CSE 396 Lecture Tue Apr 24 Spr 2018

Top Hat #1825

Tue Week 8

Recall algorithm for nullable vars using $\boxed{\text{BFS}}$

Initialize $N =$ ~~$\emptyset$~~ Σ

```
bool changed = true;
while (changed) {
  changed = false;
  for each B in V \ N.
    if (B has a rule B → X⃗ with X⃗ ∈ N*):
      N = N ∪ {B}
      changed = true;
  }
 }
}
```

If we initialize $N = \emptyset$, we still get final N = the set of Nullable Variables. First generation is $A \to X$ with $X \in \emptyset^* = \{\epsilon\}$.

Eps_{CFG}

Solves CFG_ϵ: INST: A CFG G. QUES: Does $S \Rightarrow^* \epsilon$? i.e., Is $\epsilon \in L(G)$?

Algm: yes iff S is in the final N.

Suppose we init $N = \Sigma$ instead. First generation: $B \to X$ with $x \in \Sigma^*$. Those are the variables that have terminal rules.

By inductive analysis, all variables added to N derive terminal strings.

Solves NE_{CFG}: INST: A CFG G. QUES: Is $L(G) \neq \emptyset$?

Algorithm Decider: Answer yes iff loop exits with $S \in N$.

Hence the Eps_{CFG} and NE_{CFG} problems are decidable [for later: indeed in poly(|G|) time

m

Note: If $L \in REC$ then so is $\tilde{L}$ by switching yes and no answers so $E_{CFG} \in REC$ too

How about "Empty Intersection" problems?

EI: INST M_1, M_2

QUES Is $L(M_1) \cap L(M_2) = \emptyset$?

Intersection Non-Emptiness

INE_{CFG}: Inst G_1, G_2.

Ques: Is $L(G_1) \cap L(G_2) \neq \emptyset$?

INE_{DFA}: Given DFAs M_1 & M_2, is $L(M_1) \cap L(M_2) \neq \emptyset$

Previous lecture solved $L(M) \neq \emptyset$ for one DFA by BFS.

Deciding Alg^m:
1. Use Cart. prod. construction for $\cap$ to make M_3 st $L(M_3) = L(M_1) \cap L(M_2)$
2. Run the NE_{DFA} decider on M_3
3. Answer 'yes' if and only if the NE_{DFA} decider answers yes.

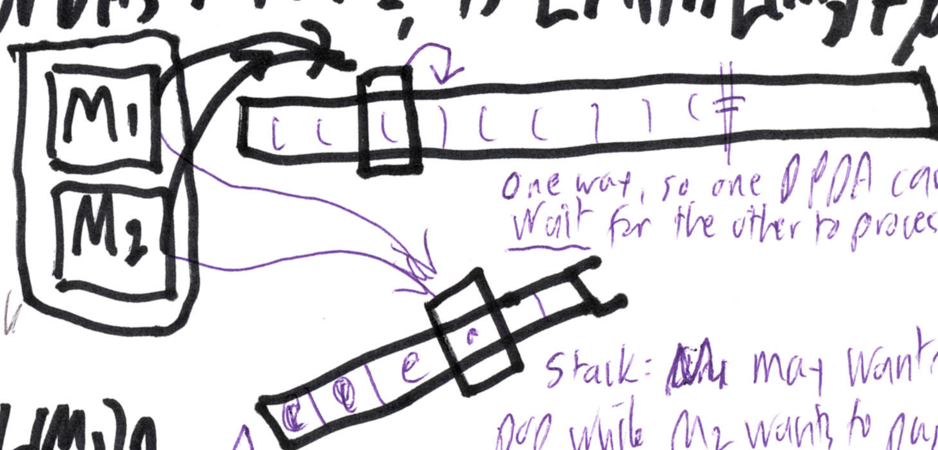
Correct because $L(M_3) \neq \emptyset \Leftrightarrow L(M_1) \cap L(M_2) \neq \emptyset$.

Forward ref to Ch 5: Step 1 computes a function $f(\vec{x})$ that reduces INE_{DFA} to NE_{DFA}

INE_{DPDA}: Given DPDAs M_1 & M_2, is $L(M_1) \cap L(M_2) \neq \emptyset$.

Can we do a similar Cartesian Product idea?

Issue: We can combine states but M_1 and M_2 will fight over one stack



One way, so one DPDA can wait for the other to process

Stack: M_1 may want to pop while M_2 wants to push

Cannot work because $L(M_1) \cap L(M_2)$ might not even be a CFL!

$L(M_1)$ = Balanced Parens, $L(M_2)$ = marked # mirror image

Note: The complement of a DCFL is a DCFL because a DPDA can be made total. Hence DCFL is not closed under $\cup$ either, though the union of two DCFLs is a CFL.

The "other shoe to drop" is we will see that the problems INE_{DPDA} and (hence) INE_{CFG} are undecidable!

③

How about A_{CFG}: INST: A CFG $G = (V, \Sigma, R, S)$, an $x \in \Sigma^*$

QUES: Is $x \in L(G)$, i.e. Does $S \Rightarrow^* x$?

Doing brute-force search on derivations may be open ended when variables are nullable: when to halt?

Decider: 1. First convert G to G' in ChNF. $n = |x|$

If $x \in \epsilon$ use EPS_{CFG} decider.

2. Then if G' derives x, its S' can derive x in exactly $2|x|-1$ steps or not at all.

$\boxed{\text{So } A_{CFG} \in REC}$ Hence brute force to depth $2n-1$ is a decider.

FYI (last week) Let $m = |G|$. Instance size of $\langle G, x \rangle$ is $\approx m+n$.

Step 1 as given in the text can expand a long rule $A \to B_1 B_2 \cdots B_r$ where each B_i is nullable into 2^r rules. If $r \approx \frac{m}{10}$, $2^{\Theta(m)}$ time.

FYI: If you do the "reduce long rules" step first, then step 1 becomes ~~O(m)~~ in $O(m^2)$ time. (Cocke Younger Kasami (CYK or CKY))

Step 2 can be made to run in $O(m^2 n)$ time by dynamic programming.

So A_{CFG} is in P

Diagonalization
Clean, no infinities!

$\underline{D_{CFG}}$ INST: A CFG G encoded as $\langle G \rangle$ over the Σ of G

QUES: Is $\langle G \rangle \notin L(G)$?

Fact 1: DCFG "flip-reduces" to ACFG.

flip answer

$$\langle G \rangle \in L_{D_{CFG}} \equiv D_c \Leftrightarrow \langle G \rangle \notin L(G) \Leftrightarrow \langle G, G \rangle \notin A_{CFG}$$

Hence DCFG is decidable. But it cannot be a CFL.

Theorem There is no encoding scheme $\langle G \rangle$ for CFGs G that makes D_{CFG} into a CFL.

Proof: Suppose there were a "Quixotic" CFG Q s.t. $L(Q) = \overset{=D_c}{D_{CFG}}$ (referencing the encoding scheme $\langle \cdots \rangle$) Then we could take $q = \langle Q \rangle$. Let us ask whether $q \in D_c$:

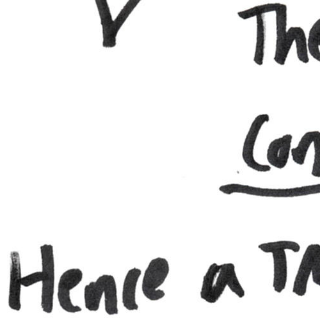
$q \in D_c \iff \boxed{q \notin L(Q)}$ by $D_c = \{\langle G \rangle : \langle G \rangle \notin L(G)\}$

$\iff \boxed{q \in L(Q)}$ because by assumption, $L(Q) = D_c$

A statement can never be equivalent to its negation. So this contradiction "rolls back" the assertion that Q exists. So D_c has no CFG Q, so D_c is not a CFL.

Now define $D_{TM} = \{\langle M \rangle : M \text{ is a DTM and } \langle M \rangle \notin L(M)\}$

Theorem: D_{TM} is undecidable, indeed not even c.e.



Proof: Suppose there were a TM Q s.t. $L(Q) = D_{TM}$. Then we could take $q = \langle Q \rangle$, so $q \in \Sigma^*$.

Then $q \in D_{TM} \iff q \notin L(Q)$ by $D_{TM} = \{q : q \notin L(\ldots$

Contradiction. $\iff q \in L(Q)$ by $L(Q) = D_{TM}$.

Hence a TM Q for D_{TM} cannot exist, so D_{TM} is not Turing acceptable, hence not decidable either. ☒ $\therefore A_{TM} \notin DEC$ Next Tue.