**Red** = the class of regular languages.

**PCFL** = the class of languages accepted by DPDA aka "deterministic CFLs." But there is no universally agreed notion of "deterministic CFL."

**CFL** = the class of $L(G)$ for CFGs $G$, which equals the class of $L(N)$ for NPDAs $N$. 

**Briefer Idea:** $S \rightarrow AB \rightarrow PooS$, push B, push A. 

Therefore, $S \rightarrow AC \rightarrow ABAC \rightarrow \text{ read } a$ and pop A.

This is the CFG->DPDA proof. The conversion is very hard.

\[ A, \text{ also } \{a^ib^i \mid i \geq 1 \} \]
\[ A' = \{x \in \sum^* \mid x = a^ib^ic^i \} \]
\[ \forall a^nb^n: n \geq 0? : A \]

For any class $C$ of languages, $\text{co-}C = \{ \overline{L} \mid L \in C \}$.

**Classes contain what's below HBM but show different Mirror symmetry = closure under ~.
The diagram does not tell about closure under $V$ or $U$, only $\cap$.

Fact: $REC$ is closed under all of them. $DCEL$ is closed under $\cap$.

(A $DFA$ can compute $a, b, c$; ignore $c$.)

Also: $A_1 = \{a^i b^j c^k : i \neq j \}$ is a DCEL. A $DFA$ can compute $a, b, c$; ignore $c$.

$A_2 = \{a^i b^j c^k : i \neq j \}$ is a DCEL.

And: $A_1 \cap A_2$ is a union of two DCELs, but not a DCEL.

Finally: Marked pair $\{a^i b^j c^k : i = j \}$ is $\notin$ DCEL, but even pal $\{a^i b^j c^k : i = j \}$ is not.

Some New Nomenclature and Classes.

**Definition:** A language $L$ is 
- Turing-recognizable
- Turing-acceptable
- Recursively enumerable (r.e.)
- Computable enumerable (c.e.)

If also the TM $M$ is total, i.e., halts for all inputs, then

$L = L(M)$ is called decidable.

The class of such languages is usually called $REC$, sometimes $DEC$.

Also $RE = REC$ and $RE$ are both closed under $\cap$ and $\cup$.

Thm: For every $N$-state DFA $L$ with $L(M) \cap L(N)$ there is a $DFA$ that accepts $L(M)$.

**Facts we will prove:**

$REC$ is closed under $\cap$.

$REC \cup \overline{REC}$ does not equal $REC$.

We will fill all regions above $REC$ with undecidable problems.
New Content: Algorithm that Decide Problems

An algorithm that always halts is called a decision procedure and (technically by the Church-Turing thesis) can be implemented in a Turing machine that always halts.

Key Tool: Breadth-First Search.

Ch 4: Problems about grammars.

Sipser's Problem Naming Scheme.

A DFA: Given a DFA $M$ and an input $x \in \Sigma^*$, does $M$ accept $x$? A for Acceptance Problem. The language of this problem is $\{<M, x> : M$ is a DFA and $M$ accepts $x \}$.

$\implies$ refers to some "transparent" way of coding the contents as a string.

$E_{DFA}$: Given just the DFA $M$, is $L(M) = \emptyset$? $E$ for Empty.

Language: $\{<M> : L(M) = \emptyset \text{ when } M \text{ is a DFA} \}$. $E_{DFA}$ or just $E_{DFA}$.

Generally, the language $L_{DT}$ of a problem $T = \{<M> \in \Sigma^* : \text{the answer to } T \text{ is } \text{YES} \}$.

Note $N_{DFA} = \exists \text{ DFAs } M : L(M) \neq \emptyset$.

"Nonemptiness Problem:"

Theorem: A DFA, $E_{DFA}$, and $N_{DFA}$ are all decidable problems/languages.

$A_{DFA}$: decision procedure is simple. "Given $M$ and $x$, just run $M(x)$ and say YES if and only if $M$ accepts $x$.

$E_{DFA}$: Fact: $L(M) = \emptyset \iff$ some final state is reachable from $s$. Breadth first search thus gives a decision procedure to tell. 18