HOBBS[1978]: Resolving Pronoun References

Presenter: Özgür Demir
Computerlinguistik, Universität des Saarlandes

13. Dezember 2002

Two approaches

- syntactic approach

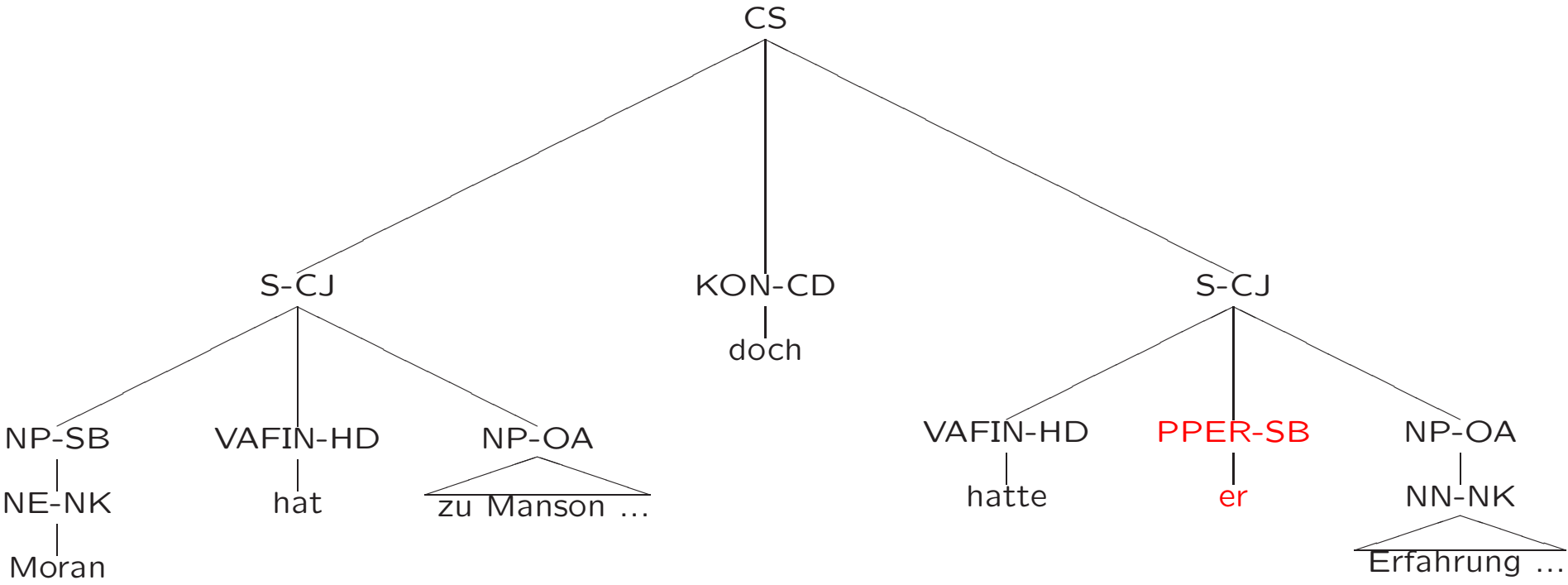
- semantic approach

Syntactic approach: The algorithm (pseudo-code)

```
(1) cn = get_mother_node(pronoun_leaf)
(2) do {
        cn = get_mother_node(cn)
    } while cn.type != {"NP","S"}
    x = cn
(3) do {
        cn = traverse(cn)
(4)     if cn == sentence.top then sentence.prev()
    } while cn != NP and not path.contains({"NP", "S"})
(5) cn = get_next_NP_or_S()
(6) if x.type == "NP" and not path.contains(get_N_daughter(x))
        then propose(x)
```

```
(7) do {
        cn = traverse(cn)
        if cn == NP then
            propose(cn)
            break
    } while traverse(cn) != ERROR
(8) if x.type == "S" then
        do {
            cn = traverse(cn)
            if cn.type == "NP" then
                propose(cn)
                break
        } while traverse(cn).type != "S"
(9) goto(4)
```

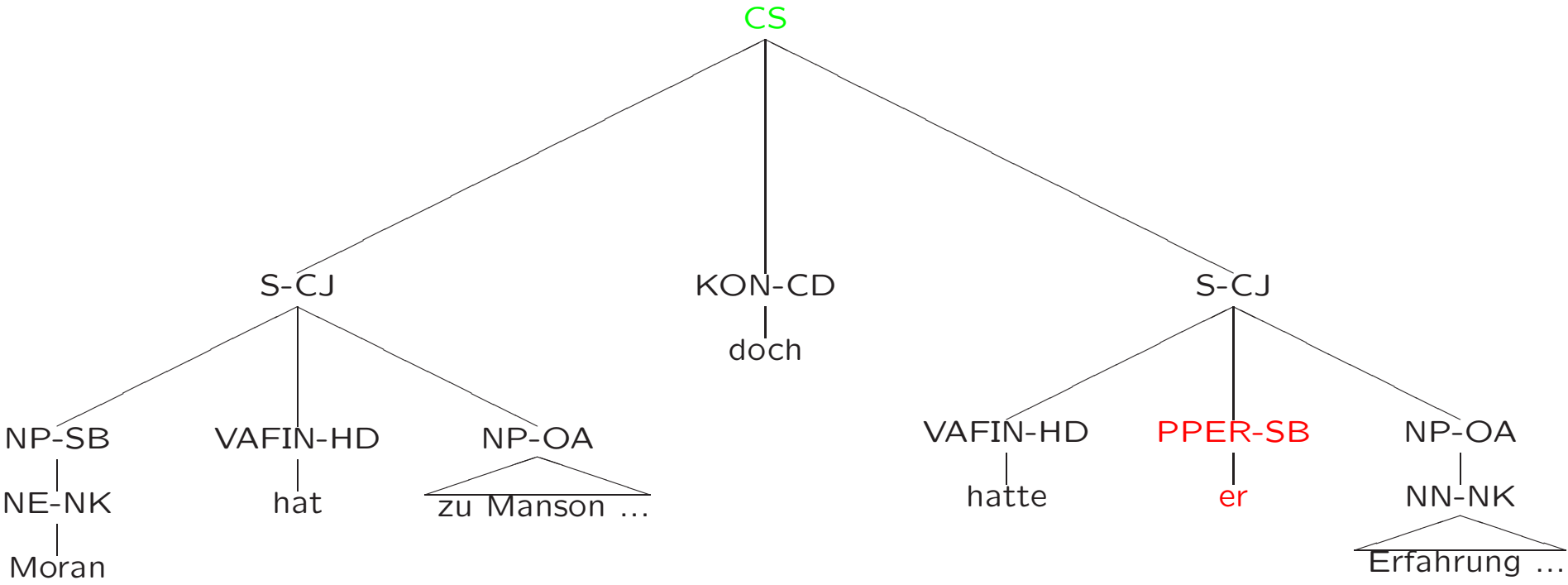
Example



red: current node

green: X

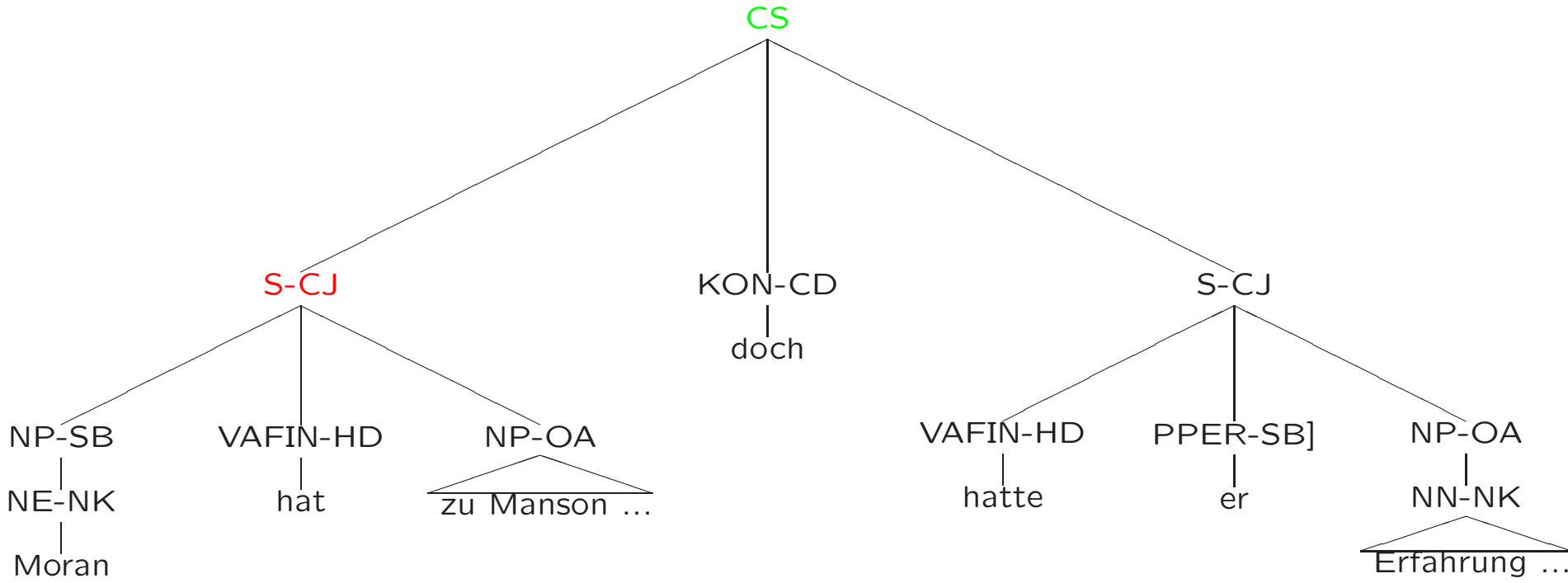
Example



red: current node

green: X

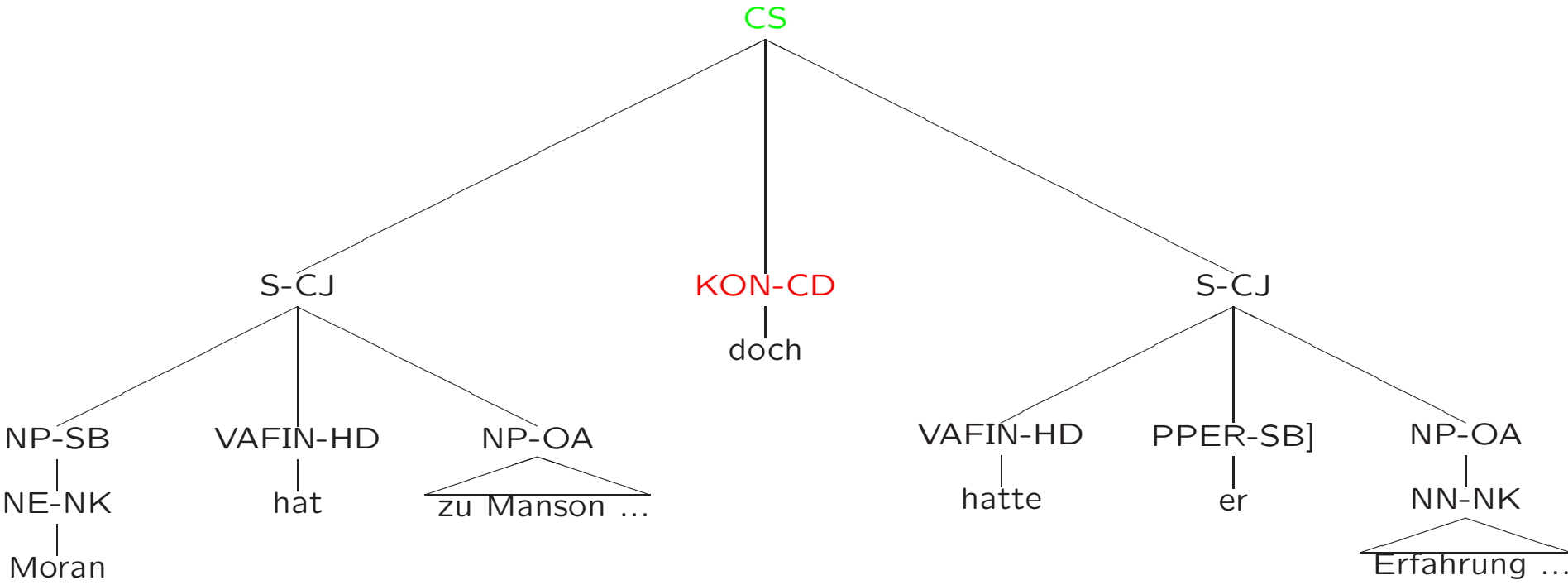
Example



red: current node

green: X

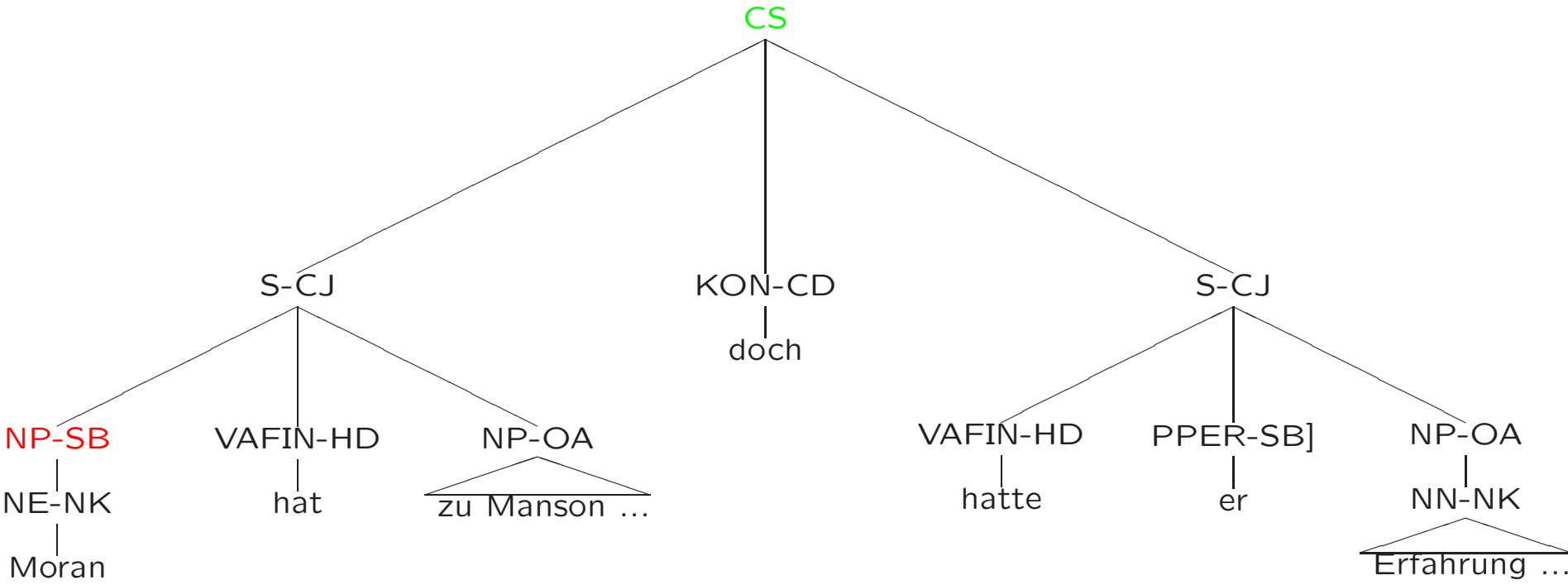
Example



red: current node

green: X

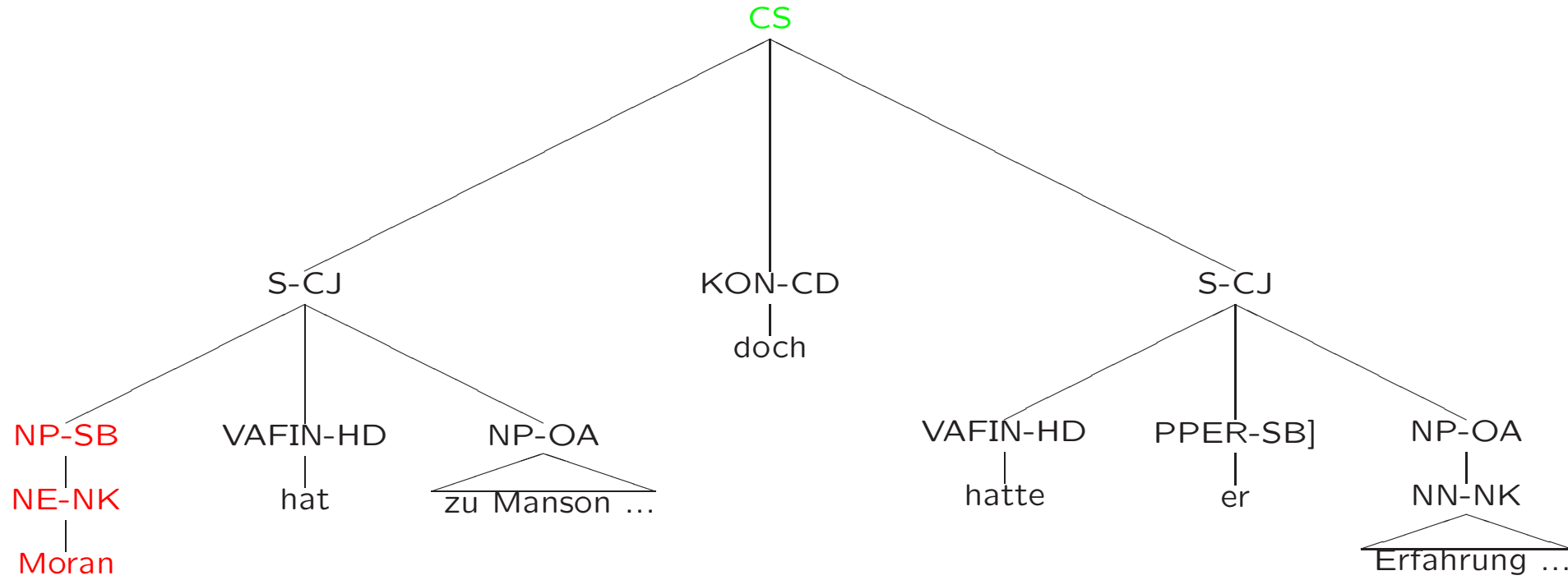
Example



red: current node

green: X

Example



red: current node

green: X

Rejection and constraints

- The proposed node can be rejected
- Implemented by adding constraints
- Examples
  - POS:PPER:"sie" cannot refer to a male name
  - SEM:LOC cannot be moved

Special case: "they" /"sie" (pl.)

- Only plural and collective sing. NPs are accepted
- Entities are collected if discontinuous

Sentence pronominalization

- SP is not handled

Peter war in Gefahr, und er wusste es.

Irreflexive pronouns

- The algorithm fails in cases like:

Peter sah ihn.

Pronoun before antecendent

- is handled by step (8)

```
(8) if x.type == "S" then
    do {
        cn = traverse(cn)
        if cn.type == "NP" then
            propose(cn)
            break
    } while traverse(cn).type != "S"
```

Als er ging, lachte Peter.

Evaluation

- Returns the correct antecedent in 88.3% of the cases
- Achieves 91.7% using selectional constraints
- Performs particularly well for "he"/"she" (93.5% without using constraints)

Implementation using the NEGRA-corpus

1. Convert NEGRA.penn to XML (penn2xml.py)

2. Parse the output with a DOM-parser

3. Use the DOM-tree to navigate through the tree

Difficult part: implementing the constraints

GAME OVER