Source Problem: 

**Inst:** A TM M and an input w to M

**Ques:** Does M accept w?

Target Problem: 

**Inst:** A program P and an input x to P

**Type:** \( \langle P, x \rangle \)

**Ques:** Does \( P(x) \) throw an \( \text{ARITH} \) exception?

Answer in Chk 6.5.1

Mindset: Suppose \( \text{EXC THROW} \) were decidable by a total program \( S \)

By defn of decider \( S \) can correctly decide all cases, even "silly" ones of the form

\[ P = \begin{cases} \text{input } x & \text{(input required)} \\ \text{simulate } M(w) & \text{(ignoring)} \end{cases} \]

Thus we would get a decider \( R \) for the \text{Arith} problem by

1. Given \( \langle M, w \rangle \), build \( P_{M,w} \)
2. Run \( S \) on \( \langle P_{M,w}, x \rangle \)
3. Accept \( \langle M, w \rangle \) if \( S \) accepts \( \langle P_{M,w}, x \rangle \)

Why is \( R \) correct?

\[ (M, w) \in \text{Arith} \iff M \text{ accept } w \Rightarrow \text{on any } x \text{, } P_{M,w}(x) \text{ sees the accept and throws the exception} \]

Since \( S \) is a decider for all cases of the target problem, even silly ones, \( R \) is correct.

\[ \text{If } M \text{ does not accept } w \Rightarrow \text{on any } x \text{, } P_{M,w}(x) \text{ does not throw the exception there, nor anywhere} \]

\[ \Rightarrow S \text{ does not accept } \langle P_{M,w}, x \rangle \Rightarrow R \text{ does not accept } \langle M, w \rangle \]

Define \( f(M, w) = \langle P_{M,w}, x \rangle \). This computable function gives \( \text{Arith} \leq_{m} \text{PART Prob} \)

\[ f(M, w) \neq \text{Arith} \leq_{m} \text{PART Prob} \]
$\langle M, w \rangle \in A_{Pm} \Rightarrow \text{on any } x, P_{M,w}(x) \text{ executes } \text{System-exit}(0)
$

Java

$\text{yes output} \Downarrow \sum^* \Rightarrow P_{M,w} \text{ accepts its own code}$

Java

$\langle M, w \rangle \in A_{Pm} \Rightarrow \text{on any } x, P_{M,w}(x) \text{ does not reach the accept/throw}$

Java

$\Rightarrow L(P_{M,w}) = \emptyset \Rightarrow P_{M,w} \text{ does not accept its own code.}$

Java

$A_{Pm} \subseteq K_{Java}$

Java

Also $A_{Pm} \subseteq \text{ALL}_{Java}$ and $A_{Pm} \subseteq \text{NP}_{Java}.$

Java

$K_{Java} \subseteq A_{Pm}$ by $f(p) = \langle p, p \rangle$ instance of $A_{Java}$

Java

$A_{Pm} \subseteq K_{pm}$ similarly $= \langle M, M \rangle$ where $M$ is a TM that simulates $P$ and decodes $\langle M \rangle$ before $\langle P \rangle.$

Java

3) Source $E_{TM}: \text{Input } "M"$

Java

Questions $\Rightarrow$ Is $L(M) = \emptyset$?

Java

Fact: $\Leftrightarrow ACH_M = \emptyset$.

Java

If $\langle M \rangle \in E_{TM} \Rightarrow L(G) = \sum^* \Rightarrow L(G) \text{ is regular} \Rightarrow f(M) = \text{REGULAR CFG}$

Java

If $\langle M \rangle \not\in E_{TM} \Rightarrow ACH_M = \emptyset \Rightarrow L(G) \text{ is not regular} \Rightarrow f(M) \not\in \text{REGULAR CFG}$.

Java

So $E_{TM} \subseteq \text{REGULAR CFG}$ and since $E_{TM}$ is decidable, REGULAR CFG is also.

Java

If we were able to show $A_{Pm} \subseteq \text{REGULAR CFG}$ then it would show the language $\text{REGULAR CFG}$ to be neither RE nor CO-RE.
Example: Let $L = \{x \in \{a,b\}^* : x \text{ is not a palindrome}\}$. Is $L$ a CFL?

$G: S \rightarrow aSb bSb | aTb bTb | \varepsilon$

$L(6) = \text{NPAL} 
= \{aTb bTb | aTb bTb | \varepsilon\}$

Each $M \in (a+b)^*$ is NPAL

Neither regular nor context-free

$D_M$ and $E_M$ are ALLCFG

$A_{NP}$ is in $P$

because we can convert NFA to DFA
not because we can trace the computation in quadratic time.

$\{a, b, c : \exists k \in \mathbb{N} \forall n \in \mathbb{N} \exists m \in \mathbb{N} anbncn \notin L\}$

$\text{PAL} \notin \text{NP}$

$\text{CFL} \subseteq \text{REG}$

$\text{PAL} \subseteq \text{NPAL}$

$\text{NPAL} \subseteq \text{REG}$

$\text{MARK60PAL} \subseteq \text{NPAL}$

$\text{WHAL} = \{a^* \}$

$A_{CFG}$ is in $P$.

(not in text!)
\[(5') \quad A = \{ x : \#a(x) > \#b(x) \}\]

\[B = \{ x : \#a(x) < \#b(x) \}\]

Show that \(L = A \cap B\) is nonregular.

\(x \in A \cap B\) means

\[
\begin{array}{|c|}
\hline
Y & Z \\
\hline
\#a > \#b & \#b > \#a \\
\hline
\end{array}
\]

\(x = a b b b b b b b b b b \underbrace{a a a a a a a a a}_{n} b\), only have one possible breakdown.

Take \(S = a \underbrace{b b b b b b b b b b}_{n} a\).

Let any \(x \cdot y \in S\) be given. Then

\(x = a b^m. y = a b^n\) where \(m, n \in \mathbb{N}\).

\(y = a^{n-1} b\). Then \(y \cdot z = a^{n+1} b^n \in L\) but \(x \cdot z = a b^m a^{n-1} b^n \notin L\).

\(L(x z) \notin L(y z)\). \(L\) is nonregular because it has too many possibilities as.

Guessing as big \(y \neq S\) is like an unbraded loop

while (true)

for \(x = 1 \rightarrow 1^n\)

try "\(x \cdot y \in L(6)\)"

if not break and acen

\(\text{num}\)