A_x = acceptance (interpreter)
E_x = emptiness
EQ_x = equality
ALL_x = oppositeness

$A_{CFG} = \{ <G, w> \mid w \in L(G) \}$
If G is in CNF, $w \in L(G)$ means there must be $2^{|w|}-1$ derivation steps.

yacc – yet another compiler compiler
bison – GNU yacc
antlr – Java yacc

$E_{CFG} = \{ <G> \mid L(G) = \emptyset \}$
$S \to \varepsilon$ V, R
$A \to BC$ $V \in T$ iff V can reach only terminals
$A \to a$

1. V times do:
2. consider each R:
3. if $R = (S \to \varepsilon)$, then $S \in T$
4. if $R = (A \to a)$, then $A \in T$
5. if $R = (A \to BC)$ and $B \in T$ and $C \in T$, then $A \in T$
6. $S \notin T$

$V_0 \to V_1 V_1$ $V_1 \to V_2 V_2$ $V_2 \to V_3 V_3$ ... $V_n \to a$

$EQ_{CFG} = \{ <G, H> \mid L(G) = L(H) \}$ $\notin \Sigma_0$

$CFL \subset \Sigma_0$: $\forall G \in CFG, \exists m \in TM, L(m) = L(G)$
m = On input w, runs $A_{CFG}(<G, w>)$

2/4-2/ $A_{TM} = \{ \langle M, w \rangle \mid M \in TM \text{ and } w \in L(M) \}$

Σ_1 : On input $\langle M, w \rangle$,
simulate M on w
if accepts, accept
o.w., reject

$\underbrace{\qquad\qquad}$
↓
U : universal Turing machine

Σ_0: On input $\langle M, w \rangle$
for $\underbrace{N}_{??}$ steps, simulate M on w

Meta-circular LISP interpreter

Proof by contradiction.

Assume that H decides A_{TM}. (ie H never ~~loops~~ diverges)

$H(\langle M, w \rangle) = \{$ accept ~~if $w \in L(M)$~~ if M accepts w
reject if M does not accept w $\}$

$H \in \Sigma_0$

M rejects w
M loops on w

Construct D = "On input $\langle M \rangle$ st $M \in TM$,
1. Run H on $\langle M, \langle M \rangle \rangle$
2. Output the opposite of H."

$D(\langle M \rangle) = \{$ accept if M does not accept $\langle M \rangle$
reject if M accept $\langle M \rangle$ $\}$

$D \in \Sigma_0$? ✓

Consider running $D(\langle D \rangle)$
$= \{$ accept if D does not accept $\langle D \rangle$
reject if D accepts $\langle D \rangle$ $\}$

accept ⇒ H is wrong
reject ⇒ H is wrong
diverges ⇒ $H \notin \Sigma_0$
→ FALSE.

Quine Liar's Paradox
"This statement is false"
Russell's Paradox