

Formale Syntax: HPSG

o6. Lexikon und Lexikonregeln

Roland Schäfer

Institut für Germanistische Sprachwissenschaft
Friedrich-Schiller-Universität Jena

Stets aktuelle Fassungen: <https://github.com/rsling/VL-HPSG>

Basiert teilweise auf Folien von Stefan Müller: <https://hpsg.hu-berlin.de/~stefan/Lehre/S2021/hpsg.html>

Grundlage ist Stefans HPSG-Buch: <https://hpsg.hu-berlin.de/~stefan/Pub/hpsg-lehrbuch.html.de>

Stefan trägt natürlich keinerlei Verantwortung für meine Fehler und Missverständnisse!

Übersicht

- 1 Phrasenstruktur und Phrasenstrukturgrammatiken
- 2 Merkmalstrukturen und Merkmalbeschreibungen
- 3 Komplementation und Grammatikregeln
- 4 Verbsemantik und Linking (Semantik 1)
- 5 Adjunktion und Spezifikation
- 6 Lexikon und Lexikonregeln
- 7 Konstituentenreihenfolge und Verbbewegung
- 8 Nicht-lokale Abhängigkeiten und Vorfelddbesetzung
- 9 Quantorenspeicher (Semantik 2)
- 10 Unterspezifikationssemantik (Semantik 3)

<https://rolandschaefer.net/archives/2805>

<https://github.com/rsling/VL-HPSG/tree/main/output>

<https://hpsg.hu-berlin.de/~stefan/Pub/hpsg-lehrbuch.html>

Einleitung

Lexikalistische Theorien lösen so viel wie möglich im Lexikon

Lexikalistische Theorien lösen so viel wie möglich im Lexikon

- Welche Information ist bei Wörtern wirklich idiosynkratisch (= individuell)?

Lexikalistische Theorien lösen so viel wie möglich im Lexikon

- Welche Information ist bei Wörtern wirklich idiosynkratisch (= individuell)?
- Wie kann man Generalisierungen im Lexikon erfassen (Typen)?

Lexikalistische Theorien lösen so viel wie möglich im Lexikon

- Welche Information ist bei Wörtern wirklich idiosynkratisch (= individuell)?
- Wie kann man Generalisierungen im Lexikon erfassen (Typen)?
- Wie helfen Typhierarchien, die sehr komplexen Lexikoneinträge zu strukturieren.

Lexikalistische Theorien lösen so viel wie möglich im Lexikon

- Welche Information ist bei Wörtern wirklich idiosynkratisch (= individuell)?
- Wie kann man Generalisierungen im Lexikon erfassen (Typen)?
- Wie helfen Typhierarchien, die sehr komplexen Lexikoneinträge zu strukturieren.
- Wie funktionieren Lexikonregeln für das Passiv?

Lexikalistische Theorien lösen so viel wie möglich im Lexikon

- Welche Information ist bei Wörtern wirklich idiosynkratisch (= individuell)?
- Wie kann man Generalisierungen im Lexikon erfassen (Typen)?
- Wie helfen Typhierarchien, die sehr komplexen Lexikoneinträge zu strukturieren.
- Wie funktionieren Lexikonregeln für das Passiv?
- Wie modelliert man Flexion und Wortbildung in HPSG? (Kapitel 19)

Lexikalistische Theorien lösen so viel wie möglich im Lexikon

- Welche Information ist bei Wörtern wirklich idiosynkratisch (= individuell)?
- Wie kann man Generalisierungen im Lexikon erfassen (Typen)?
- Wie helfen Typhierarchien, die sehr komplexen Lexikoneinträge zu strukturieren.
- Wie funktionieren Lexikonregeln für das Passiv?
- Wie modelliert man Flexion und Wortbildung in HPSG? (Kapitel 19)

Müller (2013: Kapitel 6)

Idiosynkrasie und Typbildung

Welche Informationen sind wirklich idiosynkratisch?

Welche Informationen sind wirklich idiosynkratisch?

- *Buch*

Welche Informationen sind wirklich idiosynkratisch?

- *Buch*
 - ▶ PHON und *book-rel*

Welche Informationen sind wirklich idiosynkratisch?

- *Buch*
 - ▶ PHON und *book-rel*
 - ▶ Ansonsten *neutrales Zählsubstantiv*

Welche Informationen sind wirklich idiosynkratisch?

- *Buch*
 - ▶ PHON und *book-rel*
 - ▶ Ansonsten **neutrales Zählsubstantiv**
- *Zement*

Welche Informationen sind wirklich idiosynkratisch?

- *Buch*
 - ▶ PHON und *book-rel*
 - ▶ Ansonsten *neutrales Zählsubstantiv*
- *Zement*
 - ▶ PHON und *cement-rel*

Welche Informationen sind wirklich idiosynkratisch?

- *Buch*

- ▶ PHON und *book-rel*
- ▶ Ansonsten **neutrales Zählsubstantiv**

- *Zement*

- ▶ PHON und *cement-rel*
- ▶ Ansonsten **maskulines Stoffsubstantiv**

Welche Informationen sind wirklich idiosynkratisch?

- *Buch*

- ▶ PHON und *book-rel*
- ▶ Ansonsten **neutrales Zählsubstantiv**

- *Zement*

- ▶ PHON und *cement-rel*
- ▶ Ansonsten **maskulines Stoffsubstantiv**

- *zerlegen*

Welche Informationen sind wirklich idiosynkratisch?

- *Buch*

- ▶ PHON und *book-rel*
- ▶ Ansonsten *neutrales Zählsubstantiv*

- *Zement*

- ▶ PHON und *cement-rel*
- ▶ Ansonsten *maskulines Stoffsubstantiv*

- *zerlegen*

- ▶ PHON und *disassemble-rel*

Welche Informationen sind wirklich idiosynkratisch?

- *Buch*

- ▶ PHON und *book-rel*
- ▶ Ansonsten *neutrales Zählsubstantiv*

- *Zement*

- ▶ PHON und *cement-rel*
- ▶ Ansonsten *maskulines Stoffsubstantiv*

- *zerlegen*

- ▶ PHON und *disassemble-rel*
- ▶ Ansonsten *schwaches transitives telisches Agens/Patiens-Verb*

Welche Informationen sind wirklich idiosynkratisch?

- *Buch*

- ▶ PHON und *book-rel*
- ▶ Ansonsten **neutrales Zählsubstantiv**

- *Zement*

- ▶ PHON und *cement-rel*
- ▶ Ansonsten **maskulines Stoffsubstantiv**

- *zerlegen*

- ▶ PHON und *disassemble-rel*
- ▶ Ansonsten **schwaches transitives telisches Agens/Patiens-Verb**

- *sehen*

Welche Informationen sind wirklich idiosynkratisch?

- *Buch*

- ▶ PHON und *book-rel*
- ▶ Ansonsten **neutrales Zählsubstantiv**

- *Zement*

- ▶ PHON und *cement-rel*
- ▶ Ansonsten **maskulines Stoffsubstantiv**

- *zerlegen*

- ▶ PHON und *disassemble-rel*
- ▶ Ansonsten **schwaches transitives telisches Agens/Patiens-Verb**

- *sehen*

- ▶ PHON und *see-rel*

Welche Informationen sind wirklich idiosynkratisch?

- *Buch*

- ▶ PHON und *book-rel*
- ▶ Ansonsten **neutrales Zählsubstantiv**

- *Zement*

- ▶ PHON und *cement-rel*
- ▶ Ansonsten **maskulines Stoffsubstantiv**

- *zerlegen*

- ▶ PHON und *disassemble-rel*
- ▶ Ansonsten **schwaches transitives telisches Agens/Patiens-Verb**

- *sehen*

- ▶ PHON und *see-rel*
- ▶ Und die **Stammformen**

Welche Informationen sind wirklich idiosynkratisch?

- *Buch*

- ▶ PHON und *book-rel*
- ▶ Ansonsten **neutrales Zählsubstantiv**

- *Zement*

- ▶ PHON und *cement-rel*
- ▶ Ansonsten **maskulines Stoffsubstantiv**

- *zerlegen*

- ▶ PHON und *disassemble-rel*
- ▶ Ansonsten **schwaches transitives telisches Agens/Patiens-Verb**

- *sehen*

- ▶ PHON und *see-rel*
- ▶ Und die **Stammformen**
- ▶ Ansonsten **transitives atelisches Agens/Thema-Verb**

Was entspricht der traditionellen Wortklasse *Nomen*?

Wir schreiben jetzt reine Typangaben in AVMs ohne eckige Klammern.

Was entspricht der traditionellen Wortklasse *Nomen*?

Wir schreiben jetzt reine Typangaben in AVMs ohne eckige Klammern.

$$\begin{bmatrix} \textit{noun-sign} \\ \text{CAT|HEAD} & \textit{noun} \\ \text{CONT} & \textit{nom-obj} \end{bmatrix}$$

Nomen-Typen I

Was entspricht der traditionellen Wortklasse *Nomen*?

Wir schreiben jetzt reine Typangaben in AVMs ohne eckige Klammern.

| | | |
|--|--|--|
| $\left[\begin{array}{ll} \textit{noun-sign} & \\ \text{CAT HEAD} & \textit{noun} \\ \text{CONT} & \textit{nom-obj} \end{array} \right]$ | | |
|--|--|--|

Die Typen *noun* und *nom-obj* sind dann anderswo in der Hierarchie zu spezifizieren.

Hier nur der Illustration halber. Alle Werte haben wiederum Typen.

Nomen-Typen I

Was entspricht der traditionellen Wortklasse *Nomen*?

Wir schreiben jetzt reine Typangaben in AVMs ohne eckige Klammern.

$$\left[\begin{array}{ll} \textit{noun-sign} & \\ \text{CAT|HEAD} & \textit{noun} \\ \text{CONT} & \textit{nom-obj} \end{array} \right]$$

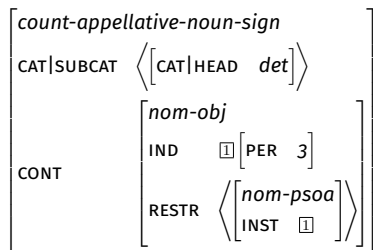
Die Typen *noun* und *nom-obj* sind dann anderswo in der Hierarchie zu spezifizieren.

Hier nur der Illustration halber. Alle Werte haben wiederum Typen.

$$\left[\begin{array}{ll} \textit{noun} & \\ \text{CAS} & \textit{case} \end{array} \right] \quad \left[\begin{array}{ll} \textit{nom-obj} & \\ \text{IND} & \textit{index} \\ \text{RESTR} & \left\langle \left[\begin{array}{ll} \textit{nom-psoa} & \\ \text{INST} & \textit{index} \end{array} \right] \right\rangle \end{array} \right]$$

Achtung! Die Aussagen auf der letzten Folie zu *nom-obj* gelten z. B. **nicht für Pronomina**.
Pronomina führen keine REL ein. Substantive sind dafür immer dritte Person.

Achtung! Die Aussagen auf der letzten Folie zu *nom-obj* gelten z. B. **nicht für Pronomina**. Pronomina führen keine REL ein. Substantive sind dafür immer dritte Person.

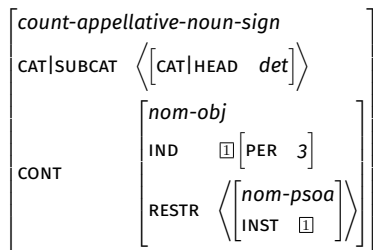


Achtung! Die Aussagen auf der letzten Folie zu *nom-obj* gelten z. B. **nicht für Pronomina**. Pronomina führen keine REL ein. Substantive sind dafür immer dritte Person.

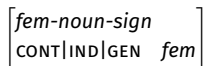


Für Feminina gilt zusätzlich:

Achtung! Die Aussagen auf der letzten Folie zu *nom-obj* gelten z. B. **nicht für Pronomina**. Pronomina führen keine REL ein. Substantive sind dafür immer dritte Person.

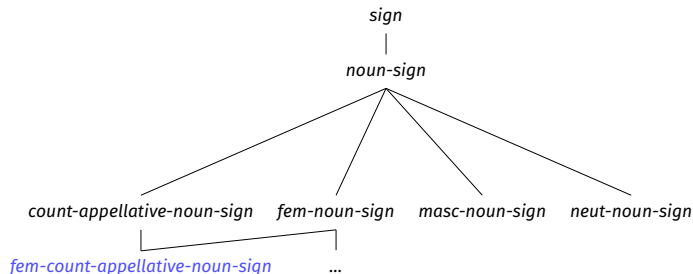


Für Feminina gilt zusätzlich:

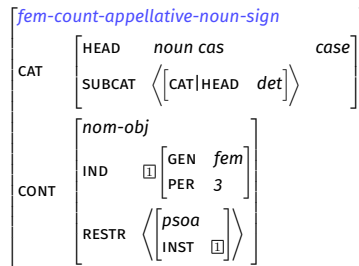
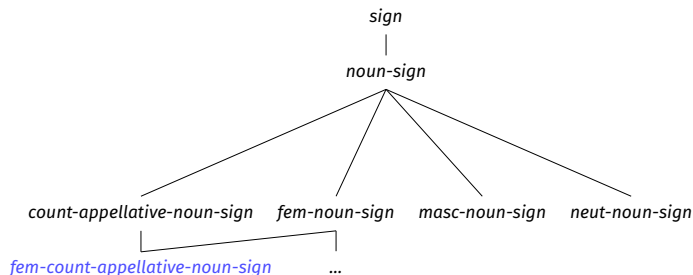


Aus Typen, die Teilinformationen kodieren, werden terminale spezifische Typen gebildet.
Hier nur beispielhafte Toy-Hierarchie. Eine größere Hierarchie weiter unten und später in *Trale*.

Aus Typen, die Teilinformationen kodieren, werden terminale spezifische Typen gebildet.
Hier nur beispielhafte Toy-Hierarchie. Eine größere Hierarchie weiter unten und später in *Trale*.



Aus Typen, die Teilinformationen kodieren, werden terminale spezifische Typen gebildet.
Hier nur beispielhafte Toy-Hierarchie. Eine größere Hierarchie weiter unten und später in *Trale*.



Verben an sich

Verben an sich

| | |
|------------|-------------|
| CAT HEAD | <i>verb</i> |
| CONT RESTR | <i>psoa</i> |

Verben an sich

| | |
|------------|-------------|
| CAT HEAD | <i>verb</i> |
| CONT RESTR | <i>psoa</i> |

Dativverben

Verben an sich

$$\begin{bmatrix} \text{CAT|HEAD} & \textit{verb} \\ \text{CONT|RESTR} & \textit{psoa} \end{bmatrix}$$

Dativverben

$$\begin{bmatrix} \text{CAT|SUBCAT} & \langle \text{NP}_{\text{NOM}}, \text{NP}_{\text{DAT}} \rangle \end{bmatrix}$$

Verben an sich

$$\begin{bmatrix} \text{CAT|HEAD} & \textit{verb} \\ \text{CONT|RESTR} & \textit{psoa} \end{bmatrix}$$

Dativverben

$$\begin{bmatrix} \text{CAT|SUBCAT} & \langle \text{NP}_{\text{NOM}}, \text{NP}_{\text{DAT}} \rangle \end{bmatrix}$$

Agentivische Experienterverben

Verben an sich

$$\begin{bmatrix} \text{CAT|HEAD} & \textit{verb} \\ \text{CONT|RESTR} & \textit{psoa} \end{bmatrix}$$

Dativverben

$$\begin{bmatrix} \text{CAT|SUBCAT} & \langle \text{NP}_{\text{NOM}}, \text{NP}_{\text{DAT}} \rangle \end{bmatrix}$$

Agentivische Experienterverben

$$\begin{bmatrix} \text{CAT|SUBCAT} & \langle [\text{CONT|IND } \boxed{1}], [\text{CONT|IND } \boxed{2}] \rangle \\ \text{CONT|RESTR} & \begin{bmatrix} \textit{agens-exp-rel} \\ \text{AGENS } \boxed{1} \\ \text{EXP } \boxed{2} \end{bmatrix} \end{bmatrix}$$

Verben an sich

$$\left[\begin{array}{l|l} \text{CAT|HEAD} & \text{verb} \\ \text{CONT|RESTR} & \text{psoa} \end{array} \right]$$

Dativverben

$$\left[\text{CAT|SUBCAT} \left\langle \text{NP}_{\text{NOM}}, \text{NP}_{\text{DAT}} \right\rangle \right]$$

Agentivische Experienterverben

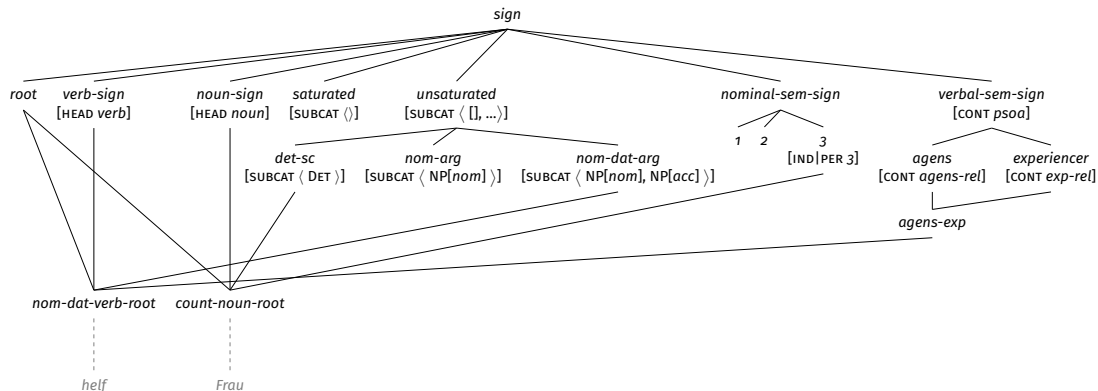
$$\left[\begin{array}{l|l} \text{CAT|SUBCAT} & \left\langle [\text{CONT|IND } \boxed{1}], [\text{CONT|IND } \boxed{2}] \right\rangle \\ \text{CONT|RESTR} & \left[\begin{array}{l} \text{agens-exp-rel} \\ \text{AGENS } \boxed{1} \\ \text{EXP } \boxed{2} \end{array} \right] \end{array} \right]$$

Im Ergebnis:

$$\left[\begin{array}{l} \text{CAT} \\ \text{SUBCAT} \left\langle \begin{array}{l} \left[\begin{array}{l|l} \text{HEAD} & \text{verb} \\ \text{CAT} & \left[\begin{array}{l|l} \text{HEAD} & \left[\begin{array}{l} \text{noun} \\ \text{CAS} & \text{nom} \end{array} \right] \\ \text{SUBCAT} & \langle \rangle \end{array} \right] \\ \text{CONT|IND} & \boxed{1} \end{array} \right] , \\ \left[\begin{array}{l|l} \text{HEAD} & \left[\begin{array}{l} \text{noun} \\ \text{CAS} & \text{dat} \end{array} \right] \\ \text{SUBCAT} & \langle \rangle \end{array} \right] \end{array} \right\rangle \\ \text{CONT|IND } \boxed{2} \end{array} \right] \\ \text{CONT|RESTR} \left\langle \left[\begin{array}{l} \text{agens-exp-rel} \\ \text{AGENS } \boxed{1} \\ \text{EXP } \boxed{2} \end{array} \right] \right\rangle \end{array} \right]$$

Möglicher größerer Ausschnitt der Typhierarchie

Möglicher größerer Ausschnitt der Typhierarchie



Platitüde aus der Morphologie- oder Syntax-Einführung: Wortarten sind immer nur ein Konstrukt. Wir teilen Wörter grob so ein, wie wir es für die Grammatik brauchen.

Platitüde aus der Morphologie- oder Syntax-Einführung: Wortarten sind immer nur ein Konstrukt. Wir teilen Wörter grob so ein, wie wir es für die Grammatik brauchen.

- Solche Äußerungen treffen auf nicht-formalisierte Grammatiken zu.

Platitüde aus der Morphologie- oder Syntax-Einführung: Wortarten sind immer nur ein Konstrukt. Wir teilen Wörter grob so ein, wie wir es für die Grammatik brauchen.

- Solche Äußerungen treffen auf nicht-formalisierte Grammatiken zu.
- In der Formalisierung verschwinden diese Einschränkungen:

Platitüde aus der Morphologie- oder Syntax-Einführung: Wortarten sind immer nur ein Konstrukt. Wir teilen Wörter grob so ein, wie wir es für die Grammatik brauchen.

- Solche Äußerungen treffen auf nicht-formalisierte Grammatiken zu.
- In der Formalisierung verschwinden diese Einschränkungen:
 - ▶ Typen erfassen Generalisierungen über Wörter und Wortformen.

Platitüde aus der Morphologie- oder Syntax-Einführung: Wortarten sind immer nur ein Konstrukt. Wir teilen Wörter grob so ein, wie wir es für die Grammatik brauchen.

- Solche Äußerungen treffen auf nicht-formalisierte Grammatiken zu.
- In der Formalisierung verschwinden diese Einschränkungen:
 - ▶ Typen erfassen Generalisierungen über Wörter und Wortformen.
 - ▶ Konkrete Wörter erben von diversen Typen und haben einen maximal spezifischen Typ.

Platitüde aus der Morphologie- oder Syntax-Einführung: Wortarten sind immer nur ein Konstrukt. Wir teilen Wörter grob so ein, wie wir es für die Grammatik brauchen.

- Solche Äußerungen treffen auf nicht-formalisierte Grammatiken zu.
- In der Formalisierung verschwinden diese Einschränkungen:
 - ▶ Typen erfassen Generalisierungen über Wörter und Wortformen.
 - ▶ Konkrete Wörter erben von diversen Typen und haben einen maximal spezifischen Typ.
 - ▶ Wörter bringen zusätzlich idiosynkratische Informationen mit.
Frau ist ein *noun-sign*, *det-sc*, *nominal-sem-sign/3* mit *Frau* als PHON-Wert.

Platitüde aus der Morphologie- oder Syntax-Einführung: Wortarten sind immer nur ein Konstrukt. Wir teilen Wörter grob so ein, wie wir es für die Grammatik brauchen.

- Solche Äußerungen treffen auf nicht-formalisierte Grammatiken zu.
- In der Formalisierung verschwinden diese Einschränkungen:
 - ▶ Typen erfassen Generalisierungen über Wörter und Wortformen.
 - ▶ Konkrete Wörter erben von diversen Typen und haben einen maximal spezifischen Typ.
 - ▶ Wörter bringen zusätzlich idiosynkratische Informationen mit.
Frau ist ein *noun-sign*, *det-sc*, *nominal-sem-sign/3* mit *Frau* als PHON-Wert.
 - ▶ Wortarten erfassen brutal vereinfacht Teilaspekte dieser Typhierarchie.

Platitüde aus der Morphologie- oder Syntax-Einführung: Wortarten sind immer nur ein Konstrukt. Wir teilen Wörter grob so ein, wie wir es für die Grammatik brauchen.

- Solche Äußerungen treffen auf nicht-formalisierte Grammatiken zu.
- In der Formalisierung verschwinden diese Einschränkungen:
 - ▶ Typen erfassen Generalisierungen über Wörter und Wortformen.
 - ▶ Konkrete Wörter erben von diversen Typen und haben einen maximal spezifischen Typ.
 - ▶ Wörter bringen zusätzlich idiosynkratische Informationen mit.
Frau ist ein *noun-sign*, *det-sc*, *nominal-sem-sign/3* mit *Frau* als PHON-Wert.
 - ▶ Wortarten erfassen brutal vereinfacht Teilaspekte dieser Typhierarchie.
- Wortarten sind Konstrukte, Typen (und Typhierarchien) modellieren Realität.

Platitüde aus der Morphologie- oder Syntax-Einführung: Wortarten sind immer nur ein Konstrukt. Wir teilen Wörter grob so ein, wie wir es für die Grammatik brauchen.

- Solche Äußerungen treffen auf nicht-formalisierte Grammatiken zu.
- In der Formalisierung verschwinden diese Einschränkungen:
 - ▶ Typen erfassen Generalisierungen über Wörter und Wortformen.
 - ▶ Konkrete Wörter erben von diversen Typen und haben einen maximal spezifischen Typ.
 - ▶ Wörter bringen zusätzlich idiosynkratische Informationen mit.
Frau ist ein *noun-sign*, *det-sc*, *nominal-sem-sign*/3 mit *Frau* als PHON-Wert.
 - ▶ Wortarten erfassen brutal vereinfacht Teilaspekte dieser Typhierarchie.
- Wortarten sind Konstrukte, Typen (und Typhierarchien) modellieren Realität.
- Wenn Sie sonst nichts aus dieser Vorlesung übrig behalten:
Daran sollten Sie sich erinnern, wenn Sie Wortarten unterrichten.

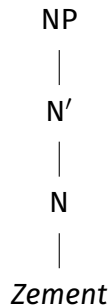
Lexikonregeln und Passiv

Nichts verbietet unäre Projektionen in HPSG. Analog zu X-Bar-Syntax:

Aus Kontexten wie: *Wir brauchen dringend Zement.*

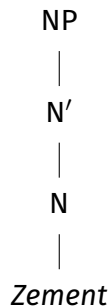
Nichts verbietet unäre Projektionen in HPSG. Analog zu X-Bar-Syntax:

Aus Kontexten wie: *Wir brauchen dringend Zement.*

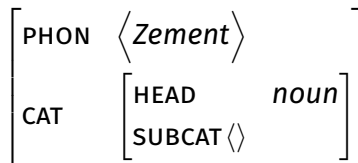


Nichts verbietet unäre Projektionen in HPSG. Analog zu X-Bar-Syntax:

Aus Kontexten wie: *Wir brauchen dringend Zement.*

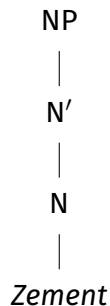


Wir brauchen solche Projektionen nicht.

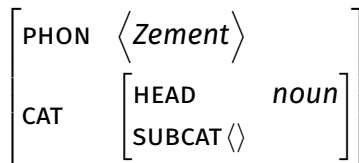


Nichts verbietet unäre Projektionen in HPSG. Analog zu X-Bar-Syntax:

Aus Kontexten wie: *Wir brauchen dringend Zement.*



Wir brauchen solche Projektionen nicht.



Das Wort kommt als NP aus dem Lexikon.

Unäre Syntaxregeln

Man kann aber **unäre Regeln** einführen und daran beliebige Funktionen aufhängen.
Hypothetisches Schema, das ein Stoffsubstantiv zu einem sortalen Nomen macht.

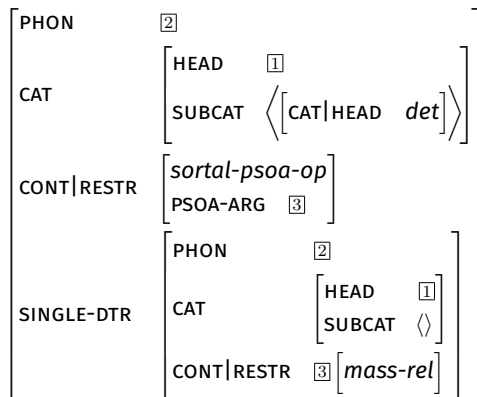
(ein bisschen) Zement → (ein) Zement

Unäre Syntaxregeln

Man kann aber **unäre Regeln** einführen und daran beliebige Funktionen aufhängen.
Hypothetisches Schema, das ein Stoffsubstantiv zu einem sortalen Nomen macht.

(ein bisschen) Zement → (ein) Zement

sortal-noun-unary-phrase ⇒

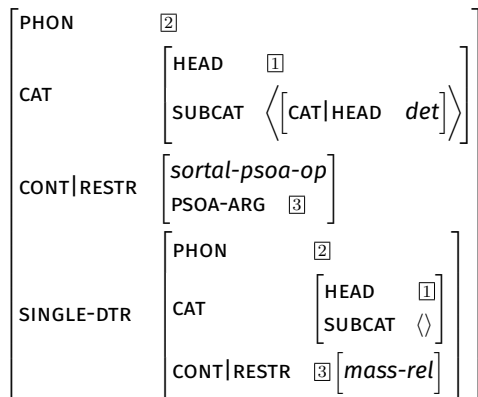


Unäre Syntaxregeln

Man kann aber **unäre Regeln** einführen und daran beliebige Funktionen aufhängen.
Hypothetisches Schema, das ein Stoffsubstantiv zu einem sortalen Nomen macht.

(ein bisschen) Zement → (ein) Zement

sortal-noun-unary-phrase ⇒



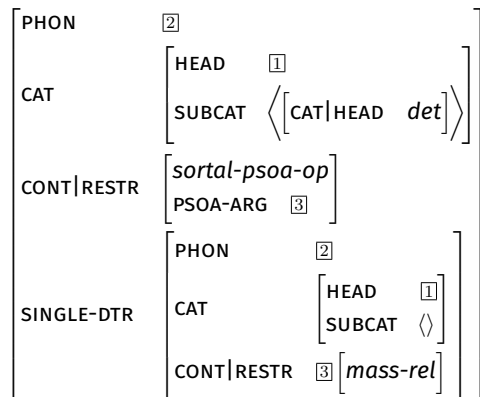
- Die einzige Tochter ist ein Stoffsubstantiv.

Unäre Syntaxregeln

Man kann aber **unäre Regeln** einführen und daran beliebige Funktionen aufhängen. Hypothetisches Schema, das ein Stoffsubstantiv zu einem sortalen Nomen macht.

(ein bisschen) Zement → (ein) Zement

sortal-noun-unary-phrase ⇒



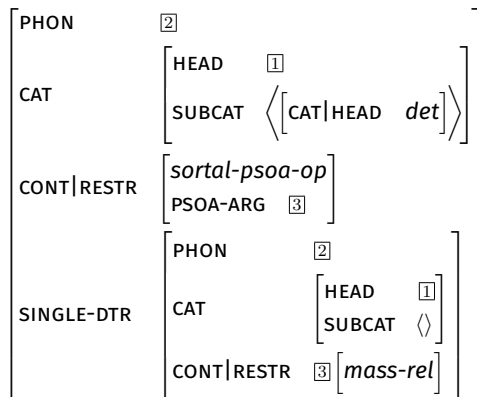
- Die einzige Tochter ist ein Stoffsubstantiv.
- Es kommt ein sortales Nomen heraus (CONT-Magie).

Unäre Syntaxregeln

Man kann aber **unäre Regeln** einführen und daran beliebige Funktionen aufhängen. Hypothetisches Schema, das ein Stoffsubstantiv zu einem sortalen Nomen macht.

(ein bisschen) Zement → (ein) Zement

sortal-noun-unary-phrase ⇒



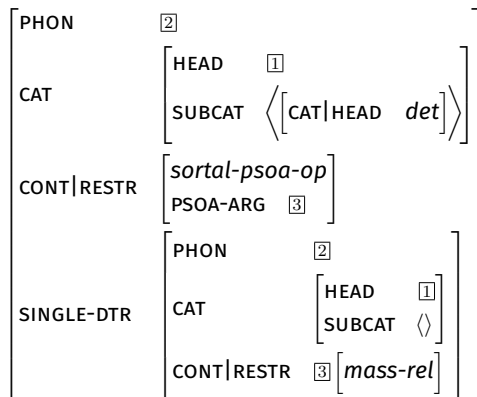
- Die einzige Tochter ist ein Stoffsubstantiv.
- Es kommt ein sortales Nomen heraus (CONT-Magie).
- Das sortale Nomen braucht einen Determinierer (im Gegensatz zum Stoffsubstantiv).

Unäre Syntaxregeln

Man kann aber **unäre Regeln** einführen und daran beliebige Funktionen aufhängen. Hypothetisches Schema, das ein Stoffsubstantiv zu einem sortalen Nomen macht.

(ein bisschen) Zement → (ein) Zement

sortal-noun-unary-phrase ⇒



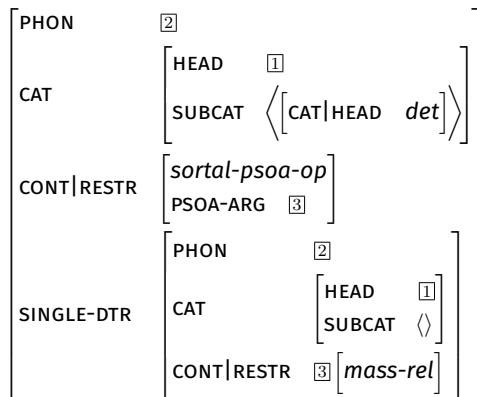
- Die einzige Tochter ist ein Stoffsubstantiv.
- Es kommt ein sortales Nomen heraus (CONT-Magie).
- Das sortale Nomen braucht einen Determinierer (im Gegensatz zum Stoffsubstantiv).
- PHON und HEAD bleiben gleich.

Unäre Syntaxregeln

Man kann aber **unäre Regeln** einführen und daran beliebige Funktionen aufhängen.
Hypothetisches Schema, das ein Stoffsubstantiv zu einem sortalen Nomen macht.

(ein bisschen) Zement → (ein) Zement

sortal-noun-unary-phrase ⇒



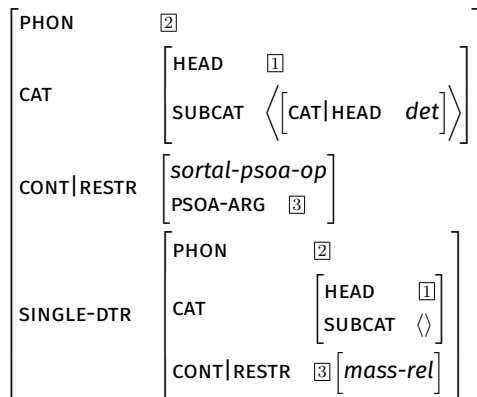
- Die einzige Tochter ist ein Stoffsubstantiv.
- Es kommt ein sortales Nomen heraus (CONT-Magie).
- Das sortale Nomen braucht einen Determinierer (im Gegensatz zum Stoffsubstantiv).
- PHON und HEAD bleiben gleich.
- Das könnten wir so machen und hätten damit eine Art **syntaktischer Konversion**.

Unäre Syntaxregeln

Man kann aber **unäre Regeln** einführen und daran beliebige Funktionen aufhängen. Hypothetisches Schema, das ein Stoffsubstantiv zu einem sortalen Nomen macht.

(ein bisschen) Zement → (ein) Zement

sortal-noun-unary-phrase ⇒



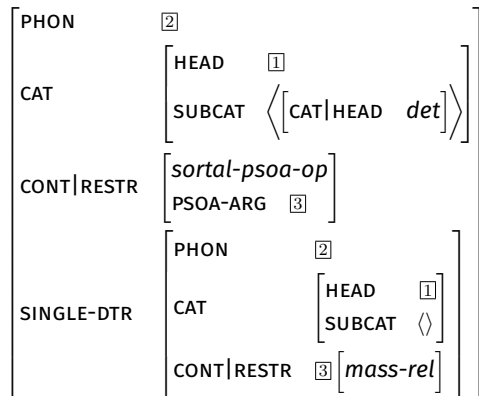
- Die einzige Tochter ist ein Stoffsubstantiv.
- Es kommt ein sortales Nomen heraus (CONT-Magie).
- Das sortale Nomen braucht einen Determinierer (im Gegensatz zum Stoffsubstantiv).
- PHON und HEAD bleiben gleich.
- Das könnten wir so machen und hätten damit eine Art **syntaktischer Konversion**.
- Probleme gäbe es, **wenn das Nomen bereits Adjunkte zu sich genommen hat**.

Unäre Syntaxregeln

Man kann aber **unäre Regeln** einführen und daran beliebige Funktionen aufhängen.
Hypothetisches Schema, das ein Stoffsubstantiv zu einem sortalen Nomen macht.

(ein bisschen) Zement → (ein) Zement

sortal-noun-unary-phrase ⇒



- Die einzige Tochter ist ein Stoffsubstantiv.
- Es kommt ein sortales Nomen heraus (CONT-Magie).
- Das sortale Nomen braucht einen Determinierer (im Gegensatz zum Stoffsubstantiv).
- PHON und HEAD bleiben gleich.
- Das könnten wir so machen und hätten damit eine Art **syntaktischer Konversion**.
- Probleme gäbe es, **wenn das Nomen bereits Adjunkte zu sich genommen hat**.
- **Man vermeidet solche Regeln möglichst in der Syntax.**

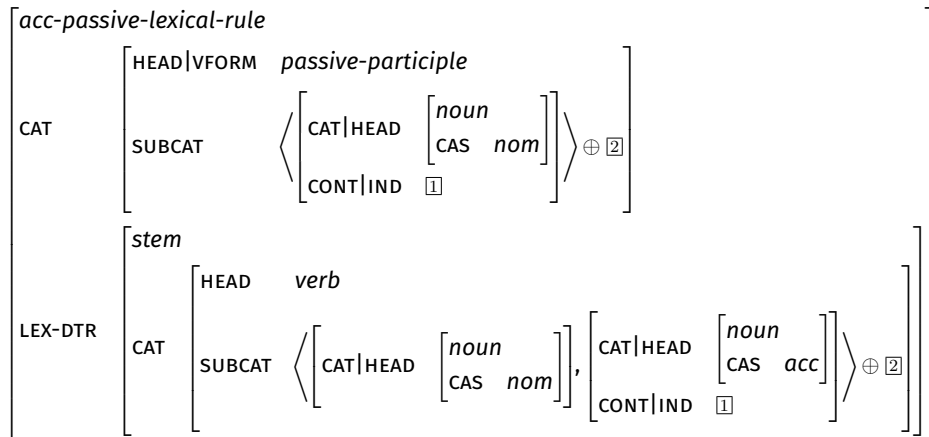
Description-Level-Lexical Rules (DLR)

Description-Level-Lexical Rules (DLR)

Lexikonregeln funktionieren ähnlich. Aber ihre Töchter sind immer Lexikoneinträge.

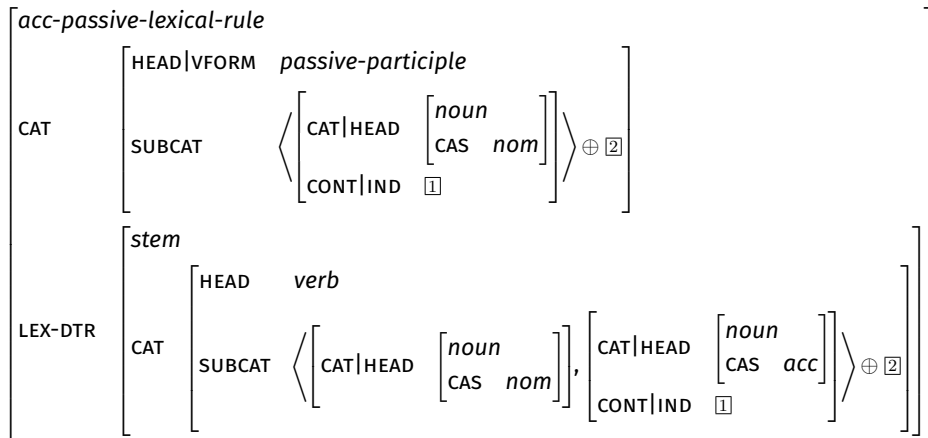
Description-Level-Lexical Rules (DLR)

Lexikonregeln funktionieren ähnlich. Aber ihre Töchter sind immer Lexikoneinträge.



Description-Level-Lexical Rules (DLR)

Lexikonregeln funktionieren ähnlich. Aber ihre Töchter sind immer Lexikoneinträge.



Deswegen erkläre ich in Schäfer (2018), dass Passiv lexikalisch ist.

Vollständige Argumentation: Ackerman & Webelhuth (1998)

Wir betrachten hier nur DLR-Lexikonregeln.

Alternativen s. Müller (2013: Kapitel 6).

Wir betrachten hier nur DLR-Lexikonregeln.

Alternativen s. Müller (2013: Kapitel 6).

- Es gibt keinen formalen Unterschied zwischen Morphologie und Syntax.

Wir betrachten hier nur DLR-Lexikonregeln.

Alternativen s. Müller (2013: Kapitel 6).

- Es gibt keinen formalen Unterschied zwischen Morphologie und Syntax.
- Lexikonregeln sind Teil des Formalismus.

Wir betrachten hier nur DLR-Lexikonregeln.

Alternativen s. Müller (2013: Kapitel 6).

- Es gibt keinen formalen Unterschied zwischen Morphologie und Syntax.
- Lexikonregeln sind Teil des Formalismus.
- Sie sind unäre Regeln, die auf Lexikoneinträge beschränkt sind.

Wir betrachten hier nur DLR-Lexikonregeln.

Alternativen s. Müller (2013: Kapitel 6).

- Es gibt keinen formalen Unterschied zwischen Morphologie und Syntax.
- Lexikonregeln sind Teil des Formalismus.
- Sie sind unäre Regeln, die auf Lexikoneinträge beschränkt sind.
- Die LEX-DTR ist der lexikalische Input.

Wir betrachten hier nur DLR-Lexikonregeln.

Alternativen s. Müller (2013: Kapitel 6).

- Es gibt keinen formalen Unterschied zwischen Morphologie und Syntax.
- Lexikonregeln sind Teil des Formalismus.
- Sie sind unäre Regeln, die auf Lexikoneinträge beschränkt sind.
- Die LEX-DTR ist der lexikalische Input.
- Alles, worüber die Regel nichts aussagt, wird übernommen.

Wir betrachten hier nur DLR-Lexikonregeln.

Alternativen s. Müller (2013: Kapitel 6).

- Es gibt keinen formalen Unterschied zwischen Morphologie und Syntax.
- Lexikonregeln sind Teil des Formalismus.
- Sie sind unäre Regeln, die auf Lexikoneinträge beschränkt sind.
- Die LEX-DTR ist der lexikalische Input.
- Alles, worüber die Regel nichts aussagt, wird übernommen.
- So funktionieren Passiv, Flexion, Wortbildung usw. in HPSG.

Wir betrachten hier nur DLR-Lexikonregeln.

Alternativen s. Müller (2013: Kapitel 6).

- Es gibt keinen formalen Unterschied zwischen Morphologie und Syntax.
- Lexikonregeln sind Teil des Formalismus.
- Sie sind unäre Regeln, die auf Lexikoneinträge beschränkt sind.
- Die LEX-DTR ist der lexikalische Input.
- Alles, worüber die Regel nichts aussagt, wird übernommen.
- So funktionieren Passiv, Flexion, Wortbildung usw. in HPSG.
- Theorien wie HPSG sind Theorien des gesamten Sprachsystems inkl. Lexikon, sie sind keine reinen Syntaxen im engen Sinn.

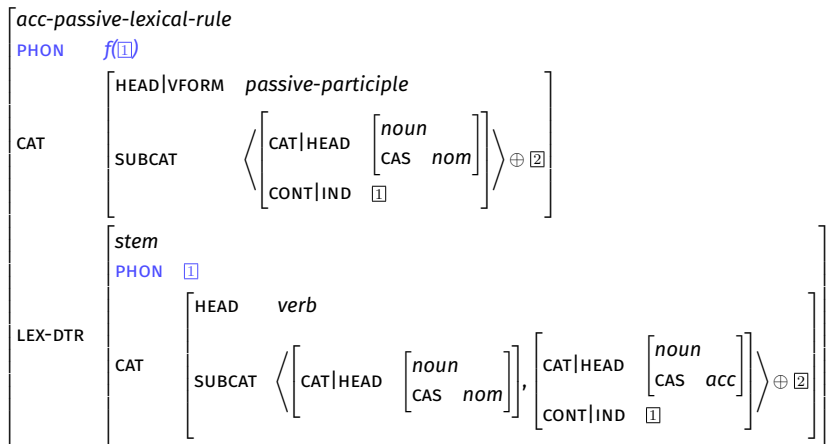
Passiv mit Morphonolomagie

Passiv mit Morphonolomagie

Um die Form kümmert sich *f*!

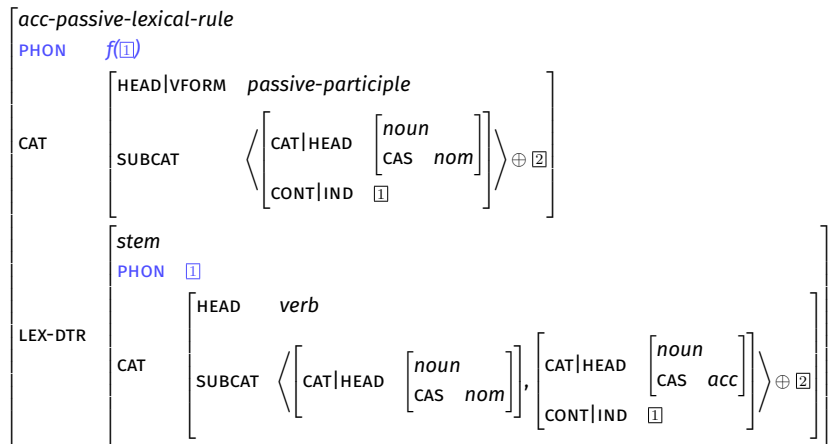
Passiv mit Morphonologie

Um die Form kümmert sich f !



Passiv mit Morphonologie

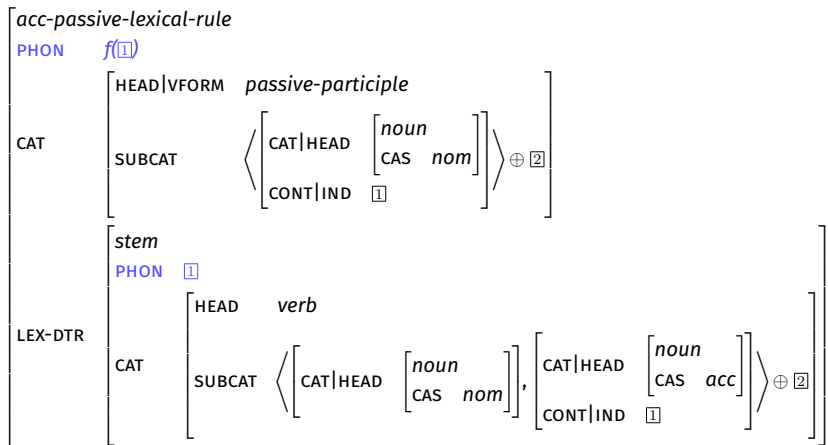
Um die Form kümmert sich f !



Die Funktion f baut die Form *gekauft* zu *kauf* usw.

Passiv mit Morphonologie

Um die Form kümmert sich *f*!



Die Funktion *f* baut die Form *gekauft* zu *kauf* usw.

Und starke Verben?

Wenn man nicht f noch mehr externes Wissen mitgeben will, muss man irgendwo die **Information über die Stammallomorphie** in *stem*-Typen repräsentieren.

Starke Verben

Wenn man nicht *f* noch mehr externes Wissen mitgeben will, muss man irgendwo die **Information über die Stammallomorphie** in *stem*-Typen repräsentieren.

Starke Verben:

| <i>verb-stem-phon</i> | |
|-----------------------|------------|
| PRES-STEM | <i>geb</i> |
| PRET-STEM | <i>gab</i> |
| PART-STEM | <i>geb</i> |

Starke Verben

Wenn man nicht *f* noch mehr externes Wissen mitgeben will, muss man irgendwo die **Information über die Stammallomorphie** in *stem*-Typen repräsentieren.

Starke Verben:

| | |
|-----------------------|----------------|
| <i>verb-stem-phon</i> | |
| PRES-STEM | ⟨ <i>geb</i> ⟩ |
| PRET-STEM | ⟨ <i>gab</i> ⟩ |
| PART-STEM | ⟨ <i>geb</i> ⟩ |

Alternativ die Information über das Ablautmuster für *f* hinterlegen:

| | | | | | |
|-----------------------------|---|-----------------------------|--|--|----------------|
| <i>stem</i> | | | | | |
| PHON | <table><tr><td colspan="2"><i>verb-ablaut-eae-phon</i></td></tr><tr><td></td><td>⟨<i>g_b</i>⟩</td></tr></table> | <i>verb-ablaut-eae-phon</i> | | | ⟨ <i>g_b</i> ⟩ |
| <i>verb-ablaut-eae-phon</i> | | | | | |
| | ⟨ <i>g_b</i> ⟩ | | | | |

Starke Verben

Wenn man nicht *f* noch mehr externes Wissen mitgeben will, muss man irgendwo die **Information über die Stammallomorphie** in *stem*-Typen repräsentieren.

Starke Verben:

| <i>verb-stem-phon</i> | |
|-----------------------|----------------|
| PRES-STEM | ⟨ <i>geb</i> ⟩ |
| PRET-STEM | ⟨ <i>gab</i> ⟩ |
| PART-STEM | ⟨ <i>geb</i> ⟩ |

Alternativ die Information über das Ablautmuster für *f* hinterlegen:

| <i>stem</i> | | | | | |
|-----------------------------|---|-----------------------------|--|--|----------------|
| PHON | <table><tr><th colspan="2"><i>verb-ablaut-eae-phon</i></th></tr><tr><td></td><td>⟨<i>g_b</i>⟩</td></tr></table> | <i>verb-ablaut-eae-phon</i> | | | ⟨ <i>g_b</i> ⟩ |
| <i>verb-ablaut-eae-phon</i> | | | | | |
| | ⟨ <i>g_b</i> ⟩ | | | | |

Nomina:

| <i>noun-stem-phon</i> | |
|-----------------------|-----------------|
| SG-STEM | ⟨ <i>Haus</i> ⟩ |
| PL-STEM | ⟨ <i>Häus</i> ⟩ |

| <i>noun-stem-phon</i> | |
|-----------------------|-------------------|
| SG-ALLOFORM | ⟨ <i>Haus</i> ⟩ |
| PL-ALLOFORM | ⟨ <i>Häuser</i> ⟩ |

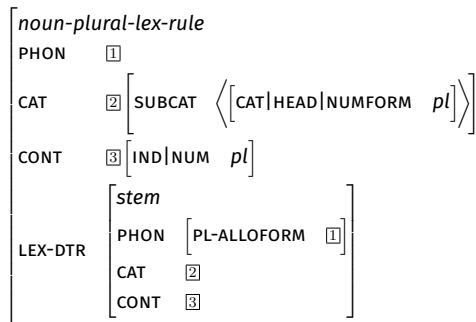
Siehe Crysmann (2021) für richtige Morphologie in HPSG.

Flexion und Wortbildung mit Lexikonregeln

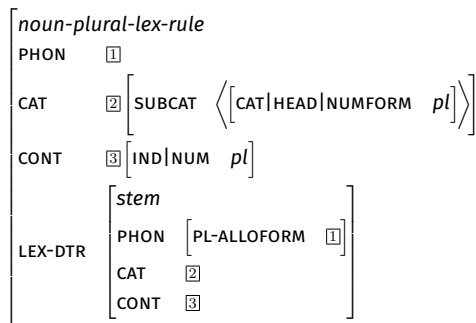
Worin besteht Pluralbildung bei Nomina?

Worin besteht Pluralbildung bei Nomina? – Formänderung und NUM:

Worin besteht Pluralbildung bei Nomina? – Formänderung und NUM:

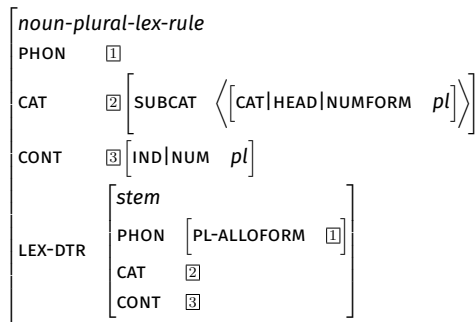


Worin besteht Pluralbildung bei Nomina? – Formänderung und NUM:



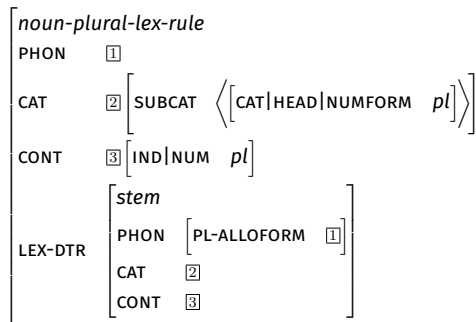
- PHON ist hier für *stem* komplex.

Worin besteht Pluralbildung bei Nomina? – Formänderung und NUM:



- PHON ist hier für *stem* komplex.
- Die eigentliche Quantifikation macht der Quantor (Artikel/Determinierer).

Worin besteht Pluralbildung bei Nomina? – Formänderung und NUM:



- PHON ist hier für *stem* komplex.
- Die eigentliche Quantifikation macht der Quantor (Artikel/Determinierer).
- Der Quantor muss aber ein pluralischer (*zwei, mehrere, ...*) sein.
Das wurde hier behelfsmäßig mit NUMFORM implementiert.

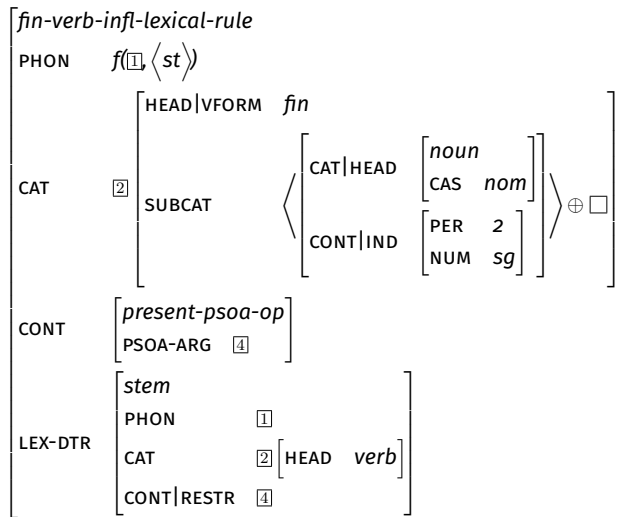
Worin besteht Verbflexion?

Worin besteht Verbflexion? – Formänderung, Tempus, Person, Numerus(, Modus)

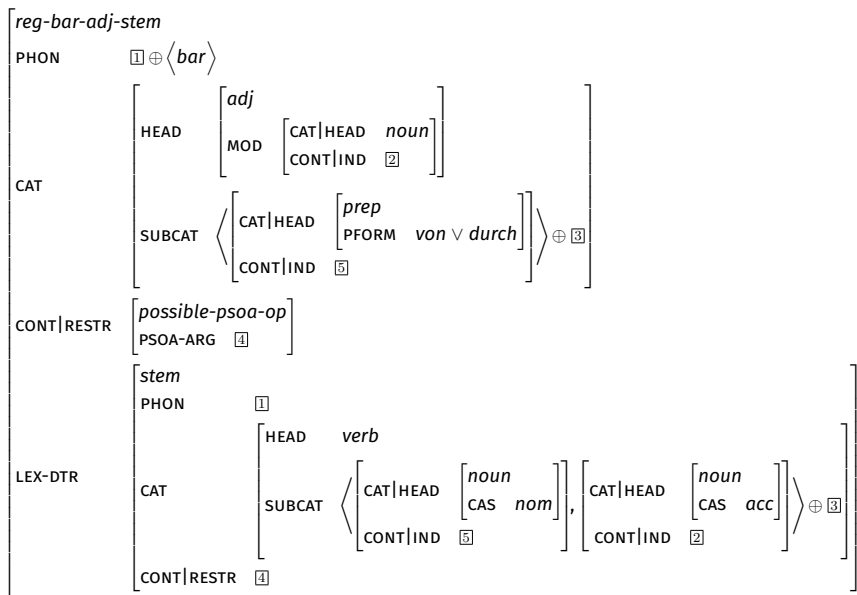
Vereinfachung für unsere Zwecke nach Müller (2013: 380).

Worin besteht Verbflexion? – Formänderung, Tempus, Person, Numerus(, Modus)

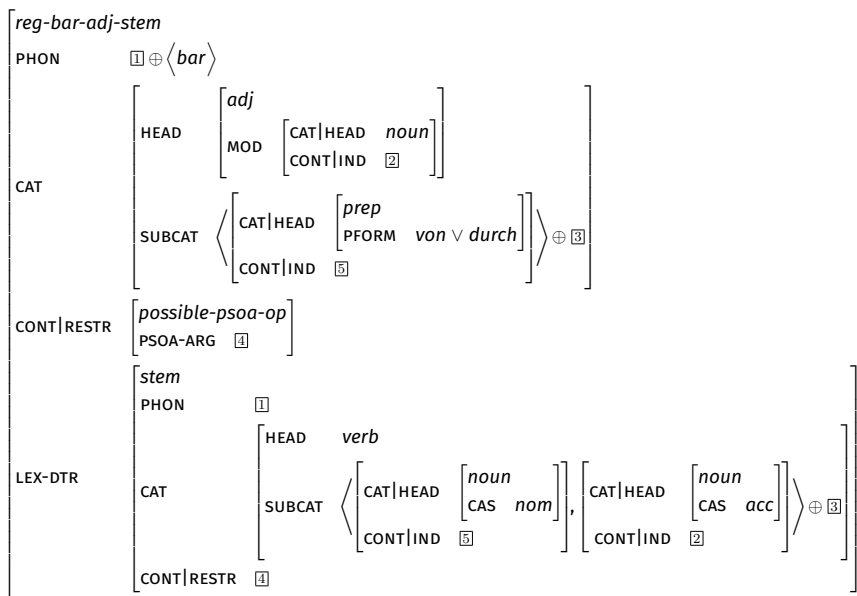
Vereinfachung für unsere Zwecke nach Müller (2013: 380).



Derivation mit *-bar*



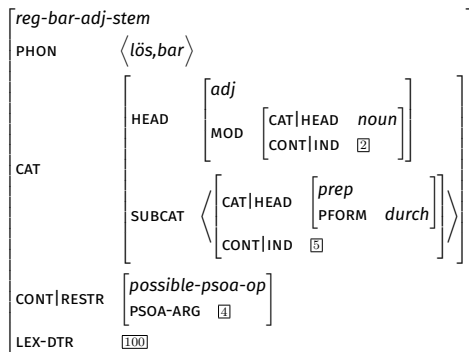
Derivation mit *-bar*



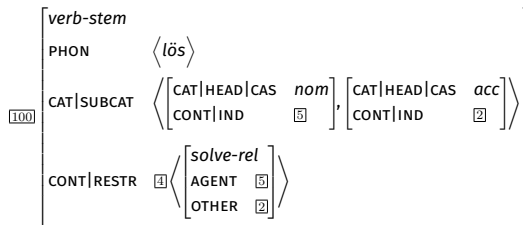
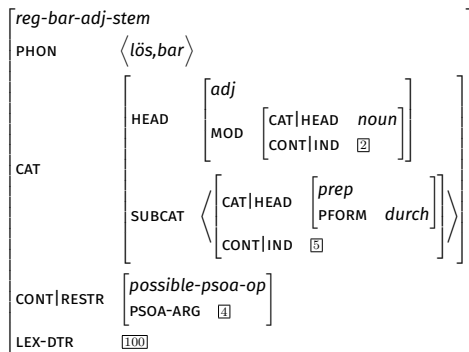
Die Version in Müller (2013: 382) ist allgemeiner, aber dadurch schwerer nachvollziehbar.

Beispielkontext: *die durch mich lösbare Gleichung*

Beispielkontext: *die durch mich lösbare Gleichung*



Beispielkontext: *die durch mich lösbare Gleichung*



Nächste Woche

Nächste Woche reden wir über Konstituentenstellung und V1-Sätze.

Nächste Woche reden wir über Konstituentenstellung und V1-Sätze.

Sie sollten dringend vorher aus dem HPSG-Buch
von Kapitel 9 die Seiten 129–148 lesen!

Nächste Woche reden wir über Konstituentenstellung und V1-Sätze.

Sie sollten dringend vorher aus dem HPSG-Buch
von Kapitel 9 die Seiten 129–148 lesen!

Das sind 20 Seiten.

Etwas mehr als sonst, aber durchaus machbar.

Nächste Woche reden wir über Konstituentenstellung und V1-Sätze.

Sie sollten dringend vorher aus dem HPSG-Buch
von Kapitel 9 die Seiten 129–148 lesen!

Das sind 20 Seiten.

Etwas mehr als sonst, aber durchaus machbar.

Achtung! In der Woche darauf sind die Seiten 163–147 dran.
Lesen Sie ggf. im Voraus!

- Ackerman, Farrell & Gert Webelhuth. 1998. *A Theory of Predicates*. (CSLI Lecture Notes 76).
- Crysmann, Berthold. 2021. Morphology. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (Hrsg.), *Head-Driven Phrase Structure Grammar: The Handbook*, 947–999. Berlin.
- Müller, Stefan. 2013. *Head-Driven Phrase Structure Grammar: Eine Einführung*. 3. Aufl. (Stauffenburg Einführungen 17). Tübingen: Stauffenburg Verlag.
- Schäfer, Roland. 2018. *Einführung in die grammatische Beschreibung des Deutschen*. 3. Aufl. (Textbooks in Language Sciences 2). Berlin.

Kontakt

Prof. Dr. Roland Schäfer
Institut für Germanistische Sprachwissenschaft
Friedrich-Schiller-Universität Jena
Fürstengraben 30
07743 Jena

<https://rolandschaefer.net>
roland.schaefer@uni-jena.de

Creative Commons BY-SA-3.0-DE

Dieses Werk ist unter einer Creative Commons Lizenz vom Typ *Namensnennung - Weitergabe unter gleichen Bedingungen 3.0 Deutschland* zugänglich. Um eine Kopie dieser Lizenz einzusehen, konsultieren Sie

<http://creativecommons.org/licenses/by-sa/3.0/de/> oder wenden Sie sich brieflich an Creative Commons, Postfach 1866, Mountain View, California, 94042, USA.