0. Vorbemerkungen

a) Vorsicht! Der Inhalt dieser Datei wird von Zeit zu Zeit aktualisiert.

b) Die folgende Grammatik enthält grobe Hacks, die sich unter praktischen Gesichtspunkten als sinnvoll erwiesen haben. Sie enspricht aus Lesbarkeitsgründen nicht völlig der implementierten Grammatik, mit der tatsächlich gematcht wird.Ich kann nicht aussschließen, dass aus diesem Grund manche Matches (relativ zur folgenden, vereinfachten Grammatik) unverständlich sind.

c) In der Grammatik werden die Symbole *,+,?,| in der üblichen Bedeutung verwendet.

d) Das Symbol : bedeutet "Vorangehende Kategorie ist Kopf der Phrase".

1. Lexikalische Kategorien

```
n          -->  Nomen
n_gen      -->  Nomen im Genitiv        # hitchhiker's
en         -->  en+
en         -->  Eigenname
en_gen     -->  Eigenname im Genitiv    # John's
v          -->  Verb
adj        -->  Adjektiv
adv        -->  Adverb
det        -->  Determinator
possdet    -->  Possessivdeterminator    # my, your, his, ...
pron       -->  Pronomen
prep       -->  Präposition
pres_part  -->  Partizip Präsens         # walking, thinking ...
past_part  -->  Partizip Perfekt         # walked, thought
v_inf      -->  Verb im Infinitiv
cnj        -->  Konjunktion
infp       -->  Infinitivpartikel        # to
intj       -->  Interjektion
```

2. Grammatik

```
NP         -->  DetP? ANP                                    # implementiert
```

```
ANP       -->  AP? NNP                                    # NP ohne DetP # implementiert
NNP       -->  (n* n:)|en:                                # NP ohne DetP und ohne AP # implementiert
NP        -->  pron:|en:|(det? AP? n* n:)
DetP      -->  det
DetP      -->  PossNP
AdvP      -->  adv* adv:                                  # implementiert
AP        -->  AdvP? adj* adj:                            # implementiert
PossNP    -->  (det? ap? n_gen:)|en_gen:|possdet          # 'possessive NP'
Inp       -->  AdvP? v_inf: AdvP? (PP?|NP?|AP?|AdvP?)
Inp       -->  v_inf: AP AdvP? (NP|PP)?
S         -->  NP v NP? PP*                               # ungefähr; tatsächlich finden wir
                                                            Sätze einfach mittels "cnj .* \."
Sqwh      --> [wh-Frage]
Sqalt     --> [Entscheidungsfrage]
```