Knowledge in Norms: A Sketch

Jesse Hughes

jesseh@cs.kun.nl

University of Nijmegen

Aims

Some initial goals:

- Express: "A function of o is f."

Aims

Some initial goals:

- Express: "A function of o is f." (in terms of user plans U)

Aims

Some initial goals:

- Express: "A function of o is f." (in terms of user plans U)
- Require: "If x executes U, then f ought to attain."

Aims

Some initial goals:

- Express: "A function of o is f." (in terms of user plans U)
- Require: "If x executes U, then f ought to attain."
- Express: "x knows the plan U."

Aims

Some initial goals:

- Express: "A function of o is f." (in terms of user plans U)
- Require: "If x executes U, then f ought to attain."
- Express: "x knows the plan U."
- Require: "For x to execute U, x must know U."

Basic ingredients

Constants: Users $(x, y, z, \ldots \in User)$

Basic ingredients

Constants: Users $(x, y, z, \ldots \in User)$

Artifacts $(o, p, q, \ldots \in Art)$

Basic ingredients

Constants:	Users	$(x, y, z, \ldots \in \mathit{User})$
	Artifacts	$(o, p, q, \ldots \in \mathit{Art})$
Relations:	Atomic	$(R, S, T, \ldots \in \mathit{Atom})$

Basic ingredients

Constants:	Users	$(x, y, z, \ldots \in User)$
	Artifacts	$(o, p, q, \ldots \in Art)$
Relations:	Atomic	$(R, S, T, \ldots \in Atom)$

Relations come with types

$$User \times User \times \ldots \times User \times Art \times Art \times \ldots \times Art\,.$$

Basic ingredients

Constants:	Users	$(x, y, z, \ldots \in \mathit{User})$
	Artifacts	$(o, p, q, \ldots \in \mathit{Art})$
Relations:	Atomic	$(R, S, T, \ldots \in \mathit{Atom})$

Relations come with types

$$\mathit{User} \times \mathit{User} \times \ldots \times \mathit{User} \times \mathit{Art} \times \mathit{Art} \times \ldots \times \mathit{Art}\,.$$

Let **Q** be first-order logic built on these ingredients.

Basic ingredients

Constants:	Users	$(x, y, z, \ldots \in \mathit{User})$
	Artifacts	$(o, p, q, \ldots \in \mathit{Art})$
Relations:	Atomic	$(R, S, T, \ldots \in \mathit{Atom})$

Relations come with types

$$\mathit{User} \times \mathit{User} \times \ldots \times \mathit{User} \times \mathit{Art} \times \mathit{Art} \times \ldots \times \mathit{Art}\,.$$

Let **Q** be first-order logic built on these ingredients.

To **Q**, we add a deontic operator $\bigcirc$, obtaining $\mathbf{QD}^*$.

Basic ingredients

Constants:	Users	$(x, y, z, \ldots \in \mathit{User})$
	Artifacts	$(o, p, q, \ldots \in \mathit{Art})$
Relations:	Atomic	$(R, S, T, \ldots \in \mathit{Atom})$

Relations come with types

$$\mathit{User} \times \mathit{User} \times \ldots \times \mathit{User} \times \mathit{Art} \times \mathit{Art} \times \ldots \times \mathit{Art}\,.$$

Let **Q** be first-order logic built on these ingredients.

To **Q**, we add a deontic operator $\bigcirc$, obtaining $\mathbf{QD}^*$.

The logic $\mathbf{QD}^*$ includes the constant domain assumption.

Functions and plans

Houkes: Knowledge of a user plan is necessary and sufficient evidence of knowledge of artifact function.

Functions and plans

Houkes: Knowledge of a user plan is necessary and sufficient evidence of knowledge of artifact function.

This suggests that we identify user plans and functions.

Functions and plans

Houkes: Knowledge of a user plan is necessary and sufficient evidence of knowledge of artifact function.

This suggests that we identify user plans and functions.

But: functions are goal-directed.

Functions and plans

Houkes: Knowledge of a user plan is necessary and sufficient evidence of knowledge of artifact function.

This suggests that we identify user plans and functions.

But: functions are goal-directed.

We must represent the end of a user plan.

Functions and plans

Houkes: Knowledge of a user plan is necessary and sufficient evidence of knowledge of artifact function.

This suggests that we identify user plans and functions.

But: functions are goal-directed.

We must represent the end of a user plan.

A goal is a state of affairs, a condition of the world . . .

Functions and plans

Houkes: Knowledge of a user plan is necessary and sufficient evidence of knowledge of artifact function.

This suggests that we identify user plans and functions.

But: functions are goal-directed.

We must represent the end of a user plan.

A goal is a state of affairs, a condition of the world … a formula of Q!

Abstract user plans

We augment $\mathbf{QD}^*$ with a new type $Plan$ (variables $U, U', \ldots$).

Abstract user plans

We augment $\mathbf{QD}^*$ with a new type $Plan$ (variables $U, U', \ldots$).
Each plan involves an artifact and an end.

Abstract user plans

We augment $\mathbf{QD}^*$ with a new type $Plan$ (variables $U, U', \ldots$).
Each plan involves an artifact and an end.

$$\mathsf{obj} : Plan \longrightarrow Art$$
$$\mathsf{end} : Plan \longrightarrow \mathbf{Q}$$

Abstract user plans

We augment $\mathbf{QD}^*$ with a new type $Plan$ (variables $U, U', \ldots$).
Each plan involves an artifact and an end.

$$\mathsf{obj} : Plan \longrightarrow Art$$
$$\mathsf{end} : Plan \longrightarrow \mathbf{Q}$$

Admittedly, this type looks a bit funky.

Abstract user plans

We augment $\mathbf{QD}^*$ with a new type $Plan$ (variables $U, U', \ldots$).
Each plan involves an artifact and an end.

$$\mathsf{obj} : Plan \longrightarrow Art$$

$$\mathsf{end} : Plan \longrightarrow \mathbf{Q}$$

We could also add preconditions to a user plan, as in dynamic logic.

$$\mathsf{pre} : Plan \longrightarrow \mathbf{Q}$$

Applications of Plans

Users apply user plans to achieve ends.

Applications of Plans

Users apply user plans to achieve ends.

Applications of plans (ought to) change the world.

Applications of Plans

Users apply user plans to achieve ends.

Applications of plans (ought to) change the world.

Application provides a transition structure on our set of worlds.

$$\mathsf{app}\_(-,-) : World \times User \times Plan \longrightarrow World$$

Applications of Plans

Users apply user plans to achieve ends.

Applications of plans (ought to) change the world.

Application provides a transition structure on our set of worlds.

$$\mathsf{app}\_(-,-) : World \times User \times Plan \longrightarrow World$$

$\mathsf{app}_w(x, U)$ is the world resulting from user x applying plan U in world w.

Applications of Plans

Users apply user plans to achieve ends.

Applications of plans (ought to) change the world.

Application provides a transition structure on our set of worlds.

$$\mathsf{app}_{\_}(-,-): World \times User \times Plan \longrightarrow World$$

$\mathsf{app}_w(x, U)$ is the world resulting from user x applying plan U in world w.

Assumes: every user can execute every plan.

Alternative transitions

$$\mathsf{app\_}(-,-) : \mathit{World} \times \mathit{User} \times \mathit{Plan} \longrightarrow ???$$

Type	Assumptions
World	Every user can perform every plan; deterministic

Alternative transitions

$$\mathsf{app\_}(-,-) : World \times User \times Plan \longrightarrow ???$$

Type	Assumptions
$World$	Every user can perform every plan; deterministic
$1 + World$	Users may not perform certain plans; deterministic

Alternative transitions

$$\mathsf{app}\_(-,-) : World \times User \times Plan \longrightarrow ???$$

Type	Assumptions
$World$	Every user can perform every plan; deterministic
$1 + World$	Users may not perform certain plans; deterministic
$\mathcal{P}(World)$	Users may not perform certain plans; non-deterministic

Alternative transitions

$$\mathsf{app\_}(-,-) : World \times User \times Plan \longrightarrow ???$$

Type	Assumptions
$World$	Every user can perform every plan; deterministic
$1 + World$	Users may not perform certain plans; deterministic
$\mathcal{P}(World)$	Users may not perform certain plans; non-deterministic

Note: this perspective on applications of plans is fundamentally coalgebraic!

A dynamic logic for applications

For each pair $x \in User$, $U \in Plan$, we add a modal operator $[x, U]$.

A dynamic logic for applications

For each pair $x \in User$, $U \in Plan$, we add a modal operator $[x, U]$.

$w \models [x, U]\varphi \quad \Leftrightarrow \quad$ for all $w' \in \mathsf{app}_w(x, U)$, we have $w' \models \varphi$.

A dynamic logic for applications

For each pair $x \in User$, $U \in Plan$, we add a modal operator $[x, U]$.

$w \models [x, U]\varphi \quad \Leftrightarrow \quad$ After x applies U in w, the formula φ attains.

A dynamic logic for applications

For each pair $x \in User$, $U \in Plan$, we add a modal operator $[x, U]$.

$w \models [x, U]\varphi \quad \Leftrightarrow \quad$ After x applies U in w, the formula φ attains.

Example:

$$[x, U] \bigcirc \mathsf{end}(U).$$

A dynamic logic for applications

For each pair $x \in User$, $U \in Plan$, we add a modal operator $[x, U]$.

$w \models [x, U]\varphi \quad \Leftrightarrow \quad$ After x applies U in w, the formula φ attains.

Example:

$$[x, U] \bigcirc \mathsf{end}(U).$$

After x applies U, the end of U ought to hold.

A dynamic logic for applications

For each pair $x \in User$, $U \in Plan$, we add a modal operator $[x, U]$.

$w \models [x, U]\varphi \quad \Leftrightarrow \quad$ After x applies U in w, the formula φ attains.

Example:

$$[x, U] \bigcirc \mathsf{end}(U).$$

After x applies U, the end of U ought to hold.

Compare: $\bigcirc [x, U]\, \mathsf{end}(U)$

Knowledge operator?

Do we need an epistemic operator?

Knowledge operator?

Do we need an epistemic operator?

At first glance, it appears not.

Knowledge operator?

Do we need an epistemic operator?

At first glance, it appears not.

We want to express "x knows the plan U."

Knowledge operator?

Do we need an epistemic operator?

At first glance, it appears not.

We want to express "x knows the plan U."

But, U is not a formula. It is a term in our language.

Knowledge operator?

Do we need an epistemic operator?

At first glance, it appears not.

We want to express "x knows the plan U."

But, U is not a formula. It is a term in our language.

Thus, we may as well use a relation to express this.

Knowledge operator?

Do we need an epistemic operator?

At first glance, it appears not.

We want to express "x knows the plan U."

But, U is not a formula. It is a term in our language.

Thus, we may as well use a relation to express this.

Introduce: $\mathsf{groks} : \mathit{User} \times \mathit{Plan}$.

Knowledge operator?

Do we need an epistemic operator?

At first glance, it appears not.

We want to express "x knows the plan U."

But, U is not a formula. It is a term in our language.

Thus, we may as well use a relation to express this.

Introduce: $\mathsf{groks} : \mathit{User} \times \mathit{Plan}$.

Epistemic operators can be added as the situation requires, of course.

The total sketch

We have:

- a basic logic **Q** for describing the worlds;

The total sketch

We have:

- a basic logic $\mathbf{Q}$ for describing the worlds;
- an extension $\mathbf{QD}^*$ of $\mathbf{Q}$ for ought-statements;

The total sketch

We have:

- a basic logic $\mathbf{Q}$ for describing the worlds;
- an extension $\mathbf{QD}^*$ of $\mathbf{Q}$ for ought-statements;
- an extension $\mathbf{QD}^* + \mathbf{DL}$ for statements involving plan execution;

The total sketch

We have:

- a basic logic $\mathbf{Q}$ for describing the worlds;
- an extension $\mathbf{QD}^*$ of $\mathbf{Q}$ for ought-statements;
- an extension $\mathbf{QD}^* + \mathbf{DL}$ for statements involving plan execution;
- a relation groks for expressing whether a user knows a plan.

The total sketch

We have:

- a basic logic $\mathbf{Q}$ for describing the worlds;
- an extension $\mathbf{QD}^*$ of $\mathbf{Q}$ for ought-statements;
- an extension $\mathbf{QD}^* + \mathbf{DL}$ for statements involving plan execution;
- a relation groks for expressing whether a user knows a plan.

With this starting point, one can work to represent functional knowledge.

How to proceed

Further development requires:

- Philosophical resources on functional knowledge;

How to proceed

Further development requires:

- Philosophical resources on functional knowledge;
- Further research in “multi-dimensional” modal logic.

How to proceed

Further development requires:

- Philosophical resources on functional knowledge;
- Further research in “multi-dimensional” modal logic.

Clearly, these tasks must go hand-in-hand.

How to proceed

Concrete steps:

- Clarify the logic $\mathbf{QD}^* + \mathbf{DL}$!

How to proceed

Concrete steps:

- Clarify the logic $\mathbf{QD}^* + \mathbf{DL}$!
- Incorporate proper functions.

How to proceed

Concrete steps:

- Clarify the logic $\mathbf{QD}^* + \mathbf{DL}$!
- Incorporate proper functions.
  - Represent “designer”, “proper”, etc.

How to proceed

Concrete steps:

- Clarify the logic $\mathbf{QD}^* + \mathbf{DL}$!
- Incorporate proper functions.
  - Represent "designer", "proper", etc.
  - Norms for proper function.

How to proceed

Concrete steps:

- Clarify the logic $\mathbf{QD}^* + \mathbf{DL}$!
- Incorporate proper functions.
  - Represent “designer”, “proper”, etc.
  - Norms for proper function.
- Include epistemic operator for practical reasoning?