

Sachliche Validität

Referenz der satzweisen Prüfung in ai-lektorat

2026-09-24

Inhaltsverzeichnis

1 Ein Satz im Durchgang	1
2 Grundbegriffe	2
3 Zerlegung: der Regelkatalog	3
4 Prüfungen der Zerlegung: P1 bis P3	3
5 Das Urteil	4
6 Bedingung 5: Verträglichkeit mit den früheren Sätzen	4
7 Namensverknüpfung ohne Mustervergleich	5
8 Fachterminologie	5
9 Resolution	5
10 Meta-Validierung	6
11 Schnittstelle	6
Bibliografie	6

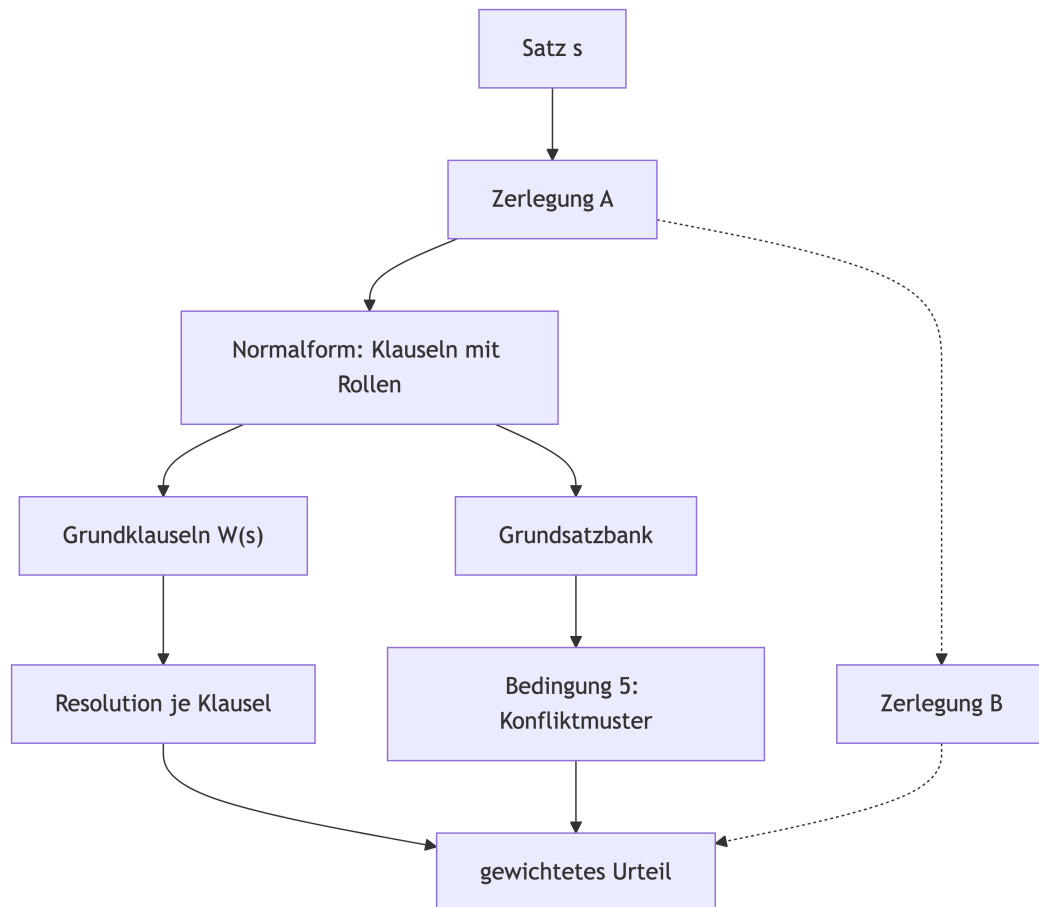
i In einem Satz

Ein Satz ist **sachlich validiert**, wenn jede seiner tragenden Aussagen aus Belegen folgt, keine widerlegt ist und der Satz nichts sagt, was einem früheren Satz des Aufsatzes oder einem Register widerspricht. Jeder andere Satz ist **nicht validiert**. Ein drittes Urteil gibt es nur für Sätze ohne Sachgehalt, etwa Fragen oder Aufforderungen.

Die Prüfung ist ein Rechtfertigungsverfahren. Sie behauptet nicht, dass ein nicht validierter Satz falsch ist. Sie hält fest, ob die vorhandenen Belege den Satz tragen. „Nicht validiert“, ist die kontradiktorische Negation von „sachlich validiert“; einen Zwischenwert wie „offen“, gibt es nicht. Die Anforderung an das Verfahren ist asymmetrisch: kein falsches „validiert“, und möglichst wenige wahre Sätze, die unbegründet bleiben.

1 Ein Satz im Durchgang

Rechtsklick auf einen Satz im Lektorat-Viewer (Port 5114) oder im Famulus Notebook (Port 5118) öffnet das Menü „Gründe der sachlichen Validität“. Das Popup zeigt das Urteil, die Zerlegung, die Beweise und die Lücken.



1. Ein Modell zerlegt den Satz nach dem Regelkatalog R1–R15 in Klauseln (Zerlegung A). Jede Klausel trägt eine Rolle.
2. Belege aus den Registern und Quellen werden zu Grundklauseln. Zusammen mit den Hintergrundregeln bilden sie die Belegbasis $W(s)$.
3. Für jede Klausel entscheidet die Resolution, ob sie aus $W(s)$ folgt, ob sie widerlegt ist oder ob sie eine Lücke bleibt.
4. Die Grundsätze des Satzes gehen in die Grundsatzbank. Konfliktmuster prüfen sie gegen alle früheren Sätze und gegen die Register.
5. Kann der Satz validiert werden, zerlegt ein zweiter, unabhängiger Aufruf ihn noch einmal (Zerlegung B). Das Urteil gilt nur, wenn A und B in den tragenden Klauseln übereinstimmen.

2 Grundbegriffe

Ein **Literal** ist ein Atom $R(c_1, \dots, c_k; \tau)$ oder seine Negation. R ist ein Prädikat aus dem Register `standards/validity-relations.yaml` (Bestand, Art, Ort, Inventar, Wert, Zustand, StehtIn, Existiert, Literatur, Erschienen, Datiert, Brief, Frei). Die c_j sind kanonische Konstanten, τ ist der Zeitindex.

Eine **Klausel** ist eine Disjunktion von Literalen. Die meisten Klauseln haben ein Literal; mehrgliedrige entstehen aus „oder,, aus Bereichen und aus ungefähren Angaben. Die **Normalform** eines Satzes ist die Konjunktion seiner Klauseln.

Die **Belegbasis** ist

$$W(s) = F(s) \cup H|_s$$

$H|_s$ sind die Hintergrundregeln, die auf die Terme von s passen (Disjunktheit von Beständen, Funktionalität von Merkmalen, Vollständigkeit erklärter Inventare). $F(s)$ sind die Befunde aus den Quellen: Register, Messläufe, Objektakten, PDAI-Datenbanken, die Bibliographie des Aufsatzes und Referenzwerke. Dass ein Eintrag fehlt, ist kein negatives Literal; es gibt keine Negation durch Fehlschlag.

Relativ zu W hat eine Klausel K einen von vier Zuständen:

Zustand	Bedingung
begründet	$W \cup \neg K \vdash \circ$
widerlegt	$W \vdash \neg l$ für jedes Literal l von K
Konflikt	beides, oder W ist in den Atomen von K unverträglich
Lücke	weder begründet noch widerlegt

3 Zerlegung: der Regelkatalog

Die Zerlegungsanweisung liegt in `skills/apparatus/validitaet/SKILL.md` (Version 1.4). Sie übersetzt Satzformen in Klauseln:

Regel	Satzform	Ergebnis
R1	Hauptprädikation	Kernklausel
R2	Kennzeichnung, Attribut, Apposition	je eine Identifizierungsklausel
R3	Koordination mit „und„	getrennte Klauseln
R4	„oder„, „zwischen a und b“	mehrgliedrige Klausel, Intervall
R5	„nicht„, „kein“, „ohne„	negiertes Literal
R6	„fünf Messreihen„	<code>Wert(gegenstand, merkmal, zahl)</code>
R7	Datum, Messlauf	Zeitindex τ
R8	„laut Manifest„, Zitierangabe	Quellenangabe
R9	„weil„, „daher“	beide Teilsätze, Zusammenhang als <code>Frei</code>
R10	Frage, Aufforderung, Wertung, innerer Verweis	Nicht-Sachgehalt
R11	definite Kennzeichnung	<code>Existiert(c)</code> als Voraussetzung
R12	„etwa„, „rund“, „mehr als„	Vergleichsprädikat mit Toleranz
R13	Datierungen	EDTF-Intervall
R14	Personen	Nennung für die Namensverknüpfung
R15	Fachtermini	Gebrauchsrolle für die Terminologie

Zwei Regeln haben sich im Betrieb als heikel erwiesen:

- **R6.** Das Merkmal muss nach seiner Bedeutung genau das zählen, was der Satz zählt. „Bestand · fünf Sammlungen„ zählt die Sammlungen, die eine Karte zeigt, und nicht die Manifeste. Passt kein Merkmal, entsteht ein `Frei`-Literal.
- **R15.** Die Gebrauchsrolle ist die Rolle, die *dieser Satz* dem Terminus durch sein Prädikat zuweist. Wahrheitsträger ist, wem ein Wahrheitswert zugesprochen wird, nie der Wahrheitswert selbst: in „ihre Wahrheit„ ist „Wahrheit“ eine Eigenschaft, die Behauptung ist der Träger.

4 Prüfungen der Zerlegung: P1 bis P3

Die Zerlegung eines Modells ist selbst ein Glied der Rechtfertigung. Sie wird geprüft, bevor ein Urteil gilt.

P1 Termtreue. Jeder Term des Satzes (Kennung, Sammlung, Zahl, Jahr, Datum, Zitat) ist Argument, Bedingung oder Zeitindex eines Literals, oder er steht im Wortlaut einer R10-Einordnung. Jede Negation ist abgebildet. Jeder Wortlaut ist Teilausdruck des Satzes.

P2 Keine Fremdterme. Jedes Argument steht im Satz, im Kontext oder im Register; jedes Prädikat steht im Register.

P3 Gegenableitung. Eine zweite, unabhängige Zerlegung B stimmt in den tragenden Klauseln (Rollen Kern und Identifizierung) überein und ordnet dieselben Terme als Nicht-Sachgehalt ein. P3 wird nur geprüft, wenn der Satz nach A validiert wäre oder keinen Sachgehalt hätte. Ein nicht validierter Satz braucht keine zweite Bestätigung seines Urteils.

5 Das Urteil

Jede Rolle hat ein Gewicht w : Kern 1,0, Identifizierung 0,8, Quellenangabe 0,6, Verknüpfung 0,4, Voraussetzung 0,2. Der **Begründungsgrad** ist

$$\delta(s) = \frac{\sum_K w(\text{rolle}_K) [K \text{ begründet}]}{\sum_K w(\text{rolle}_K)}.$$

Ein Satz ist **sachlich validiert** genau dann, wenn

1. P1, P2 und P3 erfüllt sind,
2. die ganze Normalform mit $W(s)$ verträglich ist,
3. jede Kernklausel begründet ist,
4. keine Klausel widerlegt ist und keine tragende Klausel im Konflikt steht, und $\delta(s) \geq \theta = 0,6$,
5. der Satz mit allem verträglich ist, was der Aufsatz vorher behauptet hat (Bedingung 5, nächster Abschnitt).

Ist eine dieser Bedingungen verletzt, ist der Satz **nicht validiert**, und die Begründung nennt die verletzte Bedingung. Enthält die Zerlegung keine Klausel und sind P1 bis P3 erfüllt, lautet das Urteil **keine Sachaussage**.

Die Gewichte und θ stehen in `standards/validity-rules.yaml`. Eine Nebenklausel darf unbegründet bleiben, solange δ die Schwelle erreicht; eine Kernklausel nie.

6 Bedingung 5: Verträglichkeit mit den früheren Sätzen

$$W(s) \cup \Sigma_{<s} \cup \{K\} \not\models \circ$$

$\Sigma_{<s}$ sind die Grundsätze aller früheren Sätze des Aufsatzes. Sie liegen in einer DuckDB-Grundsatzbank (`grundsätze.duckdb`): Atome mit Zahl- und Zeitintervallen, Ereignisse, Personennennungen und Terminusverwendungen. Jeder Satz hat eine Ordnung aus Absatz und Stelle; nur Paare mit einem früheren Satz zählen.

Statt einer allgemeinen Suche nach Widersprüchen prüft das Verfahren negative binäre Horn-Muster $\neg A(\bar{x}) \vee \neg B(\bar{y})$. Jedes Muster ist ein indizierter SQL-Verbund.

Muster	Unverträglich ist
K-POL	eine Aussage und ihre Negation
K-FUN	zwei verschiedene Werte eines funktionalen Merkmals mit gleichem Schlüssel
K-DIS	zwei Herkunftsbestände ohne gemeinsamen Pfad
K-ANZ	ein Teil größer als sein Ganzes, ein Anteil ungleich Zähler durch Nenner
K-EREIG	zwei disjunkte Datierungen eines einmaligen Ereignisses
K-ORD	Tod vor Geburt, eine Lebensspanne über 125 Jahre
K-LEB	ein datiertes Ereignis außerhalb der Lebenszeit eines Beteiligten

Muster	Unverträglich ist
K-ZUK	ein vergangenes Ereignis nach dem Datum des Aufsatzes
K-SCHR	ein Personennamen, dessen Nahschreibung eindeutig einen Menschen bezeichnet
K-TERM	ein Terminus in einer Rolle, die sein Wörterbucheintrag ausschließt

Zahlen werden als Intervalle nach der angegebenen Genauigkeit verglichen („16 000“ deckt 15 500 bis 16 500, „etwa., $\pm 10\%$), Daten als EDTF-Intervalle. Verschiedene Bedingungen (Sprache, Zustand, Abfrage) werden nicht verglichen: 325 702 lateinische und 137 400 deutsche Sätze von `kepler-prim` widersprechen einander nicht.

i Ein Befund blockiert erst, wenn beide Zerlegungen ihn tragen

Jede Grundaussage, auf der ein Befund ruht, muss auch in der Gegenableitung B ihres Satzes stehen. Ein Befund, den nur Zerlegung A trägt, kann ein Zerlegungsfehler sein; er erscheint als Hinweis beim Satz und hält das Urteil nicht auf. Fehlt einem früheren Satz die Gegenableitung, wird sie nachgeholt. Die Bestätigungen stehen in den Tabellen `gegenableitung` und `bestaetigt` der Grundsatzbank.

Die Bestätigung fängt Fehler auf, die nur in einer Zerlegung auftreten. Irren A und B gleich, hilft sie nicht; dann ist die Zerlegungsanweisung zu schärfen, wie bei R6 und R15 geschehen.

7 Namensverknüpfung ohne Mustervergleich

Personennamen werden nicht über reguläre Ausdrücke erkannt, sondern verknüpft:

1. gleicher Normalschlüssel mit einem Register- oder Bibliographieeintrag, auch nach Abtrennen einer Genitivendung (Keplers $\rightarrow$ Kepler);
2. sonst Nahschreibung: Jaro-Winkler $\geq 0,92$ und Damerau-Levenshtein ≤ 2 , gleicher Anfangsbuchstabe;
3. Referenzwerk: Wikidata-Suche nach Menschen (`haswbstatement:P31=Q5`), zwischengespeichert.

„Abweichend,“ heißt eine Schreibung nur, wenn ein Referenzwerk geprüft wurde und die nahe Form nachweislich einen Menschen bezeichnet, entweder durch die Bibliographie des Aufsatzes oder durch einen eindeutigen Wikidata-Treffer. Sonst bleibt es bei „nahe, ungeprüft“, und es gibt keinen Befund. Ein `bge-m3-Kosinus` ab 0,85 ergibt nur einen Hinweis.

8 Fachterminologie

Jeder Fachterminus eines Aufsatzes wird **einmal** bestimmt, im Hintergrund. Eine Warteschlange (`termin_jobs`) und ein Arbeitsfaden im Viewer erzeugen je Terminus einen Wörterbucheintrag: welche Gebrauchsrollen der Aufsatz dem Terminus geben darf und welche ausgeschlossen sind, jede mit Grund. Jede spätere Prüfung liest den Eintrag nur und vergleicht die Rolle im Satz mit einem Join (K-TERM).

- **Verbindlich** ist ein aktiver LWgrammar-Eintrag oder eine aktive Gebrauchsregel, etwa R-GG-001: nur Aussagen sind wahr oder falsch; Sachverhalte und Tatsachen sind Teile der Welt.
- Ohne verbindlichen Bestand bestimmt das Modell den Eintrag aus dem Fachgebrauch und aus den Sätzen des Aufsatzes (`skills/apparatus/terminus/SKILL.md`). Setzt der Aufsatz selbst fest, als was er einen Terminus gebraucht, ist diese Rolle zulässig.
- Der Eintrag geht als Kandidatsinn an LWgrammar zurück. Solche Kandidaten sind bei der nächsten Bestimmung Material, keine Vorgabe; sonst bestätigte sich ein früherer Eintrag selbst.

9 Resolution

Die Folgerung prüft eine Einheitspropagation mit zwei beobachteten Literalen je Klausel und Begründungszeigern [1]. Für Horn-Klauseln entscheidet sie in linearer Zeit [2]. Nur ein Rest, der nicht Horn ist, geht an DPLL [3], und nur bei unverträglicher Basis wird mit Resolution saturiert [4]. Der Beweis wird aus den Begründungszeigern rückwärts als lineare Eingaberesolution gelesen und im Popup gezeigt.

Die Laufzeit liegt fast ganz bei der Zerlegung durch das Modell. Im Audit des Famulus-Aufsatzes (557 Sätze) kostete die Resolution über alle Sätze zusammen unter einer Sekunde, die übrigen lokalen Prüfungen im Median rund 60 ms je Satz. Das Audit mit den Zeiten jedes Satzes erzeugt `scripts/validity_audit.py` (Typst, lettrea).

10 Meta-Validierung

Das Verfahren kann sich irren. Eine **Meldung** sagt: ein validierter Satz ist falsch, oder ein nicht validierter Satz ist wahr und belegt. Meldet der Eigentümer, verliert der Satz sofort das Urteil „validiert“. Die Diagnose (`viewer/validity/meta.py`) führt die Meldung auf genau eine Fehlerstelle zurück:

Fehlerstelle	Beispiel
Zerlegungslücke	eine Aussage des Satzes fehlt in der Normalform
Rollenfehler	eine Kernaussage steht als Voraussetzung
Regelfehler in H	eine Namensgleichung setzt Verschiedenes gleich
Registerfehler	ein Prädikat liest das falsche Belegfeld
Quellenfehler	ein Beleg ist selbst falsch
Quellenlücke	der Beleg für eine wahre Aussage fehlt

Jede Fehlerstelle erzeugt einen Regelkandidaten, einen Patch an `validity-rules.yaml`, `validity-relations.yaml`, `evidence-sources.yaml` oder `SKILL.md`. Der Kandidat besteht das Gate nur, wenn der gemeldete Satz das erwartete Urteil erhält, die Goldens (`tests/validity_goldens.yaml`) keine falsch-positiven Urteile bekommen, das Mutations-Gate bei null falsch-positiven Urteilen bleibt und alle gespeicherten Urteile neu berechnet werden, wobei jede Änderung aufgelistet wird. Übernommen wird nur nach Bestätigung durch den Eigentümer, als neue Version mit Historie.

11 Schnittstelle

Endpunkt	Zweck
POST <code>/api/validity/split</code>	Absätze in Sätze zerlegen
POST <code>/api/validity/check</code>	einen Satz prüfen
GET <code>/api/validity/sentence</code>	gespeichertes Urteil mit Gründen
GET <code>/api/validity/summary</code>	Urteile eines Aufsatzes
POST <code>/api/validity/run</code>	Prüflauf über einen Aufsatz, mit Ereignisstrom
GET <code>/api/validity/bank</code>	Stand der Grundsatzbank
GET <code>/api/validity/terms/status</code>	Stand der Terminuseinträge
POST <code>/api/validity/override</code>	Autor-Beleg zu einer Klausel
POST <code>/api/validity/dispute</code>	Meldung eines Fehlurteils
POST <code>/api/validity/disputes/{id}/decide</code>	Regelkandidat übernehmen oder ablehnen

Kommandozeile: `python -m ai_lektorat.viewer.validity.run --paper <name>` prüft einen Aufsatz, `--recompute` rechnet alle Urteile aus den gespeicherten Zerlegungen neu und holt fehlende Gegenableitungen nach, `--termini` arbeitet die Terminuseinträge ab.

Bibliografie

- [1] M. W. Moskewicz, C. F. Madigan, Y. Zhao, L. Zhang, und S. Malik, „Chaff: Engineering an Efficient SAT Solver“, in *Proceedings of the 38th Design Automation Conference*, 2001. doi: 10.1109/dac.2001.935565.
- [2] W. F. Dowling und J. H. Gallier, „Linear-Time Algorithms for Testing the Satisfiability of Propositional Horn Formulae“, *The Journal of Logic Programming*, Bd. 1, Nr. 3, S. 267–284, 1984, doi: 10.1016/0743-1066(84)90014-1.

- [3] M. Davis, G. Logemann, und D. Loveland, „A Machine Program for Theorem-Proving“, *Communications of the ACM*, Bd. 5, Nr. 7, S. 394–397, 1962, doi: 10.1145/368273.368557.
- [4] J. A. Robinson, „A Machine-Oriented Logic Based on the Resolution Principle“, *Journal of the ACM*, Bd. 12, Nr. 1, S. 23–41, 1965, doi: 10.1145/321250.321253.