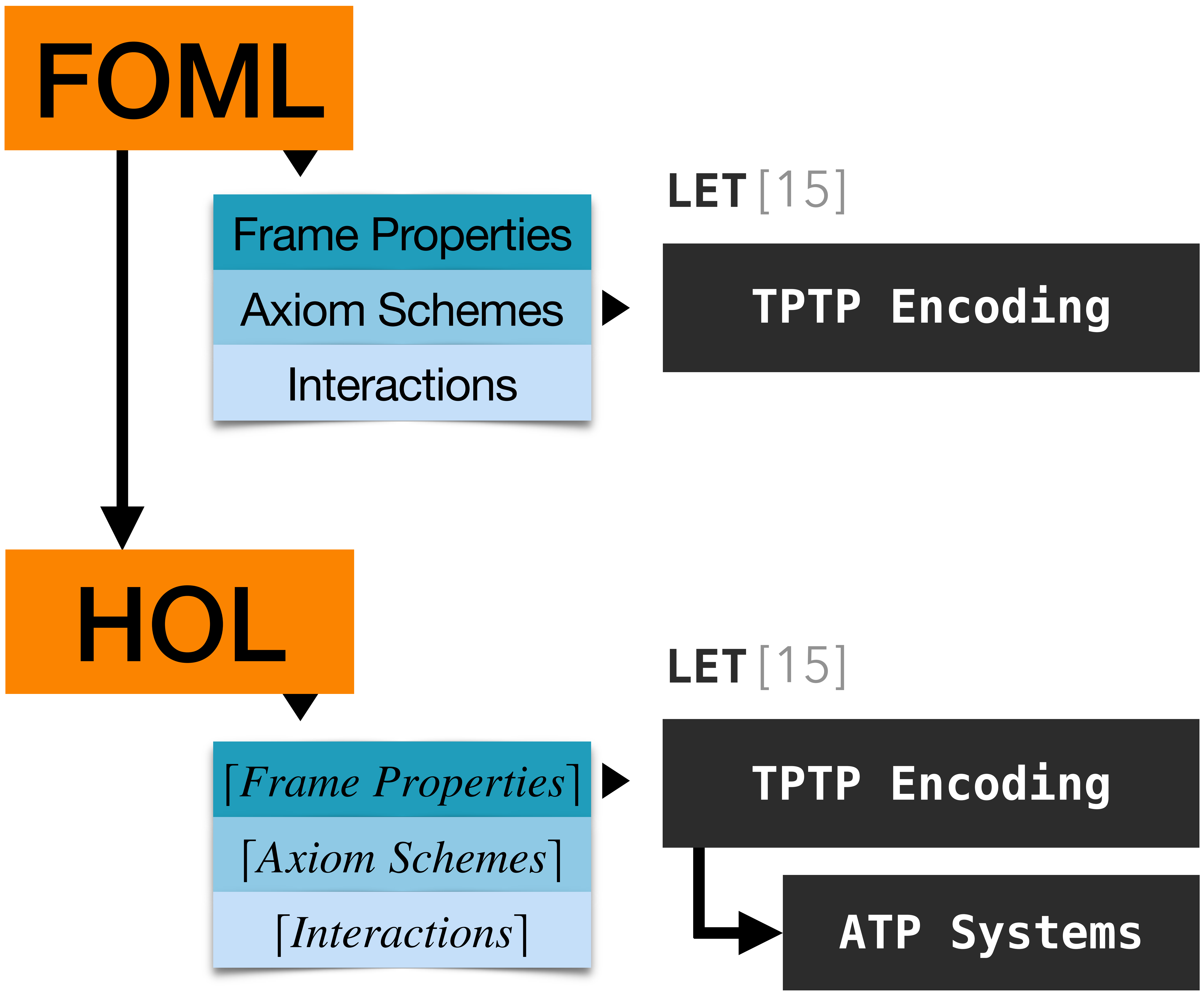
Embedding of FOML into HOL [2,3]

- Introduction of new type μ for worlds
- Definition of auxiliary structures representing elements from Kripke semantics, eg. :

$$wR_i v \longrightarrow r^i_{\mu\to\mu\to o} W_\mu V_\mu$$

$$\Box_i A \longrightarrow \big(\lambda X_{\mu\to o} \,.\, \lambda W_\mu \,.\, \forall V_\mu \,.\, \neg(r^i\ W\ V) \vee (X\ V)\big)\ \lceil A \rceil$$

- Recursive embedding of formulas


FOML
Frame Properties
Axiom Schemes
Interactions
LET [15]
TPTP Encoding
HOL
[Frame Properties]
[Axiom Schemes]
[Interactions]
LET [15]
TPTP Encoding
ATP Systems

Thousands of Problems for Theorem Provers (TPTP)
Introduction [16]

- community standard platform for ATP development
- Several languages for ATP systems, e.g.
  - THF for classical higher-order logic
  - NXF for first-order non-classical reasoning [15]
- Core building block: annotated formulas of form

```
language(name,role,formula).
```

TPTP-Syntax

$\neg xRx$

TPTP-Syntax

$\neg xRx$

```
(~$ki_accessible(X,X))
```

xRx : `$ki_accessible(X,X)`

$\neg A$: `~A`

TPTP-Syntax

$\neg xRx$

```
![X: $ki_world] :
    (~$ki_accessible(X,X))
```

xRx	:	`$ki_accessible(X,X)`
$\neg A$	:	`~A`
$\forall X\ A$	:	`![X:`*type*`]:(A)`

TPTP-Syntax

$\neg xRx$

```
![X: $ki_world] :
   (~$ki_accessible(X,X))
```

$\Box \Diamond A \supset \Diamond \Box A$

```
{$box} @ (                  )
```

$\Box A$: `{$box} @ A`

TPTP-Syntax

$\neg xRx$

```
![X: $ki_world] :
    (~$ki_accessible(X,X))
```

$\Box\Diamond A \supset \Diamond\Box A$

```
{$box} @ ({$dia} @ (A))
```

$\Box A$: `{$box} @ A`

$\Diamond A$: `{$dia} @ A`

TPTP-Syntax

$\neg xRx$

```
![X: $ki_world] :
   (~$ki_accessible(X,X))
```

$\Box \Diamond A \supset \Diamond \Box A$

```
{$box} @ ({$dia} @ (A))
   => {$dia} @ ({$box} @ (A))
```

$\Box A$: `{$box} @ A`

$\Diamond A$: `{$dia} @ A`

$(A \supset B)$: `A => B`

TPTP-Syntax

$$\neg xRx$$

```
![X: $ki_world] :
   (~$ki_accessible(X,X))
```

$$\Box \Diamond A \supset \Diamond \Box A$$

```
{$box} @ ({$dia} @ (A))
   => {$dia} @ ({$box} @ (A))
```

$$\Box_1 A \supset \Box_2 A$$

```
{$box(#1)} @ (A)
   => {$box(#2)} @ (A)
```

TPTP-Syntax

$\neg xRx$

```
![X: $ki_world] :
   (~$ki_accessible(X,X))
```

$\Box \Diamond A \supset \Diamond \Box A$

```
{$box} @ ({$dia} @ (A))
   => {$dia} @ ({$box} @ (A))
```

$\Box_1 A \supset \Box_2 A$

```
{$box(#1)} @ (A)
   => {$box(#2)} @ (A)
```

Logic Specification [15]

```
thf(logic_spec, logic, $modal == [
  $designation == $rigid,
  $domains == $constant,
     $modalities == [
        $modal_system_T,
        {$box(#1)} == 
           [$modal_system_D]
        . . .
        ]]).
```

TPTP-Syntax

$\neg xRx$

```
![X: $ki_world] :
   (~$ki_accessible(X,X))
```

$\Box \Diamond A \supset \Diamond \Box A$

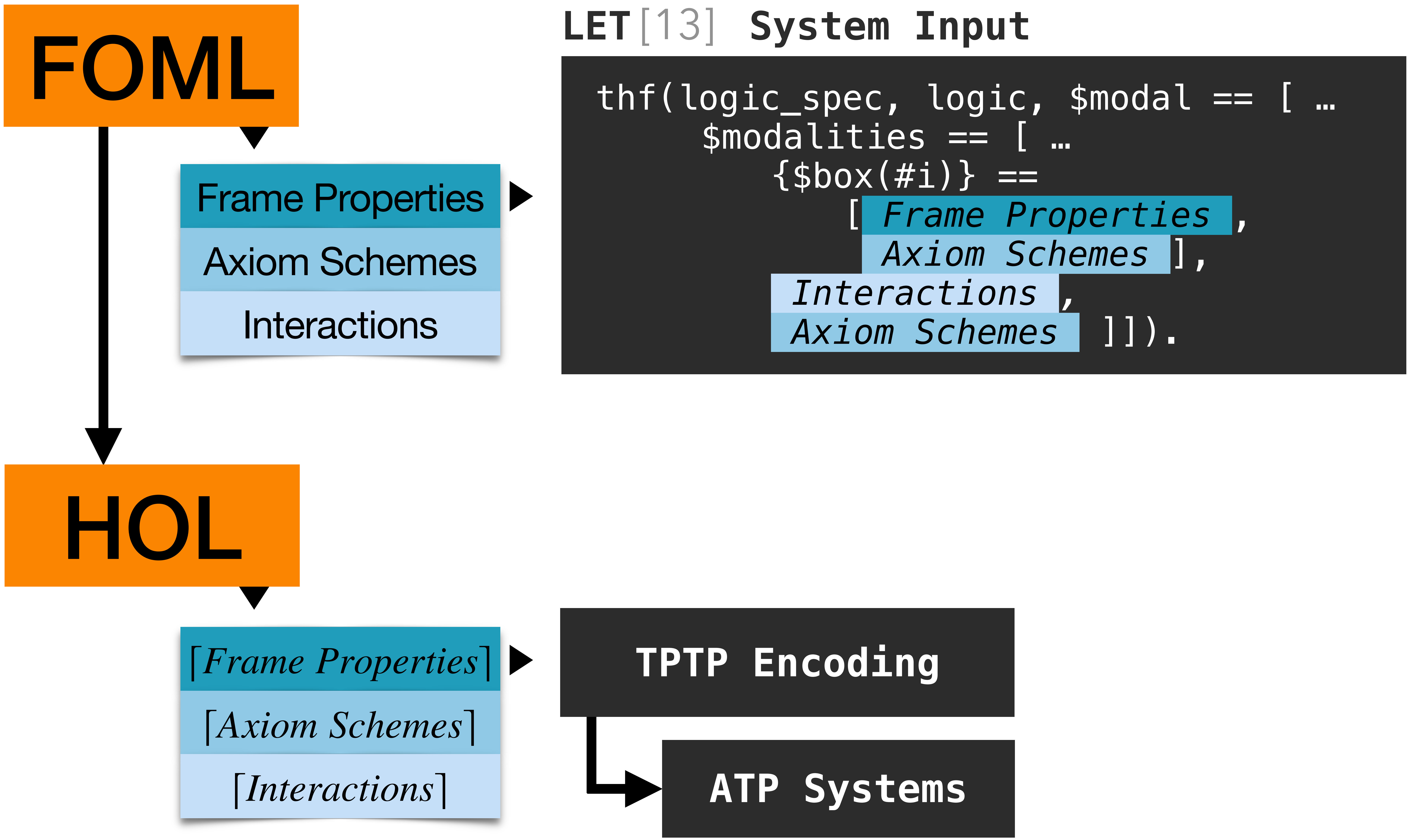
```
{$box} @ ({$dia} @ (A))
   => {$dia} @ ({$box} @ (A))
```

$\Box_1 A \supset \Box_2 A$

```
{$box(#1)} @ (A)
   => {$box(#2)} @ (A)
```

Logic Specification [15]

```
thf(logic_spec, logic, $modal == [
  $designation == $rigid,
  $domains == $constant,
     $modalities == [
        $modal_system_T,
        {$box(#1)} ==
           [$modal_system_D
            ![X: $ki_world] :
             (~$ki_accessible(X,X)),
            {$box} @ ({$dia} @ (A))
             => {$dia} @ ({$box} @ (A))],
        . . .
        {$box(#1)} @ (A)
          => {$box(#2)} @ (A),
        . . .
        ]]).
```


FOML
Frame Properties
Axiom Schemes
Interactions
HOL
[Frame Properties]
[Axiom Schemes]
[Interactions]
LET [13] System Input
thf(logic_spec, logic, $modal == […
$modalities == […
{$box(#i)} ==
[Frame Properties ,
Axiom Schemes],
Interactions ,
Axiom Schemes]]).
TPTP Encoding
ATP Systems

FOML-SPEC

HOL-EMBEDDING

$\neg xRx$

```
![X: $ki_world] :
   (~$ki_accessible(X,X))
```

```
! [X:mworld]:
   ~ (mrel @'#1' @ X @ X )
```

$\Box \Diamond A \supset \Diamond \Box A$

```
{$box} @ ({$dia} @ (A))
   => {$dia} @ ({$box} @ (A))
```

$\Box_1 A \supset \Box_2 A$

```
{$box(#1)} @ (A)
   => {$box(#2)} @ (A)
```

$$\Box_i A \quad \rightarrow \quad \left(\lambda X_{\mu\to o}\,.\,\lambda W_\mu\,.\,\forall V_\mu\,.\,\neg(r^i\ W\ V) \vee (X\ V)\right)\ \lceil A \rceil$$

Embedding

FOML-SPEC | HOL

$$\Box_i A \;\rightarrow\; \big(\lambda X_{\mu\to o}\,.\,\lambda W_\mu\,.\,\forall V_\mu\,.\,\neg(r^i\; W\; V) \vee (X\; V)\big)\; \lceil A \rceil$$

```
{box(#1)} @ (A)
```

[10] →

```
mbox: (mindex > ((mworld > $o) > (mworld > $o)))

 mbox = (^ [R:mindex,Phi:(mworld > $o),W:mworld]:
   ! [V:mworld]: ((mrel @ R @ W @ V)
    => (Phi @ V)))
```

```
mbox @‘#1‘ @ A
```

Embedding

FOML-SPEC | HOL-EMBEDDING

FOML-SPEC

$\neg xRx$

```
![X: $ki_world] :
   (~$ki_accessible(X,X))
```

$\Box\Diamond A \supset \Diamond\Box A$

```
{$box} @ ({$dia} @ (A))
   => {$dia} @ ({$box} @ (A))
```

$\Box_1 A \supset \Box_2 A$

```
{$box(#1)} @ (A)
   => {$box(#2)} @ (A)
```

HOL-EMBEDDING

```
! [X:mworld]:
   ~ (mrel @'#1' @ X @ X )
```

```
 mglobal @ ^[W:mworld]:
((mbox @'#2' @(mdia @,#2' @A )@ W)
=> (mdia @'#2'@(mbox@'#2' @A )@ W))
```

```
 mglobal @ ^ [W:mworld]:
((mbox @ '#1' @ A @ W)
=> (mbox @ '#2' @ A @ W))
```

FOML-SPEC	HOL-EMBEDDING

$\neg xRx$
`![X: $ki_world] :`
`   (~$ki_accessible(X,X))`
	`! [X:mworld]:`
	`   ~ (mrel @'#1' @ X @ X )`

$\Box\Diamond A \supset \Diamond\Box A$
`{$box} @ ({$dia} @ (A))`
`   => {$dia} @ ({$box} @ (A))`
	`! [A:(mworld > $o)]:`
	`   (mglobal @ ^[W:mworld]:`
	`   ((mbox @'#2' @(mdia @,#2' @A )@ W)`
	`   => (mdia @'#2'@(mbox@'#2' @A )@ W)))`

$\Box_1 A \supset \Box_2 A$
`{$box(#1)} @ (A)`
`   => {$box(#2)} @ (A)`
	`! [A:(mworld > $o)]:`
	`   (mglobal @ ^ [W:mworld]:`
	`   ((mbox @ '#1' @ A @ W)`
	`   => (mbox @ '#2' @ A @ W)))`

FOML-SPEC	HOL-EMBEDDING

$\neg xRx$
`![X: $ki_world] :`
`(~$ki_accessible(X,X))`
	`thf(mrel_2_semantic1,axiom,`
	`! [X:mworld]:`
	`~ (mrel @'#1' @ X @ X )).`

$\Box\Diamond A \supset \Diamond\Box A$
`{$box} @ ({$dia} @ (A))`
`=> {$dia} @ ({$box} @ (A))`
	`thf(mrel_2_syntactic1,axiom,`
	`! [A:(mworld > $o)]:`
	`(mglobal @ ^[W:mworld]:`
	`((mbox @'#2' @(mdia @,#2' @A )@ W)`
	`=> (mdia @'#2'@(mbox@'#2' @A )@ W)))).`

$\Box_1 A \supset \Box_2 A$
`{$box(#1)} @ (A)`
`=> {$box(#2)} @ (A)`
	`thf(interaction_scheme_1,axiom,`
	`! [A:(mworld > $o)]:`
	`(mglobal @ ^ [W:mworld]:`
	`((mbox @ '#1' @ A @ W)`
	`=> (mbox @ '#2' @ A @ W)))).`

HOL-EMBEDDING

```
thf(mrel_2_semantic1,axiom,
! [X:mworld]:
   ~ (mrel @'#1' @ X @ X )).
```

```
thf(mrel_2_syntactic1,axiom,
! [A:(mworld > $o)]:
   (mglobal @ ^[W:mworld]:
   ((mbox @'#2' @(mdia @,#2' @A )@ W)
   => (mdia @'#2'@(mbox@'#2' @A )@ W))))).
```

```
thf(interaction_scheme_1,axiom,
! [A:(mworld > $o)]:
   (mglobal @ ^ [W:mworld]:
   ((mbox @ '#1' @ A @ W)
   => (mbox @ '#2' @ A @ W)))).
```

SYSTEM OUTPUT (LET) [13]

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Meta-logical definitions %%%

. . .

. . .

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Converted problem %%%%%%%%%

. . .
```

FOML

Frame Properties
Axiom Schemes
Interactions

HOL

[*Frame Properties*]
[*Axiom Schemes*]
[*Interactions*]

LET [13] **System Input**

```
thf(logic_spec, logic, $modal == [ …
      $modalities == [ …
          {$box(#i)} ==
              [ Frame Properties ,
                Axiom Schemes ],
          Interactions ,
          Axiom Schemes  ]]).
```

LET [13] **System Output**

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Meta-logical definitions %%%
. . .
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Converted problem %%%%%%%%%%
. . .
```

LEO-III [14]

References

[1] Andrews, P.: General models and extensionality. J. Symbolic Logic 37(2), 395–397 (1972)
[2] Benzmüller, C., Paulson, L.: Quantified multimodal logics in simple type theory. Logica Univ. 7(1), 7–20 (2013)
[3] Benzmüller, C., Woltzenlogel Paleo, B.: Higher-order modal logics: automation and applications. In: Faber, W., Paschke, A. (eds.) Reasoning Web 2015. LNCS, vol. 9203, pp. 32–74. Springer, Cham (2015)
[4] Benzmüller, C., Andrews, P.: Church's Type Theory. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University, summer 2019 edn. (2019)
[5] Blackburn, P., van Benthem, J.F.A.K., Wolter, F. (eds.): Handbook of Modal Logic, Studies in logic and practical reasoning, vol. 3. North-Holland (2007)
[6] Church, A.: A formulation of the simple theory of types. J. Symbolic Logic 5, 56–68 (1940)
[7] Fitting, M., Mendelsohn, R.: First-Order Modal Logic. Kluwer (1998)
[8] Gabbay, D.M., Hogger, C.J., Robinson, J.A. (eds.): Handbook of Logic in Artificial Intelligence and Logic Programming (Vol. 4): Epistemic and Temporal Reasoning. Oxford University Press Inc, USA (1995)
[9] Garson, J.: Modal Logic. In: Zalta, E. (ed.) Stanford Encyclopedia of Philosophy. Stanford University (2018)
[10] Gleißner, T., Steen, A., Benzmüller, C.: Theorem provers for every normal modal logic. In: Eiter, T., Sands, D. (eds.) LPAR-21. 21st International Conference on Logic for Programming, Artificial Intelligence and Reasoning. EPiC Series in Computing, vol. 46, pp. 14–30. EasyChair (2017).
[11] Goldblatt, R.: The mckinsey axiom is not canonical. J. Symbolic Logic 56(2), 554–562 (1991)
[12] Henkin, L.: Completeness in the theory of types. J. Symbolic Logic 15(2), 81–91 (1950)
[13] Steen, A.: An Extensible Logic Embedding Tool for Lightweight Non-Classical Reasoning. In: PAAR@IJCAR. CEUR Workshop Proceedings, vol. 3201. CEUR-WS.org (2022)
[14] Steen, A., Benzmüller, C.: Extensional higher-order paramodulation in Leo-III. J. Autom. Reason. 65(6), 775–807 (2021)
[15] Steen, A., Fuenmayor, D., Gleißner, T., Sutcliffe, G., Benzmüller, C.: Automated reasoning in non-classical logics in the TPTP world. In: PAAR@IJCAR. CEUR Workshop Proceedings, vol. 3201. CEUR-WS.org (2022)
[16] Sutcliffe, G.: The TPTP problem library and associated infrastructure. From CNF to TH0, TPTP v6.4.0. J. Autom. Reasoning 59(4), 483–502 (2017)

Application Example
Simplified Shooting Problem [A1]

- The logic is defined as follows:

$$\begin{array}{rl} T: & \Box_{always} A \supset A \\ 4: & \Box_{always} A \supset \Box_{always} \Box_{always} A \\ B_1: & \Box_{always} A \supset \Box_{load} A \\ B_2: & \Box_{always} A \supset \Box_{shoot} A \end{array}$$

[A1] Baldoni, M.: Normal multimodal logics: automatic deduction and logic program- ming extension. Ph.D. thesis, Università degli Studi di Torino, Dipartimento di Informatica (1998)

Application Example

Simplified Shooting Problem [A1]

- The logic is defined as follows:

$$
\begin{array}{ll}
T: & \Box_{always} A \supset A \\
4: & \Box_{always} A \supset \Box_{always} \Box_{always} A \\
B_1: & \Box_{always} A \supset \Box_{load} A \\
B_2: & \Box_{always} A \supset \Box_{shoot} A
\end{array}
$$

- The reasoning problem:

$$
\begin{array}{ll}
1: & \Box_{always} \Box_{load}\, loaded \\
2: & \Box_{always} \left(loaded \supset \Box_{shoot} \neg alive \right) \\
C: & \Box_{load} \Box_{shoot} \neg alive
\end{array}
$$

[A1] Baldoni, M.: Normal multimodal logics: automatic deduction and logic program- ming extension. Ph.D. thesis, Università degli Studi di Torino, Dipartimento di Informatica (1998)

Application Example

Simplified Shooting Problem [A1]

- The logic is defined as follows:

$T:$	$\Box_{always} A \supset A$
$4:$	$\Box_{always} A \supset \Box_{always} \Box_{always} A$
$B_1:$	$\Box_{always} A \supset \Box_{load} A$
$B_2:$	$\Box_{always} A \supset \Box_{shoot} A$

- The reasoning problem:

$$1: \quad \Box_{always} \Box_{load}\, loaded$$

$$2: \quad \Box_{always} \left(loaded \supset \Box_{shoot} \neg alive \right)$$

$$C: \quad \Box_{load} \Box_{shoot} \neg alive$$

QMLTP Version [A2]

Attempt at including B_1 as regular axioms:

$$\Box_{always}\, loaded \supset \Box_{load}\, loaded$$

$$\Box_{always} \neg loaded \supset \Box_{load} \neg loaded$$

$$\Box_{always}\, alive \supset \Box_{load}\, alive$$

$$\Box_{always} \neg alive \supset \Box_{load} \neg alive$$

➡ Does not result in a provable reasoning problem

[A1] Baldoni, M.: Normal multimodal logics: automatic deduction and logic program- ming extension. Ph.D. thesis, Università degli Studi di Torino, Dipartimento di Informatica (1998)
[A2] Raths, T., Otten, J.: The QMLTP problem library for first-order modal logics. In: Gramlich, B., Miller, D., Sattler, U. (eds.) IJCAR 2012. LNCS (LNAI), vol. 7364, pp. 454–461. Springer, Heidelberg (2012).

Application Example

Simplified Shooting Problem: LET Version

```
tff(modal_system, logic,
    $modal ==
      [ $modalities == [
          {$box(#always)} == [$modal_axiom_T, $modal_axiom_4],
          {$box(#load)} == $modal_system_K,
          {$box(#shoot)} == $modal_system_K,
                                {$box(#always)} @ (P) => {$box(#load)} @ (P),
                                {$box(#always)} @ (P) => {$box(#shoot)} @ (P) ] ] ).

tff(alive_decl, type, alive: $o ).
tff(loaded_decl, type, loaded: $o ).

tff(axiom_1,hypothesis, {$box(#always)} @ ({$box(#load)} @ (loaded)) ).
tff(axiom_2,hypothesis, {$box(#always)} @ (loaded => ( {$box(#shoot)} @ (~
alive) )) ).

tff(conj, conjecture, {$box(#load)} @ ({$box(#shoot)} @ (~ alive)) ).
```

Application Example
Simplified Shooting Problem: LET Version

```
tff(modal_system, logic,
    $modal ==
      [ $modalities == [
          {$box(#always)} == [$modal_axiom_T, $modal_axiom_4],
          {$box(#load)} == $modal_system_K,
          {$box(#shoot)} == $modal_system_K,
                                 {$box(#always)} @ (P) => {$box(#load)} @ (P),
                                 {$box(#always)} @ (P) => {$box(#shoot)} @ (P) ] ] ).

tff(alive_decl, type, alive: $o ).
tff(loaded_decl, type, loaded: $o ).

tff(axiom_1,hypothesis, {$box(#always)} @ ({$box(#load)} @ (loaded)) ).
tff(axiom_2,hypothesis, {$box(#always)} @ (loaded => ( {$box(#shoot)} @ (~
alive) )) ).

tff(conj, conjecture, {$box(#load)} @ ({$box(#shoot)} @ (~ alive)) ).
```

➡ Proven by LEO-III