Means-End Relations and Artifactual Functions

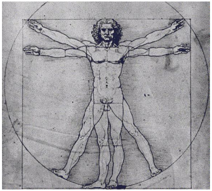
A Sketch

Jesse Hughes

Technical University of Eindhoven

June 4, 2005

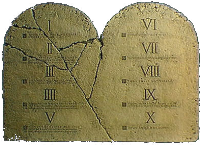
Introduction to *Norms in Knowledge*



Epistemology:

- Knowledge of descriptive claims

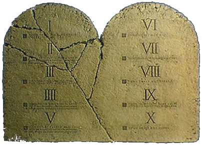
Introduction to *Norms in Knowledge*



Epistemology:

- Knowledge of descriptive claims
- Knowledge of normative claims

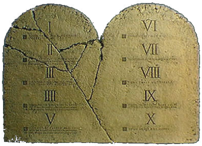
Introduction to *Norms in Knowledge*



Epistemology:

- Knowledge of descriptive claims
- Knowledge of normative claims
    - Non-moral

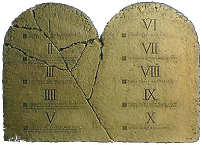
Introduction to *Norms in Knowledge*



Epistemology:

- Knowledge of descriptive claims
- Knowledge of normative claims
  - Non-moral
    - Prescriptive — ought to do

Introduction to *Norms in Knowledge*



Epistemology:

- Knowledge of descriptive claims
- Knowledge of normative claims
  - Non-moral
    - Prescriptive — ought to do
    - Functional — things ought to do

Introduction to *Norms in Knowledge*

Applied to technical artifacts:

- Knowledge of normative claims
  - Non-moral
    - Prescriptive — ought to do
    - Functional — things ought to do

Introduction to *Norms in Knowledge*

Applied to technical artifacts:

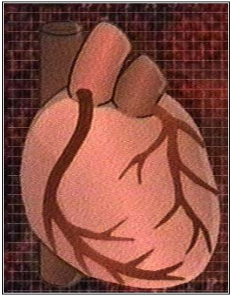
- Knowledge of normative claims
  - Non-moral
    - Prescriptive — ought to do
      Artifacts: HOWTOs
    - Functional — things ought to do

Introduction to *Norms in Knowledge*

Applied to technical artifacts:

- Knowledge of normative claims
  - Non-moral
    - Prescriptive — ought to do
      Artifacts: HOWTOs
    - Functional — things ought to do
      Artifacts: artifactual functions

Some examples of functional ascriptions

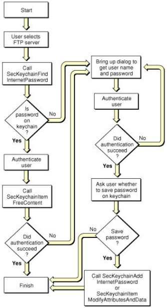


- "The function of the heart is to pump blood."

Some examples of functional ascriptions

- "The function of the heart is to pump blood."
- "That switch mutes the television."

Some examples of functional ascriptions

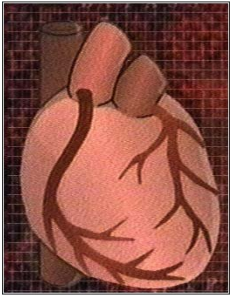


- "The function of the heart is to pump blood."
- "That switch mutes the television."
- "The subroutine ensures that the user is authorized."

Some examples of functional ascriptions

- "The function of the heart is to pump blood."
- "That switch mutes the television."
- "The subroutine ensures that the user is authorized."
- "The magician's assistant is for distracting the audience."

Some examples of functional ascriptions



- "The function of the heart is to pump blood."
- "That switch mutes the television."
- "The subroutine ensures that the user is authorized."
- "The magician's assistant is for distracting the audience."

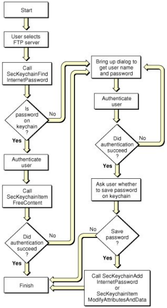
We ascribe functions to biological stuff,

Some examples of functional ascriptions

- "The function of the heart is to pump blood."
- "That switch mutes the television."
- "The subroutine ensures that the user is authorized."
- "The magician's assistant is for distracting the audience."

We ascribe functions to biological stuff, artifacts,

Some examples of functional ascriptions



- "The function of the heart is to pump blood."
- "That switch mutes the television."
- "The subroutine ensures that the user is authorized."
- "The magician's assistant is for distracting the audience."

We ascribe functions to biological stuff, artifacts, algorithms,

Some examples of functional ascriptions

- "The function of the heart is to pump blood."
- "That switch mutes the television."
- "The subroutine ensures that the user is authorized."
- "The magician's assistant is for distracting the audience."

We ascribe functions to biological stuff, artifacts, algorithms, personal roles. . .

How functions relate to means and ends

“That switch mutes the television.”

How functions relate to means and ends

"That switch mutes the television."

$\Downarrow$

One can *use* the switch to mute the television.

How functions relate to means and ends

"That switch mutes the television."

$\Downarrow$

One can *use* the switch to mute the television.

$\Downarrow$

Some *action* involving the switch will cause the television to be muted.

How functions relate to means and ends

"That switch mutes the television."

$\Downarrow$

One can *use* the switch to mute the television.

$\Downarrow$

Some *action* involving the switch will cause the television to be muted.

- Functions imply means-end relations.

How functions relate to means and ends

"That switch mutes the television."

$\Downarrow$

One can *use* the switch to mute the television.

$\Downarrow$

Some *action* involving the switch will cause the television to be muted.

- Functions imply means-end relations.
- Step one: Provide a semantics for means-end relations.

Outline

1. Means-end relations
   - Propositional Dynamic Logic
   - Means-end relations in PDL

Outline

1. Means-end relations
   - Propositional Dynamic Logic
   - Means-end relations in PDL
2. Artifactual functions
   - Functional ascriptions and fulfillment
   - Normal contexts

Outline

1. Means-end relations
   - Propositional Dynamic Logic
   - Means-end relations in PDL
2. Artifactual functions
   - Functional ascriptions and fulfillment
   - Normal contexts

PDL syntax

Propositional Dynamic Logic is a logic of actions.

PDL syntax

Propositional Dynamic Logic is a logic of actions.

Basic types:

- a set **act** of *actions*,

PDL syntax

Propositional Dynamic Logic is a logic of actions.

Basic types:

- a set **act** of *actions*,
  - Closed under:
    - *sequential composition* $\alpha; \beta$
    - *non-deterministic choice* $\alpha \cup \beta$.

PDL syntax

Propositional Dynamic Logic is a logic of actions.

Basic types:

- a set **act** of *actions*,
  - Closed under:
    - *sequential composition* $\alpha; \beta$
    - *non-deterministic choice* $\alpha \cup \beta$.
- a set **prop** of *propositions*.

PDL syntax

Propositional Dynamic Logic is a logic of actions.

Basic types:

- a set **act** of *actions*,
  - Closed under:
    - *sequential composition* $\alpha; \beta$
    - *non-deterministic choice* $\alpha \cup \beta$.
- a set **prop** of *propositions*.
  - Closed under:
    - boolean connectives,
    - dynamic operators $[\alpha]\varphi$, $\langle\alpha\rangle\varphi$.

PDL syntax

Propositional Dynamic Logic is a logic of actions.

Basic types:

- a set **act** of *actions*,
  - Closed under:
    - *sequential composition* $\alpha ; \beta$
    - *non-deterministic choice* $\alpha \cup \beta$.
- a set **prop** of *propositions*.
  - Closed under:
    - boolean connectives,
    - dynamic operators $[\alpha]\varphi$, $\langle\alpha\rangle\varphi$.

Intuitions:

- $[\alpha]\varphi$: after doing α, φ *will* hold.

PDL syntax

Propositional Dynamic Logic is a logic of actions.

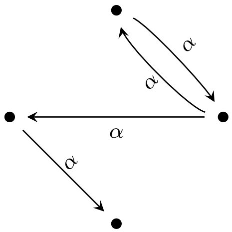
Basic types:

- a set **act** of *actions*,
  - Closed under:
    - *sequential composition* $\alpha;\beta$
    - *non-deterministic choice* $\alpha \cup \beta$.
- a set **prop** of *propositions*.
  - Closed under:
    - boolean connectives,
    - dynamic operators $[\alpha]\varphi$, $\langle\alpha\rangle\varphi$.

Intuitions:

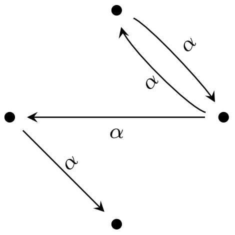
- $[\alpha]\varphi$: after doing α, φ *will* hold.
- $\langle\alpha\rangle\varphi$: after doing α, φ *might* hold.

PDL semantics



Possible world semantics with transition systems for each action α.

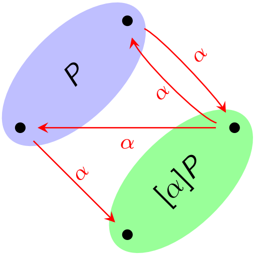
PDL semantics



Possible world semantics with transition systems for each action α.

$w \xrightarrow{\alpha} w'$ means:
one can reach w' by doing α in w.

PDL semantics

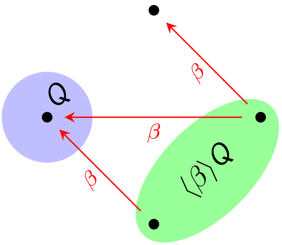


Possible world semantics with transition systems for each action α.

$w \xrightarrow{\alpha} w'$ means:
one can reach w' by doing α in w.

$w \models [\alpha]\varphi$ *iff* $\forall\, w \xrightarrow{\alpha} w' \;.\; w' \models \varphi$.

PDL semantics



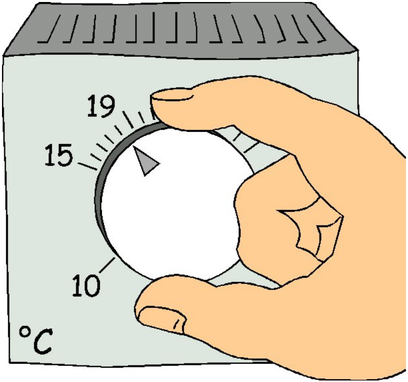
Possible world semantics with transition systems for each action α.

$w \xrightarrow{\alpha} w'$ means:
one can reach w' by doing α in w.

$$w \models [\alpha]\varphi \quad \underline{\textit{iff}} \quad \forall\, w \xrightarrow{\alpha} w' \ .\ w' \models \varphi.$$

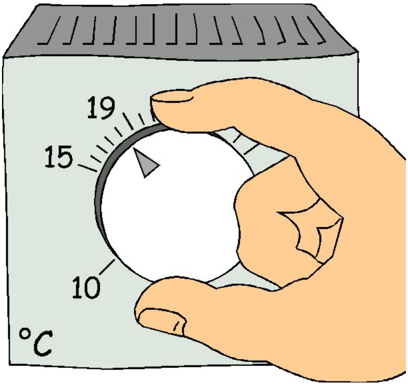
$$w \models \langle\alpha\rangle\varphi \quad \underline{\textit{iff}} \quad \exists\, w \xrightarrow{\alpha} w' \ .\ w' \models \varphi.$$

A thermostat example



Thermostat connected to heater.

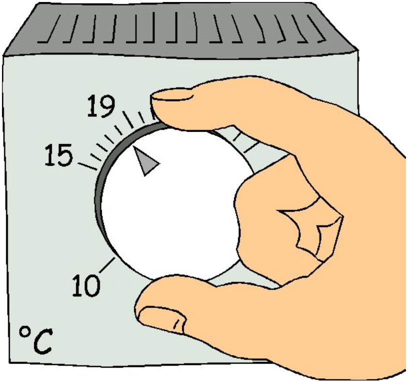
A thermostat example



Thermostat connected to heater.

Three settings: l, m, h

A thermostat example

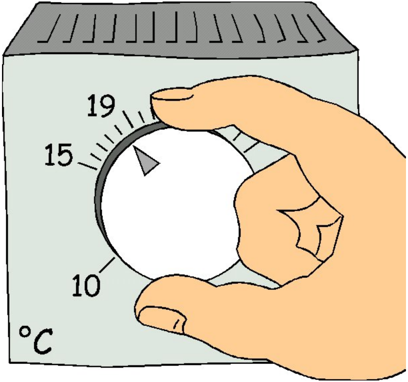


Thermostat connected to heater.
Three settings: l, m, h

Propositions:

- Setting:
  - $S = l$
  - $S = m$
  - $S = h$

A thermostat example



Thermostat connected to heater.

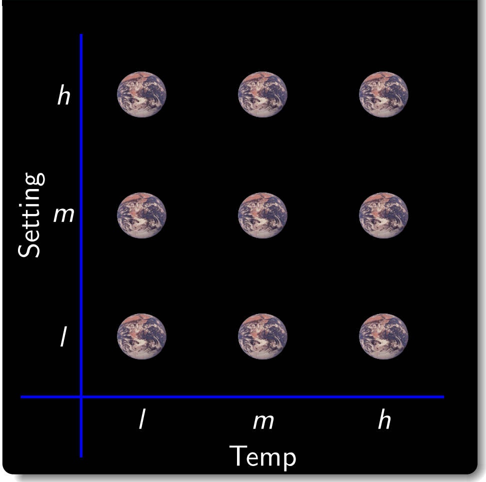
Three settings: l, m, h

Propositions:

- Setting:
  - $S = l$
  - $S = m$
  - $S = h$
- Temperature:
  - $T \geq l$
  - $T \geq m$
  - $T \geq h$

A thermostat example

Thermostat



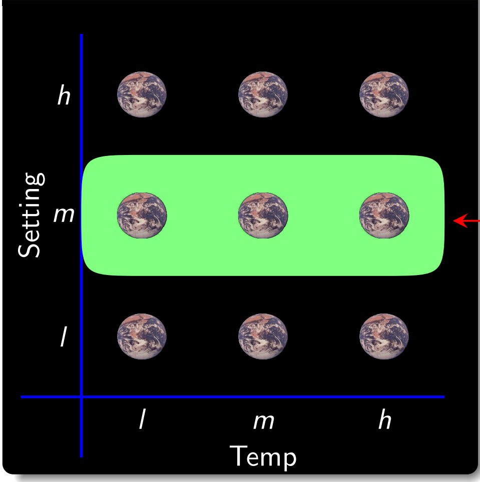
Thermostat connected to heater.
Three settings: l, m, h

Propositions:

- Setting:
  - $S = l$
  - $S = m$
  - $S = h$
- Temperature:
  - $T \geq l$
  - $T \geq m$
  - $T \geq h$

A thermostat example

Thermostat

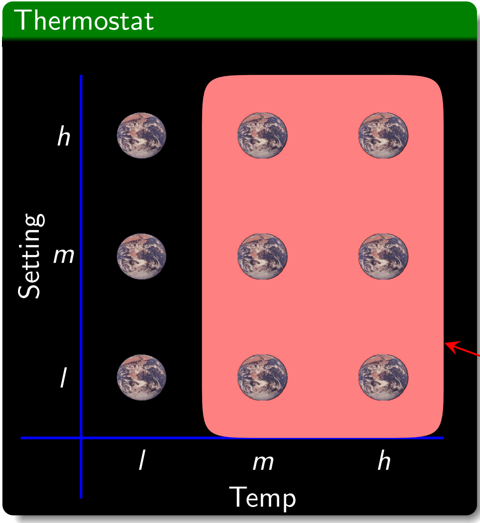


Thermostat connected to heater.
Three settings: l, m, h

Propositions:

- Setting:
  - $S = l$
  - $S = m$
  - $S = h$
- Temperature:
  - $T \geq l$
  - $T \geq m$
  - $T \geq h$

A thermostat example



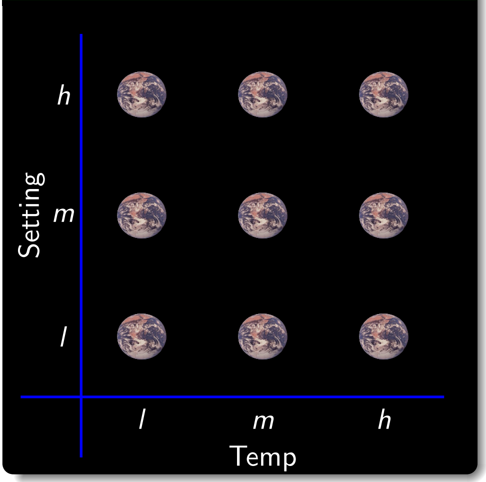
Thermostat connected to heater.
Three settings: l, m, h

Propositions:

- Setting:
  - $S = l$
  - $S = m$
  - $S = h$
- Temperature:
  - $T \geq l$
  - $T \geq m$
  - $T \geq h$

A thermostat example

Thermostat



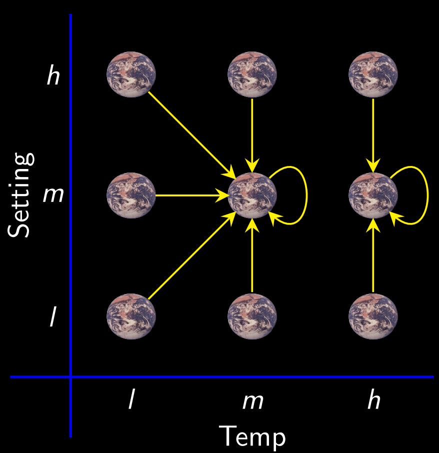
Thermostat connected to heater.
Three settings: l, m, h

Actions:

- Change setting:
  - $\textbf{set}(l)$
  - $\textbf{set}(m)$
  - $\textbf{set}(h)$

A thermostat example

Thermostat



Thermostat connected to heater.
Three settings: l, m, h

Actions:

- Change setting:
  - **set**(l)
  - **set**(m)
  - **set**(h)
- **set**(m) changes:
  - setting to m,
  - temp $\geq m$.

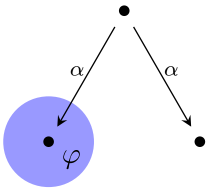
Weak and strong means-end relations

A means is an action α that can realize one's end φ.

Weak and strong means-end relations

A means is an action α that can realize one's end φ.

Two interpretations:

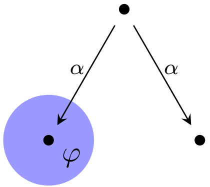


Weak: α *might* realize φ.

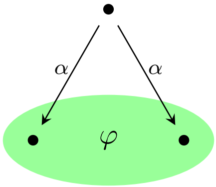
Weak and strong means-end relations

A means is an action α that can realize one's end φ.

<u>Two interpretations:</u>



Weak: α <u>*might*</u> realize φ.



Strong: α <u>*will*</u> realize φ.

Weak and strong means-end relations

A means is an action α that can realize one's end φ.

Two interpretations:

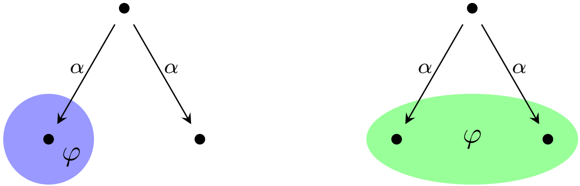
Weak: α *might* realize φ. Strong: α *will* realize φ.

α is a *weak means* to φ in w $\iff$ $w \models \langle\alpha\rangle\varphi$.

Weak and strong means-end relations

A means is an action α that can realize one's end φ.

Two interpretations:



Weak: α *might* realize φ. Strong: α *will* realize φ.

α is a *weak means* to φ in w $\iff$ $w \models \langle\alpha\rangle\varphi$.

Strong is slightly subtler.

Strong means-end relations in PDL

In *w*, α is a *strongly sufficient means* to φ

Doing α in *w* will yield φ

Strong means-end relations in PDL

In w, α is a *strongly sufficient means* to φ

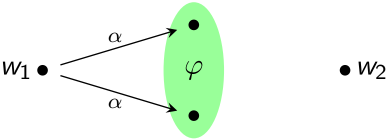
$$\underbrace{\text{Doing } \alpha \text{ in } w \text{ will yield } \varphi}_{w \models [\alpha]\varphi}$$

Strong means-end relations in PDL

In w, α is a <u>*strongly sufficient means*</u> to φ

$\underbrace{\text{Doing } \alpha \text{ in } w \text{ \underline{will} yield } \varphi}$

$w \models [\alpha]\varphi$



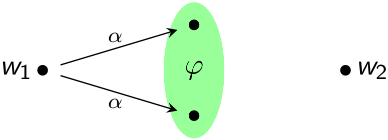
But. . .
both w_1 and w_2 satisfy $[\alpha]\varphi$!

Strong means-end relations in PDL

In w, α is a *strongly sufficient means* to φ

$\Updownarrow$

$\underbrace{\text{Doing } \alpha \text{ in } w \text{ will yield } \varphi}_{w \models [\alpha]\varphi}$ and one can do α in w.



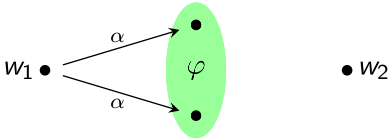
But…
both w_1 and w_2 satisfy $[\alpha]\varphi$!

Strong means-end relations in PDL

In w, α is a <u>*strongly sufficient means*</u> to φ

$$\Updownarrow$$

$\underbrace{\text{Doing } \alpha \text{ in } w \text{ \underline{will} yield } \varphi}_{w \models [\alpha]\varphi}$ and one <u>can</u> do α in w.



But...
both w_1 and w_2 satisfy $[\alpha]\varphi$!

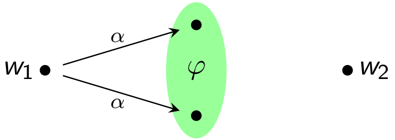
Fix: $w \models \langle\alpha\rangle$**True**
$\Updownarrow$
one <u>*can*</u> do α in w.

Strong means-end relations in PDL

In w, α is a *strongly sufficient means* to φ

$$\Updownarrow$$

$$\underbrace{\text{Doing } \alpha \text{ in } w \text{ will yield } \varphi}_{w \models [\alpha]\varphi} \quad \overset{\text{and}}{\underset{\&}{\Updownarrow}} \quad \underbrace{\text{one can do } \alpha \text{ in } w.}_{w \models \langle\alpha\rangle \textbf{True}}$$



But...
both w_1 and w_2 satisfy $[\alpha]\varphi$!

Fix: $w \models \langle\alpha\rangle \textbf{True}$
$\Updownarrow$
one *can* do α in w.

Additional topics on means-end relations
(All the thrilling details we won't discuss)

- Necessary means to an end.

Additional topics on means-end relations

(All the thrilling details we won't discuss)

- Necessary means to an end.
- Conditional means-end relations.

Additional topics on means-end relations

(All the thrilling details we won't discuss)

- Necessary means to an end.
- Conditional means-end relations.
- Practical consequences of means-end relations.

Additional topics on means-end relations
(All the thrilling details we won't discuss)

- Necessary means to an end.
- Conditional means-end relations.
- Practical consequences of means-end relations.
- Efficacy via fuzzy logic.

Outline

Where do functions come from?

Historic account:

The function of o is f

$$\Updownarrow$$

the fact that o does f explains the existence of o.

Where do functions come from?

Historic account:

The function of o is f

$\Updownarrow$

the fact that o does f explains the existence of o.

Biological function same as artifactual function.

Where do functions come from?

Historic account:

The function of o is f

$\Updownarrow$

the fact that o does f explains the existence of o.

Biological function same as artifactual function.

Intentional account:

The function of o is f

$\Updownarrow$

Someone intends to use o to do f.

Where do functions come from?

Historic account:

The function of o is f

$\Updownarrow$

the fact that o does f explains the existence of o.

Biological function same as artifactual function.

Intentional account:

The function of o is f

$\Updownarrow$

Someone intends to use o to do f.

Includes a *social* aspect.

Where do functions come from?

Historic account:

The function of o is f

$\Updownarrow$

the fact that o does f explains the existence of o.

Biological function same as artifactual function.

Intentional account:

The function of o is f

$\Updownarrow$

Someone intends to use o to do f.

Includes a *social* aspect.

Tough question. Let's avoid it.

The structure of functional ascriptions

A *functional ascription* f includes the following components.

- an artifact type T,

The structure of functional ascriptions

A *functional ascription* f includes the following components.

- an artifact type T,
- a list σ of parameter types,

The structure of functional ascriptions

A *functional ascription* f includes the following components.

- an artifact type T,
- a list σ of parameter types,
- an action α,

The structure of functional ascriptions

A *functional ascription* f includes the following components.

- an artifact type T,
- a list σ of parameter types,
- an action α,
- an end φ

The structure of functional ascriptions

A *functional ascription* f includes the following components.

- an artifact type T,
- a list σ of parameter types,
- an action α,
- an end φ

Context types

The structure of functional ascriptions

A *functional ascription* f includes the following components.

- an artifact type T,
- a list σ of parameter types,
- an action α,
- an end φ

Context types

Takes parameters from

The structure of functional ascriptions

A *functional ascription* f includes the following components.

- an artifact type T,
- a list σ of parameter types,
- an action α,
- an end φ

Context types

Takes parameters from

Expected means-end relation:

Given: a T-token o
a list τ of σ-tokens

One expects: $\alpha(o, \tau)$ is a means to $\varphi(o, \tau)$.

The structure of functional ascriptions

A *functional ascription* f includes the following components.

- an artifact type T,
- a list σ of parameter types,
- an action α,
- an end φ

Context types

Takes parameters from

Expected means-end relation:

Given: a T-token o
a list τ of σ-tokens

f-context

One expects: $\alpha(o, \tau)$ is a means to $\varphi(o, \tau)$.

The structure of functional ascriptions

A *functional ascription* f includes the following components.

- an artifact type T,
- a list σ of parameter types,
- an action α,
- an end φ

Context types

Takes parameters from

Expected means-end relation:

Given: a T-token o
a list τ of σ-tokens

f-context

One expects: $\alpha(o, \tau)$ is a means to $\varphi(o, \tau)$.

A return to the thermostat

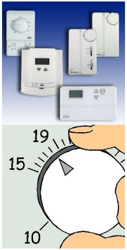
Thermostats are used to regulate temperature.

Type: $\mathcal{T}hermo$

A return to the thermostat

Thermostats are used to regulate temperature.

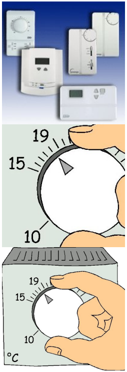
Type:	$\mathcal{T}hermo$
Parameter:	$\{l, m, h\}$



A return to the thermostat

Thermostats are used to regulate temperature.

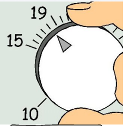
Type:	$\mathcal{T}hermo$
Parameter:	$\{l, m, h\}$
Action:	$\mathbf{set}_?(?)$

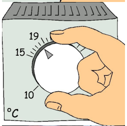


A return to the thermostat

Thermostats are used to regulate temperature.

Type:	$\mathcal{T}hermo$
Parameter:	$\{l, m, h\}$
Action:	$\mathbf{set}_?(?)$
End:	$T \geq ?$





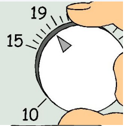
A return to the thermostat

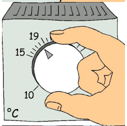
Thermostats are used to regulate temperature.

Type:	$\mathcal{T}\mathit{hermo}$
Parameter:	$\{l, m, h\}$
Action:	$\mathbf{set}_?(?)$
End:	$T \geq ?$

An *f-context* is given by

- a thermostat o,





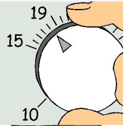
A return to the thermostat

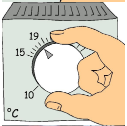
Thermostats are used to regulate temperature.

Type:	$\mathcal{T}hermo$
Parameter:	$\{l, m, h\}$
Action:	$\mathbf{set}_?(?)$
End:	$T \geq ?$

An <u>*f-context*</u> is given by

- a thermostat o,
- a setting $x \in \{l, m, h\}$.





A return to the thermostat

Thermostats are used to regulate temperature.

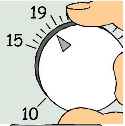
Type:	$\mathcal{T}hermo$
Parameter:	$\{l, m, h\}$
Action:	$\mathbf{set}_?(?)$
End:	$T \geq ?$

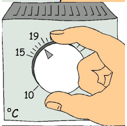
An <u>*f-context*</u> is given by

- a thermostat o,
- a setting $x \in \{l, m, h\}$.

In an f-context $\langle o, x \rangle$,

- our action is $\mathbf{set}_o(x)$: set thermostat o to x.





A return to the thermostat

Thermostats are used to regulate temperature.

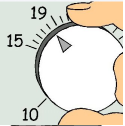
Type:	$\mathcal{T}hermo$
Parameter:	$\{l, m, h\}$
Action:	$\mathbf{set}_?(?)$
End:	$T \geq ?$

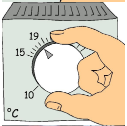
An *f-context* is given by

- a thermostat o,
- a setting $x \in \{l, m, h\}$.

In an f-context $\langle o, x \rangle$,

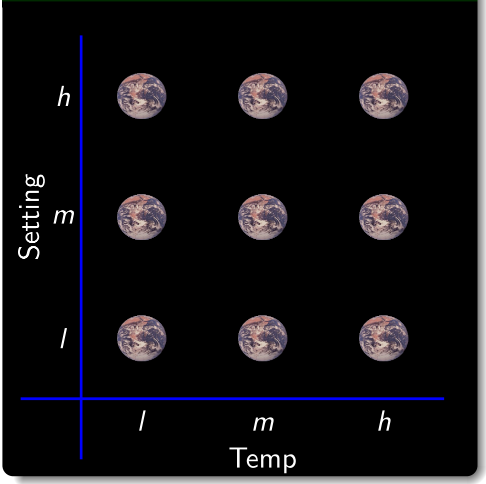
- our action is $\mathbf{set}_o(x)$: set thermostat o to x.
- our end is $T \geq x$.





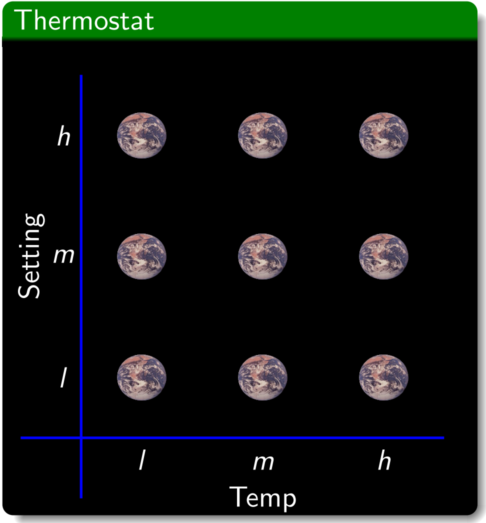
Contexts and transition systems

Thermostat



Each f-context $\langle o, x\rangle$ determines a PDL model.

Contexts and transition systems

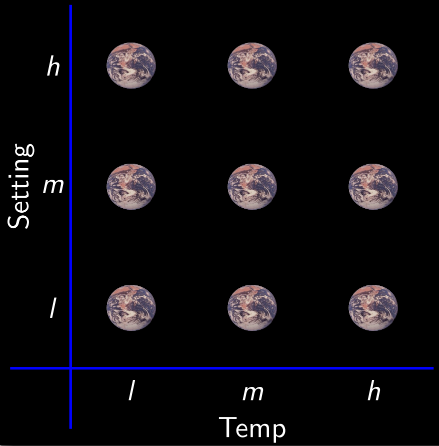


Each f-context $\langle o, x \rangle$ determines a PDL model.

- o: the artifact used.

Contexts and transition systems

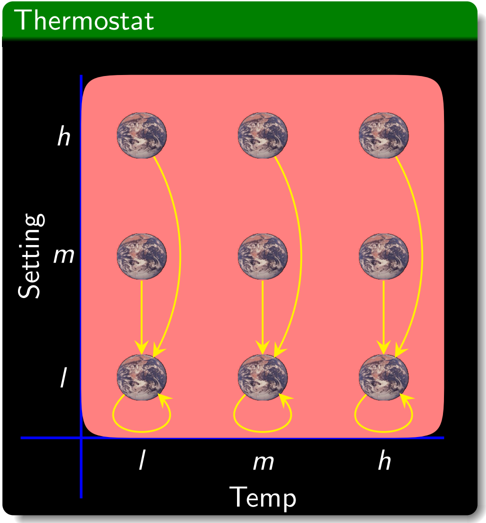
Thermostat



Each f-context $\langle o, x\rangle$ determines a PDL model.

- o: the artifact used.
- x: the setting.

Contexts and transition systems



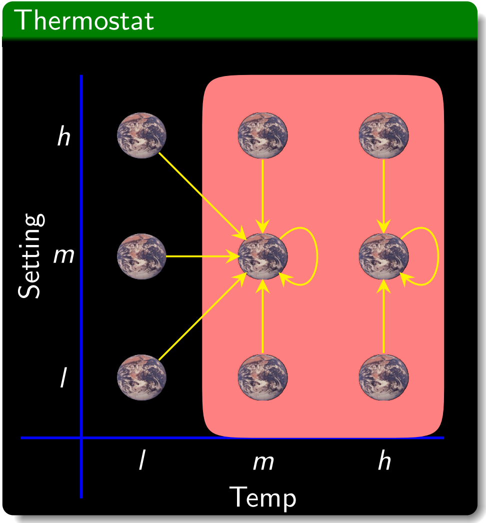
Each f-context $\langle o, x \rangle$ determines a PDL model.

- o: the artifact used.
- x: the setting.

Examples:

- $\langle$Working, $l\rangle$.

Contexts and transition systems



Each f-context $\langle o, x\rangle$ determines a PDL model.

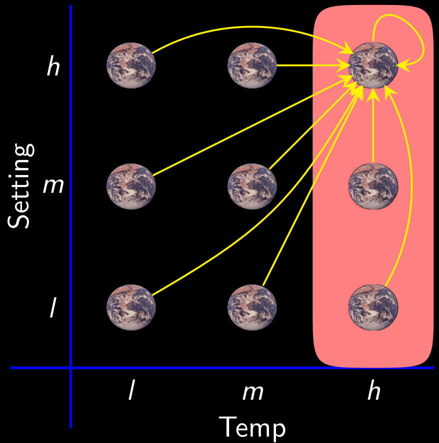
- o: the artifact used.
- x: the setting.

Examples:

- $\langle$Working, $l\rangle$.
- $\langle$Working, $m\rangle$.

Contexts and transition systems

Thermostat



Each f-context $\langle o, x\rangle$ determines a PDL model.

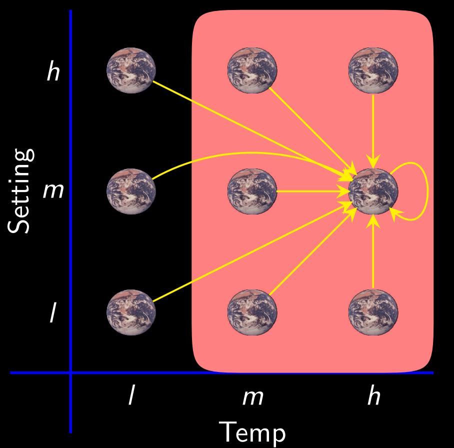
- o: the artifact used.
- x: the setting.

Examples:

- $\langle$Working, $l\rangle$.
- $\langle$Working, $m\rangle$.
- $\langle$Working, $h\rangle$.

Contexts and transition systems

Thermostat



Each f-context $\langle o, x\rangle$ determines a PDL model.

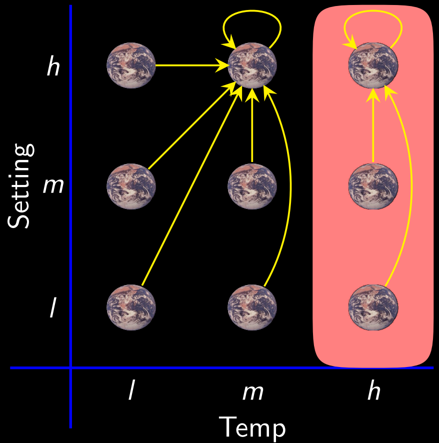
- o: the artifact used.
- x: the setting.

Examples:

- $\langle$Miscal, $m\rangle$.

Contexts and transition systems

Thermostat



Each f-context $\langle o, x \rangle$ determines a PDL model.

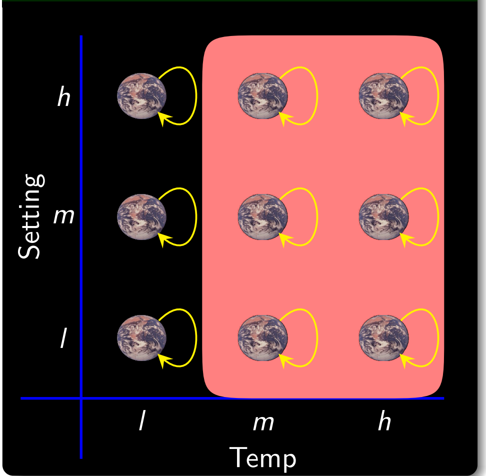
- o: the artifact used.
- x: the setting.

Examples:

- $\langle$Miscal, $m\rangle$.
- $\langle$Weak, $h\rangle$.

Contexts and transition systems

Thermostat



Each f-context $\langle o, x \rangle$ determines a PDL model.

- o: the artifact used.
- x: the setting.

Examples:

- $\langle$Miscal, $m\rangle$.
- $\langle$Weak, $h\rangle$.
- $\langle$Broke, $m\rangle$.

Fulfillment

An artifact o *(weakly/strongly) fulfills f* wrt τ

$$\Updownarrow$$

α is a *(weak/strong) means* to φ in $\mathcal{M}_{\langle o,\tau\rangle}$.

Fulfillment

An artifact o *(weakly/strongly) fulfills f* wrt τ

$$\Updownarrow$$

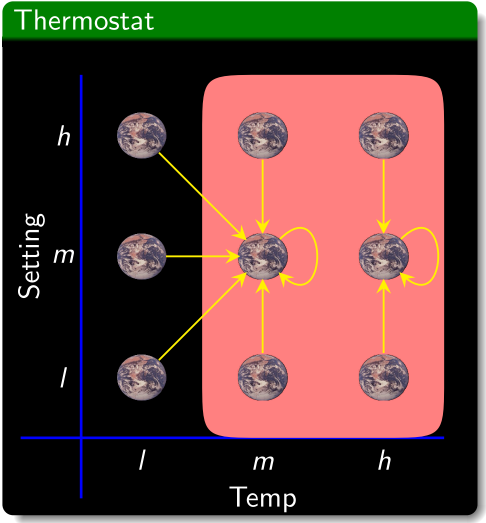
α is a *(weak/strong) means* to φ in $\mathcal{M}_{\langle o, \tau \rangle}$.

A thermostat t fulfills f wrt x

$$\Updownarrow$$

Setting t to x realizes $T \geq x$.

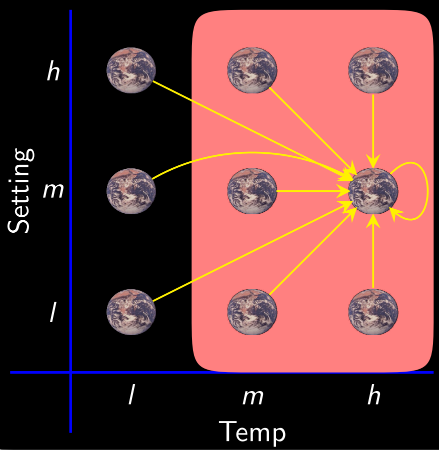
Contexts and transition systems



Token	fulfills f
Working	l, m, h

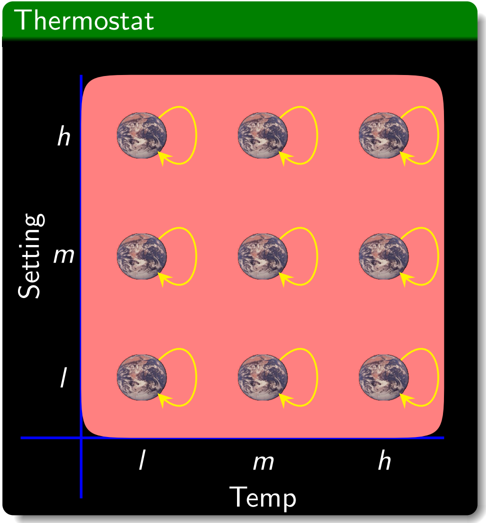
Contexts and transition systems

Thermostat



Token	fulfills f
Working	l, m, h
Miscal	l, m, h

Contexts and transition systems



Token	fulfills f
Working	$l,\ m,\ h$
Miscal	$l,\ m,\ h$
Broke	l

Fulfillment

An artifact o *(weakly/strongly) fulfills f* wrt τ

$\Updownarrow$

α is a *(weak/strong) means* to φ in $\mathcal{M}_{\langle o,\tau\rangle}$.

A thermostat t fulfills f wrt x

$\Updownarrow$

Setting t to x realizes $T \geq x$.

A thermostat t *universally fulfills* f

$\Updownarrow$

t fulfills f wrt every x.

Type fulfillment

Defined: token fulfills a function f.

Type fulfillment

Defined: token fulfills a function f.

When does a *subtype* $T' \leq T$ fulfill f?

Type fulfillment

Defined: token fulfills a function f.

When does a *subtype* $T' \leq T$ fulfill f?

Universal fulfillment:

every $o \in T'$ fulfills f.

Type fulfillment

Defined: token fulfills a function f.

When does a *subtype* $T' \leq T$ fulfill f?

Universal fulfillment:

every $o \in T'$ fulfills f.

Normal fulfillment:

every "*normal*" $o \in T'$ fulfills f.

Normal tokens: the controversial bits

Each type T comes with a set N_T of *normal* tokens.

Normal tokens: the controversial bits

Each type T comes with a set N_T of *normal* tokens.

Are normal tokens "real" tokens?

Normal tokens: the controversial bits

Each type T comes with a set N_T of *normal* tokens.

Are normal tokens "real" tokens? NO!

every T-token is broken

$$\not\Downarrow$$

normal T-tokens are broken.

Normal tokens: the controversial bits

Each type T comes with a set N_T of *normal* tokens.

Are normal tokens "real" tokens? NO!

$$\text{every } T\text{-token is broken} \;\not\Downarrow\; \text{normal } T\text{-tokens are broken.}$$

Normal tokens are useful fictions.
Express how T-things are expected to behave.

Normal tokens: the excuses

We add fictional objects to our semantics?

What are you thinking?

Normal tokens: the excuses

We add fictional objects to our semantics?
What are you thinking?

- Counterfactuals bad. Fictions barely worse.

Normal tokens: the excuses

We add fictional objects to our semantics?
What are you thinking?

- Counterfactuals bad. Fictions barely worse.
- Fictional tokens approximate intuitions.

Normal tokens: the excuses

We add fictional objects to our semantics?
What are you thinking?

- Counterfactuals bad. Fictions barely worse.
- Fictional tokens approximate intuitions.
- Formally simple, conceptually opaque.

Normal tokens: the excuses

We add fictional objects to our semantics?
What are you thinking?

- Counterfactuals bad. Fictions barely worse.
- Fictional tokens approximate intuitions.
- Formally simple, conceptually opaque.
- Gives sense of malfunction.

Normal tokens: the excuses

We add fictional objects to our semantics?
What are you thinking?

- Counterfactuals bad. Fictions barely worse.
- Fictional tokens approximate intuitions.
- Formally simple, conceptually opaque.
- Gives sense of malfunction.
- Distinguishes subtypes.

Normal tokens: subtypes

Subtypes do not always inherit functional ascriptions.

Normal tokens: subtypes

Subtypes do not always inherit functional ascriptions.

$$f \text{ is a function of } T \text{ and } T' \leq T$$

$$\not\Downarrow$$

$$T' \text{ fulfills } f.$$

Normal tokens: subtypes

Subtypes do not always inherit functional ascriptions.

$$f \text{ is a function of } T \text{ and } T' \leq T$$

$$\not\Downarrow$$

$$T' \text{ fulfills } f.$$

Universal fulfillment:

$$T \text{ fulfills } f \quad \Rightarrow \quad T' \text{ fulfills } f$$

Normal tokens: subtypes

Subtypes do not always inherit functional ascriptions.

$$f \text{ is a function of } T \text{ and } T' \leq T$$

$$\not\Downarrow$$

$$T' \text{ fulfills } f.$$

<u>Universal fulfillment:</u>

$$T \text{ fulfills } f \quad \Rightarrow \quad T' \text{ fulfills } f$$

<u>Normal fulfillment:</u>

$$T \text{ fulfills } f \ \underline{\textit{and}}\ N_{T'} \subseteq N_T \quad \Rightarrow \quad T' \text{ fulfills } f$$

Normal tokens: subtypes

Subtypes do not always inherit functional ascriptions.

$$f \text{ is a function of } T \text{ and } T' \leq T$$

$$\not\Downarrow$$

$$T' \text{ fulfills } f.$$

<u>Universal fulfillment:</u>

$$T \text{ fulfills } f \quad \Rightarrow \quad T' \text{ fulfills } f$$

<u>Normal fulfillment:</u>

$$T \text{ fulfills } f \ \underline{\text{and}}\ N_{T'} \subseteq N_T \quad \Rightarrow \quad T' \text{ fulfills } f$$

Normal flare guns aren't normal guns.

Outstanding issues

- A philosophical treatment of “normal tokens”.

Outstanding issues

- A philosophical treatment of "normal tokens".
- Add efficacy to functions.

Outstanding issues

- A philosophical treatment of “normal tokens”.
- Add efficacy to functions.
- A formalization of malfunction.

Outstanding issues

- A philosophical treatment of "normal tokens".
- Add efficacy to functions.
- A formalization of malfunction.
- Types and function inheritance.

Outstanding issues

- A philosophical treatment of "normal tokens".
- Add efficacy to functions.
- A formalization of malfunction.
- Types and function inheritance.
- Everything else.