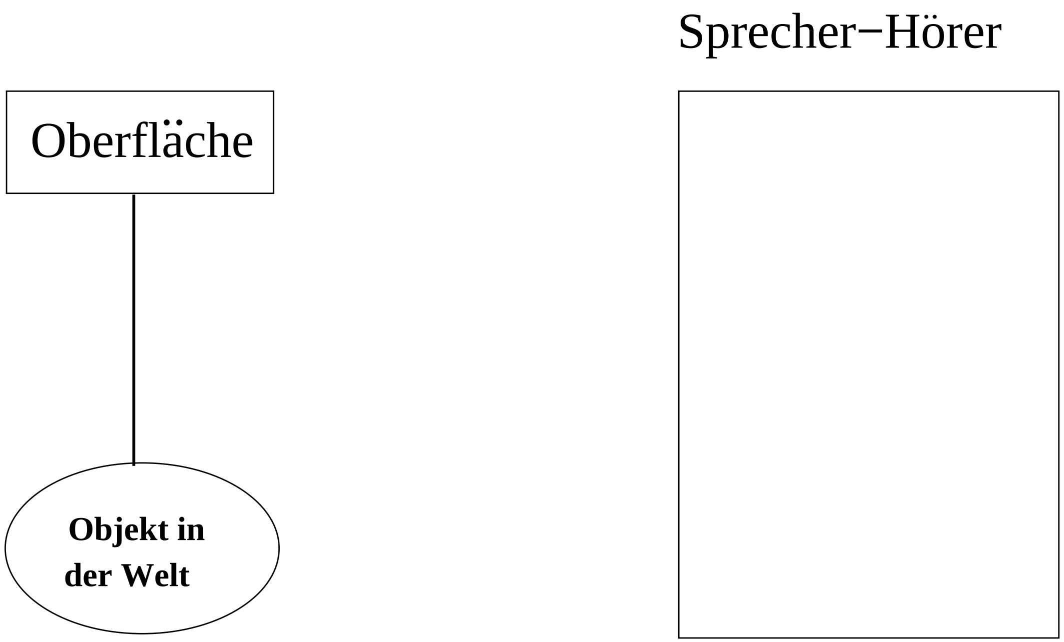
19. Die Drei Typen der Semantik

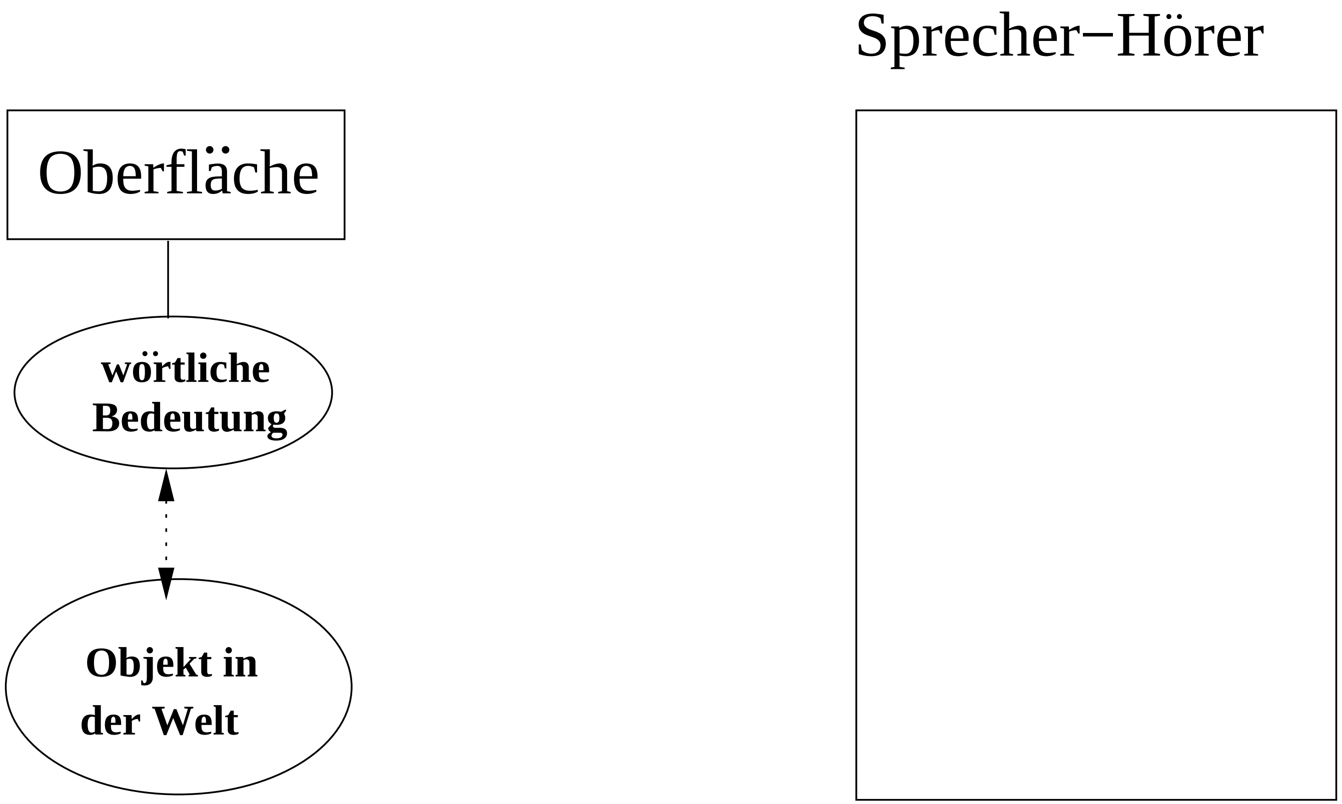
Vortheoretische Überlegungen

Externes Zeichen ohne Bedeutung



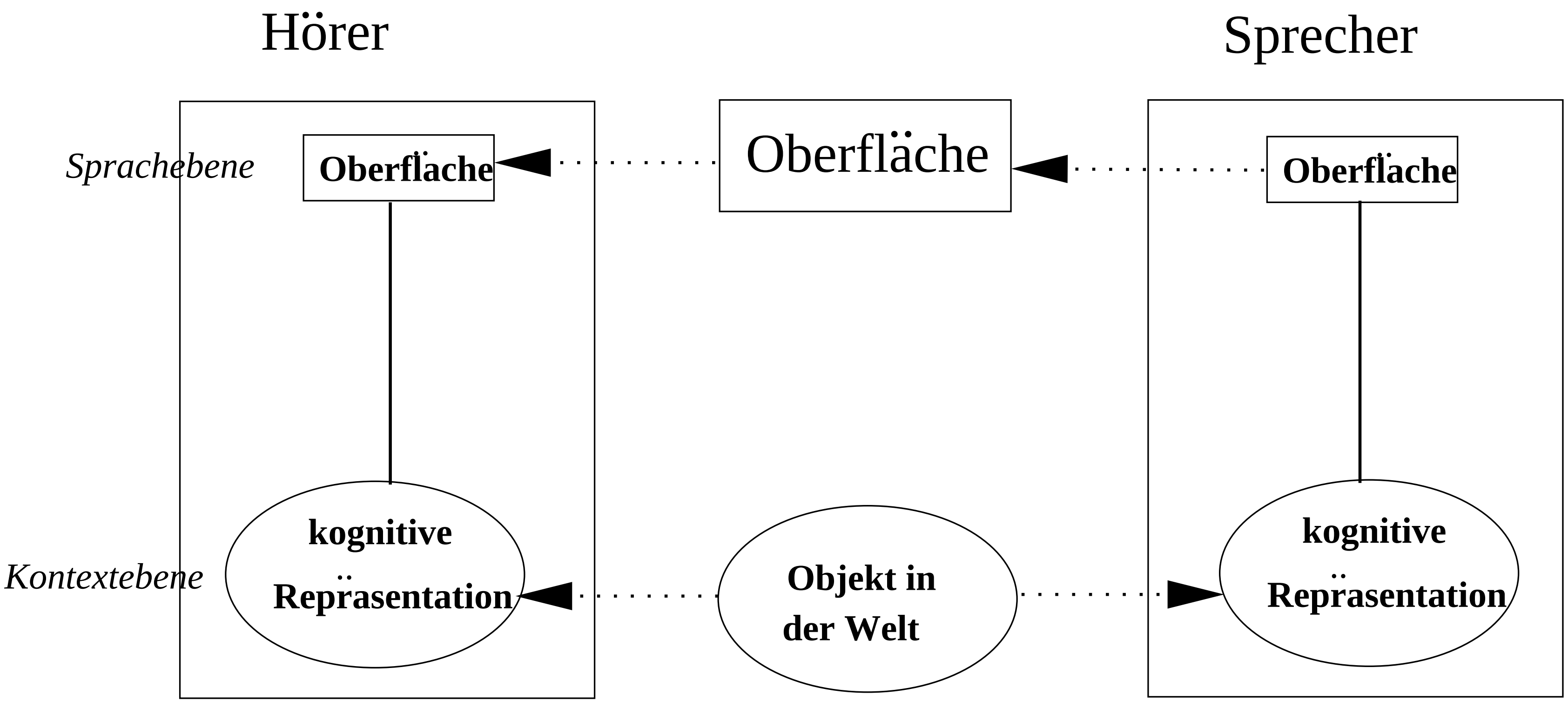
Sprecher-Hörer beobachtet externe Beziehung zwischen Zeichenoberfläche und Welt
Keine interne kognitive Struktur von Sprecher-Hörer
Referent in der Welt dient als Zeichenbedeutung
Auf geschlossene Welten beschränkt

Externes Zeichen mit Bedeutung



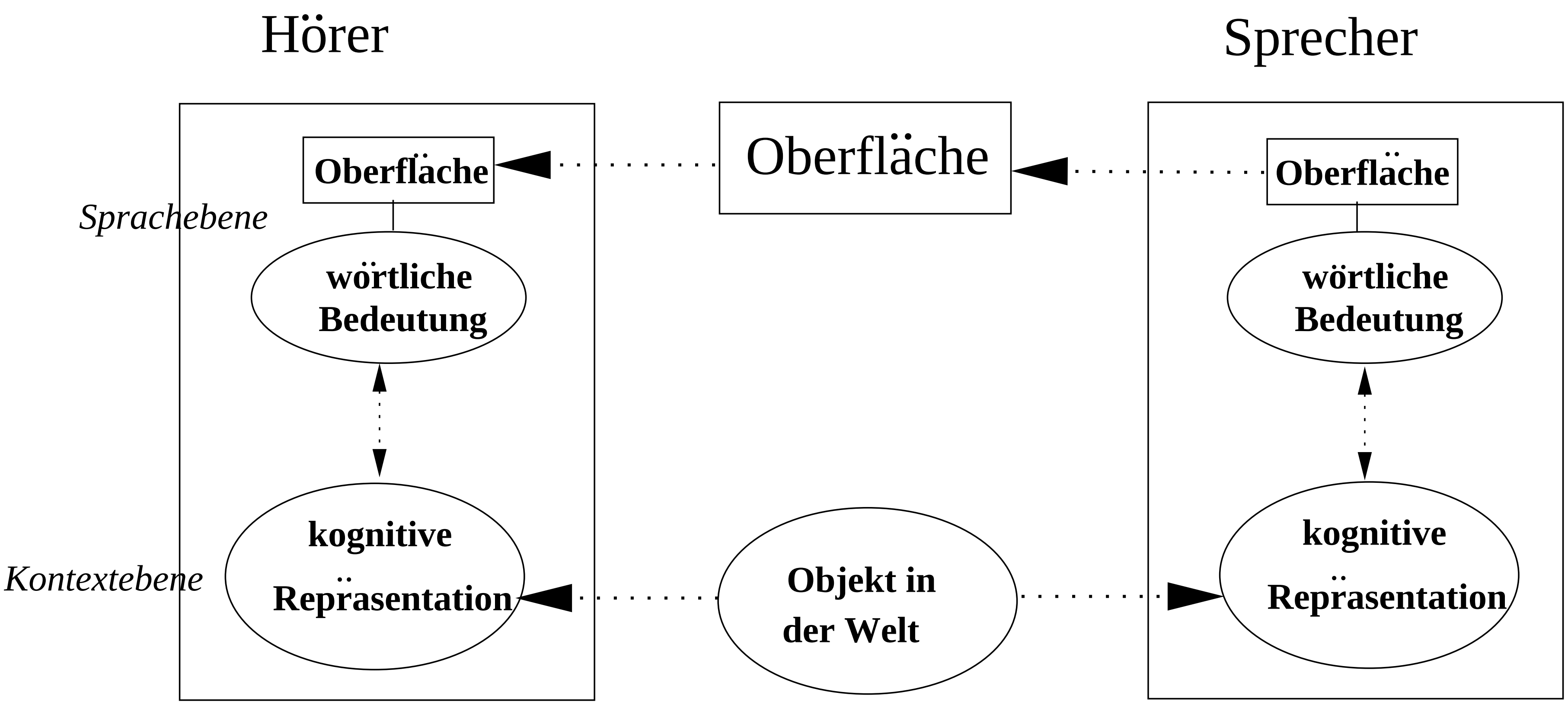
Sprecher-Hörer beobachtet externe Beziehung zwischen Zeichen, Bedeutung und Welt
Keine interne kognitive Struktur von Sprecher-Hörer
Unterscheidung von Zeichenbedeutung und Referent in der Welt

Internes Zeichen ohne Bedeutung



Keine externe Beziehung zwischen Zeichenoberfläche und Welt
Einfache interne kognitive Struktur von Sprecher-Hörer
Kognitive Repräsentation des Referenten dient als Bedeutung des internen Zeichens
Auf geschlossene Welten beschränkt

Internes Zeichen mit Bedeutung



Keine externe Beziehung zwischen Zeichen und Welt
Interne kognitive Struktur von Sprecher-Hörer

19.1 Grundstruktur der semantischen Interpretation

19.1.1 Unterschiedliche Verwendungen des Begriffs *Semantik*

- Linguistik: Komponente der Grammatik, die sprachlichen Ausdrücken Bedeutungen zuweist.
- Philosophie: Mengentheoretische Strukturen, die den Formeln einer Logiksprache zugeordnet werden, um (i) den Begriff der Wahrheit zu charakterisieren und (ii) formale Beweise zu ermöglichen.
- Informatik: Umsetzung sprachlicher Oberflächen (Befehle) als Operationen auf der Maschinenebene.

19.1.2 Schema der Zwei-Ebenenstruktur der semantischen Interpretation

1. EBENE: sprachliche Oberfläche

$\updownarrow$ syntaktisch-semantischer ZUORDNUNGSALGORITHMUS

2. EBENE: semantischer Inhalt

19.1.3 Funktionen der semantischen Interpretation

Zum Zweck der Übermittlung oder Speicherung wird semantischer Inhalt in eine sprachliche Oberfläche encodiert (Representation). Im Bedarfsfall wird dieser Inhalt durch eine Analyse der Oberfläche wieder decodiert (Rekonstruktion).

Die Ausdrucksstärke von semantisch interpretierten Sprachen liegt darin, daß sowohl die Repräsentation als auch die Rekonstruktion *automatisch* geschehen: Eine semantisch interpretierte Sprache kann korrekt verwendet werden, ohne daß dem Benutzenden diese Prozeduren bewußt sind oder er ihre Details kennt und versteht.

19.2 Logiksprachen, Programmiersprachen, natürliche Sprachen

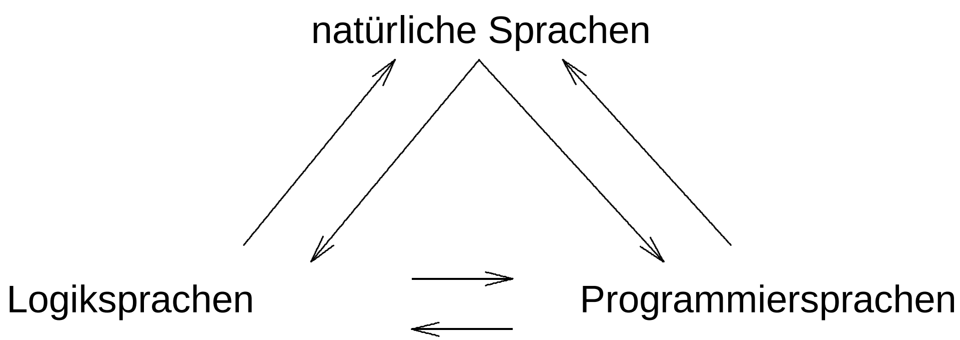
19.2.1 Drei unterschiedliche Systemtypen der Semantik

1. *Logiksprachen*
   Entstanden im Rahmen der Philosophie. Wurden entwickelt, um die Wahrheit beliebiger Propositionen in Bezug auf beliebige Modelle der Welt zu charakterisieren. Sie haben eine *metasprachliche Semantik*, weil die Korrelation zwischen den beiden Ebenen auf dem Prinzip der metasprachlichen Definition beruht.
2. *Programmiersprachen*
   Sie entstanden aus dem praktischen Bedürfnis, den Umgang mit Computern und die Entwicklung von Programmen zu vereinfachen. Sie haben eine *prozedurale Semantik*, weil die Korrelation zwischen den beiden Ebenen auf dem Prinzip der Ausführung beruht, d.h. der operationalen Umsetzung von Befehlen auf einer abstrakten Maschine, die meist elektronisch realisiert ist.
3. *Natürliche Sprachen*
   Entwickeln sich auf natürliche Weise in ihrer jeweiligen Sprachgemeinschaft. Linguisten analysieren die vorgegebenen natürlichen Sprachen syntaktisch, indem sie die Kombinatorik der Oberflächen explizit als generative Grammatiken rekonstruieren. Die zugehörigen semantischen Repräsentationen müssen über allgemeine Prinzipien der natürlichsprachlichen Kommunikation erschlossen werden, weil die Meaning$_1$ im Unterschied zur Oberfläche keine konkrete externe Manifestation hat. Natürliche Sprachen haben eine *konventionsbasierte Semantik*, weil die Korrelation zwischen den beiden Ebenen auf Konventionen innerhalb der Sprachgemeinschaft beruht.

19.2.2 Drei Arten der semantischen Interpretation

	Logiksprachen	**Programmiersprachen**	**Natürliche Sprachen**
1. Ebene:	Propositionen	Befehle	natürliche Oberflächen
	metasprachliche Definition	prozedurale Ausführung	konventionelle Zuordnung
2. Ebene:	*mengentheoretisches Modell der Welt*	*Operationen auf einer abstrakten Maschine*	*wörtl. Bedeutungen, die vom Sprecher/Hörer relativ zum Interpretationskontext verwendet werden*

19.2.3 Abbildungsrelationen zwischen den drei Arten der Semantik



19.2.4 Charakterisierung der Abbildungen

- *Nachbildung*
  Die Logiksprachen entwickelten sich ursprünglich als künstliche Nachbildung ausgewählter Phänomene der natürlichen Sprachen (N⟶L). Programmiersprachen wie LISP und Prolog wiederum entstanden als prozedurale Nachbildung von Teilaspekten bestimmter Logiksprachen (L⟶P). Die Programmiersprachen haben auch Phänomene der natürlichen Sprachen direkt nachgebildet, z. B. den *Befehl* (N⟶P).
- *Rekonstruktion*
  Wenn sich ein neuer künstlicher Sprachtyp etabliert und verselbständigt hat, ergibt sich die Möglichkeit, die Ausgangssprache zumindest teilweise mit den Methoden der neuen Sprache zu rekonstruieren. So bemüht sich die theoretische Linguistik, mit logischen Methoden künstliche Fragmente der natürlichen Sprachen zu rekonstruieren (L⟶N). Die Computerlinguistik hat zum Ziel, natürliche Sprachen mit Hilfe von Programmiersprachen zu rekonstruieren (P⟶N). Eine Rekonstruktion programmiersprachlicher Konstrukte in einer neuen Logiksprache ist ebenfalls denkbar (P⟶L).
- *Übertragung*
  Der gezielte Versuch, Beweismethoden und Ergebnisse aus einer künstlichen Sprache in eine andere zu übertragen, hat sich insbesondere bei der Anwendung logischer Ergebnisse auf die Programmiersprachen als sehr erfolgreich erwiesen (L⟶P). Eine allgemeine Übertragung ist jedoch aufgrund der unterschiedlichen Methoden, Strukturen und Ziele dieser beiden Semantiksysteme nicht möglich. Auch in der Sprachphilosophie war der (L⟶N) Versuch, die Semantik der Logiksprachen auf möglichst alle Phänomene der natürlichen Sprachen zu erweitern, nur teilweise erfolgreich.

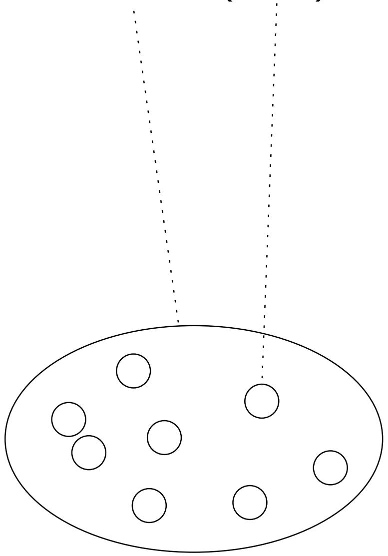
- *Kombinierbarkeit*

  Die Computerlinguistik hat das Ziel, natürliche Sprachen mit Hilfe von Programmiersprachen zu modellieren (P$\longrightarrow$N), wobei Methoden und Ergebnisse der Logiksprachen sowohl für die Konstruktion der Programmiersprachen (L$\longrightarrow$P) als auch für die Analyse der natürlichen Sprachen (L$\longrightarrow$N) berücksichtigt werden. Hierfür muß eine theoretische Struktur gefunden werden, die es erlaubt, die drei Sprachsysteme in einem einheitlichen funktionalen Rahmen zu kombinieren. Dabei sollten die unterschiedlichen Eigenschaften der verschiedenen Systeme einerseits optimal genutzt, andererseits aber Konflikte oder Redundanzen vermieden werden.

19.3 Funktionsweise der logischen Semantik

19.3.1 Interpretation einer Proposition

1. Ebene: Logik-Sprache schlafen (Julia)



2. Ebene: Welt (Modell)

19.3.2 Definition einer minimalen Logik

1. **Lexikon**

   Menge der einstelligen Prädikate: {schlafen, singen}
   Menge der Namen: {Julia, Susanne}

2. **Modell**
   Ein Modell $\mathcal{M}$ ist ein Zweitupel (A, F), wobei A eine nichtleere Menge von Entitäten und F eine Denotationsfunktion ist (siehe 3).

3. **Mögliche Denotationen**
   (a) Wenn P_1 ein einstelliges Prädikat ist, dann sind die möglichen Denotationen von P_1 in bezug auf ein Modell $\mathcal{M}$ eine Untermenge von A. Formal: $F(P_1)^{\mathcal{M}} \subseteq A$.
   z.B: F(schläft)$^{\mathcal{M}}$ = $\{a_1, a_2, a_3\} \subseteq A = \{a_1, a_2, a_3, a_4, a_5, a_6\}$
   (b) Wenn α ein Name ist, dann sind die möglichen Denotationen von α in bezug auf ein Modell $\mathcal{M}$ Elemente von A. Formal: $F(\alpha)^{\mathcal{M}} \in A$.
   z.B: F(Julia)$^{\mathcal{M}}$ = a_1
   (c) Wenn ϕ ein Satz ist, dann sind die möglichen Denotationen von ϕ in bezug auf ein Modell $\mathcal{M}$ die Zahlen 0 und 1, die als Wahrheitswerte *falsch* und *wahr* interpretiert werden. Formal: $F(\phi)^{\mathcal{M}} \in \{0,1\}$.

   In bezug auf ein Modell $\mathcal{M}$ ist ein Satz ϕ ein wahrer Satz dann und nur dann, wenn das Denotat von ϕ in $\mathcal{M}$ der Wert 1 ist.

4. **Syntax**
   (a) Wenn P_1 ein einstelliges Prädikat und α ein Name ist, dann ist $\mathrm{P}_1(\alpha)$ ein Satz.
   (b) Wenn ϕ ein Satz ist, dann ist auch $\neg\phi$ ein Satz.
   (c) Wenn ϕ ein Satz ist und ψ ein Satz ist, dann ist ϕ & ψ ein Satz.
   (d) Wenn ϕ ein Satz ist und ψ ein Satz ist, dann ist $\phi \vee \psi$ ein Satz.
   (e) Wenn ϕ ein Satz ist und ψ ein Satz ist, dann ist $\phi \rightarrow \psi$ ein Satz.
   (f) Wenn ϕ ein Satz ist und ψ ein Satz ist, dann ist $\phi = \psi$ ein Satz.

5. **Semantik**
   (a) In bezug auf ein Modell $\mathcal{M}$ ist '$\mathrm{P}_1(\alpha)$' ein wahrer Satz dann und nur dann, wenn das Denotat von α in $\mathcal{M}$ Element des Denotats von P_1 in $\mathcal{M}$ ist.
   (b) In bezug auf ein Modell $\mathcal{M}$ ist '$\neg\ \phi$' ein wahrer Satz dann und nur dann, wenn das Denotat von ϕ in bezug auf $\mathcal{M}$ gleich 0 ist.
   (c) In bezug auf ein Modell $\mathcal{M}$ ist 'ϕ & ψ' ein wahrer Satz dann und nur dann, wenn die Denotate von ϕ und von ψ in bezug auf $\mathcal{M}$ gleich 1 sind.
   (d) In bezug auf ein Modell $\mathcal{M}$ ist '$\phi \vee \psi$' ein wahrer Satz dann und nur dann, wenn das Denotat von ϕ oder von ψ in bezug auf $\mathcal{M}$ gleich 1 ist.
   (e) In bezug auf ein Modell $\mathcal{M}$ ist '$\phi \rightarrow \psi$' ein wahrer Satz dann und nur dann, wenn das Denotat von ϕ in bezug auf $\mathcal{M}$ gleich 0 ist oder das Denotat von ψ in bezug auf $\mathcal{M}$ gleich 1 ist.
   (f) In bezug auf ein Modell $\mathcal{M}$ ist '$\phi = \psi$' ein wahrer Satz dann und nur dann, wenn das Denotat von ϕ in bezug auf $\mathcal{M}$ gleich dem Denotat von ψ in bezug auf $\mathcal{M}$ ist.

19.3.3 Schema von Tarskis T-Bedingung

T: x ist ein wahrer Satz dann und nur dann, wenn p.

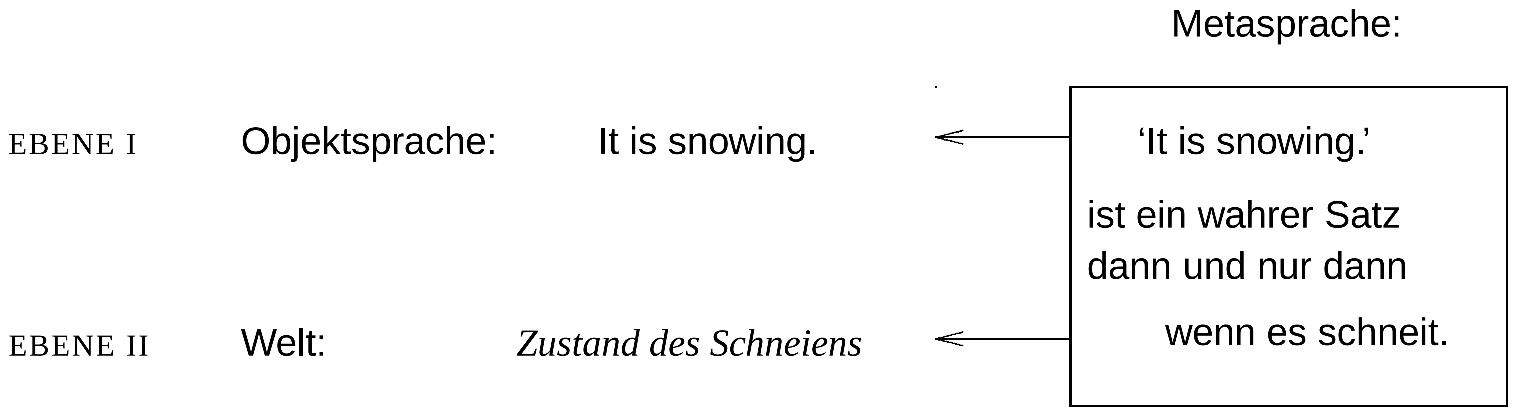
Die T-Bedingung als Ganzes ist ein Satz der Metasprache, der den Satz x der Objektsprache zitiert und als p *übersetzt*. Tarski illustriert seine Methode mit dem folgenden Beispiel:

19.3.4 Instantiierung von Tarskis T-Bedingung

'It is snowing' ist ein wahrer Satz dann und nur dann, wenn es schneit.

Dieses Beispiel ist von trügerischer Einfachheit, die zu Mißverständnissen geführt hat. Was 19.3.3 und 19.3.4 in ihrer provokativen Simplizität für sich allein betrachtet nicht zum Ausdruck bringen, ist der genaue Charakter der *Zwei-Ebenenstruktur* (siehe 19.1.2). Diese liegt allen Formen der semantischen Interpretationen zugrunde und wird daher auch von Tarskis spezieller Methode exemplifiziert.

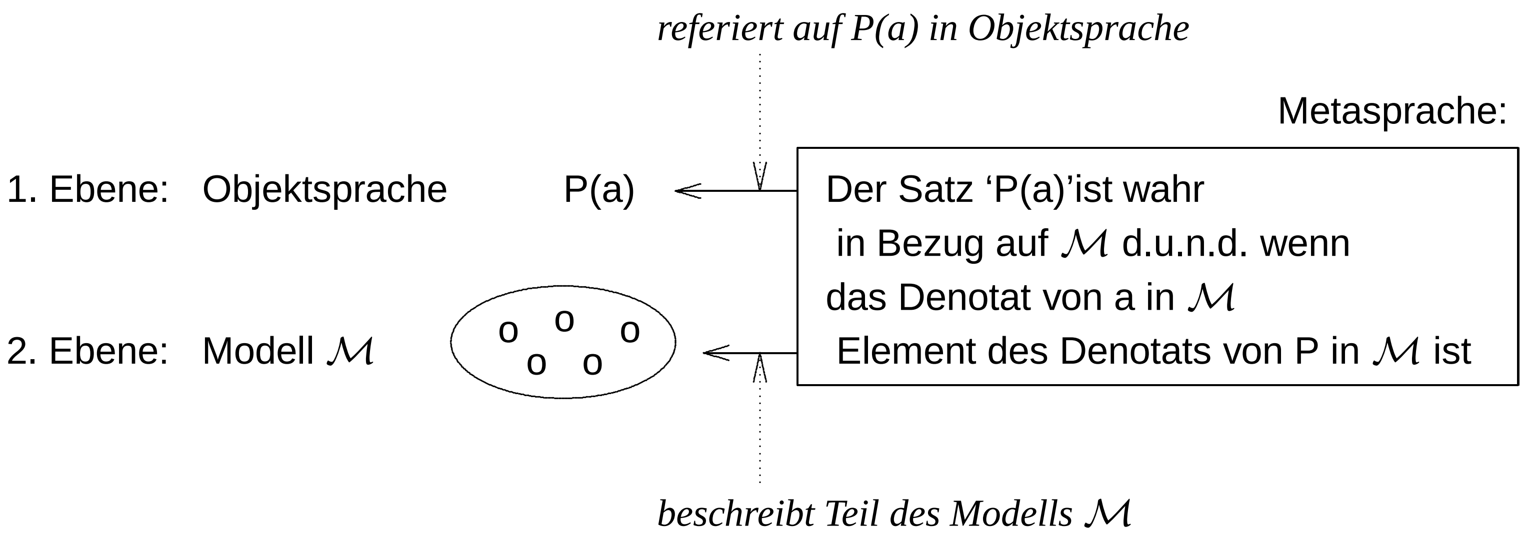
19.3.5 Verhältnis von Objekt- und Metasprache



Der direkte Bezug der Metasprache auf die Welt wird *Verifikation* genannt. Die Verifikation von T besteht in der praktischen Möglichkeit, tatsächlich festzustellen ob p zutrifft oder nicht.

19.3.6 T-Bedingung bei logischer Definition

Die Möglichkeit, diese T-Bedingung zu verifizieren, wird durch nicht mehr und nicht weniger als die Tatsache garantiert, daß bei jedem beliebigen Modell $\mathcal{M}$ jeder Sprecher des Deutschen mit einer minimalen Kenntnis der Mengentheorie in der Lage ist, zu *sehen* (im Sinne der unmittelbaren Anschauung), ob die im Übersetzungsteil von T angegebene Relation von $\mathcal{M}$ erfüllt wird oder nicht.



19.3.7 Die “unmittelbare Anschauung” in der Mathematik

Die Berufung auf die unmittelbare Anschauung ist in der Geschichte der Mathematik schon immer als die letzte Instanz herangezogen worden.

> 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.
>
> [Im mathematischen Sinn sind diese Prinzipien offensichtlich, aber sie werden normalerweise nicht verwendet. Durch diesen Mangel an Gewohnheit hat man Schwierigkeiten sich ihnen zuzuwenden: Aber sobald man sich ihnen zuwendet, werden diese Prinzipien ganz offenbar und es bräuchte einen ganz und gar verwirrten Geist, um auf der Basis dieser offensichtlichen Prinzipien falsche Schlüsse zu ziehen.]
>
> B. PASCAL (1623 -1662), *Pensées*, 1951:340

19.4 Metasprachliche oder prozedurale Semantik?

19.4.1 Konstruktion der Metasprache

Tarski fordert von der Metasprache, daß (i) alle Zeichen und Ausdrücke explizit aufgezählt werden und (ii) jedes Zeichen und jeder Ausdruck der Metasprache eine klare Bedeutung hat (*has a clear meaning*. (Tarski 1935., S. 172).

19.4.2 Tarski's Beispiel einer Metasprache

Tarskis Beispiel ist der Klassenkalkül. Die einzigen Ausdrücke, die Tarski hierbei benutzt, sind Begriffe wie nicht, und, ist enthalten in, ist Element von, Individuum, Klasse und Relation. Die Bedeutung dieser Ausdrücke ist insofern unmittelbar klar und verständlich, als es sich ausschließlich um mathematische Objekte und mengentheoretische Operationen handelt. Tarski beschränkt also seine Metasprache auf die Elementarbegriffe einer vollentwickelten (Meta)theorie, in diesem Fall einen Teilbereich der Grundlagenmathematik.

19.4.3 Parallele zwischen Logiksprachen und Programmiersprachen

So wie Tarski von einer Metasprache fordert, dass sie nur *unmittelbar offensichtliche* Begriffe und Operationen enthalten darf, ist es für eine Programmiersprache erforderlich, dass sie nur *operationalisierbare* Begriffe und Prozeduren enthält.

Allerdings: unmittelbar offensichtlich $\neq$ operationalisierbar.

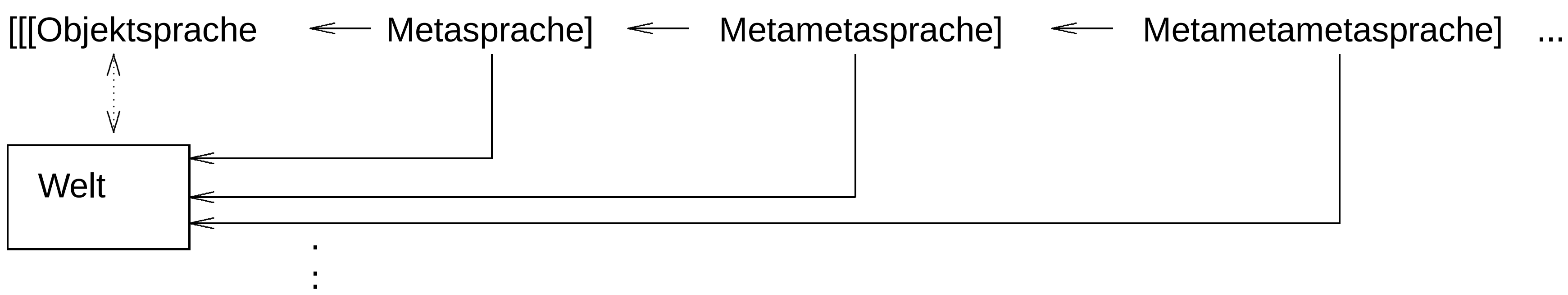
19.4.4 Beispiel einer inhaltsleeren T-Bedingung

'A ist rot' ist ein wahrer Satz dann und nur dann, wenn A rot ist.

19.4.5 Verbesserte T-Bedingung für rot

'A ist rot' ist ein wahrer Satz dann und nur dann, wenn A Licht im Frequenzbereich zwischen α und β refraktiert.

19.4.6 Hierarchie der Metasprachen



19.4.7 Unabhängigkeit von der Metasprache

Unabhängigkeit von der Metasprache bedeutet nicht, daß Computer auf uninterpretierte, rein syntaktische Deduktionssystem beschränkt bleiben, sondern das Tarskis Methode der semantischen Interpretation nicht die einzig mögliche ist. Anstatt objektsprachlichen Ausdrücken mit Hilfe einer Metasprache semantische Repräsentationen zuzuordnen, verwenden Computer die operationale Methode, in der die Bedeutung der Programmiersprache automatisch als Operationen der Maschine realisiert werden.

19.4.8 Beispiel für Unabhängigkeit von einer Metasprache

Eine adäquate metasprachliche Definition für die Regeln der Grundrechenarten zu erstellen, stellt kein Problem dar. Der Weg von dieser metasprachlichen Definition zu einer funktionierenden Rechenmaschine ist allerdings weit, und die Rechenmaschine wird schließlich rein mechanisch funktionieren – ohne Bezug zu dieser Metasprache und ohne die Metasprache verstehen zu müssen.

19.4.9 Programmierung logischer Systeme

Es gibt viele logische Calculi, die niemals als Computerprogramme umgesetzt worden sind und es auch niemals sein werden. Der Grund dafür liegt darin, daß ihre metasprachliche Beschreibung Teile enthält, die zwar von ihren Entwicklern als unmittelbar einsichtig angesehen werden (z.B. Quantifizierung über die unendliche Menge aller möglichen Welten in der Modallogik), die aber trotzdem für die Umsetzung als “mechanische” Prozedur vollkommen ungeeignet sind.

19.5 Tarskis Problem der natürlichsprachlichen Semantik

19.5.1 Logische Semantik für natürliche Sprache?

> The attempt to set up a structural definition of the term 'true sentence' – applicable to colloquial language – is confronted with insuperable difficulties.
> [Der Versuch, zu einer strukturellen Definition des Begriffs 'wahrer Satz' zu gelangen – anwendbar auf die Umgangssprache – steht vor unüberwindlichen Schwierigkeiten.]
>
> Tarski 1935, S. 164.

19.5.2 Epimenides Paradox, schwache Version

Das Paradox beruht auf Selbstreferenz. In seiner ursprünglichen ‘schwachen’ Form lautet es wie folgt:
Wenn ein Kreter sagt, Alle Kreter lügen (immer), gibt es zwei Möglichkeiten.
● Entweder der Kreter spricht die Wahrheit, in welchem Fall es falsch ist, daß *alle* Kreter lügen – da er ja selbst ein Kreter ist.
● Oder der Kreter lügt, d. h. es gibt mindestens einen anderen Kreter, der nicht lügt.
In beiden Fällen ist der Satz falsch.

19.5.3 Epimenides Paradox, starke Version (nach Leśniewski)

c = Satz in Zeile X (sogenannte ‘Abkürzung’)

Zeile X: c ist nicht ein wahrer Satz

Substitution der c-Abkürzung: Satz in Zeile X ist nicht ein wahrer Satz

Interpretation der c-Abkürzung: c ist nicht ein wahrer Satz ist nicht ein wahrer Satz.

Eliminierung der doppelten Negation: c ist ein wahrer Satz.

19.5.4 Tarskis Beweis

Um der besseren Klarheit willen werden wir das Symbol ‘c’ als typographische Abkürzung des Ausdrucks ‘der Satz, der auf Seite 418 in der vierten Zeile von oben gedruckt steht’ verwenden. Man betrachte nun den folgenden Satz:

c ist nicht ein wahrer Satz.

Bezüglich der Bedeutung des Symbols ‘c’ können wir empirisch feststellen:

(a) ‘c ist nicht ein wahrer Satz’ ist identisch mit c.

Für das Zitat des Satzes c geben wir eine Erklärung im Sinn der T-Bedingung 19.3.3:

(b) ‘c ist nicht ein wahrer Satz’ ist ein wahrer Satz dann – und nur dann – wenn c nicht ein wahrer Satz ist.

Die Prämissen (a) und (b) ergeben zusammen sofort einen Widerspruch:
c ist ein wahrer Satz dann und nur dann, wenn c nicht ein wahrer Satz ist.

Tarski 1935

19.5.5 Inkonsistente T-Bedingung beim Epimenides-Paradox

c	ist	wahr	d.u.n.d.w.	c	ist nicht	wahr
Substitution basierend auf Abkürzung		*‘wahr’ von metasprachlicher T-Bedingung*		*metasprachl. Korrelat zu objektsprachl. Wort ‘c’*		*metasprachl. Korrelat zu objektsprachl. Wort ‘wahr’*

19.5.6 Drei Möglichkeiten zur Vermeidung des Widerspruchs in der metasprachlichen T-Bedingung

1. Verbieten der Abkürzung und der daraus folgenden Substitution. Von Tarski abgelehnt, "weil es keinen rationalen Grund gibt, die Substitution im allgemeinen zu verbieten."
2. Unterscheidung der Wahrheitsprädikate 'wahr^{O}' der Objektsprache und 'wahr^{M}' der Metasprache. Damit wäre

   c ist wahr^{O} dann und nur dann, wenn c nicht wahr^{M} ist.

   nicht widersprüchlich, weil wahr^{O} $\neq$ wahr^{M}. Diese Möglichkeit zieht Tarski nicht in Betracht, vermutlich weil die Postulierung mehrerer Wahrheitsprädikate mit dem eigentlichen Anliegen der logischen Semantik, nämlich einer formalen Charakterisierung *der* Wahrheit, kollidiert.
3. Die von Tarski gewählte Option besteht darin, die Verwendung des Wahrheitsprädikats in der zu interpretierenden Objektsprache zu verbieten. Für die ursprünglichen Ziele der logischen Semantik stellt die Wahl dieser dritten Möglichkeit, also die Verbannung der Wörter *wahr* und *falsch* aus der Objektsprache, keinerlei Problem dar.

19.5.7 Gründe für die dritte Option

> “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.”
>
> Frege 1896 (1967, S. 221)

19.5.8 Gründe gegen die dritte Option

Wenn die logische Semantik auf die natürlichen Sprachen angewendet wird, folgt aus der dritte Möglichkeit ein ernstes Problem. Denn die natürlichen Sprachen *müssen* die Wörter *wahr* und *falsch* enthalten. Die logisch-semantische Interpretation einer natürlichen (Objekt-)Sprache in ihrer Gesamtheit führt also unvermeidlich zum Widerspruch.

19.5.9 Montagues Wahl: Ignorieren des Problems

> Ich weise die Behauptung zurück, daß zwischen den logischen und den natürlichen Sprachen ein wichtiger theoretischer Unterschied bestünde. . . . Wie Donald Davidson betrachte ich die Konstruktion einer Theorie der Wahrheit – oder allgemeiner, der Wahrheit unter beliebigen Interpretationen – als das elementare Ziel ernsthafter Syntax und Semantik.
>
> Montague 1970, *English as a formal language*

19.5.10 Davidsons Wahl: Aufschieben des Problems

> Tarskis Punkt ist, daß wir die natürlichen Sprachen bis zur Unkenntlichkeit reformieren müßten, bevor wir die formal-semantische Methode auf sie anwenden könnten. Sollte dies wahr sein, so ist es tödlich für mein Projekt.
>
> Davidson 1967