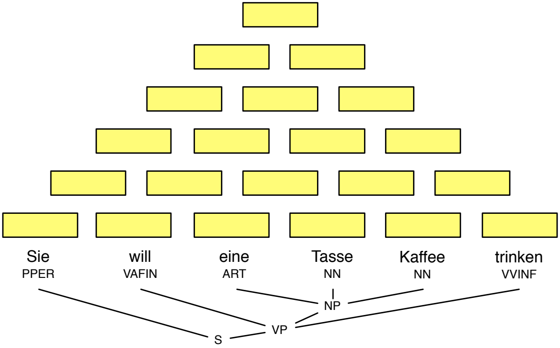
# Tree-based Models

Dr. Mariana Neves
(adapted from the original slides
of Prof. Philipp Koehn)

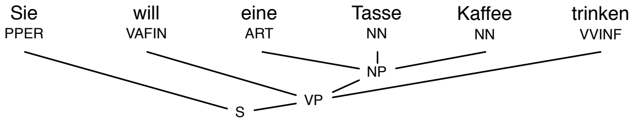
January 25th, 2016

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

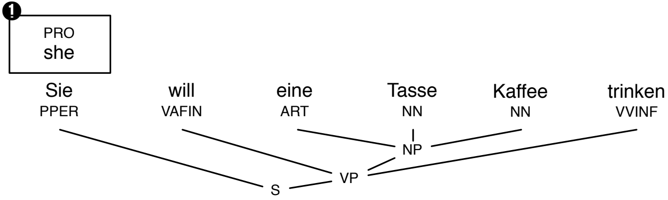


# Syntax Decoding



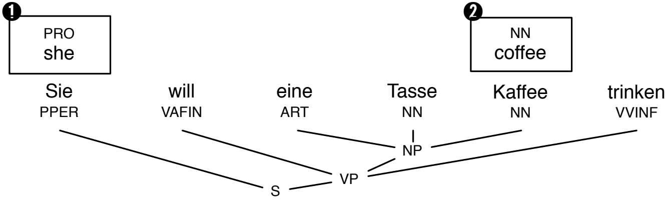

German input sentence with tree

# Syntax Decoding



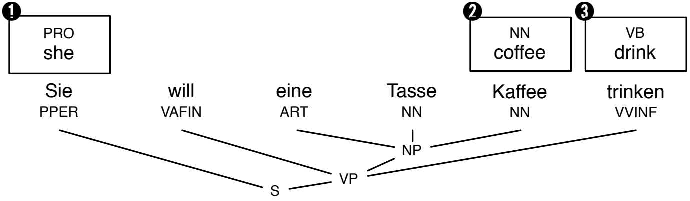

Purely lexical rule: filling a span with a translation (a constituent in the chart)

# Syntax Decoding



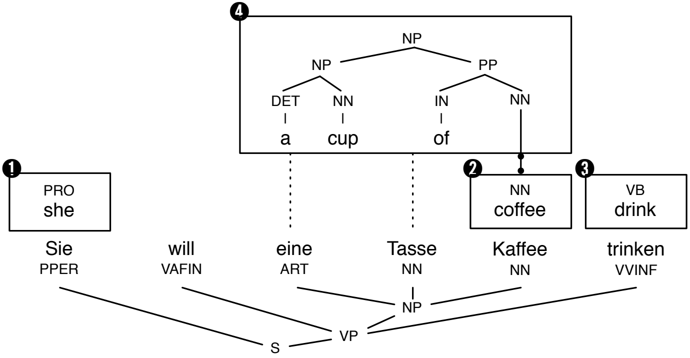

Purely lexical rule: filling a span with a translation (a constituent in the chart)

# Syntax Decoding



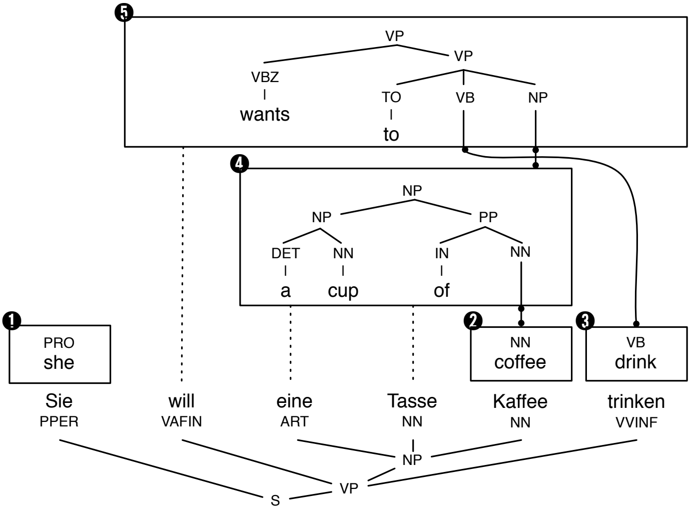

Purely lexical rule: filling a span with a translation (a constituent in the chart)

# Syntax Decoding



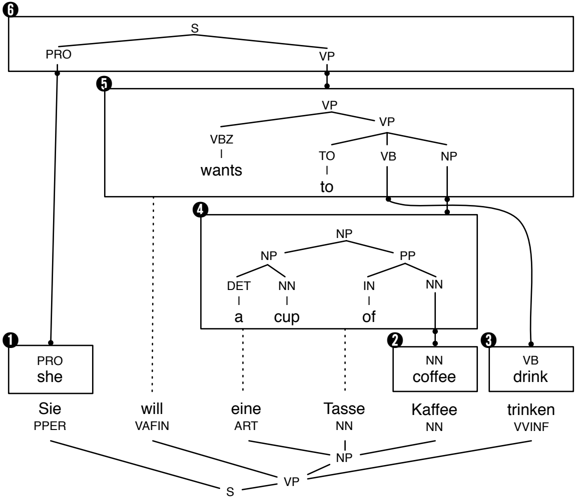

Complex rule: matching underlying constituent spans, and covering words

# Syntax Decoding



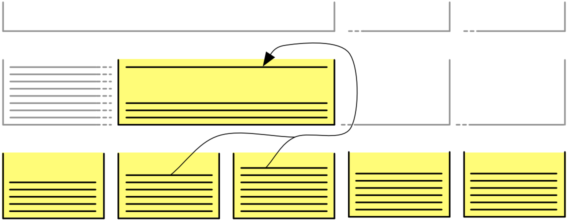

Complex rule with reordering

# Syntax Decoding



# Bottom-Up Decoding

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

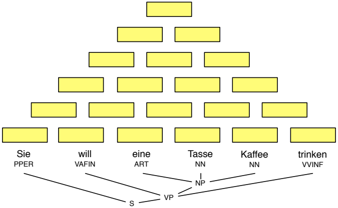


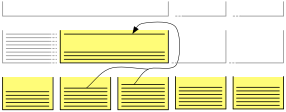
# Naive Algorithm

**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



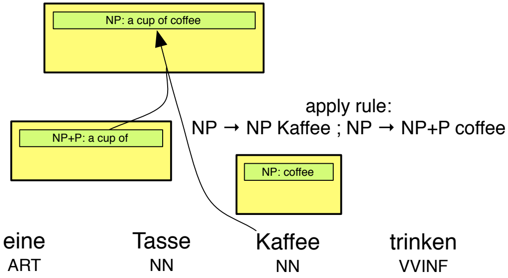



- Chart consists of cells that cover contiguous spans over the input sentence
- Each cell contains a set of hypotheses[1]
- Hypothesis = translation of span with target-side constituent

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

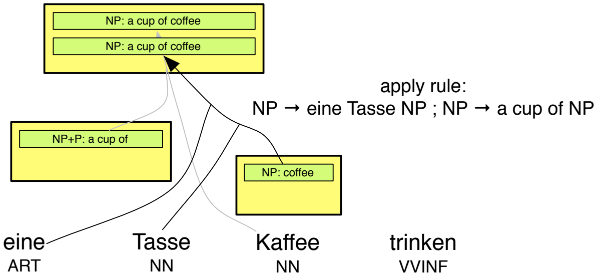
# Dynamic Programming

Applying rule creates new hypothesis



# Dynamic Programming

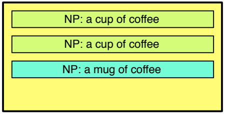
Another hypothesis




Both hypotheses are indistiguishable in future search
→ can be recombined

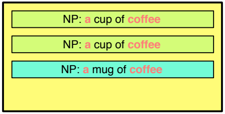
# Recombinable States

Recombinable?



# 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

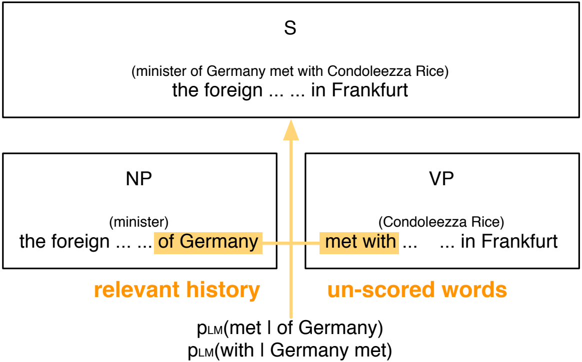
still affect scoring of subsequently added words,

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 $s$

⇒ 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

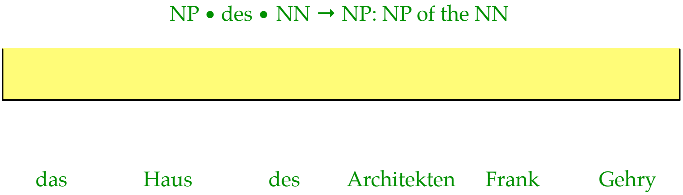
$$\text{NP} \rightarrow \text{X}_1 \text{ des X}_2 \mid \text{NP}_1 \text{ of the NN}_2$$

- 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

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

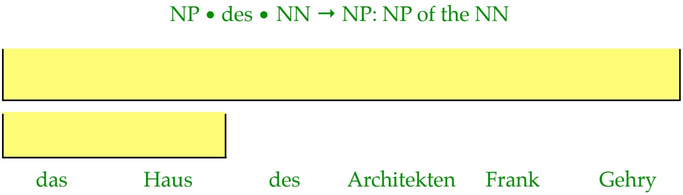
# Rule Applicability Check

Trying to cover a span of six words with given rule



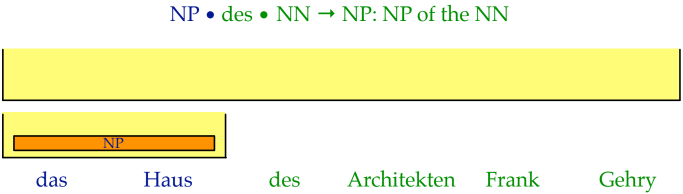
# Rule Applicability Check

First: check for hypotheses with output constituent label NP



# Rule Applicability Check

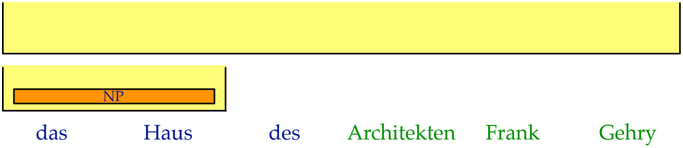
Found NP hypothesis in cell, matched first symbol of rule



# Rule Applicability Check

Matched word des, matched second symbol of rule

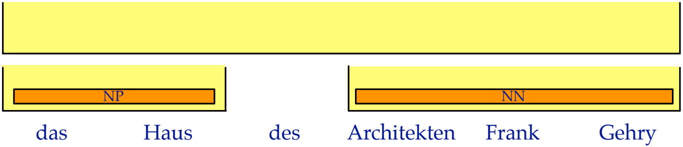
NP • des • NN → NP: NP of the NN



# Rule Applicability Check

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

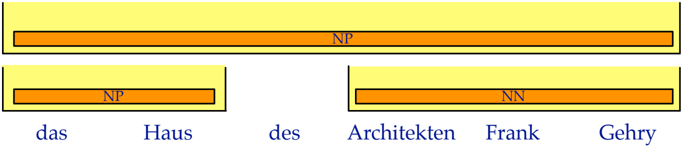
NP • des • NN → NP: NP of the NN



# Rule Applicability Check

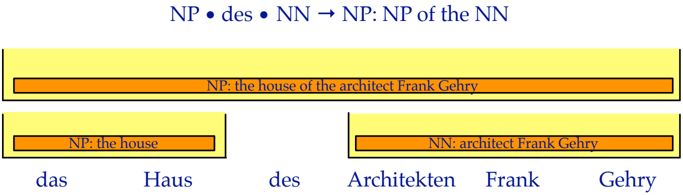
Matched entire rule → apply to create a NP hypothesis

NP • des • NN → NP: NP of the NN



# Rule Applicability Check

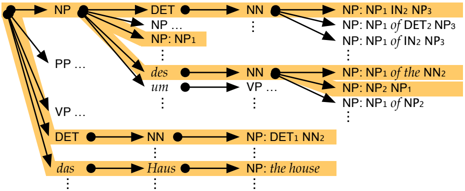
Look up output words to create new hypothesis
(note: there may be many matching underlying NP and NN hypotheses)



# 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**

$$NP \rightarrow NP_1\ DET_2\ NN_3 \mid NP_1\ IN_2\ NN_3$$
$$NP \rightarrow NP_1 \mid NP_1$$
$$NP \rightarrow NP_1\ \text{des}\ NN_2 \mid NP_1\ \text{of the}\ NN_2$$
$$NP \rightarrow NP_1\ \text{des}\ NN_2 \mid NP_2\ NP_1$$
$$NP \rightarrow DET_1\ NN_2 \mid DET_1\ NN_2$$
$$NP \rightarrow \text{das Haus} \mid \text{the house}$$

# Dotted Rules: Key Insight

- If we can apply a rule like

$$p \to \text{A B C} \mid x$$

  to a span

- Then we could have applied a rule like

$$q \to \text{A B} \mid y$$

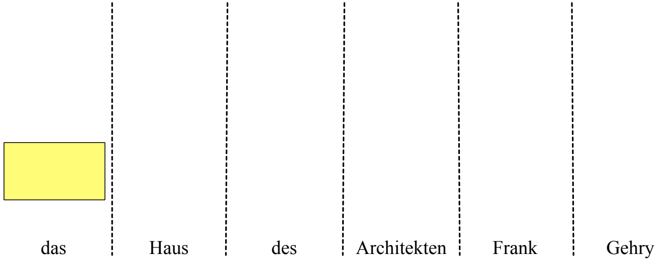
  to a sub-span with the same starting word

$\Rightarrow$ We can re-use rule lookup by storing A B • (dotted rule)

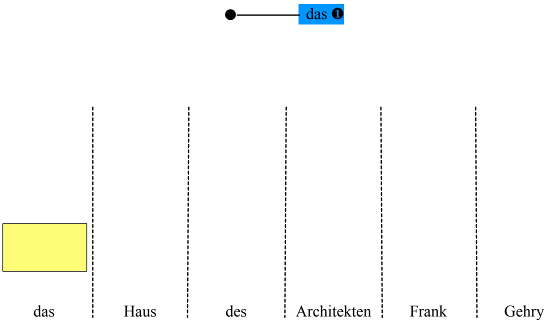
# Finding Applicable Rules in Prefix Tree

das | Haus | des | Architekten | Frank | Gehry

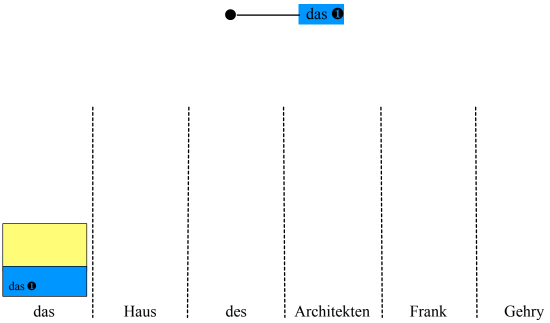
# Covering the First Cell



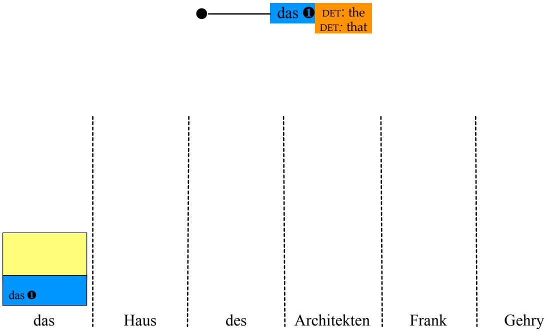
# Looking up Rules in the Prefix Tree



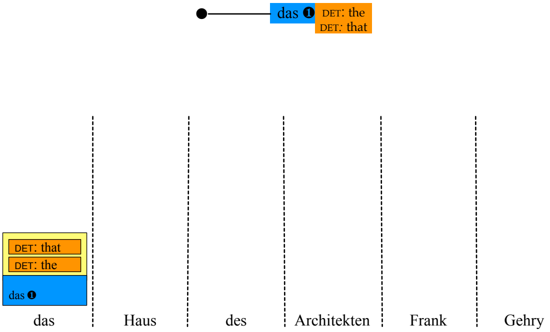
# Taking Note of the Dotted Rule



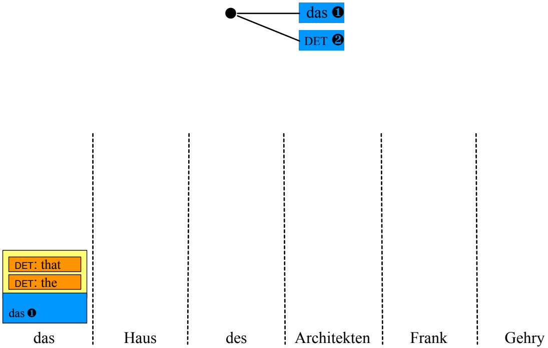
# Checking if Dotted Rule has Translations



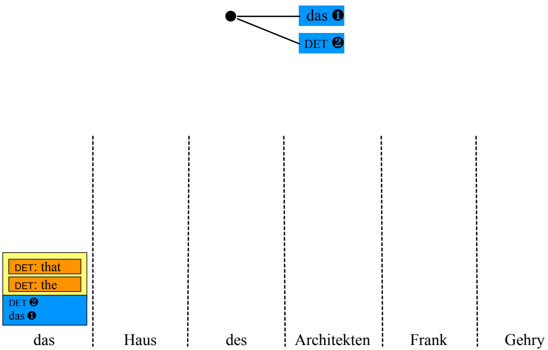
# Applying the Translation Rules



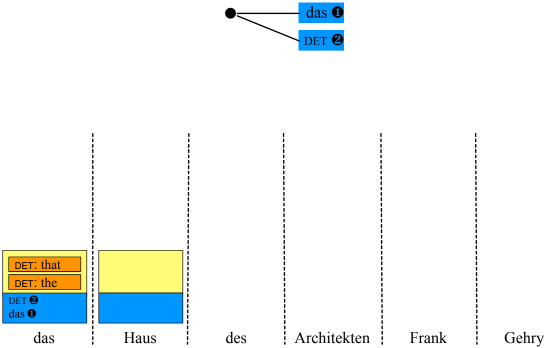
# Looking up Constituent Label in Prefix Tree



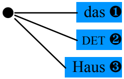
# Add to Span's List of Dotted Rules

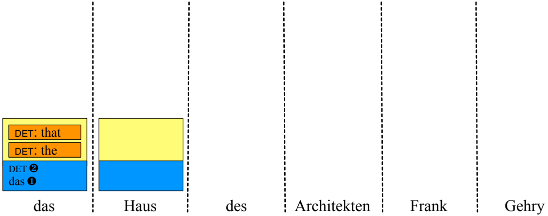


# Moving on to the Next Cell

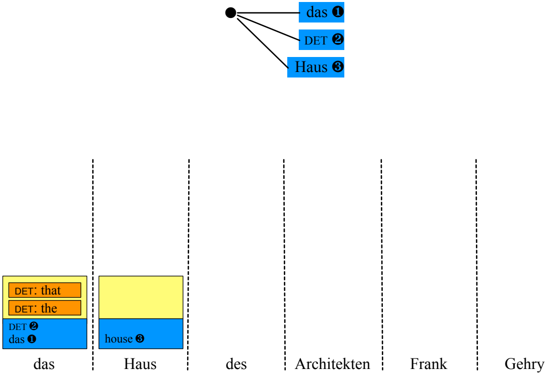


# Looking up Rules in the Prefix Tree

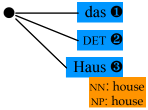


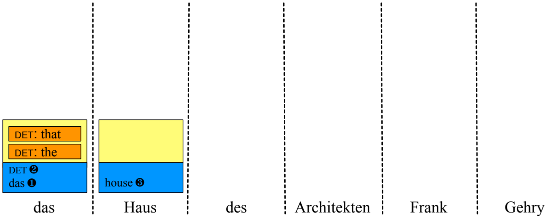



# Taking Note of the Dotted Rule

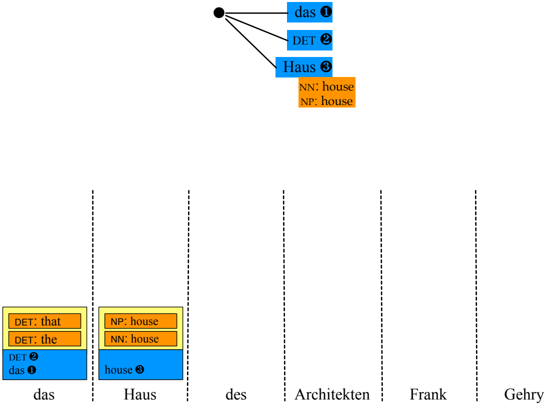


# Checking if Dotted Rule has Translations

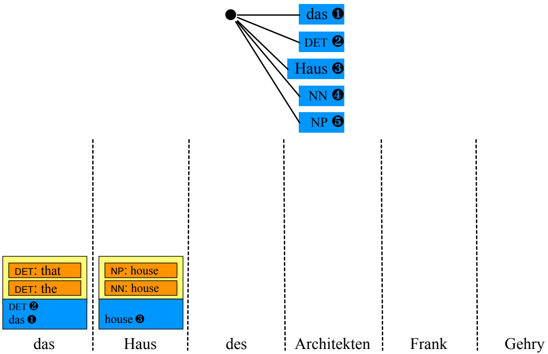


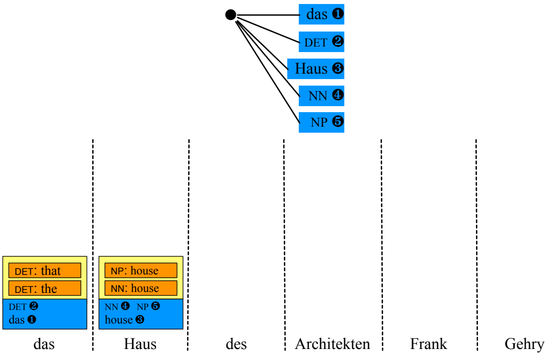
# Applying the Translation Rules



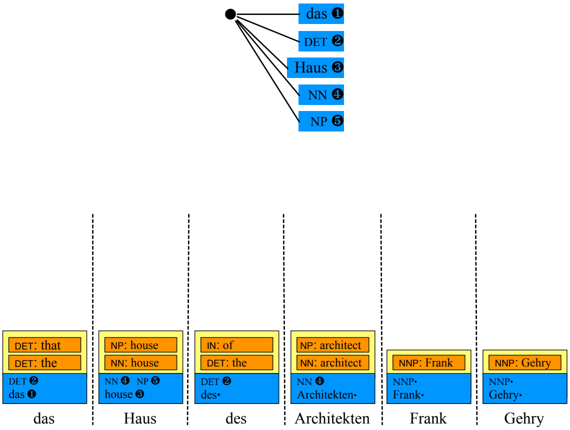
# Looking up Constituent Label in Prefix Tree



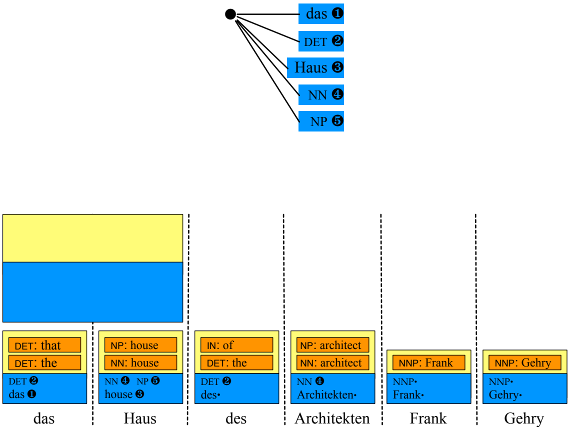
# Add to Span's List of Dotted Rules



# More of the Same



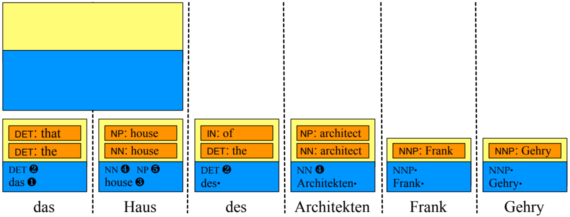
# Moving on to the Next Cell



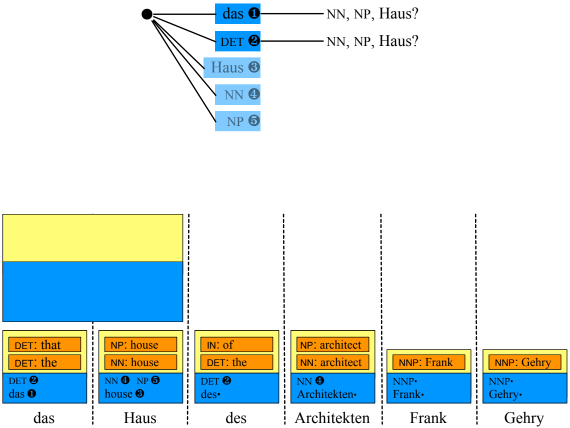
Cannot consume multiple words at once

All rules are extensions of existing dotted rules

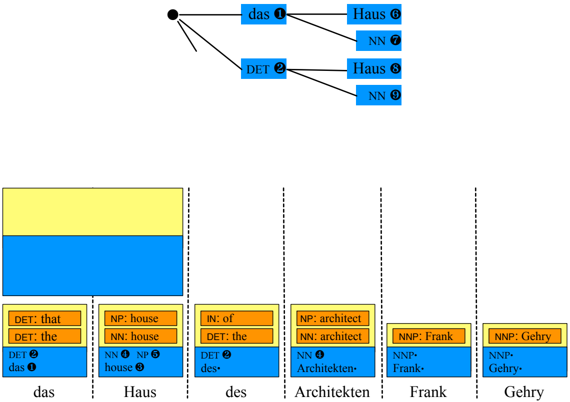
Here: only extensions of span over das possible



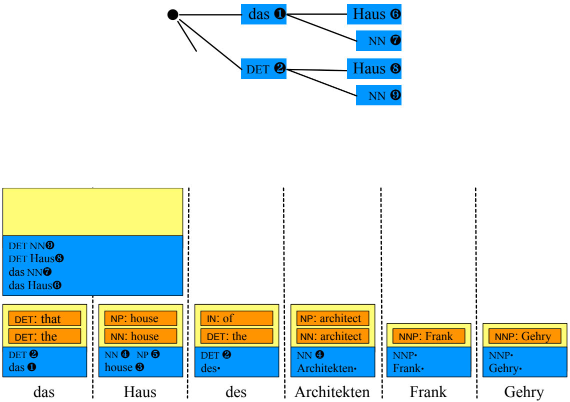
# Extensions of Span over das



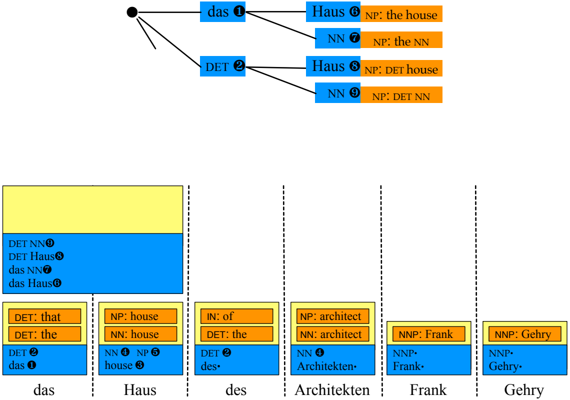
# Looking up Rules in the Prefix Tree



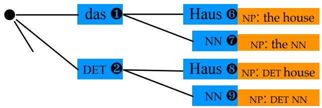
# Taking Note of the Dotted Rule

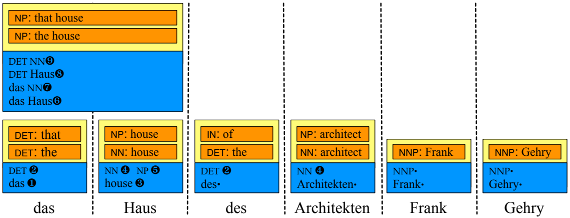


# Checking if Dotted Rules have Translations

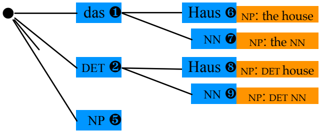


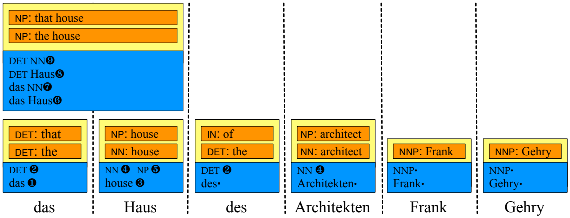
# Applying the Translation Rules



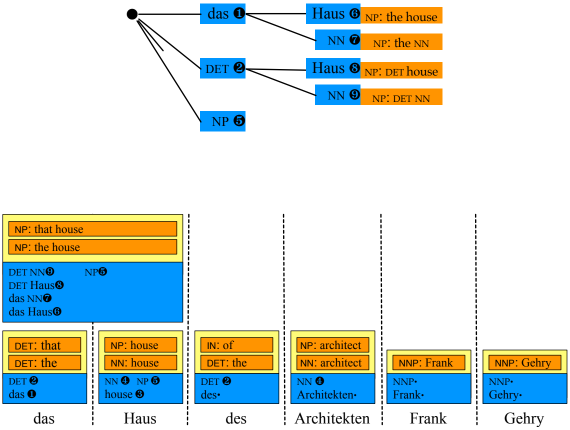



# Looking up Constituent Label in Prefix Tree



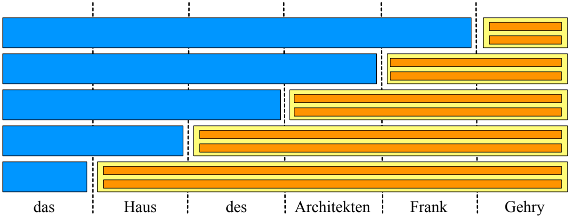
# Add to Span's List of Dotted Rules



# 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

$$\text{VP} \rightarrow \text{gibt } X_1 \ X_2 \mid \text{gives } \text{NP}_2 \text{ to } \text{NP}_1$$

  - 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

$$\text{VP} \rightarrow \text{trifft } X_1 \ X_2 \ X_3 \text{ heute} \mid \text{meets } \text{NP}_1 \text{ today } \text{PP}_2 \ \text{PP}_3$$

  - will get out of hand even for syntax models

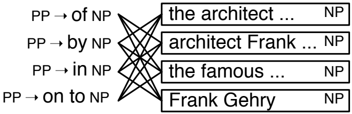
# Where are we now?

- 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

  → number choices exponential with number of non-terminals

# Rules with One Non-Terminal

Found applicable rules PP $\rightarrow$ des X | ... NP ...



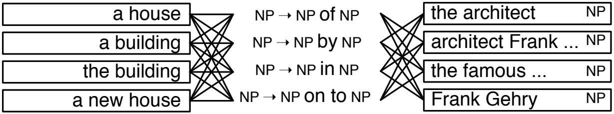

- Non-terminal will be filled any of $h$ underlying matching hypotheses
- Choice of $t$ lexical translations

$\Rightarrow$ Complexity $O(ht)$

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

# Rules with Two Non-Terminals

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

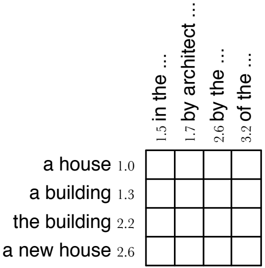



- Two non-terminal will be filled any of $h$ underlying matching hypotheses each
- Choice of $t$ lexical translations

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

(note: rules may also reorder differently)

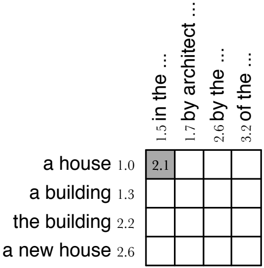
# Cube Pruning




Arrange all the choices in a "cube"

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

# Create the First Hypothesis



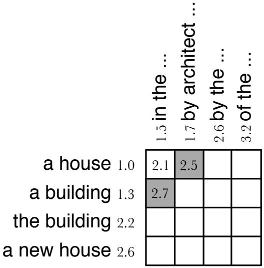

- Hypotheses created in cube: (0,0)

# 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 | | | | |

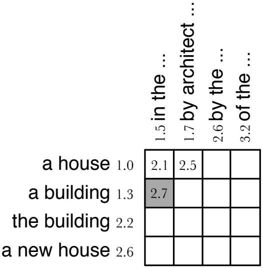
- Hypotheses created in cube: $\epsilon$
- Hypotheses in chart cell stack: (0,0)

# Create Neighboring Hypotheses




- 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

| | 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 | | |
| the building 2.2 | | | | |
| a new house 2.6 | | | | |

- 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

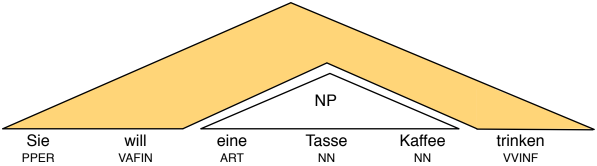
- May have separate queues for different target constituent labels

# 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
```

# Outside Cost Estimation

- Which spans should be more emphasized in search?
- Initial decoding stage can provide outside cost estimates




- For instance, use of a restricted grammar

# 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

# Suggested reading

- Statistical Machine Translation, Philipp Koehn (chapter 11).