19. Three system types of semantics

19.1 Basic structure of semantic interpretation

19.1.1 The 2-level structure of semantic interpretation

LEVEL I: language surface

$\updownarrow$ syntactic-semantic ASSIGNMENT ALGORITHM

LEVEL II: semantic content

19.1.2 The function of semantic interpretation

For purposes of transmission and storage, semantic content is coded into surfaces of language (representation). When needed, the content may be decoded by analyzing the surface (reconstruction).

The expressive power of semantically interpreted languages resides in the fact that representing and reconstructing are realized *automatically*: a semantically interpreted language may be used correctly without the user having to be conscious of these procedures, or even having to know or understand their details.

19.2 Logical, programming, and natural languages

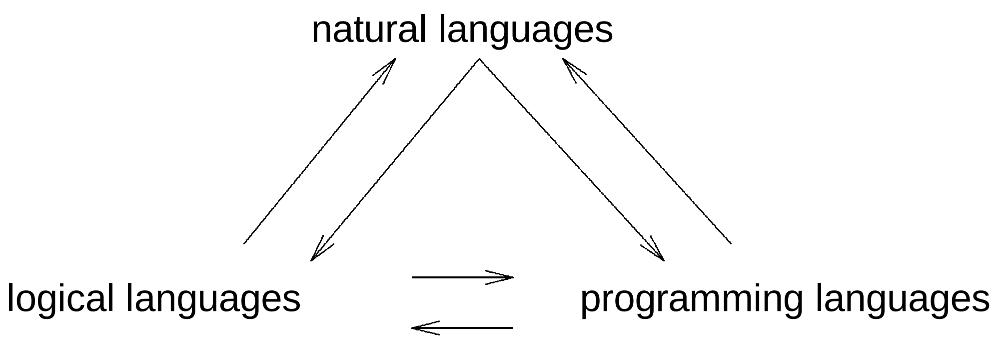
19.2.1 Three different types of semantic systems

1. *Logical languages*
   Designed to determine the truth value of arbitrary propositions relative to arbitrary models. The correlation between the two levels is based on *metalanguage definitions*.
2. *Programming languages*
   Designed to simplify the interaction with computers and the development of software. The correlation between the two levels is based on the *procedural execution* on an abstract machine, usually implemented electronically.
3. *Natural languages*
   Preexisting in the language community, they are analyzed syntactically by reconstructing the combinatorics of their surfaces. The associated semantic representations have to be deduced via the general principles of natural communication. The correlation between the two levels is based on *conventional association*.

19.2.2 Three types of semantic interpretation

	logical languages	**programming languages**	**natural languages**
LEVEL I:	propositions	commands	surfaces
	metalanguage definition	procedural execution	conventional association
LEVEL II:	*set theoretic model of the 'world'*	*operations on an abstract machine*	*literal meanings used by the speaker-hearer relative to the internal context*

19.2.3 Mapping relations between the three types of semantics



19.2.4 Characterizing the mapping relations: Replication, Reconstruction, Transfer, and Combination

- *Replication*
  Selected natural language phenomena are replicated in logical languages (N→L). Selected aspects of logical languages are replicated procedurally in programming languages like LISP and Prolog (L→P). The programming languages also replicate natural language concepts directly, e.g. 'command' (N→P).
- *Reconstruction*
  Theoretical linguistics attempts to reconstruct fragments of natural language in terms of logic (L→N). Computational linguistic aims at reconstructing natural languages by means of programming languages (P→N). One may also imagine a reconstruction of programming concepts in a new logical language (P→L).
- *Transfer*
  Computer science attempts to transfer methods and results of logical proof theory into the programming languages (L→P). Philosophy of language attempts to transfer the model-theoretic method to the semantic analysis of natural language (L→N).
- *Combination*
  Computational linguistics aims at modeling natural communication with the help of programming languages (P→N). Thereby methods and results of the logical languages play a role in both, the construction of programming languages (L→P) and the analysis of natural language (L→N). This requires a functional overall framework for combining the three types of language in a way that utilizes their different properties while avoiding redundancy as well as conflict.

19.3 Functioning of logical semantics

19.3.1 Interpretation of a proposition

LEVEL I	logical language:	*sleep (Julia)*
LEVEL II	world (model):	o o o o o o

19.3.2 Definition of a minimal logic

1. **Lexicon**

   Set of one-place predicates: {sleep, sing}
   Set of names: {Julia, Susanne}

2. **Model**
   A model $\mathcal{M}$ is a two-tuple (A, F), where A is a non-empty set of entities and F a denotation function (see 3).

3. **Possible Denotations**
   (a) If P_1 is a one-place predicate, then a possible denotation of P_1 relative to a model $\mathcal{M}$ is a subset of A. Formally, $F(P_1)^{\mathcal{M}} \subseteq A$.
   (b) If α is a name, then the possible denotations of α relative to a model $\mathcal{M}$ are elements of A. Formally, $F(\alpha)^{\mathcal{M}} \in A$.
   (c) If ϕ is a sentence, then the possible denotations of ϕ relative to a model $\mathcal{M}$ are the numbers 0 and 1, interpreted as the truth values 'true' and 'false.' Formally, $F(\phi)^{\mathcal{M}} \in \{0,1\}$.

   Relative to a model $\mathcal{M}$ a sentence ϕ is a true sentence, if and only if the denotation ϕ in $\mathcal{M}$ is the value 1.

4. **Syntax**
   (a) If P_1 is a one-place predicate and α is a name, then $P_1(\alpha)$ is a sentence.
   (b) If ϕ is a sentence, then $\neg\phi$ is a sentence.
   (c) If ϕ is a sentence and ψ is a sentence, then ϕ & ψ is a sentence.

(d) If ϕ is a sentence and ψ is a sentence, then $\phi \vee \psi$ is a sentence.
(e) If ϕ is a sentence and ψ is a sentence, the $\phi \rightarrow \psi$ is a sentence.
(f) If ϕ is a sentence and ψ is a sentence, then $\phi = \psi$ is a sentence.

5. **Semantics**

(a) '$P_1(\alpha)$' is a true sentence relative to a model $\mathcal{M}$ if and only if the denotation of α in $\mathcal{M}$ is element of the denotation of P_1 in $\mathcal{M}$.
(b) '$\neg\, \phi$' is a true sentence relative to a model $\mathcal{M}$ if and only if the denotation of ϕ is 0 relative to $\mathcal{M}$.
(c) 'ϕ & ψ' is a true sentence relative to a model $\mathcal{M}$ if and only if the denotations of ϕ and of ψ are 1 relative to $\mathcal{M}$.
(d) '$\phi \vee \psi$' is a true sentence relative to a model $\mathcal{M}$ if and only if the denotation of ϕ or ψ is 1 relative to $\mathcal{M}$.
(e) '$\phi \rightarrow \psi$' is a true sentence relative to a model $\mathcal{M}$ if and only if the denotation of ϕ relative to $\mathcal{M}$ is 0 or the denotation of ψ is 1 relative to $\mathcal{M}$.
(f) '$\phi = \psi$' is a true sentence relative to a model $\mathcal{M}$ if- and only if the denotation of ϕ relative to $\mathcal{M}$ equals the denotation of ψ relative to $\mathcal{M}$.

19.3.3 Schema of Tarski's T-condition

T: x is a true sentence if and only if p.

19.3.4 Instantiation of Tarski's T-condition

'Es schneit' is a true sentence if and only if it snows.

19.3.5 Relation between object and metalanguage

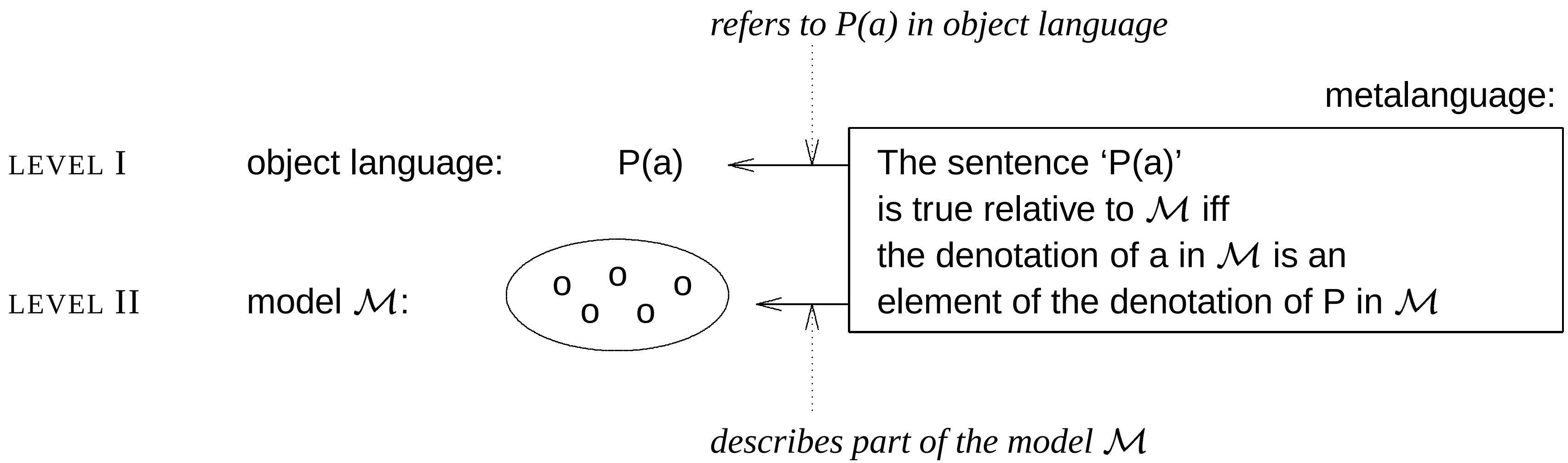
metalanguage:

LEVEL I object language: Es schneit. ← 'Es schneit.'

is a true sentence
if and only if

LEVEL II world: *state of snowing* ← it snows.

19.3.6 T-condition in a logical definition



19.3.7 The appeal to immediate obviousness in mathematics

En l’un les principes sont palpables mais éloignés de l’usage commun de sorte qu’on a peine à tourner late tête de ce côte-la, manque d’habitude : mais pour peu qu’on l’y tourne, on voit les principes à peine; et il faudrait avoir tout à fait l’esprit faux pour mal raisonner sur des principes si gros qu’il est presque impossible qu’ils échappent.

[In [the mathematical mind] the principles are obvious, but remote from ordinary use, such that one has difficulty to turn to them for lack of habit : but as soon as one turns to them, one can see the principles in full; and it would take a thoroughly unsound mind to reason falsely on the basis of principles which are so obvious that they can hardly be missed.]

B. PASCAL (1623 -1662), *Pensées*, 1951:340

19.4 Metalanguage-based versus procedural semantics

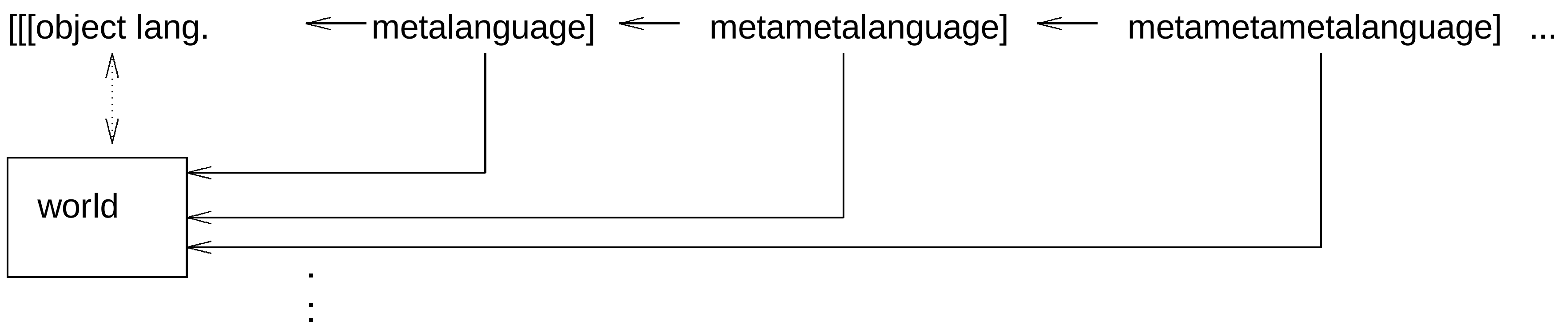
19.4.1 Example of a vacuous T-condition

‘A is red’ is a true sentence if and only if A is red.

19.4.2 Improved T-condition for red

‘A is red’ is a true sentence if and only if A refracts light in the electromagnetic frequency interval between α and β.

19.4.3 Hierarchy of metalanguages



19.4.4 Autonomy from the metalanguage

Autonomy from the metalanguage does not mean that computers would be limited to uninterpreted, purely syntactic deduction systems, but rather that Tarski's method of semantic interpretation is not the only one possible. Instead of assigning semantic representations to an object language by means of a metalanguage, computers use an operational method in which the notions of the programming language are realized automatically as machine operations.

19.4.5 Example of autonomy from metalanguage

There is no problem to provide an adequate metalanguage definition for the rules of basic addition, multiplication, etc. However, the road from such a metalanguage definition to a working calculator is quite long and in the end the calculator will function mechanically – without any reference to these metalanguage definitions and without any need to understand the metalanguage.

19.4.6 Programming logical systems

There exist many logical calculi which have not been and never will be realized as computer programs. The reason is that their metalanguage translations contain parts which may be considered immediately obvious by their designers (e.g. quantification over infinite sets of possible worlds in modal logic), but which are nevertheless unsuitable to be realized as empirically meaningful mechanical procedures.

19.5 Tarski's problem for natural language semantics

19.5.1 Logical semantics for natural language?

> The attempt to set up a structural definition of the term 'true sentence' – applicable to colloquial language – is confronted with insuperable difficulties.
>
> A. Tarski 1935, p. 164.

19.5.2 Tarski's proof

For the sake of greater perspicuity we shall use the symbol 'c' as a typological abbreviation of the expression 'the sentence printed on page 355 , line 8 from the bottom.' Consider now the following sentence:

c is not a true sentence

Having regard to the meaning of the symbol 'c', we can establish empirically:
(a) 'c is not a true sentence' is identical with c.
For the quotation-mark name of the sentence c we set up an explanation of type (2) [i.e. the T-condition 19.3.3]:
(b) 'c is not a true sentence' is a true sentence if and only if c is not a true sentence.
The premise (a) and (b) together at once give a contradiction:
c is a true sentence if and only if c is not a true sentence.

19.5.3 Inconsistent T-condition using Epimenides paradox

c	is	true	iff	c	is not	true
substitution based on abbreviation		*'true' from metalanguage T-condition*		*metalang. correlate to object language statement 'c'*		*metalang. correlate to object language word 'true'*

19.5.4 Three options for avoiding Tarski's contradiction in logical semantics

1. Forbidding the abbreviation and the substitution based on it. This possibility is rejected by Tarski because "no rational ground can be given why substitution should be forbidden in general."
2. Distinguishing between the truth predicate true^m of the metalanguage and true^o of the object language. On this approach

   c is true^m if and only if c is not true^o

   is not contradictory because true^m $\neq$ true^o.
3. This option, chosen by Tarski, consists in forbidding the use of truth predicates in the object language.

19.5.5 Reasons for the third option

If the goal is to characterize scientific theories like physics as true relations between logical propositions and states of affairs, then the vagueness and contradictions of the natural languages must be avoided – as formulated by G. Frege 1896:

> Der Grund, weshalb die Wortsprachen zu diesem Zweck [d.h. Schlüsse nur nach rein logischen Gesetzen zu ziehen] wenig geeignet sind, liegt nicht nur an der vorkommenden Vieldeutigkeit der Ausdrücke, sondern vor allem in dem Mangel fester Formen für das Schließen. Wörter wie $>$also$<$, $>$folglich$<$, $>$weil$<$ deuten zwar darauf hin, daß geschlossen wird, sagen aber nichts über das Gesetz, nach dem geschlossen wird, und können ohne Sprachfehler auch gebraucht werden, wo gar kein logisch gerechtfertigter Schluß vorliegt.
>
> [The reason why the word languages are suited little for this purpose [i.e., draw inferences based on purely logical laws] is not only the existing ambiguity of the expressions, but mainly the lack of clear forms of inference. Even though words like ‘therefore,’ ‘consequently,’ ‘because’ indicate inferencing, they do not specify the rule on which the inference is based and they may be used without violating the well-formedness of the language even if there is no logically justified inference.]

The goal of characterizing scientific truth precludes reconstructing the object language as a natural language. Therefore there is no need for a truth predicate in the object language – which is in line with Tarski’s option.

19.5.6 Reasons against the third option

If the goal is to apply logical semantics to natural language, then the third option poses a serious problem. This is because natural language as the pretheoretical metalanguage *must* contain the words true and false. Therefore a logical semantic interpretation of a natural (object-)language in its entirety will unavoidably result in a contradiction.

19.5.7 Montague's choice: Ignoring the problem

> I reject the contention that an important theoretical difference exists between formal and natural languages. ... Like Donald Davidson I regard the construction of a theory of truth – or rather the more general notion of truth under an arbitrary interpretation – as the basic goal of serious syntax and semantics.
>
> R. Montague 1970

19.5.8 Davidson's choice: Suspending the problem

> Tarski's ... point is that we should have to reform natural language out of all recognition before we could apply formal semantic methods. If this is true, it is fatal to my project.
>
> D. Davidson 1967