Storing Rules

- First concern: do they apply to span?
  → have to match available hypotheses and input words
- Example rule

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

- Check for applicability
  - is there an initial sub-span that matches 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

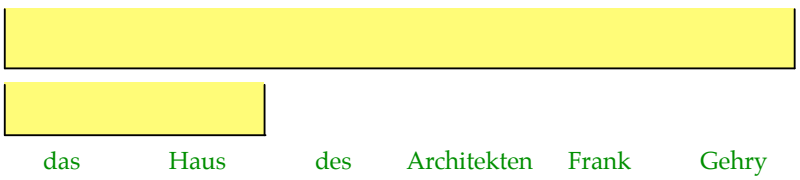
$$\text{NP} \bullet \text{des} \bullet \text{NN} \bullet \text{NP}_1 \text{ of the } \text{NN}_2$$

Check Applicability of Rule to Span

1. 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

2. Check for hypotheses with output constituent label NP
   NP • des • NN → NP: NP of the NN



das Haus des Architekten Frank Gehry

3. Found NP hypothesis in cell, matched first symbol of rule

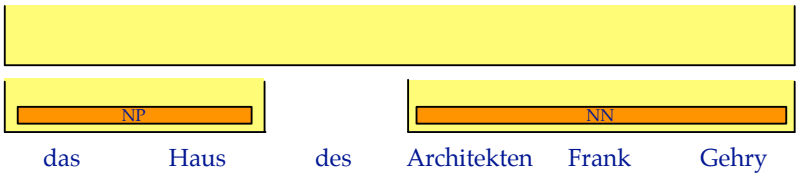
NP • des • NN → NP: NP of the NN

4. Matched word des, matched second symbol of rule

NP • des • NN → NP: NP of the NN

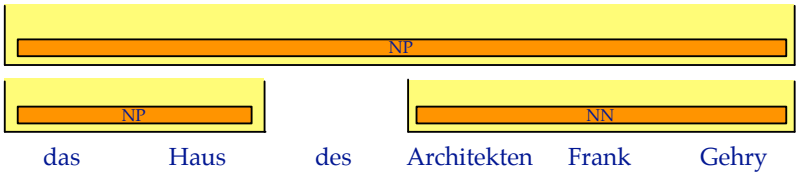
5. Found a NN hypothesis in cell, matched last symbol of rule

NP • des • NN → NP: NP of the NN



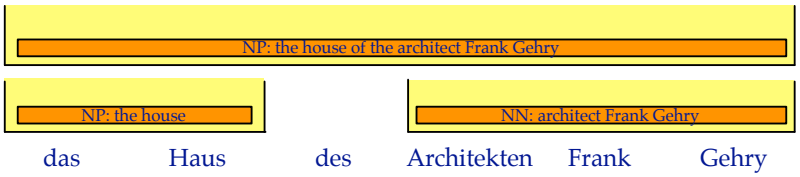
6. Matched entire rule → apply to create a NP hypothesis

NP • des • NN → NP: NP of the NN



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

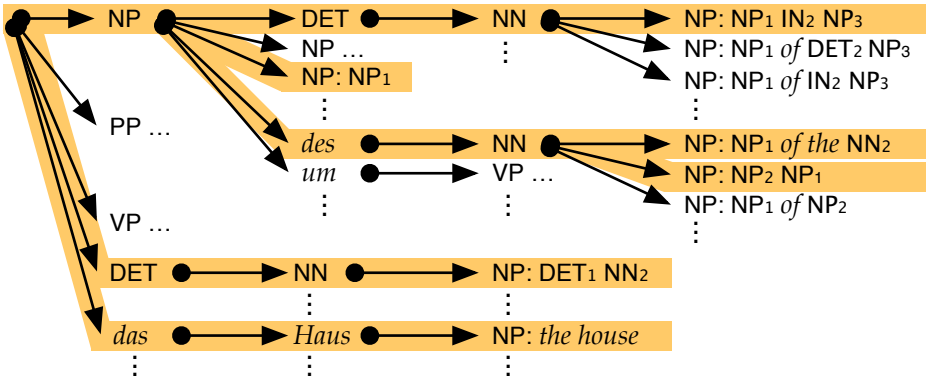
NP • des • NN → NP: NP of the NN



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

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

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

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

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

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

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

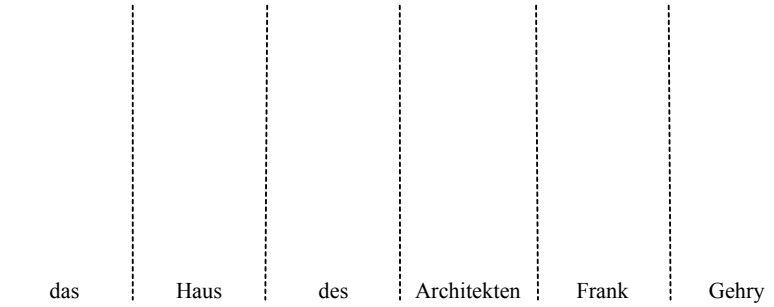
Dotted Rules: Key Insight

- If we can apply a rule like
  $$\text{p} \rightarrow \text{A B C} \mid \text{x}$$
  to a span
- Then we could have applied a rule like
  $$\text{q} \rightarrow \text{A B} \mid \text{y}$$
  to a sub-span with the same starting word

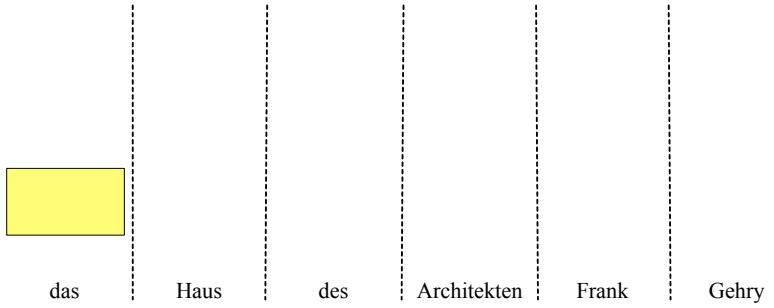
$\Rightarrow$ We can re-use rule lookup by storing $\text{A B} \bullet$ (dotted rule)

Finding Applicable Rules in Prefix Tree

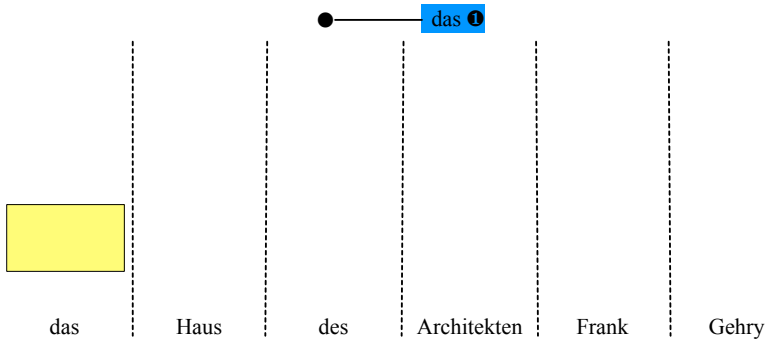
Input Sentence



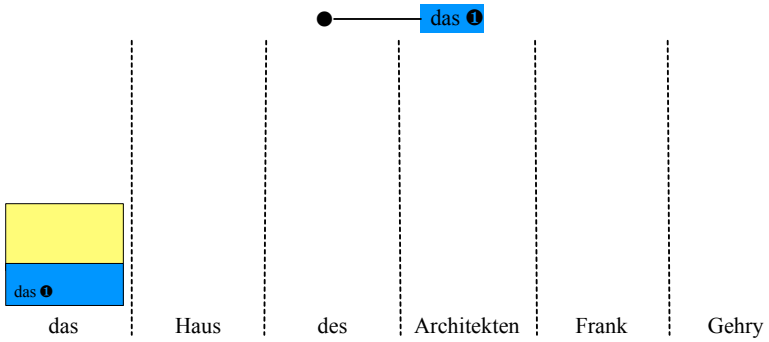
Covering the First Cell



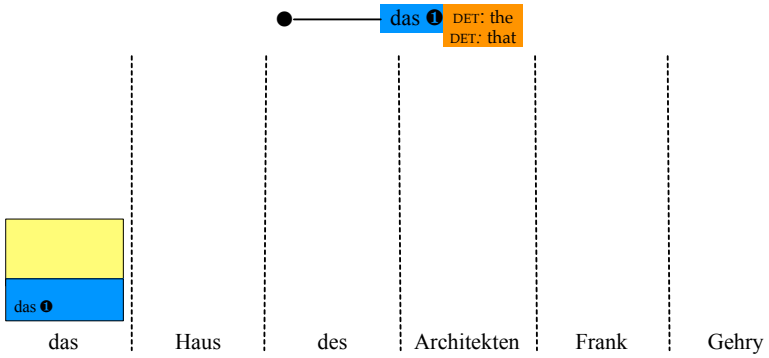
Looking up Rules in the Prefix Tree



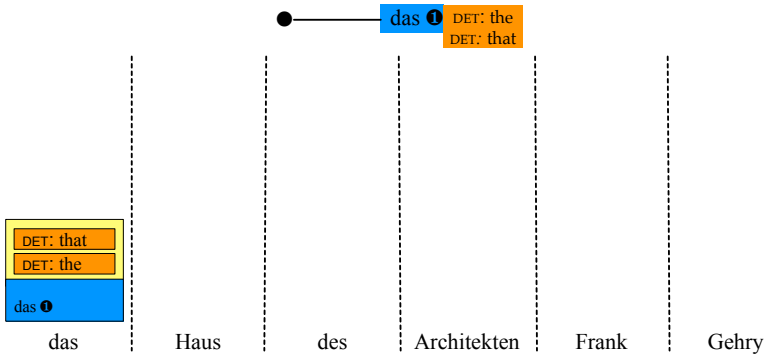
Taking Note of the Dotted Rule



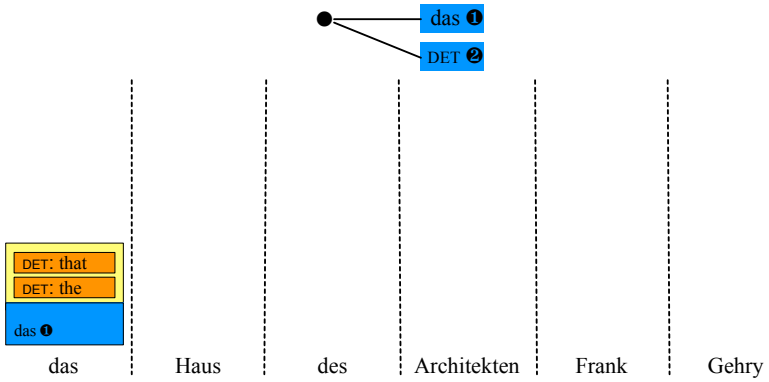
Checking if Dotted Rule has Translations



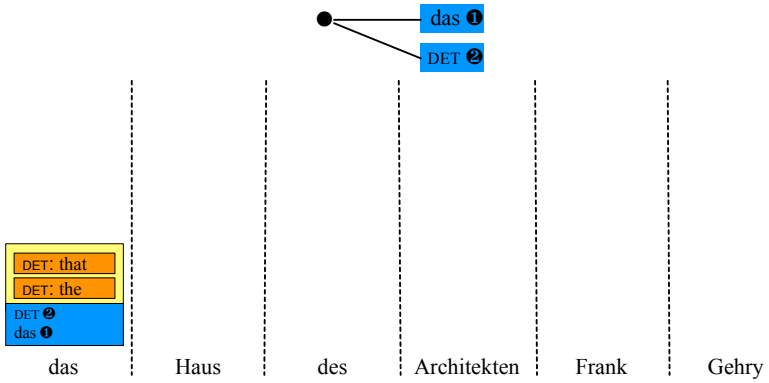
Applying the Translation Rules



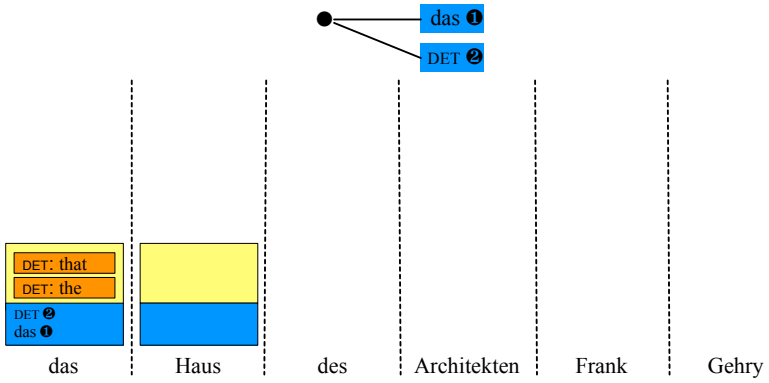
Looking up Constituent Label in Prefix Tree



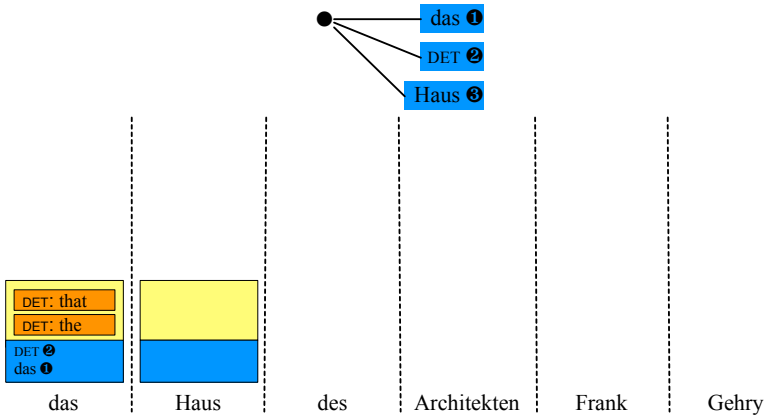
Add to Span's List of Dotted Rules



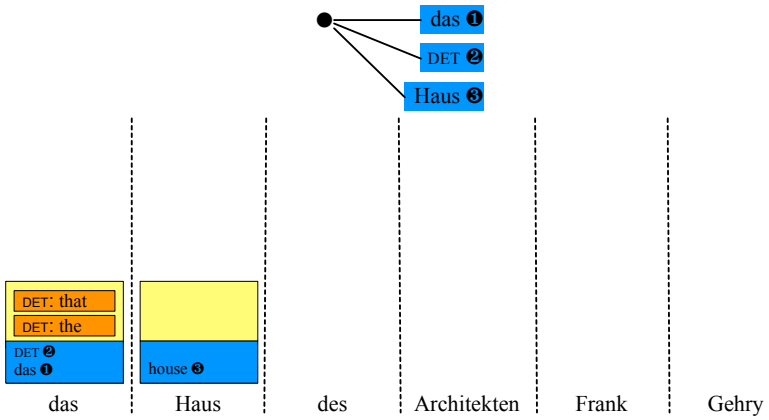
Moving on to the Next Cell



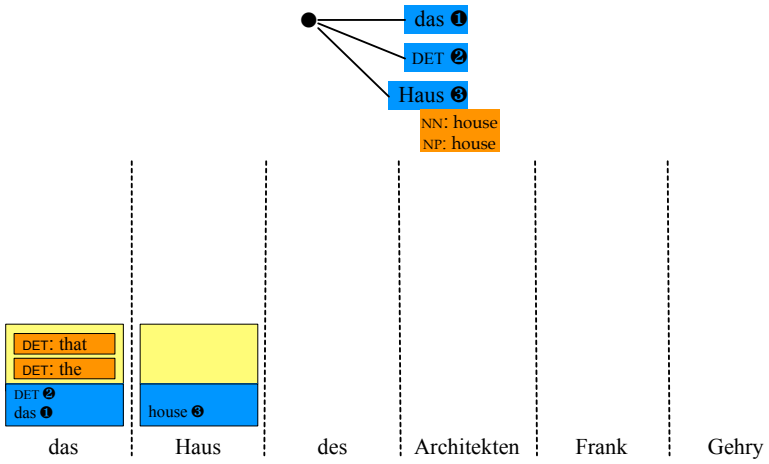
Looking up Rules in the Prefix Tree



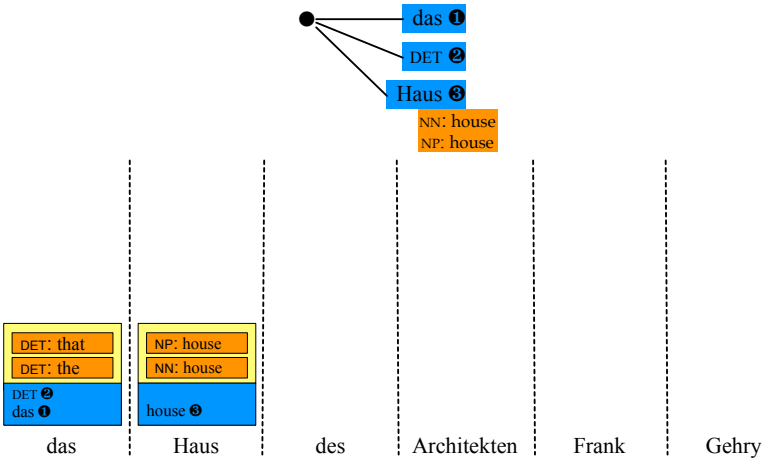
Taking Note of the Dotted Rule



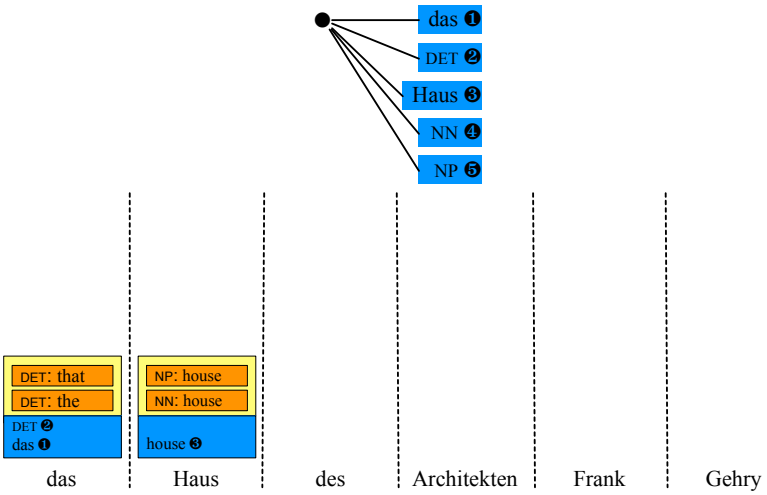
Checking if Dotted Rule has Translations



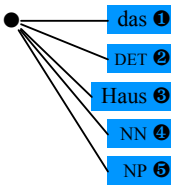
Applying the Translation Rules

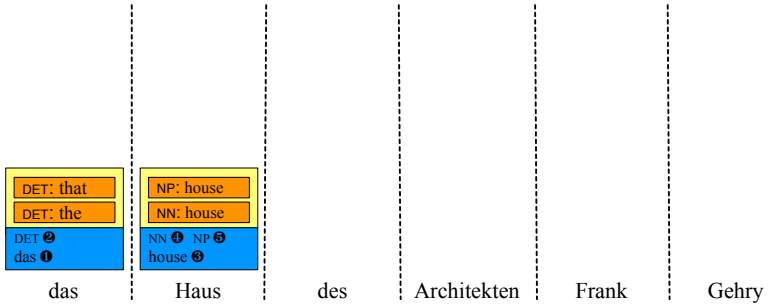


Looking up Constituent Label in Prefix Tree

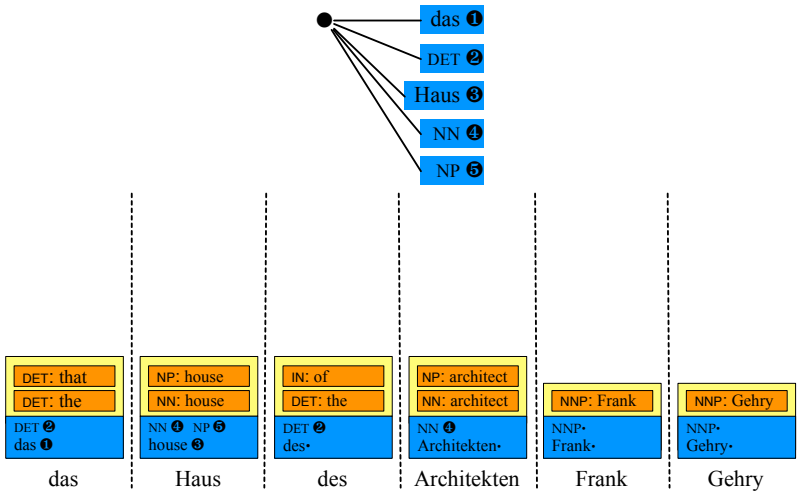


Add to Span's List of Dotted Rules

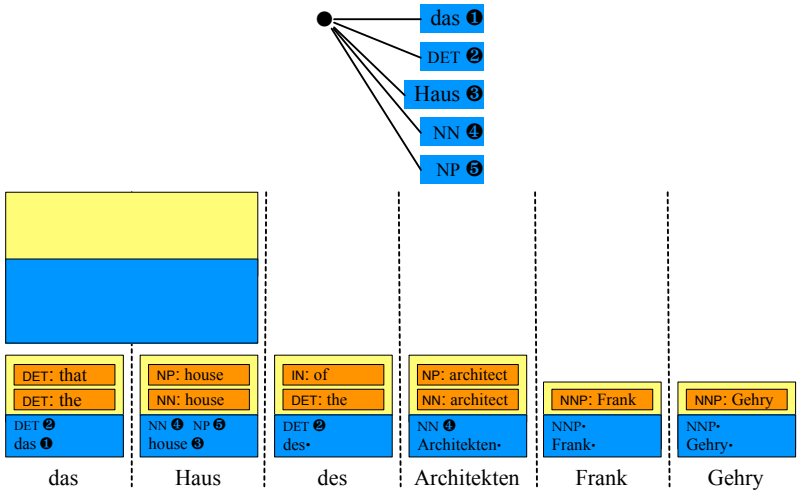




More of the Same



Moving on to the Next Cell

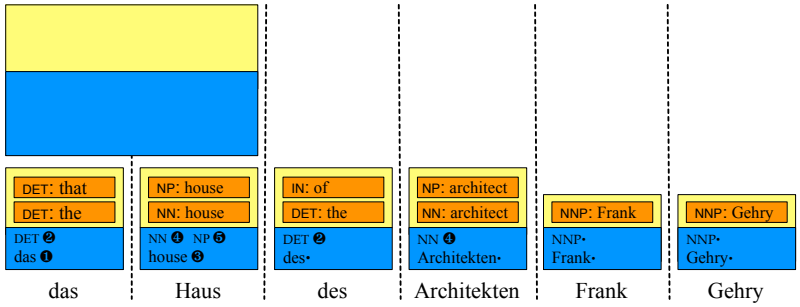


Covering a Longer Span

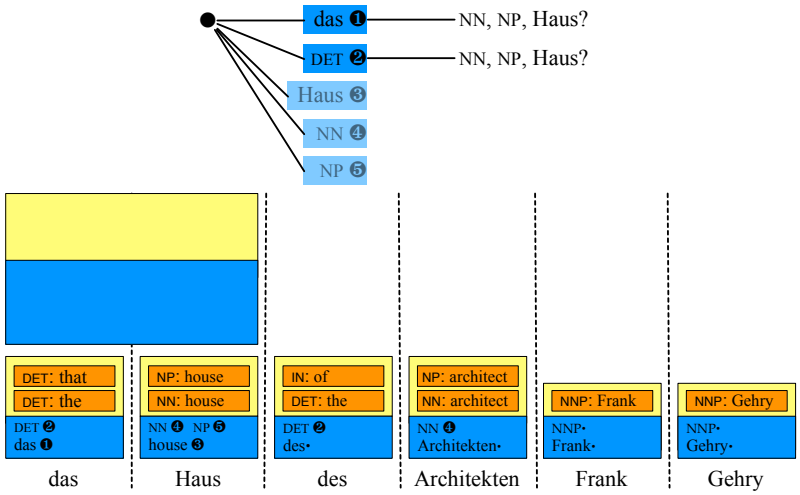
Cannot consume multiple words at once

All rules are extensions of existing dotted rules

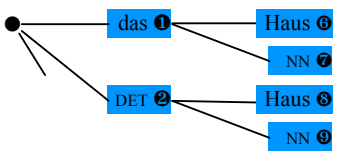
Here: only extensions of span over das possible

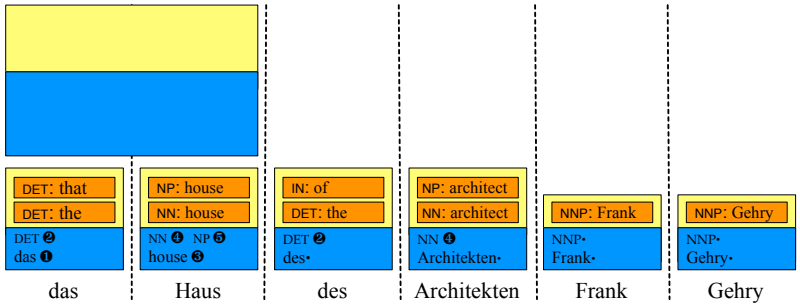


Extensions of Span over das

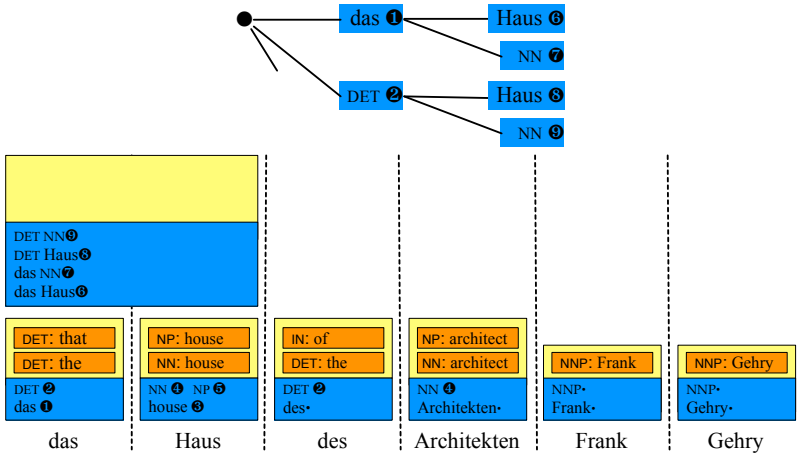


Looking up Rules in the Prefix Tree

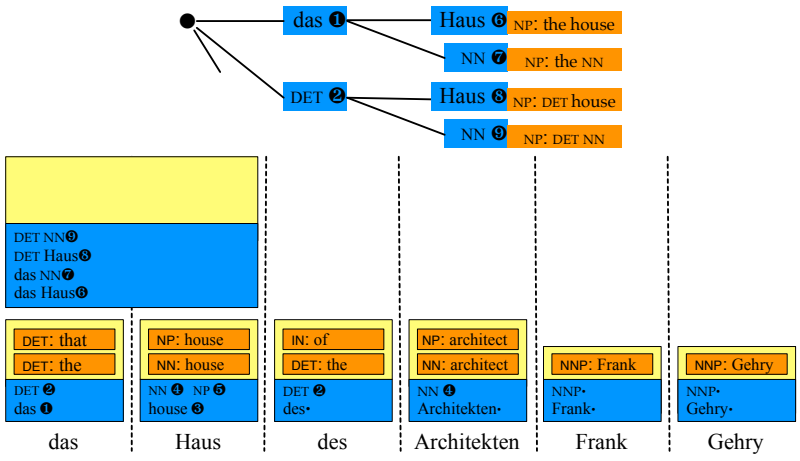




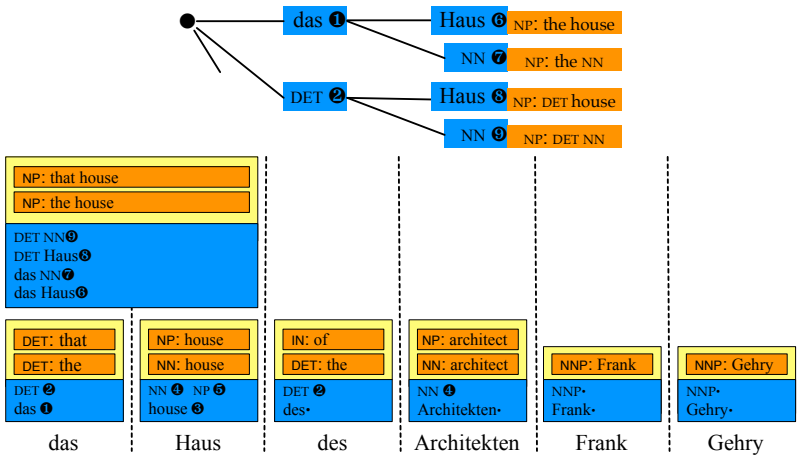
Taking Note of the Dotted Rule



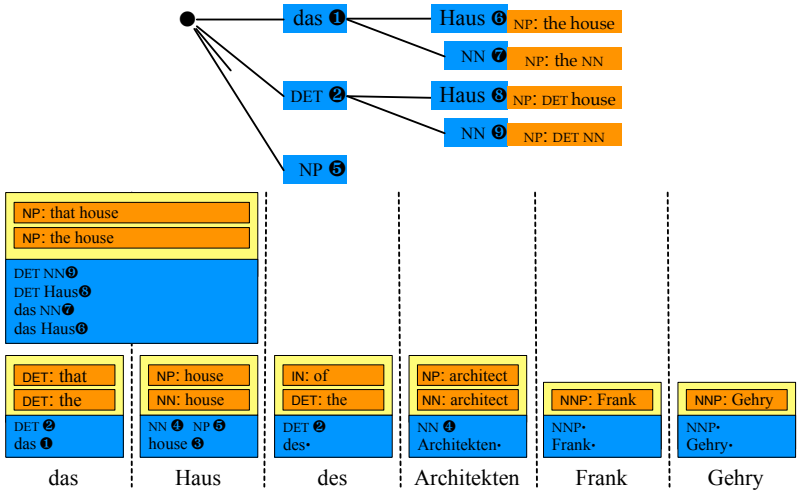
Checking if Dotted Rules have Translations



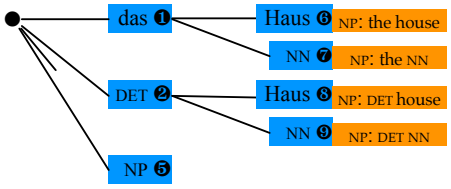
Applying the Translation Rules

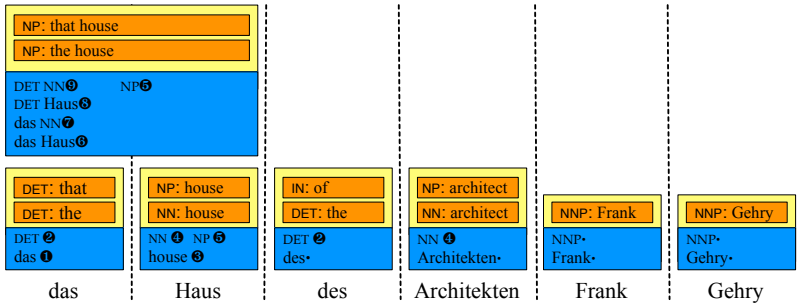


Looking up Constituent Label in Prefix Tree



Add to Span's List of Dotted Rules





Reflections

- Complexity $O(rn^3)$ with sentence length n and number dotted rules 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

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

  - 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

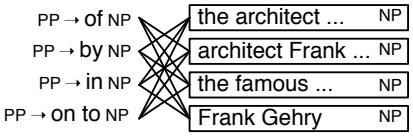
$$\text{VP} \rightarrow \text{trifft } \text{X}_1 \ \text{X}_2 \ \text{X}_3 \text{ heute} \mid \text{meets } \text{NP}_1 \text{ today } \text{PP}_2 \ \text{PP}_3$$

  - will get out of hand even for syntax models

  → number choices exponential with number of non-terminals

Rules with One Non-Terminal

Found applicable rules PP → des X | ... NP ...

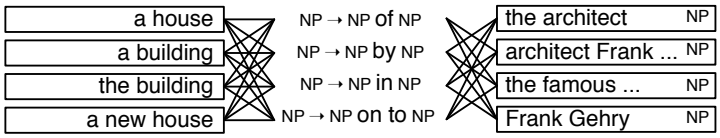


- Non-terminal will be filled any of h underlying matching hypotheses
- Choice of t lexical translations

⇒ Complexity $O(ht)$

Rules with Two Non-Terminals

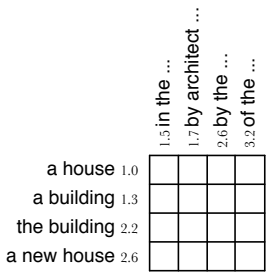
Found applicable rule NP → X_1 des X_2 | NP_1 ... NP_2



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

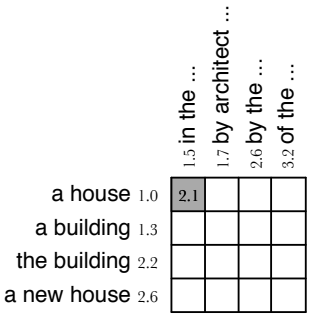
⇒ Complexity $O(h^2t)$ — a three-dimensional "cube" of choices

Cube Pruning



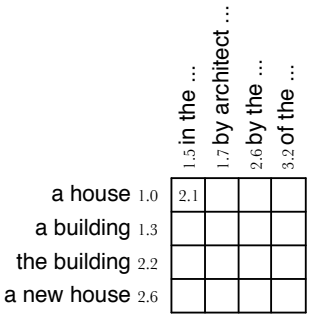
Arrange all the choices in a "cube"
(here: a square, generally a orthotope, also called a hyperrectangle)

Create the First Hypothesis



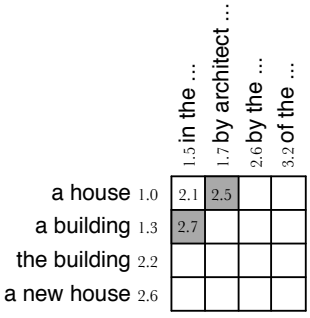
- Hypotheses created in cube: (0,0)

Add ("Pop") Hypothesis to Chart Cell



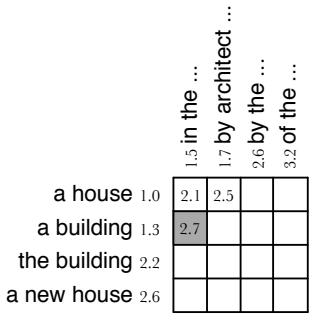
- Hypotheses created in cube: ϵ
- Hypotheses in chart cell stack: (0,0)

Create Neighboring Hypotheses



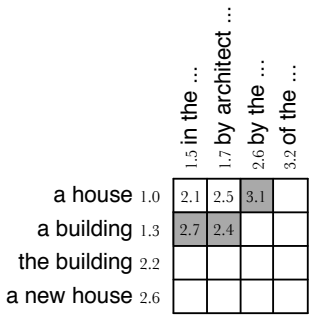
- Hypotheses created in cube: (0,1), (1,0)
- Hypotheses in chart cell stack: (0,0)

Pop Best Hypothesis to Chart Cell



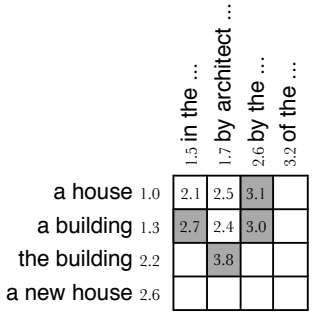
- Hypotheses created in cube: (0,1)
- Hypotheses in chart cell stack: (0,0), (1,0)

Create Neighboring Hypotheses



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

More of the Same



- 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

```
1: for all spans (bottom up) do
2:    extend dotted rules
3:    for all dotted rules do
4:       find group of applicable rules
5:       create a cube for it
6:       create first hypothesis in cube
7:       place cube in queue
8:    end for
9:    for specified number of pops do
10:      pop off best hypothesis of any cube in queue
11:      add it to the chart cell
12:      create its neighbors
13:   end for
14:   extend dotted rules over constituent labels
15: end for
```