Τεχνητή Νοημοσύνη

4η Άσκηση

Δεμερτζόγλου Ευστράτιος | TH20580

Table of Contents

Περιγραφή του Προβλήματος

Η άσκηση αφορά την υλοποίηση ενός συντακτικού αναλυτή (parser) για απλές προτάσεις της αγγλικής γλώσσας χρησιμοποιώντας γραμματικούς κανόνες σε Prolog μέσω DCG (Definite Clause Grammars). Ο στόχος είναι να ελεγχθεί αν μια πρόταση είναι γραμματικά σωστή, λαμβάνοντας υπόψη τη συμφωνία αριθμού ανάμεσα στο υποκείμενο (NP) και το ρήμα (VP).

Προσέγγιση Επίλυσης

Η επίλυση του προβλήματος βασίζεται στον ορισμό μιας γραμματικής μέσω DCG σε Prolog. Οι βασικοί κανόνες καλύπτουν την παραγωγή προτάσεων (S), ονομάτων (NP), και ρημάτων (VP), με συμφωνία στον αριθμό. Ο κώδικας επιτρέπει τόσο πλήρεις φράσεις με άρθρα όσο και φράσεις χωρίς άρθρα, καλύπτοντας βασικές μορφές προτάσεων στην αγγλική γλώσσα.

Επεξήγηση Κώδικα

1. Κανόνας Πρότασης (Sentence - S)

```
s --> np(Number), vp(Number).
```

% Απαιτεί συμφωνία αριθμού μεταξύ του υποκειμένου (NP) και του ρήματος (VP).

2. Ονοματική Φράση (Noun Phrase - NP)

```
np(Number) --> det(DetType, Number), n(Number).
np(Number) --> n(Number).
```

% Υποστηρίζεται είτε ως άρθρο και ουσιαστικό είτε μόνο ως ουσιαστικό:

3. Ρηματική Φράση (Verb Phrase - VP)

```
vp(Number) --> v(Number), np(_).
vp(Number) --> v(Number).
```

% Υποστηρίζονται και μεταβατικά και αμετάβατα ρήματα:

4. Οριστικά Άρθρα (Determiners - Det)

```
% Det μπορεί να είναι "a", "the", ή τίποτα (εφόσον το επόμενο είναι ουσιαστικό)
det(a, sg)   --> [a].         % 'a' μόνο με ενικό
det(the, _)  --> [the].
det(none, _) --> [].          % 'τίποτα' ως Det
```

% Γίνεται διάκριση μεταξύ a, the και απουσίας άρθρου:

5. Ουσιαστικά (Nouns - N)

```
n(sg) --> [dog].
n(sg) --> [cat].
n(sg) --> [boy].
n(sg) --> [girl].
n(pl) --> [dogs].
n(pl) --> [cats].
n(pl) --> [boys].
n(pl) --> [girls].
```

%Γίνεται διάκριση ενικού και πληθυντικού:

6. Ρήματα (Verbs - V)

```
v(sg) --> [chases].
v(sg) --> [sees].
v(sg) --> [says].
v(sg) --> [believes].
v(pl) --> [chase].
v(pl) --> [see].
v(pl) --> [say].
v(pl) --> [believe].
```

% Υποστηρίζεται και ο ενικός και ο πληθυντικός:

Παράδειγμα Εκτέλεσης

?- phrase(s,[the,dogs,chases,cats]).

false.

?- phrase(s,[the,dogs,chase,cats]).

true.

?- phrase(s,[boy,sees,girl]).

true.

?- phrase(s,[a,boy,say]).

false.

Κώδικας σε Prolog που εκτελέστηκε

```
% Η βασική πρόταση: S -> NP VP, όπου το NP και το ρήμα στη VP πρέπει να συμφωνούν σε αριθμό
s --> np(Number), vp(Number).

% NP -> Det N
np(Number) --> det(DetType, Number), n(Number).
% NP μπορεί να είναι και μόνο το ουσιαστικό (χωρίς Det)
np(Number) --> n(Number).

% VP -> V NP | V
vp(Number) --> v(Number), np(_).     % Ο αριθμός του αντικειμένου αδιάφορος
vp(Number) --> v(Number).

% Det μπορεί να είναι "a", "the", ή τίποτα (εφόσον το επόμενο είναι ουσιαστικό)
det(a, sg)   --> [a].         % 'a' μόνο με ενικό
det(the, _)  --> [the].
det(none, _) --> [].          % 'τίποτα' ως Det

% Ουσιαστικά, με ετικέτα αριθμού
n(sg) --> [dog].
n(sg) --> [cat].
n(sg) --> [boy].
n(sg) --> [girl].
n(pl) --> [dogs].
n(pl) --> [cats].
n(pl) --> [boys].
n(pl) --> [girls].

% Ρήματα, με ετικέτα αριθμού
v(sg) --> [chases].
v(sg) --> [sees].
v(sg) --> [says].
v(sg) --> [believes].
v(pl) --> [chase].
v(pl) --> [see].
v(pl) --> [say].
v(pl) --> [believe].
```