1 Tracing des Parsers (10 Punkte)

In der Vorlesung wurde ein Parser für kontextfreie Grammatiken vorgestellt
(siehe `http://www.dfki.de/~becker/lispkurs/vorlesungen.ss00/td-parser.lisp`).

a) Mache dich mit der Arbeitsweise des Parsers vertraut und teste die Funktionen, indem du unterschiedliche Sätze parsen läßt.

b) Schreibe eine Funktion `ptrace()`, die während des Ablaufs des Parsers Ausgaben ähnlich denen in der Vorlesung gesehenen erzeugt. Füge Aufrufe von `ptrace()` an geeigneter Stelle im Parser ein, so daß nach Aufruf von `(parse '(the woman saw the man))` die Ausgabe etwa so aussieht:

```
S --> . NP VP  [THE WOMAN SAW THE MAN]
NP --> . DET N  [THE WOMAN SAW THE MAN]
DET --> . THE  [THE WOMAN SAW THE MAN]
DET --> THE .  [WOMAN SAW THE MAN]
DET --> . A  [THE WOMAN SAW THE MAN]
NP --> (DET THE) . N  [WOMAN SAW THE MAN]
N --> . WOMAN  [WOMAN SAW THE MAN]
N --> WOMAN .  [SAW THE MAN]
N --> . MAN  [WOMAN SAW THE MAN]
N --> . BALL  [WOMAN SAW THE MAN]
N --> . SCOPE  [WOMAN SAW THE MAN]
NP --> (DET THE) (N WOMAN) .  [SAW THE MAN]
S --> (NP (DET THE) (N WOMAN)) . VP  [SAW THE MAN]
VP --> . V  [SAW THE MAN]
V --> . SNORED  [SAW THE MAN]
V --> . SAW  [SAW THE MAN]
V --> SAW .  [THE MAN]
VP --> (V SAW) .  [THE MAN]
VP --> . V NP  [SAW THE MAN]
V --> . SNORED  [SAW THE MAN]
V --> . SAW  [SAW THE MAN]
V --> SAW .  [THE MAN]
VP --> (V SAW) . NP  [THE MAN]
NP --> . DET N  [THE MAN]
DET --> . THE  [THE MAN]
DET --> THE .  [MAN]
DET --> . A  [THE MAN]
NP --> (DET THE) . N  [MAN]
N --> . WOMAN  [MAN ]
N --> . MAN  [MAN]
N --> MAN .  []
N --> . BALL  [MAN]
N --> . SCOPE  [MAN]
NP --> (DET THE) (N MAN) .  []
VP --> (V SAW) (NP (DET THE) (N MAN)) .  []
S --> (NP (DET THE) (N WOMAN)) (VP (V SAW)) .  [THE MAN]
S --> (NP (DET THE) (N WOMAN)) (VP (V SAW) (NP (DET THE) (N MAN))) . []
```

Hinweis: `ptrace()` wird an geeigneten Stellen innerhalb der Funktionen des Parsers aufgerufen und soll die benötigte Information als Parameter bekommen

2 Erweiterung des *top down* Parsers (1 + 9 Punkte)

a) Erweitere die Struktur '`parse`' um die Komponenten start und span, die (i) die Stringposition, an der eine (möglicherweise partielle) Konstituentenstruktur beginnt, und (ii) deren Länge enthalten.

   Hinweis: Stringpositionen für Parser werden typischerweise durch nullbasierte Zählung der Wortzwischenräume repräsentiert. So läßt sich die erste NP im String

   $$_0 \text{ the } _1 \text{ man } _2 \text{ snored } _3$$

   beschreiben als

   ```
   (make-parse :analyzed '(NP (Det the) (N man)) :unanalyzed '(snored) :start 0 span 2)
   ```

b) Erweitere den Parser um eine Buchführung über die aktuelle Position in der Eingabe und die Länge berechneter Ergebnisse (analog zu *start* und *span* aus Aufgabe a)). Verwende dazu die Funktionen `parse` und `extend-parse` aus
   `http://www.dfki.de/~becker/lispkurs/vorlesungen.ss00/td-parser.lisp`

Abgabetermin ist Freitag, der 26. Mai 2000