• Language is so important to us that it most clearly identifies us as human.

• Creating computer programs that can understand human language has been one of the long-standing goals of AI.

• Prolog was first implemented in 1971 by Alan Colmerauer at Marseilles, France, as a tool for studying “natural language procession.”

• Prolog provides a powerful and flexible programming framework for natural language processing. Many Prolog systems have a notational extension called DCG (definite clause grammars) to implement formal grammars that are directly executable by Prolog as a syntax analyzer.

• Work in this area ranges from the design of database interfaces in well-understood, specialized problem areas, to translation and speech understanding and generation.
Syntactic vs Semantic Approaches

- There are syntactic and semantic approaches to be addressed.
  - Syntactic techniques are grammar based. They are used to parse sentences into their individual words/phrases and look them up in a dictionary.
  - It has been recognized that human language understanding depends on extensive background knowledge about the domain of discourse and the idioms used in the domain as well as an ability to apply general contextual knowledge to resolve omissions and ambiguities in human language.
  - *Semantic techniques are used to collect and organize the background knowledge.*
Stages of Natural Language Understanding

- **Parsing**: analyze the syntactic structure of sentences.
  
  It not only verifies that sentences are syntactically well formed but also determines their linguistic structure, normally in the form of a parse tree.

- **Semantic interpretation**: produce a representation of the meaning of the text, say in the form of a conceptual graph or semantic network, from the parse tree.

- **Contextual/world knowledge interpretation**: structures from the knowledge base (about the domain of discourse) are added to the internal representation of the sentence to produce an expanded representation of the sentence’s meaning. This adds the necessary world knowledge required for complete understanding.
The Parsing Problem

sentence --> noun_phrase, verb_phrase.
noun_phrase --> determiner, noun.
verb_phrase --> verb, noun_phrase.
verb_phrase --> verb.
determiner --> [the].
noun --> [apple].
noun --> [man].
verb --> [eats].
verb --> [sings].

sentence-|-noun_phrase-|-determiner--|--’’the’’
| |-noun--------|--’’apple’’
| |-verb_phrase-|verb--------|--’’eats’’
| |-noun_phrase-|determiner--|--’’the’’
| |-noun--------|--’’man’’
Writing a Prolog Program

?- sentence([the,apple,eats,man]).
Yes
the [determiner] apple [noun] eats [verb] the [determiner] man [noun].

Prolog code:

sentence(Sentence):-
    append(Noun_phrase,Verb_phrase,Sentence),
    noun_phrase(Noun_phrase),
    verb_phrase(Verb_phrase).

... 

determiner([the]).
noun([man]).
noun([apple]).
verb([eats]).
verb([sings]).
Some Implementation Issues

- Allowing capital letters in your vocabulary:
  The data type for each word should be defined as a string, enclosed in “double quotes”.

- Sentence decomposition before parsing:
  \texttt{frontchar/3}, \texttt{frontstr/4}, \texttt{str_int/2}, \texttt{upper_lower/2}, and \texttt{concat/3} for string manipulation.