A Uniform Architecture for Parsing and Generation of Natural Language

Günter Neumann

DFKI GmbH

66123 Saarbrücken
neumann@dfki.de

Overview

Work based on Neumann:94 (Ph.D. thesis), Neumann:98 (AIJ)

1. **Parsing and Generation**
2. **Results**
3. **Motivation**
4. **State of the Art**
5. **A New Uniform Architecture**
6. **Parsing and Generation: UTA can do both**
7. **Interleaving of Parsing and Generation**
8. **Conclusion and Future Direction**

Uniform grammatical processing

- **Parsing: given a string, compute all possible logical forms (wrt. the given grammar)**

- **Generation: given a logical form, compute all possible strings**

- **Uniformity**
  - **use of one and the same grammar for performing both tasks**
    ⟹ reversible grammar
  - **use of the same algorithm**
    ⟹ uniform algorithm

Results

- **Uniform Tabular Algorithm (UTA):**
  - **constraint-based grammars**
  - **generalized Early deduction**
  - **flexible agenda mechanism**
  - **On-line**
  - **Input as essential feature**
    - **dynamic selection function**
    - **uniform chart mechanism**

  $\Longrightarrow$ **uniform <u>and</u> task-oriented processing**

- **Performanz model on the basis of a uniform architecture:**
  - **item-sharing between parsing and generation**
  - **incremental self-monitoring/revison strategies**
  - **generation of un-ambiguous strings**
  - **generation of paraphrases**
  - **any-time mode**

  $\longrightarrow$ **interleaved parsing and generation**

- **Implementation in Common Lisp and CLOS**

Why uniform grammatical processing ?

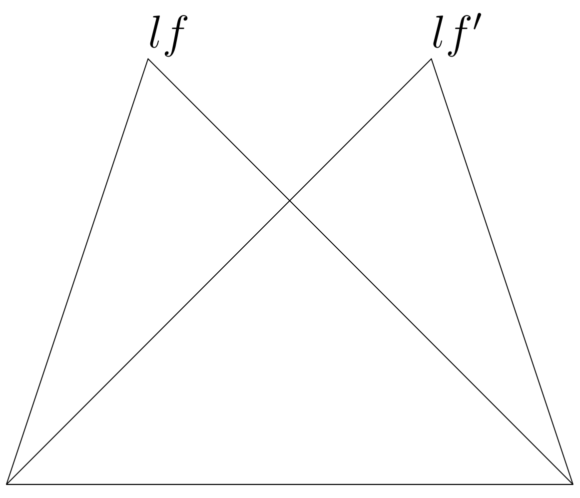
- **Theoretical:**
  - **Occam's razor**
  - **psycho-linguistic motivations**

- **Practical:**
  - **reduced redundancies**
  - **simpler consistency tests**
  - **knowledge acquistion**
  - **compact and modular systems**

- **Application:**
  - **grammar development**
  - **interactive grammar-/style-checker**
  - **incremental text processing**
  - **monitoring and revision**
  - **generation of paraphrases**
  - **processing of elliptic expressions**
  - **combination of learning-/ preference-based methods**
  - ...

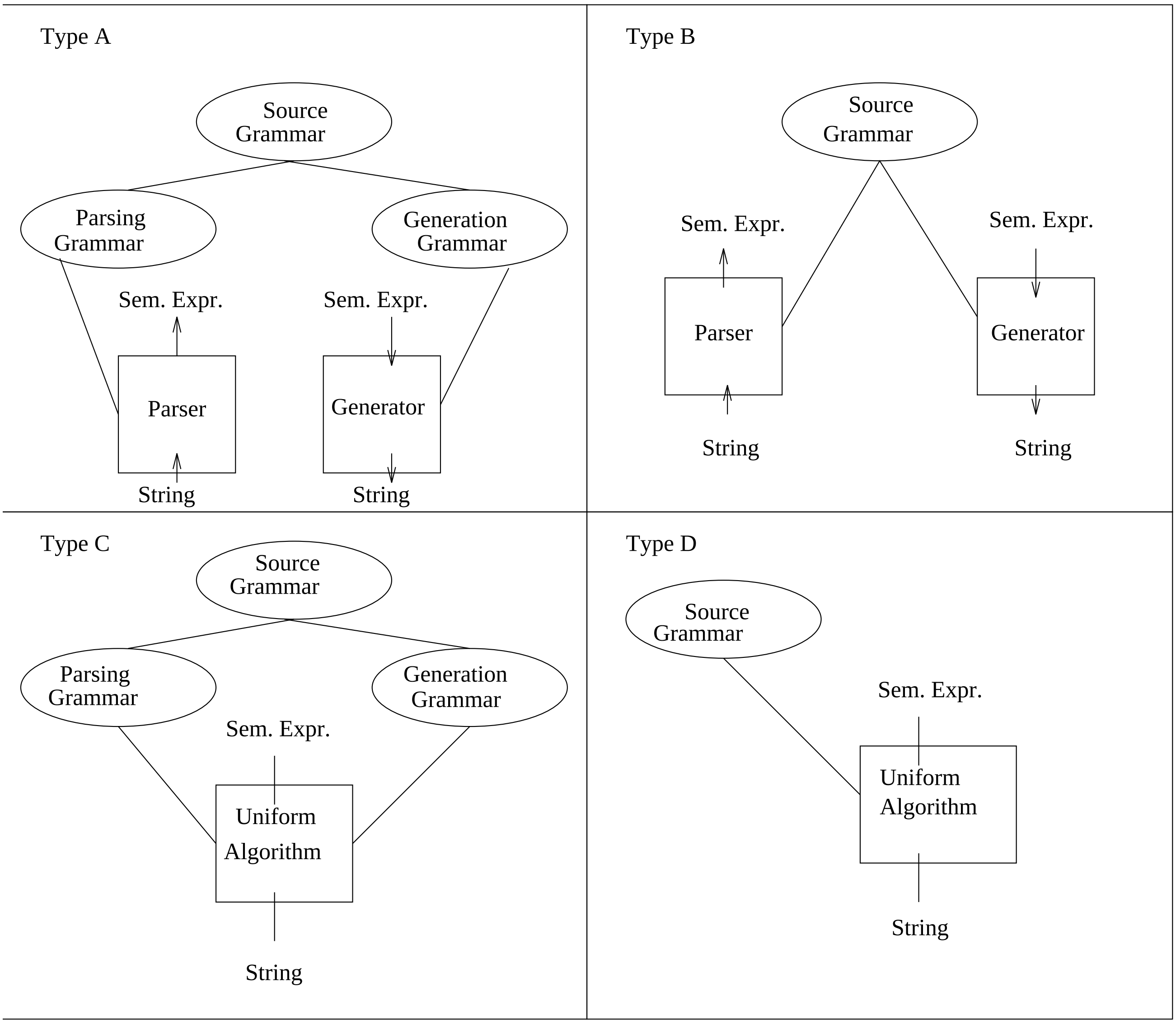
Reversible grammars

- **Language as a relation R:**
  wellformed strings $\times$ logical forms ($R \subseteq S \times LF$)
- **Parsing:** s, **compute** $\{lf_i| < s, lf_i > \in R\}$
- **Generation:** lf, **compute** $\{s_i| < s_i, lf > \in R\}$
- **Reversible grammar: define R with one grammar**
- **Ambiguity and paraphrases**



Lösche das Verzeichnis mit den Systemtools!

Current state of the art

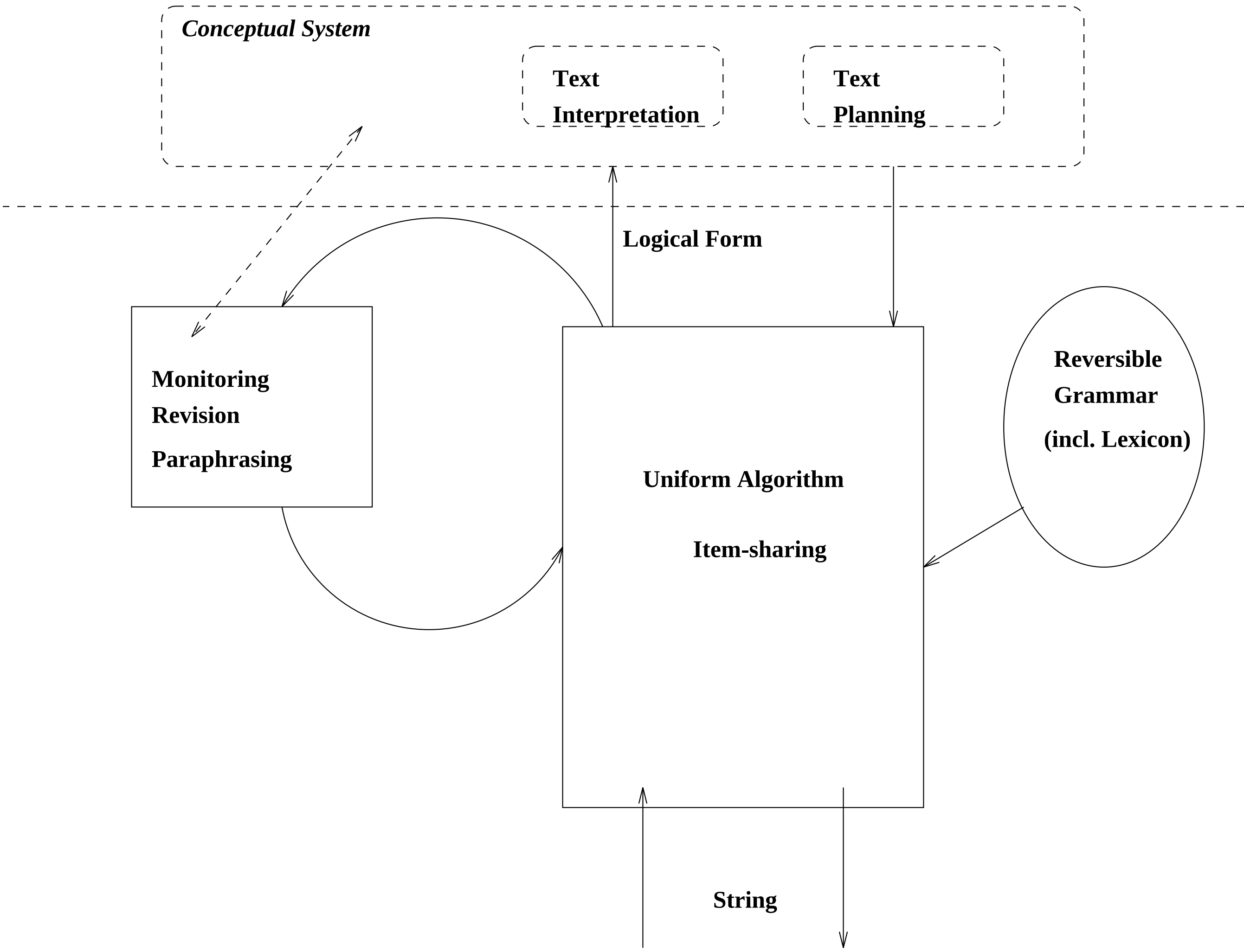


Disadvantages of current models

- **Types A, B, C**
  - **approaches: Block (A), Strzalkowski (C), Dymetman et al. (C)**
  - **high degree of redundancies(A,C)**
  - **testing of source grammar not possible (A,C)**
  - **interleaved parsing and generation not meaningful**

- **Type D**
  - **approaches: Shieber, van Noord, Gerdeman**
  - **interleaved approach possible**
  - **poor dynamic of the models**
  - **parsing-oriented chart**
  - **restricted view on uniformity**

A New Uniform Model



Constraint-based grammars

- **e.g., LFG, HPSG, CUG**
- **Reversibility**
  - **uniform representation** (PHON, SYN, SEM)
  - **word, phrase, clause level**
  - **structure sharing**
  - **declarative**

Example:

$$
\begin{bmatrix}
\texttt{cat} & \textit{sentence} \\
\texttt{phon} & \langle \textit{peter}, \textit{cries} \rangle \\
\texttt{syntax} & \ldots \\
\texttt{dtrs} & \left\langle
\begin{bmatrix}
\texttt{cat} & \textit{noun} \\
\texttt{phon} & \textit{peter} \\
\texttt{syntax} & \begin{bmatrix} \texttt{agr} & \begin{bmatrix} \texttt{per} & \textit{3} \\ \texttt{num} & \textit{sg} \end{bmatrix} \end{bmatrix} \\
\texttt{semantics} & {}_{\textit{Arg}}\begin{bmatrix} \texttt{rel} & \textit{the-peter'} \end{bmatrix}
\end{bmatrix},
\begin{bmatrix}
\texttt{cat} & \textit{verb} \\
\texttt{phon} & \textit{cries} \\
\texttt{syntax} & \begin{bmatrix} \texttt{agr} & \begin{bmatrix} \texttt{per} & \textit{3} \\ \texttt{num} & \textit{sg} \end{bmatrix} \end{bmatrix} \\
\texttt{semantics} & {}_{\textit{Sem}}\begin{bmatrix} \texttt{rel} & \textit{cry} \\ \texttt{arg} & \textit{Arg} \end{bmatrix}
\end{bmatrix}
\right\rangle \\
\texttt{semantics} & \textit{Sem}
\end{bmatrix}
$$

Constraint Logic programming CLP

- **Generalization of conventional logic programming to arbitrary constraint-languages (Hoefeld&Smolka:88)**

- **Representation of grammar as definite clauses**
  - **rule:** $q \leftarrow p_1, \ldots, p_n, \phi$
  - **lexical element:** $q \leftarrow \phi$

- **Goal-reduction rule:**

$$\begin{array}{rl} \textbf{goal:} & p_1, \ldots, p(\vec{x}), \ldots p_n, \phi \\ \textbf{clause:} & p(\vec{x}) \leftarrow q_1, \ldots, q_m, \psi \\ \Longrightarrow & \\ \textbf{new goal:} & p_1, \ldots, q_1, \ldots, q_m, \ldots, p_n, \phi, \psi \end{array}$$

- **Constraint-solver:** $unify(\phi, \psi)$

- **Parsing and generation: queries of form** $\leftarrow q, \phi$

UTA: A Uniform Algorithm for Parsing and Generation

- **Goal: uniform <u>and</u> task-oriented Processing**
- **Uniform control logic: generalized Earley deduction (based on Pereira&Warren:83)**
  - **grammar (rule, lexicon), item sets**
  - **item: lemma with selected element (SEL)**
  - **active item (AI):** $\langle h \leftarrow b_0 \ldots b_n \ ; \ i \ ; idx \rangle$
  - **passive item(PI):** $\langle h \leftarrow \epsilon \ ; \ \epsilon \ ; idx \rangle$
  - **blocking-test: subsumption**

UTA: A Uniform Algorithm for Parsing and Generation

- **Inference rules:**
  - **prediction:** ABSTR(SEL(**AI**)) *unify* **head(rule)**
  - **completion: AI minus** SEL
    - **scanning:** SEL(**AI**) *unify* **lexical element**
    - **active completion:** SEL(**AI**) *unify* **PI**
    - **passive completion: PI** *unify* SEL(**AI**)

- **New clauses (Items): determine** SEL **using dynamic selection function** SF

  Prediction: $\langle \Phi[\textbf{Rule}]; \textsc{sf}(\Phi[\textbf{Rule}], EF); Idx \rangle$

Parametrization of UTA

- **Relevant parameter: Essential Feature EF $\Longrightarrow$ the feature, that carries the input (e.g., PHON or SEM)**
  - **parametrized selection function**
    - **EF guides ordering of processing of rhs(rule)**
  - **paramertized item set**
    - **EF used for defining equivalence classes**
- **Parsing and generation with UTA $\Longrightarrow$ main difference is the different input structure**

Parametrizable selection function

- **Choose that element, whose Essential Feature is instantiated, else take the left-most one**
- **Implications:**
  - **data-driven selection, e.g.,**
    - **left-to-right (e.g., for parsing)**
    - **functor-first (e.g., for generation)**
    - **or both**
    - **integration of preferences**
  - **grammar itself has influence on control**

Beispiel:

$$sign\left(\begin{bmatrix}\texttt{cat:} & vp\\ \texttt{sc:} & Tail\\ \texttt{sem:} & Sem\\ \texttt{lex:} & no\\ \texttt{v2:} & V\\ \texttt{phon:} & P_0\text{-}P\end{bmatrix}\right) \longleftarrow sign\left({}_{Arg}\begin{bmatrix}\texttt{phon:} & P_0\text{-}P_1\end{bmatrix}\right),\ sign\left(\begin{bmatrix}\texttt{cat:} & vp\\ \texttt{sc:} & \langle Arg|Tail\rangle\\ \texttt{sem:} & Sem\\ \texttt{v2:} & V\\ \texttt{phon:} & P_1\text{-}P\end{bmatrix}\right)$$

Structured item set

- **Idea: divide item set into equivalence classes**

- **Determination of equivalence classes by means of Essential Feature**

$\Longrightarrow$ **item set is structured according to input structure, e.g.,**
  - **as sequence in case of parsing**
  - **as funktor/argument tree in case of generation**
  - **set in case of MRS**

- **Advantage:**
  - **application of inference rules on subsets**
  - **blocking-test only on subsets**
  - **on-the-fly creation**

- **Details:**
  - **item set:** $\langle AI, PI, Idx \rangle$
  - $\forall$ **items: EF compatible** $\longrightarrow$ **Idx**
  - **PI: EF of Head, AI: EF of SEL**

Flexible agenda mechanism

- **Guides order of processing of new items**
- **Sorts items according to preference**
- **Activation of clauses and insertion into item set according to preference**
- **Advantage:**
  - **depth-first, breadth-first, best-first, random**
  - **blocking-test only on "activated" clauses**
  - **interleaved parsing and generation: different preference rules**

Parsing and Generation

- **Parsing and generation as queries with instantiated essential feature**

- **Parsing:**
$\leftarrow sign(\left[\texttt{phon} \quad \langle heute, erz\ddot{a}hlt, peter, l\ddot{u}gen\rangle\text{-}\langle\ \rangle\right])$

- **Generation:**
$$\leftarrow sign(\left[\texttt{sem} \left[\begin{array}{ll}\texttt{mod} & heute \\ \texttt{arg1} & \left[\begin{array}{ll}\texttt{pred} & erz\ddot{a}hlen \\ \texttt{arg1} & peter \\ \texttt{arg2} & l\ddot{u}gen\end{array}\right]\end{array}\right]\right])$$

- **EF-proof problem (s.a. VanNoord:93):**
Value of $\mathbf{EF}_{Query}$ = **Value of** $\mathbf{EF}_{Answer}$

$22\langle ans;\epsilon;0\rangle 22$
$21\langle vp;\epsilon;0\rangle 21$
$1\langle vp\leftarrow v\ np\ pp;0;0\rangle 18$
$18\langle ans;\epsilon;0\rangle 16$
$17\langle vp;\epsilon;0\rangle 15$
$2\langle vp\leftarrow v\ np;0;0\rangle 2$
$0\langle ans\leftarrow vp;0;0\rangle 1$

sPmM_0

$19\langle vp\leftarrow np\ pp;0;1\rangle 19$
$16\langle np;\epsilon;1\rangle 14$
$8\langle np;\epsilon;1\rangle 7$
$7\langle np\leftarrow n;0;1\rangle 6$
$4\langle np\leftarrow np\ pp;0;1\rangle 5$
$5\langle np\leftarrow n;0;1\rangle 4$
$3\langle vp\leftarrow np;0;1\rangle 3$

PmM_1

$20\langle vp\leftarrow pp;0;2\rangle 20$
$15\langle pp;\epsilon;2\rangle 13$
$10\langle pp\leftarrow p\ np;0;2\rangle 9$
$9\langle np\leftarrow pp;0;2\rangle 8$

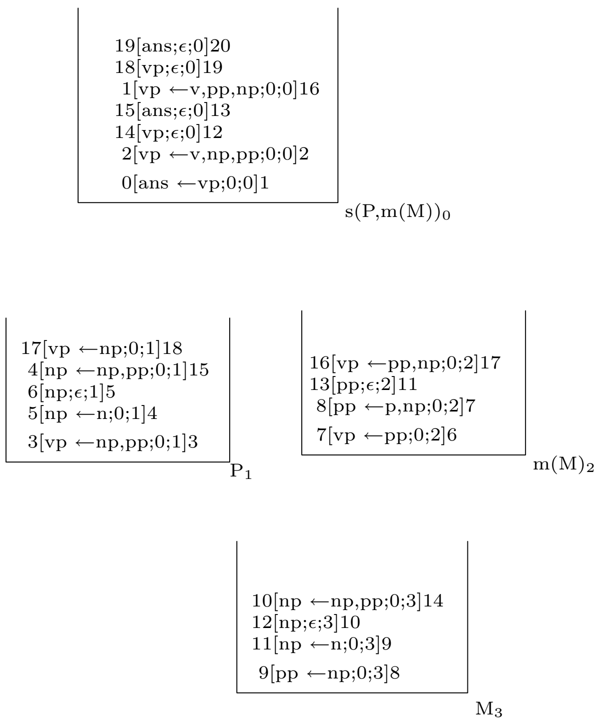
mM_2

$12\langle np\leftarrow np\ pp;0;3\rangle 17$
$14\langle np;\epsilon;3\rangle 12$
$13\langle np\leftarrow n;0;3\rangle 11$
$11\langle pp\leftarrow np;0;3\rangle 10$

M_3

Agenda	CurrentTask	Item of alternative
0	0	$1\langle vp\leftarrow v\ nppp;0;0\rangle$
1,2	2	
1,3	3	
1,4,5	5	$4\langle np\leftarrow np\ pp;0;1\rangle$
1,4	4	
1,6,7	7	$6\langle np\leftarrow np\ pp;0;1\rangle$
1,6,8	8	
1,6,9	9	
1,6,10	10	
1,6,11	11	
1,6,12,13	13	$12\langle np\leftarrow np\ pp;0;3\rangle$
1,6,12,14	14	
1,6,12,15	15	
1,6,12,16	16	
1,6,12,17	17	
1,6,12,18	18	First Result
1,6,12	12	
1,6	6	
1	1	
19	19	
20	20	
21	21	
22	22	Second Result
ϵ		

Parsing of “sieht Peter mit Maria”



Agenda	Current Task	Item of alternative
0	0	
1,2	2	1[vp ←v,pp,np;0;0]
1,3	3	
1,4,5	5	4[np ←np,pp;0;1]
1,4,6	6	
1,4,7	7	
1,4,8	8	
1,4,9	9	
1,4,10,11	11	10[np ←np,pp;0;3]
1,4,10,12	12	
1,4,10,13	13	
1,4,10,14	14	
1,4,10,15	15	First paraphrase
1,4,10	10	
1,4	4	
1	1	
16	16	
17	17	
18	18	
19	19	Second paraphrase
ϵ		

Generation of "sehen(Peter,mit(Maria))"

Interleaving of parsing and generation

- **Consider parsing and generation not in isolation**
  - **use of partial results of the other direction**
    $\Longrightarrow$ item-sharing between parsing & generation
  - **use of one direction**
    as additional control of the other
    $\Longrightarrow$ incremental self-monitoring
- **Uniform algorithm required**

Item-Sharing

- **Idea: exchange of partial results between parsing and generation**

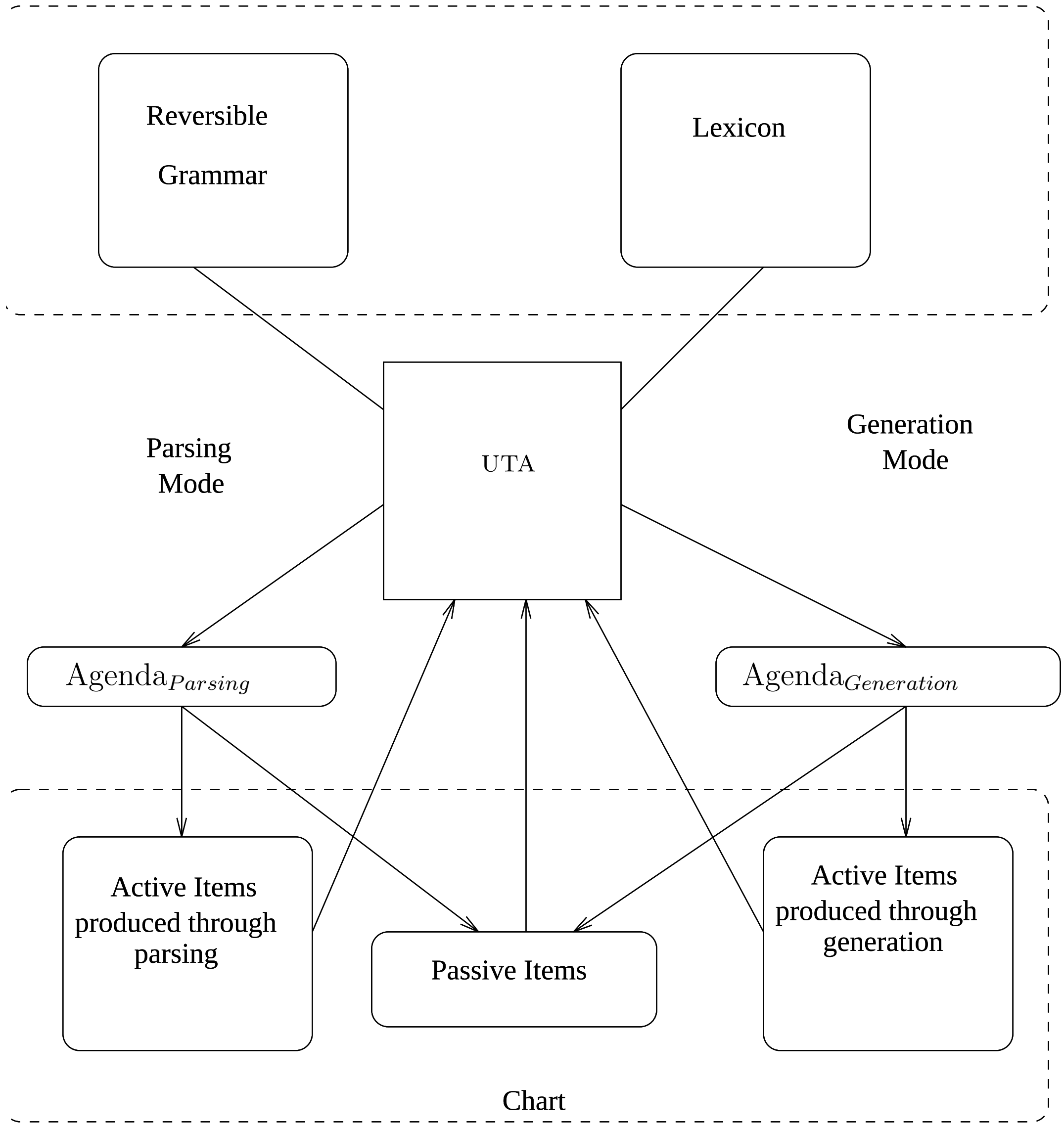
  $\Longrightarrow$

  computed passive items of one direction are automatically provided for the other direction

- **Parsing and generation use same passive items $\Longrightarrow$ item-sharing**

- **Advantage: re-use of partial results for parsing <u>and</u> generation**

Item sharing: architecture



Monitoring and revision during generation

- **Psycholing. motivation (cf. Levelt:89)**
- **Here: avoid mis-understandings**

  "Lösche das Verzeichnis mit den Systemtools"

  "Lösche mit den Systemtools das Verzeichnis"

- **Generation of paraphrases $\Longrightarrow$ interactive desambiguierung**

  "Do you mean: X or Y ?"

- **Problem:**
  choice between possible paraphrases

Monitoring and revision: Idea

- **'Revise' relevante structures of an ambiguous utterance**

  ⟹ parsing to recover ambiguities

- **Assumption: It is possible to revise an utterance locally in order to generate an un-ambiguous utterance with the same meaning**

- **Neumann und VanNoord 92, non-incremental algorithm**

Incremental
Monitoring and revision

- **If local, then revision applicable on partial structures $\Longrightarrow$ incremental**

– **Removing the folder with the system tools can be very dangerous.**

– **Visiting relatives can be boring.**

– **Visiting relatives are boring.**

– **During the ball I danced with a lot of people.**

– **I know of no better ball.**

Problems for incremental self-monitoring

- **When should ambiguity check take place?**
- **For which just generated partial string α should revision take place?**

$\Longrightarrow$ **Lookback(n)-stratgy:**

  - **parse α relative to adjacent already generated partial string β**
  - **if β does not exist, or $\beta\alpha$ not parsable or not ambigious, then no revision**

Fundamental strategy

- **Generate/Parse/Revise feedback-loop:**
  - **during generation for just generated substring α**
  - **monitoring: parse α relative to context**
  - **revision: if ambigious, then revision**

    "[Lösche][das Verzeichnis]...[mit den S.t.]"
    $\Longrightarrow$ parse
    "das Verzeichnis mit den S.t."
    $\Longrightarrow$ revise
    "[Lösche][mit den S.t.][das Verzeichnis]"

Details

p-completion(Pi) **is:**

For every active item $Ai \in I_{idx}$**:**

if $\Phi = $ UNIFY(SEL(Ai), h) **and** $\Phi \neq fail$ **then**

if NOT(AND($Monitor?$,REVISION-P($\Phi[Ai]$,Pi))) **then**

with reduced lemma $Rl = \Phi[Ai - \text{SEL}(Ai)]$ **do**

...

od

revision-p(Ai, Pi) **is:**

with $ExtendedString$ = GET-CONTEXT(Ai, Pi,n);

if $ExtendedString$ **then**

with $ParseRes$ = PARSE($ExtendedString$);

if AND($ParseRes$,AMBIGUOUS(Ai,$ParseRes$))

then true else false fi

else false fi.

Incremental self-monitoring with UTA

- **Self-embedded control of generation module**
- **Agenda mechansism automatically realizes revision**
- **Revision: prune possible set of answers**
- **Chart-based incremental interleaved parsing & generation**
- **Specific control:**
  - **any-time mode**
  - **lookback(n)-strategy**
  - **use of grammar-specific information can easily be integrated (preferences, no complements, subset of adjuncts)**

Conclusion: the main results

- **Novel: NLP on the basis of interleaved parsing and generation**
- **Theoretical: competence-based performance model**
- **Practical: uniformity as application impact**
- **For whole NLP:**
  - **self-monitoring/self-control $\Longrightarrow$**
    - **flexibility**
    - **robustness**
    - **adaptibility**

Possible future directions

- **Incremental generation with MRS**
- **Integration of Machine Learning**
  - **Explanation-based Learning (EBL): automatic computation of prototypical constructions (templates)**
  - **UTA: reversible EBL, Template-sharing**
  - **principle-guided induction of reversible grammars**

- **Preference-based methods:**
  - **stochastic lexicalized tree grammars (NeumannFlickinger:99)**
  - **hearer-adaptible monitoring/revision**

- **bidirectional dialog systems**