Three Improvements to the HPSG Model Theory

Adam Przepiórkowski[1,2,3]

[1]University of Warsaw (Faculty of Philosophy)
[2]Polish Academy of Sciences (Institute of Computer Science)
[3]University of Oxford (Faculty of Linguistics)

HPSG 2021

28 July 2021

Problems with HPSG model theory

Aim: propose **three orthogonal improvements to the standard HPSG model theory** (RSRL; Richter 2004):

- solve the long-standing **"Höhle's problem"** of spurious ambiguities,
- solve various problems stemming from the insistence on **exhaustive models**,
- propose a **second-order extension** to deal with the analysis of unlike category coordination in Yatabe 2004.

The first two problems are to some extent dealt with in Richter 2007: "Closer to the Truth: A New Model Theory for HPSG".

Problems with HPSG model theory

Aim: propose **three orthogonal improvements to the standard HPSG model theory** (RSRL; Richter 2004):

- solve the long-standing **"Höhle's problem"** of spurious ambiguities,
- solve various problems stemming from the insistence on **exhaustive models**,
- propose a **second-order extension** to deal with the analysis of unlike category coordination in Yatabe 2004.

The first two problems are to some extent dealt with in Richter 2007: "Closer to the Truth: A New Model Theory for HPSG".

Problems with HPSG model theory

Aim: propose **three orthogonal improvements to the standard HPSG model theory** (RSRL; Richter 2004):

- solve the long-standing **"Höhle's problem"** of spurious ambiguities,
- solve various problems stemming from the insistence on **exhaustive models**,
- propose a **second-order extension** to deal with the analysis of unlike category coordination in Yatabe 2004.

The first two problems are to some extent dealt with in Richter 2007: "Closer to the Truth: A New Model Theory for HPSG".

Problems with HPSG model theory

Aim: propose **three orthogonal improvements to the standard HPSG model theory** (RSRL; Richter 2004):

- solve the long-standing **"Höhle's problem"** of spurious ambiguities,
- solve various problems stemming from the insistence on **exhaustive models**,
- propose a **second-order extension** to deal with the analysis of unlike category coordination in Yatabe 2004.

The first two problems are to some extent dealt with in Richter 2007: "Closer to the Truth: A New Model Theory for HPSG".

Problems with HPSG model theory

Aim: propose **three orthogonal improvements to the standard HPSG model theory** (RSRL; Richter 2004):

- solve the long-standing **"Höhle's problem"** of spurious ambiguities,
- solve various problems stemming from the insistence on **exhaustive models**,
- propose a **second-order extension** to deal with the analysis of unlike category coordination in Yatabe 2004.

The first two problems are to some extent dealt with in Richter 2007: "Closer to the Truth: A New Model Theory for HPSG".

Höhle's problem

The following problem dubbed "**Höhle's problem**", e.g., in Pollard 2001, 2014: 113.

- *She says she loves you.*
- 2 × *she* (assume different *ref*s):

```
[ word
  PHON      ⟨she⟩
  SS|LOCAL  [ local
              CAT   [ category
                      HEAD  [ noun
                              CASE nom ]
                      VAL   [ valence
                              SUBJ  ⟨ ⟩
                              COMPS ⟨ ⟩ ] ]
              CONT  [ ppro
                      INDEX [ ref
                              PERS 3
                              NUMB sg
                              GEND f ] ] ] ]
```

- same *3* or not?
- same *sg* or not?
- same *f* or not?
- same *category* or not?
- if not, same *noun* or not?
- if not, same *nom* or not?
- if *category* not same, same *valence* or not?
- same PHON values?
- if not, same *she*?
- *assuming there is just one elist*, there are **168 different configurations** for 2 × *she*!
- **5208**, taking into account verbal HEAD values,
- **billions** (hint: Bell numbers), taking into account *elists*.

Höhle's problem

The following problem dubbed "**Höhle's problem**", e.g., in Pollard 2001, 2014: 113.

- *She says she loves you.*
- 2 × *she* (assume different *ref*s):

```
[ word
  PHON      ⟨she⟩
  SS|LOCAL  [ local
              CAT   [ category
                      HEAD [ noun
                             CASE nom ]
                      VAL  [ valence
                             SUBJ  ⟨ ⟩
                             COMPS ⟨ ⟩ ] ]
              CONT  [ ppro
                      INDEX [ ref
                              PERS 3
                              NUMB sg
                              GEND f ] ] ] ]
```

- same *3* or not?
- same *sg* or not?
- same *f* or not?
- same *category* or not?
- if not, same *noun* or not?
- if not, same *nom* or not?
- if *category* not same, same *valence* or not?
- same PHON values?
- if not, same *she*?
- *assuming there is just one elist*, there are **168 different configurations** for 2 × *she*!
- **5208**, taking into account verbal HEAD values,
- **billions** (hint: Bell numbers), taking into account *elists*.

Höhle's problem

The following problem dubbed "**Höhle's problem**", e.g., in Pollard 2001, 2014: 113.

- *She says she loves you.*
- 2 × *she* (assume different *ref*s):

```
[ word
  PHON ⟨she⟩
  SS|LOCAL [ local
             CAT [ category
                   HEAD [ noun
                          CASE nom ]
                   VAL [ valence
                         SUBJ ⟨ ⟩
                         COMPS ⟨ ⟩ ] ]
             CONT [ ppro
                    INDEX [ ref
                            PERS 3
                            NUMB sg
                            GEND f ] ] ] ]
```

- same *3* or not?
- same *sg* or not?
- same *f* or not?
- same *category* or not?
- if not, same *noun* or not?
- if not, same *nom* or not?
- if *category* not same, same *valence* or not?
- same PHON values?
- if not, same *she*?
- *assuming there is just one elist*, there are **168 different configurations** for 2 × *she*!
- **5208**, taking into account verbal HEAD values,
- **billions** (hint: Bell numbers), taking into account *elists*.

Höhle's problem

The following problem dubbed "**Höhle's problem**", e.g., in Pollard 2001, 2014: 113.

- *She says she loves you.*
- 2 × *she* (assume different *ref*s):

```
[ word
  PHON ⟨she⟩
  SS|LOCAL [ local
             CAT  [ category
                    HEAD [ noun
                           CASE nom ]
                    VAL  [ valence
                           SUBJ ⟨⟩
                           COMPS ⟨⟩ ] ]
             CONT [ ppro
                    INDEX [ ref
                            PERS 3
                            NUMB sg
                            GEND f ] ] ] ]
```

- same *3* or not?
- same *sg* or not?
- same *f* or not?
- same *category* or not?
- if not, same *noun* or not?
- if not, same *nom* or not?
- if *category* not same, same *valence* or not?
- same PHON values?
- if not, same *she*?
- *assuming there is just one elist*, there are **168 different configurations** for 2 × *she*!
- **5208**, taking into account verbal HEAD values,
- **billions** (hint: Bell numbers), taking into account *elists*.

Höhle's problem

The following problem dubbed "**Höhle's problem**", e.g., in Pollard 2001, 2014: 113.

- *She says she loves you.*
- 2 × *she* (assume different *ref*s):

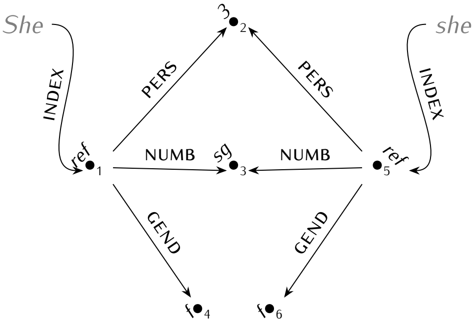
$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{she} \rangle \\ \text{SS}|\text{LOCAL} & \begin{bmatrix} \textit{local} & \\ \text{CAT} & \begin{bmatrix} \textit{category} & \\ \text{HEAD} & \begin{bmatrix} \textit{noun} \\ \text{CASE} & \textit{nom} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \textit{valence} & \\ \text{SUBJ} & \langle\rangle \\ \text{COMPS} & \langle\rangle \end{bmatrix} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \textit{ppro} & \\ \text{INDEX} & \begin{bmatrix} \textit{ref} & \\ \text{PERS} & \textit{3} \\ \text{NUMB} & \textit{sg} \\ \text{GEND} & \textit{f} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

- same *3* or not?
- same *sg* or not?
- same *f* or not?
- same *category* or not?
- if not, same *noun* or not?
- if not, same *nom* or not?
- if *category* not same, same *valence* or not?
- same PHON values?
- if not, same *she*?
- *assuming there is just one elist*, there are **168 different configurations** for 2 × *she*!
- **5208**, taking into account verbal HEAD values,
- **billions** (hint: Bell numbers), taking into account *elists*.

Höhle's problem

The following problem dubbed "**Höhle's problem**", e.g., in Pollard 2001, 2014: 113.

- *She says she loves you.*
- 2 × *she* (assume different *ref*s):



- same *3* or not?
- same *sg* or not?
- same *f* or not?
- same *category* or not?
- if not, same *noun* or not?
- if not, same *nom* or not?
- if *category* not same, same *valence* or not?
- same PHON values?
- if not, same *she*?
- *assuming there is just one elist*, there are **168 different configurations** for 2 × *she*!
- **5208**, taking into account verbal HEAD values,
- **billions** (hint: Bell numbers), taking into account *elists*.

Höhle's problem

The following problem dubbed "**Höhle's problem**", e.g., in Pollard 2001, 2014: 113.

- *She says she loves you.*
- 2 × *she* (assume different *refs*):

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{she} \rangle \\ \text{SS}|\text{LOCAL} & \begin{bmatrix} \textit{local} & \\ \text{CAT} & \begin{bmatrix} \textit{category} & \\ \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{CASE} & \textit{nom} \end{bmatrix} \\ \text{VAL} & \begin{bmatrix} \textit{valence} & \\ \text{SUBJ} & \langle \rangle \\ \text{COMPS} & \langle \rangle \end{bmatrix} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \textit{ppro} & \\ \text{INDEX} & \begin{bmatrix} \textit{ref} & \\ \text{PERS} & \mathit{3} \\ \text{NUMB} & \textit{sg} \\ \text{GEND} & \textit{f} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

- same *3* or not?
- same *sg* or not?
- same *f* or not?
- same *category* or not?
- if not, same *noun* or not?
- if not, same *nom* or not?
- if *category* not same, same *valence* or not?
- same PHON values?
- if not, same *she*?
- *assuming there is just one elist*, there are **168 different configurations** for 2 × *she*!
- **5208**, taking into account verbal HEAD values,
- **billions** (hint: Bell numbers), taking into account *elists*.

Höhle's problem

The following problem dubbed "**Höhle's problem**", e.g., in Pollard 2001, 2014: 113.

- *She says she loves you.*
- 2 × *she* (assume different *refs*):

word
- PHON ⟨*she*⟩
- SS|LOCAL *local*
  - CAT *category*
    - HEAD *noun*
      - CASE *nom*
    - VAL *valence*
      - SUBJ ⟨⟩
      - COMPS ⟨⟩
  - CONT *ppro*
    - INDEX *ref*
      - PERS *3*
      - NUMB *sg*
      - GEND *f*

- same *3* or not?
- same *sg* or not?
- same *f* or not?
- same *category* or not?
- if not, same *noun* or not?
- if not, same *nom* or not?
- if *category* not same, same *valence* or not?
- same PHON values?
- if not, same *she*?
- *assuming there is just one elist*, there are **168 different configurations** for 2 × *she*!
- **5208**, taking into account verbal HEAD values,
- **billions** (hint: Bell numbers), taking into account *elists*.

Höhle's problem

The following problem dubbed "**Höhle's problem**", e.g., in Pollard 2001, 2014: 113.

- *She says she loves you.*
- 2 × *she* (assume different *ref*s):

$$\begin{bmatrix} \textit{word} \\ \text{PHON} \ \langle \textit{she} \rangle \\ \text{SS}|\text{LOCAL} \begin{bmatrix} \textit{local} \\ \text{CAT} \begin{bmatrix} \textit{category} \\ \text{HEAD} \begin{bmatrix} \textit{noun} \\ \text{CASE} \ \textit{nom} \end{bmatrix} \\ \text{VAL} \begin{bmatrix} \textit{valence} \\ \text{SUBJ} \ \langle\,\rangle \\ \text{COMPS} \ \langle\,\rangle \end{bmatrix} \end{bmatrix} \\ \text{CONT} \begin{bmatrix} \textit{ppro} \\ \text{INDEX} \begin{bmatrix} \textit{ref} \\ \text{PERS} \ \mathit{3} \\ \text{NUMB} \ \textit{sg} \\ \text{GEND} \ \textit{f} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

- same *3* or not?
- same *sg* or not?
- same *f* or not?
- same *category* or not?
- if not, same *noun* or not?
- if not, same *nom* or not?
- if *category* not same, same *valence* or not?
- same PHON values?
- if not, same *she*?
- *assuming there is just one elist*, there are **168 different configurations** for 2 × *she*!
- **5208**, taking into account verbal HEAD values,
- **billions** (hint: Bell numbers), taking into account *elists*.

Höhle's problem

The following problem dubbed "**Höhle's problem**", e.g., in Pollard 2001, 2014: 113.

- *She says she loves you.*
- 2 × *she* (assume different *ref*s):

```
[ word
  PHON ⟨she⟩
  SS|LOCAL [ local
             CAT  [ category
                    HEAD [ noun
                           CASE nom ]
                    VAL  [ valence
                           SUBJ ⟨ ⟩
                           COMPS ⟨ ⟩ ] ]
             CONT [ ppro
                    INDEX [ ref
                            PERS 3
                            NUMB sg
                            GEND f ] ] ] ]
```

- same *3* or not?
- same *sg* or not?
- same *f* or not?
- same *category* or not?
- if not, same *noun* or not?
- if not, same *nom* or not?
- if *category* not same, same *valence* or not?
- same PHON values?
- if not, same *she*?
- *assuming there is just one elist*, there are **168 different configurations** for 2 × *she*!
- **5208**, taking into account verbal HEAD values,
- **billions** (hint: Bell numbers), taking into account *elists*.

Höhle's problem

The following problem dubbed "**Höhle's problem**", e.g., in Pollard 2001, 2014: 113.

- *She says she loves you.*
- 2 × *she* (assume different *ref*s):

[*word*
  PHON ⟨*she*⟩
  SS|LOCAL [*local*
    CAT [*category*
      HEAD [*noun*
        CASE *nom*]
      VAL [*valence*
        SUBJ ⟨⟩
        COMPS ⟨⟩]]
    CONT [*ppro*
      INDEX [*ref*
        PERS *3*
        NUMB *sg*
        GEND *f*]]]]

- same *3* or not?
- same *sg* or not?
- same *f* or not?
- same *category* or not?
- if not, same *noun* or not?
- if not, same *nom* or not?
- if *category* not same, same *valence* or not?
- same PHON values?
- if not, same *she*?
- *assuming there is just one elist*, there are **168 different configurations** for 2 × *she*!
- **5208**, taking into account verbal HEAD values,
- **billions** (hint: Bell numbers), taking into account *elists*.

Höhle's problem

The following problem dubbed "**Höhle's problem**", e.g., in Pollard 2001, 2014: 113.

- *She says she loves you.*
- 2 × *she* (assume different *ref*s):

$$\begin{bmatrix} \textit{word} \\ \text{PHON} \ \langle \textit{she} \rangle \\ \text{SS|LOCAL} \begin{bmatrix} \textit{local} \\ \text{CAT} \begin{bmatrix} \textit{category} \\ \text{HEAD} \begin{bmatrix} \textit{noun} \\ \text{CASE} \ \textit{nom} \end{bmatrix} \\ \text{VAL} \begin{bmatrix} \textit{valence} \\ \text{SUBJ} \ \langle\rangle \\ \text{COMPS} \ \langle\rangle \end{bmatrix} \end{bmatrix} \\ \text{CONT} \begin{bmatrix} \textit{ppro} \\ \text{INDEX} \begin{bmatrix} \textit{ref} \\ \text{PERS} \ \textit{3} \\ \text{NUMB} \ \textit{sg} \\ \text{GEND} \ \textit{f} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

- same *3* or not?
- same *sg* or not?
- same *f* or not?
- same *category* or not?
- if not, same *noun* or not?
- if not, same *nom* or not?
- if *category* not same, same *valence* or not?
- same PHON values?
- if not, same *she*?
- *assuming there is just one elist*, there are **168 different configurations** for 2 × *she*!
- **5208**, taking into account verbal HEAD values,
- **billions** (hint: Bell numbers), taking into account *elists*.

Höhle's problem

The following problem dubbed "**Höhle's problem**", e.g., in Pollard 2001, 2014: 113.

- *She says she loves you.*
- 2 × *she* (assume different *refs*):

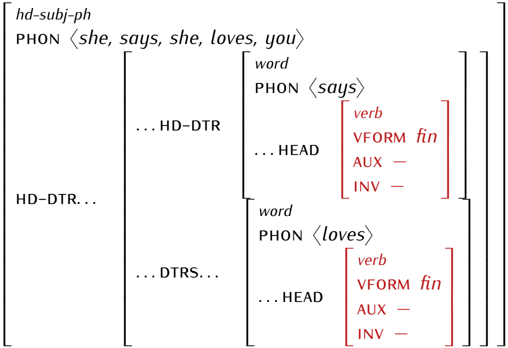
```
[ word
  PHON ⟨she⟩
  SS|LOCAL [ local
             CAT [ category
                   HEAD [ noun
                          CASE nom ]
                   VAL [ valence
                         SUBJ ⟨ ⟩
                         COMPS ⟨ ⟩ ] ]
             CONT [ ppro
                    INDEX [ ref
                            PERS 3
                            NUMB sg
                            GEND f ] ] ] ]
```

- same *3* or not?
- same *sg* or not?
- same *f* or not?
- same *category* or not?
- if not, same *noun* or not?
- if not, same *nom* or not?
- if *category* not same, same *valence* or not?
- same PHON values?
- if not, same *she*?
- *assuming there is just one elist*, there are **168 different configurations** for 2 × *she*!
- 5208, taking into account verbal HEAD values,
- **billions** (hint: Bell numbers), taking into account *elists*.

Höhle's problem

The following problem dubbed "**Höhle's problem**", e.g., in Pollard 2001, 2014: 113.

- *She says she loves you.*



- same *3* or not?
- same *sg* or not?
- same *f* or not?
- same *category* or not?
- if not, same *noun* or not?
- if not, same *nom* or not?
- if *category* not same, same *valence* or not?
- same PHON values?
- if not, same *she*?
- *assuming there is just one elist*, there are **168 different configurations** for 2 × *she*!
- **5208**, taking into account verbal HEAD values,
- **billions** (hint: Bell numbers), taking into account *elists*.

Höhle's problem

The following problem dubbed "**Höhle's problem**", e.g., in Pollard 2001, 2014: 113.

- *She says she loves you.*

hd-subj-ph
PHON ⟨*she, says, she, loves, you*⟩
HD-DTR…
- …HD-DTR
  - *word*
  - PHON ⟨*says*⟩
  - …HEAD [*verb*, VFORM *fin*, AUX −, INV −]
- …DTRS…
  - *word*
  - PHON ⟨*loves*⟩
  - …HEAD [*verb*, VFORM *fin*, AUX −, INV −]

- same *3* or not?
- same *sg* or not?
- same *f* or not?
- same *category* or not?
- if not, same *noun* or not?
- if not, same *nom* or not?
- if *category* not same, same *valence* or not?
- same PHON values?
- if not, same *she*?
- *assuming there is just one elist*, there are **168 different configurations** for 2 × *she*!
- **5208**, taking into account verbal HEAD values,
- **billions** (hint: Bell numbers), taking into account *elists*.

Partial solutions

There are **partial solutions** in the literature.

Unique Empty List Condition (Richter 2007: 102):

- $\forall \boxed{1} \forall \boxed{2} ((\boxed{1} \sim elist \wedge \boxed{2} \sim elist) \rightarrow \boxed{1} \approx \boxed{2})$

General Identity Principle (GIP; Sailer 2003: 116):

- $\forall \boxed{1} \forall \boxed{2} (\texttt{are-copies}(\boxed{1}, \boxed{2}) \rightarrow \boxed{1} \approx \boxed{2})$
- $\texttt{are-copies}(\boxed{1}, \boxed{2})$ informally (cf. Sailer 2003: 116, (130)):
  - $\boxed{1}$ and $\boxed{2}$ are of the same species, and
  - values of corresponding attributes of $\boxed{1}$ and $\boxed{2}$ stand in the `are-copies` relation.
- In Sailer 2003, the scope of GIP is constrained to configurations encoding Ty2 semantic formulae.

Partial solutions

There are **partial solutions** in the literature.

Unique Empty List Condition (Richter 2007: 102):

- $\forall \boxed{1} \forall \boxed{2} ((\boxed{1} \sim \mathit{elist} \wedge \boxed{2} \sim \mathit{elist}) \rightarrow \boxed{1} \approx \boxed{2})$

General Identity Principle (GIP; Sailer 2003: 116):

- $\forall \boxed{1} \forall \boxed{2} (\texttt{are-copies}(\boxed{1}, \boxed{2}) \rightarrow \boxed{1} \approx \boxed{2})$
- $\texttt{are-copies}(\boxed{1}, \boxed{2})$ informally (cf. Sailer 2003: 116, (130)):
  - $\boxed{1}$ and $\boxed{2}$ are of the same species, and
  - values of corresponding attributes of $\boxed{1}$ and $\boxed{2}$ stand in the `are-copies` relation.
- In Sailer 2003, the scope of GIP is constrained to configurations encoding Ty2 semantic formulae.

Partial solutions

There are **partial solutions** in the literature.

Unique Empty List Condition (Richter 2007: 102):

- $\forall \boxed{1} \forall \boxed{2} ((\boxed{1} \sim \textit{elist} \wedge \boxed{2} \sim \textit{elist}) \rightarrow \boxed{1} \approx \boxed{2})$

General Identity Principle (GIP; Sailer 2003: 116):

- $\forall \boxed{1} \forall \boxed{2} (\texttt{are-copies}(\boxed{1}, \boxed{2}) \rightarrow \boxed{1} \approx \boxed{2})$
- $\texttt{are-copies}(\boxed{1}, \boxed{2})$ informally (cf. Sailer 2003: 116, (130)):
  - $\boxed{1}$ and $\boxed{2}$ are of the same species, and
  - values of corresponding attributes of $\boxed{1}$ and $\boxed{2}$ stand in the `are-copies` relation.
- In Sailer 2003, the scope of GIP is constrained to configurations encoding Ty2 semantic formulae.

Partial solutions

There are **partial solutions** in the literature.

Unique Empty List Condition (Richter 2007: 102):

- $\forall \boxed{1} \forall \boxed{2} ((\boxed{1} \sim \textit{elist} \land \boxed{2} \sim \textit{elist}) \rightarrow \boxed{1} \approx \boxed{2})$

General Identity Principle (GIP; Sailer 2003: 116):

- $\forall \boxed{1} \forall \boxed{2} (\texttt{are-copies}(\boxed{1}, \boxed{2}) \rightarrow \boxed{1} \approx \boxed{2})$
- `are-copies`($\boxed{1}$, $\boxed{2}$) informally (cf. Sailer 2003: 116, (130)):
  - $\boxed{1}$ and $\boxed{2}$ are of the same species, and
  - values of corresponding attributes of $\boxed{1}$ and $\boxed{2}$ stand in the `are-copies` relation.
- In Sailer 2003, the scope of GIP is constrained to configurations encoding Ty2 semantic formulae.

Partial solutions

There are **partial solutions** in the literature.

Unique Empty List Condition (Richter 2007: 102):

- $\forall \boxed{1} \forall \boxed{2} \left(\left(\boxed{1} \sim \mathit{elist} \wedge \boxed{2} \sim \mathit{elist} \right) \rightarrow \boxed{1} \approx \boxed{2} \right)$

General Identity Principle (GIP; Sailer 2003: 116):

- $\forall \boxed{1} \forall \boxed{2} \left(\texttt{are-copies}(\boxed{1}, \boxed{2}) \rightarrow \boxed{1} \approx \boxed{2} \right)$
- `are-copies`($\boxed{1}$, $\boxed{2}$) informally (cf. Sailer 2003: 116, (130)):
  - $\boxed{1}$ and $\boxed{2}$ are of the same species, and
  - values of corresponding attributes of $\boxed{1}$ and $\boxed{2}$ stand in the `are-copies` relation.
- In Sailer 2003, the scope of GIP is constrained to configurations encoding Ty2 semantic formulae.

Fuller solution

Problem with making GIP universal:

- the **standard HPSG binding theory** relies on same-looking INDEX values not being necessarily same,
- **otherwise**, reliance on the non-identity of indiscernibles **very rare** in HPSG, but not unheard of:
  - Höhle 1999: §2.4 assumes that some "identities are not token but type identities",
  - Meurers 1998: 326, fn. 42 assumes "that the HEAD values of different head projections are never (accidentally) token identical, which could be explicitly enforced by a constraint on unembedded signs."

Solution:

- make **GIP universal by default**,
- but allow grammars to specify that **objects of certain sorts** (e.g., *ref*) **escape GIP**.

Fuller solution

Problem with making GIP universal:

- the **standard HPSG binding theory** relies on same-looking INDEX values not being necessarily same,
- **otherwise**, reliance on the non-identity of indiscernibles **very rare** in HPSG, but not unheard of:
  - Höhle 1999: §2.4 assumes that some "identities are not token but type identities",
  - Meurers 1998: 326, fn. 42 assumes "that the HEAD values of different head projections are never (accidentally) token identical, which could be explicitly enforced by a constraint on unembedded signs."

Solution:

- make **GIP universal by default**,
- but allow grammars to specify that **objects of certain sorts** (e.g., *ref*) **escape GIP**.

Fuller solution

Problem with making GIP universal:

- the **standard HPSG binding theory** relies on same-looking INDEX values not being necessarily same,
- **otherwise**, reliance on the non-identity of indiscernibles **very rare** in HPSG, but not unheard of:
  - Höhle 1999: §2.4 assumes that some "identities are not token but type identities",
  - Meurers 1998: 326, fn. 42 assumes "that the HEAD values of different head projections are never (accidentally) token identical, which could be explicitly enforced by a constraint on unembedded signs."

Solution:

- make **GIP universal by default**,
- but allow grammars to specify that **objects of certain sorts** (e.g., *ref*) **escape GIP**.

Fuller solution

Problem with making GIP universal:

- the **standard HPSG binding theory** relies on same-looking INDEX values not being necessarily same,
- **otherwise**, reliance on the non-identity of indiscernibles **very rare** in HPSG, but not unheard of:
  - Höhle 1999: §2.4 assumes that some "identities are not token but type identities",
  - Meurers 1998: 326, fn. 42 assumes "that the HEAD values of different head projections are never (accidentally) token identical, which could be explicitly enforced by a constraint on unembedded signs."

Solution:

- make **GIP universal by default**,
- but allow grammars to specify that **objects of certain sorts** (e.g., *ref*) **escape GIP**.

Fuller solution

Problem with making GIP universal:

- the **standard HPSG binding theory** relies on same-looking INDEX values not being necessarily same,
- **otherwise**, reliance on the non-identity of indiscernibles **very rare** in HPSG, but not unheard of:
  - Höhle 1999: §2.4 assumes that some "identities are not token but type identities",
  - Meurers 1998: 326, fn. 42 assumes "that the HEAD values of different head projections are never (accidentally) token identical, which could be explicitly enforced by a constraint on unembedded signs."

Solution:

- make **GIP universal by default**,
- but allow grammars to specify that **objects of certain sorts** (e.g., *ref*) escape GIP.

Fuller solution

Problem with making GIP universal:

- the **standard HPSG binding theory** relies on same-looking INDEX values not being necessarily same,
- **otherwise**, reliance on the non-identity of indiscernibles **very rare** in HPSG, but not unheard of:
  - Höhle 1999: §2.4 assumes that some "identities are not token but type identities",
  - Meurers 1998: 326, fn. 42 assumes "that the HEAD values of different head projections are never (accidentally) token identical, which could be explicitly enforced by a constraint on unembedded signs."

Solution:

- make **GIP universal by default**,
- but allow grammars to specify that **objects of certain sorts** (e.g., *ref*) escape GIP.

Fuller solution

Problem with making GIP universal:

- the **standard HPSG binding theory** relies on same-looking INDEX values not being necessarily same,
- **otherwise**, reliance on the non-identity of indiscernibles **very rare** in HPSG, but not unheard of:
  - Höhle 1999: §2.4 assumes that some "identities are not token but type identities",
  - Meurers 1998: 326, fn. 42 assumes "that the HEAD values of different head projections are never (accidentally) token identical, which could be explicitly enforced by a constraint on unembedded signs."

Solution:

- make **GIP universal by default**,
- but allow grammars to specify that **objects of certain sorts** (e.g., *ref*) **escape GIP**.

Technicalities

Technicalities:

- If objects of sort σ are meant to be intensional, then σ is specified for **the attribute** INT **(intensional)** with values of sort σ.
- Universal **Intensionality Principle**:
  $\forall \boxed{1} \forall \boxed{2} \left(\boxed{1} \left[\text{INT } \boxed{2} \right] \rightarrow \boxed{1} \approx \boxed{2} \right)$

For **example**, INDEX values of $2 \times$ *she*: $\boxed{1} \begin{bmatrix} \textit{ref} \\ \text{PERS } \textit{3} \\ \text{NUM } \textit{sg} \\ \text{GEND } \textit{f} \\ \text{INT } \boxed{1} \end{bmatrix}$ and $\boxed{2} \begin{bmatrix} \textit{ref} \\ \text{PERS } \textit{3} \\ \text{NUM } \textit{sg} \\ \text{GEND } \textit{f} \\ \text{INT } \boxed{2} \end{bmatrix}$.

- **Does GIP enforce** $\boxed{1} \approx \boxed{2}$**?**
  (recall GIP: $\forall \boxed{1} \forall \boxed{2} (\texttt{are-copies}(\boxed{1}, \boxed{2}) \rightarrow \boxed{1} \approx \boxed{2}))$
- That is, **are** $\boxed{1}$ **and** $\boxed{2}$ **copies?**
- `are-copies`($\boxed{1}$, $\boxed{2}$) **iff** 1) same species (**yes**), and 2) values of PERS are copies (**yes**), and 3) NUM (**yes**), and 4) GEND (**yes**), and 5) values of INT.
- So, the definition of `are-copies` **does not determine** whether `are-copies`($\boxed{1}$, $\boxed{2}$) and, hence, whether $\boxed{1} \approx \boxed{2}$.

Technicalities

Technicalities:

- If objects of sort σ are meant to be intensional, then σ is specified for **the attribute INT (intensional)** with values of sort σ.
- Universal **Intensionality Principle**:
  $\forall \boxed{1} \forall \boxed{2} \ (\boxed{1}[\text{INT } \boxed{2}] \rightarrow \boxed{1} \approx \boxed{2})$

For **example**, INDEX values of 2 × *she*: $\boxed{1}\begin{bmatrix} \textit{ref} \\ \text{PERS } 3 \\ \text{NUM } sg \\ \text{GEND } f \\ \text{INT } \boxed{1} \end{bmatrix}$ and $\boxed{2}\begin{bmatrix} \textit{ref} \\ \text{PERS } 3 \\ \text{NUM } sg \\ \text{GEND } f \\ \text{INT } \boxed{2} \end{bmatrix}$.

- **Does GIP enforce** $\boxed{1} \approx \boxed{2}$**?**
  (recall GIP: $\forall \boxed{1} \forall \boxed{2} \ (\texttt{are-copies}(\boxed{1}, \boxed{2}) \rightarrow \boxed{1} \approx \boxed{2})$)
- That is, **are** $\boxed{1}$ **and** $\boxed{2}$ **copies?**
- $\texttt{are-copies}(\boxed{1}, \boxed{2})$ **iff** 1) same species (**yes**), and 2) values of PERS are copies (**yes**), and 3) NUM (**yes**), and 4) GEND (**yes**), and 5) values of INT.
- So, the definition of are-copies **does not determine** whether $\texttt{are-copies}(\boxed{1}, \boxed{2})$ and, hence, whether $\boxed{1} \approx \boxed{2}$.

Technicalities

Technicalities:

- If objects of sort σ are meant to be intensional, then σ is specified for **the attribute INT (intensional)** with values of sort σ.
- Universal **Intensionality Principle**:
  $\forall\boxed{1}\,\forall\boxed{2}\ (\boxed{1}[\text{INT}\ \boxed{2}] \rightarrow \boxed{1} \approx \boxed{2})$

For **example**, INDEX values of 2 × *she*: $\boxed{1}\begin{bmatrix} \textit{ref} \\ \text{PERS}\ \ \textit{3} \\ \text{NUM}\ \ \textit{sg} \\ \text{GEND}\ \ \textit{f} \\ \text{INT}\ \ \boxed{1} \end{bmatrix}$ and $\boxed{2}\begin{bmatrix} \textit{ref} \\ \text{PERS}\ \ \textit{3} \\ \text{NUM}\ \ \textit{sg} \\ \text{GEND}\ \ \textit{f} \\ \text{INT}\ \ \boxed{2} \end{bmatrix}$.

- **Does GIP enforce** $\boxed{1} \approx \boxed{2}$?
  (recall GIP: $\forall\boxed{1}\forall\boxed{2}\,(\texttt{are-copies}(\boxed{1},\boxed{2}) \rightarrow \boxed{1} \approx \boxed{2})$)
- That is, **are** $\boxed{1}$ **and** $\boxed{2}$ **copies?**
- are-copies($\boxed{1}$,$\boxed{2}$) **iff** 1) same species (**yes**), and 2) values of PERS are copies (**yes**), and 3) NUM (**yes**), and 4) GEND (**yes**), and 5) values of INT.
- So, the definition of are-copies **does not determine** whether are-copies($\boxed{1}$,$\boxed{2}$) and, hence, whether $\boxed{1} \approx \boxed{2}$.

Technicalities

Technicalities:

- If objects of sort σ are meant to be intensional, then σ is specified for **the attribute** INT **(intensional)** with values of sort σ.
- Universal **Intensionality Principle**:
  $\forall\boxed{1}\,\forall\boxed{2}\ (\boxed{1}[\text{INT}\ \boxed{2}] \rightarrow \boxed{1} \approx \boxed{2})$

For **example**, INDEX values of 2 × *she*: $\boxed{1}\begin{bmatrix} \textit{ref} \\ \text{PERS}\ \ 3 \\ \text{NUM}\ \ \textit{sg} \\ \text{GEND}\ \ \textit{f} \\ \text{INT}\ \ \boxed{1} \end{bmatrix}$ and $\boxed{2}\begin{bmatrix} \textit{ref} \\ \text{PERS}\ \ 3 \\ \text{NUM}\ \ \textit{sg} \\ \text{GEND}\ \ \textit{f} \\ \text{INT}\ \ \boxed{2} \end{bmatrix}$.

- **Does GIP enforce** $\boxed{1} \approx \boxed{2}$**?**
  (recall GIP: $\forall\boxed{1}\forall\boxed{2}\,(\texttt{are-copies}(\boxed{1},\boxed{2}) \rightarrow \boxed{1} \approx \boxed{2})$)
- That is, **are** $\boxed{1}$ **and** $\boxed{2}$ **copies?**
- $\texttt{are-copies}(\boxed{1},\boxed{2})$ **iff** 1) same species (**yes**), and 2) values of PERS are copies (**yes**), and 3) NUM (**yes**), and 4) GEND (**yes**), and 5) values of INT.
- So, the definition of `are-copies` **does not determine** whether $\texttt{are-copies}(\boxed{1},\boxed{2})$ and, hence, whether $\boxed{1} \approx \boxed{2}$.

Technicalities

Technicalities:

- If objects of sort σ are meant to be intensional, then σ is specified for **the attribute** INT **(intensional)** with values of sort σ.
- Universal **Intensionality Principle**:
  $\forall \boxed{1}\, \forall \boxed{2}\; (\boxed{1}[\text{INT } \boxed{2}] \rightarrow \boxed{1} \approx \boxed{2})$

For **example**, INDEX values of 2 × *she*: $\boxed{1}\begin{bmatrix} \textit{ref} \\ \text{PERS } \textit{3} \\ \text{NUM } \textit{sg} \\ \text{GEND } \textit{f} \\ \text{INT } \boxed{1} \end{bmatrix}$ and $\boxed{2}\begin{bmatrix} \textit{ref} \\ \text{PERS } \textit{3} \\ \text{NUM } \textit{sg} \\ \text{GEND } \textit{f} \\ \text{INT } \boxed{2} \end{bmatrix}$.

- **Does GIP enforce $\boxed{1} \approx \boxed{2}$?**
  (recall GIP: $\forall \boxed{1} \forall \boxed{2}\, (\texttt{are-copies}(\boxed{1}, \boxed{2}) \rightarrow \boxed{1} \approx \boxed{2}))$
- That is, **are $\boxed{1}$ and $\boxed{2}$ copies?**
- $\texttt{are-copies}(\boxed{1}, \boxed{2})$ **iff** 1) same species (**yes**), and 2) values of PERS are copies (**yes**), and 3) NUM (**yes**), and 4) GEND (**yes**), and 5) values of INT.
- So, the definition of are-copies **does not determine** whether $\texttt{are-copies}(\boxed{1}, \boxed{2})$ and, hence, whether $\boxed{1} \approx \boxed{2}$.

Technicalities

Technicalities:

- If objects of sort σ are meant to be intensional, then σ is specified for **the attribute INT (intensional)** with values of sort σ.
- Universal **Intensionality Principle**:
  $\forall \boxed{1} \forall \boxed{2} \; (\boxed{1}[\text{INT } \boxed{2}] \rightarrow \boxed{1} \approx \boxed{2})$

For **example**, INDEX values of 2 × *she*: $\boxed{1}\begin{bmatrix} \textit{ref} \\ \text{PERS } \textit{3} \\ \text{NUM } \textit{sg} \\ \text{GEND } \textit{f} \\ \text{INT } \boxed{1} \end{bmatrix}$ and $\boxed{2}\begin{bmatrix} \textit{ref} \\ \text{PERS } \textit{3} \\ \text{NUM } \textit{sg} \\ \text{GEND } \textit{f} \\ \text{INT } \boxed{2} \end{bmatrix}$.

- **Does GIP enforce $\boxed{1} \approx \boxed{2}$?**
  (recall GIP: $\forall \boxed{1} \forall \boxed{2} \, (\texttt{are-copies}(\boxed{1}, \boxed{2}) \rightarrow \boxed{1} \approx \boxed{2})$)
- That is, **are $\boxed{1}$ and $\boxed{2}$ copies?**
- $\texttt{are-copies}(\boxed{1}, \boxed{2})$ **iff** 1) same species (**yes**), and 2) values of PERS are copies (**yes**), and 3) NUM (**yes**), and 4) GEND (**yes**), and 5) values of INT.
- So, the definition of $\texttt{are-copies}$ **does not determine** whether $\texttt{are-copies}(\boxed{1}, \boxed{2})$ and, hence, whether $\boxed{1} \approx \boxed{2}$.

Technicalities

Technicalities:

- If objects of sort σ are meant to be intensional, then σ is specified for **the attribute INT (intensional)** with values of sort σ.
- Universal **Intensionality Principle**:
  $\forall \boxed{1} \forall \boxed{2} \ (\boxed{1}[\text{INT } \boxed{2}] \rightarrow \boxed{1} \approx \boxed{2})$

For **example**, INDEX values of 2 × *she*: $\boxed{1}\begin{bmatrix} \textit{ref} \\ \text{PERS } \textit{3} \\ \text{NUM } \textit{sg} \\ \text{GEND } \textit{f} \\ \text{INT } \boxed{1} \end{bmatrix}$ and $\boxed{2}\begin{bmatrix} \textit{ref} \\ \text{PERS } \textit{3} \\ \text{NUM } \textit{sg} \\ \text{GEND } \textit{f} \\ \text{INT } \boxed{2} \end{bmatrix}$.

- **Does GIP enforce** $\boxed{1} \approx \boxed{2}$?
  (recall GIP: $\forall \boxed{1} \forall \boxed{2} \ (\texttt{are-copies}(\boxed{1}, \boxed{2}) \rightarrow \boxed{1} \approx \boxed{2})$)
- That is, **are** $\boxed{1}$ **and** $\boxed{2}$ **copies?**
- `are-copies`($\boxed{1}$, $\boxed{2}$) **iff** 1) same species (**yes**), and 2) values of PERS are copies (**yes**), and 3) NUM (**yes**), and 4) GEND (**yes**), and 5) values of INT.
- So, the definition of `are-copies` **does not determine** whether `are-copies`($\boxed{1}$, $\boxed{2}$) and, hence, whether $\boxed{1} \approx \boxed{2}$.

Technicalities

Technicalities:

- If objects of sort σ are meant to be intensional, then σ is specified for **the attribute INT (intensional)** with values of sort σ.
- Universal **Intensionality Principle**:
  $\forall \boxed{1} \forall \boxed{2} \left(\boxed{1}\left[\text{INT } \boxed{2}\right] \rightarrow \boxed{1} \approx \boxed{2}\right)$

For **example**, INDEX values of 2 × *she*: $\boxed{1}\begin{bmatrix} \textit{ref} \\ \text{PERS } \mathit{3} \\ \text{NUM } \mathit{sg} \\ \text{GEND } \mathit{f} \\ \text{INT } \boxed{1} \end{bmatrix}$ and $\boxed{2}\begin{bmatrix} \textit{ref} \\ \text{PERS } \mathit{3} \\ \text{NUM } \mathit{sg} \\ \text{GEND } \mathit{f} \\ \text{INT } \boxed{2} \end{bmatrix}$.

- **Does GIP enforce** $\boxed{1} \approx \boxed{2}$**?**
  (recall GIP: $\forall \boxed{1} \forall \boxed{2} \left(\texttt{are-copies}(\boxed{1}, \boxed{2}) \rightarrow \boxed{1} \approx \boxed{2}\right)$)
- That is, **are** $\boxed{1}$ **and** $\boxed{2}$ **copies?**
- `are-copies`$(\boxed{1}, \boxed{2})$ **iff** 1) same species (**yes**), and 2) values of PERS are copies (**yes**), and 3) NUM (**yes**), and 4) GEND (**yes**), and 5) values of INT.
- So, the definition of `are-copies` **does not determine** whether `are-copies`$(\boxed{1}, \boxed{2})$ and, hence, whether $\boxed{1} \approx \boxed{2}$.

Technicalities

Technicalities:

- If objects of sort σ are meant to be intensional, then σ is specified for **the attribute INT (intensional)** with values of sort σ.
- Universal **Intensionality Principle**:
  $\forall\boxed{1}\,\forall\boxed{2}\ (\boxed{1}[\textsc{int}\ \boxed{2}] \rightarrow \boxed{1} \approx \boxed{2})$

For **example**, INDEX values of 2 × *she*: $\boxed{1}\begin{bmatrix} \textit{ref} \\ \textsc{pers}\ \ 3 \\ \textsc{num}\ \ sg \\ \textsc{gend}\ \ f \\ \textsc{int}\ \ \boxed{1} \end{bmatrix}$ and $\boxed{2}\begin{bmatrix} \textit{ref} \\ \textsc{pers}\ \ 3 \\ \textsc{num}\ \ sg \\ \textsc{gend}\ \ f \\ \textsc{int}\ \ \boxed{2} \end{bmatrix}$.

- **Does GIP enforce $\boxed{1} \approx \boxed{2}$?**
  (recall GIP: $\forall\boxed{1}\forall\boxed{2}\,(\texttt{are-copies}(\boxed{1},\boxed{2}) \rightarrow \boxed{1} \approx \boxed{2})$)
- That is, **are $\boxed{1}$ and $\boxed{2}$ copies?**
- $\texttt{are-copies}(\boxed{1},\boxed{2})$ **iff** 1) same species (**yes**), and 2) values of PERS are copies (**yes**), and 3) NUM (**yes**), and 4) GEND (**yes**), and 5) values of INT.
- So, the definition of are-copies **does not determine** whether $\texttt{are-copies}(\boxed{1},\boxed{2})$ and, hence, whether $\boxed{1} \approx \boxed{2}$.

Technicalities

Technicalities:

- If objects of sort σ are meant to be intensional, then σ is specified for **the attribute INT (intensional)** with values of sort σ.
- Universal **Intensionality Principle**:
  $\forall\boxed{1}\,\forall\boxed{2}\ (\boxed{1}[\text{INT}\ \boxed{2}] \rightarrow \boxed{1} \approx \boxed{2})$

For **example**, INDEX values of 2 × *she*: $\boxed{1}\begin{bmatrix} \textit{ref} \\ \text{PERS}\ \ \mathit{3} \\ \text{NUM}\ \ \mathit{sg} \\ \text{GEND}\ \ \mathit{f} \\ \text{INT}\ \ \boxed{1} \end{bmatrix}$ and $\boxed{2}\begin{bmatrix} \textit{ref} \\ \text{PERS}\ \ \mathit{3} \\ \text{NUM}\ \ \mathit{sg} \\ \text{GEND}\ \ \mathit{f} \\ \text{INT}\ \ \boxed{2} \end{bmatrix}$.

- **Does GIP enforce** $\boxed{1} \approx \boxed{2}$**?**
  (recall GIP: $\forall\boxed{1}\forall\boxed{2}\,(\texttt{are-copies}(\boxed{1},\boxed{2}) \rightarrow \boxed{1} \approx \boxed{2})$)
- That is, **are** $\boxed{1}$ **and** $\boxed{2}$ **copies?**
- $\texttt{are-copies}(\boxed{1},\boxed{2})$ **iff** 1) same species (**yes**), and 2) values of PERS are copies (**yes**), and 3) NUM (**yes**), and 4) GEND (**yes**), and 5) values of INT.
- So, the definition of `are-copies` **does not determine** whether $\texttt{are-copies}(\boxed{1},\boxed{2})$ and, hence, whether $\boxed{1} \approx \boxed{2}$.

Technicalities

Technicalities:

- If objects of sort σ are meant to be intensional, then σ is specified for **the attribute** INT **(intensional)** with values of sort σ.
- Universal **Intensionality Principle**:
  $\forall \boxed{1} \forall \boxed{2} \; (\boxed{1}[\text{INT } \boxed{2}] \rightarrow \boxed{1} \approx \boxed{2})$

For **example**, INDEX values of 2 × *she*: $\boxed{1}\begin{bmatrix} \textit{ref} \\ \text{PERS } \mathit{3} \\ \text{NUM } \mathit{sg} \\ \text{GEND } \mathit{f} \\ \text{INT } \boxed{1} \end{bmatrix}$ and $\boxed{2}\begin{bmatrix} \textit{ref} \\ \text{PERS } \mathit{3} \\ \text{NUM } \mathit{sg} \\ \text{GEND } \mathit{f} \\ \text{INT } \boxed{2} \end{bmatrix}$.

- **Does GIP enforce** $\boxed{1} \approx \boxed{2}$**?**
  (recall GIP: $\forall \boxed{1} \forall \boxed{2} \, (\texttt{are-copies}(\boxed{1}, \boxed{2}) \rightarrow \boxed{1} \approx \boxed{2}))$
- That is, **are** $\boxed{1}$ **and** $\boxed{2}$ **copies?**
- $\texttt{are-copies}(\boxed{1}, \boxed{2})$ **iff** 1) same species (**yes**), and 2) values of PERS are copies (**yes**), and 3) NUM (**yes**), and 4) GEND (**yes**), and 5) values of INT.
- So, the definition of are-copies **does not determine** whether $\texttt{are-copies}(\boxed{1}, \boxed{2})$ and, hence, whether $\boxed{1} \approx \boxed{2}$.

Technicalities

Technicalities:

- If objects of sort σ are meant to be intensional, then σ is specified for **the attribute** INT **(intensional)** with values of sort σ.
- Universal **Intensionality Principle**:
  $\forall\boxed{1}\,\forall\boxed{2}\ (\boxed{1}[\text{INT}\ \boxed{2}] \rightarrow \boxed{1} \approx \boxed{2})$

For **example**, INDEX values of 2 × *she*: $\boxed{1}\begin{bmatrix} \textit{ref} \\ \text{PERS}\ \ \textit{3} \\ \text{NUM}\ \ \textit{sg} \\ \text{GEND}\ \ \textit{f} \\ \text{INT}\ \ \boxed{1} \end{bmatrix}$ and $\boxed{2}\begin{bmatrix} \textit{ref} \\ \text{PERS}\ \ \textit{3} \\ \text{NUM}\ \ \textit{sg} \\ \text{GEND}\ \ \textit{f} \\ \text{INT}\ \ \boxed{2} \end{bmatrix}$.

- **Does GIP enforce** $\boxed{1} \approx \boxed{2}$**?**
  (recall GIP: $\forall\boxed{1}\forall\boxed{2}\,(\texttt{are-copies}(\boxed{1},\boxed{2}) \rightarrow \boxed{1} \approx \boxed{2})$)
- That is, **are** $\boxed{1}$ **and** $\boxed{2}$ **copies?**
- $\texttt{are-copies}(\boxed{1},\boxed{2})$ **iff** 1) same species (**yes**), and 2) values of PERS are copies (**yes**), and 3) NUM (**yes**), and 4) GEND (**yes**), and 5) values of INT.
- So, the definition of `are-copies` **does not determine** whether $\texttt{are-copies}(\boxed{1},\boxed{2})$ and, hence, whether $\boxed{1} \approx \boxed{2}$.

Scope of the solution

This is a **complete solution** if the grammar does not produce any other cyclic structures.

If there are **other cyclic structures**:

- the grammar writer deals with that (aka 'not my problem'),
- a **more complex solution** – see the extended slides.

Scope of the solution

This is a **complete solution** if the grammar does not produce any other cyclic structures.

If there are **other cyclic structures**:

- the grammar writer deals with that (aka 'not my problem'),
- a **more complex solution** – see the extended slides.

Scope of the solution

This is a **complete solution** if the grammar does not produce any other cyclic structures.

If there are **other cyclic structures**:

- the grammar writer deals with that (aka ‘not my problem’),
- a **more complex solution** – see the extended slides.

Scope of the solution

This is a **complete solution** if the grammar does not produce any other cyclic structures.

If there are **other cyclic structures**:

- the grammar writer deals with that (aka 'not my problem'),
- a **more complex solution** – see the extended slides.

Preventing accidental intensionality 1

Typical **cyclicity** in HPSG grammars:

$$\begin{bmatrix} \textit{word} \\ \text{PHON} \ \langle \textit{the} \rangle \\ \text{SYNSEM} \ \boxed{1} \begin{bmatrix} \text{L}|\text{C}|\text{HEAD} \begin{bmatrix} \textit{det} \\ \text{SPEC}|\text{L}|\text{C} \begin{bmatrix} \text{HEAD} \ \textit{noun} \\ \text{SPR} \ \langle [\text{L}|\text{C}|\text{HEAD}|\text{SPEC} \ \boxed{1}] \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

In the following, assume the two NPs *the dog* have the same INDEX value:

- *A man with **the dog** likes **the dog**.*

Then there are **two analyses**, with the indiscernible NPs identical or not.

A **possible** (ugly) **solution**:

- **if** two *different* phrases $\boxed{1}$ and $\boxed{2}$ (here: *with the dog* and *likes the dog*)
- have on their DTRS lists elements $\boxed{1d}$ and $\boxed{2d}$ (here: 2 × *the dog*),
- such that both $\boxed{1d}$ and $\boxed{2d}$ are of appropriate shape (involve specifiers),
- **then** $\neg(\boxed{1d} \approx \boxed{2d})$.

A **more general solution**? On the next two slides.

Preventing accidental intensionality 1

Typical **cyclicity** in HPSG grammars:

$$\begin{bmatrix} \textit{word} \\ \text{PHON } \langle \textit{the} \rangle \\ \text{SYNSEM } \boxed{1} \begin{bmatrix} \text{L}|\text{C}|\text{HEAD} \begin{bmatrix} \textit{det} \\ \text{SPEC}|\text{L}|\text{C} \begin{bmatrix} \text{HEAD } \textit{noun} \\ \text{SPR } \langle [\text{L}|\text{C}|\text{HEAD}|\text{SPEC } \boxed{1}] \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

In the following, assume the two NPs *the dog* have the same INDEX value:

- *A man with* ***the dog*** *likes* ***the dog****.*

Then there are **two analyses**, with the indiscernible NPs identical or not.

A **possible** (ugly) **solution**:

- **if** two *different* phrases $\boxed{1}$ and $\boxed{2}$ (here: *with the dog* and *likes the dog*)
- have on their DTRS lists elements $\boxed{1d}$ and $\boxed{2d}$ (here: 2 × *the dog*),
- such that both $\boxed{1d}$ and $\boxed{2d}$ are of appropriate shape (involve specifiers),
- **then** $\neg(\boxed{1d} \approx \boxed{2d})$.

A **more general solution**? On the next two slides.

Preventing accidental intensionality

Typical **cyclicity** in HPSG grammars:

$$\begin{bmatrix} \textit{word} \\ \text{PHON} \ \langle \textit{the} \rangle \\ \text{SYNSEM} \ \boxed{1} \begin{bmatrix} \text{L}|\text{C}|\text{HEAD} \ \begin{bmatrix} \textit{det} \\ \text{SPEC}|\text{L}|\text{C} \ \begin{bmatrix} \text{HEAD} \ \textit{noun} \\ \text{SPR} \ \langle [\text{L}|\text{C}|\text{HEAD}|\text{SPEC} \ \boxed{1}] \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

In the following, assume the two NPs *the dog* have the same INDEX value:

- *A man with **the dog** likes **the dog**.*

Then there are **two analyses**, with the indiscernible NPs identical or not.

A **possible** (ugly) **solution**:

- **if** two *different* phrases $\boxed{1}$ and $\boxed{2}$ (here: *with the dog* and *likes the dog*)
- have on their DTRS lists elements $\boxed{1d}$ and $\boxed{2d}$ (here: 2 × *the dog*),
- such that both $\boxed{1d}$ and $\boxed{2d}$ are of appropriate shape (involve specifiers),
- **then** $\neg(\boxed{1d} \approx \boxed{2d})$.

A **more general solution**? On the next two slides.

Preventing accidental intensionality 1

Typical **cyclicity** in HPSG grammars:

$$\begin{bmatrix} \textit{word} \\ \text{PHON} \ \langle \textit{the} \rangle \\ \text{SYNSEM} \ \boxed{1} \begin{bmatrix} \text{L}|\text{C}|\text{HEAD} \ \begin{bmatrix} \textit{det} \\ \text{SPEC}|\text{L}|\text{C} \ \begin{bmatrix} \text{HEAD} \ \textit{noun} \\ \text{SPR} \ \langle [\text{L}|\text{C}|\text{HEAD}|\text{SPEC} \ \boxed{1}] \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

In the following, assume the two NPs *the dog* have the same INDEX value:

- *A man with **the dog** likes **the dog**.*

Then there are **two analyses**, with the indiscernible NPs identical or not.

A possible (ugly) **solution**:

- **if** two *different* phrases $\boxed{1}$ and $\boxed{2}$ (here: *with the dog* and *likes the dog*)
- have on their DTRS lists elements $\boxed{1d}$ and $\boxed{2d}$ (here: $2 \times$ *the dog*),
- such that both $\boxed{1d}$ and $\boxed{2d}$ are of appropriate shape (involve specifiers),
- **then** $\neg(\boxed{1d} \approx \boxed{2d})$.

A **more general solution**? On the next two slides.

Preventing accidental intensionality 1

Typical **cyclicity** in HPSG grammars:

$$\begin{bmatrix} \textit{word} \\ \text{PHON} \ \langle \textit{the} \rangle \\ \text{SYNSEM} \ \boxed{1} \begin{bmatrix} \text{L}|\text{C}|\text{HEAD} \begin{bmatrix} \textit{det} \\ \text{SPEC}|\text{L}|\text{C} \begin{bmatrix} \text{HEAD} \ \textit{noun} \\ \text{SPR} \ \langle [\text{L}|\text{C}|\text{HEAD}|\text{SPEC} \ \boxed{1}] \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

In the following, assume the two NPs *the dog* have the same INDEX value:

- *A man with **the dog** likes **the dog**.*

Then there are **two analyses**, with the indiscernible NPs identical or not.

A possible (ugly) **solution**:

- **if** two *different* phrases $\boxed{1}$ and $\boxed{2}$ (here: *with the dog* and *likes the dog*)
- have on their DTRS lists elements $\boxed{1d}$ and $\boxed{2d}$ (here: 2 $\times$ *the dog*),
- such that both $\boxed{1d}$ and $\boxed{2d}$ are of appropriate shape (involve specifiers),
- **then** $\neg(\boxed{1d} \approx \boxed{2d})$.

A **more general solution**? On the next two slides.

Preventing accidental intensionality 2

Sailer 2003: 116:

$$\forall \boxed{1} \forall \boxed{2}\ \texttt{are-copies}(\boxed{1}, \boxed{2}) \leftrightarrow$$

$$\left(\bigvee_{\sigma \in \mathcal{S}} (\boxed{1} \sim \sigma \wedge \boxed{2} \sim \sigma) \wedge \bigwedge_{\alpha \in \mathcal{A}} (\boxed{1}\alpha \approx \boxed{1}\alpha \rightarrow \texttt{are-copies}(\boxed{1}\alpha, \boxed{2}\alpha)) \right)$$

Idea: **recursion with memory**, immune to cycles – **nothing is intensional**:

$$\forall \boxed{1} \forall \boxed{2}\ \texttt{are-copies}(\boxed{1}, \boxed{2}) \leftrightarrow \texttt{Rcopies}(\langle\,\rangle, \boxed{1}, \langle\,\rangle, \boxed{2})$$

$$\forall \boxed{1} \forall \boxed{2}\ \texttt{Rcopies}(\boxed{\llcorner 1}, \boxed{1}, \boxed{\llcorner 2}, \boxed{2}) \leftrightarrow$$

$$\texttt{member2}(\boxed{1}, \boxed{\llcorner 1}, \boxed{2}, \boxed{\llcorner 2}) \vee$$

$$\left(\bigvee_{\sigma \in \mathcal{S}} (\boxed{1} \sim \sigma \wedge \boxed{2} \sim \sigma) \wedge \bigwedge_{\alpha \in \mathcal{A}} (\boxed{1}\alpha \approx \boxed{1}\alpha \rightarrow \texttt{Rcopies}(\langle \boxed{1} | \boxed{\llcorner 1} \rangle, \boxed{1}\alpha, \langle \boxed{2} | \boxed{\llcorner 2} \rangle, \boxed{2}\alpha)) \right)$$

$$\texttt{member2}(\boxed{1}, \langle \boxed{\llcorner 1 h} | \boxed{\llcorner 1 t} \rangle, \boxed{2}, \langle \boxed{\llcorner 2 h} | \boxed{\llcorner 2 t} \rangle) \leftrightarrow$$

$$(\boxed{1} \approx \boxed{\llcorner 1 h} \wedge \boxed{2} \approx \boxed{\llcorner 2 h}) \vee \texttt{member2}(\boxed{1}, \boxed{\llcorner 1 t}, \boxed{2}, \boxed{\llcorner 2 t})$$

Preventing accidental intensionality 2

Sailer 2003: 116:

$$\forall \boxed{1} \forall \boxed{2}\ \texttt{are-copies}(\boxed{1}, \boxed{2}) \leftrightarrow$$
$$\left(\bigvee_{\sigma \in \mathcal{S}} (\boxed{1} \sim \sigma \wedge \boxed{2} \sim \sigma) \wedge \bigwedge_{\alpha \in \mathcal{A}} (\boxed{1}\alpha \approx \boxed{1}\alpha \rightarrow \texttt{are-copies}(\boxed{1}\alpha, \boxed{2}\alpha)) \right)$$

Idea: **recursion with memory**, immune to cycles – **nothing is intensional**:

$$\forall \boxed{1} \forall \boxed{2}\ \texttt{are-copies}(\boxed{1}, \boxed{2}) \leftrightarrow \texttt{Rcopies}(\langle\,\rangle, \boxed{1}, \langle\,\rangle, \boxed{2})$$

$$\forall \boxed{1} \forall \boxed{2}\ \texttt{Rcopies}(\boxed{L1}, \boxed{1}, \boxed{L2}, \boxed{2}) \leftrightarrow$$
$$\texttt{member2}(\boxed{1}, \boxed{L1}, \boxed{2}, \boxed{L2}) \vee$$
$$\left(\bigvee_{\sigma \in \mathcal{S}} (\boxed{1} \sim \sigma \wedge \boxed{2} \sim \sigma) \wedge \bigwedge_{\alpha \in \mathcal{A}} (\boxed{1}\alpha \approx \boxed{1}\alpha \rightarrow \texttt{Rcopies}(\langle \boxed{1} | \boxed{L1} \rangle, \boxed{1}\alpha, \langle \boxed{2} | \boxed{L2} \rangle, \boxed{2}\alpha)) \right)$$

$$\texttt{member2}(\boxed{1}, \langle \boxed{L1h} | \boxed{L1t} \rangle, \boxed{2}, \langle \boxed{L2h} | \boxed{L2t} \rangle) \leftrightarrow$$
$$(\boxed{1} \approx \boxed{L1h} \wedge \boxed{2} \approx \boxed{L2h}) \vee \texttt{member2}(\boxed{1}, \boxed{L1t}, \boxed{2}, \boxed{L2t})$$

Preventing accidental intensionality 2

Sailer 2003: 116:

$$\forall \boxed{1} \forall \boxed{2}\ \texttt{are-copies}(\boxed{1}, \boxed{2}) \leftrightarrow \left(\bigvee_{\sigma \in \mathcal{S}} (\boxed{1} \sim \sigma \wedge \boxed{2} \sim \sigma) \wedge \bigwedge_{\alpha \in \mathcal{A}} (\boxed{1}\alpha \approx \boxed{1}\alpha \rightarrow \texttt{are-copies}(\boxed{1}\alpha, \boxed{2}\alpha)) \right)$$

Idea: **recursion with memory**, immune to cycles – **nothing is intensional**:

$$\forall \boxed{1} \forall \boxed{2}\ \texttt{are-copies}(\boxed{1}, \boxed{2}) \leftrightarrow \texttt{Rcopies}(\langle\,\rangle, \boxed{1}, \langle\,\rangle, \boxed{2})$$

$$\forall \boxed{1} \forall \boxed{2}\ \texttt{Rcopies}(\boxed{L1}, \boxed{1}, \boxed{L2}, \boxed{2}) \leftrightarrow \texttt{member2}(\boxed{1}, \boxed{L1}, \boxed{2}, \boxed{L2}) \vee \left(\bigvee_{\sigma \in \mathcal{S}} (\boxed{1} \sim \sigma \wedge \boxed{2} \sim \sigma) \wedge \bigwedge_{\alpha \in \mathcal{A}} (\boxed{1}\alpha \approx \boxed{1}\alpha \rightarrow \texttt{Rcopies}(\langle \boxed{1} | \boxed{L1} \rangle, \boxed{1}\alpha, \langle \boxed{2} | \boxed{L2} \rangle, \boxed{2}\alpha)) \right)$$

$$\texttt{member2}(\boxed{1}, \langle \boxed{L1h} | \boxed{L1t} \rangle, \boxed{2}, \langle \boxed{L2h} | \boxed{L2t} \rangle) \leftrightarrow (\boxed{1} \approx \boxed{L1h} \wedge \boxed{2} \approx \boxed{L2h}) \vee \texttt{member2}(\boxed{1}, \boxed{L1t}, \boxed{2}, \boxed{L2t})$$

Preventing accidental intensionality 2

Sailer 2003: 116:

$$\forall \boxed{1} \forall \boxed{2}\ \texttt{are-copies}(\boxed{1}, \boxed{2}) \leftrightarrow$$
$$\left(\bigvee_{\sigma \in \mathcal{S}} (\boxed{1} \sim \sigma \wedge \boxed{2} \sim \sigma) \wedge \bigwedge_{\alpha \in \mathcal{A}} (\boxed{1}\alpha \approx \boxed{1}\alpha \rightarrow \texttt{are-copies}(\boxed{1}\alpha, \boxed{2}\alpha)) \right)$$

Idea: **recursion with memory**, immune to cycles – **nothing is intensional**:

$$\forall \boxed{1} \forall \boxed{2}\ \texttt{are-copies}(\boxed{1}, \boxed{2}) \leftrightarrow \texttt{Rcopies}(\langle\,\rangle, \boxed{1}, \langle\,\rangle, \boxed{2})$$

$$\forall \boxed{1} \forall \boxed{2}\ \texttt{Rcopies}(\boxed{L1}, \boxed{1}, \boxed{L2}, \boxed{2}) \leftrightarrow$$
$$\texttt{member2}(\boxed{1}, \boxed{L1}, \boxed{2}, \boxed{L2}) \vee$$
$$\left(\bigvee_{\sigma \in \mathcal{S}} (\boxed{1} \sim \sigma \wedge \boxed{2} \sim \sigma) \wedge \bigwedge_{\alpha \in \mathcal{A}} (\boxed{1}\alpha \approx \boxed{1}\alpha \rightarrow \texttt{Rcopies}(\langle \boxed{1} | \boxed{L1} \rangle, \boxed{1}\alpha, \langle \boxed{2} | \boxed{L2} \rangle, \boxed{2}\alpha)) \right)$$

$$\texttt{member2}(\boxed{1}, \langle \boxed{L1h} | \boxed{L1t} \rangle, \boxed{2}, \langle \boxed{L2h} | \boxed{L2t} \rangle) \leftrightarrow$$
$$(\boxed{1} \approx \boxed{L1h} \wedge \boxed{2} \approx \boxed{L2h}) \vee \texttt{member2}(\boxed{1}, \boxed{L1t}, \boxed{2}, \boxed{L2t})$$

Preventing accidental intensionality 2

Sailer 2003: 116:

$$\forall \boxed{1} \forall \boxed{2}\ \texttt{are-copies}(\boxed{1}, \boxed{2}) \leftrightarrow$$
$$\left(\bigvee_{\sigma \in \mathcal{S}} (\boxed{1} \sim \sigma \wedge \boxed{2} \sim \sigma) \wedge \bigwedge_{\alpha \in \mathcal{A}} (\boxed{1}\alpha \approx \boxed{1}\alpha \rightarrow \texttt{are-copies}(\boxed{1}\alpha, \boxed{2}\alpha)) \right)$$

Idea: **recursion with memory**, immune to cycles – **nothing is intensional**:

$$\forall \boxed{1} \forall \boxed{2}\ \texttt{are-copies}(\boxed{1}, \boxed{2}) \leftrightarrow \texttt{Rcopies}(\langle\ \rangle, \boxed{1}, \langle\ \rangle, \boxed{2})$$

$$\forall \boxed{1} \forall \boxed{2}\ \texttt{Rcopies}(\boxed{L1}, \boxed{1}, \boxed{L2}, \boxed{2}) \leftrightarrow$$
$$\texttt{member2}(\boxed{1}, \boxed{L1}, \boxed{2}, \boxed{L2}) \vee$$
$$\left(\bigvee_{\sigma \in \mathcal{S}} (\boxed{1} \sim \sigma \wedge \boxed{2} \sim \sigma) \wedge \bigwedge_{\alpha \in \mathcal{A}} (\boxed{1}\alpha \approx \boxed{1}\alpha \rightarrow \texttt{Rcopies}(\langle \boxed{1} | \boxed{L1} \rangle, \boxed{1}\alpha, \langle \boxed{2} | \boxed{L2} \rangle, \boxed{2}\alpha)) \right)$$

$$\texttt{member2}(\boxed{1}, \langle \boxed{L1h} | \boxed{L1t} \rangle, \boxed{2}, \langle \boxed{L2h} | \boxed{L2t} \rangle) \leftrightarrow$$
$$(\boxed{1} \approx \boxed{L1h} \wedge \boxed{2} \approx \boxed{L2h}) \vee \texttt{member2}(\boxed{1}, \boxed{L1t}, \boxed{2}, \boxed{L2t})$$

Preventing accidental intensionality

Now we need to **make some structures intensional again**:

$$\forall \boxed{1} \forall \boxed{2}\ \texttt{Rcopies}(\boxed{L1}, \boxed{1}, \boxed{L2}, \boxed{2}) \leftrightarrow$$

$$\texttt{member2}(\boxed{1}, \boxed{L1}, \boxed{2}, \boxed{L2}) \lor$$

$$\bigvee_{\sigma \in \mathcal{S}} (\boxed{1} \sim \sigma \land \boxed{2} \sim \sigma) \land$$

$$\bigwedge_{\alpha \in \mathcal{A}} (\boxed{1}\alpha \approx \boxed{1}\alpha \rightarrow \texttt{Rcopies}(\langle \boxed{1} | \boxed{L1} \rangle, \boxed{1}\alpha, \langle \boxed{2} | \boxed{L2} \rangle, \boxed{2}\alpha)) \land$$

$$\boxed{1} \sim \mathit{ref} \rightarrow \texttt{Rcopies}(\boxed{L1}, \boxed{1}, \boxed{L2}, \boxed{2})$$

$$\forall \boxed{1}\ \texttt{int}(\boxed{1}) \leftrightarrow (\boxed{1} \sim \mathit{ref} \lor \ldots)$$

Preventing accidental intensionality

Now we need to **make some structures intensional again**:

$$\forall \boxed{1} \forall \boxed{2}\ \texttt{Rcopies}(\boxed{L1}, \boxed{1}, \boxed{L2}, \boxed{2}) \leftrightarrow$$

$$\texttt{member2}(\boxed{1}, \boxed{L1}, \boxed{2}, \boxed{L2}) \ \lor$$

$$\bigvee_{\sigma \in \mathcal{S}} (\boxed{1} \sim \sigma \land \boxed{2} \sim \sigma) \ \land$$

$$\bigwedge_{\alpha \in \mathcal{A}} (\boxed{1}\alpha \approx \boxed{1}\alpha \rightarrow \texttt{Rcopies}(\langle \boxed{1} | \boxed{L1} \rangle, \boxed{1}\alpha, \langle \boxed{2} | \boxed{L2} \rangle, \boxed{2}\alpha)) \ \land$$

$$\boxed{1} \sim \mathit{ref} \rightarrow \texttt{Rcopies}(\boxed{L1}, \boxed{1}, \boxed{L2}, \boxed{2})$$

$$\forall \boxed{1}\ \texttt{int}(\boxed{1}) \leftrightarrow (\boxed{1} \sim \mathit{ref} \lor \ldots)$$

Preventing accidental intensionality

Now we need to **make some structures intensional again**:

$$\forall \boxed{1} \forall \boxed{2}\ \mathtt{Rcopies}(\boxed{L1}, \boxed{1}, \boxed{L2}, \boxed{2}) \leftrightarrow$$

$$\mathtt{member2}(\boxed{1}, \boxed{L1}, \boxed{2}, \boxed{L2}) \ \vee$$

$$\bigvee_{\sigma \in \mathcal{S}} (\boxed{1} \sim \sigma \wedge \boxed{2} \sim \sigma) \ \wedge$$

$$\bigwedge_{\alpha \in \mathcal{A}} (\boxed{1}\alpha \approx \boxed{1}\alpha \rightarrow \mathtt{Rcopies}(\langle \boxed{1} | \boxed{L1} \rangle, \boxed{1}\alpha, \langle \boxed{2} | \boxed{L2} \rangle, \boxed{2}\alpha)) \ \wedge$$

$$\boxed{1} \sim \mathit{ref} \rightarrow \mathtt{Rcopies}(\boxed{L1}, \boxed{1}, \boxed{L2}, \boxed{2})$$

$$\forall \boxed{1}\ \mathtt{int}(\boxed{1}) \leftrightarrow (\boxed{1} \sim \mathit{ref} \vee \ldots)$$

Preventing accidental intensionality

Now we need to **make some structures intensional again**:

$$\forall \boxed{1} \forall \boxed{2}\ \texttt{Rcopies}(\boxed{\text{L}1}, \boxed{1}, \boxed{\text{L}2}, \boxed{2}) \leftrightarrow$$

$$\texttt{member2}(\boxed{1}, \boxed{\text{L}1}, \boxed{2}, \boxed{\text{L}2}) \lor$$

$$\bigvee_{\sigma \in \mathcal{S}} (\boxed{1} \sim \sigma \land \boxed{2} \sim \sigma) \land$$

$$\bigwedge_{\alpha \in \mathcal{A}} (\boxed{1}\alpha \approx \boxed{1}\alpha \rightarrow \texttt{Rcopies}(\langle \boxed{1} | \boxed{\text{L}1} \rangle, \boxed{1}\alpha, \langle \boxed{2} | \boxed{\text{L}2} \rangle, \boxed{2}\alpha)) \land$$

$$\boxed{1} \sim \mathit{ref} \rightarrow \texttt{Rcopies}(\boxed{\text{L}1}, \boxed{1}, \boxed{\text{L}2}, \boxed{2})$$

$$\forall \boxed{1}\ \texttt{int}(\boxed{1}) \leftrightarrow (\boxed{1} \sim \mathit{ref} \lor \ldots)$$

Preventing accidental intensionality

Now we need to **make some structures intensional again**:

$$\forall \boxed{1} \forall \boxed{2}\ \texttt{Rcopies}(\boxed{\llcorner 1}, \boxed{1}, \boxed{\llcorner 2}, \boxed{2}) \leftrightarrow$$

$$\texttt{member2}(\boxed{1}, \boxed{\llcorner 1}, \boxed{2}, \boxed{\llcorner 2}) \ \vee$$

$$\bigvee_{\sigma \in \mathcal{S}} (\boxed{1} \sim \sigma \wedge \boxed{2} \sim \sigma) \ \wedge$$

$$\bigwedge_{\alpha \in \mathcal{A}} (\boxed{1}\alpha \approx \boxed{1}\alpha \rightarrow \texttt{Rcopies}(\langle \boxed{1} | \boxed{\llcorner 1} \rangle, \boxed{1}\alpha, \langle \boxed{2} | \boxed{\llcorner 2} \rangle, \boxed{2}\alpha)) \ \wedge$$

$$\texttt{int}(\boxed{1}) \rightarrow \texttt{Rcopies}(\boxed{\llcorner 1}, \boxed{1}, \boxed{\llcorner 2}, \boxed{2})$$

$$\forall \boxed{1}\ \texttt{int}(\boxed{1}) \leftrightarrow (\boxed{1} \sim \mathit{ref} \vee \ldots)$$

Exhaustive models

King 1999 (and Richter 2004, 2007): **(R)SRL models are exhaustive**:

- **each** licensed **configuration** must actually be **present in the model**,
- intuition: (R)SRL models are **models of whole languages**, not models of single utterances.

This idea leads to **multiple problems** (cf. Richter 2007):

- **twin structures**: shared by different utterances,
- **(monster) stranded structures**: not parts of utterances,
- **ontological problem**: compatibility with the idea that objects in models are linguistic tokens (rather than mathematical idealisations).

Also:

- this intuition is **not shared by other mathematical / formal linguists** (Blackburn, Pullum, Kaplan…),
- for them, models are **models of single utterances**,
- **languages** – if defined at all – are **collections of models**.

Exhaustive models

King 1999 (and Richter 2004, 2007): **(R)SRL models are exhaustive**:

- **each** licensed **configuration** must actually be **present in the model**,
- intuition: (R)SRL models are **models of whole languages**, not models of single utterances.

This idea leads to **multiple problems** (cf. Richter 2007):

- **twin structures**: shared by different utterances,
- **(monster) stranded structures**: not parts of utterances,
- **ontological problem**: compatibility with the idea that objects in models are linguistic tokens (rather than mathematical idealisations).

Also:

- this intuition is **not shared by other mathematical / formal linguists** (Blackburn, Pullum, Kaplan…),
- for them, models are **models of single utterances**,
- **languages** – if defined at all – are **collections of models**.

Exhaustive models

King 1999 (and Richter 2004, 2007): **(R)SRL models are exhaustive**:

- **each** licensed **configuration** must actually be **present in the model**,
- intuition: (R)SRL models are **models of whole languages**, not models of single utterances.

This idea leads to **multiple problems** (cf. Richter 2007):

- **twin structures**: shared by different utterances,
- **(monster) stranded structures**: not parts of utterances,
- **ontological problem**: compatibility with the idea that objects in models are linguistic tokens (rather than mathematical idealisations).

Also:

- this intuition is **not shared by other mathematical / formal linguists** (Blackburn, Pullum, Kaplan. . .),
- for them, models are **models of single utterances**,
- **languages** – if defined at all – are **collections of models**.

Exhaustive models

King 1999 (and Richter 2004, 2007): **(R)SRL models are exhaustive**:

- **each** licensed **configuration** must actually be **present in the model**,
- intuition: (R)SRL models are **models of whole languages**, not models of single utterances.

This idea leads to **multiple problems** (cf. Richter 2007):

- **twin structures**: shared by different utterances,
- **(monster) stranded structures**: not parts of utterances,
- **ontological problem**: compatibility with the idea that objects in models are linguistic tokens (rather than mathematical idealisations).

Also:

- this intuition is **not shared by other mathematical / formal linguists** (Blackburn, Pullum, Kaplan…),
- for them, models are **models of single utterances**,
- **languages** – if defined at all – are **collections of models**.

Exhaustive models

King 1999 (and Richter 2004, 2007): **(R)SRL models are exhaustive**:

- **each** licensed **configuration** must actually be **present in the model**,
- intuition: (R)SRL models are **models of whole languages**, not models of single utterances.

This idea leads to **multiple problems** (cf. Richter 2007):

- **twin structures**: shared by different utterances,
- **(monster) stranded structures**: not parts of utterances,
- **ontological problem**: compatibility with the idea that objects in models are linguistic tokens (rather than mathematical idealisations).

Also:

- this intuition is **not shared by other mathematical / formal linguists** (Blackburn, Pullum, Kaplan...),
- for them, models are **models of single utterances**,
- **languages** – if defined at all – are **collections of models**.

Exhaustive models

King 1999 (and Richter 2004, 2007): **(R)SRL models are exhaustive**:

- **each** licensed **configuration** must actually be **present in the model**,
- intuition: (R)SRL models are **models of whole languages**, not models of single utterances.

This idea leads to **multiple problems** (cf. Richter 2007):

- **twin structures**: shared by different utterances,
- **(monster) stranded structures**: not parts of utterances,
- **ontological problem**: compatibility with the idea that objects in models are linguistic tokens (rather than mathematical idealisations).

Also:

- this intuition is **not shared by other mathematical / formal linguists** (Blackburn, Pullum, Kaplan...),
- for them, models are **models of single utterances**,
- **languages** – if defined at all – are **collections of models**.

Exhaustive models

King 1999 (and Richter 2004, 2007): **(R)SRL models are exhaustive**:

- **each** licensed **configuration** must actually be **present in the model**,
- intuition: (R)SRL models are **models of whole languages**, not models of single utterances.

This idea leads to **multiple problems** (cf. Richter 2007):

- **twin structures**: shared by different utterances,
- **(monster) stranded structures**: not parts of utterances,
- **ontological problem**: compatibility with the idea that objects in models are linguistic tokens (rather than mathematical idealisations).

Also:

- this intuition is **not shared by other mathematical / formal linguists** (Blackburn, Pullum, Kaplan…),
- for them, models are **models of single utterances**,
- **languages** – if defined at all – are **collections of models**.

Exhaustive models

King 1999 (and Richter 2004, 2007): **(R)SRL models are exhaustive**:

- **each** licensed **configuration** must actually be **present in the model**,
- intuition: (R)SRL models are **models of whole languages**, not models of single utterances.

This idea leads to **multiple problems** (cf. Richter 2007):

- **twin structures**: shared by different utterances,
- **(monster) stranded structures**: not parts of utterances,
- **ontological problem**: compatibility with the idea that objects in models are linguistic tokens (rather than mathematical idealisations).

Also:

- this intuition is **not shared by other mathematical / formal linguists** (Blackburn, Pullum, Kaplan...),
- for them, models are **models of single utterances**,
- **languages** – if defined at all – are **collections of models**.

Exhaustive models

King 1999 (and Richter 2004, 2007): **(R)SRL models are exhaustive**:

- **each** licensed **configuration** must actually be **present in the model**,
- intuition: (R)SRL models are **models of whole languages**, not models of single utterances.

This idea leads to **multiple problems** (cf. Richter 2007):

- **twin structures**: shared by different utterances,
- **(monster) stranded structures**: not parts of utterances,
- **ontological problem**: compatibility with the idea that objects in models are linguistic tokens (rather than mathematical idealisations).

Also:

- this intuition is **not shared by other mathematical / formal linguists** (Blackburn, Pullum, Kaplan. . .),
- for them, models are **models of single utterances**,
- **languages** – if defined at all – are **collections of models**.

Exhaustive models

King 1999 (and Richter 2004, 2007): **(R)SRL models are exhaustive**:

- **each** licensed **configuration** must actually be **present in the model**,
- intuition: (R)SRL models are **models of whole languages**, not models of single utterances.

This idea leads to **multiple problems** (cf. Richter 2007):

- **twin structures**: shared by different utterances,
- **(monster) stranded structures**: not parts of utterances,
- **ontological problem**: compatibility with the idea that objects in models are linguistic tokens (rather than mathematical idealisations).

Also:

- this intuition is **not shared by other mathematical / formal linguists** (Blackburn, Pullum, Kaplan…),
- for them, models are **models of single utterances**,
- **languages** – if defined at all – are **collections of models**.

Problem: Twin structures

Richter 2007: nothing in (R)SRL prevents configurations in which **different utterances share structure**.

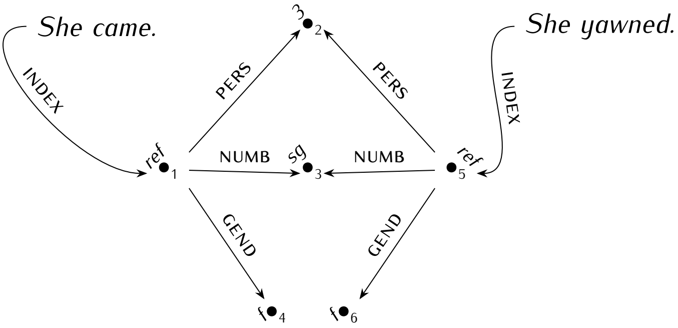
For **example**:

Of course, **much larger configurations may be shared** (e.g., the whole word *she*).

Problem: Twin structures

Richter 2007: nothing in (R)SRL prevents configurations in which **different utterances share structure**.

For **example**:

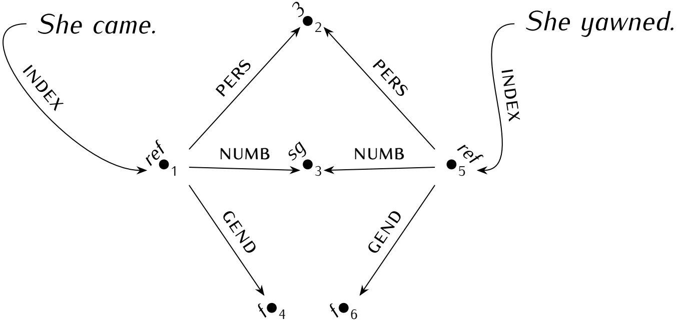


Of course, **much larger configurations may be shared** (e.g., the whole word *she*).

Problem: Twin structures

Richter 2007: nothing in (R)SRL prevents configurations in which **different utterances share structure**.

For **example**:



Of course, **much larger configurations may be shared** (e.g., the whole word *she*).

Problem: Stranded structures

Richter 2007: a typical HPSG grammar licenses as separate configurations, e.g., ***phrase*s with nonempty SLASH values**.

Elements of SLASH are **of sort *local***.

Various HPSG principles formulated as **constraints on *sign*s constrain *local* configurations** within those *sign*s.

But **unrealised elements of SLASH are not subject to such principles**; they may contain very strange *local* objects.

Hence, exhaustive models contain **configurations of *phrase*s** (with non-empty SLASH values) that **never occur in utterances**.

Problem: Stranded structures

Richter 2007: a typical HPSG grammar licenses as separate configurations, e.g., ***phrases* with nonempty SLASH values**.

Elements of SLASH are **of sort *local***.

Various HPSG principles formulated as **constraints on *signs* constrain *local* configurations** within those *signs*.

But **unrealised elements of SLASH are not subject to such principles**; they may contain very strange *local* objects.

Hence, exhaustive models contain **configurations of *phrases*** (with non-empty SLASH values) that **never occur in utterances**.

Problem: Stranded structures

Richter 2007: a typical HPSG grammar licenses as separate configurations, e.g., ***phrase*s with nonempty SLASH values**.

Elements of SLASH are **of sort *local***.

Various HPSG principles formulated as **constraints on *sign*s constrain *local* configurations** within those *sign*s.

But **unrealised elements of SLASH are not subject to such principles**; they may contain very strange *local* objects.

Hence, exhaustive models contain **configurations of *phrases*** (with non-empty SLASH values) that **never occur in utterances**.

Problem: Stranded structures

Richter 2007: a typical HPSG grammar licenses as separate configurations, e.g., ***phrase*s with nonempty SLASH values**.

Elements of SLASH are **of sort *local***.

Various HPSG principles formulated as **constraints on *sign*s constrain *local* configurations** within those *sign*s.

But **unrealised elements of SLASH are not subject to such principles**; they may contain very strange *local* objects.

Hence, exhaustive models contain **configurations of *phrases*** (with non-empty SLASH values) that **never occur in utterances**.

Problem: Stranded structures

Richter 2007: a typical HPSG grammar licenses as separate configurations, e.g., ***phrase*s with nonempty SLASH values**.

Elements of SLASH are **of sort *local***.

Various HPSG principles formulated as **constraints on *sign*s constrain *local* configurations** within those *sign*s.

But **unrealised elements of SLASH are not subject to such principles**; they may contain very strange *local* objects.

Hence, exhaustive models contain **configurations of *phrases*** (with non-empty SLASH values) that **never occur in utterances**.

Solution: Richter 2007

Extension of signature:

top EMBEDDED *u_sign*
sign ...
e_sign
e_word
e_phrase
u_sign
u_word
u_phrase
word
e_word
u_word
phrase H_DTR *e_sign*
NH_DTR *e_sign*
e_phrase
u_phrase
...
component/2

Extension of theory:

- UNIQUE U-SIGN CONDITION:
  $\forall \boxed{1} \forall \boxed{2} ((\boxed{1} \sim u\_sign \wedge \boxed{2} \sim u\_sign) \rightarrow \boxed{1} \approx \boxed{2})$
- U-SIGN COMPONENT CONDITION:
  $\forall \boxed{1} (\boxed{1} \sim top \rightarrow \exists \boxed{2} (\boxed{2} \sim u\_sign \wedge \mathtt{component}(\boxed{1}, \boxed{2})))$

Solves the above problems:

- **no twin structures**: everything is a component of a single *u_sign*,
- **no stranded structures**: ditto (assuming *u_sign* → empty SLASH, etc.).

At a cost:

- **more complex signature** (and theory),
- **conceptual problem**: the value of, e.g., CASE contains the whole utterance?

Solution: Richter 2007

Extension of signature:

top EMBEDDED *u_sign*
  sign ...
    e_sign
      e_word
      e_phrase
    u_sign
      u_word
      u_phrase
    word
      e_word
      u_word
    phrase H_DTR *e_sign*
        NH_DTR *e_sign*
      e_phrase
      u_phrase
...
`component/2`

Extension of theory:

- UNIQUE U-SIGN CONDITION:
  $\forall \boxed{1} \forall \boxed{2} ((\boxed{1} \sim \textit{u\_sign} \land \boxed{2} \sim \textit{u\_sign}) \rightarrow \boxed{1} \approx \boxed{2})$
- U-SIGN COMPONENT CONDITION:
  $\forall \boxed{1} (\boxed{1} \sim \textit{top} \rightarrow \exists \boxed{2} (\boxed{2} \sim \textit{u\_sign} \land \texttt{component}(\boxed{1}, \boxed{2})))$

Solves the above problems:

- **no twin structures**: everything is a component of a single *u_sign*,
- **no stranded structures**: ditto (assuming *u_sign* → empty SLASH, etc.).

At a cost:

- **more complex signature** (and theory),
- **conceptual problem**: the value of, e.g., CASE contains the whole utterance?

Solution: Richter 2007

Extension of signature:

top EMBEDDED *u_sign*
- *sign* …
  - *e_sign*
    - *e_word*
    - *e_phrase*
  - *u_sign*
    - *u_word*
    - *u_phrase*
  - *word*
    - *e_word*
    - *u_word*
  - *phrase* H_DTR *e_sign*
    NH_DTR *e_sign*
    - *e_phrase*
    - *u_phrase*

…

`component/2`

Extension of theory:

- UNIQUE U-SIGN CONDITION:
  $\forall \boxed{1} \forall \boxed{2} ((\boxed{1} \sim u\_sign \wedge \boxed{2} \sim u\_sign) \rightarrow \boxed{1} \approx \boxed{2})$
- U-SIGN COMPONENT CONDITION:
  $\forall \boxed{1} (\boxed{1} \sim top \rightarrow \exists \boxed{2} (\boxed{2} \sim u\_sign \wedge \texttt{component}(\boxed{1}, \boxed{2})))$

Solves the above problems:

- **no twin structures**: everything is a component of a single *u_sign*,
- **no stranded structures**: ditto (assuming *u_sign* → empty SLASH, etc.).

At a cost:

- **more complex signature** (and theory),
- **conceptual problem**: the value of, e.g., CASE contains the whole utterance?

Solution: Richter 2007

Extension of signature:

top EMBEDDED *u_sign*
- *sign* ...
  - *e_sign*
    - *e_word*
    - *e_phrase*
  - *u_sign*
    - *u_word*
    - *u_phrase*
  - *word*
    - *e_word*
    - *u_word*
  - *phrase* H_DTR *e_sign*
    NH_DTR *e_sign*
    - *e_phrase*
    - *u_phrase*

...

`component/2`

Extension of theory:

- UNIQUE U-SIGN CONDITION:
  $\forall \boxed{1} \forall \boxed{2} ((\boxed{1} \sim u\_sign \wedge \boxed{2} \sim u\_sign) \rightarrow \boxed{1} \approx \boxed{2})$
- U-SIGN COMPONENT CONDITION:
  $\forall \boxed{1} (\boxed{1} \sim top \rightarrow \exists \boxed{2} (\boxed{2} \sim u\_sign \wedge \texttt{component}(\boxed{1}, \boxed{2})))$

Solves the above problems:

- **no twin structures**: everything is a component of a single *u_sign*,
- **no stranded structures**: ditto (assuming *u_sign* → empty SLASH, etc.).

At a cost:

- **more complex signature** (and theory),
- **conceptual problem**: the value of, e.g., CASE contains the whole utterance?

Solution: Richter 2007

Extension of signature:

top EMBEDDED *u_sign*
  sign . . .
    e_sign
      e_word
      e_phrase
    u_sign
      u_word
      u_phrase
    word
      e_word
      u_word
    phrase H_DTR *e_sign*
          NH_DTR *e_sign*
      e_phrase
      u_phrase
...
component/2

Extension of theory:

- UNIQUE U-SIGN CONDITION:
  $\forall \boxed{1} \forall \boxed{2} ((\boxed{1} \sim \mathit{u\_sign} \wedge \boxed{2} \sim \mathit{u\_sign}) \rightarrow \boxed{1} \approx \boxed{2})$
- U-SIGN COMPONENT CONDITION:
  $\forall \boxed{1} (\boxed{1} \sim \mathit{top} \rightarrow \exists \boxed{2} (\boxed{2} \sim \mathit{u\_sign} \wedge \mathtt{component}(\boxed{1}, \boxed{2})))$

Solves the above problems:

- **no twin structures**: everything is a component of a single *u_sign*,
- **no stranded structures**: ditto (assuming *u_sign* → empty SLASH, etc.).

At a cost:

- **more complex signature** (and theory),
- **conceptual problem**: the value of, e.g., CASE contains the whole utterance?

Solution: Richter 2007

Extension of signature:

top EMBEDDED *u_sign*
  sign ...
    e_sign
      e_word
      e_phrase
    u_sign
      u_word
      u_phrase
    word
      e_word
      u_word
    phrase H_DTR *e_sign*
          NH_DTR *e_sign*
      e_phrase
      u_phrase
...
`component/2`

Extension of theory:

- UNIQUE U-SIGN CONDITION:
  $\forall \boxed{1} \forall \boxed{2} ((\boxed{1} \sim u\_sign \wedge \boxed{2} \sim u\_sign) \rightarrow \boxed{1} \approx \boxed{2})$
- U-SIGN COMPONENT CONDITION:
  $\forall \boxed{1} (\boxed{1} \sim top \rightarrow \exists \boxed{2} (\boxed{2} \sim u\_sign \wedge \texttt{component}(\boxed{1}, \boxed{2})))$

Solves the above problems:

- **no twin structures**: everything is a component of a single *u_sign*,
- **no stranded structures**: ditto (assuming *u_sign* → empty SLASH, etc.).

At a cost:

- **more complex signature** (and theory),
- **conceptual problem**: the value of, e.g., CASE contains the whole utterance?

Solution: Richter 2007

Extension of signature:

top EMBEDDED *u_sign*
  sign ...
    e_sign
      e_word
      e_phrase
    u_sign
      u_word
      u_phrase
    word
      e_word
      u_word
    phrase H_DTR *e_sign*
        NH_DTR *e_sign*
      e_phrase
      u_phrase
...
`component/2`

Extension of theory:

- UNIQUE U-SIGN CONDITION:
  $\forall \boxed{1} \forall \boxed{2} ((\boxed{1} \sim u\_sign \wedge \boxed{2} \sim u\_sign) \rightarrow \boxed{1} \approx \boxed{2})$
- U-SIGN COMPONENT CONDITION:
  $\forall \boxed{1} (\boxed{1} \sim top \rightarrow \exists \boxed{2} (\boxed{2} \sim u\_sign \wedge \mathtt{component}(\boxed{1}, \boxed{2})))$

Solves the above problems:

- **no twin structures:** everything is a component of a single *u_sign*,
- **no stranded structures:** ditto (assuming *u_sign* → empty SLASH, etc.).

At a cost:

- **more complex signature** (and theory),
- **conceptual problem:** the value of, e.g., CASE contains the whole utterance?

Solution: Richter 2007

Extension of signature:

top EMBEDDED *u_sign*
  sign ...
    e_sign
      e_word
      e_phrase
    u_sign
      u_word
      u_phrase
    word
      e_word
      u_word
    phrase H_DTR *e_sign*
        NH_DTR *e_sign*
      e_phrase
      u_phrase

...

`component/2`

Extension of theory:

- UNIQUE U-SIGN CONDITION:
  $\forall \boxed{1} \forall \boxed{2} ((\boxed{1} \sim u\_sign \land \boxed{2} \sim u\_sign) \rightarrow \boxed{1} \approx \boxed{2})$
- U-SIGN COMPONENT CONDITION:
  $\forall \boxed{1} (\boxed{1} \sim top \rightarrow \exists \boxed{2} (\boxed{2} \sim u\_sign \land \mathtt{component}(\boxed{1}, \boxed{2})))$

Solves the above problems:

- **no twin structures**: everything is a component of a single *u_sign*,
- **no stranded structures**: ditto (assuming *u_sign* → empty SLASH, etc.).

At a cost:

- **more complex signature** (and theory),
- **conceptual problem**: the value of, e.g., CASE contains the whole utterance?

Solution: Richter 2007

Extension of signature:

top EMBEDDED *u_sign*
- *sign* ...
  - *e_sign*
    - *e_word*
    - *e_phrase*
  - *u_sign*
    - *u_word*
    - *u_phrase*
  - *word*
    - *e_word*
    - *u_word*
  - *phrase* H_DTR *e_sign*
    NH_DTR *e_sign*
    - *e_phrase*
    - *u_phrase*

...

`component/2`

Extension of theory:

- UNIQUE U-SIGN CONDITION:
  $\forall \boxed{1} \forall \boxed{2} ((\boxed{1} \sim u\_sign \wedge \boxed{2} \sim u\_sign) \rightarrow \boxed{1} \approx \boxed{2})$
- U-SIGN COMPONENT CONDITION:
  $\forall \boxed{1} (\boxed{1} \sim top \rightarrow \exists \boxed{2} (\boxed{2} \sim u\_sign \wedge \mathtt{component}(\boxed{1}, \boxed{2})))$

Solves the above problems:

- **no twin structures**: everything is a component of a single *u_sign*,
- **no stranded structures**: ditto (assuming *u_sign* → empty SLASH, etc.).

At a cost:

- **more complex signature** (and theory),
- **conceptual problem**: the value of, e.g., CASE contains the whole utterance?

Solution: Richter 2007

Extension of signature:

top EMBEDDED *u_sign*
- *sign* ...
  - *e_sign*
    - *e_word*
    - *e_phrase*
  - *u_sign*
    - *u_word*
    - *u_phrase*
  - *word*
    - *e_word*
    - *u_word*
  - *phrase* H_DTR *e_sign*
    NH_DTR *e_sign*
    - *e_phrase*
    - *u_phrase*

...

`component/2`

Extension of theory:

- UNIQUE U-SIGN CONDITION:
  $\forall \boxed{1} \forall \boxed{2} ((\boxed{1} \sim \mathit{u\_sign} \wedge \boxed{2} \sim \mathit{u\_sign}) \rightarrow \boxed{1} \approx \boxed{2})$
- U-SIGN COMPONENT CONDITION:
  $\forall \boxed{1} (\boxed{1} \sim \mathit{top} \rightarrow \exists \boxed{2} (\boxed{2} \sim \mathit{u\_sign} \wedge \mathtt{component}(\boxed{1}, \boxed{2})))$

Solves the above problems:

- **no twin structures:** everything is a component of a single *u_sign*,
- **no stranded structures:** ditto (assuming *u_sign* → empty SLASH, etc.).

At a cost:

- **more complex signature** (and theory),
- **conceptual problem:** the value of, e.g., CASE contains the whole utterance?

Solution: Richter 2007

Extension of signature:

top EMBEDDED *u_sign*
- *sign* …
  - *e_sign*
    - *e_word*
    - *e_phrase*
  - *u_sign*
    - *u_word*
    - *u_phrase*
  - *word*
    - *e_word*
    - *u_word*
  - *phrase* H_DTR *e_sign*
    NH_DTR *e_sign*
    - *e_phrase*
    - *u_phrase*

…

component/2

Extension of theory:

- UNIQUE U-SIGN CONDITION:
  $\forall \boxed{1} \forall \boxed{2} ((\boxed{1} \sim u\_sign \wedge \boxed{2} \sim u\_sign) \rightarrow \boxed{1} \approx \boxed{2})$
- U-SIGN COMPONENT CONDITION:
  $\forall \boxed{1} (\boxed{1} \sim top \rightarrow \exists \boxed{2} (\boxed{2} \sim u\_sign \wedge \mathtt{component}(\boxed{1}, \boxed{2})))$

Solves the above problems:

- **no twin structures:** everything is a component of a single *u_sign*,
- **no stranded structures:** ditto (assuming *u_sign* → empty SLASH, etc.).

At a cost:

- **more complex signature** (and theory),
- **conceptual problem:** the value of, e.g., CASE contains the whole utterance?

Problem: Linguistic tokens in models

King 1999 posits two **contradictory requirements about models**:

- that they be **exhaustive**,
- that model **objects are (components of) utterance tokens**.

Exhaustivity implies that models contain all licensed configurations.

But **not all configurations correspond to actual utterance tokens**.

So there must be **non-actual** utterance **tokens**.

"To Pollard [(p.c. to Richter)], **the concept of non-actual tokens is contradictory and nonsensical**" (Richter 2004: 119).

Problem: Linguistic tokens in models

King 1999 posits two **contradictory requirements about models**:

- that they be **exhaustive**,
- that model **objects are (components of) utterance tokens**.

Exhaustivity implies that models contain all licensed configurations.

But **not all configurations correspond to actual utterance tokens**.

So there must be **non-actual** utterance **tokens**.

"To Pollard [(p.c. to Richter)], **the concept of non-actual tokens is contradictory and nonsensical**" (Richter 2004: 119).

Problem: Linguistic tokens in models

King 1999 posits two **contradictory requirements about models**:

- that they be **exhaustive**,
- that model **objects are (components of) utterance tokens**.

Exhaustivity implies that models contain all licensed configurations.

But **not all configurations correspond to actual utterance tokens**.

So there must be **non-actual** utterance **tokens**.

"To Pollard [(p.c. to Richter)], **the concept of non-actual tokens is contradictory and nonsensical**" (Richter 2004: 119).

Problem: Linguistic tokens in models

King 1999 posits two **contradictory requirements about models**:

- that they be **exhaustive**,
- that model **objects are (components of) utterance tokens**.

Exhaustivity implies that models contain all licensed configurations.

But **not all configurations correspond to actual utterance tokens.**

So there must be **non-actual** utterance **tokens**.

"To Pollard [(p.c. to Richter)], **the concept of non-actual tokens is contradictory and nonsensical**" (Richter 2004: 119).

Problem: Linguistic tokens in models

King 1999 posits two **contradictory requirements about models**:

- that they be **exhaustive**,
- that model **objects are (components of) utterance tokens**.

Exhaustivity implies that models contain all licensed configurations.

But **not all configurations correspond to actual utterance tokens**.

So there must be **non-actual** utterance **tokens**.

"To Pollard [(p.c. to Richter)], **the concept of non-actual tokens is contradictory and nonsensical**" (Richter 2004: 119).

Problem: Linguistic tokens in models

King 1999 posits two **contradictory requirements about models**:

- that they be **exhaustive**,
- that model **objects are (components of) utterance tokens**.

Exhaustivity implies that models contain all licensed configurations.

But **not all configurations correspond to actual utterance tokens**.

So there must be **non-actual** utterance **tokens**.

"To Pollard [(p.c. to Richter)], **the concept of non-actual tokens is contradictory and nonsensical**" (Richter 2004: 119).

Problem: Linguistic tokens in models

King 1999 posits two **contradictory requirements about models**:

- that they be **exhaustive**,
- that model **objects are (components of) utterance tokens**.

Exhaustivity implies that models contain all licensed configurations.

But **not all configurations correspond to actual utterance tokens**.

So there must be **non-actual** utterance **tokens**.

"To Pollard [(p.c. to Richter)], **the concept of non-actual tokens is contradictory and nonsensical**" (Richter 2004: 119).

Solution: Rooted models of utterances

A **solution** to all of the above (see extended slides for RSRL technicalities):

- **rooted models**,
- **of single utterances**.

Solves all the problems above:

- **no twin structures**: everything is a component of a single utterance,
- **no stranded structures**: ditto (assuming we can say: utterance → empty SLASH, etc.),
- **no non-actual tokens**:
  - we may separate the questions (Pullum and Scholz 2001):
    - Is a given configuration a model of a grammar?
    - What is the exact collection of models of a grammar?
  - we don't have to enumerate all possible (actual and non-actual) tokens.

Solution: Rooted models of utterances

A **solution** to all of the above (see extended slides for RSRL technicalities):

- **rooted models**,
- **of single utterances**.

Solves all the problems above:

- **no twin structures**: everything is a component of a single utterance,
- **no stranded structures**: ditto (assuming we can say: utterance → empty SLASH, etc.),
- **no non-actual tokens**:
  - we may separate the questions (Pullum and Scholz 2001):
    - Is a given configuration a model of a grammar?
    - What is the exact collection of models of a grammar?
  - we don't have to enumerate all possible (actual and non-actual) tokens.

Solution: Rooted models of utterances

A **solution** to all of the above (see extended slides for RSRL technicalities):

- **rooted models**,
- **of single utterances**.

Solves all the problems above:

- **no twin structures**: everything is a component of a single utterance,
- **no stranded structures**: ditto (assuming we can say: utterance → empty SLASH, etc.),
- **no non-actual tokens**:
  - we may separate the questions (Pullum and Scholz 2001):
    - Is a given configuration a model of a grammar?
    - What is the exact collection of models of a grammar?
  - we don't have to enumerate all possible (actual and non-actual) tokens.

Solution: Rooted models of utterances

A **solution** to all of the above (see extended slides for RSRL technicalities):

- **rooted models**,
- **of single utterances**.

Solves all the problems above:

- **no twin structures**: everything is a component of a single utterance,
- **no stranded structures**: ditto (assuming we can say: utterance → empty SLASH, etc.),
- **no non-actual tokens**:
  - we may separate the questions (Pullum and Scholz 2001):
    - Is a given configuration a model of a grammar?
    - What is the exact collection of models of a grammar?
  - we don't have to enumerate all possible (actual and non-actual) tokens.

Solution: Rooted models of utterances

A **solution** to all of the above (see extended slides for RSRL technicalities):

- **rooted models**,
- **of single utterances**.

Solves all the problems above:

- **no twin structures**: everything is a component of a single utterance,
- **no stranded structures**: ditto (assuming we can say: utterance → empty SLASH, etc.),
- **no non-actual tokens**:
  - we may separate the questions (Pullum and Scholz 2001):
    - Is a given configuration a model of a grammar?
    - What is the exact collection of models of a grammar?
  - we don't have to enumerate all possible (actual and non-actual) tokens.

Solution: Rooted models of utterances

A **solution** to all of the above (see extended slides for RSRL technicalities):

- **rooted models**,
- **of single utterances**.

Solves all the problems above:

- **no twin structures**: everything is a component of a single utterance,
- **no stranded structures**: ditto (assuming we can say: utterance → empty SLASH, etc.),
- **no non-actual tokens**:
  - we may separate the questions (Pullum and Scholz 2001):
    - Is a given configuration a model of a grammar?
    - What is the exact collection of models of a grammar?
  - we don't have to enumerate all possible (actual and non-actual) tokens.

Solution: Rooted models of utterances

A **solution** to all of the above (see extended slides for RSRL technicalities):

- **rooted models**,
- **of single utterances**.

Solves all the problems above:

- **no twin structures**: everything is a component of a single utterance,
- **no stranded structures**: ditto (assuming we can say: utterance → empty SLASH, etc.),
- **no non-actual tokens**:
  - we may separate the questions (Pullum and Scholz 2001):
    - Is a given configuration a model of a grammar?
    - What is the exact collection of models of a grammar?
  - we don't have to enumerate all possible (actual and non-actual) tokens.

Solution: Rooted models of utterances

A **solution** to all of the above (see extended slides for RSRL technicalities):

- **rooted models**,
- **of single utterances**.

Solves all the problems above:

- **no twin structures**: everything is a component of a single utterance,
- **no stranded structures**: ditto (assuming we can say: utterance → empty SLASH, etc.),
- **no non-actual tokens**:
  - we may separate the questions (Pullum and Scholz 2001):
    - Is a given configuration a model of a grammar?
    - What is the exact collection of models of a grammar?
  - we don't have to enumerate all possible (actual and non-actual) tokens.

Solution: Rooted models of utterances

A **solution** to all of the above (see extended slides for RSRL technicalities):

- **rooted models**,
- **of single utterances**.

Solves all the problems above:

- **no twin structures**: everything is a component of a single utterance,
- **no stranded structures**: ditto (assuming we can say: utterance → empty SLASH, etc.),
- **no non-actual tokens**:
  - we may separate the questions (Pullum and Scholz 2001):
    - Is a given configuration a model of a grammar?
    - What is the exact collection of models of a grammar?
  - we don't have to enumerate all possible (actual and non-actual) tokens.

Solution: Technicalities 1

Assume the standard notion of **signature** (we ignore chains throughout):

Definition (signature)

Σ is a signature iff
Σ is a septuple $\langle S, \sqsubseteq, S_{max}, A, F, R, Ar \rangle$,
$\langle S, \sqsubseteq \rangle$ is a partial order,
$S_{max} = \{\sigma \in S \mid \text{for each } \sigma' \in S, \text{ if } \sigma' \sqsubseteq \sigma \text{ then } \sigma = \sigma'\}$,
A is a set,
F is a partial function from $S \times A$ to S,
for each $\sigma_1 \in S$, for each $\sigma_2 \in S$, for each $\phi \in A$,
    if $F(\sigma_1, \phi)$ is defined and $\sigma_2 \sqsubseteq \sigma_1$
    then $F(\sigma_2, \phi)$ is defined and $F(\sigma_2, \phi) \sqsubseteq F(\sigma_1, \phi)$,
R is a finite set, and
Ar is a total function from R to the positive integers.

Solution: Technicalities 2

We extend the notion of **terms** – $\mathring{r}$ will refer to the root of the utterance:

Definition (terms)

For each signature $\Sigma = \langle S, \sqsubseteq, S_{max}, A, F, R, Ar \rangle$, the set of terms T^{Σ} is the smallest set such that
$\mathring{r} \in T^{\Sigma}$,
$: \in T^{\Sigma}$,
for each $x \in V$, $x \in T^{\Sigma}$,
for each $\phi \in A$ and each $\tau \in T^{\Sigma}$, $\tau\phi \in T^{\Sigma}$.

Solution: Technicalities 2

We extend the notion of **terms** – $\mathring{r}$ will refer to the root of the utterance:

Definition (terms)

For each signature $\Sigma = \langle S, \sqsubseteq, S_{max}, A, F, R, Ar \rangle$, the set of terms T^{Σ} is the smallest set such that
$\mathring{r} \in T^{\Sigma}$,
$: \in T^{\Sigma}$,
for each $x \in V$, $x \in T^{\Sigma}$,
for each $\phi \in A$ and each $\tau \in T^{\Sigma}$, $\tau\phi \in T^{\Sigma}$.

Solution: Technicalities 3

Standard definition of **formulæ**:

Definition (formulæ)

For each signature $\Sigma = \langle S, \sqsubseteq, S_{max}, A, F, R, Ar \rangle$, the set of formulæ D^{Σ} is the smallest set such that

for each $\sigma \in S$, for each $\tau \in T^{\Sigma}$, $\tau \sim \sigma \in D^{\Sigma}$,

for each $\tau_1, \tau_2 \in T^{\Sigma}$, $\tau_1 \approx \tau_2 \in D^{\Sigma}$,

for each $\rho \in R$, for each $x_1, \ldots, x_{Ar(\rho)} \in V$, $\rho(x_1, \ldots, x_{Ar(\rho)}) \in D^{\Sigma}$,

for each $x \in V$, for each $\delta \in D^{\Sigma}$, $\exists x \delta \in D^{\Sigma}$, (analogous for $\forall$)

for each $\delta \in D^{\Sigma}$, $\neg\delta \in D^{\Sigma}$,

for each $\delta_1, \delta_2 \in D^{\Sigma}$, and $(\delta_1 \wedge \delta_2) \in D^{\Sigma}$. (analogous for $\vee, \rightarrow, \leftrightarrow$)

Solution: Technicalities 4

Trivial extension of the definition of **free variables** to handle the term $\mathring{r}$:

Definition (free variables)

For each signature $\Sigma = \langle S, \sqsubseteq, S_{max}, A, F, R, Ar \rangle$, FV is the function from terms and formulæ to free variables in them:

$FV(\mathring{r}) = \{\}$,

$FV(:) = \{\}$,

for each $x \in V$, $FV(x) = \{x\}$,

for each $\tau \in T^{\Sigma}$, for each $\phi \in A$, $FV(\tau\phi) = FV(\tau)$,

for each $\tau \in T^{\Sigma}$, for each $\sigma \in S$, $FV(\tau \sim \sigma) = FV(\tau)$,

for each $\tau_1, \tau_2 \in T^{\Sigma}$, $FV(\tau_1 \approx \tau_2) = FV(\tau_1) \cup FV(\tau_2)$,

for each $\rho \in R$, for each $x_1, \ldots, x_{Ar(\rho)} \in V$,

$FV(\rho(x_1, \ldots, x_{Ar(\rho)})) = \{x_1, \ldots, x_{Ar(\rho)}\}$,

for each $\delta \in D^{\Sigma}$, for each $x \in V$, $FV(\exists x \delta) = FV(\delta) \backslash \{x\}$, (also $\forall$)

for each $\delta \in D^{\Sigma}$, $FV(\neg\delta) = FV(\delta)$,

for each $\delta_1, \delta_2 \in D^{\Sigma}$, $FV((\delta_1 \wedge \delta_2)) = FV(\delta_1) \cup FV(\delta_2)$. (also $\vee, \rightarrow, \leftrightarrow$)

Solution: Technicalities 4

Trivial extension of the definition of **free variables** to handle the term $\mathring{r}$:

Definition (free variables)

For each signature $\Sigma = \langle S, \sqsubseteq, S_{max}, A, F, R, Ar \rangle$, FV is the function from terms and formulæ to free variables in them:

$FV(\mathring{r}) = \{\}$,
$FV(:) = \{\}$,
for each $x \in V$, $FV(x) = \{x\}$,
for each $\tau \in T^{\Sigma}$, for each $\phi \in A$, $FV(\tau\phi) = FV(\tau)$,
for each $\tau \in T^{\Sigma}$, for each $\sigma \in S$, $FV(\tau \sim \sigma) = FV(\tau)$,
for each $\tau_1, \tau_2 \in T^{\Sigma}$, $FV(\tau_1 \approx \tau_2) = FV(\tau_1) \cup FV(\tau_2)$,
for each $\rho \in R$, for each $x_1, \ldots, x_{Ar(\rho)} \in V$,
$FV(\rho(x_1, \ldots, x_{Ar(\rho)})) = \{x_1, \ldots, x_{Ar(\rho)}\}$,
for each $\delta \in D^{\Sigma}$, for each $x \in V$, $FV(\exists x \delta) = FV(\delta) \backslash \{x\}$, (also $\forall$)
for each $\delta \in D^{\Sigma}$, $FV(\neg\delta) = FV(\delta)$,
for each $\delta_1, \delta_2 \in D^{\Sigma}$, $FV((\delta_1 \wedge \delta_2)) = FV(\delta_1) \cup FV(\delta_2)$. (also $\vee, \rightarrow, \leftrightarrow$)

Solution: Technicalities 5

Standard definition of **descriptions**:

Definition (descriptions)

For each signature Σ, the set of descriptions $D_0^{\Sigma} = \{\delta \in D^{\Sigma} | FV(\delta) = \{\}\}$.

Solution: Technicalities 6

Definition of **interpretation** now singles out one element of the universe:

Definition (interpretation)

For each signature $\Sigma = \langle S, \sqsubseteq, S_{max}, A, F, R, Ar \rangle$, $\mathsf{I} = \langle \mathsf{U}, \mathsf{r}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$ is an Σ interpretation iff

U is a set,

$\mathsf{r} \in \mathsf{U}$,

S is a total function from U to S_{max},

A is a total function from A to the set of partial functions from U to U,

for each $\phi \in A$ and each $u \in \mathsf{U}$

if $\mathsf{A}(\phi)(u)$ is defined

then $F(\mathsf{S}(u), \phi)$ is defined, and $\mathsf{S}(\mathsf{A}(\phi)(u)) \sqsubseteq F(\mathsf{S}(u), \phi)$, and

for each $\phi \in A$ and each $u \in \mathsf{U}$,

if $F(\mathsf{S}(u), \phi)$ is defined then $\mathsf{A}(\phi)(u)$ is defined,

R is a total function from R to the power set of $\bigcup_{n \in \mathbb{N}} \mathsf{U}^n$, and

for each $\rho \in R$, $\mathsf{R}(\rho) \subseteq \mathsf{U}^{Ar(\rho)}$.

Solution: Technicalities 6

Definition of **interpretation** now singles out one element of the universe:

Definition (interpretation)

For each signature $\Sigma = \langle S, \sqsubseteq, S_{max}, A, F, R, Ar \rangle$, $\mathsf{I} = \langle \mathsf{U}, \mathsf{r}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$ is an Σ interpretation iff

U is a set,

$\mathsf{r} \in \mathsf{U}$,

S is a total function from U to S_{max},

A is a total function from A to the set of partial functions from U to U,

for each $\phi \in A$ and each $u \in \mathsf{U}$

if $\mathsf{A}(\phi)(u)$ is defined

then $F(\mathsf{S}(u), \phi)$ is defined, and $\mathsf{S}(\mathsf{A}(\phi)(u)) \sqsubseteq F(\mathsf{S}(u), \phi)$, and

for each $\phi \in A$ and each $u \in \mathsf{U}$,

if $F(\mathsf{S}(u), \phi)$ is defined then $\mathsf{A}(\phi)(u)$ is defined,

R is a total function from R to the power set of $\bigcup_{n \in \mathbb{N}} \mathsf{U}^n$, and

for each $\rho \in R$, $\mathsf{R}(\rho) \subseteq \mathsf{U}^{Ar(\rho)}$.

Solution: Technicalities 7

Standard definition of **variable assignments**:

Definition (variable assignments)

For each signature Σ, for each Σ interpretation $\mathsf{I} = \langle \mathsf{U}, \mathsf{r}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$,
$\mathsf{G}_{\mathsf{I}} = \mathsf{U}^{V}$ is the set of variable assignments in I.

Solution: Technicalities 8

Standard **term interpretation**, extended so that the interpretation of $\mathring{r}$ is r:

Definition (term interpretation)

For each signature $\Sigma = \langle S, \sqsubseteq, S_{max}, A, F, R, Ar\rangle$, for each Σ interpretation $\mathsf{I} = \langle \mathsf{U}, \mathsf{r}, \mathsf{S}, \mathsf{A}, \mathsf{R}\rangle$, for each $g \in \mathsf{G}_\mathsf{I}$, the term interpretation T^g_I is the total function from T^Σ to the set of partial functions from U to U such that for each $u \in \mathsf{U}$,

$\mathsf{T}^g_\mathsf{I}(\mathring{r})(u)$ is defined and $\mathsf{T}^g_\mathsf{I}(\mathring{r})(u) = \mathsf{r}$,

$\mathsf{T}^g_\mathsf{I}(:)(u)$ is defined and $\mathsf{T}^g_\mathsf{I}(:)(u) = u$,

for each $x \in V$, $\mathsf{T}^g_\mathsf{I}(x)(u)$ is defined and $\mathsf{T}^g_\mathsf{I}(x)(u) = g(x)$,

for each $\tau \in T^\Sigma$, for each $\phi \in A$,

$\mathsf{T}^g_\mathsf{I}(\tau\phi)(u)$ is defined iff $\mathsf{T}^g_\mathsf{I}(\tau)(u)$ is defined and $\mathsf{A}(\phi)(\mathsf{T}^g_\mathsf{I}(\tau)(u))$ is defined, and

if $\mathsf{T}^g_\mathsf{I}(\tau\phi)(u)$ is defined then $\mathsf{T}^g_\mathsf{I}(\tau\phi)(u) = \mathsf{A}(\phi)(\mathsf{T}^g_\mathsf{I}(\tau)(u))$.

Solution: Technicalities 8

Standard **term interpretation**, extended so that the interpretation of $\mathring{r}$ is r:

Definition (term interpretation)

For each signature $\Sigma = \langle S, \sqsubseteq, S_{max}, A, F, R, Ar \rangle$, for each Σ interpretation $\mathsf{I} = \langle \mathsf{U}, \mathsf{r}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$, for each $g \in \mathsf{G}_\mathsf{I}$, the term interpretation T^g_I is the total function from T^Σ to the set of partial functions from U to U such that for each $u \in \mathsf{U}$,

$\mathsf{T}^g_\mathsf{I}(\mathring{r})(u)$ is defined and $\mathsf{T}^g_\mathsf{I}(\mathring{r})(u) = \mathsf{r}$,

$\mathsf{T}^g_\mathsf{I}(:)(u)$ is defined and $\mathsf{T}^g_\mathsf{I}(:)(u) = u$,

for each $x \in V$, $\mathsf{T}^g_\mathsf{I}(x)(u)$ is defined and $\mathsf{T}^g_\mathsf{I}(x)(u) = g(x)$,

for each $\tau \in T^\Sigma$, for each $\phi \in A$,

$\mathsf{T}^g_\mathsf{I}(\tau\phi)(u)$ is defined iff $\mathsf{T}^g_\mathsf{I}(\tau)(u)$ is defined and $\mathsf{A}(\phi)(\mathsf{T}^g_\mathsf{I}(\tau)(u))$ is defined, and

if $\mathsf{T}^g_\mathsf{I}(\tau\phi)(u)$ is defined then $\mathsf{T}^g_\mathsf{I}(\tau\phi)(u) = \mathsf{A}(\phi)(\mathsf{T}^g_\mathsf{I}(\tau)(u))$.

Solution: Technicalities 9

Formula denotation – now quantification is over the whole universe (i.e., over all the components of the utterance, similarly to Richter 2007):

Definition (formula denotation)

For each signature $\Sigma = \langle S, \sqsubseteq, S_{max}, A, F, R, Ar\rangle$, for each Σ interpretation $\mathsf{I} = \langle \mathsf{U}, \mathsf{r}, \mathsf{S}, \mathsf{A}, \mathsf{R}\rangle$, for each $g \in \mathsf{G}_\mathsf{I}$, the formula denotation function D^g_I is the total function from D^Σ to the power set of U such that

for each $\tau \in T^\Sigma$, for each $\sigma \in S$,

$$\mathsf{D}^g_\mathsf{I}(\tau \sim \sigma) = \left\{ u \in \mathsf{U} \,\middle|\, \begin{array}{l} \mathsf{T}^g_\mathsf{I}(\tau)(u) \text{ is defined, and} \\ \mathsf{S}\left(\mathsf{T}^g_\mathsf{I}(\tau)(u)\right) \sqsubseteq \sigma \end{array} \right\},$$

for each $\tau_1, \tau_2 \in T^\Sigma$,

$$\mathsf{D}^g_\mathsf{I}(\tau_1 \approx \tau_2) = \left\{ u \in \mathsf{U} \,\middle|\, \begin{array}{l} \mathsf{T}^g_\mathsf{I}(\tau_1)(u) \text{ is defined,} \\ \mathsf{T}^g_\mathsf{I}(\tau_2)(u) \text{ is defined, and} \\ \mathsf{T}^g_\mathsf{I}(\tau_1)(u) = \mathsf{T}^g_\mathsf{I}(\tau_2)(u) \end{array} \right\},$$

for each $\rho \in R$, for each $x_1, \ldots, x_{Ar(\rho)} \in V$,

$$\mathsf{D}^g_\mathsf{I}\left(\rho(x_1, \ldots, x_{Ar(\rho)})\right) = \left\{ u \in \mathsf{U} \,\middle|\, \left\langle g(x_1), \ldots, g(x_{Ar(\rho)})\right\rangle \in \mathsf{R}(\rho) \right\},$$

...

Solution: Technicalities 10

Definition (formula denotation contd.)

. . . ,

for each $x \in V$, for each $\delta \in D^{\Sigma}$,

$$\mathsf{D}^{g}_{\mathsf{I}}(\exists x \delta) = \left\{ u \in \mathsf{U} \;\middle|\; \begin{array}{l} \text{for some } u' \in \mathsf{U} \\ u \in \mathsf{D}^{g[x \mapsto u']}_{\mathsf{I}}(\delta) \end{array} \right\},$$

for each $x \in V$, for each $\delta \in D^{\Sigma}$,

$$\mathsf{D}^{g}_{\mathsf{I}}(\forall x \delta) = \left\{ u \in \mathsf{U} \;\middle|\; \begin{array}{l} \text{for each } u' \in \mathsf{U} \\ u \in \mathsf{D}^{g[x \mapsto u']}_{\mathsf{I}}(\delta) \end{array} \right\},$$

for each $\delta \in D^{\Sigma}$, $\mathsf{D}^{g}_{\mathsf{I}}(\neg\delta) = \mathsf{U} \backslash \mathsf{D}^{g}_{\mathsf{I}}(\delta)$,

for each $\delta_1, \delta_2 \in D^{\Sigma}$, $\mathsf{D}^{g}_{\mathsf{I}}((\delta_1 \wedge \delta_2)) = \mathsf{D}^{g}_{\mathsf{I}}(\delta_1) \cap \mathsf{D}^{g}_{\mathsf{I}}(\delta_2)$

for each $\delta_1, \delta_2 \in D^{\Sigma}$, $\mathsf{D}^{g}_{\mathsf{I}}((\delta_1 \vee \delta_2)) = \mathsf{D}^{g}_{\mathsf{I}}(\delta_1) \cup \mathsf{D}^{g}_{\mathsf{I}}(\delta_2)$

for each $\delta_1, \delta_2 \in D^{\Sigma}$, $\mathsf{D}^{g}_{\mathsf{I}}((\delta_1 \rightarrow \delta_2)) = \big(\mathsf{U} \backslash \mathsf{D}^{g}_{\mathsf{I}}(\delta_1)\big) \cup \mathsf{D}^{g}_{\mathsf{I}}(\delta_2)$, and

for each $\delta_1, \delta_2 \in D^{\Sigma}$,

$$\mathsf{D}^{g}_{\mathsf{I}}((\delta_1 \leftrightarrow \delta_2)) = ((\mathsf{U} \backslash \mathsf{D}^{g}_{\mathsf{I}}(\delta_1)) \cap (\mathsf{U} \backslash \mathsf{D}^{g}_{\mathsf{I}}(\delta_2))) \cup (\mathsf{D}^{g}_{\mathsf{I}}(\delta_1) \cap \mathsf{D}^{g}_{\mathsf{I}}(\delta_2)).$$

Solution: Technicalities 10

Definition (formula denotation contd.)

…,

for each $x \in V$, for each $\delta \in D^{\Sigma}$,

$$\mathsf{D}_{\mathsf{I}}^{g}(\exists x \delta) = \left\{ u \in \mathsf{U} \,\middle|\, \begin{array}{l} \text{for some } u' \in \mathsf{U} \\ u \in \mathsf{D}_{\mathsf{I}}^{g[x \mapsto u']}(\delta) \end{array} \right\},$$

for each $x \in V$, for each $\delta \in D^{\Sigma}$,

$$\mathsf{D}_{\mathsf{I}}^{g}(\forall x \delta) = \left\{ u \in \mathsf{U} \,\middle|\, \begin{array}{l} \text{for each } u' \in \mathsf{U} \\ u \in \mathsf{D}_{\mathsf{I}}^{g[x \mapsto u']}(\delta) \end{array} \right\},$$

for each $\delta \in D^{\Sigma}$, $\mathsf{D}_{\mathsf{I}}^{g}(\neg\delta) = \mathsf{U} \backslash \mathsf{D}_{\mathsf{I}}^{g}(\delta)$,

for each $\delta_1, \delta_2 \in D^{\Sigma}$, $\mathsf{D}_{\mathsf{I}}^{g}((\delta_1 \wedge \delta_2)) = \mathsf{D}_{\mathsf{I}}^{g}(\delta_1) \cap \mathsf{D}_{\mathsf{I}}^{g}(\delta_2)$

for each $\delta_1, \delta_2 \in D^{\Sigma}$, $\mathsf{D}_{\mathsf{I}}^{g}((\delta_1 \vee \delta_2)) = \mathsf{D}_{\mathsf{I}}^{g}(\delta_1) \cup \mathsf{D}_{\mathsf{I}}^{g}(\delta_2)$

for each $\delta_1, \delta_2 \in D^{\Sigma}$, $\mathsf{D}_{\mathsf{I}}^{g}((\delta_1 \rightarrow \delta_2)) = \left(\mathsf{U} \backslash \mathsf{D}_{\mathsf{I}}^{g}(\delta_1)\right) \cup \mathsf{D}_{\mathsf{I}}^{g}(\delta_2)$, and

for each $\delta_1, \delta_2 \in D^{\Sigma}$,

$$\mathsf{D}_{\mathsf{I}}^{g}((\delta_1 \leftrightarrow \delta_2)) = ((\mathsf{U} \backslash \mathsf{D}_{\mathsf{I}}^{g}(\delta_1)) \cap (\mathsf{U} \backslash \mathsf{D}_{\mathsf{I}}^{g}(\delta_2))) \cup (\mathsf{D}_{\mathsf{I}}^{g}(\delta_1) \cap \mathsf{D}_{\mathsf{I}}^{g}(\delta_2)).$$

Solution: Technicalities 11

Standard definition of **description denotation**:

Definition (description denotation)

For each signature Σ, for each Σ interpretation $\mathsf{I} = \langle \mathsf{U}, \mathsf{r}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$, the description denotation function D_I is the total function from D_0^Σ to the power set of U such that $\mathsf{D}_\mathsf{I}(\delta) = \left\{ u \in \mathsf{U} \,\middle|\, \text{for each } g \in \mathsf{G}_\mathsf{I},\ u \in \mathsf{D}_\mathsf{I}^g(\delta) \right\}$.

Solution: Technicalities 12

So far we know that $\mathring{r}$ **points at a distinguished model element** r.

We want to make sure that r **is the root**...

- $\forall \boxed{1}$ component($\boxed{1}$, $\mathring{r}$) i.e.: $\exists \boxed{0}$ ($\boxed{0} \approx \mathring{r} \wedge \forall \boxed{1}$ component($\boxed{1}$, $\boxed{0}$))

... **of an utterance**:

- $\mathring{r}$ SYNSEM NONLOCAL SLASH $\sim$ *eset*
- $\mathring{r}$ SYNSEM LOCAL CAT VAL SUBJ $\sim$ *elist* $\wedge$
  $\mathring{r}$ SYNSEM LOCAL CAT VAL COMPS $\sim$ *elist*
- etc.

Solution: Technicalities 12

So far we know that $\mathring{r}$ **points at a distinguished model element** r.

We want to make sure that r **is the root**...

- $\forall\boxed{1}$ component$(\boxed{1}, \mathring{r})$ i.e.: $\exists\boxed{0}$ $(\boxed{0} \approx \mathring{r} \wedge \forall\boxed{1}$ component$(\boxed{1}, \boxed{0}))$

...**of an utterance**:

- $\mathring{r}$ SYNSEM NONLOCAL SLASH $\sim$ *eset*
- $\mathring{r}$ SYNSEM LOCAL CAT VAL SUBJ $\sim$ *elist* $\wedge$
  $\mathring{r}$ SYNSEM LOCAL CAT VAL COMPS $\sim$ *elist*
- etc.

Solution: Technicalities 12

So far we know that $\mathring{r}$ **points at a distinguished model element** r.

We want to make sure that r **is the root**…

- $\forall \boxed{1}$ $\texttt{component}(\boxed{1}, \mathring{r})$ i.e.: $\exists \boxed{0}\ (\boxed{0} \approx \mathring{r} \wedge \forall \boxed{1}\ \texttt{component}(\boxed{1}, \boxed{0}))$

…**of an utterance:**

- $\mathring{r}$ SYNSEM NONLOCAL SLASH $\sim$ *eset*
- $\mathring{r}$ SYNSEM LOCAL CAT VAL SUBJ $\sim$ *elist* $\wedge$
  $\mathring{r}$ SYNSEM LOCAL CAT VAL COMPS $\sim$ *elist*
- etc.

Solution: Technicalities 12

So far we know that $\mathring{r}$ **points at a distinguished model element** r.

We want to make sure that r **is the root**...

- $\forall \boxed{1}$ component$(\boxed{1}, \mathring{r})$ i.e.: $\exists \boxed{0}\ (\boxed{0} \approx \mathring{r} \wedge \forall \boxed{1}$ component$(\boxed{1}, \boxed{0}))$

...**of an utterance**:

- $\mathring{r}$ SYNSEM NONLOCAL SLASH $\sim$ *eset*
- $\mathring{r}$ SYNSEM LOCAL CAT VAL SUBJ $\sim$ *elist* $\wedge$
  $\mathring{r}$ SYNSEM LOCAL CAT VAL COMPS $\sim$ *elist*
- etc.

Solution: Comparisons and summary

Rooted models – not an original idea:

- discussed **in modal logic** (*rooted* or *point generated* models; Blackburn *et al.* 2010: 56),
- these are exactly **Pollard's (1999: §6) singly generated** models.

Here, we **additionally**:

- require that such rooted models correspond to **utterances**,
- allow terms to **refer to the root** directly.

This view is **ontologically agnostic**; model objects may be:

- **utterance tokens and their parts** (King 1999),
- **unique mathematical objects representing utterance types** (Pollard and Sag 1994, Pollard 1999).

Solution: Comparisons and summary

Rooted models – not an original idea:

- discussed **in modal logic** (*rooted* or *point generated* models; Blackburn *et al.* 2010: 56),
- these are exactly **Pollard's (1999: §6) singly generated** models.

Here, we **additionally**:

- require that such rooted models correspond to **utterances**,
- allow terms to **refer to the root** directly.

This view is **ontologically agnostic**; model objects may be:

- **utterance tokens and their parts** (King 1999),
- **unique mathematical objects representing utterance types** (Pollard and Sag 1994, Pollard 1999).

Solution: Comparisons and summary

Rooted models – not an original idea:

- discussed **in modal logic** (*rooted* or *point generated* models; Blackburn *et al.* 2010: 56),
- these are exactly **Pollard's (1999: §6) singly generated** models.

Here, we **additionally**:

- require that such rooted models correspond to **utterances**,
- allow terms to **refer to the root** directly.

This view is **ontologically agnostic**; model objects may be:

- **utterance tokens and their parts** (King 1999),
- **unique mathematical objects representing utterance types** (Pollard and Sag 1994, Pollard 1999).

Solution: Comparisons and summary

Rooted models – not an original idea:

- discussed **in modal logic** (*rooted* or *point generated* models; Blackburn *et al.* 2010: 56),
- these are exactly **Pollard's (1999: §6) singly generated** models.

Here, we **additionally**:

- require that such rooted models correspond to **utterances**,
- allow terms to **refer to the root** directly.

This view is **ontologically agnostic**; model objects may be:

- **utterance tokens and their parts** (King 1999),
- **unique mathematical objects representing utterance types** (Pollard and Sag 1994, Pollard 1999).

Solution: Comparisons and summary

Rooted models – not an original idea:

- discussed **in modal logic** (*rooted* or *point generated* models; Blackburn *et al.* 2010: 56),
- these are exactly **Pollard's (1999: §6) singly generated** models.

Here, we **additionally**:

- require that such rooted models correspond to **utterances**,
- allow terms to **refer to the root** directly.

This view is **ontologically agnostic**; model objects may be:

- **utterance tokens and their parts** (King 1999),
- **unique mathematical objects representing utterance types** (Pollard and Sag 1994, Pollard 1999).

Solution: Comparisons and summary

Rooted models – not an original idea:

- discussed **in modal logic** (*rooted* or *point generated* models; Blackburn *et al.* 2010: 56),
- these are exactly **Pollard's (1999: §6) singly generated** models.

Here, we **additionally**:

- require that such rooted models correspond to **utterances**,
- allow terms to **refer to the root** directly.

This view is **ontologically agnostic**; model objects may be:

- **utterance tokens and their parts** (King 1999),
- **unique mathematical objects representing utterance types** (Pollard and Sag 1994, Pollard 1999).

Solution: Comparisons and summary

Rooted models – not an original idea:

- discussed **in modal logic** (*rooted* or *point generated* models; Blackburn *et al.* 2010: 56),
- these are exactly **Pollard's (1999: §6) singly generated** models.

Here, we **additionally**:

- require that such rooted models correspond to **utterances**,
- allow terms to **refer to the root** directly.

This view is **ontologically agnostic**; model objects may be:

- **utterance tokens and their parts** (King 1999),
- **unique mathematical objects representing utterance types** (Pollard and Sag 1994, Pollard 1999).

Solution: Comparisons and summary

Rooted models – not an original idea:

- discussed **in modal logic** (*rooted* or *point generated* models; Blackburn *et al.* 2010: 56),
- these are exactly **Pollard's (1999: §6) singly generated** models.

Here, we **additionally**:

- require that such rooted models correspond to **utterances**,
- allow terms to **refer to the root** directly.

This view is **ontologically agnostic**; model objects may be:

- **utterance tokens and their parts** (King 1999),
- **unique mathematical objects representing utterance types** (Pollard and Sag 1994, Pollard 1999).

Solution: Comparisons and summary

Rooted models – not an original idea:

- discussed **in modal logic** (*rooted* or *point generated* models; Blackburn *et al.* 2010: 56),
- these are exactly **Pollard's (1999: §6) singly generated** models.

Here, we **additionally**:

- require that such rooted models correspond to **utterances**,
- allow terms to **refer to the root** directly.

This view is **ontologically agnostic**; model objects may be:

- **utterance tokens and their parts** (King 1999),
- **unique mathematical objects representing utterance types** (Pollard and Sag 1994, Pollard 1999).

Yatabe 2004 on unlike category coordination

Yatabe 2004 deals with **unlike category coordination**, e.g. (Bayer 1996):

- We emphasized [[Mr. Colson's many qualifications]$_{NP}$ and [that he had worked at the White House]$_{CP}$].
- We emphasized [[that Mr. Colson had worked at the White House]$_{CP}$ and [his many other qualifications]$_{NP}$].

His **lexical entry for *emphasized***:

- $$\begin{bmatrix} \text{PHON } \langle \textit{emphasized} \rangle \\ \text{...VALENCE} \begin{bmatrix} \text{SUBJ } \left\langle \left[\text{...HEAD } c\left(\begin{bmatrix} \textit{noun} \\ \text{CASE } \textit{nom} \end{bmatrix} \right) \right] \right\rangle \\ \text{COMPS } \langle [\text{...HEAD } c(\textit{noun} \vee \textit{comp})] \rangle \end{bmatrix} \end{bmatrix}$$

The key idea is the use of the **distributive functor,** c:

- $$\boxed{1} : c(\alpha) \equiv \boxed{1} : \alpha \vee (\boxed{1} : [\text{ARGS } \langle \boxed{a_1}, \ldots, \boxed{a_n} \rangle] \wedge \boxed{a_1} : \alpha \wedge \ldots \wedge \boxed{a_n} : \alpha)$$

The value of ARGS is the list of (heads of) conjuncts in coordination.

Yatabe 2004 on unlike category coordination

Yatabe 2004 deals with **unlike category coordination**, e.g. (Bayer 1996):

- We emphasized [[Mr. Colson's many qualifications]$_{NP}$ and [that he had worked at the White House]$_{CP}$].
- We emphasized [[that Mr. Colson had worked at the White House]$_{CP}$ and [his many other qualifications]$_{NP}$].

His **lexical entry for *emphasized***:

- $$\begin{bmatrix} \text{PHON} & \langle \textit{emphasized} \rangle \\ \ldots\text{VALENCE} & \begin{bmatrix} \text{SUBJ} & \left\langle \begin{bmatrix} \ldots\text{HEAD} & c\left(\begin{bmatrix} \textit{noun} & \\ \text{CASE} & \textit{nom} \end{bmatrix} \right) \end{bmatrix} \right\rangle \\ \text{COMPS} & \left\langle \begin{bmatrix} \ldots\text{HEAD} & c(\textit{noun} \vee \textit{comp}) \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

The key idea is the use of the **distributive functor,** c:

- $$\boxed{1} : c(\alpha) \equiv \boxed{1} : \alpha \vee \left(\boxed{1} : \left[\text{ARGS} \left\langle \boxed{a_1}, \ldots, \boxed{a_n} \right\rangle\right] \wedge \boxed{a_1} : \alpha \wedge \ldots \wedge \boxed{a_n} : \alpha\right)$$

The value of ARGS is the list of (heads of) conjuncts in coordination.

Yatabe 2004 on unlike category coordination

Yatabe 2004 deals with **unlike category coordination**, e.g. (Bayer 1996):

- We emphasized [[Mr. Colson's many qualifications]$_{NP}$ and [that he had worked at the White House]$_{CP}$].
- We emphasized [[that Mr. Colson had worked at the White House]$_{CP}$ and [his many other qualifications]$_{NP}$].

His **lexical entry for *emphasized***:

- $$\begin{bmatrix} \text{PHON} & \langle \textit{emphasized} \rangle \\ \ldots\text{VALENCE} & \begin{bmatrix} \text{SUBJ} & \left\langle \left[\ldots\text{HEAD} \; c\left(\begin{bmatrix} \textit{noun} & \\ \text{CASE} & \textit{nom} \end{bmatrix} \right) \right] \right\rangle \\ \text{COMPS} & \langle [\ldots\text{HEAD} \; c(\textit{noun} \vee \textit{comp})] \rangle \end{bmatrix} \end{bmatrix}$$

The key idea is the use of the **distributive functor,** c:

- $$\boxed{1} : c(\alpha) \equiv \boxed{1} : \alpha \vee \left(\boxed{1} : \left[\text{ARGS} \; \langle \boxed{a_1}, \ldots, \boxed{a_n} \rangle \right] \wedge \boxed{a_1} : \alpha \wedge \ldots \wedge \boxed{a_n} : \alpha \right)$$

The value of ARGS is the list of (heads of) conjuncts in coordination.

Yatabe 2004 on unlike category coordination

Yatabe 2004 deals with **unlike category coordination**, e.g. (Bayer 1996):

- We emphasized [[Mr. Colson's many qualifications]$_{NP}$ and [that he had worked at the White House]$_{CP}$].
- We emphasized [[that Mr. Colson had worked at the White House]$_{CP}$ and [his many other qualifications]$_{NP}$].

His **lexical entry for *emphasized***:

- $$\begin{bmatrix} \text{PHON} & \langle \textit{emphasized} \rangle \\ \ldots\text{VALENCE} & \begin{bmatrix} \text{SUBJ} & \left\langle \begin{bmatrix} \ldots\text{HEAD} & c\left(\begin{bmatrix} \textit{noun} \\ \text{CASE} & \textit{nom} \end{bmatrix}\right) \end{bmatrix} \right\rangle \\ \text{COMPS} & \left\langle \begin{bmatrix} \ldots\text{HEAD} & c(\textit{noun} \vee \textit{comp}) \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

The key idea is the use of the **distributive functor,** c:

- $$\boxed{1} : c(\alpha) \quad \equiv \quad \boxed{1} : \alpha \quad \vee \quad \left(\boxed{1} : \left[\text{ARGS} \ \langle \boxed{a_1}, \ldots, \boxed{a_n} \rangle\right] \ \wedge \ \boxed{a_1} : \alpha \wedge \ldots \wedge \boxed{a_n} : \alpha\right)$$

The value of ARGS is the list of (heads of) conjuncts in coordination.

Unlike category coordination in HPSG

Unlike category coordination (UCC) in HPSG:

- **linearisation-based approaches** to UCC were **popular in HPSG in 2000s** (e.g., Crysmann 2003, Beavers and Sag 2004, Chaves 2006, 2007, 2008),
- many cases of UCC **not amenable to such elliptical approaches** (Levine 2011, Dalrymple 2017, Patejuk and Przepiórkowski 2021),
- **conceded in Chaves 2013 and Abeillé and Chaves 2021**,
- so **no HPSG alternative** to Yatabe 2004.

Unlike category coordination in HPSG

Unlike category coordination (UCC) in HPSG:

- **linearisation-based approaches** to UCC were **popular in HPSG in 2000s** (e.g., Crysmann 2003, Beavers and Sag 2004, Chaves 2006, 2007, 2008),
- many cases of **UCC not amenable to such elliptical approaches** (Levine 2011, Dalrymple 2017, Patejuk and Przepiórkowski 2021),
- **conceded** in Chaves 2013 and Abeillé and Chaves 2021,
- so **no HPSG alternative** to Yatabe 2004.

Unlike category coordination in HPSG

Unlike category coordination (UCC) in HPSG:

- **linearisation-based approaches** to UCC were **popular in HPSG in 2000s** (e.g., Crysmann 2003, Beavers and Sag 2004, Chaves 2006, 2007, 2008),
- many cases of **UCC not amenable to such elliptical approaches** (Levine 2011, Dalrymple 2017, Patejuk and Przepiórkowski 2021),
- **conceded in Chaves 2013 and Abeillé and Chaves 2021**,
- so **no HPSG alternative** to Yatabe 2004.

Unlike category coordination in HPSG

Unlike category coordination (UCC) in HPSG:

- **linearisation-based approaches** to UCC were **popular in HPSG in 2000s** (e.g., Crysmann 2003, Beavers and Sag 2004, Chaves 2006, 2007, 2008),
- many cases of **UCC not amenable to such elliptical approaches** (Levine 2011, Dalrymple 2017, Patejuk and Przepiórkowski 2021),
- **conceded in Chaves 2013 and Abeillé and Chaves 2021**,
- so **no HPSG alternative** to Yatabe 2004.

Reformulation in second-order RSRL

Yatabe 2004:

- $\left[\begin{array}{ll}\text{PHON} & \langle \textit{emphasized} \rangle \\ \ldots\text{VALENCE} & \left[\begin{array}{ll}\text{SUBJ} & \left\langle \left[\ldots\text{HEAD}\ c\left(\left[\begin{array}{l}\textit{noun} \\ \text{CASE}\ \ \textit{nom}\end{array}\right]\right)\right]\right\rangle \\ \text{COMPS} & \langle [\ldots\text{HEAD}\ \ c(\textit{noun} \vee \textit{comp})] \rangle\end{array}\right]\end{array}\right]$
- $\boxed{1} : c(\alpha) \quad \equiv \quad \boxed{1} : \alpha \ \ \vee$
  $(\boxed{1} : [\text{ARGS}\ \langle \boxed{a_1}, \ldots, \boxed{a_n} \rangle] \ \wedge \ \boxed{a_1} : \alpha \wedge \ldots \wedge \boxed{a_n} : \alpha)$

Second-order RSRL:

- $\left[\begin{array}{ll}\text{PHON} & \langle \textit{emphasized} \rangle \\ \ldots\text{VALENCE} & \left[\begin{array}{ll}\text{SUBJ} & \langle [\ldots\text{HEAD}\ \boxed{1}] \rangle \\ \text{COMPS} & \langle [\ldots\text{HEAD}\ \boxed{2}] \rangle\end{array}\right]\end{array}\right] \begin{array}{l}\wedge\ \alpha_1 \approx (:\sim \textit{noun} \wedge :\text{CASE} \sim \textit{nom}) \\ \wedge\ \alpha_2 \approx (:\sim \textit{noun} \vee :\sim \textit{comp}) \\ \wedge\ \mathsf{c}(\boxed{1}, \alpha_1)\ \wedge\ \mathsf{c}(\boxed{2}, \alpha_2)\end{array}$
- $\forall \boxed{1}_e \forall \alpha_{et}\ (\mathsf{c}(\boxed{1}, \alpha) \leftrightarrow \alpha(\boxed{1}) \quad \vee$
  $\exists \boxed{a_1} \ldots \exists \boxed{a_n}\ (\boxed{1}[\text{ARGS}\ \langle \boxed{a_1}, \ldots, \boxed{a_n} \rangle] \wedge \alpha(\boxed{a_1}) \wedge \ldots \wedge \alpha(\boxed{a_n})))$

(See extended slides for some technicalities.)

Reformulation in second-order RSRL

Yatabe 2004:

- $\begin{bmatrix} \text{PHON} & \langle \textit{emphasized} \rangle \\ \ldots\text{VALENCE} & \begin{bmatrix} \text{SUBJ} & \left\langle \begin{bmatrix} \ldots\text{HEAD} & c\left(\begin{bmatrix} \textit{noun} \\ \text{CASE} & \textit{nom} \end{bmatrix}\right) \end{bmatrix} \right\rangle \\ \text{COMPS} & \left\langle \begin{bmatrix} \ldots\text{HEAD} & c(\textit{noun} \vee \textit{comp}) \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$
- $\boxed{1} : c(\alpha) \quad \equiv \quad \boxed{1} : \alpha \quad \vee$
  $(\boxed{1} : \begin{bmatrix} \text{ARGS} & \langle \boxed{a_1}, \ldots, \boxed{a_n} \rangle \end{bmatrix} \;\wedge\; \boxed{a_1} : \alpha \wedge \ldots \wedge \boxed{a_n} : \alpha)$

Second-order RSRL:

- $\begin{bmatrix} \text{PHON} & \langle \textit{emphasized} \rangle \\ \ldots\text{VALENCE} & \begin{bmatrix} \text{SUBJ} & \langle [\ldots\text{HEAD} \; \boxed{1}] \rangle \\ \text{COMPS} & \langle [\ldots\text{HEAD} \; \boxed{2}] \rangle \end{bmatrix} \end{bmatrix} \begin{array}{l} \wedge\; \alpha_1 \approx (: \sim \textit{noun} \wedge : \text{CASE} \sim \textit{nom}) \\ \wedge\; \alpha_2 \approx (: \sim \textit{noun} \vee : \sim \textit{comp}) \\ \wedge\; \mathsf{c}(\boxed{1}, \alpha_1) \;\wedge\; \mathsf{c}(\boxed{2}, \alpha_2) \end{array}$
- $\forall \boxed{1}_e \forall \alpha_{et} \; (\mathsf{c}(\boxed{1}, \alpha) \leftrightarrow \alpha(\boxed{1}) \quad \vee$
  $\exists \boxed{a_1} \ldots \exists \boxed{a_n} \; (\boxed{1}\begin{bmatrix} \text{ARGS} & \langle \boxed{a_1}, \ldots, \boxed{a_n} \rangle \end{bmatrix} \wedge \alpha(\boxed{a_1}) \wedge \ldots \wedge \alpha(\boxed{a_n})))$

(See extended slides for some technicalities.)

Reformulation in second-order RSRL

Yatabe 2004:

- $\begin{bmatrix} \text{PHON} & \langle \textit{emphasized} \rangle \\ \ldots\text{VALENCE} & \begin{bmatrix} \text{SUBJ} & \left\langle \begin{bmatrix} \ldots\text{HEAD} & c\left(\begin{bmatrix} \textit{noun} \\ \text{CASE} & \textit{nom} \end{bmatrix}\right) \end{bmatrix} \right\rangle \\ \text{COMPS} & \left\langle \begin{bmatrix} \ldots\text{HEAD} & c(\textit{noun} \vee \textit{comp}) \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$
- $\boxed{1} : c(\alpha) \quad \equiv \quad \boxed{1} : \alpha \quad \vee$
  $(\boxed{1} : \left[\text{ARGS} \ \langle \boxed{a_1}, \ldots, \boxed{a_n} \rangle\right] \ \wedge \ \boxed{a_1} : \alpha \wedge \ldots \wedge \boxed{a_n} : \alpha)$

Second-order RSRL:

- $\begin{bmatrix} \text{PHON} & \langle \textit{emphasized} \rangle \\ \ldots\text{VALENCE} & \begin{bmatrix} \text{SUBJ} & \langle [\ldots\text{HEAD} \ \boxed{1}] \rangle \\ \text{COMPS} & \langle [\ldots\text{HEAD} \ \boxed{2}] \rangle \end{bmatrix} \end{bmatrix} \begin{array}{l} \wedge \ \alpha_1 \approx (: \sim \textit{noun} \wedge :\text{CASE} \sim \textit{nom}) \\ \wedge \ \alpha_2 \approx (: \sim \textit{noun} \vee : \sim \textit{comp}) \\ \wedge \ \mathsf{c}(\boxed{1}, \alpha_1) \ \wedge \ \mathsf{c}(\boxed{2}, \alpha_2) \end{array}$
- $\forall \boxed{1}_e \forall \alpha_{et} \ (\mathsf{c}(\boxed{1}, \alpha) \leftrightarrow \alpha(\boxed{1}) \quad \vee$
  $\exists \boxed{a_1} \ldots \exists \boxed{a_n} \ (\boxed{1}\left[\text{ARGS} \ \langle \boxed{a_1}, \ldots, \boxed{a_n} \rangle\right] \wedge \alpha(\boxed{a_1}) \wedge \ldots \wedge \alpha(\boxed{a_n})))$

(See extended slides for some technicalities.)

Second-order RSRL? 1

What would it require?

- signatures **specify types of arguments of relation symbols** (each either e or et);
- **interpretations of relation symbols**:
  - sets of tuples (as before), but
  - **each element of a tuple is an object or a set of objects**,
  - depending on the type specified in the signature;
- the **set of variables**, $\mathcal{VAR}$, is the disjoint sum of:
  - $\mathcal{VAR}_e$ (first-order variables) and
  - $\mathcal{VAR}_{et}$ (second-order variables);
- **variable assignments** assign
  - objects to elements of $\mathcal{VAR}_e$ and
  - sets of objects to elements of $\mathcal{VAR}_{et}$;
- the **interpretation of quantifiers** is extended to second-order variables correspondingly;
- ...

Second-order RSRL? 1

What would it require?

- signatures **specify types of arguments of relation symbols** (each either *e* or *et*);
- **interpretations of relation symbols**:
  - sets of tuples (as before), but
  - **each element of a tuple is an object or a set of objects,**
  - depending on the type specified in the signature;
- the **set of variables**, $\mathcal{VAR}$, is the disjoint sum of:
  - $\mathcal{VAR}_e$ (first-order variables) and
  - $\mathcal{VAR}_{et}$ (second-order variables);
- **variable assignments** assign
  - objects to elements of $\mathcal{VAR}_e$ and
  - sets of objects to elements of $\mathcal{VAR}_{et}$;
- the **interpretation of quantifiers** is extended to second-order variables correspondingly;
- ...

Second-order RSRL? 1

What would it require?

- signatures **specify types of arguments of relation symbols** (each either *e* or *et*);
- **interpretations of relation symbols**:
  - sets of tuples (as before), but
  - **each element of a tuple is an object or a set of objects**,
  - depending on the type specified in the signature;
- the **set of variables**, $\mathcal{VAR}$, is the disjoint sum of:
  - $\mathcal{VAR}_e$ (first-order variables) and
  - $\mathcal{VAR}_{et}$ (second-order variables);
- **variable assignments** assign
  - objects to elements of $\mathcal{VAR}_e$ and
  - sets of objects to elements of $\mathcal{VAR}_{et}$;
- the **interpretation of quantifiers** is extended to second-order variables correspondingly;
- ...

Second-order RSRL? 1

What would it require?

- signatures **specify types of arguments of relation symbols** (each either e or et);
- **interpretations of relation symbols**:
  - sets of tuples (as before), but
  - **each element of a tuple is an object or a set of objects**,
  - depending on the type specified in the signature;
- the **set of variables**, $\mathcal{VAR}$, is the disjoint sum of:
  - $\mathcal{VAR}_e$ (first-order variables) and
  - $\mathcal{VAR}_{et}$ (second-order variables);
- **variable assignments** assign
  - objects to elements of $\mathcal{VAR}_e$ and
  - sets of objects to elements of $\mathcal{VAR}_{et}$;
- the **interpretation of quantifiers** is extended to second-order variables correspondingly;
- ...

Second-order RSRL? 1

What would it require?

- signatures **specify types of arguments of relation symbols** (each either *e* or *et*);
- **interpretations of relation symbols**:
  - sets of tuples (as before), but
  - **each element of a tuple is an object or a set of objects**,
  - depending on the type specified in the signature;
- the **set of variables**, $\mathcal{VAR}$, is the disjoint sum of:
  - $\mathcal{VAR}_e$ (first-order variables) and
  - $\mathcal{VAR}_{et}$ (second-order variables);
- **variable assignments** assign
  - objects to elements of $\mathcal{VAR}_e$ and
  - sets of objects to elements of $\mathcal{VAR}_{et}$;
- the **interpretation of quantifiers** is extended to second-order variables correspondingly;
- ...

Second-order RSRL? 1

What would it require?

- signatures **specify types of arguments of relation symbols** (each either *e* or *et*);
- **interpretations of relation symbols**:
  - sets of tuples (as before), but
  - **each element of a tuple is an object or a set of objects**,
  - depending on the type specified in the signature;
- the **set of variables**, $\mathcal{VAR}$, is the disjoint sum of:
  - $\mathcal{VAR}_e$ (first-order variables) and
  - $\mathcal{VAR}_{et}$ (second-order variables);
- **variable assignments** assign
  - objects to elements of $\mathcal{VAR}_e$ and
  - sets of objects to elements of $\mathcal{VAR}_{et}$;
- the **interpretation of quantifiers** is extended to second-order variables correspondingly;
- ...

Second-order RSRL? 1

What would it require?

- signatures **specify types of arguments of relation symbols** (each either *e* or *et*);
- **interpretations of relation symbols**:
  - sets of tuples (as before), but
  - **each element of a tuple is an object or a set of objects**,
  - depending on the type specified in the signature;
- the **set of variables**, $\mathcal{VAR}$, is the disjoint sum of:
  - $\mathcal{VAR}_e$ (first-order variables) and
  - $\mathcal{VAR}_{et}$ (second-order variables);
- **variable assignments** assign
  - objects to elements of $\mathcal{VAR}_e$ and
  - sets of objects to elements of $\mathcal{VAR}_{et}$;
- the **interpretation of quantifiers** is extended to second-order variables correspondingly;
- ...

Second-order RSRL? 2

- . . . ;
- apart from the usual first-order terms $\mathcal{T}_e^\Sigma$ (for the signature Σ), there are also **second-order terms**, $\mathcal{T}_{et}^\Sigma$, specified recursively simultaneously with the set of formulae, $\mathcal{D}^\Sigma$, as the disjoint sum of **second-order variables** ($\mathcal{VAR}_{et}$) **and all formulae** ($\mathcal{D}^\Sigma$);
- two clauses of the **definition of formulae modified** so that:
  - **arguments of relation symbols** (variables) are **of the right type** *e* or *et*,
  - $\tau_1 \approx \tau_2$ is a formula if both **terms are of the same type** (i.e., both are *e* or both are *et*);
- importantly, **a new kind of formula** is added: $\tau_1(\tau_2)$, where $\tau_1 \in \mathcal{T}_{et}^\Sigma$ and $\tau_2 \in \mathcal{T}_e^\Sigma$;
- the **interpretation** of $\tau_1(\tau_2)$ is the set of all these objects of the universe U on which the interpretation of τ_2 belongs to the interpretation of τ_1; more formally: $D_{\mathrm{I}}^{ass}(\tau_1(\tau_2)) = \{u \in \mathrm{U} : T_{\mathrm{I}}^{ass}(\tau_2)(u) \in D_{\mathrm{I}}^{ass}(\tau_1)\}$.

Second-order RSRL? 2

- . . . ;
- apart from the usual first-order terms $\mathcal{T}_e^\Sigma$ (for the signature Σ), there are also **second-order terms**, $\mathcal{T}_{et}^\Sigma$, specified recursively simultaneously with the set of formulae, $\mathcal{D}^\Sigma$, as the disjoint sum of **second-order variables** ($\mathcal{VAR}_{et}$) **and all formulae** ($\mathcal{D}^\Sigma$);
- two clauses of the **definition of formulae modified** so that:
  - **arguments of relation symbols** (variables) are **of the right type** e or et,
  - $\tau_1 \approx \tau_2$ is a formula if both **terms are of the same type** (i.e., both are e or both are et);
- importantly, **a new kind of formula** is added: $\tau_1(\tau_2)$, where $\tau_1 \in \mathcal{T}_{et}^\Sigma$ and $\tau_2 \in \mathcal{T}_e^\Sigma$;
- the **interpretation** of $\tau_1(\tau_2)$ is the set of all these objects of the universe U on which the interpretation of τ_2 belongs to the interpretation of τ_1; more formally: $D_I^{ass}(\tau_1(\tau_2)) = \{u \in U : T_I^{ass}(\tau_2)(u) \in D_I^{ass}(\tau_1)\}$.

Second-order RSRL? 2

- ...;
- apart from the usual first-order terms $\mathcal{T}_e^{\Sigma}$ (for the signature Σ), there are also **second-order terms**, $\mathcal{T}_{et}^{\Sigma}$, specified recursively simultaneously with the set of formulae, $\mathcal{D}^{\Sigma}$, as the disjoint sum of **second-order variables** ($\mathcal{VAR}_{et}$) **and all formulae** ($\mathcal{D}^{\Sigma}$);
- two clauses of the **definition of formulae modified** so that:
  - **arguments of relation symbols** (variables) are **of the right type** *e* or *et*,
  - $\tau_1 \approx \tau_2$ is a formula if both **terms are of the same type** (i.e., both are *e* or both are *et*);
- importantly, **a new kind of formula** is added: $\tau_1(\tau_2)$, where $\tau_1 \in \mathcal{T}_{et}^{\Sigma}$ and $\tau_2 \in \mathcal{T}_e^{\Sigma}$;
- the **interpretation** of $\tau_1(\tau_2)$ is the set of all these objects of the universe U on which the interpretation of τ_2 belongs to the interpretation of τ_1; more formally: $D_{\mathrm{I}}^{ass}(\tau_1(\tau_2)) = \{u \in \mathrm{U} : T_{\mathrm{I}}^{ass}(\tau_2)(u) \in D_{\mathrm{I}}^{ass}(\tau_1)\}$.

Second-order RSRL? 2

- . . . ;
- apart from the usual first-order terms $\mathcal{T}_e^\Sigma$ (for the signature Σ), there are also **second-order terms**, $\mathcal{T}_{et}^\Sigma$, specified recursively simultaneously with the set of formulae, $\mathcal{D}^\Sigma$, as the disjoint sum of **second-order variables** ($\mathcal{VAR}_{et}$) **and all formulae** ($\mathcal{D}^\Sigma$);
- two clauses of the **definition of formulae modified** so that:
  - **arguments of relation symbols** (variables) are **of the right type** *e* or *et*,
  - $\tau_1 \approx \tau_2$ is a formula if both **terms are of the same type** (i.e., both are *e* or both are *et*);
- importantly, **a new kind of formula** is added: $\tau_1(\tau_2)$, where $\tau_1 \in \mathcal{T}_{et}^\Sigma$ and $\tau_2 \in \mathcal{T}_e^\Sigma$;
- the **interpretation** of $\tau_1(\tau_2)$ is the set of all these objects of the universe U on which the interpretation of τ_2 belongs to the interpretation of τ_1; more formally: $D_I^{ass}(\tau_1(\tau_2)) = \{u \in U : T_I^{ass}(\tau_2)(u) \in D_I^{ass}(\tau_1)\}$.

Second-order RSRL? 2

- . . . ;
- apart from the usual first-order terms $\mathcal{T}_e^\Sigma$ (for the signature Σ), there are also **second-order terms**, $\mathcal{T}_{et}^\Sigma$, specified recursively simultaneously with the set of formulae, $\mathcal{D}^\Sigma$, as the disjoint sum of **second-order variables** ($\mathcal{VAR}_{et}$) **and all formulae** ($\mathcal{D}^\Sigma$);
- two clauses of the **definition of formulae modified** so that:
  - **arguments of relation symbols** (variables) are **of the right type** *e* or *et*,
  - $\tau_1 \approx \tau_2$ is a formula if both **terms are of the same type** (i.e., both are *e* or both are *et*);
- importantly, **a new kind of formula** is added: $\tau_1(\tau_2)$, where $\tau_1 \in \mathcal{T}_{et}^\Sigma$ and $\tau_2 \in \mathcal{T}_e^\Sigma$;
- the **interpretation** of $\tau_1(\tau_2)$ is the set of all these objects of the universe U on which the interpretation of τ_2 belongs to the interpretation of τ_1; more formally: $D_I^{ass}(\tau_1(\tau_2)) = \{u \in U : T_I^{ass}(\tau_2)(u) \in D_I^{ass}(\tau_1)\}$.

Second-order RSRL? 2

- ...;
- apart from the usual first-order terms $\mathcal{T}_e^{\Sigma}$ (for the signature Σ), there are also **second-order terms**, $\mathcal{T}_{et}^{\Sigma}$, specified recursively simultaneously with the set of formulae, $\mathcal{D}^{\Sigma}$, as the disjoint sum of **second-order variables** ($\mathcal{VAR}_{et}$) **and all formulae** ($\mathcal{D}^{\Sigma}$);
- two clauses of the **definition of formulae modified** so that:
    - **arguments of relation symbols** (variables) are **of the right type** *e* or *et*,
    - $\tau_1 \approx \tau_2$ is a formula if both **terms are of the same type** (i.e., both are *e* or both are *et*);
- importantly, **a new kind of formula** is added: $\tau_1(\tau_2)$, where $\tau_1 \in \mathcal{T}_{et}^{\Sigma}$ and $\tau_2 \in \mathcal{T}_e^{\Sigma}$;
- the **interpretation** of $\tau_1(\tau_2)$ is the set of all these objects of the universe U on which the interpretation of τ_2 belongs to the interpretation of τ_1; more formally: $D_I^{ass}(\tau_1(\tau_2)) = \{u \in U : T_I^{ass}(\tau_2)(u) \in D_I^{ass}(\tau_1)\}$.

Second-order RSRL? 2

- . . . ;
- apart from the usual first-order terms $\mathcal{T}_e^\Sigma$ (for the signature Σ), there are also **second-order terms**, $\mathcal{T}_{et}^\Sigma$, specified recursively simultaneously with the set of formulae, $\mathcal{D}^\Sigma$, as the disjoint sum of **second-order variables** ($\mathcal{VAR}_{et}$) **and all formulae** ($\mathcal{D}^\Sigma$);
- two clauses of the **definition of formulae modified** so that:
  - **arguments of relation symbols** (variables) are **of the right type** *e* or *et*,
  - $\tau_1 \approx \tau_2$ is a formula if both **terms are of the same type** (i.e., both are *e* or both are *et*);
- importantly, **a new kind of formula** is added: $\tau_1(\tau_2)$, where $\tau_1 \in \mathcal{T}_{et}^\Sigma$ and $\tau_2 \in \mathcal{T}_e^\Sigma$;
- the **interpretation** of $\tau_1(\tau_2)$ is the set of all these objects of the universe U on which the interpretation of τ_2 belongs to the interpretation of τ_1; more formally: $D_\mathsf{I}^{ass}(\tau_1(\tau_2)) = \{u \in \mathsf{U} : T_\mathsf{I}^{ass}(\tau_2)(u) \in D_\mathsf{I}^{ass}(\tau_1)\}$.

Open questions

Some **open questions**:

- **is it worth it**?
- a less elegant **alternative**:
  - instead of the general relation $\mathsf{c}(\boxed{0}, \alpha)$ taking a second-order argument α
  - define a standard relation $\mathsf{c}_\alpha(\boxed{0})$ for each $\mathsf{c}(\boxed{0}, \alpha)$ actually needed in the grammar.

Open questions

Some **open questions**:

- **is it worth it**?
- a less elegant **alternative**:
  - instead of the general relation $c(\boxed{0}, \alpha)$ taking a second-order argument α
  - define a standard relation $c_\alpha(\boxed{0})$ for each $c(\boxed{0}, \alpha)$ actually needed in the grammar.

Open questions

Some **open questions**:

- **is it worth it**?
- a less elegant **alternative**:
  - instead of the general relation $c(\boxed{0}, \alpha)$ taking a second-order argument α
  - define a standard relation $c_\alpha(\boxed{0})$ for each $c(\boxed{0}, \alpha)$ actually needed in the grammar.

Open questions

Some **open questions**:

- **is it worth it**?
- a less elegant **alternative**:
  - instead of the general relation $\mathsf{c}(\boxed{0}, \alpha)$ taking a second-order argument α
  - define a standard relation $\mathsf{c}_\alpha(\boxed{0})$ for each $\mathsf{c}(\boxed{0}, \alpha)$ actually needed in the grammar.

Open questions

Some **open questions**:

- **is it worth it**?
- a less elegant **alternative**:
  - instead of the general relation $\mathsf{c}(\boxed{0}, \alpha)$ taking a second-order argument α
  - define a standard relation $\mathsf{c}_{\alpha}(\boxed{0})$ for each $\mathsf{c}(\boxed{0}, \alpha)$ actually needed in the grammar.

Instead of conclusion

Solutions:

- **"Höhle's problem"**:
  - **General Identity Principle** (GIP; Sailer 2003: 116),
  - modified to exempt some structures (e.g., of sort *ref*) from its scope;
- **exhaustive models** (twin structures, stranded structures, ontological problems):
  - **modification of RSRL**,
  - each model corresponds to an **utterance**,
  - and has a distinguished element – the **root**;
- **unlike category coordination** (Yatabe 2004):
  - **extension of RSRL**,
  - values of variables – **arguments of relations** – may be objects or **properties of objects**,
  - i.e., second-order RSRL.

Instead of conclusion

Solutions:

- **"Höhle's problem"**:
  - **General Identity Principle** (GIP; Sailer 2003: 116),
  - modified to exempt some structures (e.g., of sort *ref*) from its scope;
- **exhaustive models** (twin structures, stranded structures, ontological problems):
  - **modification of RSRL**,
  - each model corresponds to an **utterance**,
  - and has a distinguished element – the **root**;
- **unlike category coordination** (Yatabe 2004):
  - **extension of RSRL**,
  - values of variables – **arguments of relations** – may be objects or **properties of objects**,
  - i.e., second-order RSRL.

Instead of conclusion

Solutions:

- **"Höhle's problem"**:
  - **General Identity Principle** (GIP; Sailer 2003: 116),
  - modified to exempt some structures (e.g., of sort *ref*) from its scope;
- **exhaustive models** (twin structures, stranded structures, ontological problems):
  - **modification of RSRL**,
  - each model corresponds to an **utterance**,
  - and has a distinguished element – the **root**;
- **unlike category coordination** (Yatabe 2004):
  - **extension of RSRL**,
  - values of variables – **arguments of relations** – may be objects or **properties of objects**,
  - i.e., second-order RSRL.

Instead of conclusion

Solutions:

- **"Höhle's problem"**:
  - **General Identity Principle** (GIP; Sailer 2003: 116),
  - modified to exempt some structures (e.g., of sort *ref*) from its scope;
- **exhaustive models** (twin structures, stranded structures, ontological problems):
  - **modification of RSRL**,
  - each model corresponds to an **utterance**,
  - and has a distinguished element – the **root**;
- **unlike category coordination** (Yatabe 2004):
  - **extension of RSRL**,
  - values of variables – **arguments of relations** – may be objects or **properties of objects**,
  - i.e., second-order RSRL.

Instead of conclusion

Solutions:

- **"Höhle's problem"**:
  - **General Identity Principle** (GIP; Sailer 2003: 116),
  - modified to exempt some structures (e.g., of sort *ref*) from its scope;
- **exhaustive models** (twin structures, stranded structures, ontological problems):
  - **modification of RSRL**,
  - each model corresponds to an **utterance**,
  - and has a distinguished element – the **root**;
- **unlike category coordination** (Yatabe 2004):
  - **extension of RSRL**,
  - values of variables – **arguments of relations** – may be objects or **properties of objects**,
  - i.e., second-order RSRL.

Thank you for your attention!

Abeillé, A. and Chaves, R. (2021). Coordination. In S. Müller, A. Abeillé, R. D. Borsley, and J.-P. Koenig, eds., *Head-Driven Phrase Structure Grammar: The Handbook*. Language Science Press, Berlin. Forthcoming.

Bayer, S. (1996). The coordination of unlike categories. *Language*, **72**(3), 579–616.

Beavers, J. and Sag, I. A. (2004). Coordinate ellipsis and apparent non-constituent coordination. In Müller (2004), pp. 48–69.

Blackburn, P., de Rijke, M., and Venema, Y. (2010). *Modal Logic*. Cambridge University Press, Cambridge. Fourth printing (with corrections).

Chaves, R. P. (2006). Coordination of unlikes without unlike categories. In S. Müller, ed., *Proceedings of the HPSG 2006 Conference*, pp. 102–122, Stanford, CA. CSLI Publications.

Chaves, R. P. (2007). *Coordinate Structures: Constraint-Based Syntax-Semantics Processing*. Ph.D. dissertation, Universidade de Lisboa.

Chaves, R. P. (2008). Linearization-based word-part ellipsis. *Linguistics and Philosophy*, **31**, 261–307.

Chaves, R. P. (2013). Grammatical alignments and the gradience of lexical categories. In P. Hofmeister and E. Norcliffe, eds., *The Core and the Periphery: Data-Driven Perspectives on Syntax Inspired by Ivan A. Sag*, pp. 167–220. CSLI Publications, Stanford, CA.

Crysmann, B. (2003). An asymmetric theory of peripheral sharing in HPSG: Conjunction reduction and coordination of unlikes. In G. Jäger, P. Monachesi, G. Penn, and S. Wintner, eds., *Proceedings of Formal Grammar 2003*, pp. 47–62.

Dalrymple, M. (2017). Unlike phrase structure category coordination. In V. Rosén and K. De Smedt, eds., *The Very Model of a Modern Linguist*, pp. 33–55. University of Bergen Library, Bergen.

Höhle, T. N. (1999). An architecture for phonology. In R. D. Borsley and A. Przepiórkowski, eds., *Slavic in Head-Driven Phrase Structure Grammar*, pp. 61–90. CSLI Publications, Stanford, CA. Reprinted in Müller *et al.* 2019.

King, P. J. (1999). Towards truth in HPSG. In V. Kordoni, ed., *Tübingen Studies in Head-Driven Phrase Structure Grammar*, pp. 301–352, Tübingen. Universität Tübingen.

Levine, R. D. (2011). Linearization and its discontents. In S. Müller, ed., *Proceedings of the HPSG 2011 Conference*, pp. 126–146, Stanford, CA. CSLI Publications.

Meurers, W. D. (1998). Raising spirits (and assigning them case). Handout for a talk delivered during the Groningen-Tübingen Kolloquium, Tübingen, 27th November 1998.

Müller, S., ed. (2004). *Proceedings of the HPSG 2004 Conference*. CSLI Publications, Stanford, CA.

Müller, S., Reis, M., and Richter, F., eds. (2019). *Beiträge zur deutschen Grammatik: Gesammelte Schriften von Tilman N. Höhle*. Language Science Press, Berlin.

Patejuk, A. and Przepiórkowski, A. (2021). Category mismatches in coordination vindicated. *Linguistic Inquiry*. Forthcoming.

Pollard, C. (1999). Strong generative capacity in HPSG. In G. Webelhuth, J.-P. Koenig, and A. Kathol, eds., *Lexical and Constructional Aspects of Linguistic Explanation*, pp. 281–297. CSLI Publications, Stanford, CA.

Pollard, C. (2001). Cleaning the HPSG garage: Some problems and some proposals. Unpublished manuscript, Ohio State University; `http://utkl.ff.cuni.cz/~rosen/public/cp_garage.ps`.

Pollard, C. (2014). Type-logical HPSG. In G. Jäger, P. Monachesi, G. Penn, and S. Wintner, eds., *Proceedings of Formal Grammar 2004*, p. 111–128. CSLI Publications.

Pollard, C. and Sag, I. A. (1994). *Head-driven Phrase Structure Grammar*. Chicago University Press / CSLI Publications, Chicago, IL.

Pullum, G. K. and Scholz, B. C. (2001). On the distinction between model-theoretic and generative-enumerative syntactic frameworks.

In P. de Groote, G. Morrill, and C. Retoré, eds., *Logical Aspects of Computational Linguistics: 4th International Conference*, pp. 17–43. Springer-Verlag, Berlin.

Richter, F. (2004). *A Mathematical Formalism for Linguistic Theories with an Application in Head-Driven Phrase Structure Grammar*. Ph.D. dissertation (defended in 2000), Universität Tübingen.

Richter, F. (2007). Closer to the truth: A new model theory for HPSG. In J. Rogers and S. Kepser, eds., *Model-Theoretic Syntax at 10*, pp. 99–108.

Sailer, M. (2003). Local semantics in Head-driven Phrase Structure Grammar. Paper presented at the Colloque de Syntaxe et Sémantique à Paris.

Yatabe, S. (2004). A comprehensive theory of coordination of unlikes. In Müller (2004), pp. 335–355.