A Semantics for Functional Efficacy

Jesse Hughes[1] Sjoerd Zwart[1,2]

[1] Technical University of Eindhoven

[2] Technical University of Delft

August 11, 2005

Outline

1. Means-end relations and artifactual functions
    - From functions to means-end relations
    - Means-end relations and function fulfillment

Outline

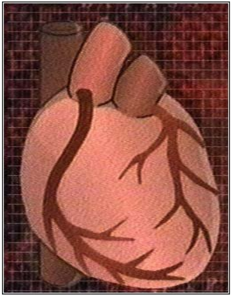
1. Means-end relations and artifactual functions
   - From functions to means-end relations
   - Means-end relations and function fulfillment
2. Efficacy and malfunction
   - Efficacy as a fuzzy property
   - Type-token comparisons and malfunction

Outline

1. Means-end relations and artifactual functions
   - From functions to means-end relations
   - Means-end relations and function fulfillment

2. Efficacy and malfunction
   - Efficacy as a fuzzy property
   - Type-token comparisons and malfunction

Functional ascriptions

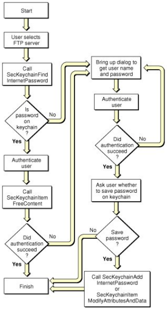


- "The function of the heart is to pump blood."

Functional ascriptions

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

Functional ascriptions

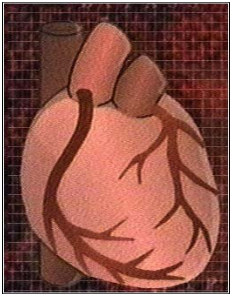

Start
User selects FTP server
Call SecKeychainFind InternetPassword
Is password on keychain ?
No
Yes
Authenticate user
Call SecKeychainItem FreeContent
Did authentication succeed ?
No
Yes
Finish
Bring up dialog to get user name and password
Authenticate user
Did authentication succeed ?
No
Yes
Ask user whether to save password on keychain
Save password ?
No
Yes
Call SecKeychainAdd InternetPassword or SecKeychainItem ModifyAttributesAndData

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

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."

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.”

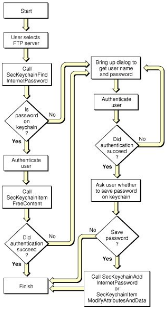
We ascribe functions to biological stuff,

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,

Functional ascriptions


Start
User selects FTP server
Call SecKeychainFind InternetPassword
Is password on keychain ?
No
Yes
Authenticate user
Call SecKeychainItem FreeContent
Did authentication succeed ?
No
Yes
Finish
Bring up dialog to get user name and password
Authenticate user
Did authentication succeed ?
No
Yes
Ask user whether to save password on keychain
Save password ?
No
Yes
Call SecKeychainAdd InternetPassword or SecKeychainItem ModifyAttributesAndData

- "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,

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. . .

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. . .

We focus on *artifactual functions*.

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.”

$\Downarrow$

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."

⇓

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.
- Aim: Use means-end semantics to analyze functions.

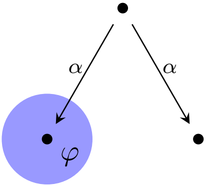
Weak and strong means-end relations

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:

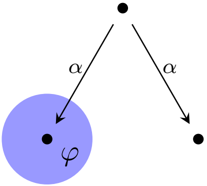


Weak: α *might* realize φ.

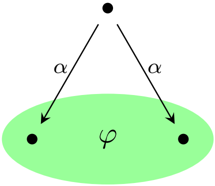
Weak and strong means-end relations

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

Two interpretations:



Weak: α *might* realize φ.

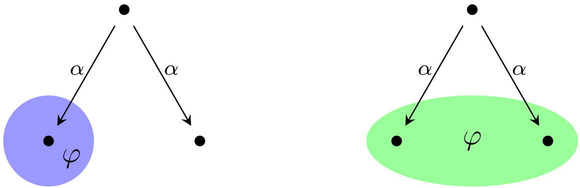


Strong: α *will* realize φ.

Weak and strong means-end relations

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

Two interpretations:



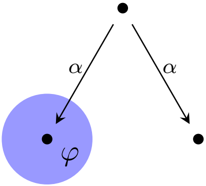
Weak: α *might* realize φ. Strong: α *will* realize φ.

Both are easily expressed in Propositional Dynamic Logic.

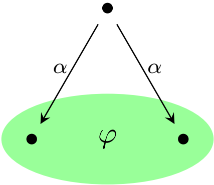
Weak and strong means-end relations

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.

Both are easily expressed in Propositional Dynamic Logic.

Result: Formal definitions for "α is a (weak/strong) means to φ."

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, $\Big\}$ Context 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

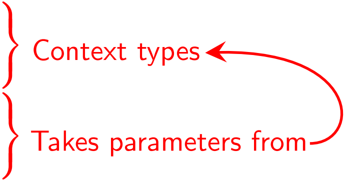
Context types

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:

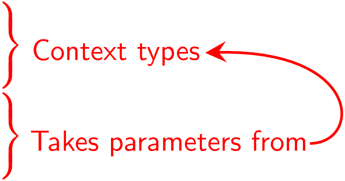
Given: a T-token o
a list τ of σ-tokens

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

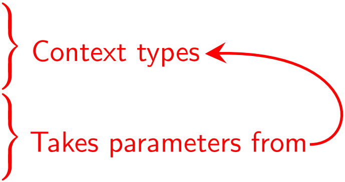
f-context

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 $\varphi(o, \tau)$.

Example: fire starters

Various artifacts are used to start fires.

Type: $\mathcal{F}ire\mathcal{S}tarter$

Example: fire starters

Various artifacts are used to start fires.

Type:	*FireStarter*
Parameters:	User
	Weather/lighting conditions
	Kindling

Example: fire starters

Various artifacts are used to start fires.

Type:	*FireStarter*
Parameters:	User
	Weather/lighting conditions
	Kindling
Action:	**ignite**$_?$(?)

Example: fire starters

Various artifacts are used to start fires.

Type:	$\mathcal{F}ire\mathcal{S}tarter$
Parameters:	User
	Weather/lighting conditions
	Kindling
Action:	**ignite**$_?$(?)
End:	burning(?)

Example: fire starters

Various artifacts are used to start fires.

Type:	*FireStarter*
Parameters:	User
	Weather/lighting conditions
	Kindling
Action:	$\mathbf{ignite}_?(?)$
End:	burning(?)

An *f-context* is given by

- a fire-starting device o,

Example: fire starters

Various artifacts are used to start fires.

Type:	*FireStarter*
Parameters:	User
	Weather/lighting conditions
	Kindling
Action:	**ignite**$_?$(?)
End:	burning(?)

An *f-context* is given by

- a fire-starting device *o*,
- particular user *u*, weather conditions *c* and kindling *k*.

Token fulfillment

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

$$\Updownarrow$$

α is a *(weak/strong) means* to φ in context $\langle o, \tau \rangle$.

Token fulfillment

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

$$\Updownarrow$$

α is a *(weak/strong) means* to φ in context $\langle o, \tau\rangle$.

A $\mathcal{F}ire\mathcal{S}tarter$ o fulfills f wrt $\langle u, c, k\rangle$

$$\Updownarrow$$

$\textbf{ignite}_{\langle u, c, k\rangle}(o)$ realizes burning(k).

Token fulfillment

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

$$\Updownarrow$$

α is a *(weak/strong) means* to φ in context $\langle o, \tau \rangle$.

A $\mathcal{F}ire\mathcal{S}tarter$ o fulfills f wrt $\langle u, c, k \rangle$

$$\Updownarrow$$

$\mathbf{ignite}_{\langle u, c, k \rangle}(o)$ realizes burning(k).

$$\Updownarrow$$

When u ignites k via o in conditions c, kindling k burns.

Functions and subtypes

Subtypes do not always fulfill supertype functions.

Functions and subtypes

Subtypes do not always fulfill supertype functions.

$\mathcal{C}ar\mathcal{L}ighter \leq \mathcal{L}ighter \leq \mathcal{F}ire\mathcal{S}tarter$

Functions and subtypes

Subtypes do not always fulfill supertype functions.

$\mathcal{C}ar\mathcal{L}ighter \leq \mathcal{L}ighter \leq \mathcal{F}ire\mathcal{S}tarter$

- Typical lighters are good means to starting fires.

Functions and subtypes

Subtypes do not always fulfill supertype functions.

$\mathcal{CarLighter} \leq \mathcal{Lighter} \leq \mathcal{FireStarter}$

- Typical lighters are good means to starting fires.
- But car lighters do not ignite kindling easily.

Functions and subtypes

Subtypes do not always fulfill supertype functions.

$\mathcal{C}ar\mathcal{L}ighter \leq \mathcal{L}ighter \leq \mathcal{F}ire\mathcal{S}tarter$

- Typical lighters are good means to starting fires.
- But car lighters do not ignite kindling easily.

$\mathcal{L}ighter$ fulfills f, but $\mathcal{C}ar\mathcal{L}ighter$ does not fulfill f.

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

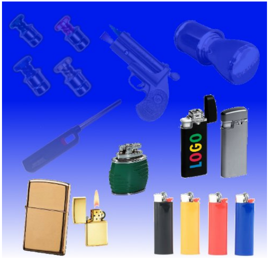
Defined: token fulfills a function f.

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!

every T-token is broken

$$\not\Downarrow$$

normal T-tokens are broken.

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

Functions and subtypes

Subtypes do not always fulfill supertype functions.

Lighter fulfills f, but *CarLighter* does not fulfill f.

Functions and subtypes

Subtypes do not always fulfill supertype functions.

Lighter fulfills f, but *CarLighter* does not fulfill f.

$$T \text{ fulfills } f \text{ and } T' \leq T$$

$$\not\Downarrow$$

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

Functions and subtypes

Subtypes do not always fulfill supertype functions.

Lighter fulfills f, but *CarLighter* does not fulfill f.

$$T \text{ fulfills } f \text{ and } T' \leq T$$

$$\xcancel{\Downarrow}$$

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

Universal fulfillment:

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

Normal fulfillment:

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

Functions and subtypes

Subtypes do not always fulfill supertype functions.

Lighter fulfills f, but *CarLighter* does not fulfill f.

$$T \text{ fulfills } f \text{ and } T' \leq T$$

$$\cancel{\Downarrow}$$

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

Universal fulfillment:

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

Normal fulfillment:

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

Normal *CarLighters* are not normal *Lighters*.

Outline

1. Means-end relations and artifactual functions
   - From functions to means-end relations
   - Means-end relations and function fulfillment

2. Efficacy and malfunction
   - Efficacy as a fuzzy property
   - Type-token comparisons and malfunction

Functions and efficacy

Different tokens can be distinguished by propensities to achieve goal.

Functions and efficacy

Different tokens can be distinguished by propensities to achieve goal.

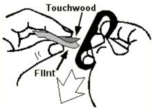
In a windy context, a Zippo may be more effective than a Bic.

Functions and efficacy

Different tokens can be distinguished by propensities to achieve goal.

In a windy context, a Zippo may be more effective than a Bic.

Both are more effective than flint and steel.



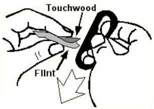
Functions and efficacy

Different tokens can be distinguished by propensities to achieve goal.

In a windy context, a Zippo may be more effective than a Bic.

Both are more effective than flint and steel.

Efficacy: the degree to which a token is reliable in fulfilling its function.



Functional efficacy requires means-end efficacy

Efficacy: the degree to which a token is reliable in fulfilling its function.

Functional efficacy requires means-end efficacy

Efficacy: the degree to which a token is reliable in fulfilling its function.

$$\text{A } \mathcal{F}\!\mathit{ire}\mathcal{S}\!\mathit{tarter}\ o \text{ fulfills } f \text{ in context } \langle u, c, k\rangle$$
$$\Updownarrow$$
$$\textbf{ignite}_{\langle u,\, c,\, k\rangle}(o) \text{ is a means to burning}(k).$$

Functional efficacy requires means-end efficacy

Efficacy: the degree to which a token is reliable in fulfilling its function.

$$\text{A } \mathcal{F}\mathit{ire}\mathcal{S}\mathit{tarter}\ o \text{ fulfills } f \text{ in context } \langle u, c, k\rangle$$

$$\Updownarrow$$

$$\mathbf{ignite}_{\langle u, c, k\rangle}(o) \text{ is a means to burning}(k).$$

Need: efficacy of means to an end.

Functional efficacy requires means-end efficacy

Efficacy: the degree to which a token is reliable in fulfilling its function.

$$\text{A } \mathcal{F}\!\mathit{ire}\mathcal{S}\!\mathit{tarter}\ o \text{ fulfills } f \text{ in context } \langle u, c, k\rangle$$
$$\Updownarrow$$
$$\textbf{ignite}_{\langle u, c, k\rangle}(o) \text{ is a means to burning}(k).$$

Need: efficacy of means to an end.

degree to which α reliably realizes φ.

Means-end relations and efficacy

Need: efficacy of means to an end.
degree to which α reliably realizes φ.

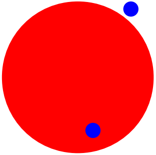
"α reliably realizes φ" is a *fuzzy* proposition.

Means-end relations and efficacy

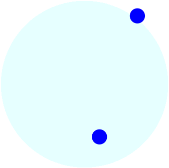
Need: efficacy of means to an end.
degree to which α reliably realizes φ.

"α reliably realizes φ" is a *fuzzy* proposition.

Fuzzy logic: the logic of vague propositions.



Crisp proposition



Fuzzy proposition

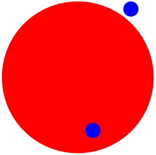
Means-end relations and efficacy

Need: efficacy of means to an end.
degree to which α reliably realizes φ.

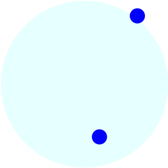
"α reliably realizes φ" is a *fuzzy* proposition.

Fuzzy logic: the logic of vague propositions.

Fuzzy semantics assigns *truth degrees* $0 \leq x \leq 1$ to formulas.



Crisp proposition



Fuzzy proposition

Means-end relations and efficacy

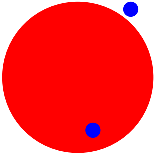
Need: efficacy of means to an end.
degree to which α reliably realizes φ.

"α reliably realizes φ" is a *fuzzy* proposition.

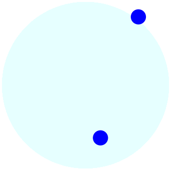
Fuzzy logic: the logic of vague propositions.

Fuzzy semantics assigns *truth degrees* $0 \leq x \leq 1$ to formulas.

Efficacy of α in φ:
Truth degree of "α reliably realizes φ."



Crisp proposition



Fuzzy proposition

Efficacy of artifact types

Token-token comparison:

Can compare efficacy of two tokens.

What about efficacy of an artifact type?

Efficacy of artifact types

Token-token comparison:

Can compare efficacy of two tokens.

What about efficacy of an artifact type?

What is the efficacy of *$\mathcal{L}$ighter* for starting fires?

Efficacy of artifact types

Token-token comparison:

Can compare efficacy of two tokens.

What about efficacy of an artifact type?

What is the efficacy of *Lighter* for starting fires?

Match?

CarLighter?

Efficacy of artifact types

Token-token comparison:
Can compare efficacy of two tokens.

What about efficacy of an artifact type?

What is the efficacy of $\mathcal{Lighter}$ for starting fires?
$\mathcal{Match}$?
$\mathcal{CarLighter}$?

f-efficacy of type T:
infimum $\{f\text{-eff. of } o \mid o \text{ is a normal } T\text{-token}\}$

Malfunction as type-token comparison

Proposal: A token malfunctions when it is ineffective.

Malfunction as type-token comparison

Proposal: A token malfunctions when it is ineffective.

Ineffective compared to what?

Malfunction as type-token comparison

Proposal: A token malfunctions when it is ineffective.

Ineffective compared to what?

Normal tokens of narrow type . . .

Malfunction as type-token comparison

Proposal: A token malfunctions when it is ineffective.

Ineffective compared to what?

Normal tokens of narrow type . . .

. . . in similar contexts.

Malfunction as type-token comparison

Proposal: A token malfunctions when it is ineffective.

Ineffective compared to what?

Normal tokens of narrow type . . .

. . . in similar contexts.

A *Lighter* malfunctions when it is ineffective at starting fires.

Malfunction as type-token comparison

Proposal: A token malfunctions when it is ineffective.

Ineffective compared to what?

Normal tokens of narrow type . . .
. . . in similar contexts.

A *Lighter* malfunctions when it is ineffective at starting fires.

A *CarLighter* malfunctions when it is less effective than normal *CarLighters*.

Concluding remarks

- Aim: Formal semantics for clarifying natural language.

Concluding remarks

- Aim: Formal semantics for clarifying natural language.
  - Relation between function and means-ends.

Concluding remarks

- Aim: Formal semantics for clarifying natural language.
  - Relation between function and means-ends.
  - Efficacy for token-token, type-type and token-type comparisons.

Concluding remarks

- Aim: Formal semantics for clarifying natural language.
  - Relation between function and means-ends.
  - Efficacy for token-token, type-type and token-type comparisons.
  - Formal definition of one kind of malfunction.

Concluding remarks

- Aim: Formal semantics for clarifying natural language.
  - Relation between function and means-ends.
  - Efficacy for token-token, type-type and token-type comparisons.
  - Formal definition of one kind of malfunction.
- Adapted PDL with fuzzy logic for "reliability".

Concluding remarks

- Aim: Formal semantics for clarifying natural language.
  - Relation between function and means-ends.
  - Efficacy for token-token, type-type and token-type comparisons.
  - Formal definition of one kind of malfunction.
- Adapted PDL with fuzzy logic for "reliability".
- Introduced *normal* tokens – but needs a philosophical analysis!

Concluding remarks

- Aim: Formal semantics for clarifying natural language.
  - Relation between function and means-ends.
  - Efficacy for token-token, type-type and token-type comparisons.
  - Formal definition of one kind of malfunction.
- Adapted PDL with fuzzy logic for "reliability".
- Introduced *normal* tokens – but needs a philosophical analysis!

Thank you.

Outline

3 Function origins

4 Extra material on normal tokens

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.

Good question, but. . .

- not easily analyzed by formal semantics;

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.

Good question, but. . .

- not easily analyzed by formal semantics;
- not immediately relevant to today's task.

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.

Good question, but. . .

- not easily analyzed by formal semantics;
- not immediately relevant to today's task.

Expectations and behavior *are* good fodder for formalization.

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.

Good question, but. . .

- not easily analyzed by formal semantics;
- not immediately relevant to today's task.

Expectations and behavior *are* good fodder for formalization.

Start with means-end relations.

Outline

3 Function origins

4 Extra material on normal tokens

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.