Chapter 11

Tree-based models

Statistical Machine Translation

Tree-Based Models

- Traditional statistical models operate on sequences of words

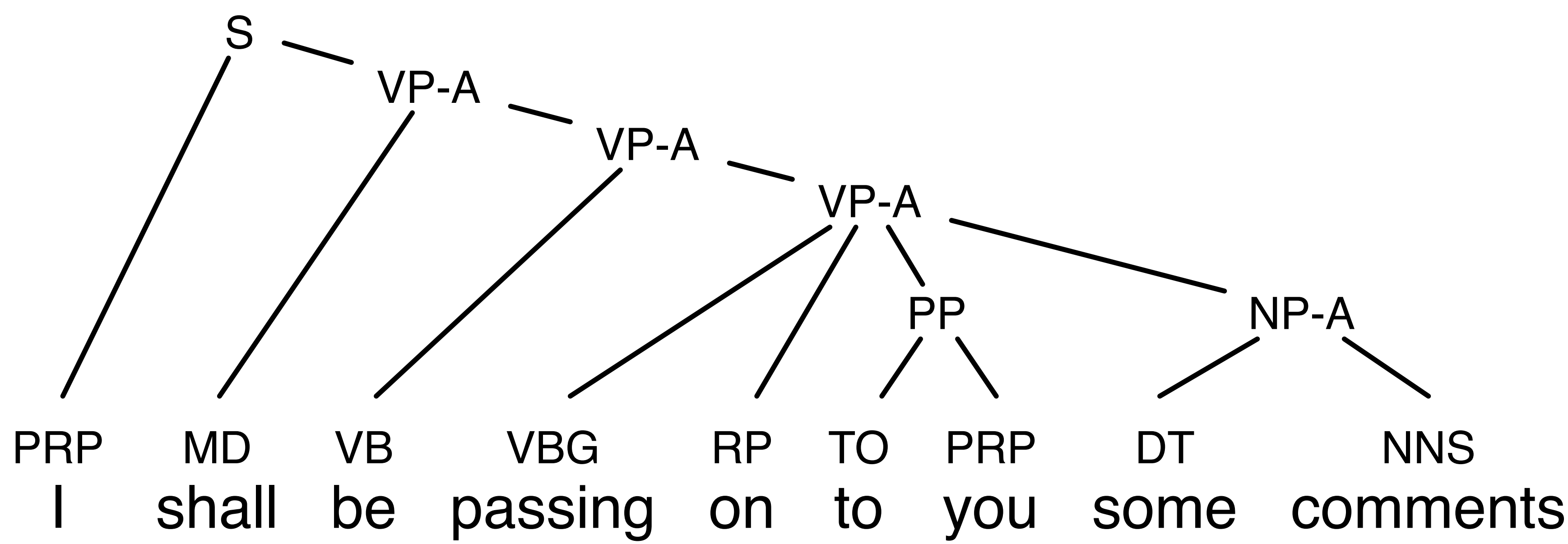
- Many translation problems can be best explained by pointing to syntax
  - reordering, e.g., verb movement in German–English translation
  - long distance agreement (e.g., subject-verb) in output

⇒ Translation models based on tree representation of language
  - significant ongoing research
  - state-of-the art for some language pairs

Phrase Structure Grammar

- Phrase structure
  - noun phrases: the big man, a house, ...
  - prepositional phrases: at 5 o'clock, in Edinburgh, ...
  - verb phrases: going out of business, eat chicken, ...
  - adjective phrases, ...
- Context-free Grammars (CFG)
  - non-terminal symbols: phrase structure labels, part-of-speech tags
  - terminal symbols: words
  - production rules: $\text{NT} \rightarrow [\text{NT,T}]+$
    example: $\text{NP} \rightarrow \text{DET NN}$

Phrase Structure Grammar



Phrase structure grammar tree for an English sentence
(as produced Collins' parser)

Synchronous Phrase Structure Grammar

- English rule

$$\text{NP} \longrightarrow \text{DET JJ NN}$$

- French rule

$$\text{NP} \longrightarrow \text{DET NN JJ}$$

- Synchronous rule (indices indicate alignment):

$$\text{NP} \longrightarrow \text{DET}_1 \ \text{NN}_2 \ \text{JJ}_3 \ \Big| \ \text{DET}_1 \ \text{JJ}_3 \ \text{NN}_2$$

Synchronous Grammar Rules

- Nonterminal rules

$$\text{NP} \rightarrow \text{DET}_1 \ \text{NN}_2 \ \text{JJ}_3 \ | \ \text{DET}_1 \ \text{JJ}_3 \ \text{NN}_2$$

- Terminal rules

$$\text{N} \rightarrow \text{maison} \ | \ \text{house}$$

$$\text{NP} \rightarrow \text{la maison bleue} \ | \ \text{the blue house}$$

- Mixed rules

$$\text{NP} \rightarrow \text{la maison } \text{JJ}_1 \ | \ \text{the } \text{JJ}_1 \text{ house}$$

Tree-Based Translation Model

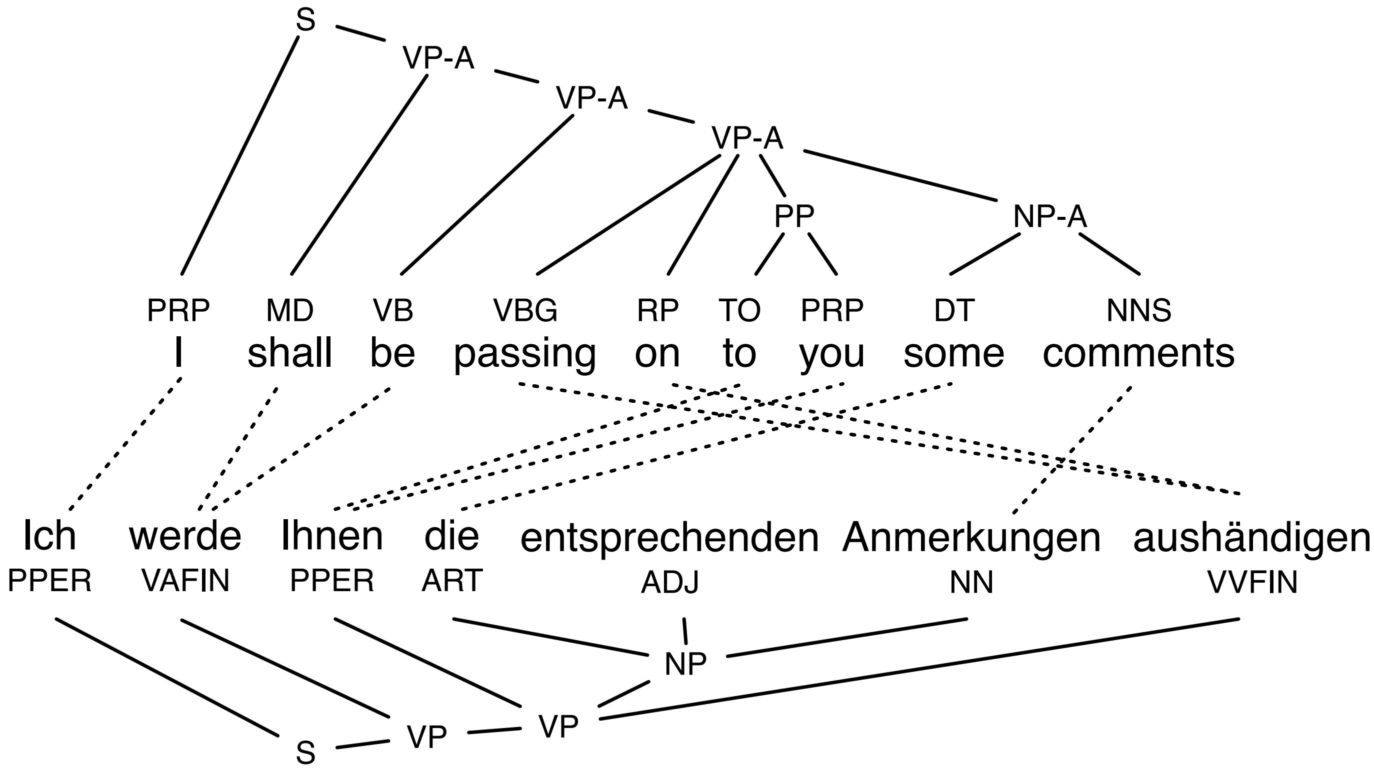
- Translation by parsing
  - synchronous grammar has to parse entire input sentence
  - output tree is generated at the same time
  - process is broken up into a number of rule applications

- Translation probability

$$\textsc{score}(\textsc{tree}, \textsc{e}, \textsc{f}) = \prod_i \textsc{rule}_i$$

- Many ways to assign probabilities to rules

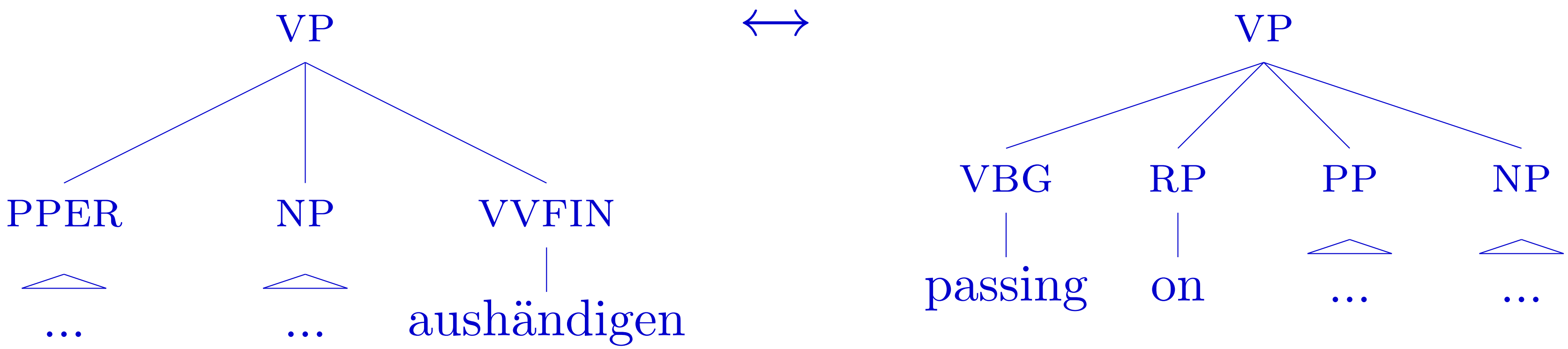
Aligned Tree Pair



Phrase structure grammar trees with word alignment
(German–English sentence pair.)

Reordering Rule

- Subtree alignment



- Synchronous grammar rule

$$\text{VP} \rightarrow \text{PPER}_1 \ \text{NP}_2 \ \text{aushändigen} \ \mid \ \text{passing on PP}_1 \ \text{NP}_2$$

- Note:
  - one word aushändigen mapped to two words passing on ok
  - but: fully non-terminal rule not possible
    (one-to-one mapping constraint for nonterminals)

Another Rule

- Subtree alignment

- Synchronous grammar rule (stripping out English internal structure)

$$\textsc{pro}/\textsc{pp} \rightarrow \text{Ihnen} \mid \text{to you}$$

- Rule with internal structure

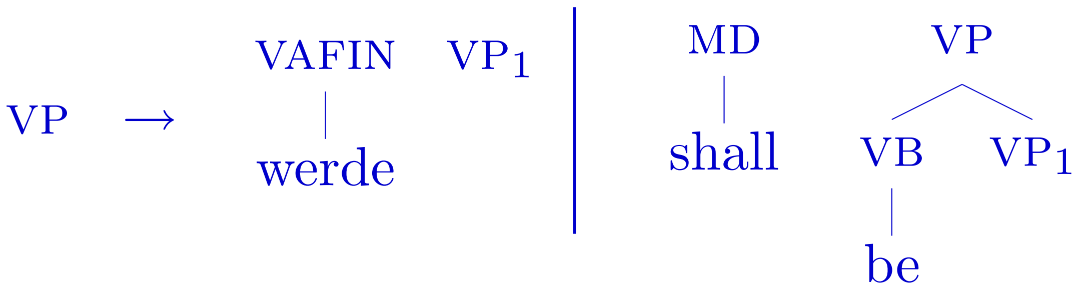
$$\textsc{pro}/\textsc{pp} \rightarrow \quad \text{Ihnen} \quad \left| \quad \begin{array}{cc} \text{TO} & \text{PRP} \\ | & | \\ \text{to} & \text{you} \end{array} \right.$$

Another Rule

- Translation of German werde to English shall be

- Translation rule needs to include mapping of VP

⇒ Complex rule



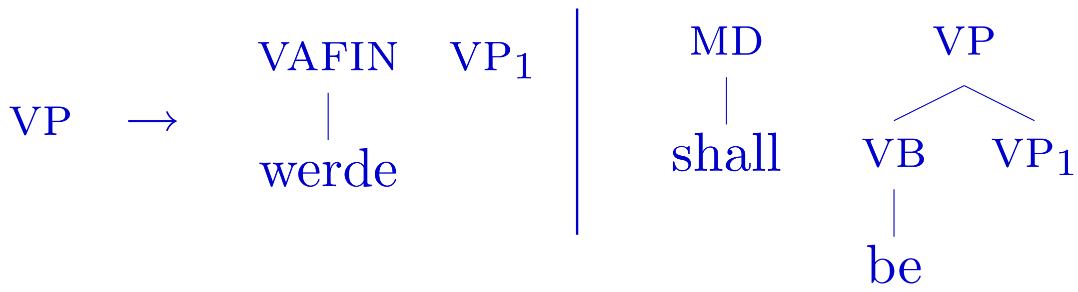
Internal Structure

- Stripping out internal structure

$$\text{VP} \rightarrow \text{werde VP}_1 \;\Big|\; \text{shall be VP}_1$$

$\Rightarrow$ synchronous context free grammar

- Maintaining internal structure



$\Rightarrow$ synchronous tree substitution grammar

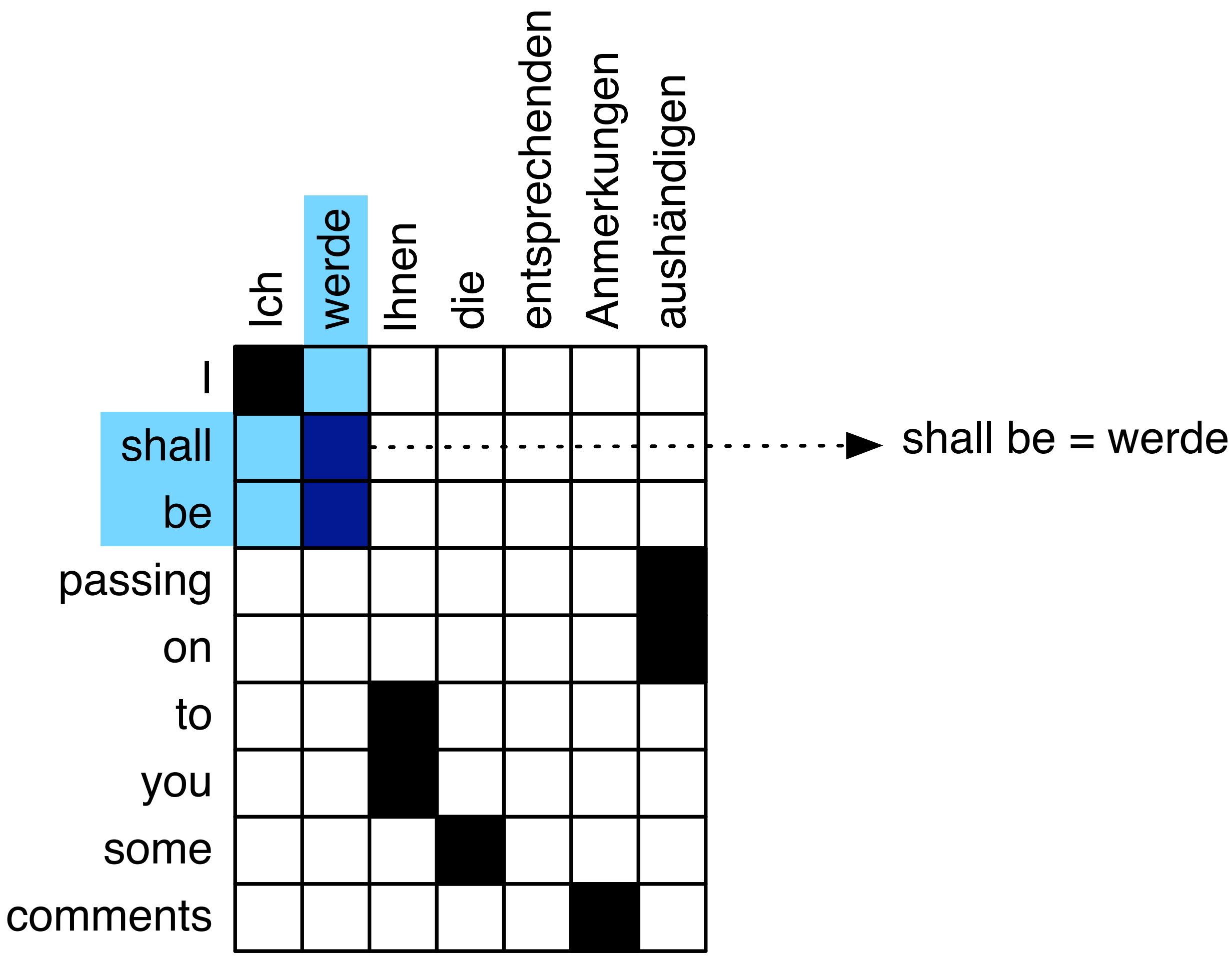
Learning Synchronous Grammars

- Extracting rules from a word-aligned parallel corpus

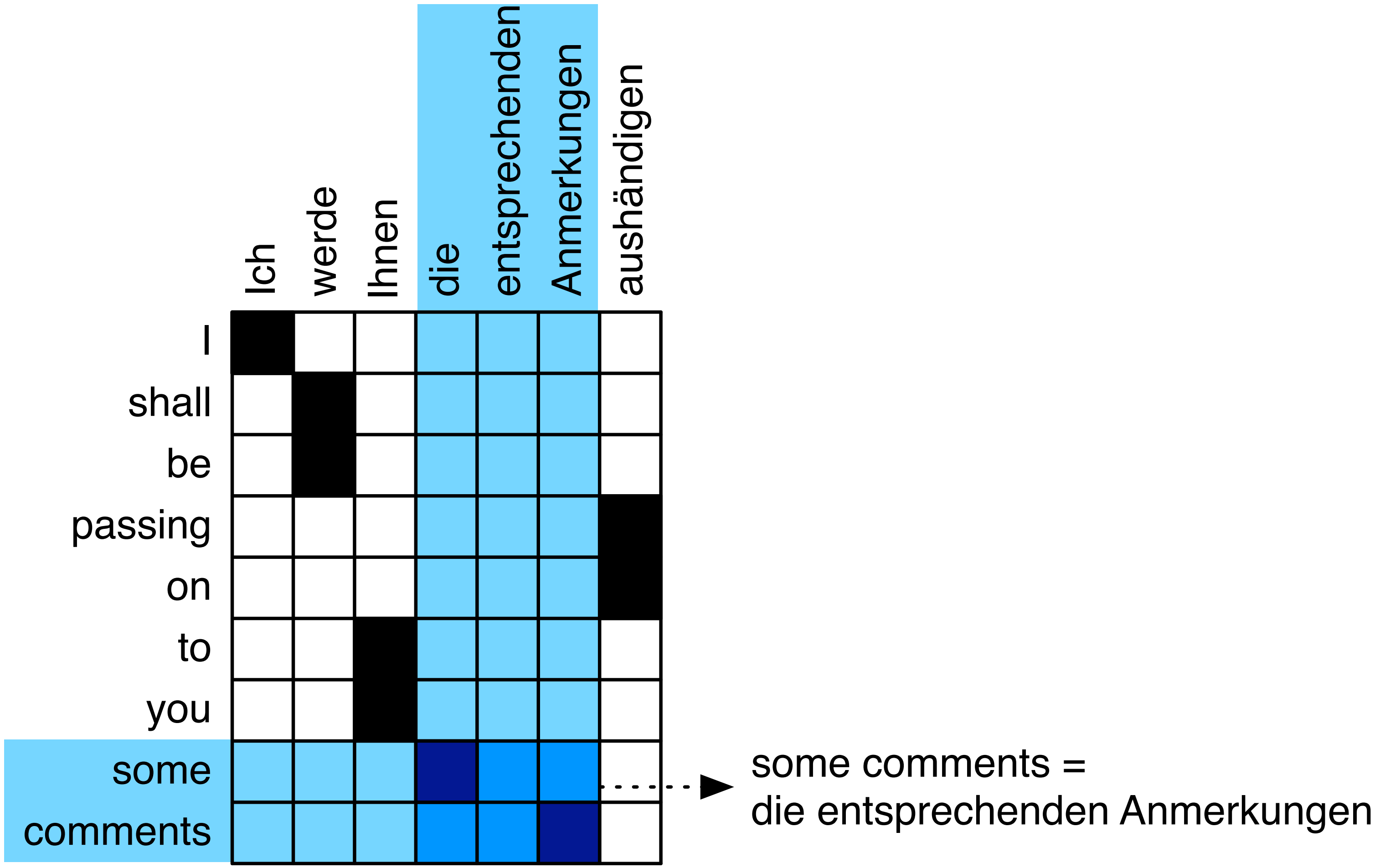
- First: Hierarchical phrase-based model
  - only one non-terminal symbol X
  - no linguistic syntax, just a formally syntactic model

- Then: Synchronous phrase structure model
  - non-terminals for words and phrases: NP, VP, PP, ADJ, ...
  - corpus must also be parsed with syntactic parser

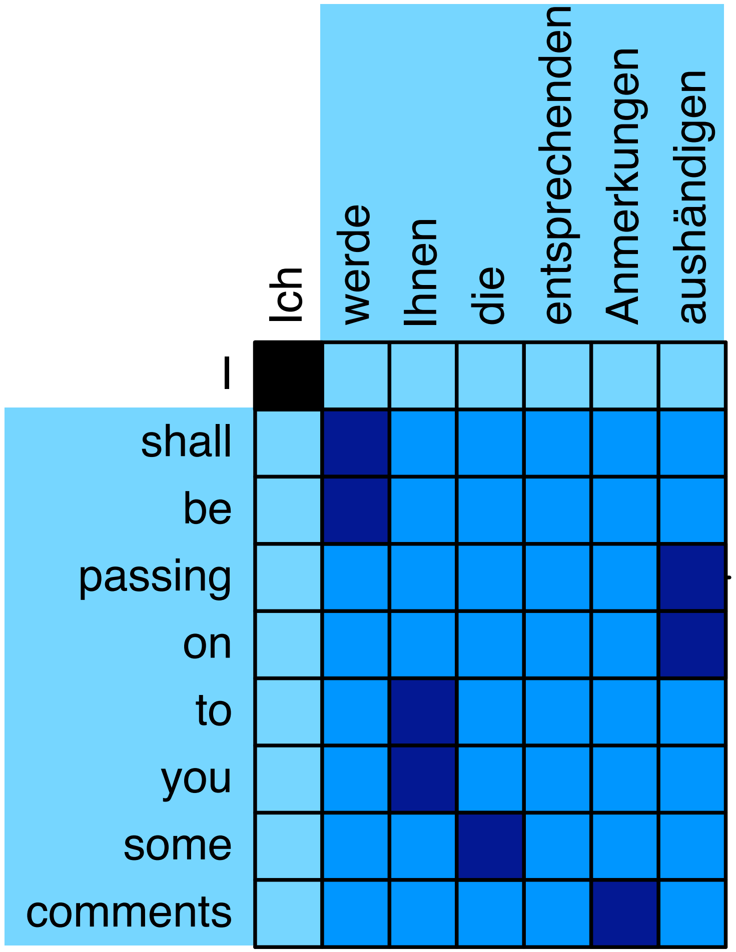
Extracting Phrase Translation Rules



Extracting Phrase Translation Rules

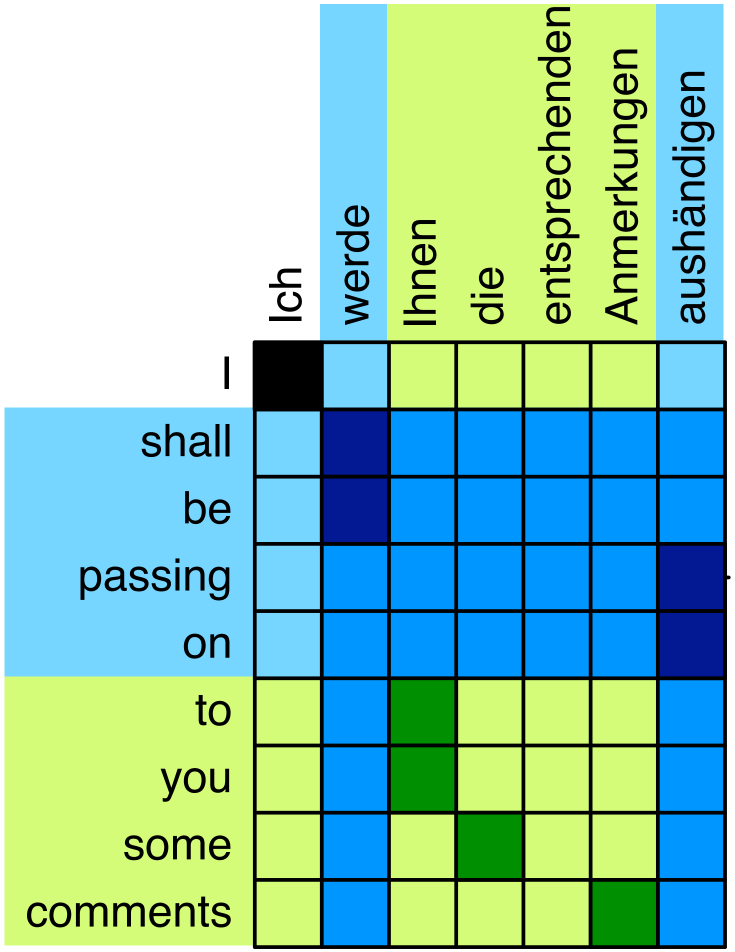


Extracting Phrase Translation Rules



werde Ihnen die entsprechenden Anmerkungen aushändigen
= shall be passing on to you some comments

Extracting Hierarchical Phrase Translation Rules



subtracting subphrase

werde X aushändigen
= shall be passing on X

Formal Definition

- Recall: consistent phrase pairs

$$(\bar{e}, \bar{f}) \text{ consistent with } A \Leftrightarrow$$
$$\forall e_i \in \bar{e} : (e_i, f_j) \in A \rightarrow f_j \in \bar{f}$$
$$\text{AND } \forall f_j \in \bar{f} : (e_i, f_j) \in A \rightarrow e_i \in \bar{e}$$
$$\text{AND } \exists e_i \in \bar{e}, f_j \in \bar{f} : (e_i, f_j) \in A$$

- Let P be the set of all extracted phrase pairs $(\bar{e}, \bar{f})$

Formal Definition

- Extend recursively:

$$\text{if } (\bar{e}, \bar{f}) \in P \text{ AND } (\bar{e}_{\text{SUB}}, \bar{f}_{\text{SUB}}) \in P$$
$$\text{AND } \bar{e} = \bar{e}_{\text{PRE}} + \bar{e}_{\text{SUB}} + \bar{e}_{\text{POST}}$$
$$\text{AND } \bar{f} = \bar{f}_{\text{PRE}} + \bar{f}_{\text{SUB}} + \bar{f}_{\text{POST}}$$
$$\text{AND } \bar{e} \neq \bar{e}_{\text{SUB}} \text{ AND } \bar{f} \neq \bar{f}_{\text{SUB}}$$
$$\text{add } (e_{\text{PRE}} + \text{X} + e_{\text{POST}}, f_{\text{PRE}} + \text{X} + f_{\text{POST}}) \text{ to } P$$

(note: any of e_{PRE}, e_{POST}, f_{PRE}, or f_{POST} may be empty)

- Set of hierarchical phrase pairs is the closure under this extension mechanism

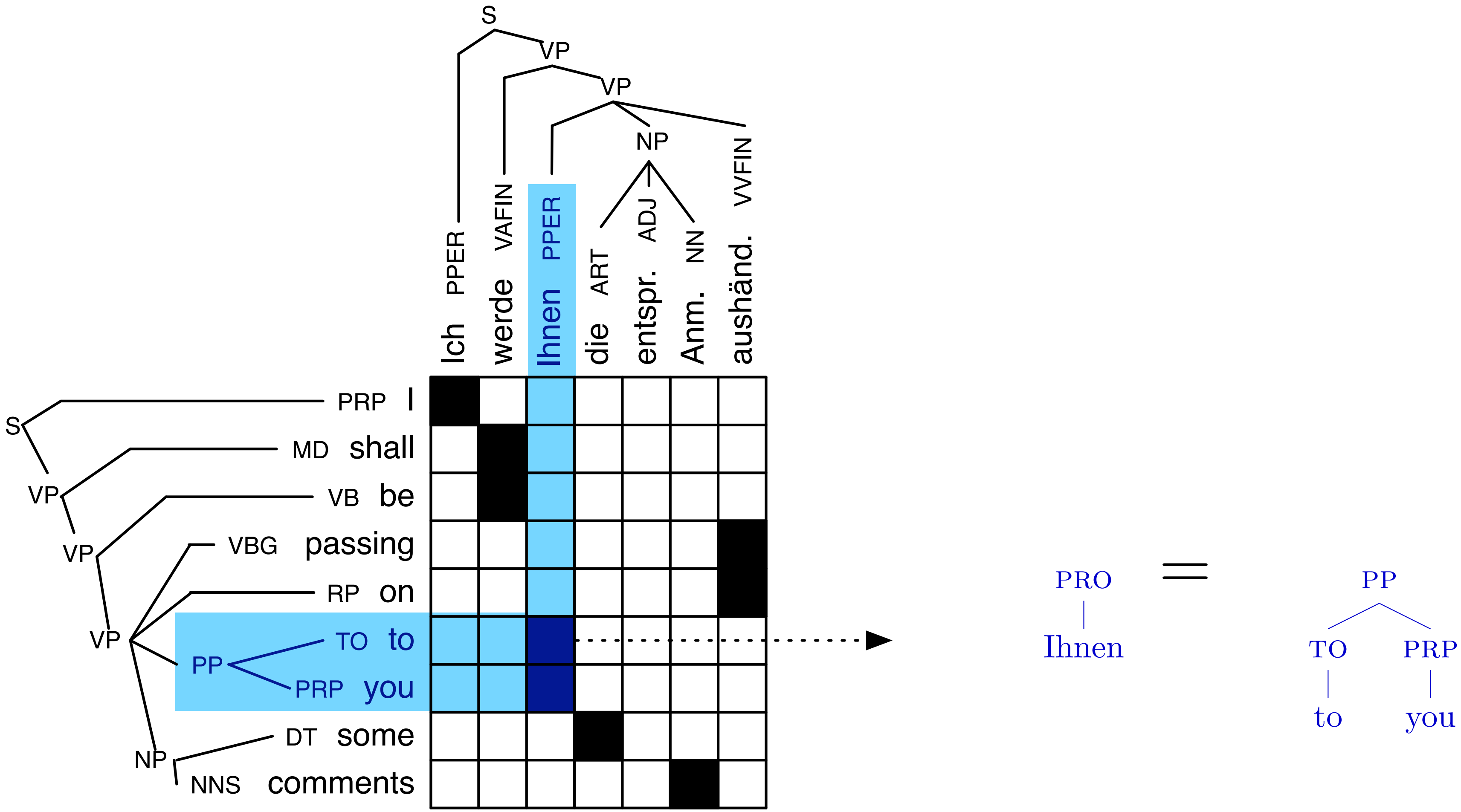
Comments

- Removal of multiple sub-phrases leads to rules with multiple non-terminals, such as:

$$\mathrm{Y} \rightarrow \mathrm{X}_1 \ \mathrm{X}_2 \ \ | \ \ \mathrm{X}_2 \ \textit{of} \ \mathrm{X}_1$$

- Typical restrictions to limit complexity [Chiang, 2005]
  - at most 2 nonterminal symbols
  - at least 1 but at most 5 words per language
  - span at most 15 words (counting gaps)

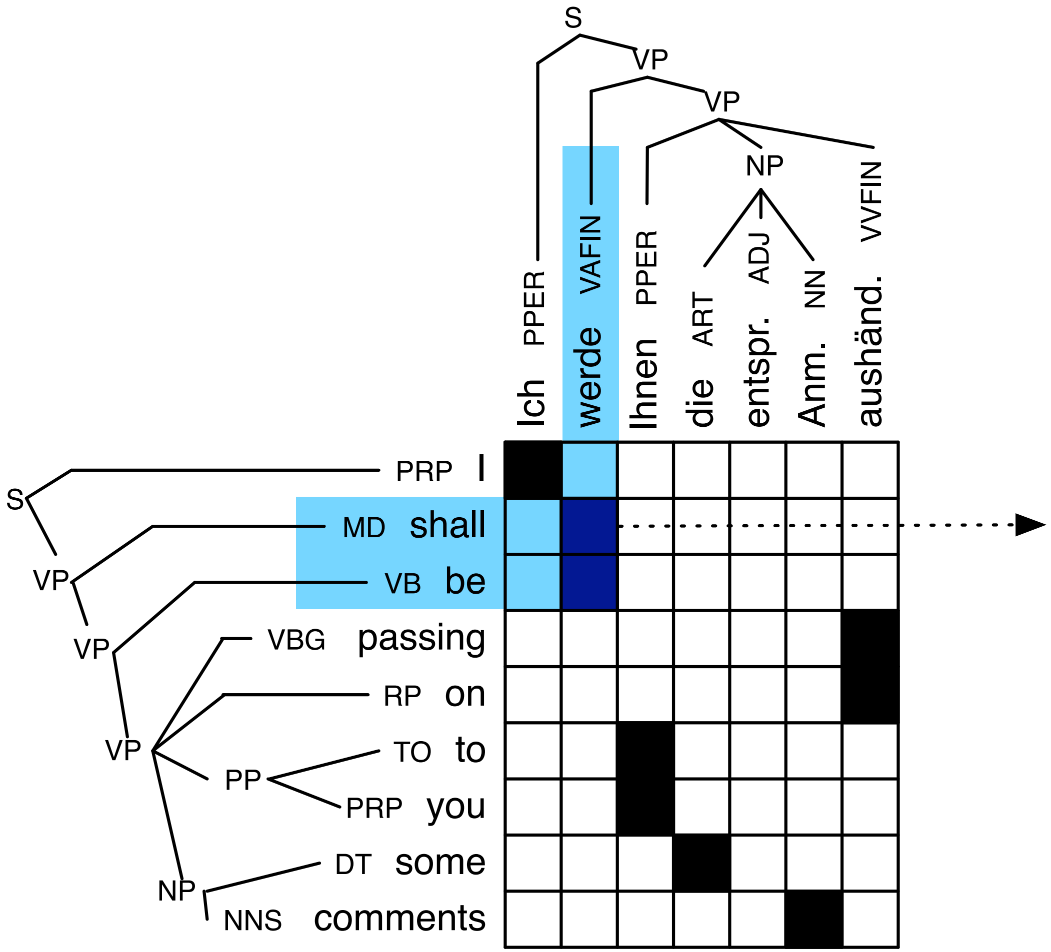
Learning Syntactic Translation Rules



Constraints on Syntactic Rules

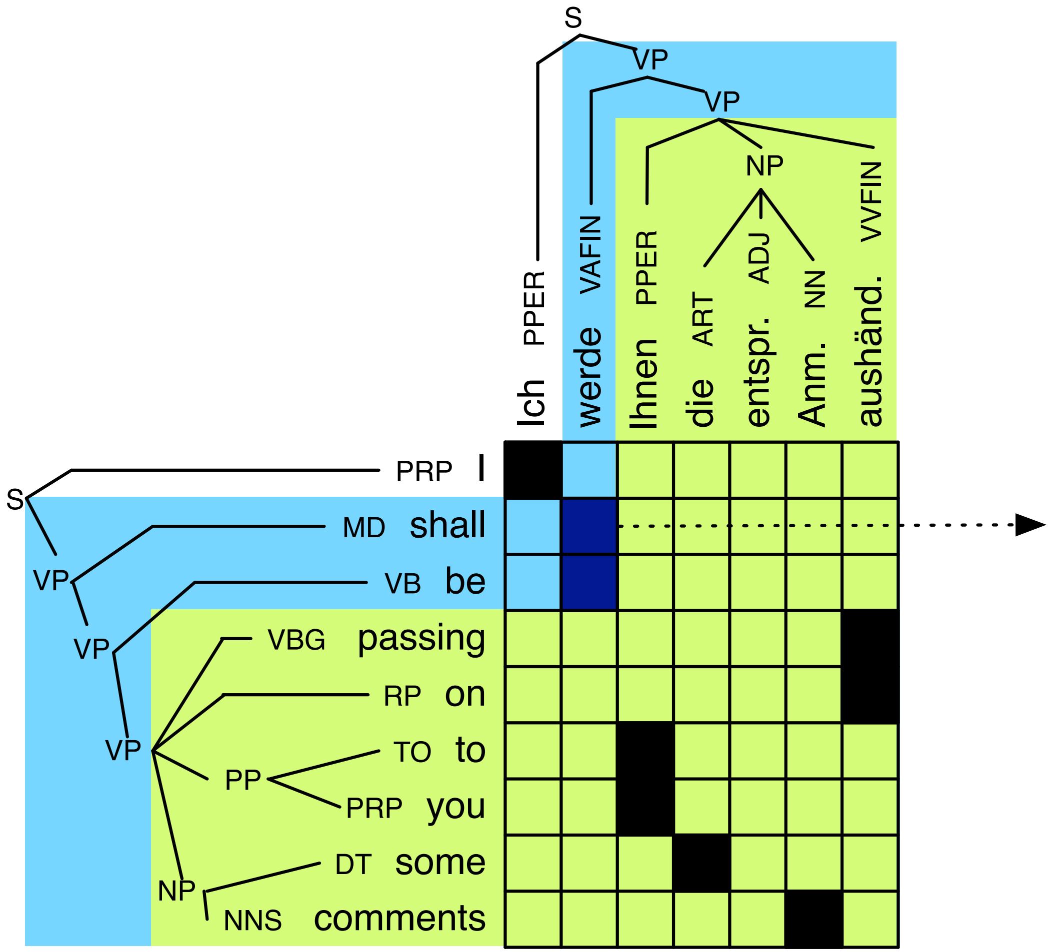
- Same word alignment constraints as hierarchical models
- Hierarchical: rule can cover any span
  $\Leftrightarrow$ syntactic rules must cover constituents in the tree
- Hierarchical: gaps may cover any span
  $\Leftrightarrow$ gaps must cover constituents in the tree
- Much less rules are extracted (all things being equal)

Impossible Rules

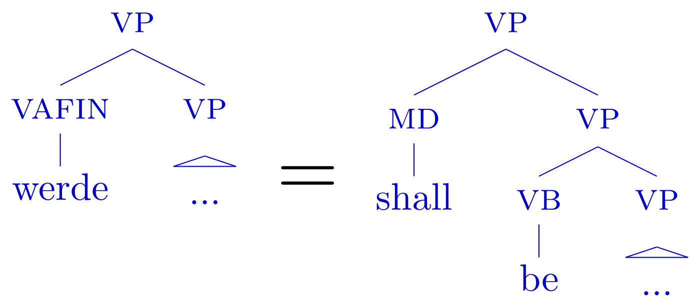


English span not a constituent
no rule extracted

Rules with Context



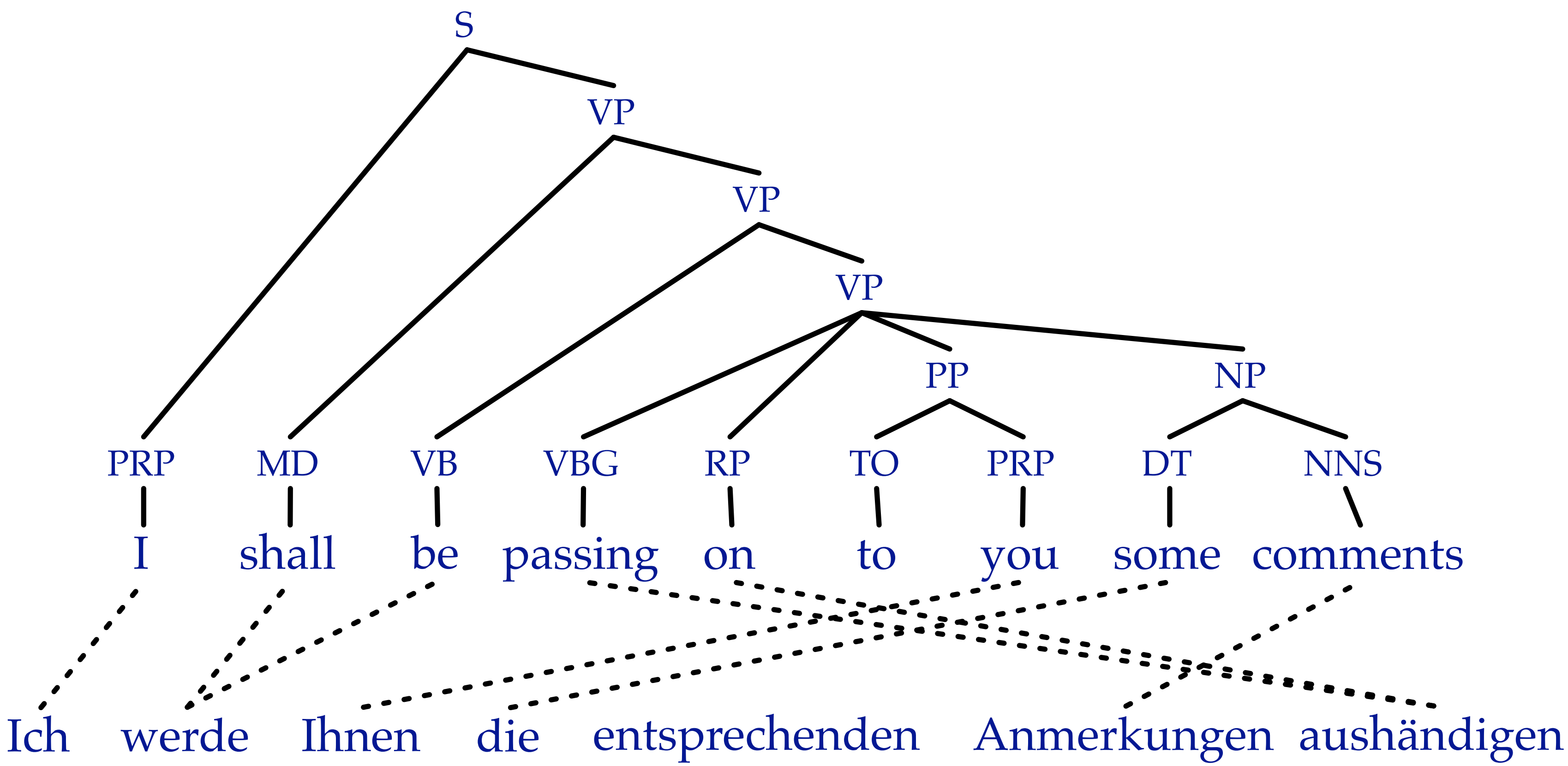
Rule with this phrase pair requires syntactic context



Too Many Rules Extractable

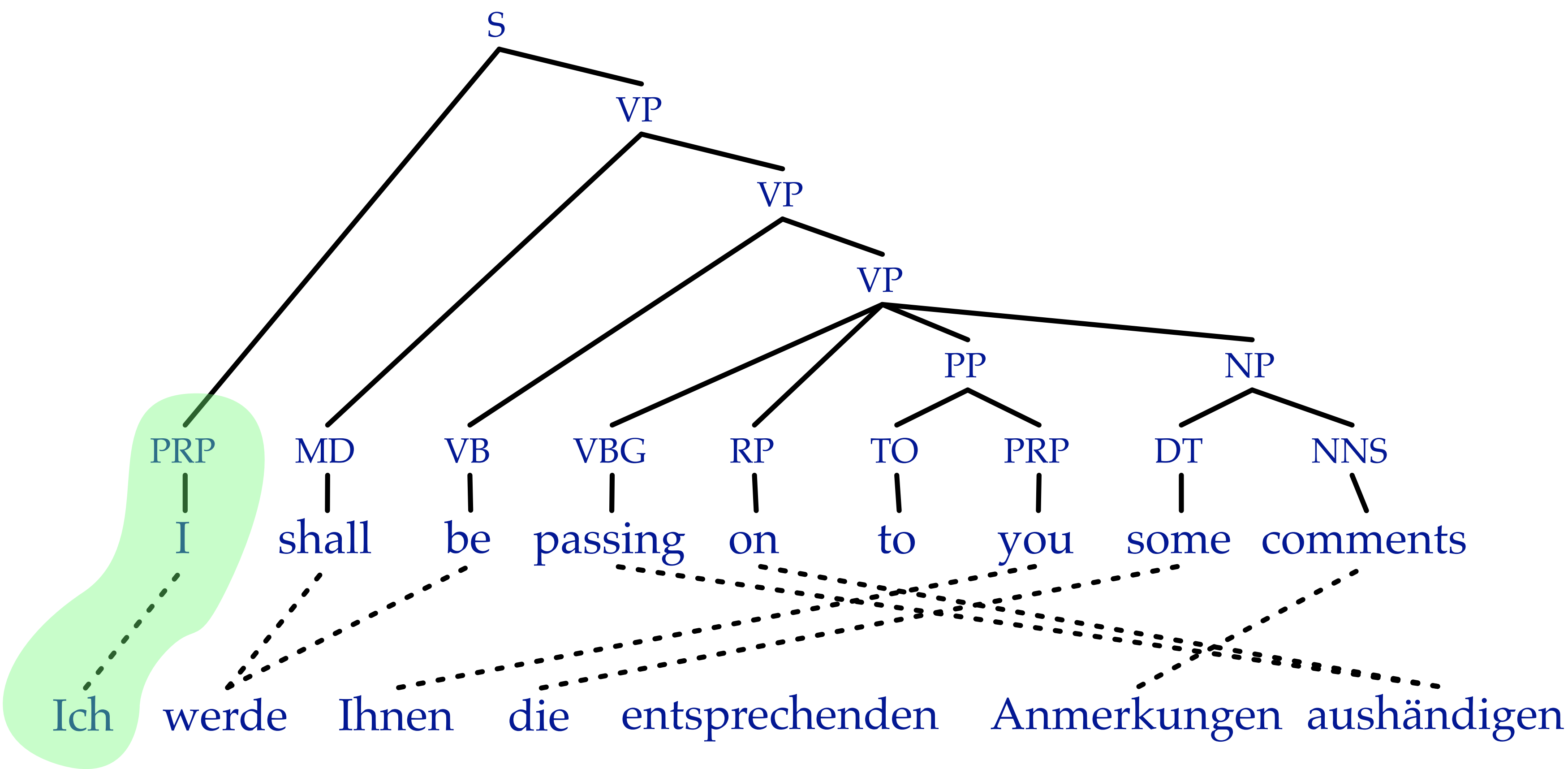
- Huge number of rules can be extracted
  (every alignable node may or may not be part of a rule → exponential number of rules)
- Need to limit which rules to extract
- Option 1: similar restriction as for hierarchical model
  (maximum span size, maximum number of terminals and non-terminals, etc.)
- Option 2: only extract minimal rules ("GHKM" rules)

Minimal Rules



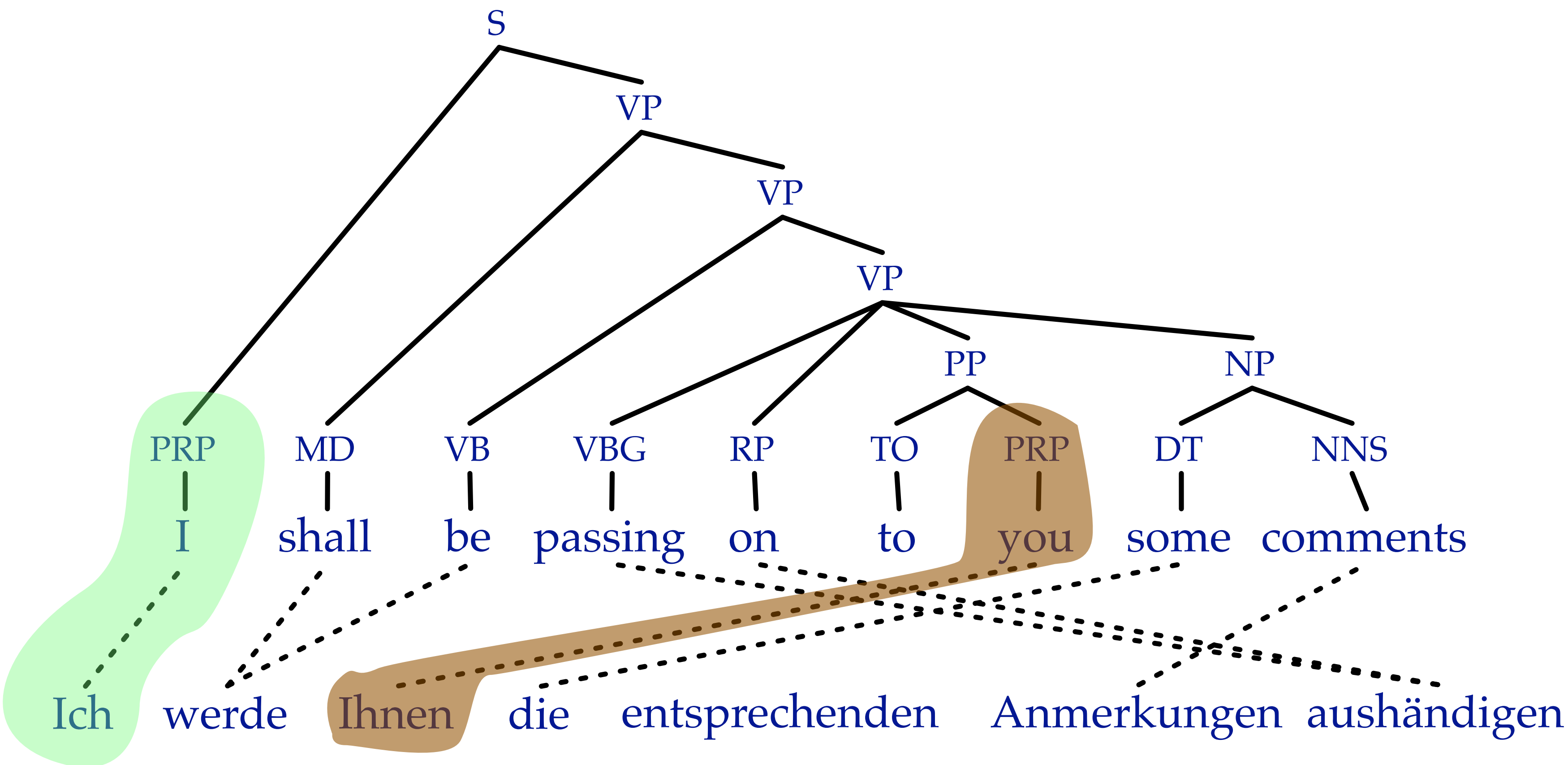
Extract: set of smallest rules required to explain the sentence pair

Lexical Rule



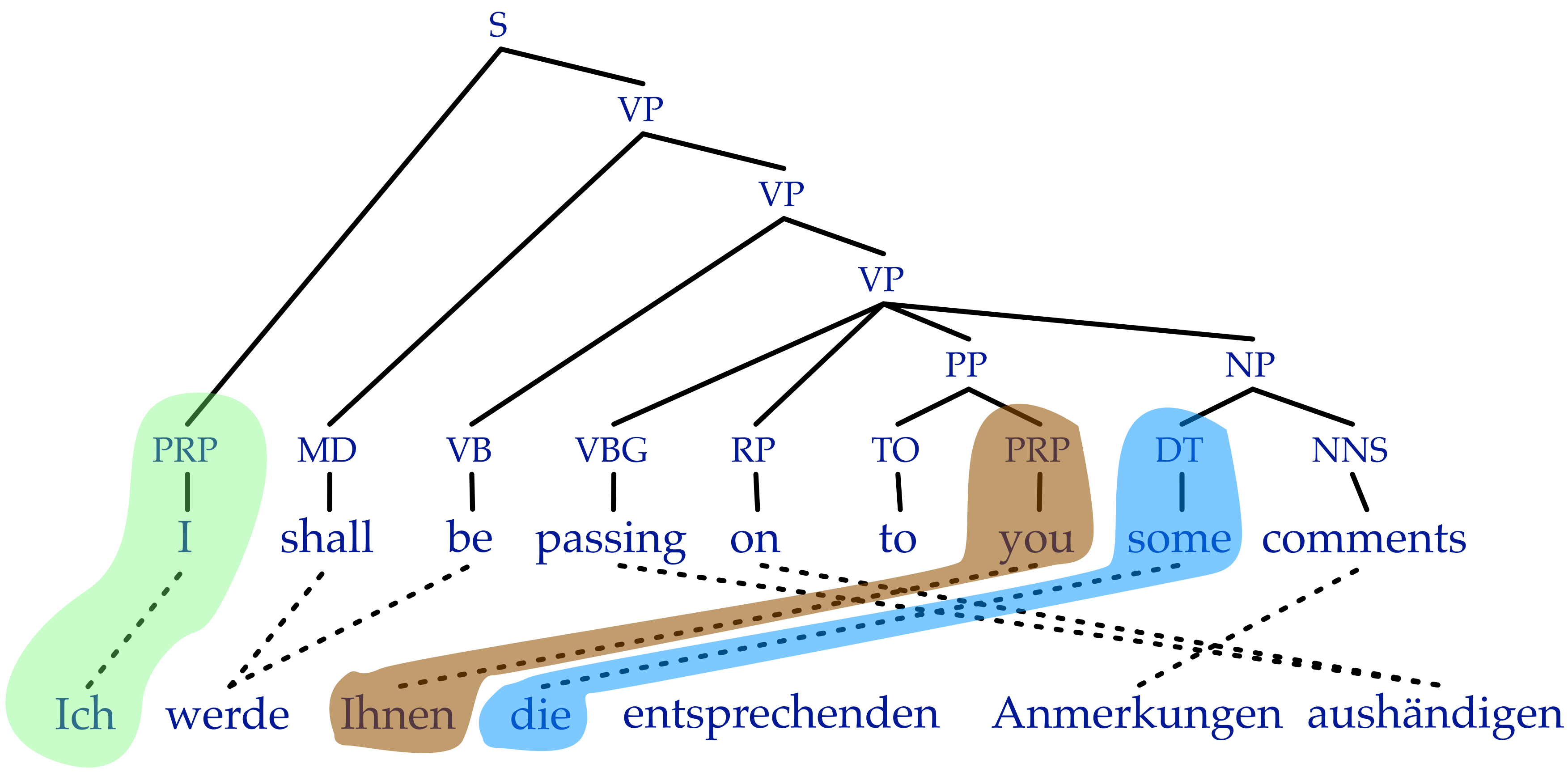
Extracted rule: PRP → Ich | I

Lexical Rule



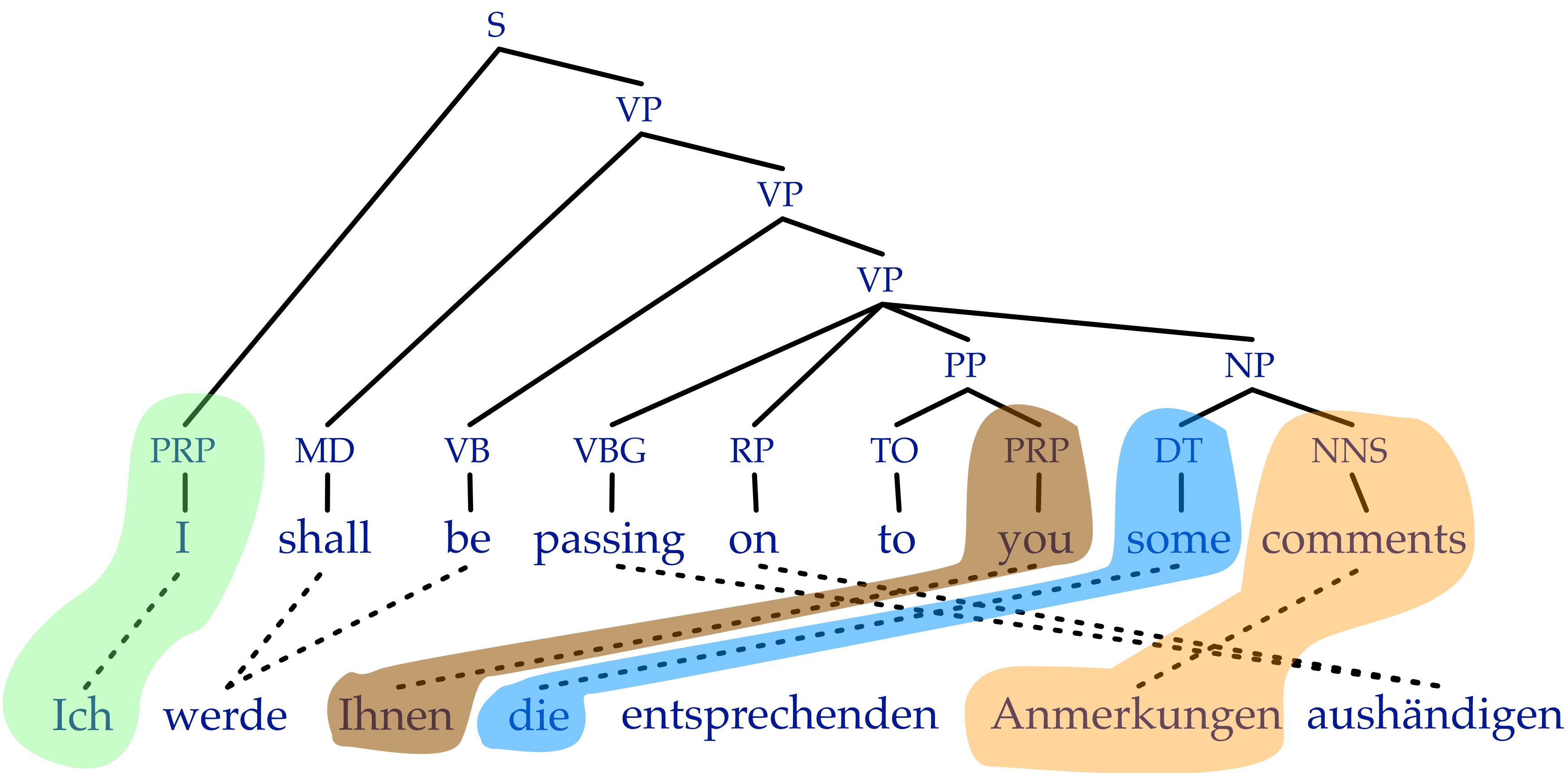
Extracted rule: PRP → Ihnen | you

Lexical Rule



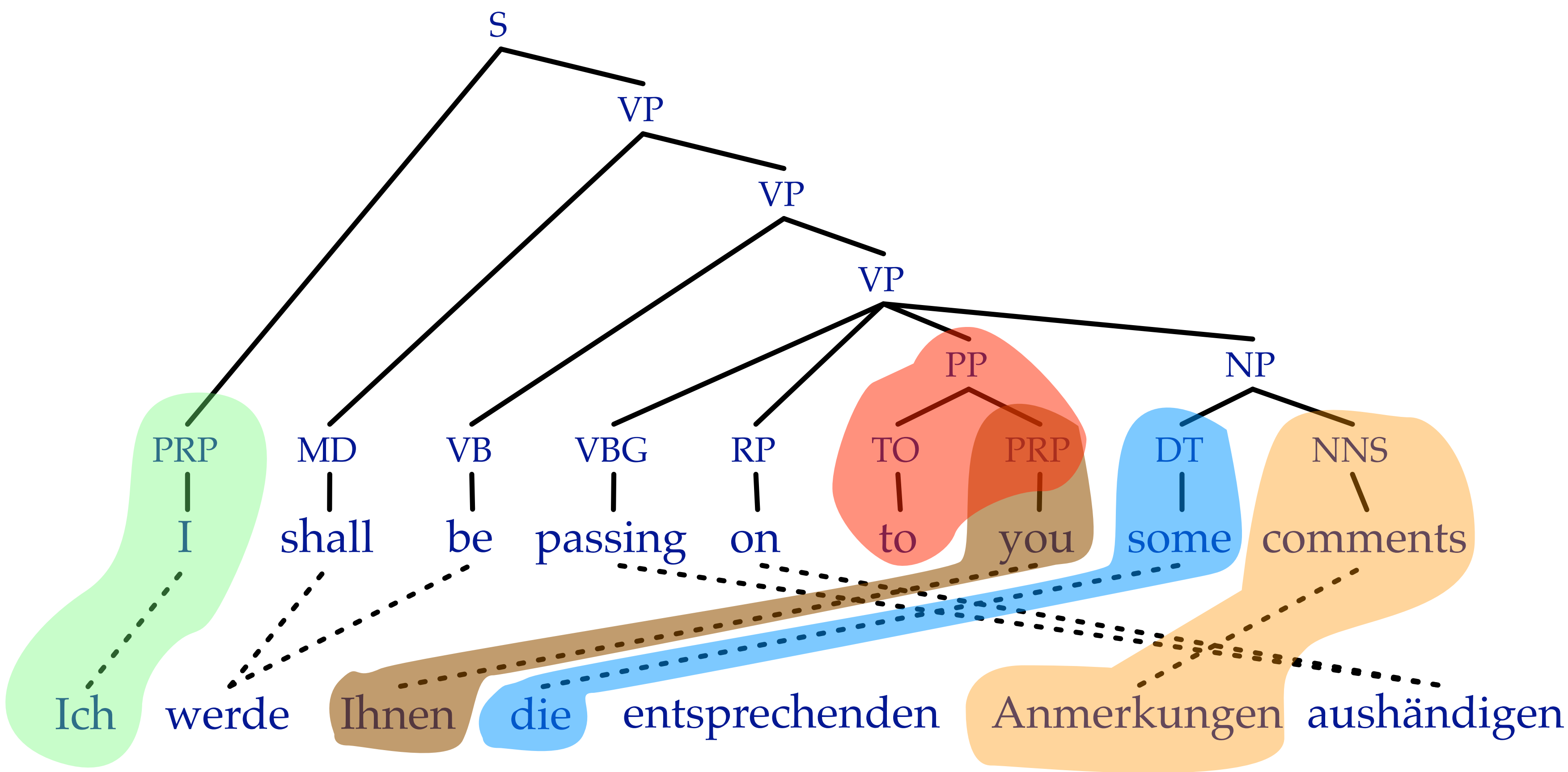
Extracted rule: $\text{DT} \rightarrow \text{die} \mid \text{some}$

Lexical Rule



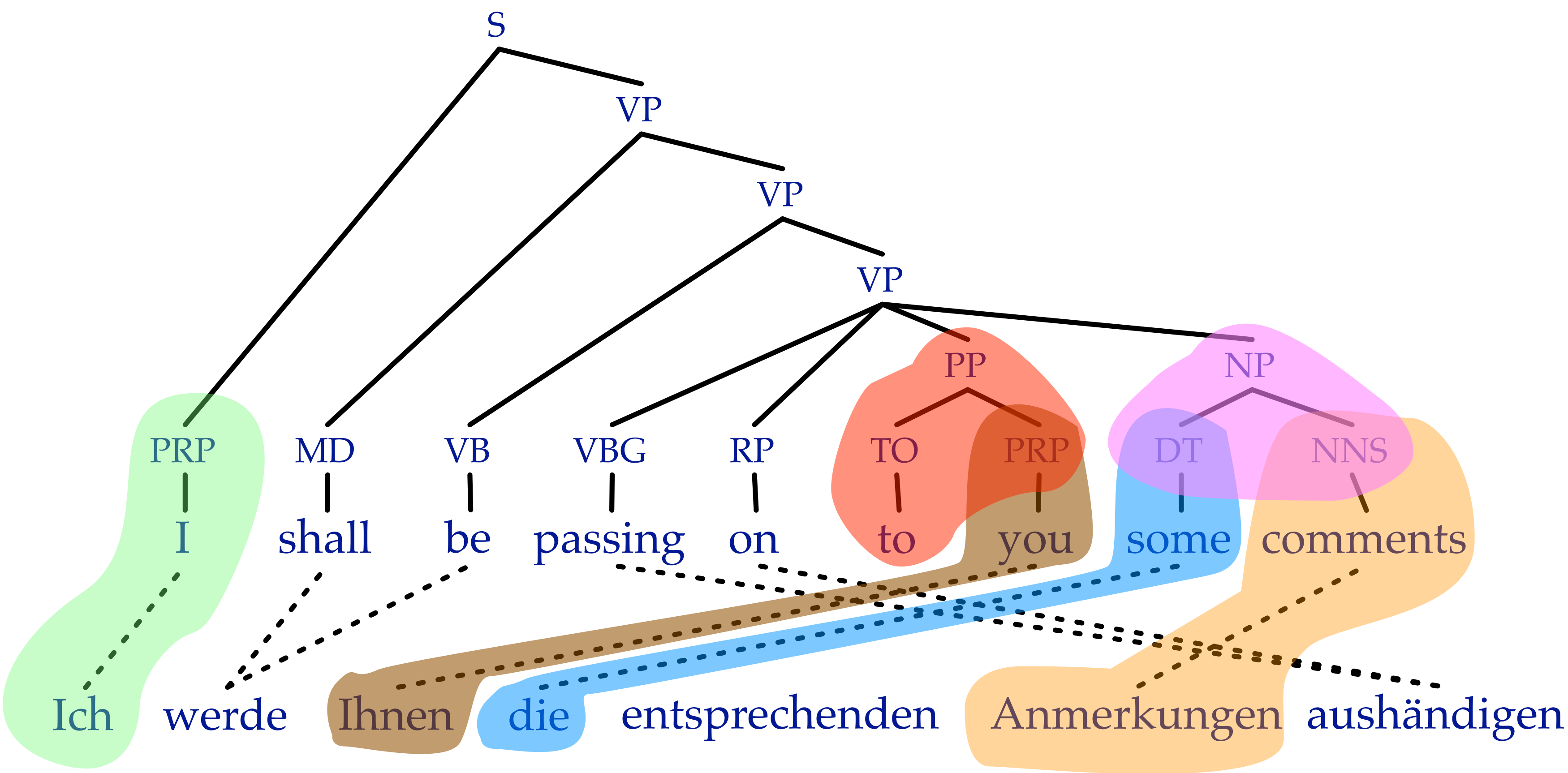
Extracted rule: NNS → Anmerkungen | comments

Insertion Rule



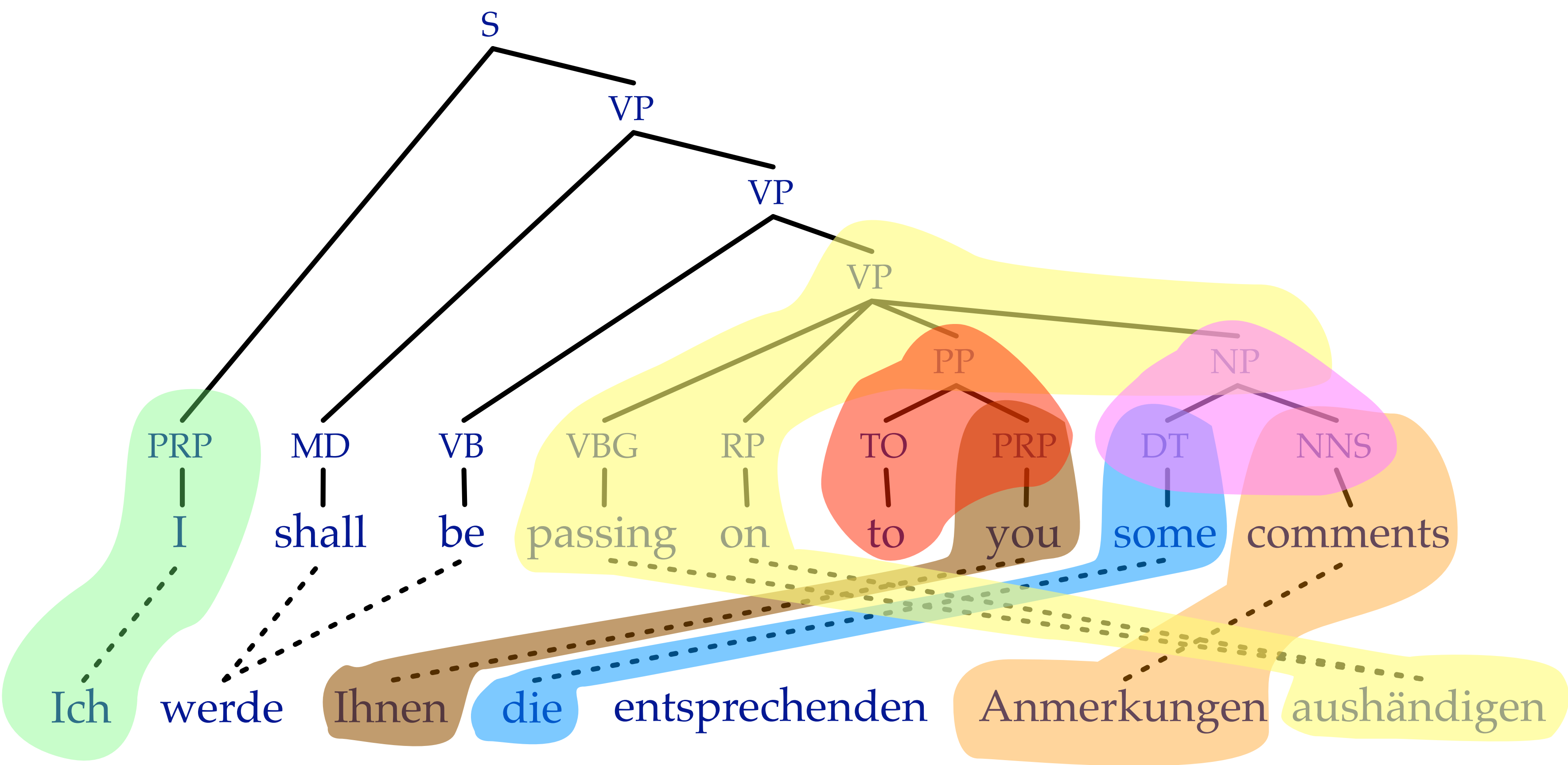
Extracted rule: PP → X | to PRP

Non-Lexical Rule



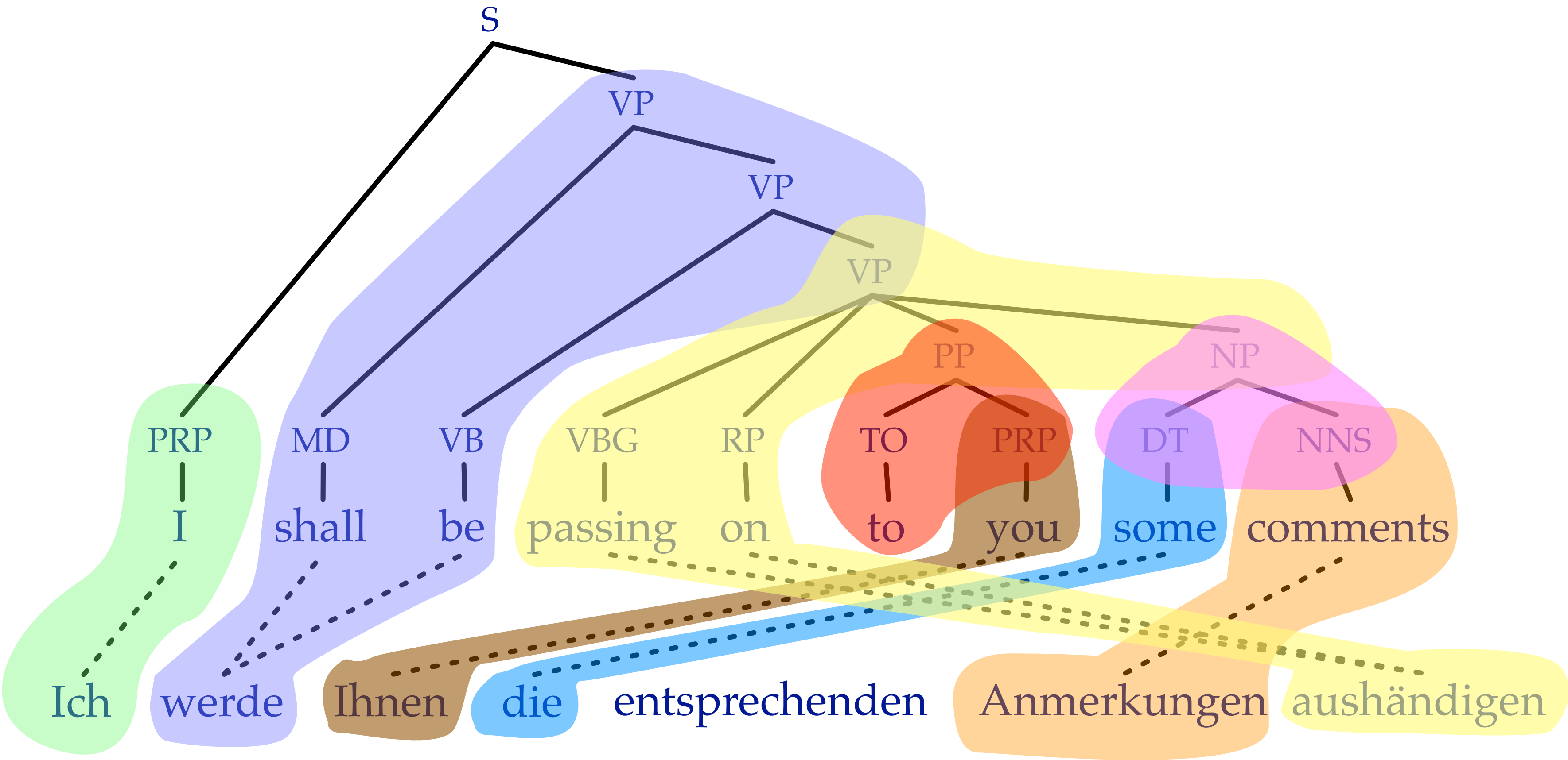
Extracted rule: $\mathrm{NP} \rightarrow \mathrm{X}_1\ \mathrm{X}_2 \mid \mathrm{DT}_1\ \mathrm{NNS}_2$

Lexical Rule with Syntactic Context



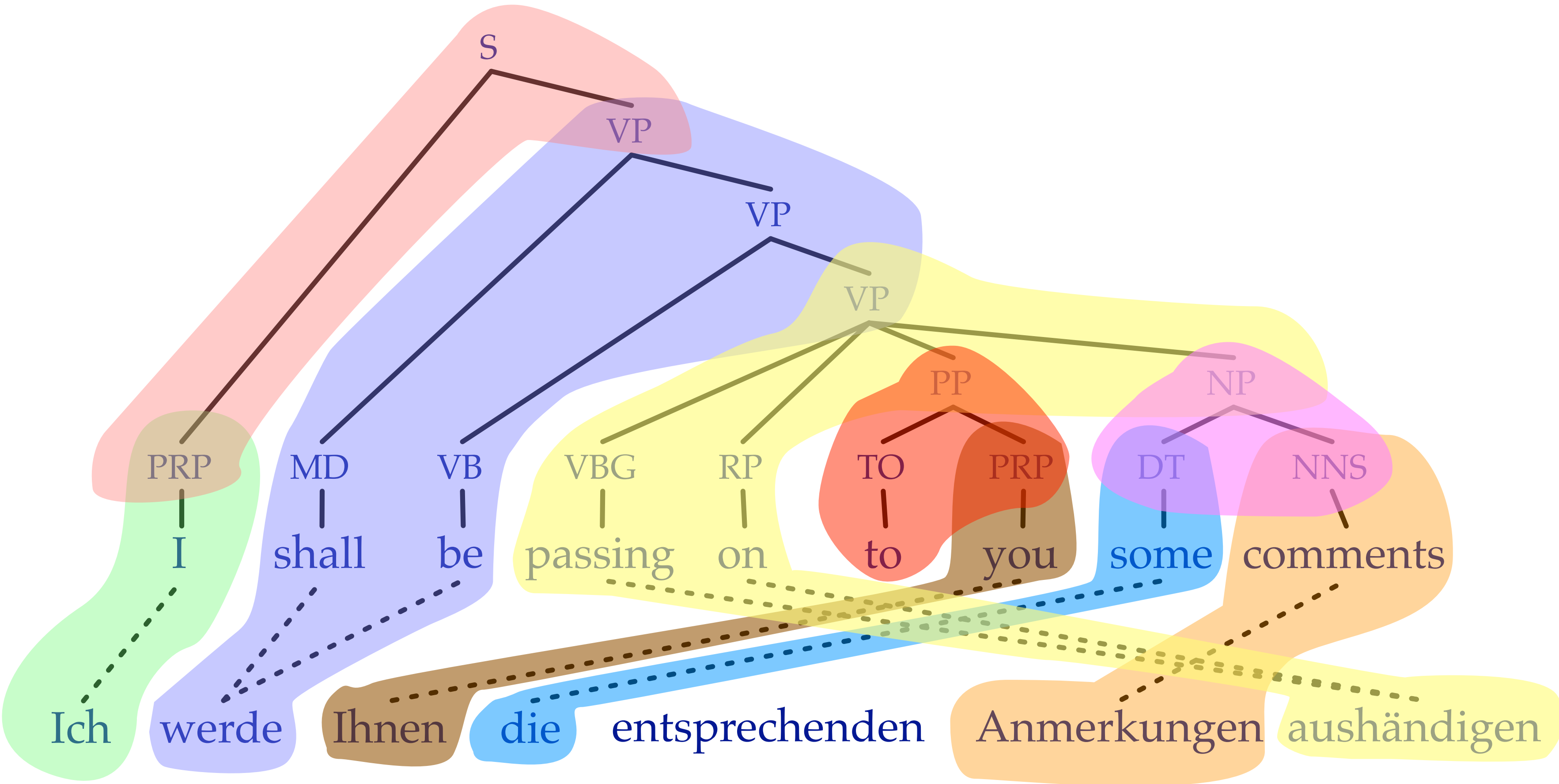
Extracted rule: $\text{VP} \rightarrow \text{X}_1\ \text{X}_2$ aushändigen | passing on $\text{PP}_1\ \text{NP}_2$

Lexical Rule with Syntactic Context



Extracted rule: VP → werde X | shall be VP (ignoring internal structure)

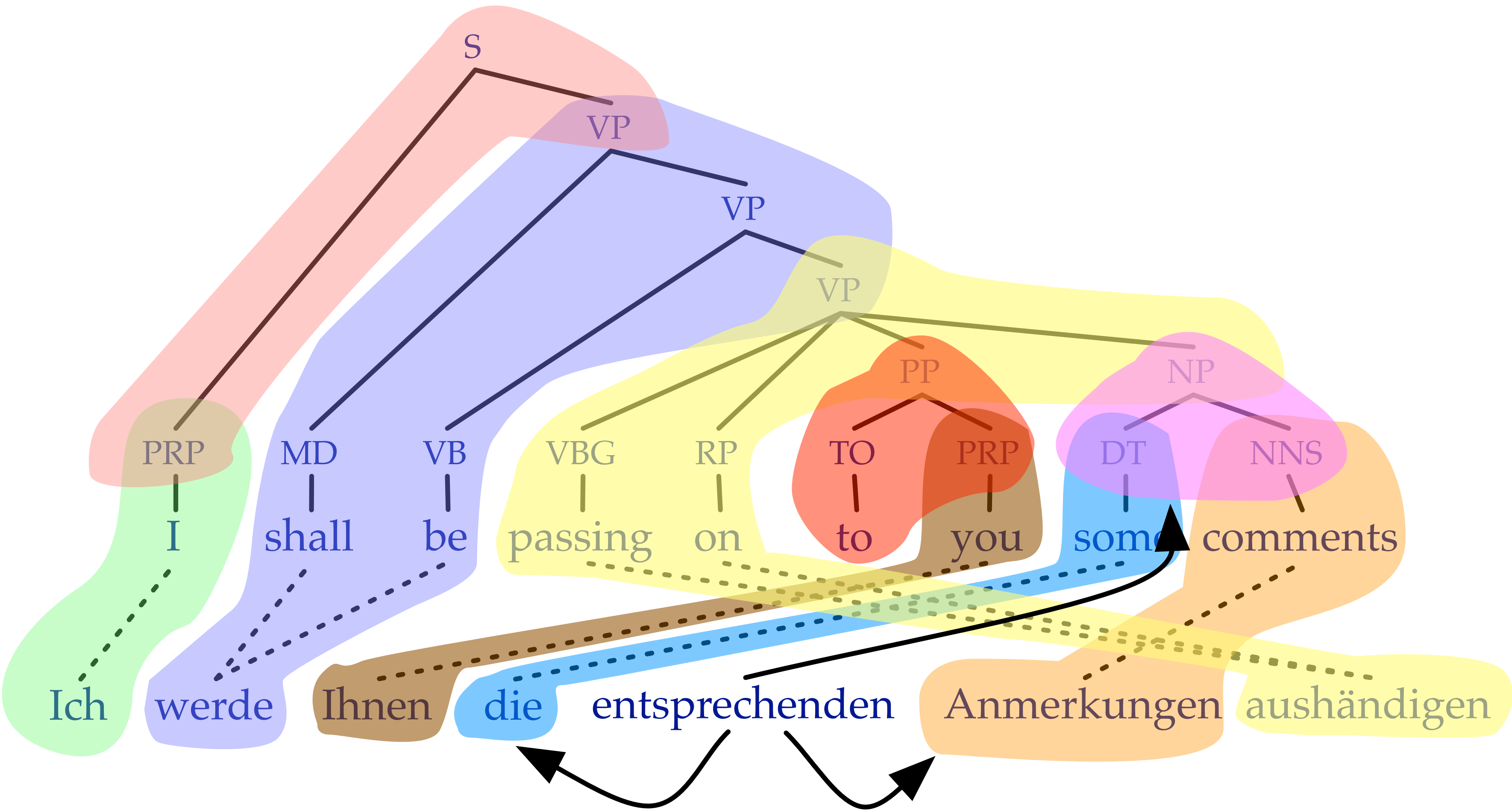
Non-Lexical Rule



Extracted rule: $\text{S} \rightarrow \text{X}_1\ \text{X}_2 \mid \text{PRP}_1\ \text{VP}_2$

DONE — note: one rule per alignable constituent

Unaligned Source Words



Attach to neighboring words or higher nodes → additional rules

Too Few Phrasal Rules?

- Lexical rules will be 1-to-1 mappings (unless word alignment requires otherwise)
- But: phrasal rules very beneficial in phrase-based models
- Solutions
  - combine rules that contain a maximum number of symbols
    (as in hierarchical models, recall: "Option 1")
  - compose minimal rules to cover a maximum number of non-leaf nodes

Composed Rules

- Current rules

$$X_1\ X_2 \;=\; \text{NP} \to \text{DT}_1\ \text{NNS}_1$$

```
X1 X2  =    NP
           /  \
        DT1   NNS1
```

```
die  =  DT          entsprechenden Anmerkungen  =  NNS
        |                                          |
       some                                     comments
```

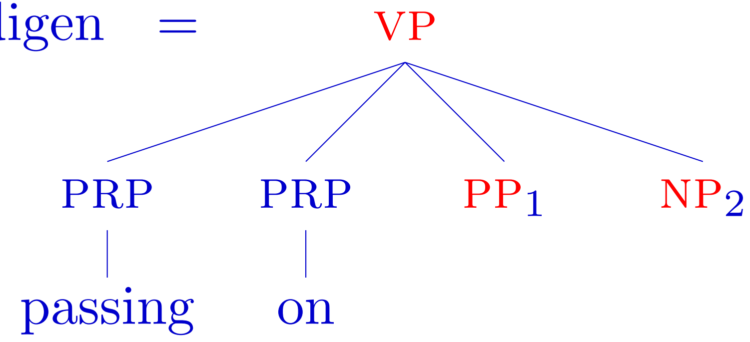
- Composed rule

```
die entsprechenden Anmerkungen  =      NP
                                      /  \
                                    DT    NNS
                                    |      |
                                   some  comments
```

(1 non-leaf node: NP)

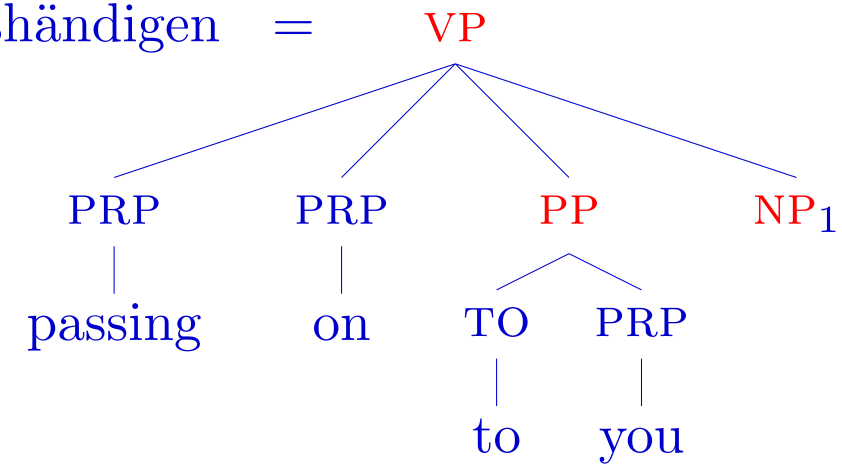
Composed Rules

- Minimal rule: x_1 x_2 aushändigen =



3 non-leaf nodes:
VP, PP, NP

- Composed rule: Ihnen x_1 aushändigen =



3 non-leaf nodes:
VP, PP and NP

Relaxing Tree Constraints

- Impossible rule

$$\underset{\text{werde}}{\text{X}} \quad = \quad \underset{\text{shall}}{\text{MD}} \quad \underset{\text{be}}{\text{VB}}$$

- Create new non-terminal label: MD+VB

⇒ New rule

$$\underset{\text{werde}}{\text{X}} \quad = \quad \text{MD+VB}\left(\underset{\text{shall}}{\text{MD}} \quad \underset{\text{be}}{\text{VB}}\right)$$

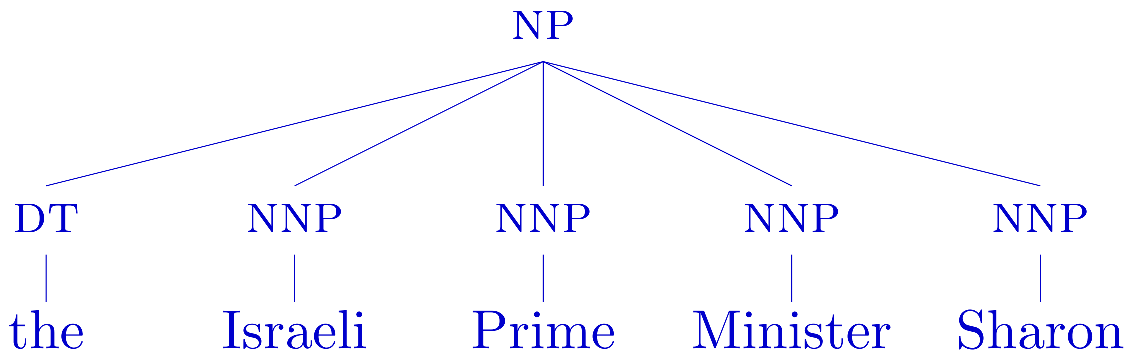
Zollmann Venugopal Relaxation

- If span consists of two constituents , join them: X+Y
- If span conststs of three constituents, join them: X+Y+Z
- If span covers constituents with the same parent X and include
  - every but the first child Y, label as X\Y
  - every but the last child Y, label as X/Y
- For all other cases, label as FAIL

⇒ More rules can be extracted, but number of non-terminals blows up

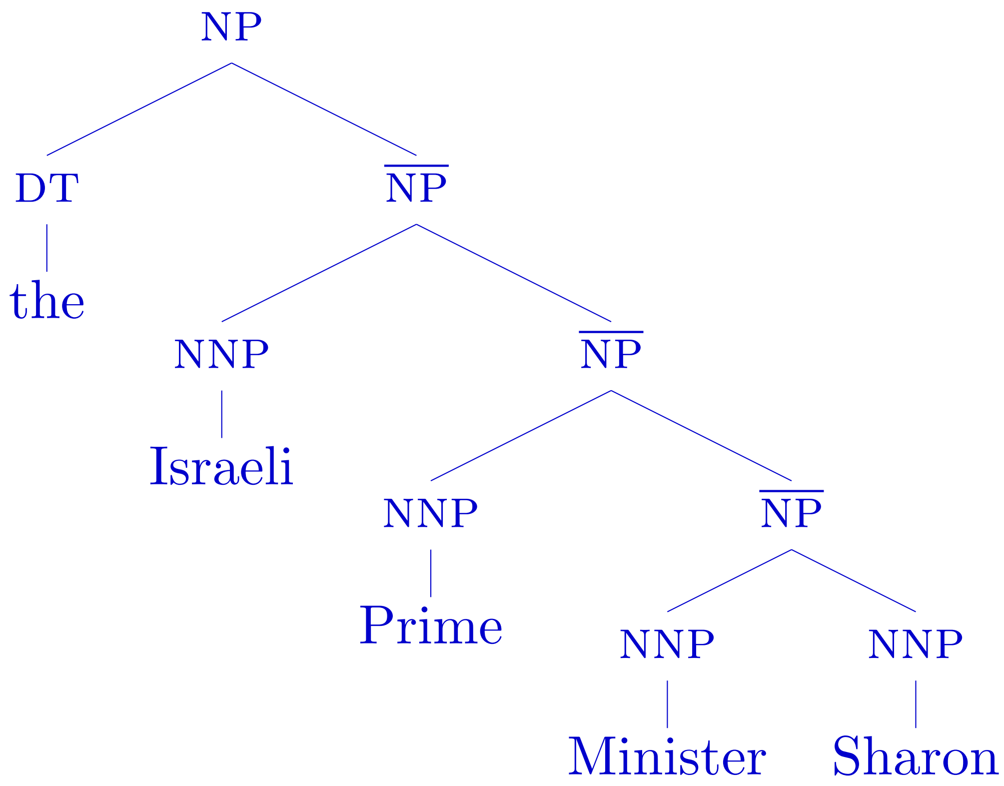
Special Problem: Flat Structures

- Flat structures severely limit rule extraction



- Can only extract rules for individual words or entire phrase

Relaxation by Tree Binarization



More rules can be extracted

Left-binarization or right-binarization?

Scoring Translation Rules

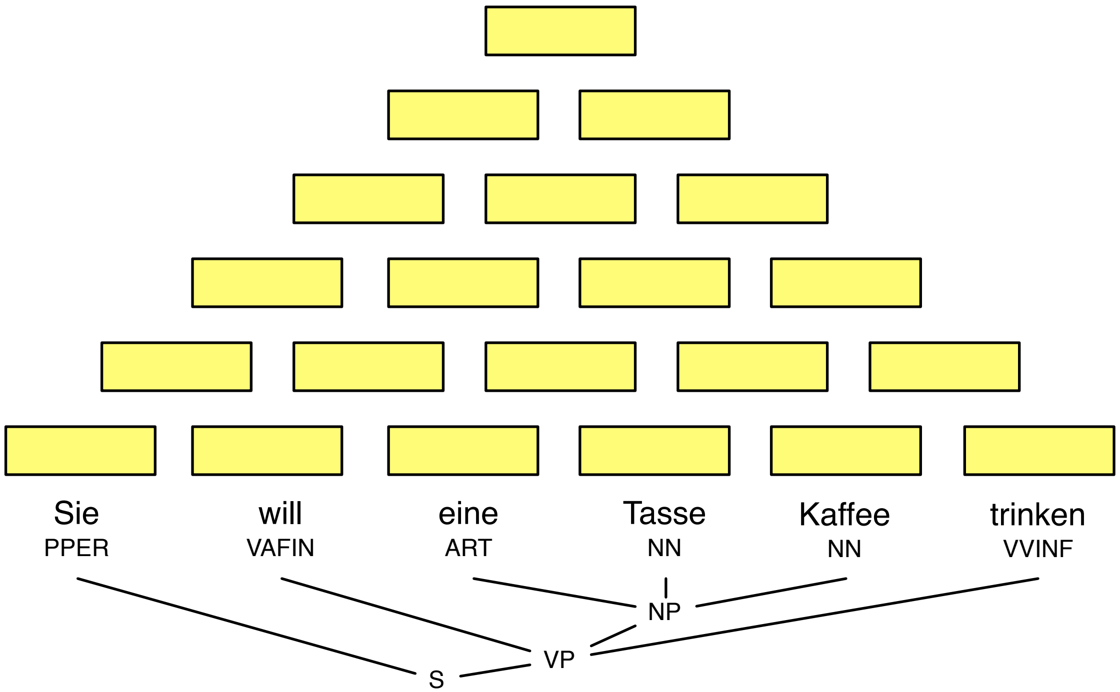
- Extract all rules from corpus

- Score based on counts
  - joint rule probability: $p(\text{LHS}, \text{RHS}_f, \text{RHS}_e)$
  - rule application probability: $p(\text{RHS}_f, \text{RHS}_e|\text{LHS})$
  - direct translation probability: $p(\text{RHS}_e|\text{RHS}_f, \text{LHS})$
  - noisy channel translation probability: $p(\text{RHS}_f|\text{RHS}_e, \text{LHS})$
  - lexical translation probability: $\prod_{e_i \in \text{RHS}_e} p(e_i|\text{RHS}_f, a)$

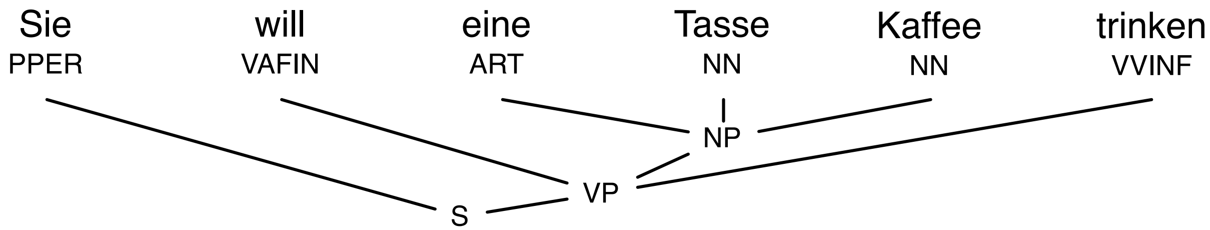
Syntactic Decoding

Inspired by monolingual syntactic chart parsing:

During decoding of the source sentence,
a chart with translations for the $O(n^2)$ spans has to be filled

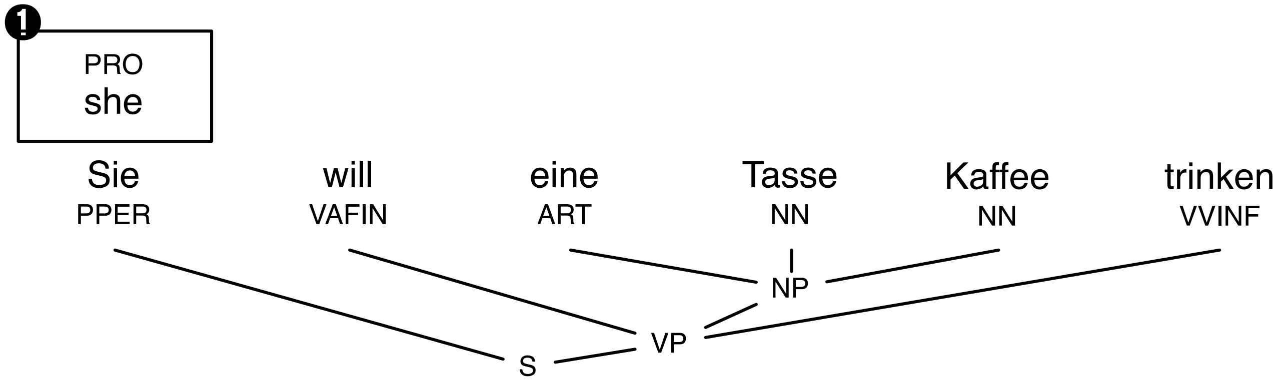


Syntax Decoding



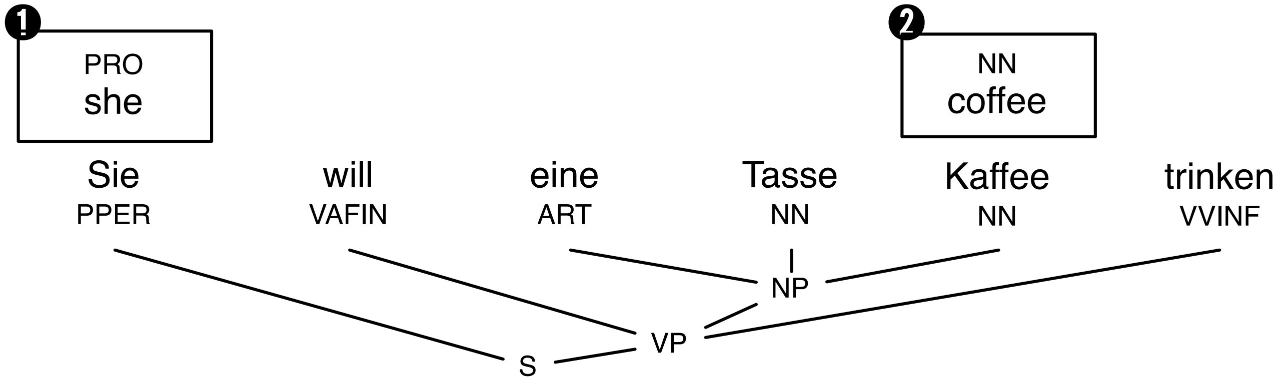
German input sentence with tree

Syntax Decoding



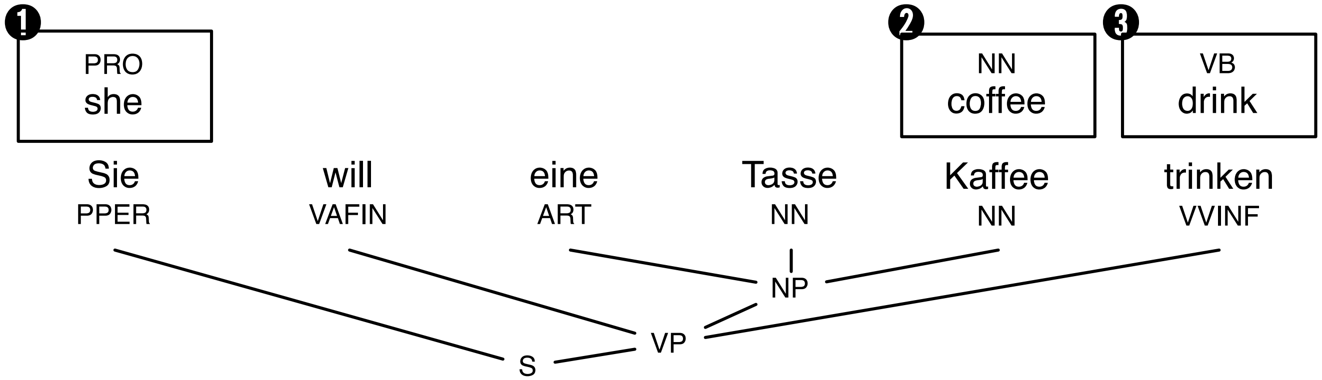
Purely lexical rule: filling a span with a translation (a constituent in the chart)

Syntax Decoding



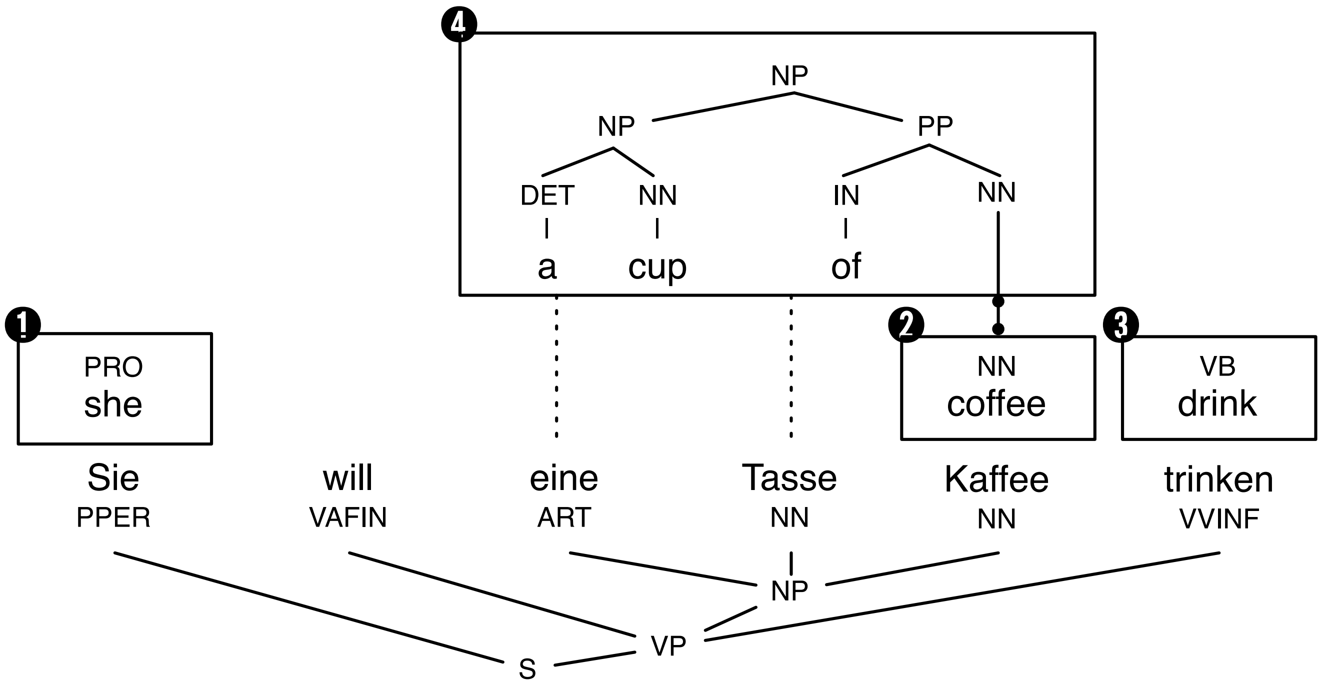
Purely lexical rule: filling a span with a translation (a constituent in the chart)

Syntax Decoding



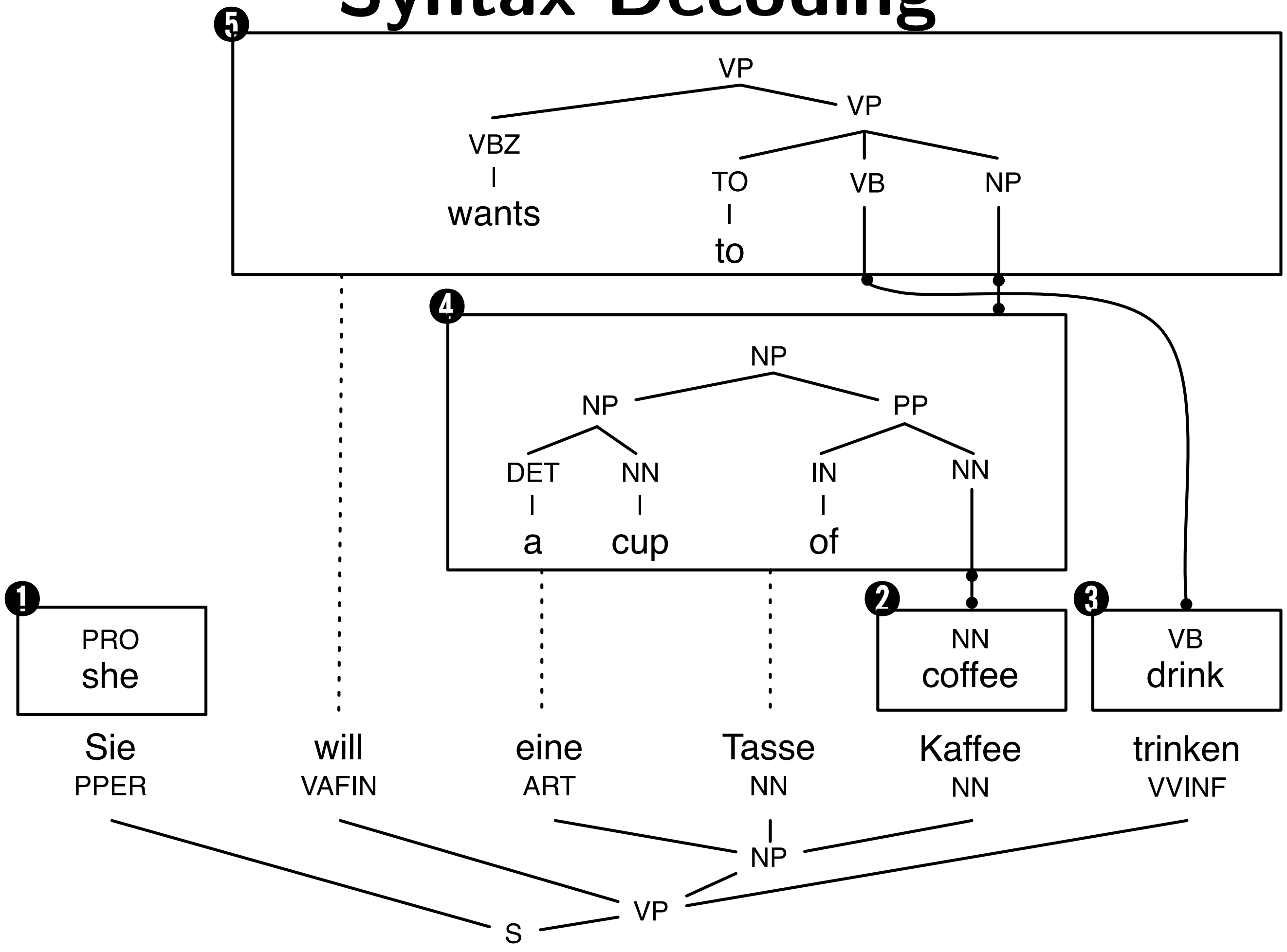
Purely lexical rule: filling a span with a translation (a constituent in the chart)

Syntax Decoding

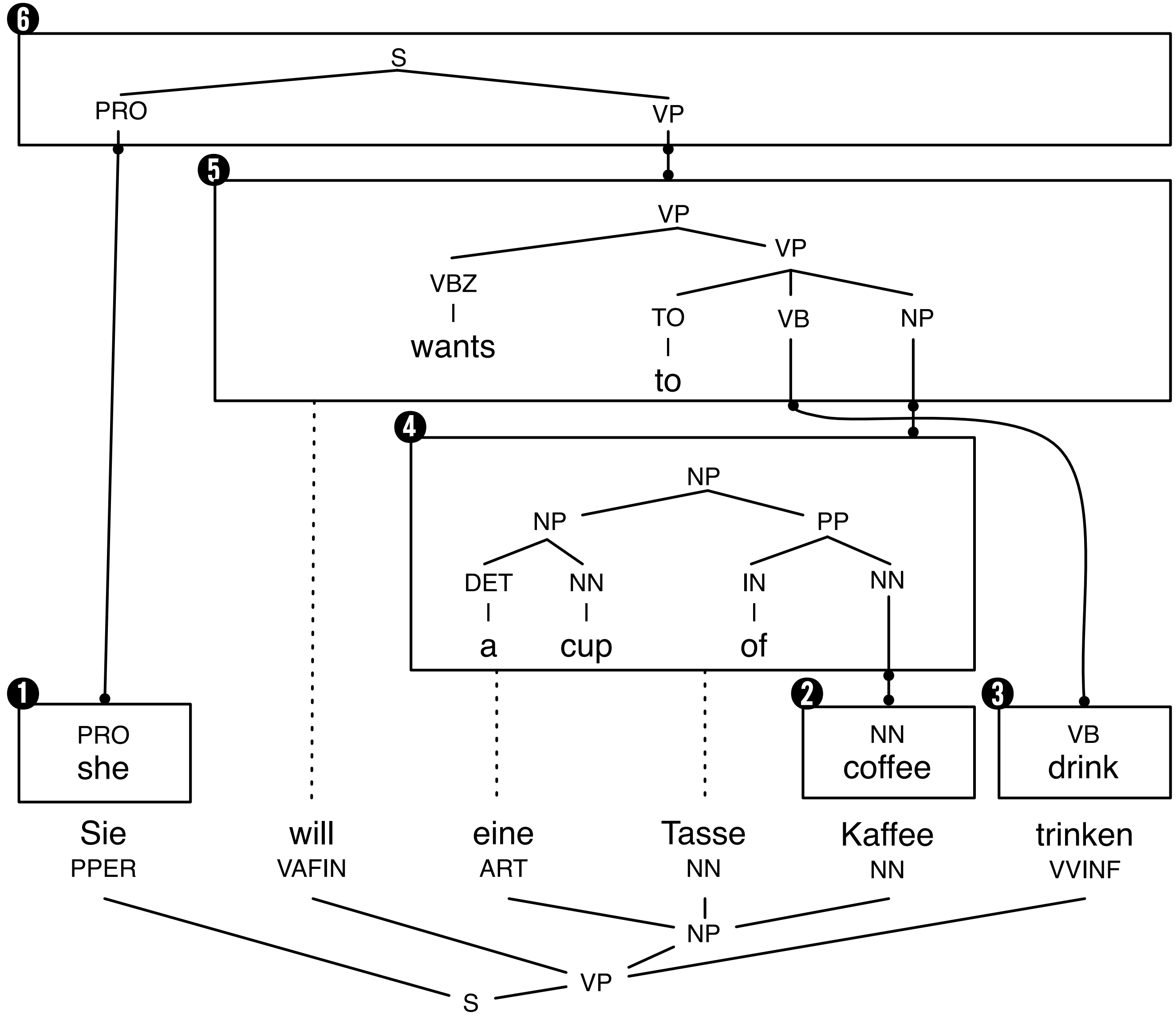


Complex rule: matching underlying constituent spans, and covering words

Syntax Decoding

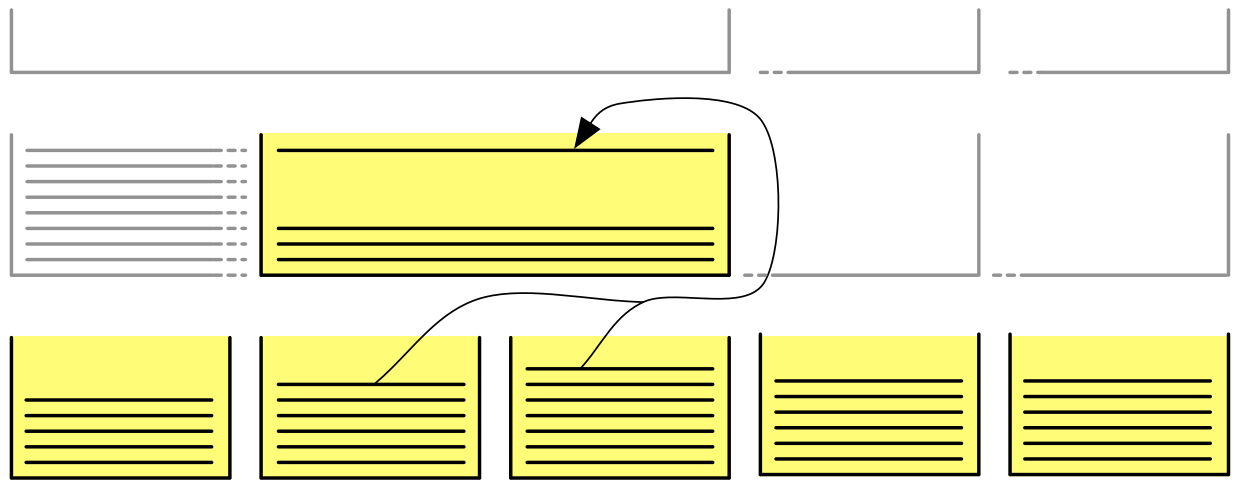


Complex rule with reordering


6
S
PRO
VP
5
VP
VP
VBZ
wants
TO
to
VB
NP
4
NP
NP
PP
DET
a
NN
cup
IN
of
NN
1
PRO
she
2
NN
coffee
3
VB
drink
Sie
PPER
will
VAFIN
eine
ART
Tasse
NN
Kaffee
NN
trinken
VVINF
NP
VP
S

Bottom-Up Decoding

- For each span, a stack of (partial) translations is maintained
- Bottom-up: a higher stack is filled, once underlying stacks are complete

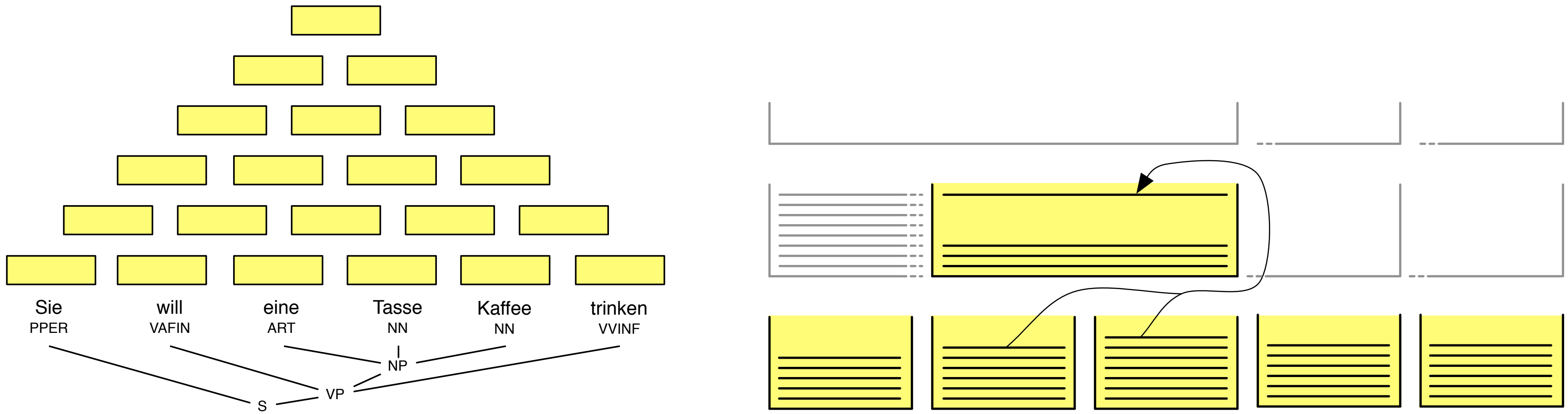


Naive Algorithm

Input: Foreign sentence $\mathbf{f} = f_1, ...f_{l_f}$, with syntax tree
Output: English translation **e**

```
for all spans [start,end] (bottom up) do
    for all sequences s of hypotheses and words in span [start,end] do
        for all rules r do
            if rule r applies to chart sequence s then
                create new hypothesis c
                add hypothesis c to chart
            end if
        end for
    end for
end for
return English translation e from best hypothesis in span [0,l_f]
```

Chart Organization

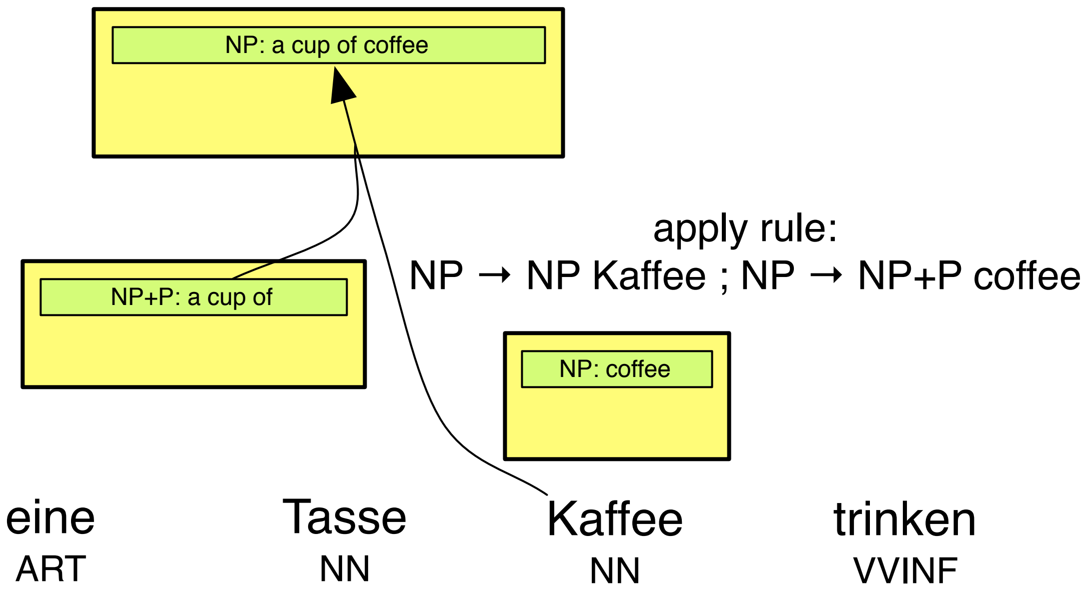


- Chart consists of cells that cover contiguous spans over the input sentence
- Each cell contains a set of hypotheses[1]
- Hypothesis = translation of span with target-side constituent

[1] In the book, they are called chart entries.

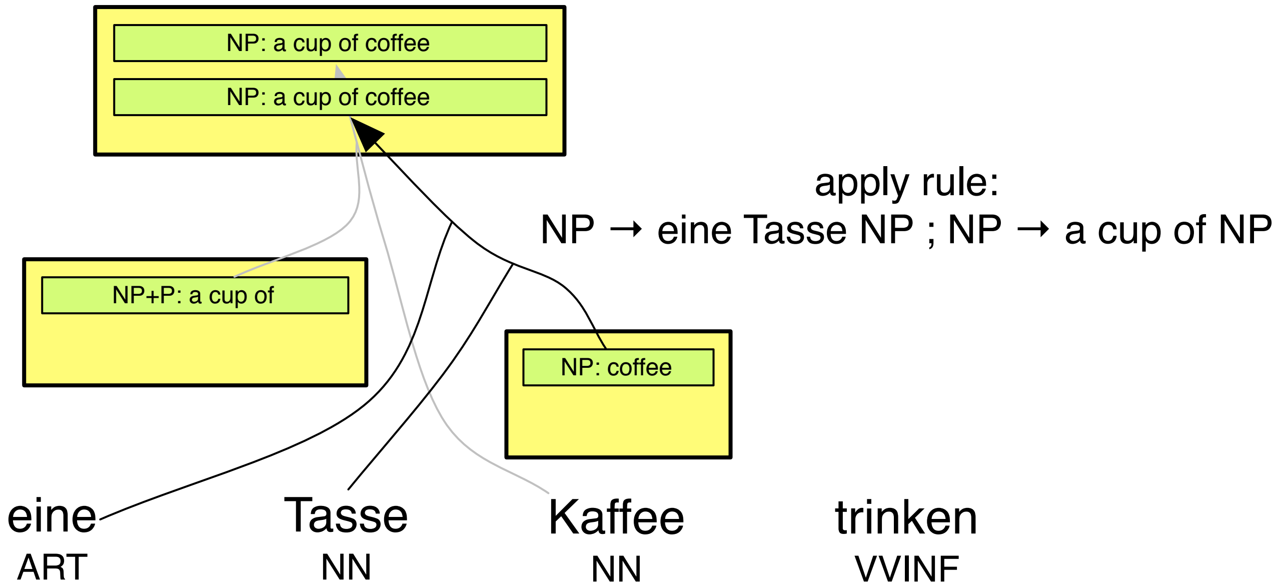
Dynamic Programming

Applying rule creates new hypothesis



Dynamic Programming

Another hypothesis



Both hypotheses are indistiguishable in future search
$\longrightarrow$ can be recombined

Recombinable States

Recombinable?

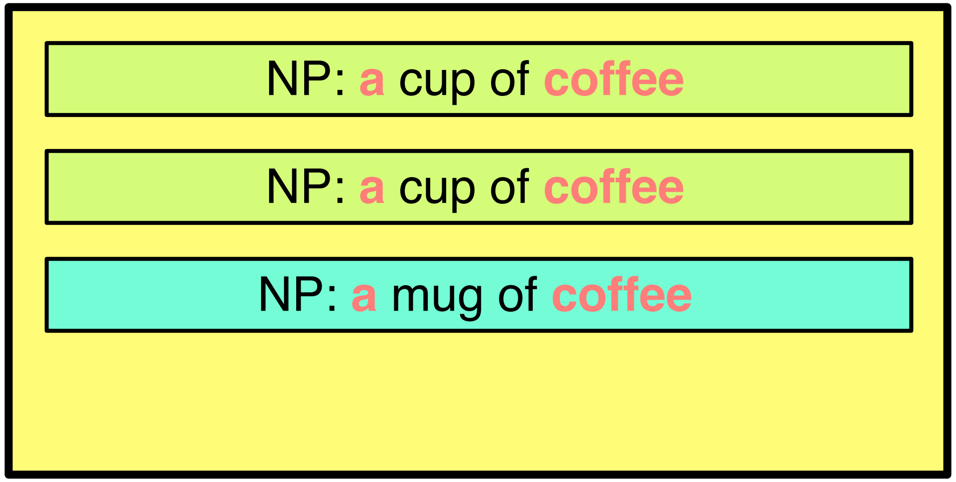
NP: a cup of coffee

NP: a cup of coffee

NP: a mug of coffee

Recombinable States

Recombinable?



Yes, iff max. 2-gram language model is used

Recombinability

Hypotheses have to match in

- span of input words covered
- output constituent label
- first n–1 output words

  not properly scored, since they lack context

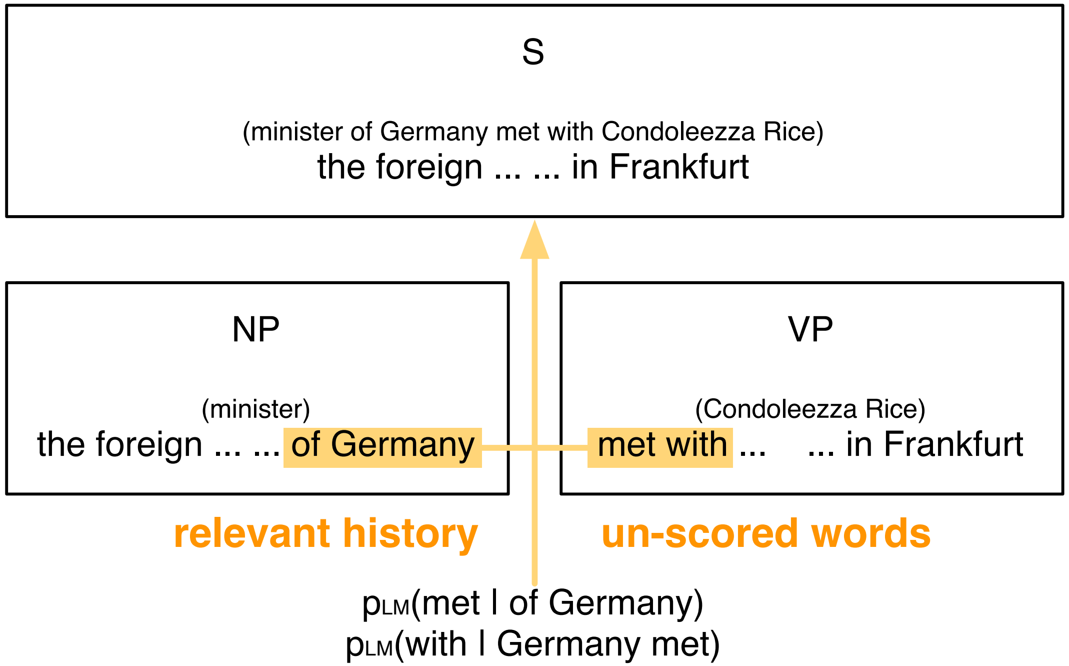
- last n–1 output words

  still affect scoring of subsequently added words, just like in phrase-based decoding

(n is the order of the n-gram language model)

Language Model Contexts

When merging hypotheses, internal language model contexts are absorbed



Stack Pruning

- Number of hypotheses in each chart cell explodes

$\Rightarrow$ need to discard bad hypotheses
e.g., keep 100 best only

- Different stacks for different output constituent labels?

- Cost estimates
  - translation model cost known
  - language model cost for internal words known
    $\rightarrow$ estimates for initial words
  - outside cost estimate?
    (how useful will be a NP covering input words 3–5 later on?)

Naive Algorithm: Blow-ups

- Many subspan sequences

  for all sequences s of hypotheses and words in span [start,end]

- Many rules

  for all rules r

- Checking if a rule applies not trivial

  rule r applies to chart sequence s

$\Rightarrow$ Unworkable

Solution

- Prefix tree data structure for rules
- Dotted rules
- Cube pruning

Storing Rules

- First concern: do they apply to span?
  $\rightarrow$ have to match available hypotheses and input words

- Example rule

$$\text{NP} \rightarrow \text{X}_1 \text{ des } \text{X}_2 \mid \text{NP}_1 \text{ of the } \text{NN}_2$$

- Check for applicability
  - is there an initial sub-span that with a hypothesis with constituent label NP?
  - is it followed by a sub-span over the word des?
  - is it followed by a final sub-span with a hypothesis with label NN?

- Sequence of relevant information

$$\text{NP} \bullet \text{des} \bullet \text{NN} \bullet \text{NP}_1 \text{ of the } \text{NN}_2$$

Rule Applicability Check

Trying to cover a span of six words with given rule

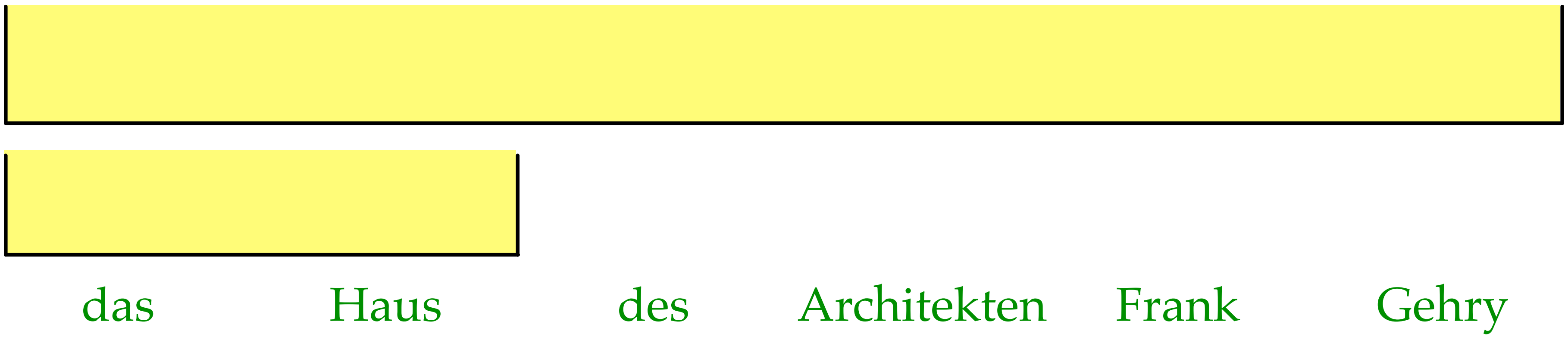
NP • des • NN → NP: NP of the NN

das Haus des Architekten Frank Gehry

Rule Applicability Check

First: check for hypotheses with output constituent label NP

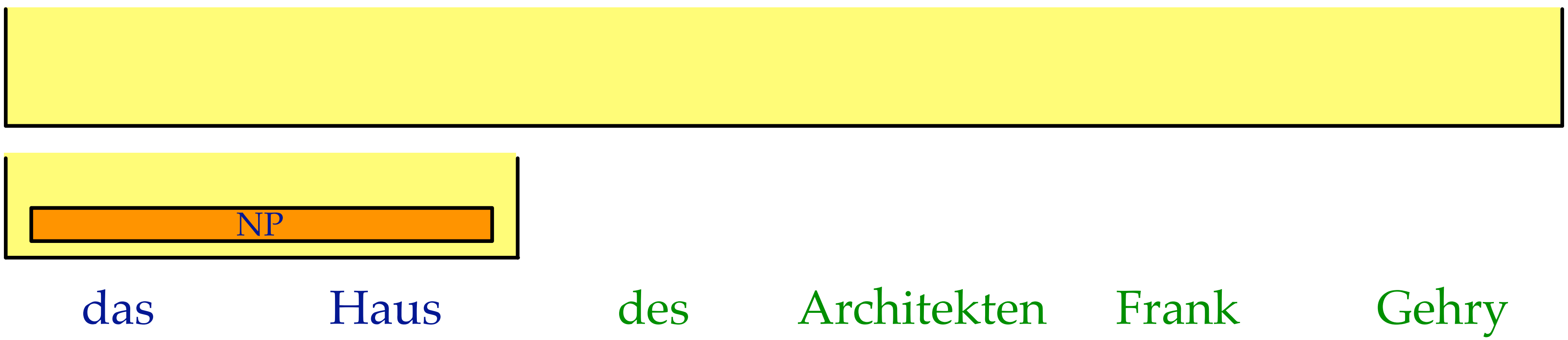
NP • des • NN → NP: NP of the NN



Rule Applicability Check

Found NP hypothesis in cell, matched first symbol of rule

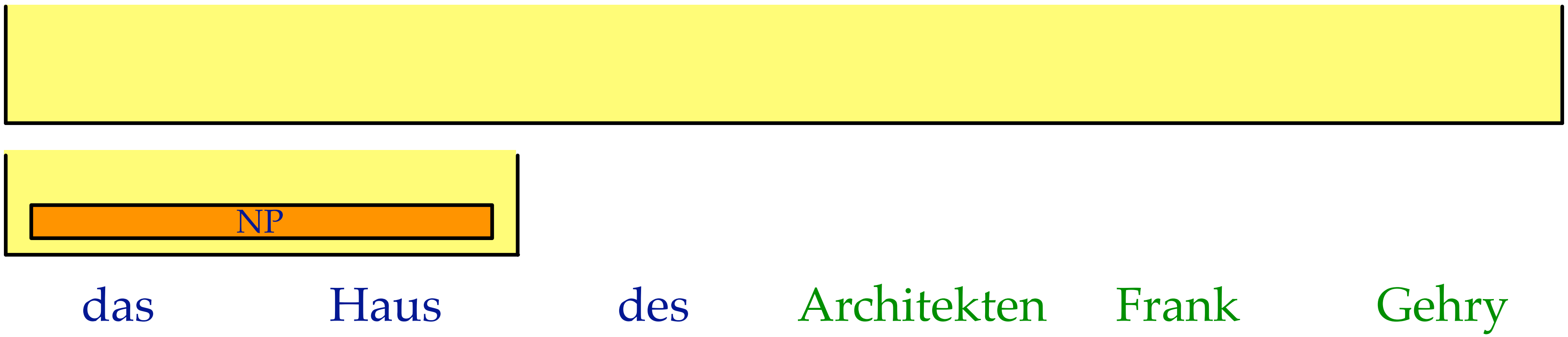
NP • des • NN → NP: NP of the NN



Rule Applicability Check

Matched word des, matched second symbol of rule

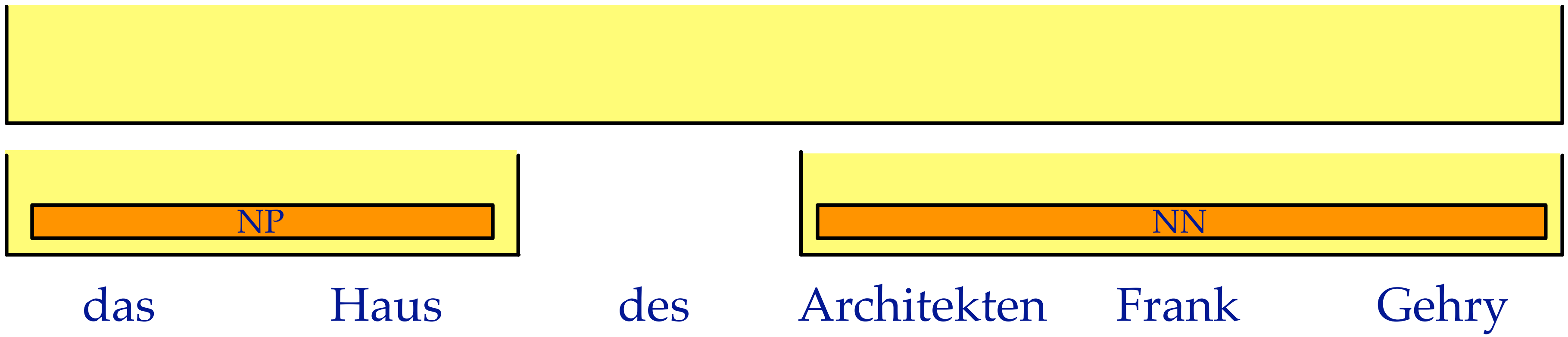
NP • des • NN → NP: NP of the NN



Rule Applicability Check

Found a NN hypothesis in cell, matched last symbol of rule

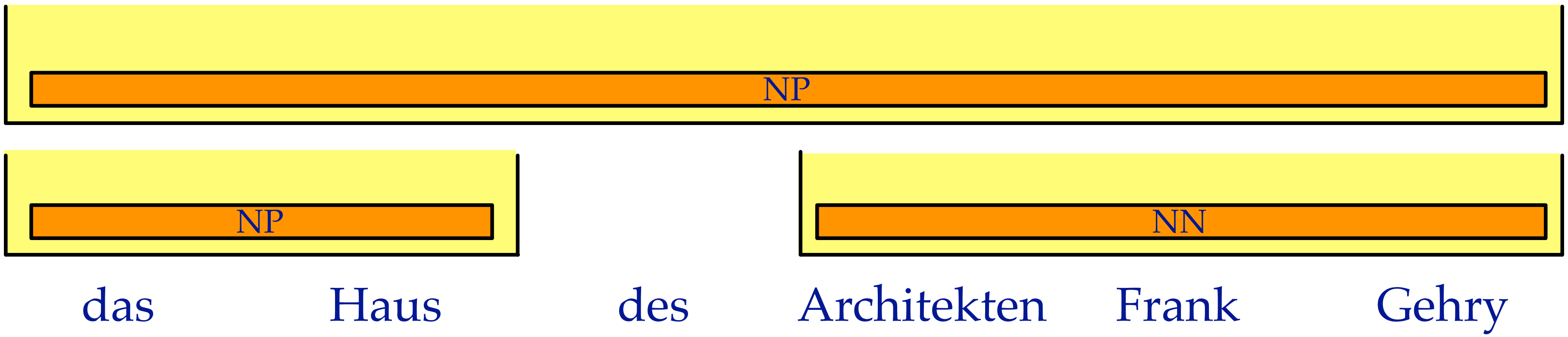
NP • des • NN → NP: NP of the NN



Rule Applicability Check

Matched entire rule → apply to create a NP hypothesis

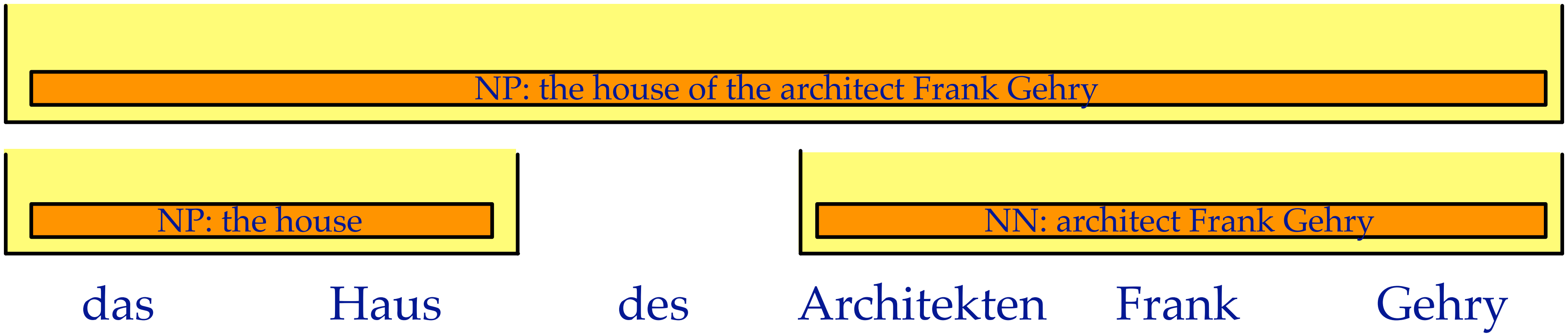
NP • des • NN → NP: NP of the NN



Rule Applicability Check

Look up output words to create new hypothesis
(note: there may be many matching underlying NP and NN hypotheses)

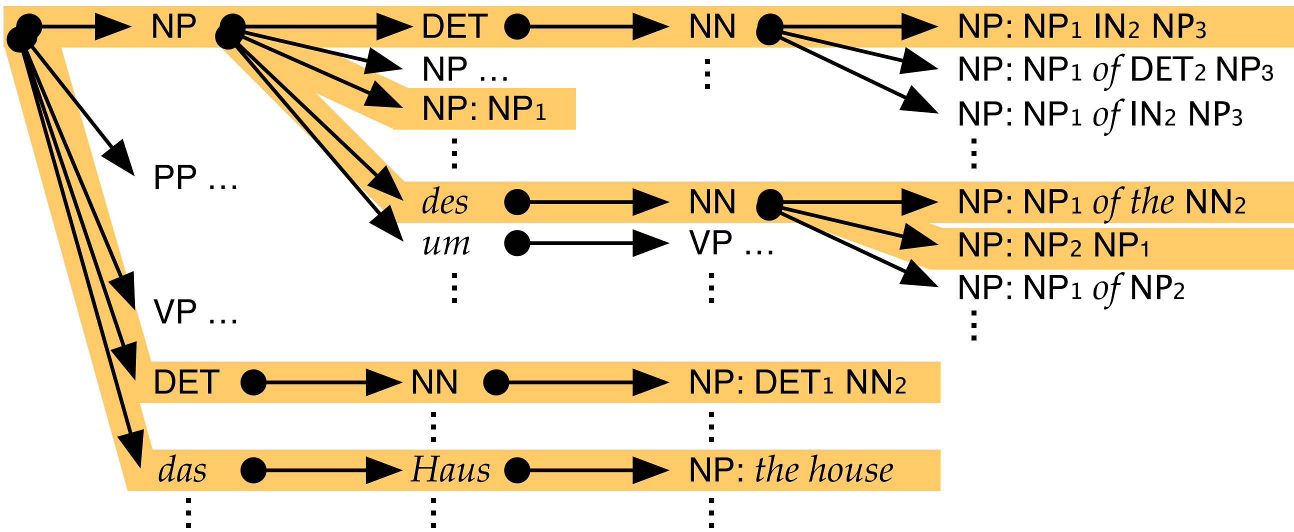
NP • des • NN → NP: NP of the NN



Checking Rules vs. Finding Rules

- What we showed:
  - given a rule
  - check if and how it can be applied
- But there are too many rules (millions) to check them all
- Instead:
  - given the underlying chart cells and input words
  - find which rules apply

Prefix Tree for Rules



Highlighted Rules

$\text{NP} \rightarrow \text{NP}_1\ \text{DET}_2\ \text{NN}_3 \mid \text{NP}_1\ \text{IN}_2\ \text{NN}_3$

$\text{NP} \rightarrow \text{NP}_1 \mid \text{NP}_1$

$\text{NP} \rightarrow \text{NP}_1\ \text{des}\ \text{NN}_2 \mid \text{NP}_1\ \text{of the}\ \text{NN}_2$

$\text{NP} \rightarrow \text{NP}_1\ \text{des}\ \text{NN}_2 \mid \text{NP}_2\ \text{NP}_1$

$\text{NP} \rightarrow \text{DET}_1\ \text{NN}_2 \mid \text{DET}_1\ \text{NN}_2$

$\text{NP} \rightarrow \text{das Haus} \mid \text{the house}$

Dotted Rules: Key Insight

- If we can apply a rule like

$$p \rightarrow \text{A B C} \ | \ x$$

  to a span

- Then we could have applied a rule like

$$q \rightarrow \text{A B} \ | \ y$$

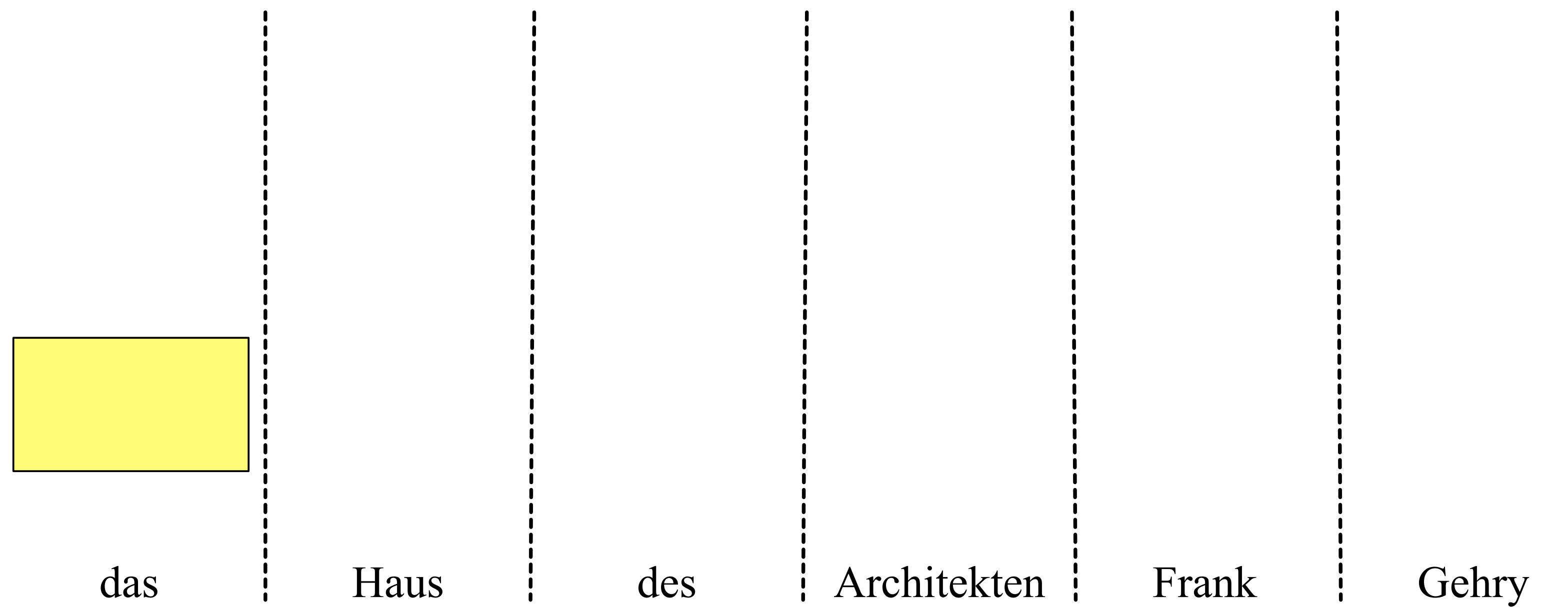
  to a sub-span with the same starting word

$\Rightarrow$ We can re-use rule lookup by storing $\text{A B} \bullet$ (dotted rule)

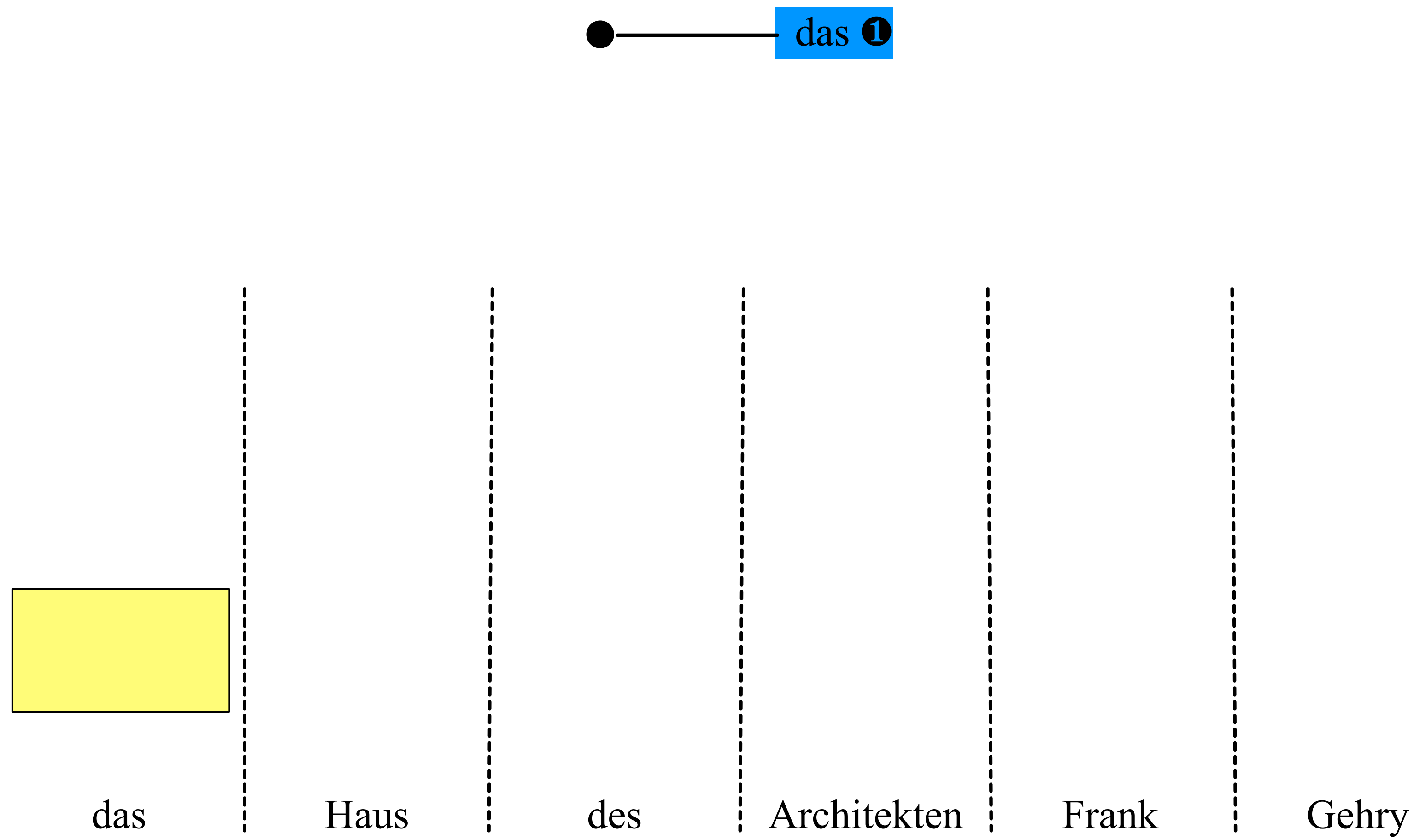
Finding Applicable Rules in Prefix Tree

das | Haus | des | Architekten | Frank | Gehry

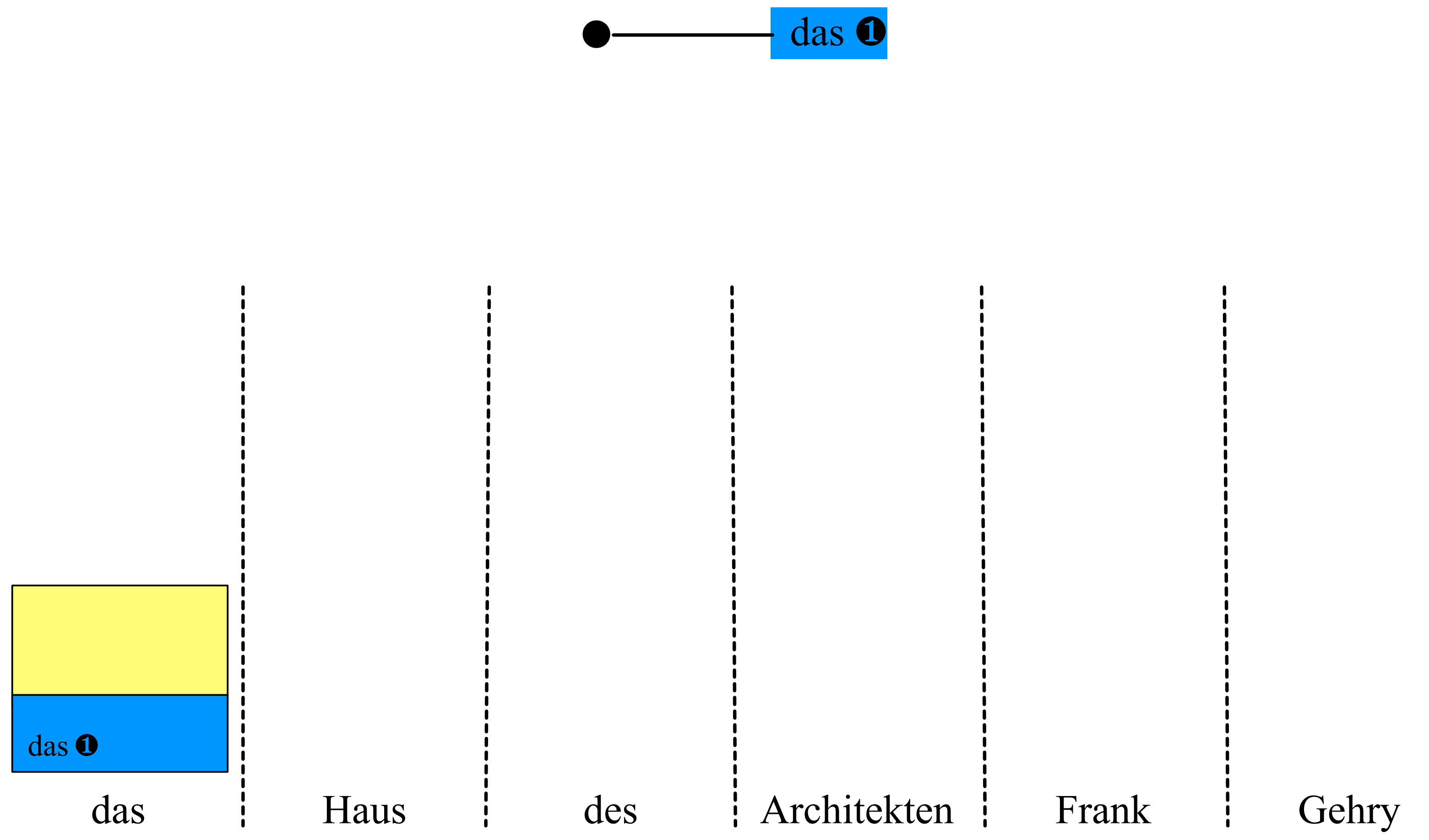
Covering the First Cell



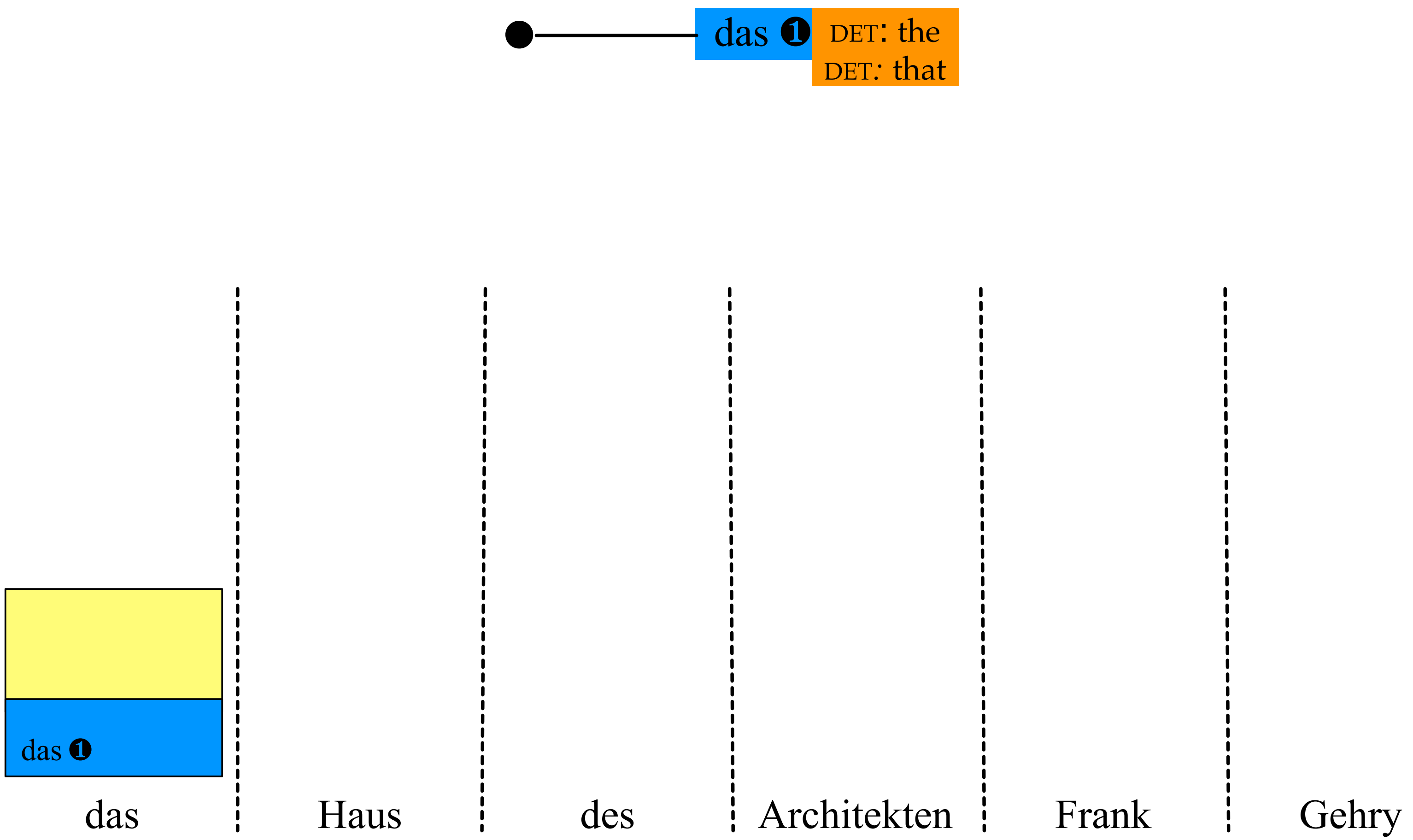
Looking up Rules in the Prefix Tree



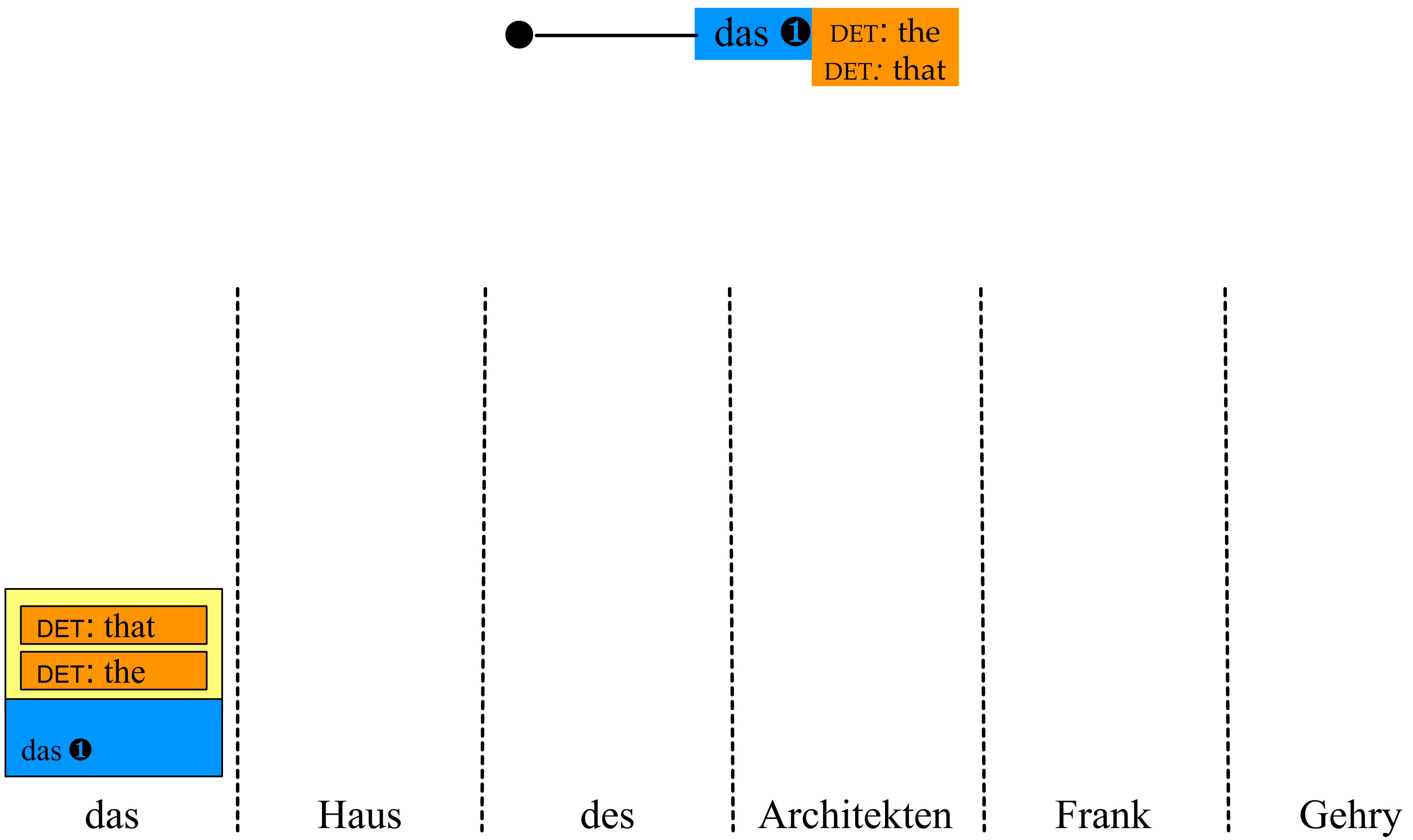
Taking Note of the Dotted Rule



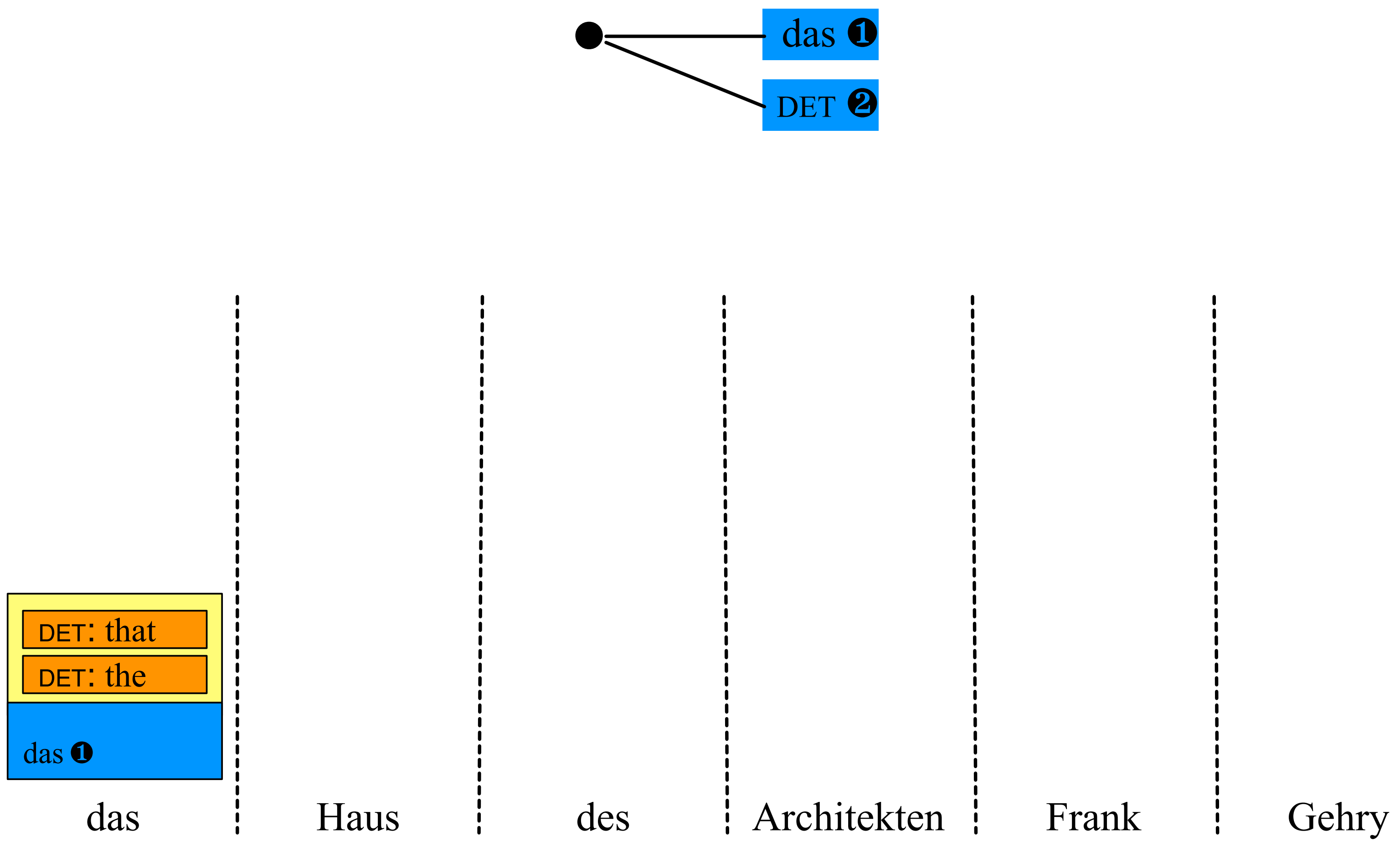
Checking if Dotted Rule has Translations



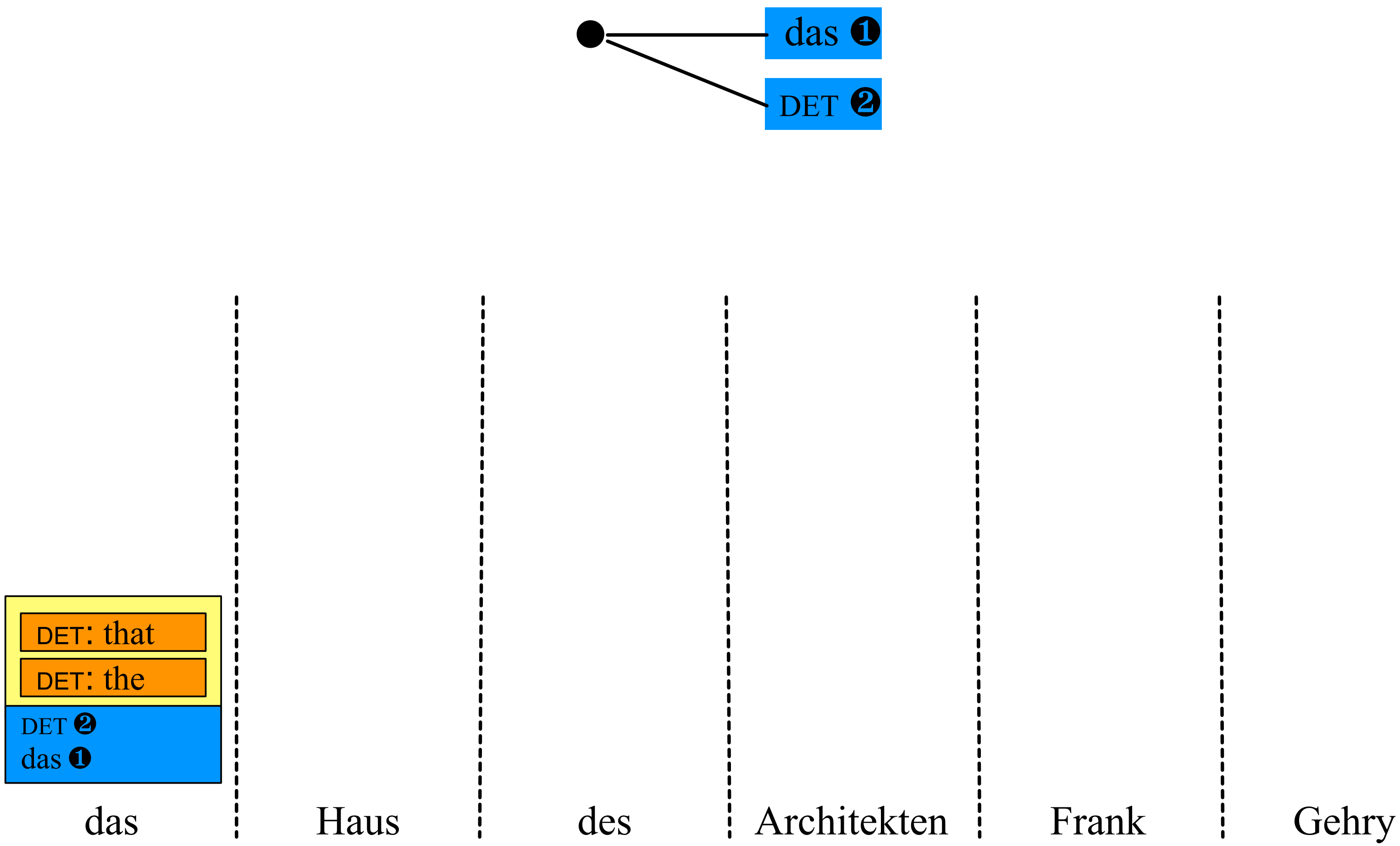
Applying the Translation Rules



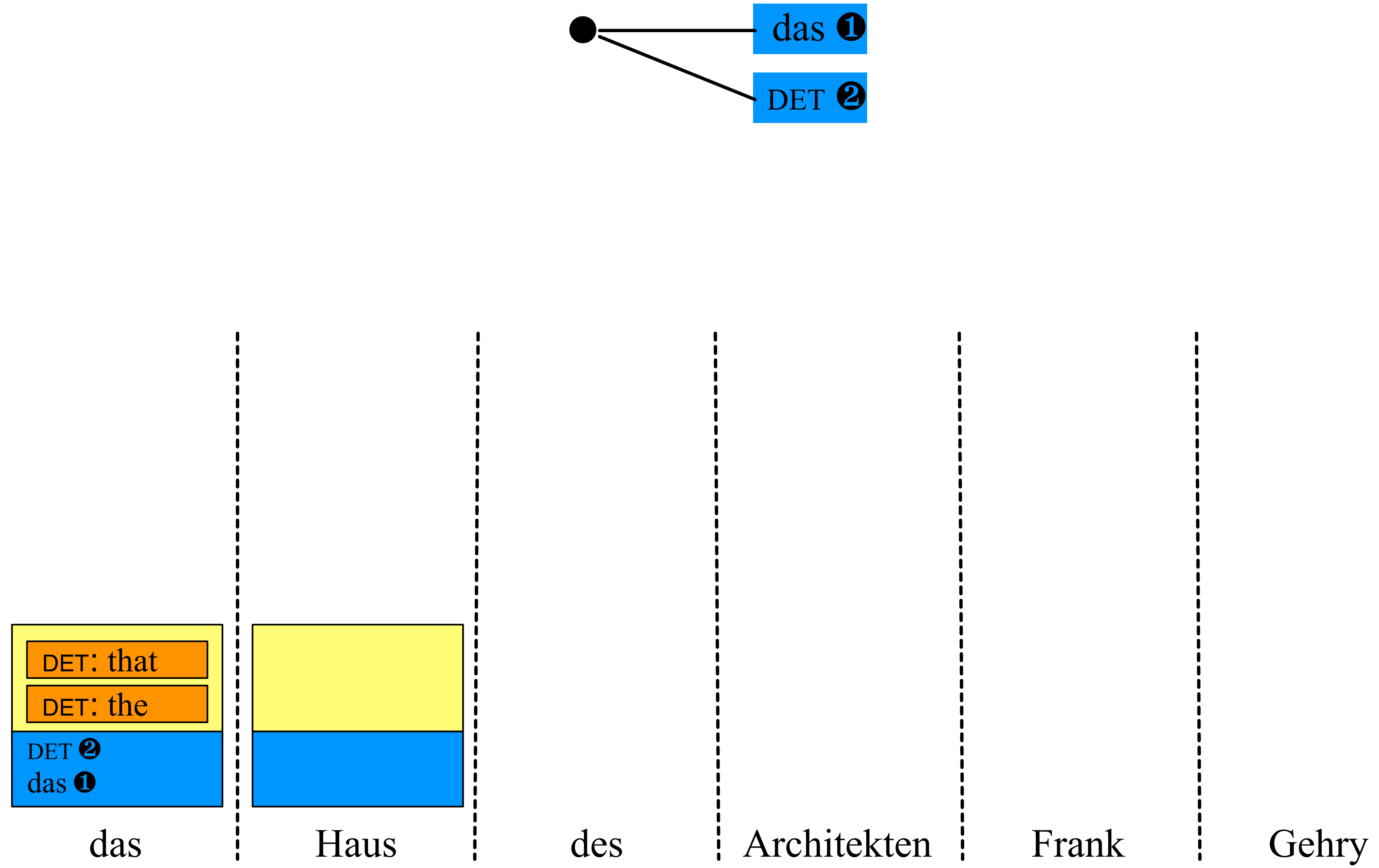
Looking up Constituent Label in Prefix Tree



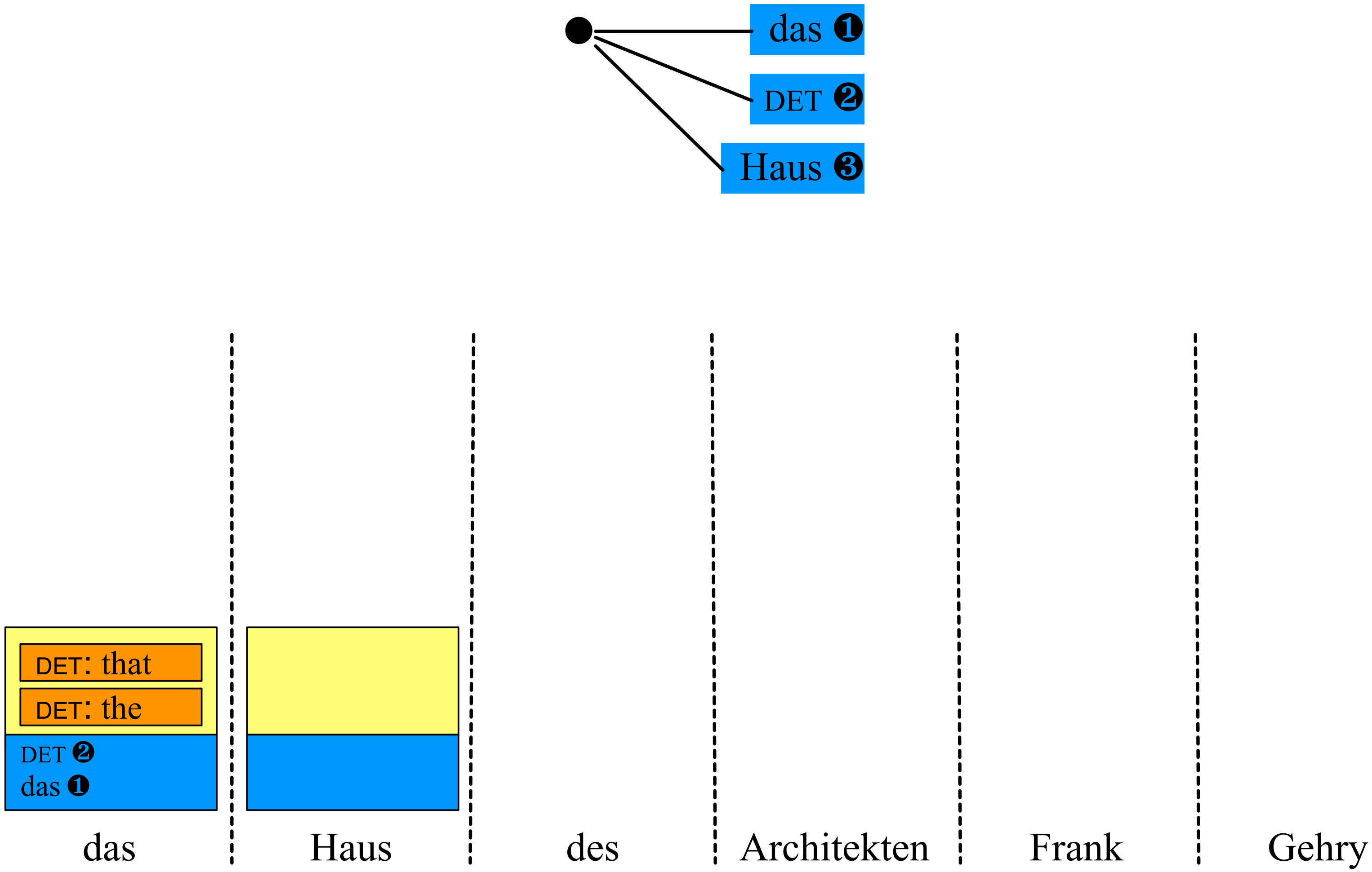
Add to Span's List of Dotted Rules



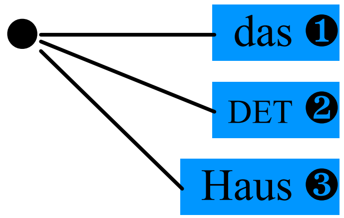
Moving on to the Next Cell

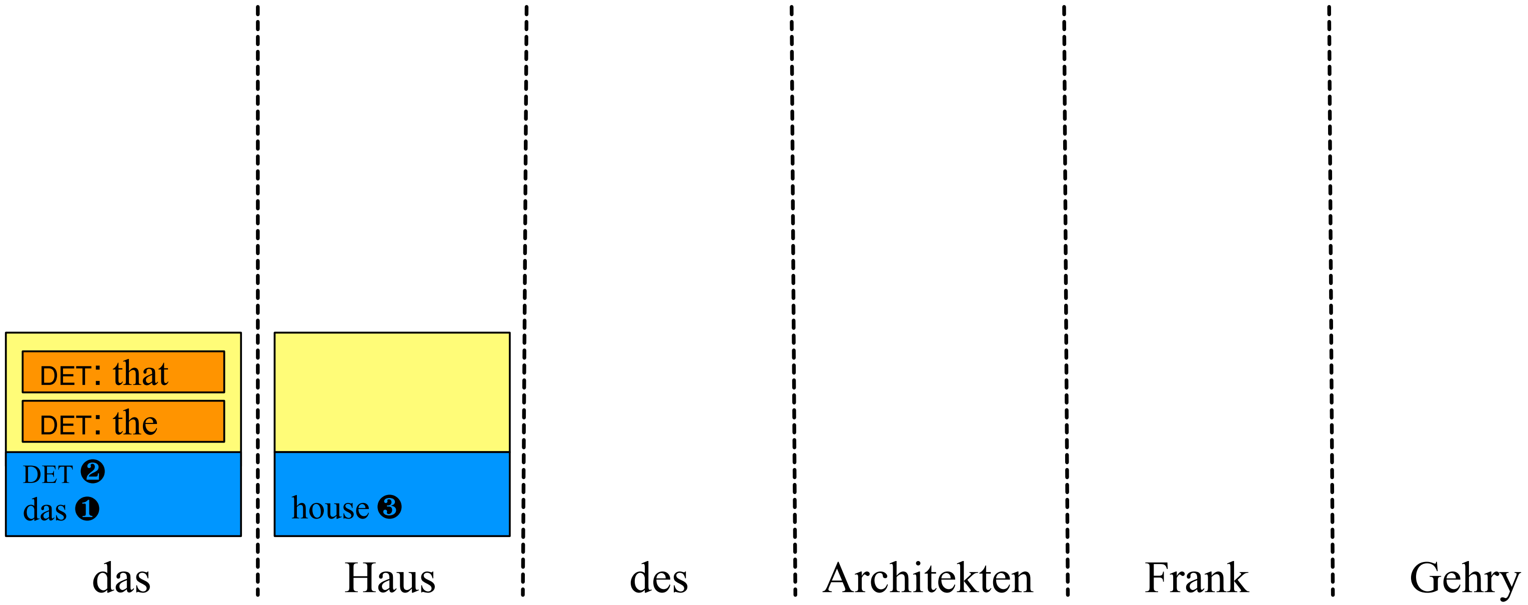


Looking up Rules in the Prefix Tree

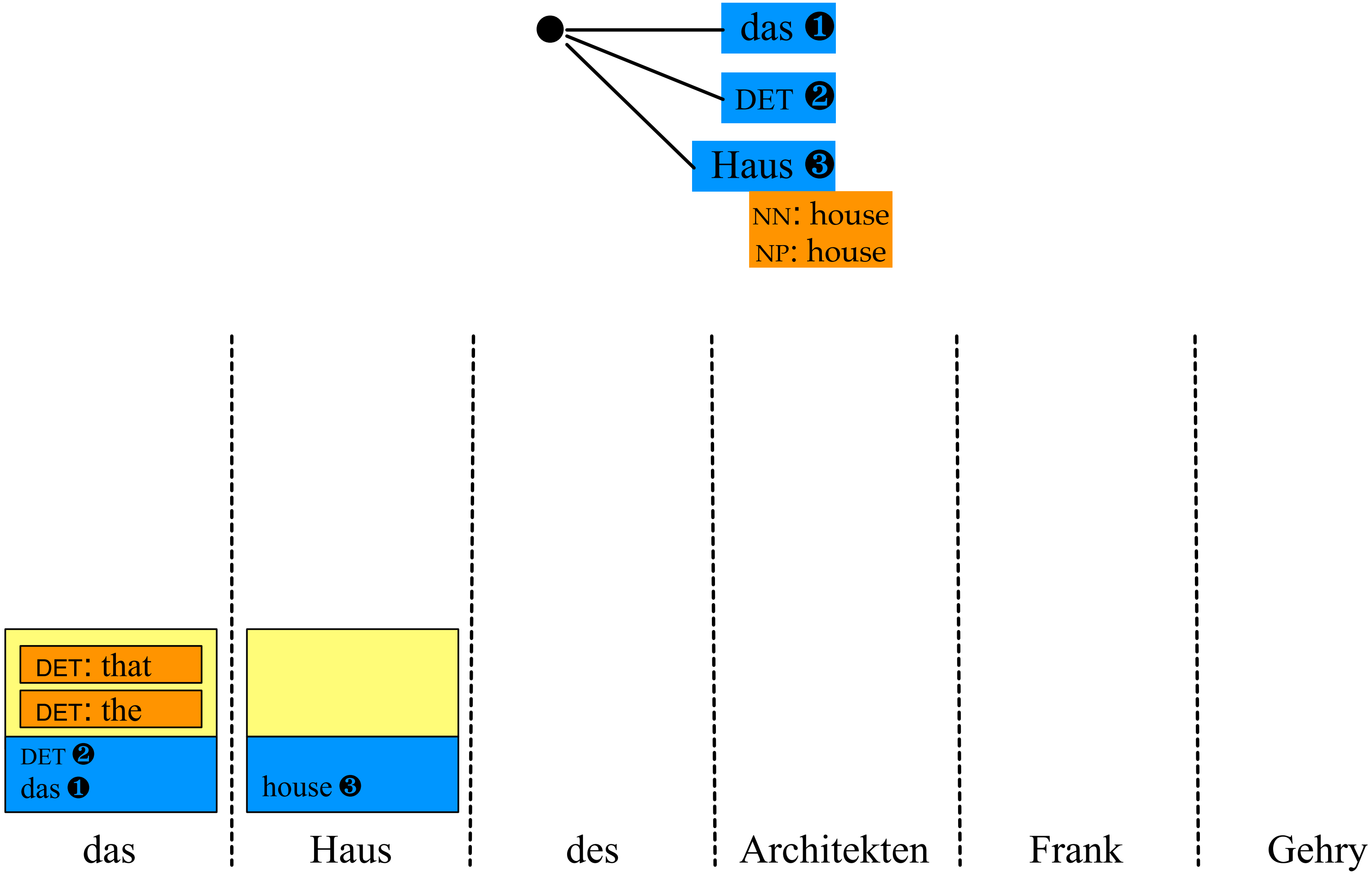


Taking Note of the Dotted Rule

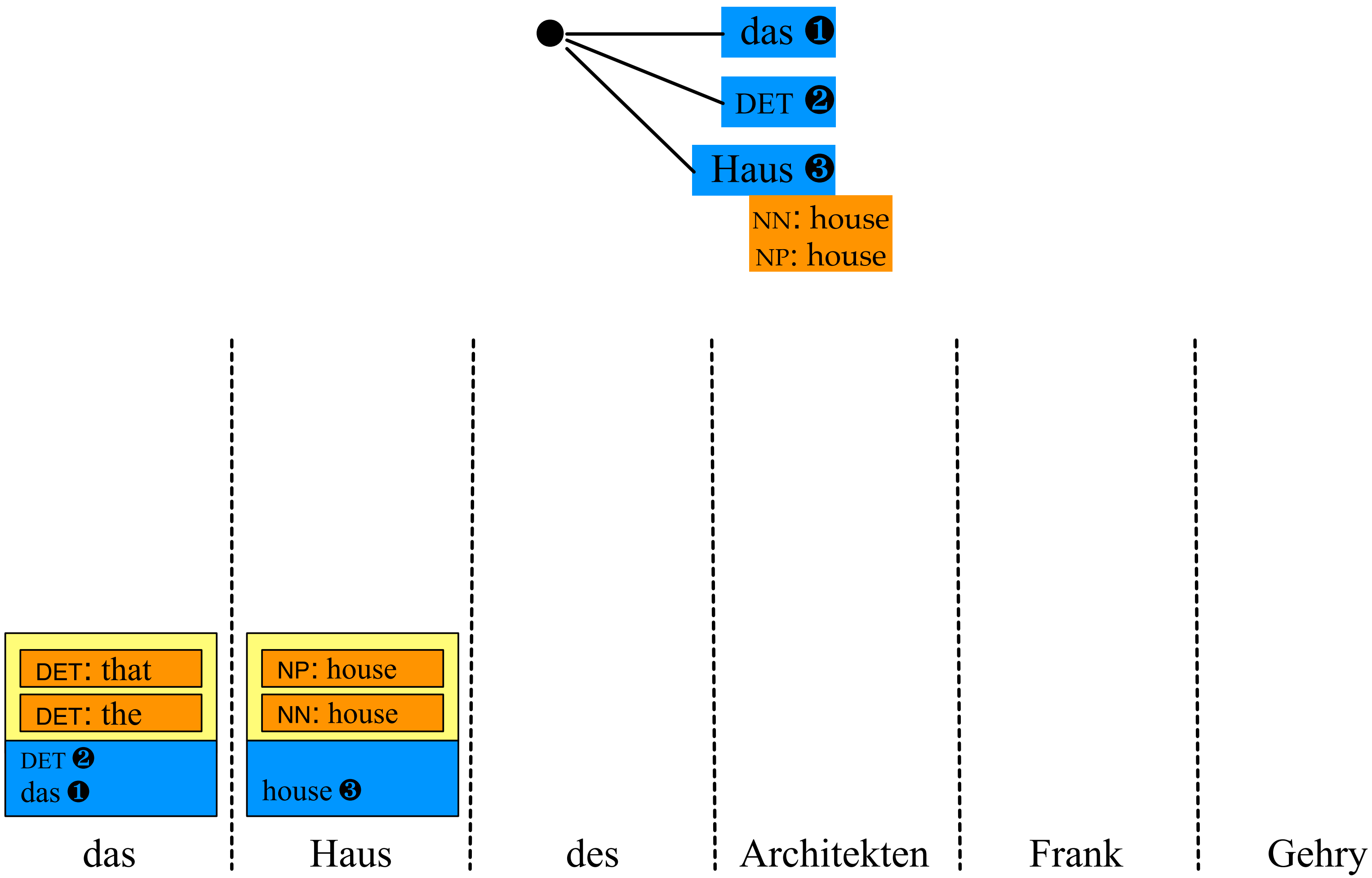




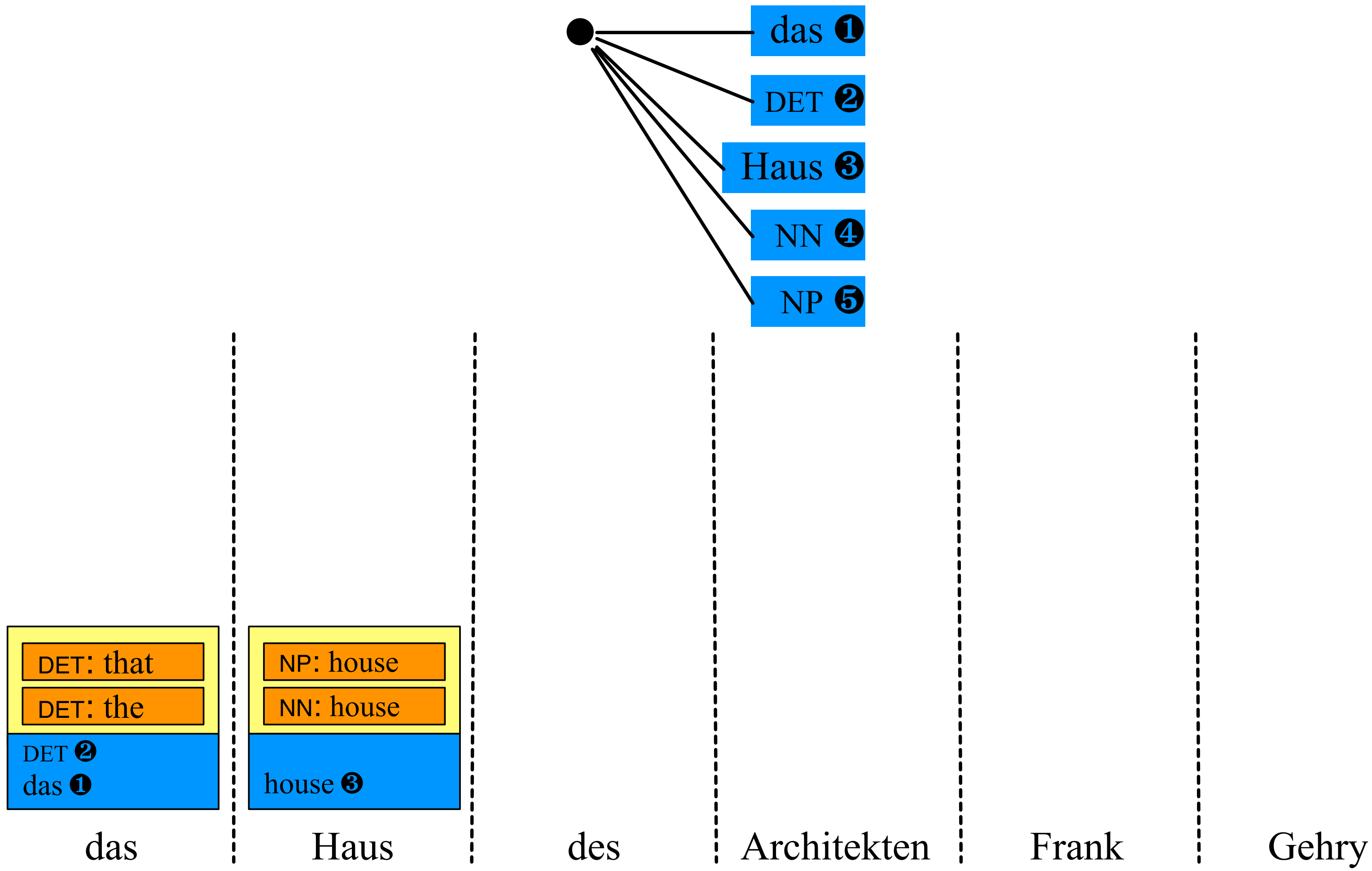
Checking if Dotted Rule has Translations



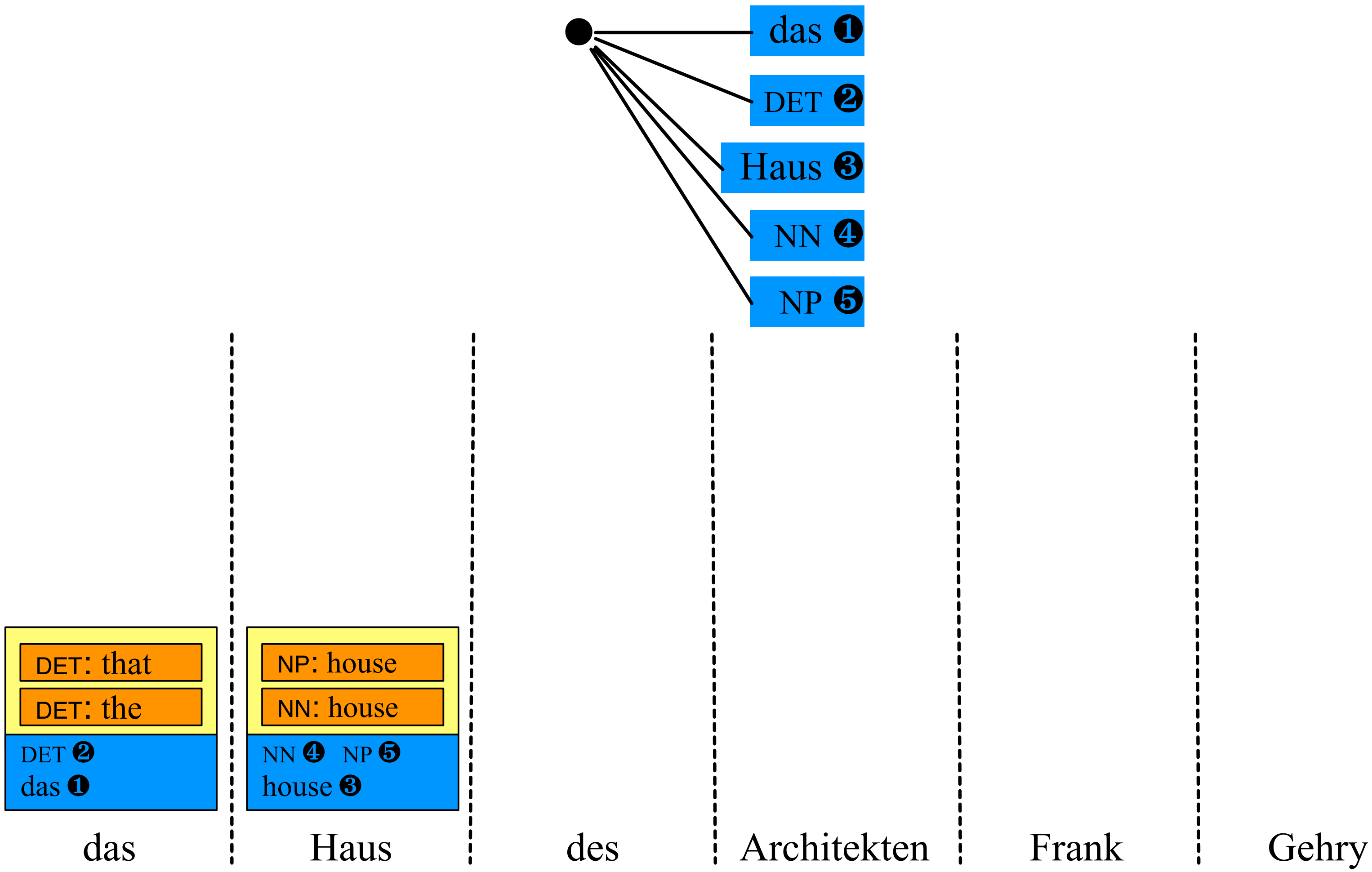
Applying the Translation Rules



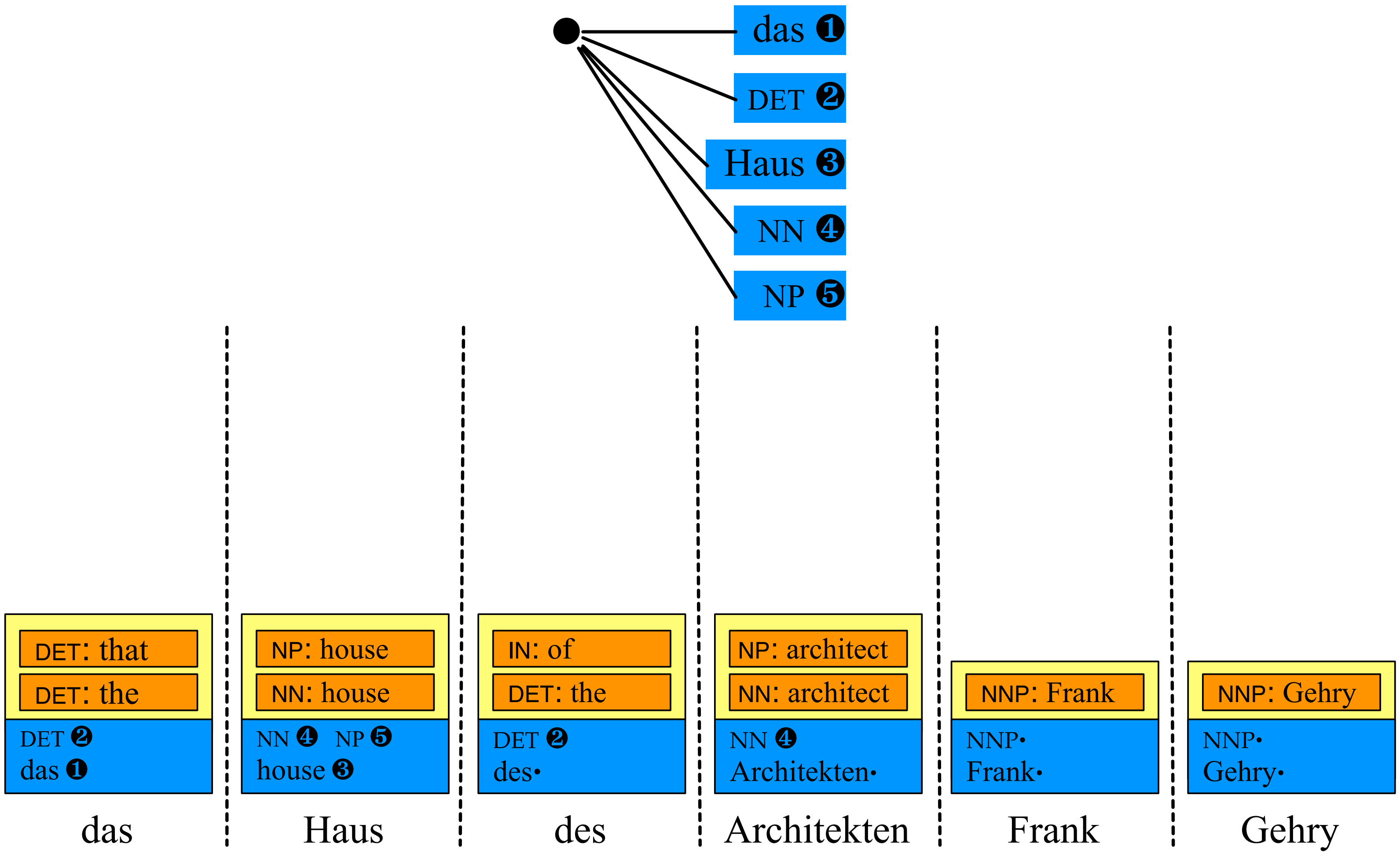
Looking up Constituent Label in Prefix Tree



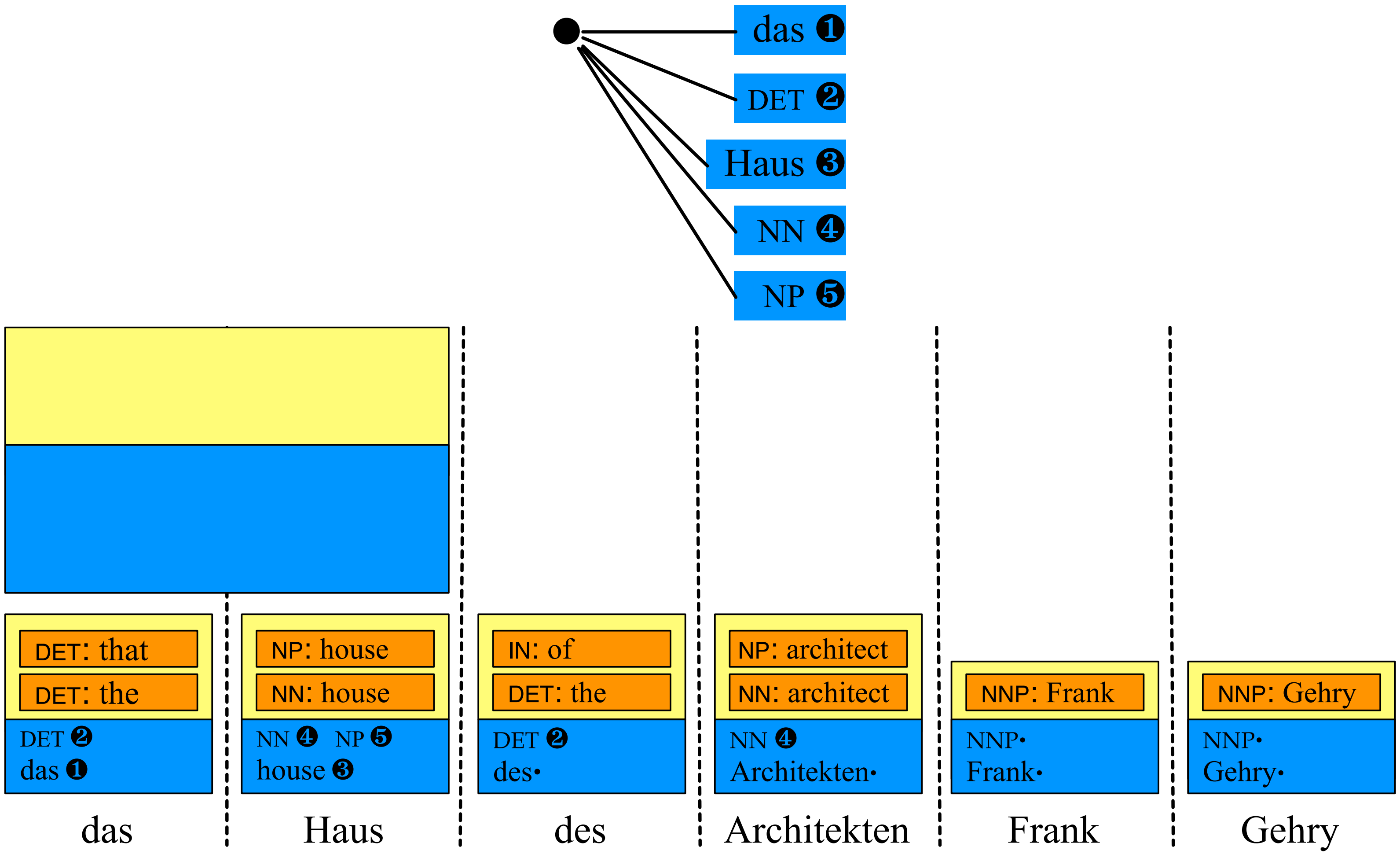
Add to Span's List of Dotted Rules



More of the Same



Moving on to the Next Cell

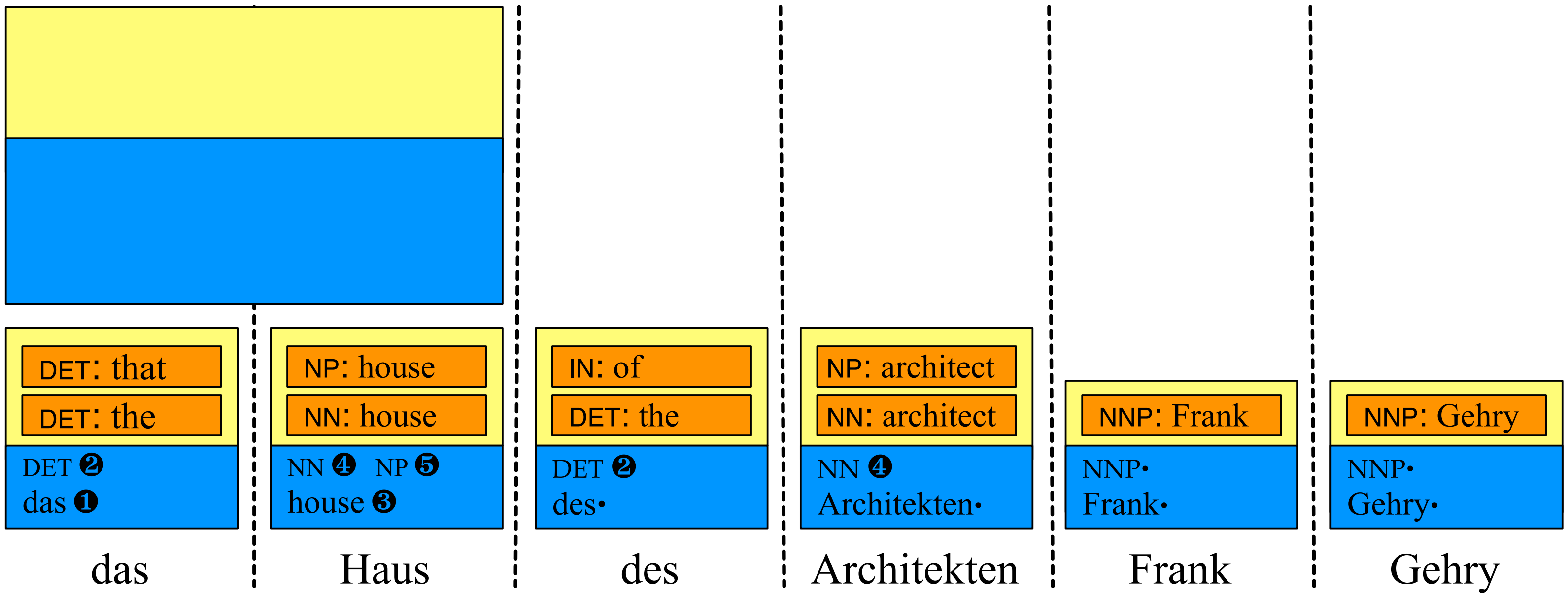


Covering a Longer Span

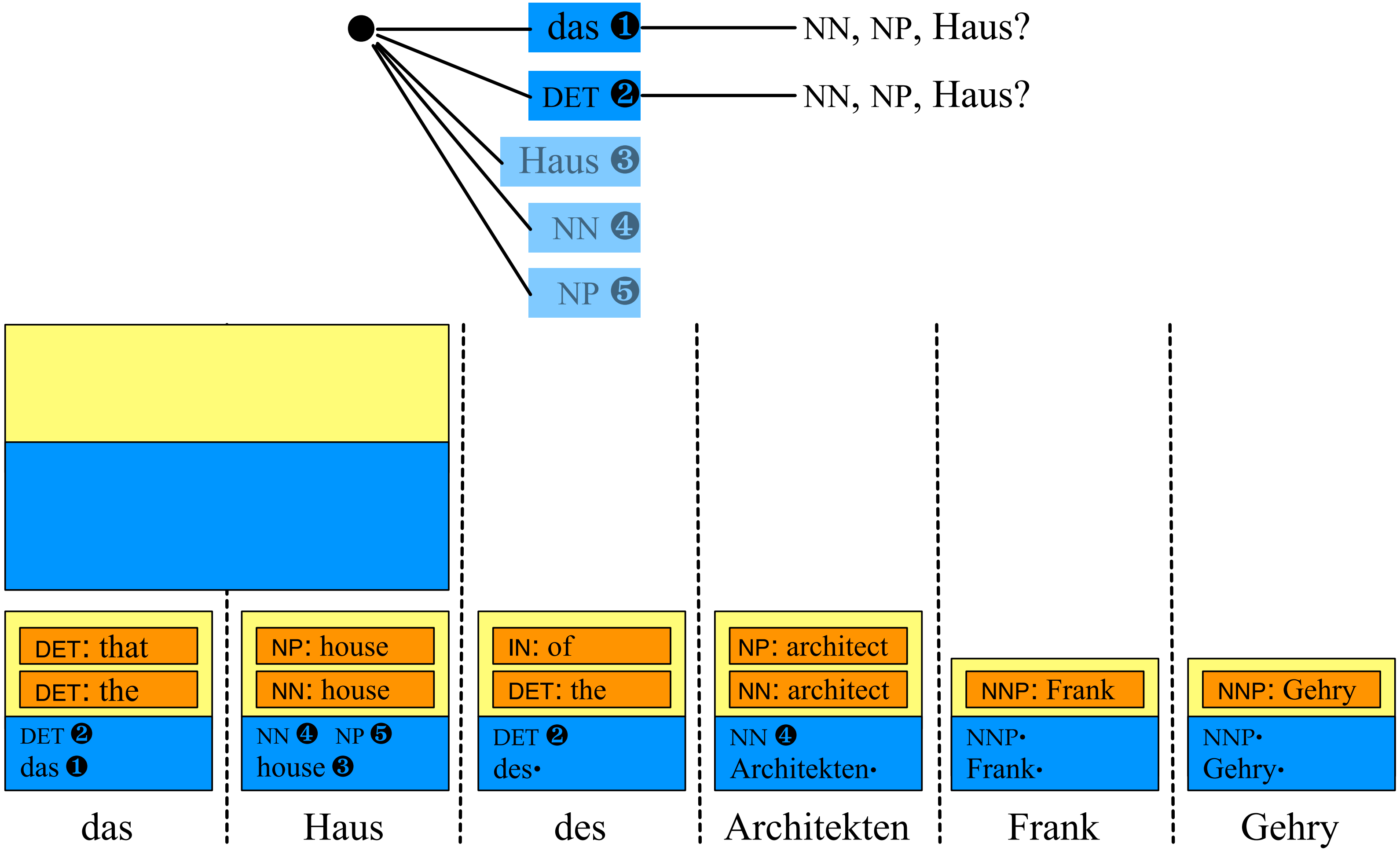
Cannot consume multiple words at once

All rules are extensions of existing dotted rules

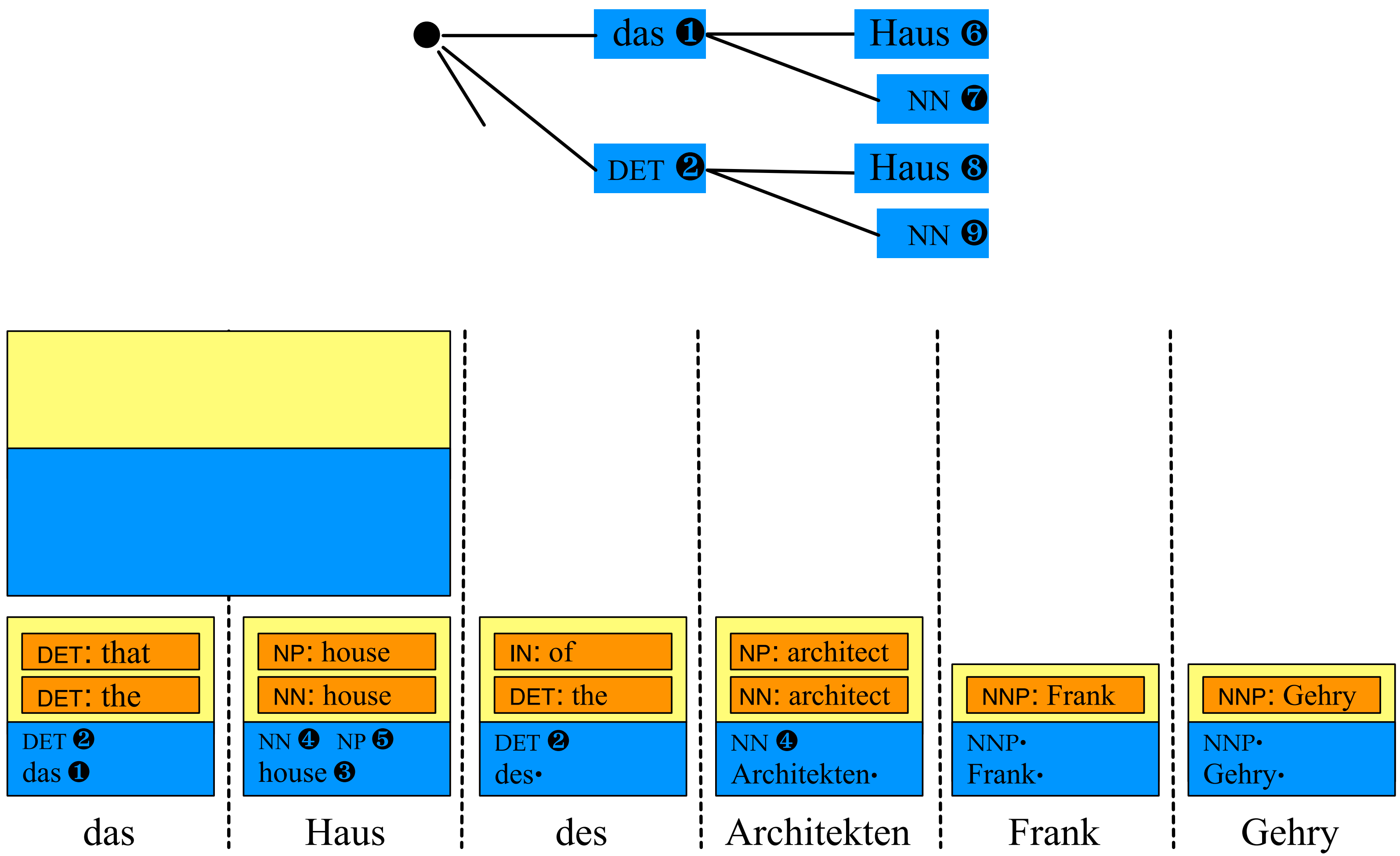
Here: only extensions of span over das possible



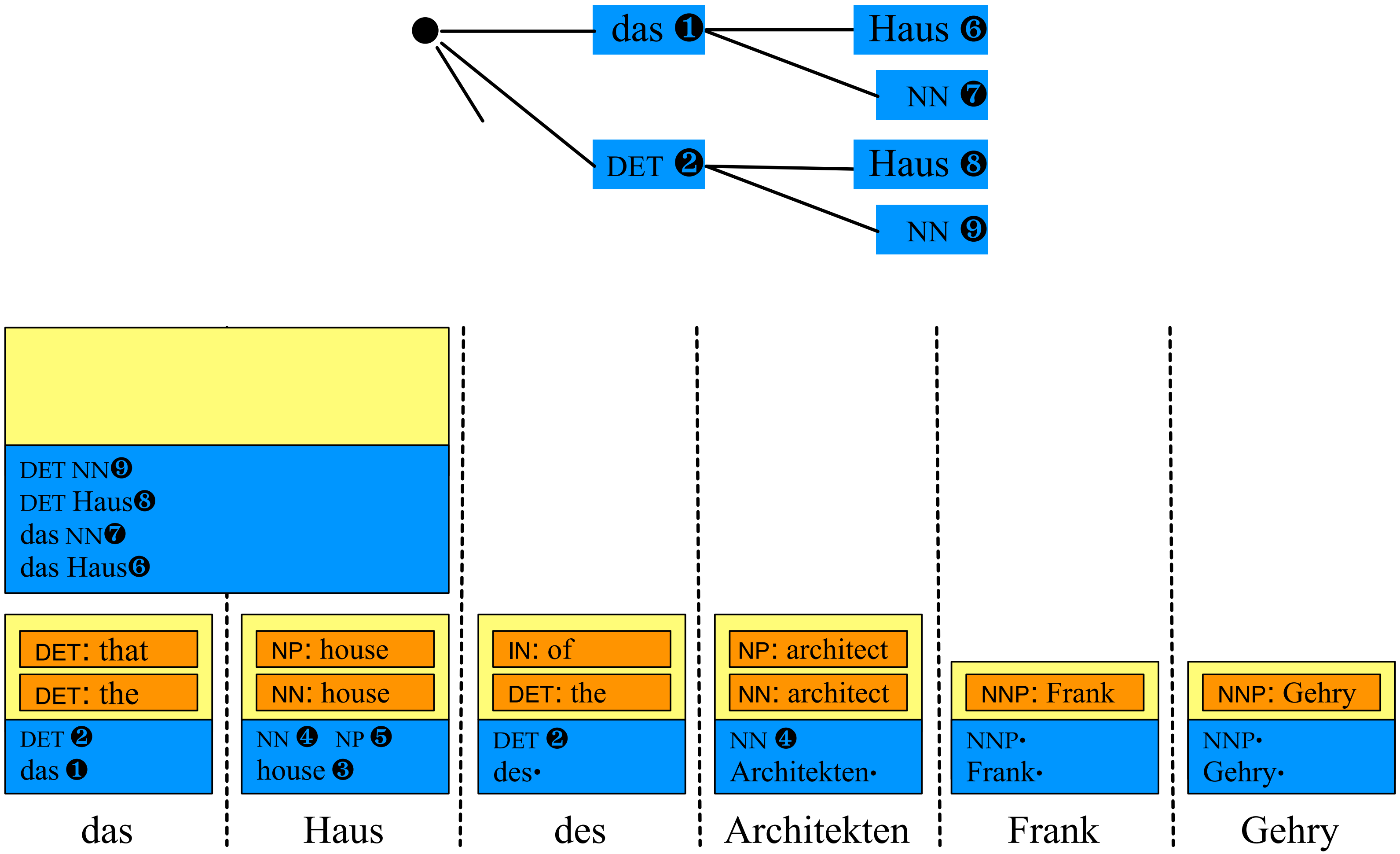
Extensions of Span over das



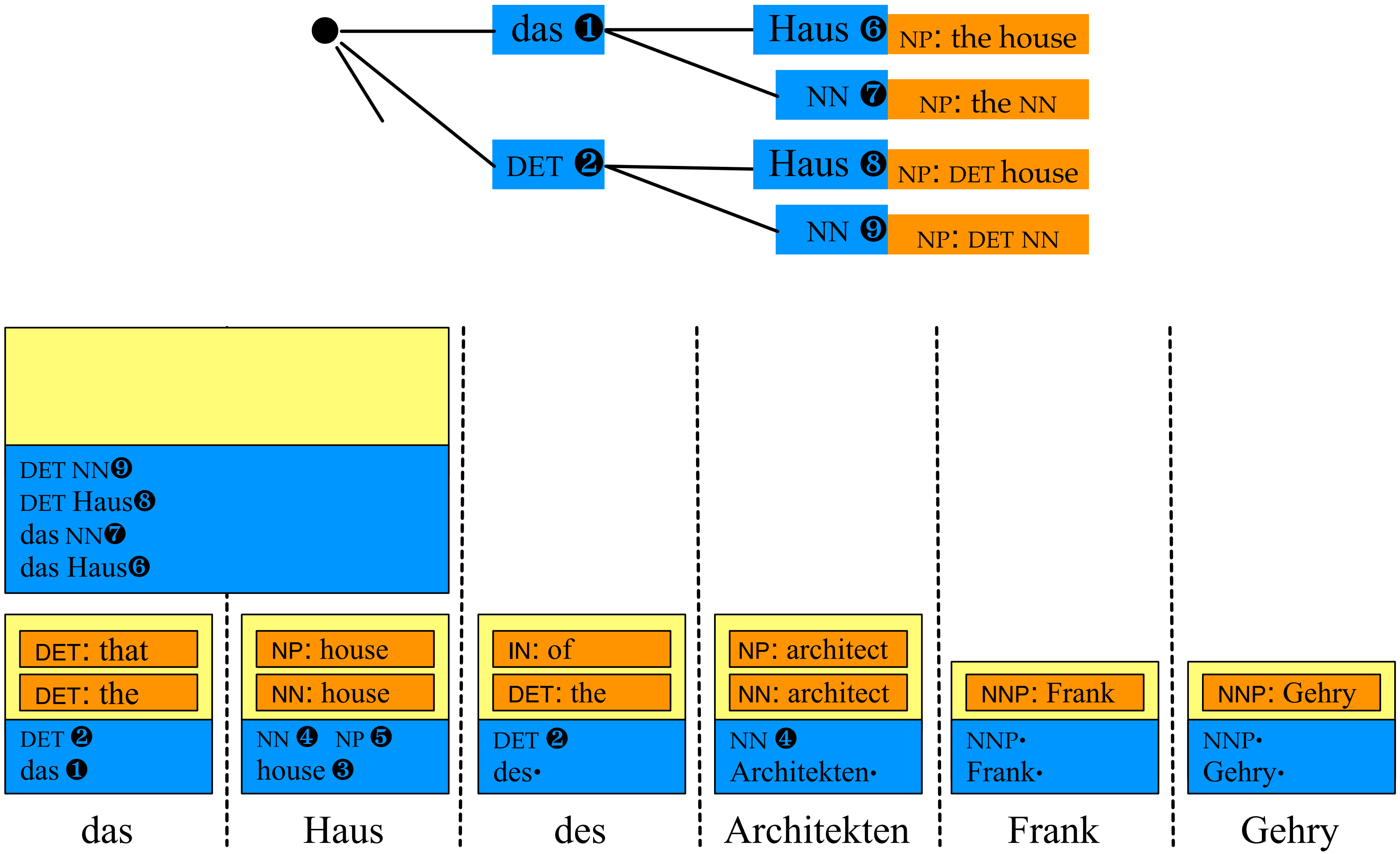
Looking up Rules in the Prefix Tree



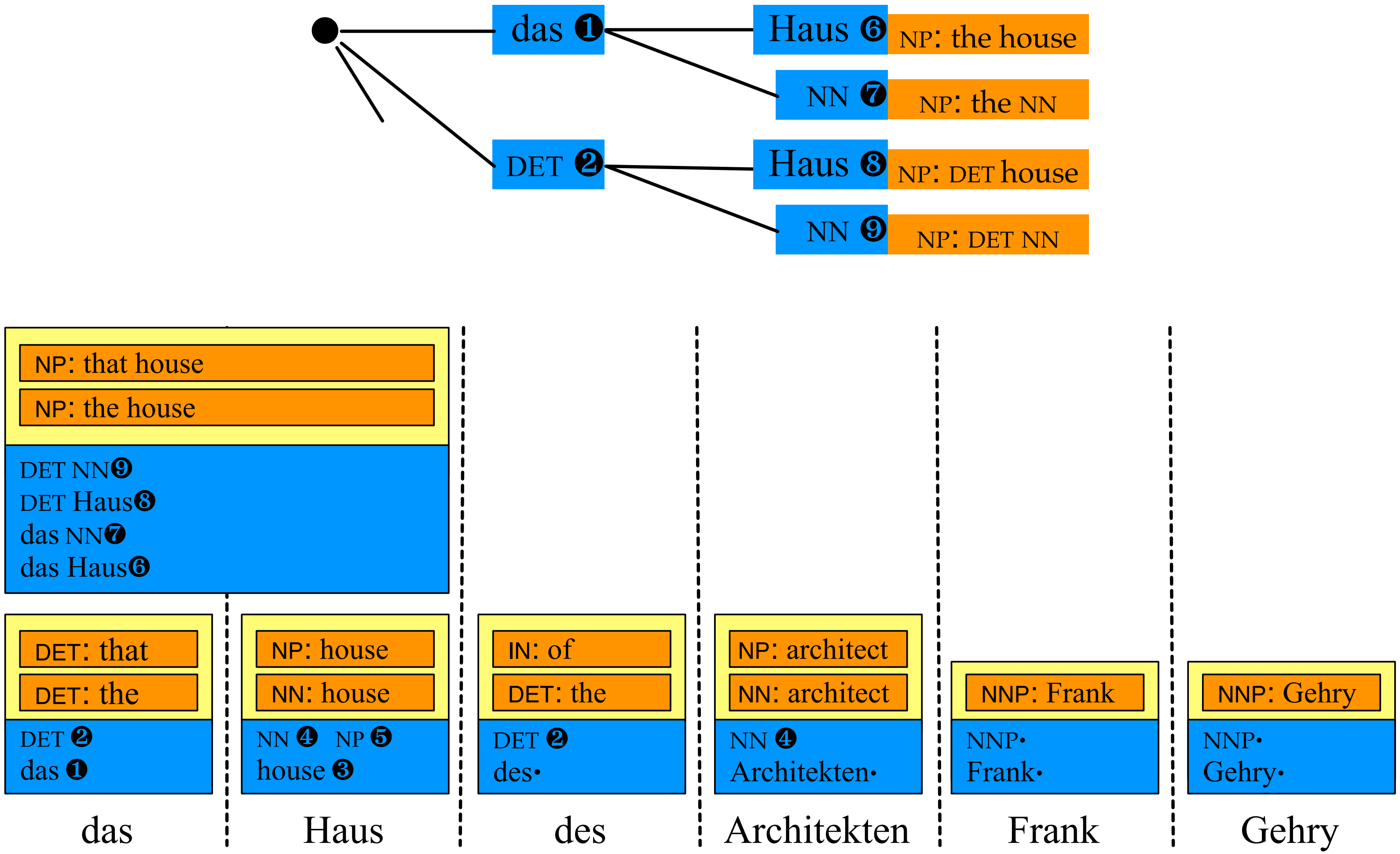
Taking Note of the Dotted Rule



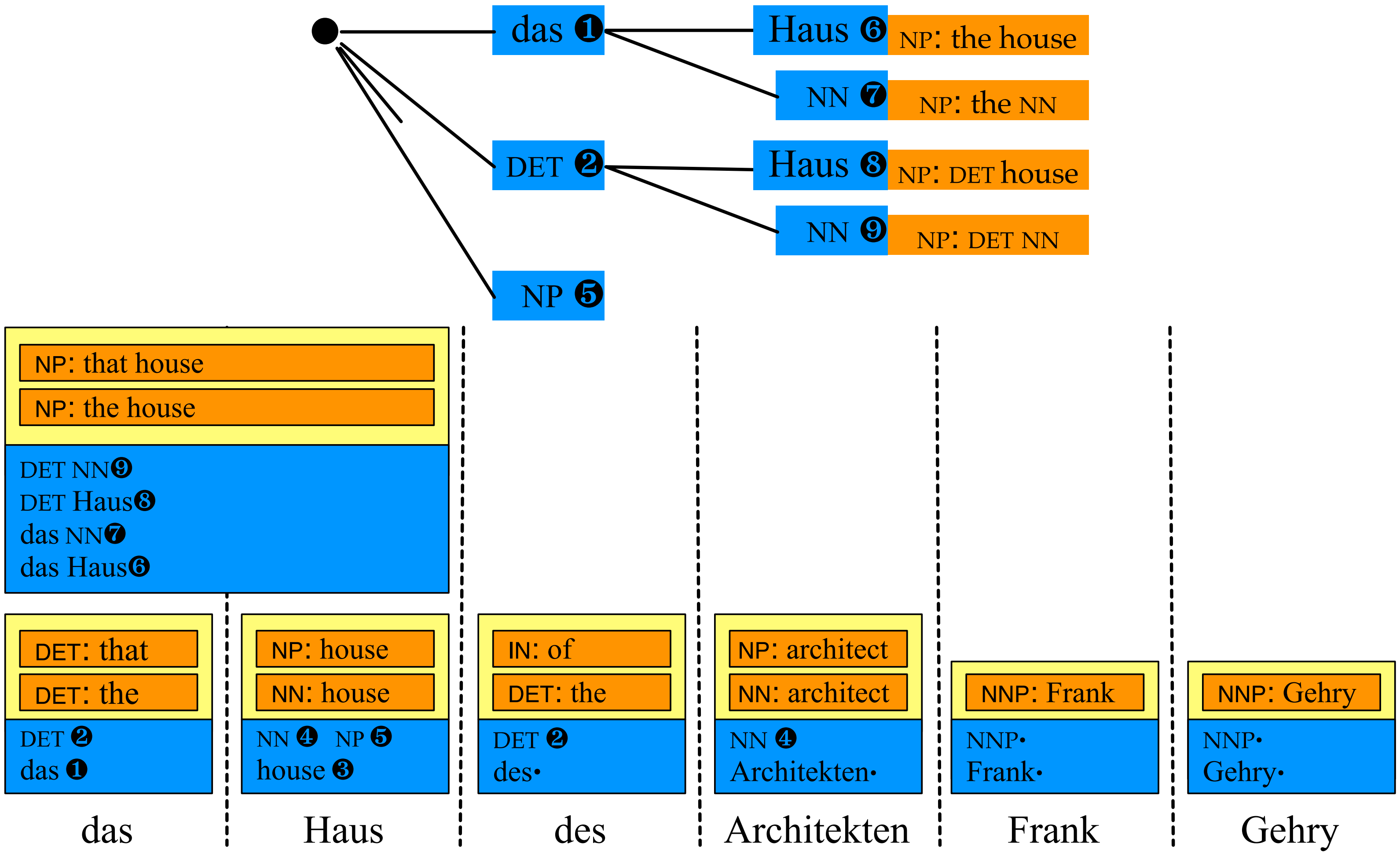
Checking if Dotted Rules have Translations



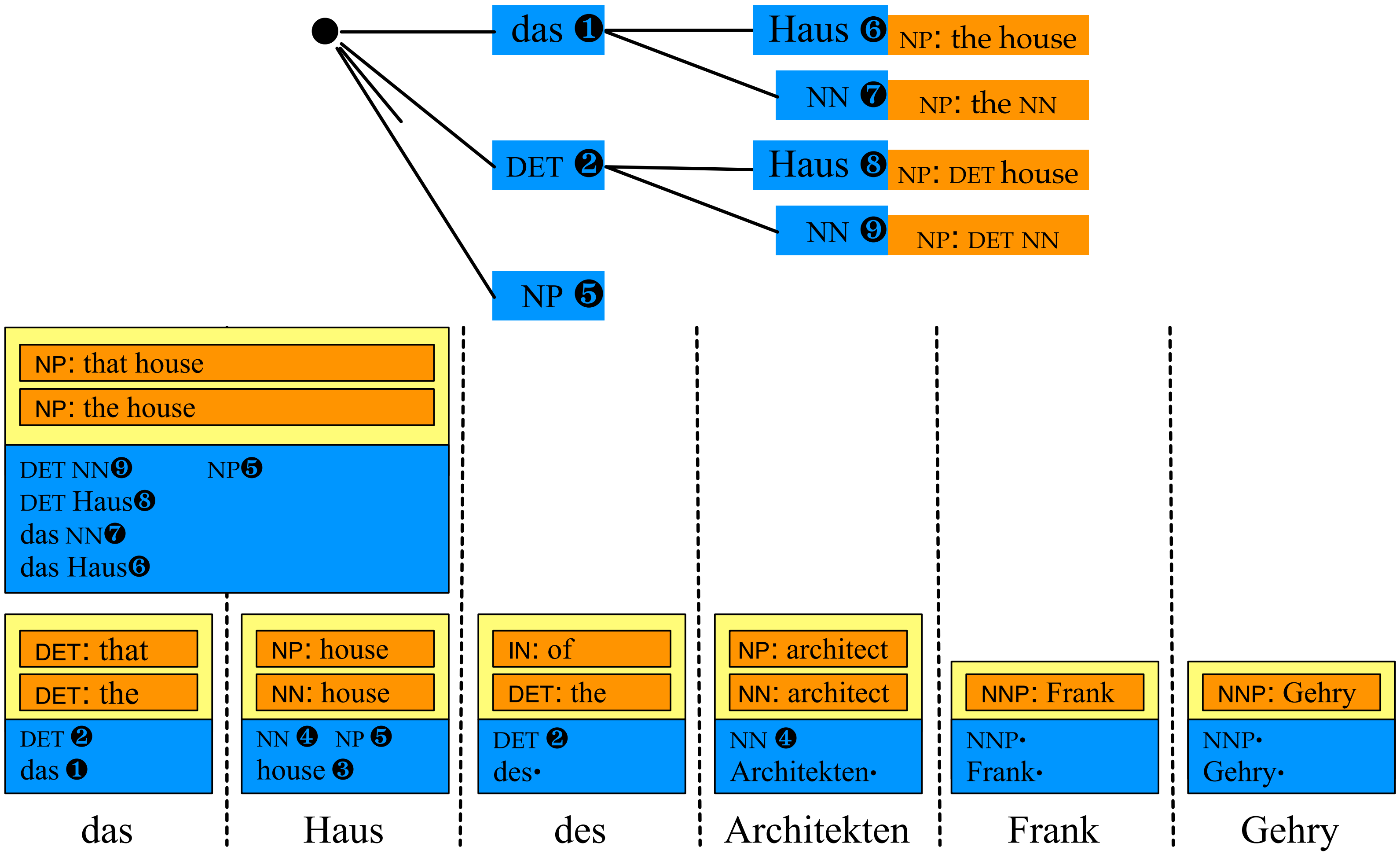
Applying the Translation Rules



Looking up Constituent Label in Prefix Tree



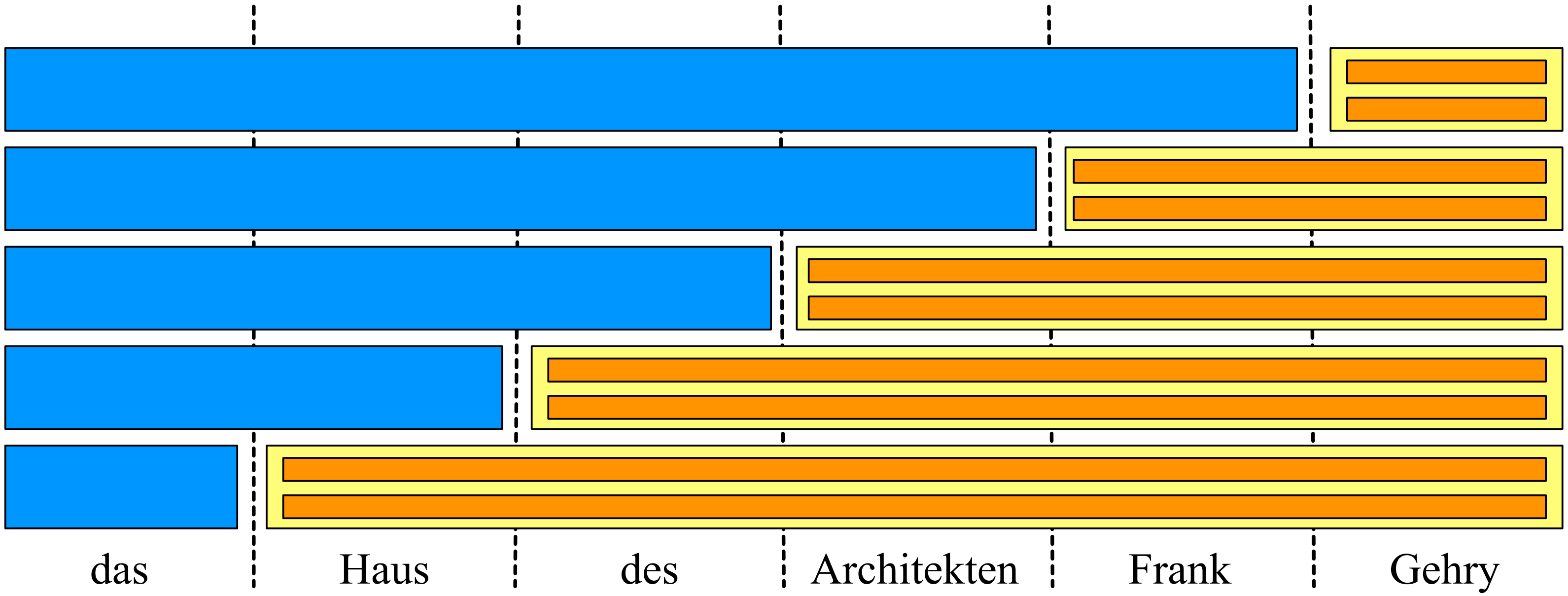
Add to Span's List of Dotted Rules



Even Larger Spans

Extend lists of dotted rules with cell constituent labels

span's dotted rule list (with same start)
plus neighboring
span's constituent labels of hypotheses (with same end)



Reflections

- Complexity $O(rn^3)$ with sentence length n and size of dotted rule list r
  - may introduce maximum size for spans that do not start at beginning
  - may limit size of dotted rule list (very arbitrary)

- Does the list of dotted rules explode?

- Yes, if there are many rules with neighboring target-side non-terminals
  - such rules apply in many places
  - rules with words are much more restricted

Difficult Rules

- Some rules may apply in too many ways

- Neighboring input non-terminals

$$\text{VP} \rightarrow \text{gibt } \text{X}_1 \text{ X}_2 \mid \text{gives NP}_2 \text{ to NP}_1$$

  - non-terminals may match many different pairs of spans
  - especially a problem for hierarchical models (no constituent label restrictions)
  - may be okay for syntax-models

- Three neighboring input non-terminals

$$\text{VP} \rightarrow \text{trifft } \text{X}_1 \text{ X}_2 \text{ X}_3 \text{ heute} \mid \text{meets NP}_1 \text{ today PP}_2 \text{ PP}_3$$

  - will get out of hand even for syntax models

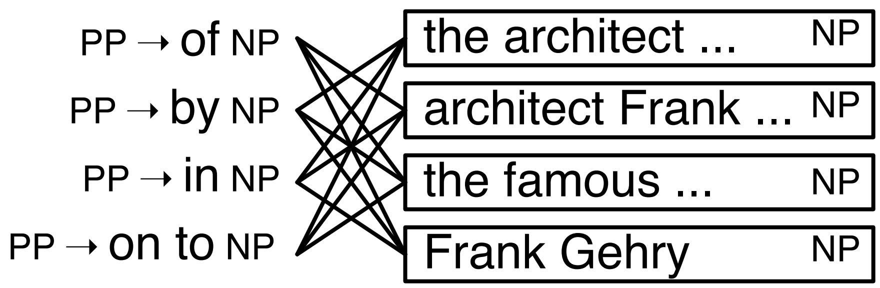
Where are we now?

- We know which rules apply
- We know where they apply (each non-terminal tied to a span)
- But there are still many choices
  - many possible translations
  - each non-terminal may match multiple hypotheses

  $\rightarrow$ number choices exponential with number of non-terminals

Rules with One Non-Terminal

Found applicable rules $\text{PP} \rightarrow \text{des X} \mid \ldots\ \text{NP}\ \ldots$



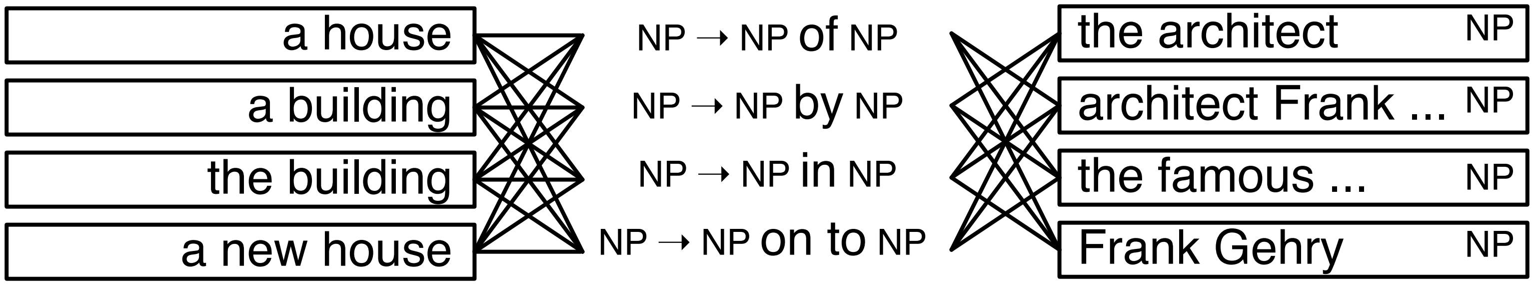
- Non-terminal will be filled any of h underlying matching hypotheses
- Choice of t lexical translations

$\Rightarrow$ Complexity $O(ht)$

(note: we may not group rules by target constituent label,
so a rule $\text{NP} \rightarrow \text{des X} \mid \text{the NP}$ would also be considered here as well)

Rules with Two Non-Terminals

Found applicable rule $\text{NP} \rightarrow \text{X}_1 \text{ des X}_2 \mid \text{NP}_1 \ \ldots \ \text{NP}_2$



- Two non-terminal will be filled any of h underlying matching hypotheses each
- Choice of t lexical translations

$\Rightarrow$ Complexity $O(h^2t)$ — a three-dimensional "cube" of choices

(note: rules may also reorder differently)

Cube Pruning

	1.5 in the ...	1.7 by architect ...	2.6 by the ...	3.2 of the ...
a house 1.0				
a building 1.3				
the building 2.2				
a new house 2.6				

Arrange all the choices in a "cube"

(here: a square, generally a orthotope, also called a hyperrectangle)

Create the First Hypothesis

	1.5 in the ...	1.7 by architect ...	2.6 by the ...	3.2 of the ...
a house 1.0	2.1			
a building 1.3				
the building 2.2				
a new house 2.6				

- Hypotheses created in cube: (0,0)

Add ("Pop") Hypothesis to Chart Cell

	1.5 in the ...	1.7 by architect ...	2.6 by the ...	3.2 of the ...
a house 1.0	2.1			
a building 1.3				
the building 2.2				
a new house 2.6				

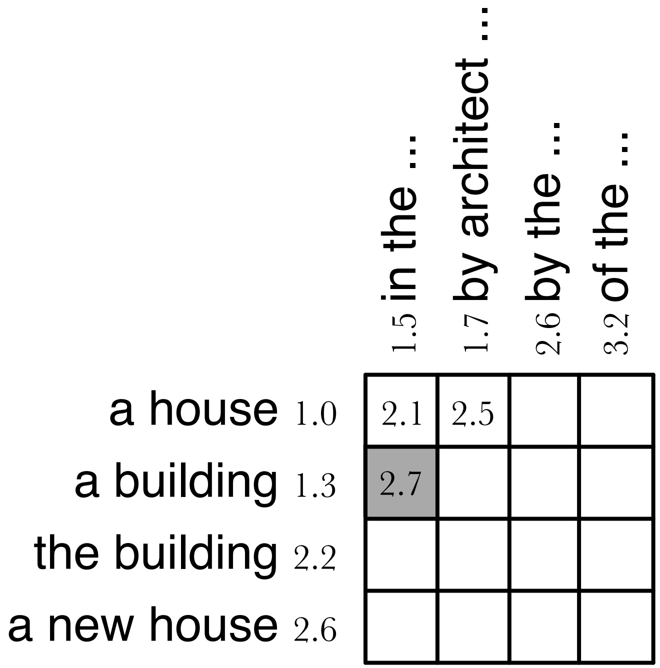
- Hypotheses created in cube: ϵ
- Hypotheses in chart cell stack: (0,0)

Create Neighboring Hypotheses

	1.5 in the ...	1.7 by architect ...	2.6 by the ...	3.2 of the ...
a house 1.0	2.1	2.5		
a building 1.3	2.7			
the building 2.2				
a new house 2.6				

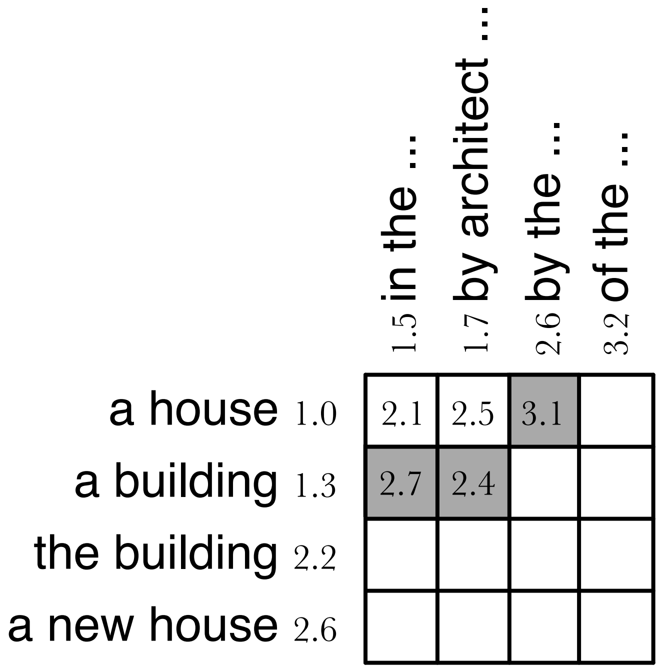
- Hypotheses created in cube: (0,1), (1,0)
- Hypotheses in chart cell stack: (0,0)

Pop Best Hypothesis to Chart Cell



- Hypotheses created in cube: (0,1)
- Hypotheses in chart cell stack: (0,0), (1,0)

Create Neighboring Hypotheses



- Hypotheses created in cube: (0,1), (1,1), (2,0)
- Hypotheses in chart cell stack: (0,0), (1,0)

More of the Same

	1.5 in the ...	1.7 by architect ...	2.6 by the ...	3.2 of the ...
a house 1.0	2.1	2.5	3.1	
a building 1.3	2.7	2.4	3.0	
the building 2.2		3.8		
a new house 2.6				

- Hypotheses created in cube: (0,1), (1,2), (2,1), (2,0)
- Hypotheses in chart cell stack: (0,0), (1,0), (1,1)

Queue of Cubes

- Several groups of rules will apply to a given span
- Each of them will have a cube
- We can create a queue of cubes

⇒ Always pop off the most promising hypothesis, regardless of cube

- May have separate queues for different target constituent labels

Bottom-Up Chart Decoding Algorithm

```
for all spans (bottom up) do
    extend dotted rules
    for all dotted rules do
        find group of applicable rules
        create a cube for it
        create first hypothesis in cube
        place cube in queue
    end for
    for specified number of pops do
        pop off best hypothesis of any cube in queue
        add it to the chart cell
        create its neighbors
    end for
    extend dotted rules over constituent labels
end for
```

Two-Stage Decoding

- First stage: decoding without a language model (-LM decoding)
  - may be done exhaustively
  - eliminate dead ends
  - optionably prune out low scoring hypotheses

- Second stage: add language model
  - limited to packed chart obtained in first stage

- Note: essentially, we do two-stage decoding for each span at a time

Coarse-to-Fine

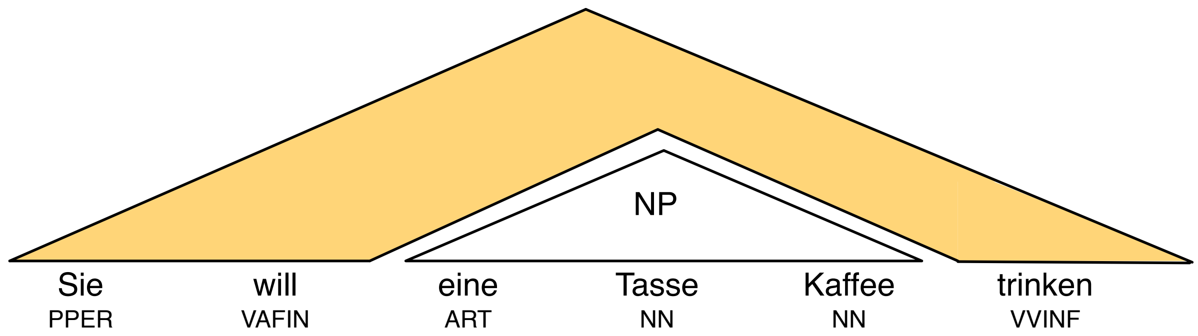
- Decode with increasingly complex model

- Examples
  - reduced language model [Zhang and Gildea, 2008]
  - reduced set of non-terminals [DeNero et al., 2009]
  - language model on clustered word classes [Petrov et al., 2008]

Outside Cost Estimation

- Which spans should be more emphasized in search?

- Initial decoding stage can provide outside cost estimates



- Use min/max language model costs to obtain admissible heuristic
(or at least something that will guide search better)

Open Questions

- Where does the best translation fall out the beam?
- How accurate are LM estimates?
- Are particular types of rules too quickly discarded?
- Are there systemic problems with cube pruning?

Summary

- Synchronous context free grammars
- Extracting rules from a syntactically parsed parallel corpus
- Bottom-up decoding
- Chart organization: dynamic programming, stacks, pruning
- Prefix tree for rules
- Dotted rules
- Cube pruning