



Feb 7-11:00 AM

$C := \emptyset$
 For all $q \in Q$, do
 if there is a $r \in S$
 with $\delta(r, a) = q$ then
 $C := C \cup \{q\}$
 return $E \cup C$

build DFA (N):
 $S_0 := \text{start_state}$
 // S_0 is start state of DFA
 $Q := \{S_0\}$
 // Q = state set of DFA
 while there is an $S \in Q$
 and char a such that
 (S, a) has not been processed
 do
 $T := \text{Trans}(S, a)$
 draw an edge $S \xrightarrow{a} T$ labeled a
 $Q := Q \cup \{T\}$
 mark (S, a) as being processed.

end while
 for each $S \in Q$, do
 if $S \neq S_0$
 // S contains an accepting state of N
 and S is a accepting (in D)

Feb 7-11:29 AM

Windows/DOS:
newline is $\backslash n$

Unix/Linux
newline is $\backslash n$ only

% ad -c ^{text} file.txt

%%
 $\backslash n$ putchar('u');
%%
(^{text} main l)
{ ^{yytext} l;
}

/a.out <file.txt> >nextfile.txt

Syntax analyses

parser → four → tokens → parser → parse trees (through tokens)

bison (yacc)

↓
GNU yacc

grammar → lex → code

productions (rules of the grammar)

"context-free grammar" CFB

Feb 7-11:38 AM

N finite set of symbols called nonterminals (or variables)
 T finite set of symbols called terminals (or tokens)
 $N \cap T = \emptyset$
grammar symbols are $N \cup T$
 Designate a symbol $S \in N$ as the start symbol
 Finite list of productions of the form
 $A \rightarrow X_1 X_2 \dots X_k$
 where $A \in N$ and $k \geq 0$ and each X_i is grammar symbol, for $1 \leq i \leq k$.
 Example: grammar for arithmetic expressions:
 start symbol: E (for "expression")
 $E \rightarrow \dots (id)$

Feb 7-11:54 AM

$E \rightarrow i$ (i for "initial")
 $E \rightarrow c$ (c for "terminal")
 $E \rightarrow E + E$
 $E \rightarrow E - E$
 $E \rightarrow E * E$
 $E \rightarrow E / E$
 $E \rightarrow E \% E$
 $E \rightarrow (E)$

start symbol E
 nonterminals: $\{E\}$
 terminals: $\{i, c, +, -, *, /, \%, ()\}$

Generally, first list the productions of the grammar. Then, nonterminals are the symbols appearing in the heads of the productions and terminals are everything else appearing in the bodies.

Grammars in abstract notation:
 Nonterminals: $A, B, C, \dots$
 Terminals: lowercase Roman letter and other nonletters
 "→" separates head from body.
 X, Y, Z (usually with subscripts) to designate non-terminal symbols

Feb 7-12:01 PM

Lower case Greek letters
 $\alpha, \beta, \gamma, \delta, \dots$
 denote strings of grammar
 symbols,
 ϵ always denotes the
 empty string

Typical grammar:

$$\begin{aligned} A_1 &\rightarrow \alpha_1 \\ A_2 &\rightarrow \alpha_2 \\ &\vdots \\ A_n &\rightarrow \alpha_n \end{aligned}$$

Start symbol is head of
 first production (by default).

string α takes
 $\downarrow$ parser
 parse-tree

```

graph TD
    Root["C * i + C"] --> C1["C"]
    Root --> E1["E"]
    Root --> C2["C"]
    E1 --> E2["E"]
    E1 --> Plus["+"]
    E1 --> E3["E"]
    E2 --> E4["E"]
    E2 --> i["i"]
    E4 --> C3["C"]
  
```

Feb 7-12:10 PM