Designing a logic problem solver using a natural language system

Aaron Cogbill, Computer Science
Faculty Advisor: Suk-Chung Yoon, Computer Science

Abstract

The zebra puzzle is a logic problem in which the solver is asked to match pet owners with pets and houses while satisfying arbitrary constraints. We outline a natural language interpreter that generates a solution from an English statement of the zebra puzzle, bypassing the need to restate the problem in a domain-specific language. Our system consists of a semantic parser, a model generator based on Prolog's default proof search, and an interactive shell. We discuss the handling of quantifiers, including the unique quantifier "the," in Prolog, which prove essential for solving the puzzle. We also identify deficiencies in our parsing and model-building system that limit our ability to describe the puzzle to the system and find workarounds. Potential improvements include support for pronouns and other words that reference preexisting logical constants, representation of semantic ambiguity via underspecification, and a more sophisticated model generator/automated reasoning system.

The puzzle

Multiple versions of the zebra puzzle exist, including one published in *Life* 1962. We use a version presented in Blackburn, Bos, and Striegnitz:

1. A street contains three neighboring houses, one red, one blue, one green
2. Each house has one resident: a Japanese man, a Spanish man, and a British man
3. Each resident owns a pet: one owns a zebra, one a jaguar, one snails
4. The British man lives in the red house
5. The Spanish man owns the jaguar
6. The Japanese man lives to the right of the snail keeper
7. The snail keeper lives to the left of the blue house

Given this information, who owns the zebra?

Worked solution

- Initial setup, using (1), (2), (3), (4), and (5):

Nationality	Pet	Color	Position
Japanese			
British		red	
Spanish	jaguar		

- By (6), we know the Japanese man is *not* the snail keeper:

Nationality	Pet	Color	Position
Japanese	*not* snails		right of snails
British		red	
Spanish	jaguar		

- Therefore the British man must own the snails and the Japanese man must own the zebra. By (7), we add that the British man lives left of the blue house:

Nationality	Pet	Color	Position
Japanese	zebra		right of snails
British	snails	red	left of blue
Spanish	jaguar		

- From the position column, we conclude that the Japanese man must live in the blue house, and by process of elimination, the Spanish man lives in the green house.

Parsing

We use a feature-based constituency grammar. A constituency grammar reduces phrases to sub-phrases, with words as terminal symbols.

- Example rules for sentence (S), noun phrase (NP), and verb phrase (VP):

```
S  --> NP, VP.
VP --> TV, NP.
NP --> DET, N.
```

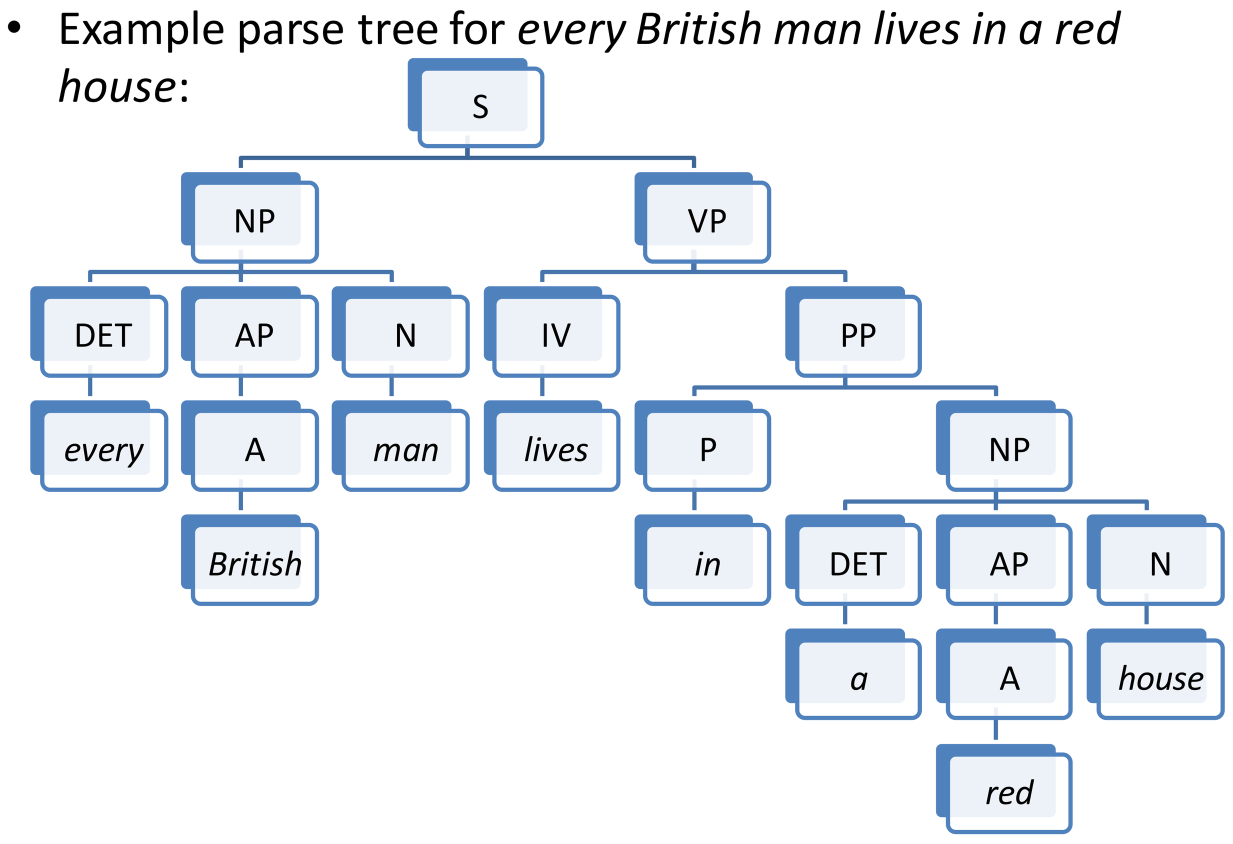
(where TV = transitive verb, DET = determiner, N = noun)

- Example rule for a sentence in terms of feature structures:

$$[category\colon S] \rightarrow \begin{bmatrix} category\colon NP \\ number\colon [1] \\ person\colon [2] \end{bmatrix}, \begin{bmatrix} category\colon VP \\ number\colon [1] \\ person\colon [2] \end{bmatrix}$$

(Here [1] and [2] denote shared features across structures)

- Example parse tree for *every British man lives in a red house*:



Semantic representation

- Lambda calculus is used to transform parsed English sentences into first-order logic with equality
- Example semantic representations at the word level:
  - "house": $\lambda x.house(x)$
  - "every": $\lambda P.\lambda Q.\forall x(P(x) \rightarrow Q(x))$
- We construct the semantic representation of a phrase through functional application of the semantic representations of its components
- Representation of *every British man lives in a red house*:

$$\forall x\left(\begin{matrix} man(x) \wedge British(x) \rightarrow \\ \exists y(house(y) \wedge red(y) \wedge live(x) \wedge in(x,y)) \end{matrix}\right) \quad (1)$$

- Naïve Montague semantics overlooks potential ambiguities. For example, the sentence *every British man lives in a red house* can also be represented as follows:

$$\exists y\left(\begin{matrix} house(y) \wedge red(y) \wedge \\ \forall x(man(x) \wedge British(x) \rightarrow live(x) \wedge in(x,y)) \end{matrix}\right) \quad (2)$$

- In this example, (2) states that a single (exceedingly spacious) house contains every living British man, while (1) admits the possibility of distinct British men living in separate houses
- Crucially, both representations are valid interpretations of the English sentence *every British man lives in a red house*
- The ambiguity of the English can be retained using *underspecified representations*, which we omitted from our implementation for simplicity

Model building

- Since our grammar does not accept pronouns or proper nouns, all incoming sentences will be quantified

<u>Adding to the knowledge base</u>

- Universal statements are inserted into the knowledge base with no changes
- Existential statements $\exists x(P \wedge Q)$ first have their bound variable instantiated to a new constant; we then recurse on P and Q (with x replaced by the new constant)
- Statements quantified by the unique quantifier $\exists! x(P \wedge Q)$ are handled as existential statements; we also add the statement $\forall y(P(y) \rightarrow y = x)$.

<u>Quantified queries to the knowledge base</u>

- To confirm a universal statement holds, we use the following predicate:

```
forall(_X, P, Q) :- \+ (P, \+ Q).
```

(where \+ denotes negation)

<u>Updating the knowledge base (via abduction)</u>

- After adding an existential statement, we identify every unsatisfied universal statements in the KB and collect the missing facts needed to make them true with the following predicate:

```
abduce_universals(Qs) :-
    findall(Q,
        (forall_rec(_X, P, Q), P, \+Q),
    Qs).
```

(where forall_rec/3 is a KB record of a universal statement)

<u>Handling Equality</u>

- Whenever we discover a fact $c_1 = c_2$ (where c_1 and c_2 are distinct constants), we replace every instance of c_2 with c_1 in the model, then remove any duplicate facts

Additional knowledge

To identify the zebra's owner, the system requires the following additional facts:

a) No man is next to himself
b) Every man owns *exactly* one animal

Because we do not implement pronouns ("himself") or the quantifier "exactly one," we compromise:

1. Apply fact (a) to clue (6) : "every man who lives beside the man who owns snails does not own snails"
2. Apply fact (b) to clue (5): "every man who is not the man who owns the jaguar does not own the jaguar"
3. Lemma of fact (b): "the man who does not own the jaguar and who does not own the snails owns the zebra"

- We also must explicitly state that the Japanese man is not the British man, the British man is not the Spanish man, and the Spanish man is not the Japanese man. (Note that this assumption is not universally true: in other discourses one person may have multiple nationalities.)

References

Blackburn, Patrick and Bos, Johan. *Representation and Inference for Natural Language*. Stanford: CLSI, 2005.
Blackburn, Patrick, Bos, Johan, and Striegnitz, Kristina. *Learn Prolog Now!* http://www.learnprolognow.org.
Nugues, Pierre M. *An Introduction to Language Processing with Perl and Prolog*. Berlin: Springer, 2006.