Automaton Theories of Human Sentence Comprehension - Ch. 3: Generalized Left-Corner Parsing

2020. 01. 09

Won Ik Cho

SEOUL NATIONAL UNIVERSITY

Human Interface Laboratory

Pushdown Automaton and CFG

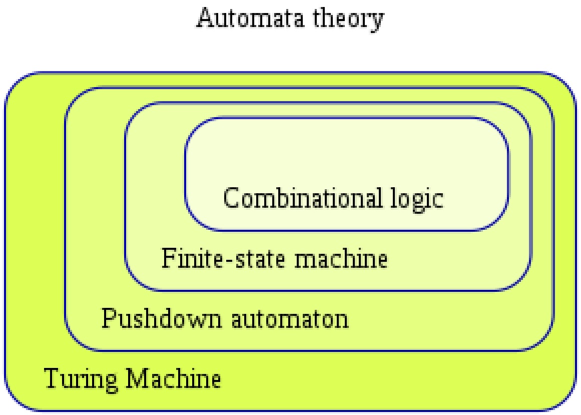
- **Question: How can we combine ‘time’ with a generative grammar?**
  - This yields a concept of ‘automaton’
    - Mathematical model of computation
- **Pushdown automaton and a CFG?**
  - Each operation in the automaton corresponds to a particular CFG rule
    - “Strong competence” or “Transparency”
  - Desirable property in an algorithmic model
    - Clarify the relationship between Marr’s level
- **Transparent processor**
  - The causal actions do what the grammar specifies
- **No canonical automaton for a given grammar**
  - Generalized left-corner parsing: a theory of one aspect of control
    - A case of Kowalski’s (1979) slogan: **ALGORITHM = LOGIC + CONTROL**

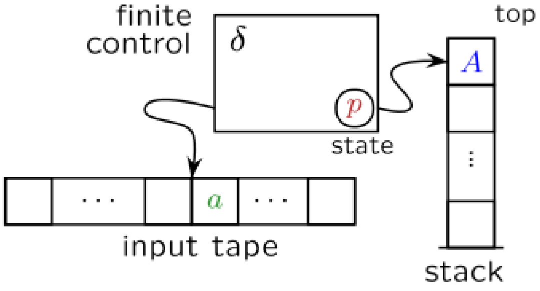
Automaton and Functionalism

- **Basic idea of automaton:**
  - Differentiate abstract 'states' from one another,
  - and specify what happens to cause transitions between them
- **A simple automaton**
  - Each node: **States**
  - Arcs between: **Vocabulary**
  - Path: **Sentence**
  - **Automaton recognize a sentence ⇔ Path exists?**
- **Automata as cognitive models ⇔ Functionalism (Putnam, 1960)**
  - Minds as a software of the brain
  - What's important is what states can do (rather than ingredient)
    - Possibly a 'relativism'?
    - What current NNs (neural networks) share?

Pushdown Automata

- **Finite automata with a stack memory? → Infinite-state machines**
  - e.g., Keep track of embedding levels (garden-pathing)
- **How PDA does this? → A recursive transition network**
  - Pushing the stack whenever it begins an embedded sentence
  - Popping the stack whenever it finishes one
  - Key difference between FA
    - Special 'PUSH' that transfers control to another network of circles and arrows
    - Elements are distinguishable arbitrary symbols





https://en.wikipedia.org/wiki/Pushdown_automaton

Top-Down Parsing

- **Goes through a series of transitions that mimic the line-by-line derivation**
- **Two main operations**
  - **Expand**: Replace the stack symbol with the sequence of symbols
  - **Scan**: Remove the terminal symbol from both the stack and the input
- **Three general points to be illustrated:**
  - Incrementality
    - Deals with each word one-by-one, L to R, via 'scan'
  - Strong Competence
    - Each grammar rule gets its own operation
  - Nondeterminism
    - Fails to specify exactly which operation should apply when more than one step is applicable

Bottom-Up Parsing

- **Write the grammatical category and then find possible rules**
- **Two main operations**
  - **Reduce**: Replace a sequence of symbols with a stack symbol
  - **Shift**: Push the terminal word on the top of the stack symbolizing it
- **Delay – a Hallmark of bottom-up parsing**
  - e.g., Subject NP is not combined with its VP until the internal structure of the VP is fully recognized
- **Nondeterminism: Not reduced**
  - Various candidates of reduction
    - Reduce/reduce conflict
  - The choice between reducing and simply going onto the next word
    - Shift/reduce conflict
- **What is the cost of remembering?**
  - Contrastive patterns of memory consumption

Left-Corner Parsing

- **Resnik: Neither of the two parsing strategies match human performance**
  - Right-branching and left-branching both available?
- **Left-Corner Parsing**
  - Combining good points of both strategy
  - From Bottom-up: Evidence must first be found before a rule can be used
  - From Top-down: Satisfied with only a limited amount of evidence
- **Two main operations incorporating nondeterminism**
  - **Project**: Replace symbol with a record and an expectation
  - **Project + Complete**: Replace symbol and expectation with the new EXP
  - **Shift**: If next word is a terminal symbol, the push it on the top of the stack
- **Fits English very well, but how about other languages?**

Generalized Left-Corner

- **Even English has phrases that aren’t head-initial**
  - e.g., John loves Mary passionately
    - Attaching adverbial post-modifier
    - ‘passionately’ not wanted by any lexical property of the head of the VP
    - Optionality makes no sense to project on the basis of some left-corner
- **GLC: one way to mix traversal orders**
  - Each rule receives an annotation on its RHS: **Announce point**
    - LHS substrings constitutes the ‘trigger’ of this particular rule
    - Putting announce point at the end of adjunction rules allows modifiers to be parsed bottom-up, without unmotivated anticipation
  - GLC parsing allows a theorist to systematically vary the degree to which parsing is predictive on a per-rule basis
    - Every hypothesis about degree-of-expectation can be formalized as a particular positioning of the announce points

Seoul National University

EndOfPresentation

Thank you!