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

Means-end relations
Artifactual functions
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.”

One can *use* the switch to mute the television.
How functions relate to means and ends

“That switch mutes the television.”

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

Some *action* involving the switch will cause the television to be muted.
How functions relate to means and ends

“That switch mutes the television.”

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

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.”

One can use the switch to mute the television.

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 of **actions**, 

Hughes

Means-End Relations and Artifactual Functions
Propositional Dynamic Logic is a logic of actions.

Basic types:
- a set \( \text{act} \) 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**.
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 \([\alpha]\varphi, \langle\alpha\rangle\varphi\).
**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 \(\alpha\), \(\varphi\) will hold.
Propositional Dynamic Logic is a logic of actions.

Basic types:
- a set \text{act} of \text{actions},
  - Closed under:
    - \text{sequential composition } \alpha; \beta
    - \text{non-deterministic choice } \alpha \cup \beta.
- a set \text{prop} of \text{propositions}.
  - Closed under:
    - boolean connectives,
    - dynamic operators \([\alpha]\varphi, \langle \alpha \rangle \varphi\).

Intuitions:
- \([\alpha]\varphi\): after doing \alpha, \varphi \text{ will} hold.
- \(\langle \alpha \rangle \varphi\): after doing \alpha, \varphi \text{ might} hold.
PDL semantics

Possible world semantics with transition systems for each action $\alpha$. 
**PDL semantics**

Possible world semantics with transition systems for each action $\alpha$.

$w \xrightarrow{\alpha} w'$ means:

one can reach $w'$ by doing $\alpha$ in $w$. 
PDL semantics

Possible world semantics with transition systems for each action $\alpha$.

$$w \xrightarrow{\alpha} w'$$ means:

one can reach $w'$ by doing $\alpha$ 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 $\alpha$.

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

\[
\begin{align*}
w & \models [\alpha] \varphi \iff \forall w \xrightarrow{\alpha} w'. w' \models \varphi. \\
w & \models \langle \alpha \rangle \varphi \iff \exists w \xrightarrow{\alpha} w'. w' \models \varphi.
\end{align*}
\]
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 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 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: \( l, m, h \)

Actions:
- Change setting:
  - \( \text{set}(l) \)
  - \( \text{set}(m) \)
  - \( \text{set}(h) \)
A thermostat example

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 $\geq m$. 

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 $\alpha$ that can realize one’s end $\varphi$. 
Weak and strong means-end relations

A means is an action $\alpha$ that can realize one’s end $\varphi$.

**Two interpretations:**

- Weak: $\alpha$ *might* realize $\varphi$.
- Strong: $\alpha$ *will* realize $\varphi$.

$\alpha$ is a weak means to $\varphi$ in $w$ if $w \models \langle \alpha \rangle \varphi$. Strong is slightly subtler.
Weak and strong means-end relations

A means is an action $\alpha$ that can realize one’s end $\varphi$.

Two interpretations:

Weak: $\alpha$ _might_ realize $\varphi$.  
Strong: $\alpha$ _will_ realize $\varphi$.  

$\alpha$ is a weak means to $\varphi$ in $w$ if $w \models \langle \alpha \rangle \varphi$.

Strong is slightly subtler.
Weak and strong means-end relations

A means is an action $\alpha$ that can realize one’s end $\varphi$.

Two interpretations:

Weak: $\alpha$ might realize $\varphi$.

Strong: $\alpha$ will realize $\varphi$.

$\alpha$ is a weak means to $\varphi$ in $w$ $\iff$ $w \models \langle \alpha \rangle \varphi$. 
Weak and strong means-end relations

A means is an action \( \alpha \) that can realize one’s end \( \varphi \).

Two interpretations:

Weak: \( \alpha \) \textit{might} realize \( \varphi \).

Strong: \( \alpha \) \textit{will} realize \( \varphi \).

\( \alpha \) is a \textit{weak means} to \( \varphi \) in \( w \) \iff \( w \models \langle \alpha \rangle \varphi \).

Strong is slightly subtler.
Strong means-end relations in PDL

In $w$, $\alpha$ is a *strongly sufficient means* to $\varphi$

Doing $\alpha$ in $w$ will yield $\varphi$
Strong means-end relations in PDL

In $w$, $\alpha$ is a strongly sufficient means to $\varphi$

Doing $\alpha$ in $w$ will yield $\varphi$

$w \models [\alpha] \varphi$
Strong means-end relations in PDL

In $w$, $\alpha$ is a \textit{strongly sufficient means} to $\varphi$

\[
\begin{align*}
\text{Doing } \alpha \text{ in } w \text{ will yield } \varphi \\
\end{align*}
\]

\[
\begin{align*}
w \models [\alpha] \varphi
\end{align*}
\]

But... both $w_1$ and $w_2$ satisfy $[\alpha] \varphi$!
Strong means-end relations in PDL

In $w$, $\alpha$ is a strongly sufficient means to $\varphi$

Doing $\alpha$ in $w$ will yield $\varphi$ and one can do $\alpha$ in $w$.

$w \models [\alpha] \varphi$

$w_1 \bullet \xrightarrow{\alpha} \bullet \varphi \xrightarrow{\alpha} \bullet w_2$

But... both $w_1$ and $w_2$ satisfy $[\alpha] \varphi$!
Strong means-end relations in PDL

In $w$, $\alpha$ is a **strongly sufficient means** to $\varphi$

$\iff$

Doing $\alpha$ in $w$ **will yield** $\varphi$ and one **can** do $\alpha$ in $w$.

$$w \models [\alpha] \varphi$$

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

Fix: $w \models \langle \alpha \rangle \text{True}$

one **can** do $\alpha$ in $w$. 
Strong means-end relations in PDL

In $w$, $\alpha$ is a **strongly sufficient means** to $\varphi$

Doing $\alpha$ in $w$ will yield $\varphi$ and one can do $\alpha$ in $w$.

$w \models [\alpha] \varphi$

$w \models \langle \alpha \rangle \text{True}$

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

**Fix:** $w \models \langle \alpha \rangle \text{True}$

one can do $\alpha$ 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

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

2. Artifactual functions
   - Functional ascriptions and fulfillment
   - Normal contexts
Where do functions come from?

Historic account:

The function of $o$ is $f$

$\uparrow$

the fact that $o$ does $f$

explains the existence of $o$. 
Where do functions come from?

**Historic account:**

The function of $o$ is $f$

$\iff$

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 $f$  
$\leftarrow\rightarrow$  
the fact that $o$ does $f$ explains the existence of $o$.

Biological function same as artifactual function.

**Intentional account:**

The function of $o$ is $f$  
$\leftarrow\rightarrow$  
*Someone* intends to use $o$ to do $f$.  

Tough question. Let’s avoid it.
Where do functions come from?

**Historic account:**

The function of $o$ is $f$  
\[ \iff \]
the fact that $o$ does $f$  
explains the existence of $o$.

Biological function same as artifactual function.

**Intentional account:**

The function of $o$ is $f$  
\[ \iff \]
*Someone* intends to use $o$ to do $f$.  

Includes a *social* aspect.
Where do functions come from?

**Historic account:**

The function of $o$ is $f$ \[\iff\] the fact that $o$ does $f$ explains the existence of $o$.

Biological function same as artifactual function.

**Intentional account:**

The function of $o$ is $f$ \[\iff\] **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 $\sigma$ of parameter types,
The structure of functional ascriptions

A functional ascription $f$ includes the following components.

- an artifact type $T$,
- a list $\sigma$ of parameter types,
- an action $\alpha$, 

Expected means-end relation:

Given:

One expects:

$$\alpha(o, \tau) \text{ is a means to } \phi(o, \tau).$$

$\forall - \text{context}$

Hughes

Means-End Relations and Artifactual Functions
The structure of functional ascriptions

A functional ascription $f$ includes the following components.

- an artifact type $T$,
- a list $\sigma$ of parameter types,
- an action $\alpha$,
- an end $\varphi$
The structure of functional ascriptions

A *functional ascription* $f$ includes the following components.

1. an artifact type $T$,
2. a list $\sigma$ of parameter types,
3. an action $\alpha$,
4. an end $\varphi$
The structure of functional ascriptions

A functional ascription $f$ includes the following components.

- an artifact type $T$,
- a list $\sigma$ of parameter types,
- an action $\alpha$,
- an end $\varphi$
A functional ascription $f$ includes the following components.

- an artifact type $T$,
- a list $\sigma$ of parameter types,
- an action $\alpha$,
- an end $\varphi$

Expected means-end relation:

Given: a $T$-token $o$

- a list $\tau$ of $\sigma$-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 $\sigma$ of parameter types,
- an action $\alpha$,
- an end $\varphi$

**Expected means-end relation:**

*Given:* a $T$-token $o$

- a list $\tau$ of $\sigma$-tokens

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

**Context types:**

Takes parameters from
The structure of functional ascriptions

A functional ascription $f$ includes the following components.

- an artifact type $T$,
- a list $\sigma$ of parameter types,
- an action $\alpha$,
- an end $\varphi$

**Expected means-end relation:**

*Given:* a $T$-token $o$

- a list $\tau$ of $\sigma$-tokens

*One expects:* $\alpha(o, \tau)$ is a means to $\varphi(o, \tau)$. 
A return to the thermostat

Thermostats are used to regulate temperature.

Type: *Thermo*
A return to the thermostat

Thermostats are used to regulate temperature.

Type: \( T \)hermo

Parameter: \( \{ l, m, h \} \)
A return to the thermostat

Thermostats are used to regulate temperature.

Type: \( T_{\text{ermo}} \)

Parameter: \( \{l, m, h\} \)

Action: \( \text{set}_? (?) \)
Thermostats are used to regulate temperature.

Type: \( T_{thermo} \)

Parameter: \( \{ l, m, h \} \)

Action: \( \text{set} \?) (?) \)

End: \( T \geq ? \)
A return to the thermostat

Thermostats are used to regulate temperature.

Type: \( \mathcal{T} \text{hermo} \)

Parameter: \( \{ l, m, h \} \)

Action: \( \text{set}_? (?) \)

End: \( T \geq ? \)

An \textit{f-context} is given by

- a thermostat \( o \),
A return to the thermostat

Thermostats are used to regulate temperature.

- **Type:** \( T \) \textit{ermo}
- **Parameter:** \( \{l, m, h\} \)
- **Action:** \texttt{set}(?)
- **End:** \( T \geq ? \)

An \textit{f-context} 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: \( T \)hermo

Parameter: \( \{l, m, h\} \)

Action: \( \text{set}_? (?) \)

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 \( \text{set}_o(x) \): set thermostat \( o \) to \( x \).
Means-end relations
Artifactual functions
Functional ascriptions and fulfillment
Normal contexts

A return to the thermostat

Thermostats are used to regulate temperature.

- **Type:** \( \mathcal{T} \)hermo
- **Parameter:** \( \{l, m, h\} \)
- **Action:** 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 \( \text{set}_o(x) \): set thermostat \( o \) to \( x \).
- our end is \( T \geq x \).
Means-end relations
Artifactual functions

Contexts and transition systems

Thermostat

<table>
<thead>
<tr>
<th>Setting</th>
<th>Temp</th>
</tr>
</thead>
<tbody>
<tr>
<td>l</td>
<td>l</td>
</tr>
<tr>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>h</td>
<td>h</td>
</tr>
</tbody>
</table>

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

- $o$: the artifact used.
Contexts and transition systems

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

- \( o \): the artifact used.
- \( x \): the setting.
Each $f$-context $\langle o, x \rangle$ determines a PDL model.

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

Examples:
- $\langle \text{Working}, l \rangle$.
Contexts and transition systems

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

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

Examples:
- $\langle \text{Working}, l \rangle$.
- $\langle \text{Working}, m \rangle$. 

Thermostat

<table>
<thead>
<tr>
<th>Setting</th>
<th>Temp</th>
</tr>
</thead>
<tbody>
<tr>
<td>$l$</td>
<td>$l$</td>
</tr>
<tr>
<td>$m$</td>
<td>$m$</td>
</tr>
<tr>
<td>$h$</td>
<td>$h$</td>
</tr>
</tbody>
</table>

$o$: 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:
- $\langle \text{Working}, l \rangle$.
- $\langle \text{Working}, m \rangle$.
- $\langle \text{Working}, h \rangle$. 
Each \( f \)-context \( \langle o, x \rangle \) determines a PDL model.

- \( o \): the artifact used.
- \( x \): the setting.

Examples:

- \( \langle \text{Miscal}, m \rangle \).
Mean-end relations and artifactual functions

Contexts and transition systems

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

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

Examples:

- $\langle \text{Miscal}, m \rangle$.
- $\langle \text{Weak}, h \rangle$.
Contexts and transition systems

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

- \( o \): the artifact used.
- \( x \): the setting.

Examples:
- \( \langle \text{Miscal}, m \rangle \).
- \( \langle \text{Weak}, h \rangle \).
- \( \langle \text{Broke}, m \rangle \).
Fulfillment

An artifact $o$ (weakly/strongly) fulfills $f$ wrt $\tau$

$\uparrow$

$\alpha$ is a (weak/strong) means to $\varphi$ in $\mathcal{M}_{\langle o, \tau \rangle}$.
Fulfillment

An artifact \( o \) (weakly/strongly) fulfills \( f \) wrt \( \tau \)
\[ \iff \]
\( \alpha \) is a (weak/strong) means to \( \varphi \) in \( M_{\langle o, \tau \rangle} \).

A thermostat \( t \) fulfills \( f \) wrt \( x \)
\[ \iff \]
Setting \( t \) to \( x \) realizes \( T \geq x \).
Contexts and transition systems

Thermostat

<table>
<thead>
<tr>
<th>Token</th>
<th>fulfills $f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working</td>
<td>$l, m, h$</td>
</tr>
</tbody>
</table>

Means-End Relations and Artifactual Functions
Means-end relations
Artifactual functions

Functional ascriptions and fulfillment
Normal contexts

Contexts and transition systems

<table>
<thead>
<tr>
<th>Token</th>
<th>fulfills f</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working</td>
<td>l, m, h</td>
</tr>
<tr>
<td>Miscal</td>
<td>l, m, h</td>
</tr>
</tbody>
</table>

Thermostat

<table>
<thead>
<tr>
<th>Setting</th>
<th>Temp</th>
</tr>
</thead>
<tbody>
<tr>
<td>h</td>
<td>l, m, h</td>
</tr>
<tr>
<td>m</td>
<td>l, m, h</td>
</tr>
<tr>
<td>l</td>
<td>l, m, h</td>
</tr>
</tbody>
</table>
**Means-End Relations and Artifactual Functions**

**Contexts and transition systems**

<table>
<thead>
<tr>
<th>Token</th>
<th>fulfills f</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working</td>
<td>l, m, h</td>
</tr>
<tr>
<td>Miscal</td>
<td>l, m, h</td>
</tr>
<tr>
<td>Broke</td>
<td>l</td>
</tr>
</tbody>
</table>

**Thermostat**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Temp</th>
</tr>
</thead>
<tbody>
<tr>
<td>l</td>
<td>l</td>
</tr>
<tr>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>h</td>
<td>h</td>
</tr>
</tbody>
</table>

- l: Low temperature
- m: Medium temperature
- h: High temperature

- Working
- Miscal
- Broke
Fulfillment

An artifact $o$ \textit{(weakly/strongly) fulfills $f$ wrt $\tau$}

$\iff$

$\alpha$ is a \textit{(weak/strong) means} to $\varphi$ in $\mathcal{M}_{(o, \tau)}$.

A thermostat $t$ fulfills $f$ wrt $x$

$\iff$

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

A thermostat $t$ \textit{universally fulfills $f$}

$\iff$

$t$ 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:**

\[
every \ o \in T' \text{ fulfills } f.\]

Hughes

Means-End Relations and Artifactual Functions
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!

\[ \text{every } T\text{-token is broken} \]

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!

every $T$-token is broken

normal $T$-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.

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

\[ \times \]

\[ T' \text{ fulfills } f. \]
Normal tokens: subtypes

Subtypes do not always inherit functional ascriptions.

\[ f \text{ is a function of } T \text{ and } T' \leq T \]
\[ \text{ } \]
\[ T' \text{ fulfills } f. \]

**Universal fulfillment:**

\[ T \text{ fulfills } f \implies T' \text{ fulfills } f \]
Normal tokens: subtypes

Subtypes do not always inherit functional ascriptions.

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

\[ \text{X} \]

\[ T' \text{ fulfills } f. \]

**Universal fulfillment:**

\[ T \text{ fulfills } f \implies T' \text{ fulfills } f \]

**Normal fulfillment:**

\[ T \text{ fulfills } f \text{ and } N_{T'} \subseteq N_T \implies T' \text{ fulfills } f \]
Normal tokens: subtypes

Subtypes do not always inherit functional ascriptions.

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

\[ \times \]

\[ T' \text{ fulfills } f. \]

**Universal fulfillment:**

\[ T \text{ fulfills } f \Rightarrow T' \text{ fulfills } f \]

**Normal fulfillment:**

\[ T \text{ fulfills } f \text{ and } N_{T'} \subseteq N_T \Rightarrow T' \text{ fulfills } f \]

Normal flare guns aren’t normal guns.
Outstanding issues

- A philosophical treatment of “normal tokens”.

Hughes

Means-End Relations and Artifactual Functions
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.