Lecture Tue 4/7. Technique for proving non-CFL nature of a language $L \subseteq \Sigma^*$:

Let any (sufficiently large) $p > 0$ be given.

Take $x =$ _____ such that $x \in L$ and $|x| \geq p$ indeed often $|x| \gg p$.

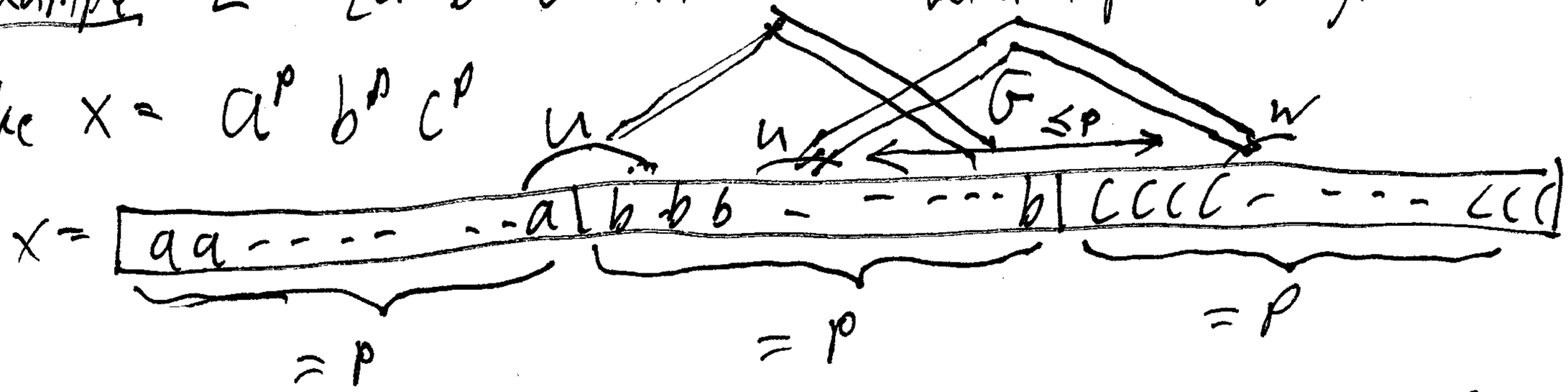
Let any breakdown $x =: yuvwz$ subject to $|uvw| \leq p$ and $uw \neq \epsilon$ be given.

Take $i =$ _____ so that you can get $x^{(i)} =_{def} yu^i v w^i z \notin L$. ($i=0$ is allowed).

Then L is not a CFL.

Example: $L = \{a^n b^n c^n : n \geq 0\}$. Let any $p > 0$ be given.

Take $x = a^p b^p c^p$



Take $x^{(2)} = yuuvwwz$. By $|uvw| \leq p$, uvw can't both contain an 'a' and a 'c', so either $\#a(x^{(2)})$ or $\#c(x^{(2)})$ remains equal to p. We get at least one other character, so we cannot have all of $\#a$, $\#b$, $\#c$ equal to p, so $x^{(2)}$ can't possibly belong to L. Hence L is not a CFL. ⊠

⊠ The same proof works for $L' = \{x \in \{a,b,c\}^* : \#a(x) = \#b(x) = \#c(x)\}$. Could also reason that since $R = a^*b^*c^*$ is a regular set, if L' were a CFL then so would be $L' \cap R$, but this equals L.

Note $L = \underbrace{\{a^n b^n c^r : n, r \geq 0\}}_{L_1} \cap \underbrace{\{a^r b^n c^n : n, r \geq 0\}}_{L_2}$. ②

The two languages being intersected are CFLs

CFG for L_1: $S \to TC$, $C \to cC \mid \varepsilon$ ($L_C = c^*$)
$T \to aTb \mid \varepsilon$.

CFG for L_2: $S \to AT'$, $A \to aA \mid \varepsilon$, $T' \to bT'c \mid \varepsilon$.

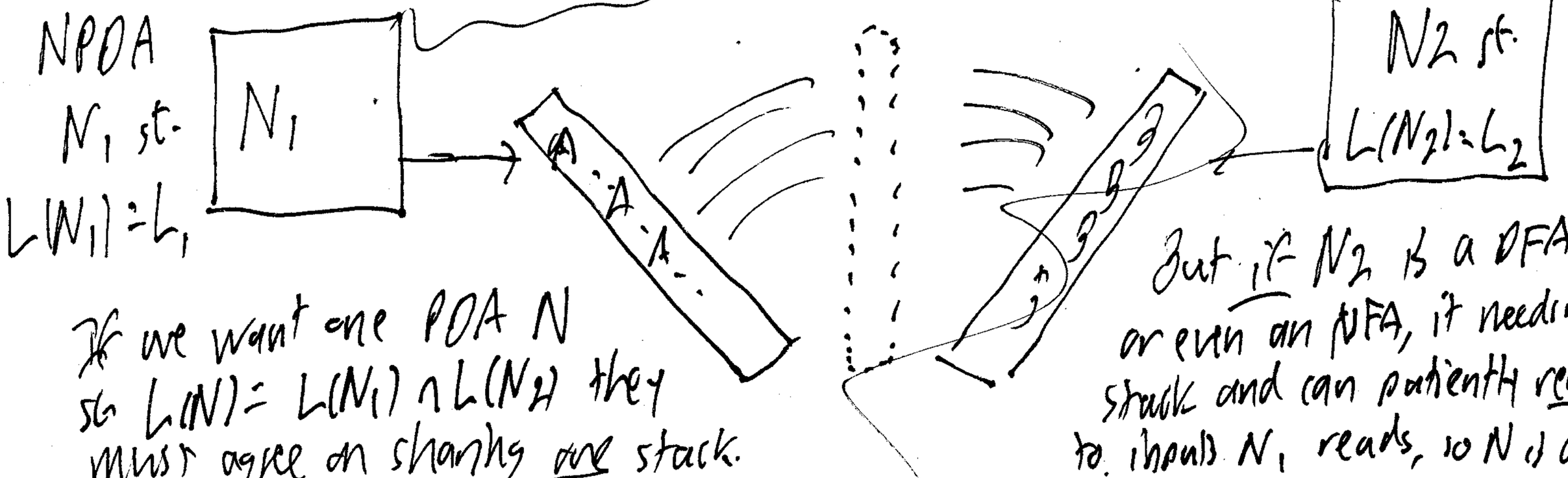
∴ The class of CFLs, unlike the class of regular languages, is not closed under intersection. $\boxed{\tilde{L} \text{ is a CFL (remarked on HW)}}$

- It is closed under Union.
- Hence it is also not closed under complements — if it were, we would get closure under $\cap$ via $L_1 \cap L_2 = \sim(\tilde{L_1} \cup \tilde{L_2})$.

- However, if L is a CFL and R is a regular set, then $L \cap R$ (and of course $L \cup R$) are CFLs.

Why the Cartesian product construction doesn't work for two PDAs

$x \stackrel{?}{\in} L_1 \cap L_2$ $\quad x = x_1 x_2 \cdots x_n$ (a^n, b^n, A)

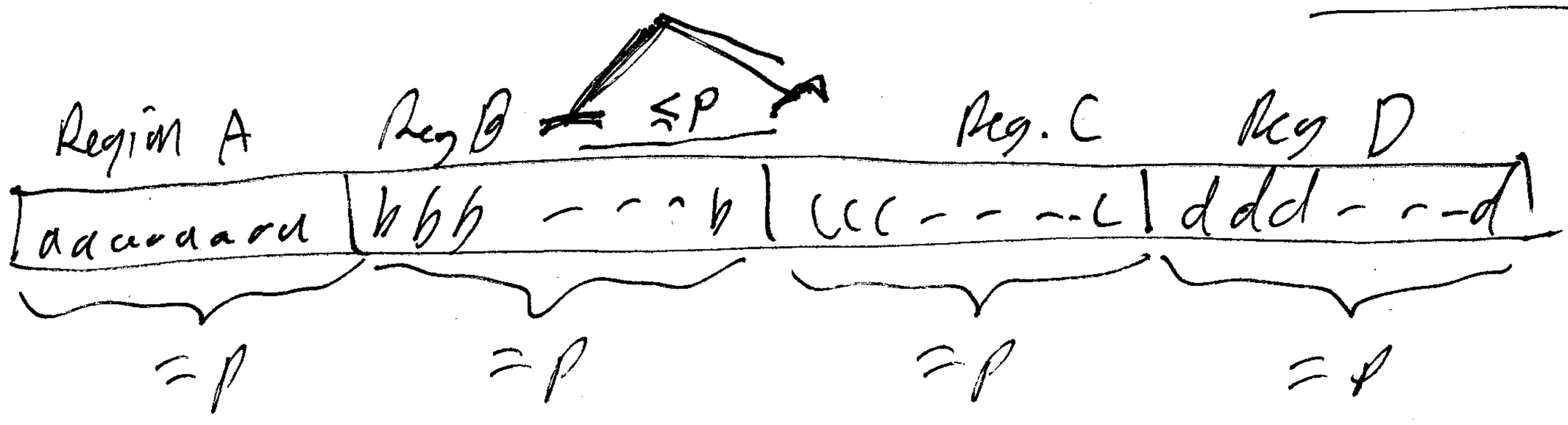


If we want one PDA N s.t. $L(N) = L(N_1) \cap L(N_2)$ they must agree on sharing one stack.

But if N_2 is a DFA, or even an NFA, it needs no stack and can patiently react to what N_1 reads, so N is OK.

Example 2: $A = \{a^m b^n c^m d^n : m,n \geq 0\}$ is not a CFL. (9)

Proof: Let any (sufficiently large) $p > 0$ be given. Take $x = a^p b^p c^p d^p$.



Let any breakdown $x = yuvwz$ with $|uvw| \leq p$, u,w not both ε, be given. The compass must "touch" at least one region A, B, C, or D.

But:
- If it touches A, it can't also touch C. $x^{(1)} = yu^1vw^1z$
- If it touches B, it can't also touch D. $= x$
- " " C, " " A $x^{(i)} = yu^ivw^iz$
- " " D, " " B. $x^{(0)} = yvz$

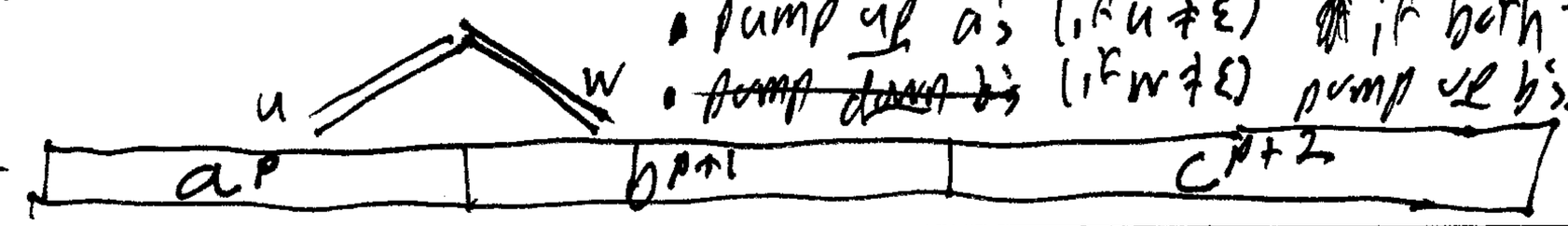
Hence in $x^{(0)}$ or $x^{(2)}$ or any $x^{(i)}$ with $i \neq 1$, the restriction $\#a = \#c$ and $\#b = \#d$ cannot both be preserved. So $x^{(i)} \notin A$, so A is not a CFL.

- Same proof works for $A'' = \{a^m b^n a^m b^n : m,n \geq 0\}$.
- Then it also works for $\{ww : w \in \{a,b\}^*\}$ (for double words). The argument is then easiest for $i = 0$. Another way: take $R = a^*b^*a^*b^*$ and use $D \cap R = A''$, so if D were a CFL then A'' would be also.
- The "$\cap R$" idea helps somewhat with languages like $L = a^*b^*c^*$

$L' = \{x \in \{a,b,c\}^* : \#a(x) > \#b(x)$ & $\#b(x) > \#c(x)\}$.

$L''' = L' \cap R = \{a^m b^n c^p : m > n$ and $n > p\}$. Enough to show L''' not a CFL.

Take $x = a^p b^{p+1} c^{p+2}$



- pump up a's (if $u \neq \varepsilon$) if both $\neq \varepsilon$, must pump up
- ~~pump down b's~~ (if $w \neq \varepsilon$) pump up b's.

If $\wedge$ is only in Cs, must pump down.

CFGs can handle nested and sequential binary dependencies:

$B_1 = \{a^m b^n a^n b^m\}$ is a CFL, by nesting.

CFG $S \to aSb \mid T$

$T \to bTa \mid \varepsilon$.

$B_2 = \{a^m b^m a^n b^n : m, n \geq 0\}$ is also a CFL,

it is in fact equal to $A \cdot A$ where $A = \{a^n b^n : n \geq 0\}$.

What kind of machine can handle $\{a^n b^n c^n\}$ and $\{a^n b^n c^n d^n\}$?

What capabilities can we add to a DFA to handle these languages?

$X \sim$ [$= a^n$? : $= b^n$? : $= c^n$?] for some $n \geq 0$?

(1) Allow the DFA to change a char on the tape.

(2) Allow the DFA to move its read head L as well as R.

(1) by itself makes no difference – the new char could be lumped in with a new state in $Q \times \Sigma = Q$

(2) by itself also makes no difference. NFA

(HARD THM, Not in text! Two-Way DFA crossing sequences.)

(1) & (2) together create a Turing Machine: $M = (Q, \Sigma, \Gamma, \delta, s, F)$ (F: q_{acc}, q_{rej})

where $\delta \subseteq (Q \times \Gamma) \times (\Gamma \times \{L, R, \underbrace{S}_{\text{stay}}\} \times Q)$.

$(p, c/d, D, q)$