LR parsing is the derivative of context free grammars

Josef Svenningsson

June 2, 2010

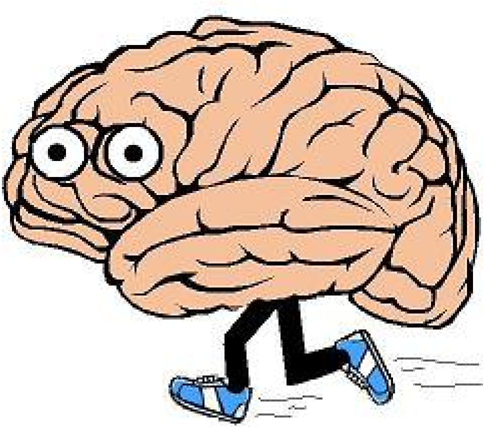
The title says it all.

LR parsing is the derivative of context free grammars

The title says it all.

But you might not be convinced by me just saying so..

Jogging your brains



Context Free Grammars

A context free grammar

$$
\begin{array}{lcl}
E & \rightarrow & E * B \\
E & \rightarrow & E + B \\
E & \rightarrow & B \\
B & \rightarrow & 0 \\
B & \rightarrow & 1
\end{array}
$$

Formal Definition

A Context Free Grammar is a four-tuple:

- A set of terminals T
- A set of nonterminals NT
- A start symbol $S \in NT$
- A set of productions. Each production consist of:
    - A nonterminal; the head
    - The symbol $\rightarrow$
    - A body consisting of a sequence of terminals and nonterminals

Semantics of a grammar

The semantics of a grammar is defined in terms of derivation trees.

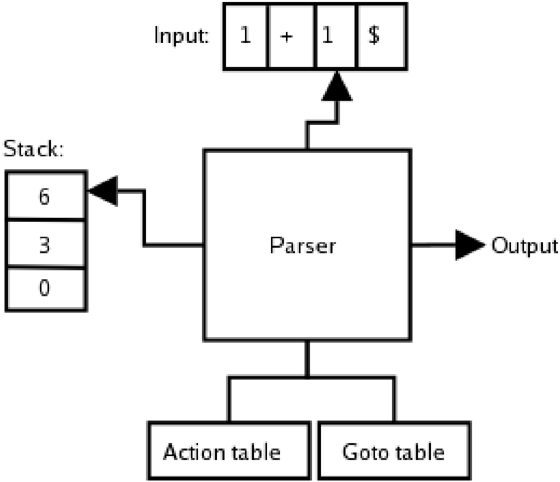
```
        E               E
        |               |
       /|\             /|\
      E * E           E + E
     /     \         /     \
    /|\     0       0      /|\
   0 + 1                  1 * 0
```

If a string has more than one derivation the grammar is ambiguous.

LR parsing

- The most common algorithm for parsing formal languages.
- Invented by Knuth
- Lots of tools for generating LR parsers: YACC, Bison, Happy, ...

LR parser machine



Producing the LR automaton

Going from a grammar to an LR automaton goes via item sets.

Item Set 0

$$
\begin{aligned}
E &\rightarrow \bullet E * B \\
E &\rightarrow \bullet E + B \\
E &\rightarrow \bullet B \\
E &\rightarrow \bullet 0 \\
E &\rightarrow \bullet 1
\end{aligned}
$$

Item Set 1

$$B \rightarrow 0\bullet$$

Item Set 2

$$B \rightarrow 1\bullet$$

Item Set 3

$$E \rightarrow E \bullet * B$$
$$E \rightarrow E \bullet + B$$

Advancing the dot one step is like recognizing one character in the input, alternatively a whole production.

Item set

Each item set corresponds to a state in the LR automaton (for LR(0) parsers).

Item sets are the derivative of the context free grammar

My claim:
Item sets are the derivative of the context free grammar

A new representation of Context Free Grammars

It's not exactly clear from the presentation of item sets that they correspond to some kind of derivative

To demonstrate this we need a new representation of context free grammars

A new syntax for syntax

$$
\begin{array}{lcl}
G, H \in \text{Grammar} & ::= & x \mid 0 \mid 1 \mid G \cdot H \mid G + H \mid T\ A \\
T \in \text{Grammar Tuples} & ::= & r \mid \mu r.\{A \mapsto G\}_{A \in N}
\end{array}
$$

An example grammar

$$\mu r.\{\quad \begin{array}{l} E \mapsto (rE\ \text{'+'}\ rB) + (rE\ \text{'*'}\ rB) + rB; \\ B \mapsto 0 + 1 \end{array} \qquad \}_{\{E,B\}}$$

$$\begin{array}{lcl} E & \rightarrow & E * B \\ E & \rightarrow & E + B \\ E & \rightarrow & B \\ B & \rightarrow & 0 \\ B & \rightarrow & 1 \end{array}$$

Semantics

- The current semantics I have is unfortunately not based on derivation trees.
- It only captures whether a grammar recognizes a language or not. There is no notion of ambiguity.
- $0, 1, +$ and $*$ form a semi-group

The Derivative

$$
\begin{array}{lcl}
\partial_x x & = & 1 \\
\partial_x (y \neq x) & = & 0 \\
\partial_x 0 & = & 0 \\
\partial_x 1 & = & 0 \\
\partial_x (G \cdot H) & = & \partial_x G \cdot H + \delta G \cdot \partial_x H \\
\partial_x (G + H) & = & \partial_x G + \partial_x H \\
\partial_x (rA) & = & 0 \\
\partial_x ((\mu r.\{A \mapsto G_A\}_{A \in N}) B) & = & \mu r.\{S \mapsto \partial_x G_B\}_{\{S\} \cup N} S
\end{array}
$$

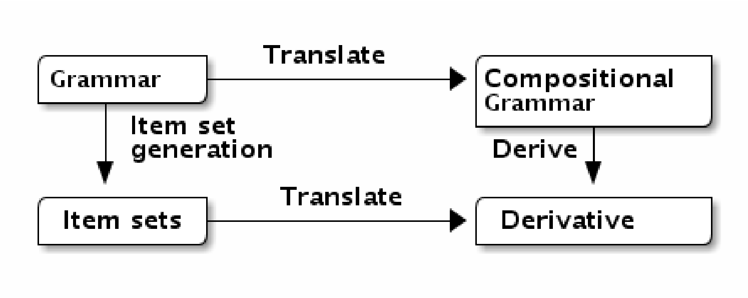
Auxiliary function, empty string recognition

$$\begin{array}{lcl}
\delta x & = & 0 \\
\delta 0 & = & 1 \\
\delta 1 & = & 0 \\
\delta(G \cdot H) & = & \delta G \cdot \delta H \\
\delta(G + H) & = & \delta G + \delta H \\
\delta(rA) & = & rA \\
\delta(\mu r.\{A \mapsto H_A\}_N A) & = & \mu r.\{A \mapsto \delta H_A\} A
\end{array}$$

Derivative wrt Non-terminals

$$
\begin{array}{lcl}
\partial_A x & = & 0 \\
\partial_A 0 & = & 0 \\
\partial_A 1 & = & 0 \\
\partial_A (G \cdot H) & = & \partial_A G \cdot H + \delta G \cdot \partial_A H \\
\partial_A (G + H) & = & \partial_A G + \partial_A H \\
\partial_A (rA) & = & 1 \\
\partial_A (rB \neq A) & = & 0 \\
\partial_A (\mu r.\{B \mapsto G_B\}_N B) & = & \mu r.\{B \mapsto \partial_A G_B\}_N B
\end{array}
$$

The Proof


Grammar
Translate
Compositional Grammar
Item set generation
Derive
Item sets
Translate
Derivative

Derivatives

The intuition behind derivatives in computer science: removing one item

- Data types: making one hole
- Regular expressions: recognizing one character
- LR parsing: recognizing one character

Questions

Questions?

The sequence rule

$$\partial_x(G \cdot H) = \partial_x G \cdot H + \delta G \cdot \partial_x H$$

$$\partial_x(G \cdot H) = \partial_x G \cdot H + G \cdot \partial_x H$$