Semantics 1: from sentences to propositions

Semantics 1: from sentences to propositions

- Assign a new atom – a *propositional indeterminate* to every propositional demonstrative: **p** to **this**, $\mathbf{q}_n$ to $\mathbf{that}_n$. The class of Russellian propositions over this extended set of atoms is *ParPROP* (parametric propositions).

Semantics 1: from sentences to propositions

- Assign a new atom – a *propositional indeterminate* to every propositional demonstrative: **p** to **this**, $\mathbf{q}_n$ to $\mathbf{that}_n$. The class of Russellian propositions over this extended set of atoms is *ParPROP* (parametric propositions).
- The function *Val* renders parametric Russellian propositions in the parameters $\mathbf{q}_i, \mathbf{q}_j, \ldots \mathbf{q}_n$ (and **p**) to the sentence φ if φ contains the demonstratives $\mathbf{that}_i, \mathbf{that}_j, \ldots$ (and loose occurrences of **this**).

Semantics 1: from sentences to propositions

- Assign a new atom – a *propositional indeterminate* to every propositional demonstrative: **p** to **this**, $\mathbf{q}_n$ to $\mathbf{that}_n$. The class of Russellian propositions over this extended set of atoms is *ParPROP* (parametric propositions).
- The function *Val* renders parametric Russellian propositions in the parameters $\mathbf{q}_i, \mathbf{q}_j, \ldots \mathbf{q}_n$ (and **p**) to the sentence φ if φ contains the demonstratives $\mathbf{that}_i, \mathbf{that}_j, \ldots$ (and loose occurrences of **this**).
- The clauses that define *Val* are all as you would expect. The single interesting one is that for the scope operator:

Semantics 1: from sentences to propositions

- Assign a new atom – a *propositional indeterminate* to every propositional demonstrative: $\mathbf{p}$ to **this**, $\mathbf{q}_n$ to $\mathbf{that}_n$. The class of Russellian propositions over this extended set of atoms is *ParPROP* (parametric propositions).
- The function *Val* renders parametric Russellian propositions in the parameters $\mathbf{q}_i, \mathbf{q}_j, \ldots \mathbf{q}_n$ (and $\mathbf{p}$) to the sentence φ if φ contains the demonstratives $\mathbf{that}_i, \mathbf{that}_j, \ldots$ (and loose occurrences of **this**).
- The clauses that define *Val* are all as you would expect. The single interesting one is that for the scope operator:
- $Val(\downarrow \varphi)$ is the solution $p \in ParPROP$ of the equation

$$\mathbf{p} = Val\varphi(\mathbf{p}, \mathbf{q}_i, \mathbf{q}_j, \ldots)$$

.

Contexts

Contexts

- The *context* for a(n use of a) sentence is a function that assigns values to every demonstrative (in our case: propositions to every loose propositional demonstrative).

Contexts

- The *context* for a(n use of a) sentence is a function that assigns values to every demonstrative (in our case: propositions to every loose propositional demonstrative).
- The context c assigns the value $q_i \in Prop$ to the demonstrative $\mathbf{that_i}$.

Contexts

- The *context* for a(n use of a) sentence is a function that assigns values to every demonstrative (in our case: propositions to every loose propositional demonstrative).
- The context c assigns the value $q_i \in Prop$ to the demonstrative $\mathbf{that_i}$.
- The sentence φ expresses the following proposition in the context c :

$$Exp(\varphi, c) = Val(\varphi)(q_1, q_2, ...q_n)$$

Contexts

- The *context* for a(n use of a) sentence is a function that assigns values to every demonstrative (in our case: propositions to every loose propositional demonstrative).
- The context c assigns the value $q_i \in Prop$ to the demonstrative $\mathbf{that_i}$.
- The sentence φ expresses the following proposition in the context c :

$$Exp(\varphi, c) = Val(\varphi)(q_1, q_2, ...q_n)$$

- What if the sentences $\varphi_1, \varphi_2, ...\varphi_n$ refer to each other (through propositional connectives)?

Contexts for sequences of sentences

Contexts for sequences of sentences

- If we have a sequence of sentences $\varphi_1, \varphi_2, ...\varphi_n$, we want that the proposition expressed by φ_1 should be assigned to $\textbf{that}_1$, etc. (and arbitrary propositions may be assigned to demonstratives with indexes greater than n).

Contexts for sequences of sentences

- If we have a sequence of sentences $\varphi_1, \varphi_2, ...\varphi_n$, we want that the proposition expressed by φ_1 should be assigned to $\mathbf{that}_1$, etc. (and arbitrary propositions may be assigned to demonstratives with indexes greater than n).
- A *context for the sequence* $\varphi_1, \varphi_2, ...\varphi_n$ is a function that assigns q_i to $\mathbf{that}_i$ for every $i \leq n$ and arbitrary (non-parametric) propositions to every $i > n$. Let us denote the function doing this assignment for $i > n$ by c and its extension for the indexes $i \leq n$ by $(q_1, q_2, ...q_n, c)$. This latter one is a context in the previous sense

Contexts for sequences of sentences

- If we have a sequence of sentences $\varphi_1, \varphi_2, ...\varphi_n$, we want that the proposition expressed by φ_1 should be assigned to $\mathbf{that}_1$, etc. (and arbitrary propositions may be assigned to demonstratives with indexes greater than n).
- A *context for the sequence* $\varphi_1, \varphi_2, ...\varphi_n$ is a function that assigns q_i to $\mathbf{that}_i$ for every $i \leq n$ and arbitrary (non-parametric) propositions to every $i > n$. Let us denote the function doing this assignment for $i > n$ by c and its extension for the indexes $i \leq n$ by $(q_1, q_2, ...q_n, c)$. This latter one is a context in the previous sense
- By the Solution Lemma, there is an unique sequence of propositions $q_1, q_2, ...q_n$ such that for every $i \leq n$

$$Exp(\varphi_i, (q_1, q_2, ...q_n, c)) = q_i$$

Semantics 2: from propositions to truth-values

Semantics 2: from propositions to truth-values

- States of affairs (*soas*) are n-tuples of the following forms:

$$< H, a, c, i >, \ < Tr, p, i >, \ < Bel, a, p, i >$$

The class of states of affairs: *SOA*

Semantics 2: from propositions to truth-values

- States of affairs (*soas*) are n-tuples of the following forms:

$$< H, a, c, i >, \ < Tr, p, i >, \ < Bel, a, p, i >$$

  The class of states of affairs: SOA

- Situations are sub*sets* of SOA. The class of situations: SIT.

Semantics 2: from propositions to truth-values

- States of affairs (*soa*s) are n-tuples of the following forms:

$$< H, a, c, i >, \ < Tr, p, i >, \ < Bel, a, p, i >$$

  The class of states of affairs: SOA
- Situations are sub*sets* of SOA. The class of situations: SIT.
- A situation s makes true (models) ($\models$) an atomic proposition iff the witnessing *soa* is in s.

Semantics 2: from propositions to truth-values

- States of affairs (*soa*s) are n-tuples of the following forms:

$$< H, a, c, i >, \ < Tr, p, i >, \ < Bel, a, p, i >$$

  The class of states of affairs: SOA
- Situations are sub*sets* of SOA. The class of situations: SIT.
- A situation s makes true (models) ($\models$) an atomic proposition iff the witnessing *soa* is in s.
- It can be uniquely extended for compound propositions.

Semantics 2: from propositions to truth-values

- States of affairs (*soa*s) are n-tuples of the following forms:

$$< H, a, c, i >, \ < Tr, p, i >, \ < Bel, a, p, i >$$

  The class of states of affairs: SOA
- Situations are sub*sets* of SOA. The class of situations: SIT.
- A situation s makes true (models) ($\models$) an atomic proposition iff the witnessing *soa* is in s.
- It can be uniquely extended for compound propositions.
- The dual is a *soa* is the *soa* with the other i.

Models

A model $\mathfrak{M}$ is a class of *soas*.

Models

A model $\mathfrak{M}$ is a class of *soa*s.

- $\mathfrak{M}$ makes true resp. false the proposition p ($\mathfrak{M} \models p / \mathfrak{M} \not\models p$) if there is resp. there is no situation $s \subseteq \mathfrak{M}$ such that $s \models p$.

Models

A model $\mathfrak{M}$ is a class of *soas*.

- $\mathfrak{M}$ makes true resp. false the proposition p ($\mathfrak{M} \models p / \mathfrak{M} \not\models p$) if there is resp. there is no situation $s \subseteq \mathfrak{M}$ such that $s \models p$.
- p is true resp. false in $\mathfrak{M}$ iff $< Tr, p, 1 > / < Tr, p, 0 > \in \mathfrak{M}$.

Models

A model $\mathfrak{M}$ is a class of *soas*.

- $\mathfrak{M}$ makes true resp. false the proposition p ($\mathfrak{M} \models p / \mathfrak{M} \not\models p$) if there is resp. there is no situation $s \subseteq \mathfrak{M}$ such that $s \models p$.
- p is true resp. false in $\mathfrak{M}$ iff $< Tr, p, 1 > / < Tr, p, 0 > \in \mathfrak{M}$.
- $\mathfrak{M}$ is coherent iff it does not contain a *soa* and its dual together.

Models

A model $\mathfrak{M}$ is a class of *soas*.

- $\mathfrak{M}$ makes true resp. false the proposition p ($\mathfrak{M} \models p / \mathfrak{M} \not\models p$) if there is resp. there is no situation $s \subseteq \mathfrak{M}$ such that $s \models p$.
- p is true resp. false in $\mathfrak{M}$ iff $< Tr, p, 1 > / < Tr, p, 0 > \in \mathfrak{M}$.
- $\mathfrak{M}$ is coherent iff it does not contain a *soa* and its dual together.
- $\mathfrak{M}$ is a weak model iff (it is coherent and if a proposition p is true resp. false in $\mathfrak{M}$ the $\mathfrak{M}$ makes p true resp. false).