# COURSEWORK 2 CSC-325 Artificial Intelligence Dr Adam Wyner

## **15 March 2022**

**Due: Thursday 4 April 2022, 11am. This is a hard deadline and set in relation to other submission deadlines.**

Read and think through the whole coursework before starting to program.

Follow the instructions fully and accurately. Marks are taken off for wrong or missing answers.

# Submitting your coursework

1. Return your coursework in one zipped file (.zip) that contains your one Prolog file (.pl).

Your filenames must be in the following format, for example:

CSC-325\_2021\_Coursework2\_YOURSTUDENTNUMBER.pl

1. Submit the coursework file on the Canvas Assignments link.
2. The Prolog file should include:
   1. All your Prolog code, that is, the grammar rules and the lexicon.

We should be able to execute and query your code to assess it;

* 1. In comments, include the input query for each test sentence (below) along with the output parse.

1. While there may be different ways to write the grammar and lexicon, the output must be precise and fixed, as the results may be automatically checked. If the output does not match the intended output, you be marked down on that output.

**Address the following in the coursework.**

1. Write a DCG which parses input sentences and outputs a parse.
2. The DCG must parse sentences with the following features, some of which appeared earlier in lectures or labs, while others are explained further below:

* 1. Transitive and intransitive verbs
  2. Common nouns
  3. Determiners (e.g., "a", "the", and "two")
  4. Subject/object singular/plural pronouns (e.g., "he", "him", "I", "we")
  5. Pronouns with grammatical person (e.g., "I", "you", "she")
  6. Singular/plural nouns (e.g., chair/chairs)
  7. Adjectives and adjectival phrases
  8. Prepositions and prepositional phrases
  9. Subject-verb agreement for person and number
  10. Determiner-common noun number agreement
  11. Animacy agreement between the subject and verb

1. The DCG should make use of separate grammar and lexicon. The lexicon must be included in your code and include all the words and word forms in The Lexicon below as well as additions for animacy.
2. The sample outputs (below) should be carefully studied and emulated by the parser. The categories, e.g., nbar, jp, adj, n, and parsing structure, e.g., nbar(jp(adj(tall),

jp(adj(young), n(men)))), should appear in the output. Output that misses categories and parsing structures will be marked down.

# Demonstration of Work

To demonstrate that your code works as intended, your code should correctly recognise and give the right parse trees for the grammatical sentences in the list of **test sentences** (below) and fail for the ungrammatical sentences. In the list of test sentences below, an ungrammatical sentence is indicated with a \* next to it, for example, "\*the men sees the apples" is ungrammatical. We are not concerned with capitalisation or punctuation. For each sentence in the list of test sentences, query your parser; if the sentence is ungrammatical, the output should be *false/fails*; if the sentence is grammatical, the outputs should be the correct parse. Copy the query and the output (parse or fails/false) into a commented section of your code. See the emphasised point about sample outputs below. Testing your parser with the test sentences is required. However, it would be wise to create further test sentences of your own.

Generally, if issues or problems arise, report these in a discussion section.

Your grammar should parse every sentence. Some of the sentences show animacy features. Intuitively, we understand that people are animate (i.e., act under their own volition), while objects such as apples, chairs, and rooms are inanimate (i.e., do not act under their own volition, but are acted upon). We see in the data that some verbs only allow animate subjects, while other verbs can take either animate or inanimate subjects. Your lexicon and parser must correctly parse such sentences. Note: you should provide one grammar and one lexicon that parses all the sentences.

While much of the lexicon is provided below, you will need to augment it with entries that reflect Animacy.

Points are taken off if there are missing or incorrect parses or if there is no distinct lexicon, i.e. separate from the grammar (as per the lecture).

If a sentence parses, do not provide more than one sample parse of a sentence. Check that it is correct.

If you have very long parses, then something is wrong with your grammar. Manually check that your parses are correct and make sense.

The numbering/order of the tests and parses must follow the numbering/order of the test sentences.

# Test sentences

1. the woman sees the apples
2. a woman knows him
3. \*two woman hires a man
4. two women hire a man
5. she knows her
6. \*she know the man
7. \*us see the apple
8. we see the apple
9. i know a short man
10. \*he hires they
11. two apples fall
12. the apple falls
13. the apples fall
14. i sleep
15. you sleep
16. she sleeps
17. \*he sleep
18. \*them sleep
19. \*a men sleep
20. \*the tall woman sees the red
21. the young tall man knows the old short woman
22. \*a man tall knows the short woman
23. a man on a chair sees a woman in a room
24. \*a man on a chair sees a woman a room in
25. the tall young woman in a room on the chair in a room in the room sees the red apples under the chair.
26. the woman sees the apples
27. a woman knows him
28. the man sleeps
29. \*the room sleeps
30. \*the apple sees the chair
31. \*the rooms know the man
32. the apple falls
33. the man falls

**Sample Outputs (this example highlights the categories and structures):**

?- s(Tree,[she,knows,her],[]).

Tree = s(np(pro(she)), vp(v(knows), np(pro(her)))).

?- s(Tree, [the, woman, on, two, chairs, in, a, room, sees, two, tall, young, men], []).

Tree = s(np(det(the), nbar(n(woman)), pp(prep(on), np(det(two), nbar(n(chairs)), pp(prep(in), np(det(a), nbar(n(room))))))), vp(v(sees), np(det(two), nbar(jp(adj(tall), jp(adj(young), n(men)))))))

The output in this example should be carefully studied and emulated by the parser. The categories, e.g., nbar, jp, adj, n, and parsing structure, e.g., nbar(jp(adj(tall), jp(adj(young), n(men)))), should appear in the output. Output that misses categories and parsing structures will be marked down.

Caveat: the grammar you are writing should recognize the relevant sentences (those above and many others) and fail on others. If you generate more sentences or provide further examples for recognition, you will quickly see that there are many odd or ungrammatical sentences that this grammar recognises. Writing a “real” grammar for a fragment of natural language must take into account a range of properties, e.g. ordering of prepositional phrases, pragmatics, etc., which we are not addressing in this exercise. Going “hard core” in the world of computational linguistic parsing and semantic representation means facing lots of hard, complex, and very interesting issues of natural language.

Extra recognition: if you are inclined to engage with further discussion, work on other examples, further extensions, or other languages, you are welcome to share it in your file in a discussion section. There is no additional mark per se. You will get remarks in return from the lecturer.

# Marking Scheme

A total score of 100 broken down as follows:

* 34 marks for the code listing.
* 66 marks total for all the correct parse trees (1-33), where 2 marks for each fully correct parse and 1 mark for a partially correct parse.
* Marks may be taken off at the discretion of the Lecturer; a comment will explain why.

# Notes on grammatical constructions

## Pronouns

Pronouns (e.g. he/him, I/we) have features such as number (singular/plural) and grammatical role (subject/object). In addition, pronouns have features such as grammatical person, e.g. first person "i", second person "you", third person "she". Grammatical person indicates a closer or more distant relationship between the speaker of the sentence and other persons: "I see the apple" represents the most personal statement (first person); "You see the apple" is between the speaker and a person who is immediately present; and "He sees the apple" is the most distant, as it can relate to a person who is not immediately present or somehow less ``relevant''. As the examples above show, there is subject-verb agreement for number, grammatical role, and grammatical person.

## Agreement

The exercise shows several forms of agreement. If any are not clear from learning languages, please ask the Lecturer, who will provide a short review.

## Structures for NPs with Adjectives and Prepositional Phrases

For our purposes, an adjective such as tall describes a property of a common noun such a man. The adjective precedes the noun. For example: the tall man sees the woman is grammatical; the man tall sees the woman is ungrammatical. You can have any number of adjectives, for example: the tall tall old man sees the woman; the tall tall old old man sees the woman.

For our purposes, a prepositional phrase modifies a noun, and it is made up of a preposition and a noun phrase. The preposition provides information about the relative locations of the nouns i.e., the noun that is modified and the noun within the prepositional phrase. The prepositional phrase follows the noun that it modifies: The man in the room sees a woman on a chair. We see the in relation between man and room. You can have any number of prepositional phrases, for example: the woman in a room on the chair in a room in the room sees the man.

As an adjective or prepositional phrase modifies a noun phrase, it can appear with the noun phrase in either the subject or the object position.

As a helpful hint about the grammar of adjectives and prepositional phrases in noun phrases, see the phrase tree for a sample sentence below. It indicates the grammatical structure of the categories and phrase structure for adjectives and prepositional phrases in noun phrases; though somewhat complicated, it shows the variety of structures. While the grammatical structure of jp and nbar are unfamiliar, we can take them as given. Use these categories and phrase structures for your grammar. Given such input (and similar), your parser should produce the same output :.

?- s(Tree, [the, woman, on, two, chairs, in, a, room, sees, two, tall, young, men], []).

Tree = s(np(det(the), nbar(n(woman)), pp(prep(on), np(det(two), nbar(n(chairs)), pp(prep(in), np(det(a), nbar(n(room))))))), vp(v(sees), np(det(two), nbar(jp(adj(tall), jp(adj(young), n(men)))))))

# The Lexicon

The lexicon should include all the following words that appear, where the components of each lexical entry are as given. This is not the form that your code requires, but is a helpful hint.

Note: you must add a feature for animacy to the entries in the lexicon.

word, grammatical category (pronoun), number (singular/plural), grammatical person (1st, 2nd, or 3rd), and grammatical role (subject or object)

i,pro,singular,1,subject you,pro,singular,2,subject he,pro,singular,3,subject she,pro,singular,3,subject it,pro,singular,3,subject we,pro,plural,1,subject you,pro,plural,2,subject they,pro,plural,3,subject me,pro,singular,1,object you,pro,singular,2,object him,pro,singular,3,object her,pro,singular,3,object it,pro,singular,3,object us,pro,plural,1,object you,pro,plural,2,object

them,pro,plural,3,object

word, grammatical category (verb), number (singular/plural), grammatical person (1st, 2nd, 3rd)

know,tv,singular,1 know,tv,singular,2 knows,tv,singular,3 know,tv,plural,\_ see,tv,singular,1 see,tv,singular,2 sees,tv,singular,3 see,tv,plural,\_ hire,tv,singular,1 hire,tv,singular,2 hires,tv,singular,3 hire,tv,plural,\_ fall,iv,singular,1 fall,iv,singular,2 falls,iv,singular,3 fall,iv,plural,\_ sleep,iv,singular,1 sleep,iv,singular,2 sleeps,iv,singular,3

sleep,iv,plural,\_

word, grammatical category (determiner), number

the,det,\_ a,det,singular

two,det,plural

word, grammatical category (noun), number

man,n,singular woman,n,singular apple,n,singular chair,n,singular room,n,singular men,n,plural women,n,plural apples,n,plural chairs,n,plural

rooms,n,plural

word, grammatical category (preposition)

on,prep in,prep under,prep

word, grammatical category (adjective)

old,adj young,adj red,adj short,adj tall,adj