Closure properties of RE, Rec

Recall that a language L is

- **recursively enumerable** (RE) if there exists a TM for L,
- **recursive** (Rec) if there exists a TM for L which halts on every input (i.e. also on strings not from L).

Is the class RE closed under union? And intersection?

Is the class Rec closed under union? And intersection?

What can we say about complement?

Rec: YES
RE: NO
Closure properties of RE, Rec

Lemma: RE languages are closed under union.

Lemma: RE languages are closed under intersection.

[Section 10.1]
Closure properties of RE, Rec

Lemma : Recursive languages are closed under union.

Lemma : Recursive languages are closed under intersection.

similar to RE
Closure properties of RE, Rec

Lemma: Recursive languages are closed under complement.

Pf: easy: switch $h_a$ and $h_r$ to get a TM for $L'$

(and the TM never goes to an infinite loop)

(assuming a total transition function – we said that every undefined transition
  goes to $h_r$, so we now just need to add
  these transitions to the transition func)

Lemma: RE languages are not closed under complement.

given a TM T which

\[
\begin{align*}
\text{accepts } x \\
\text{rejects } x \\
\text{loops on } x
\end{align*}
\]

Pf: we will assume that the RE class is bigger than Rec class.

see next thm

assume RE is closed under complement $\Rightarrow$ if $L$ is RE then $L'$ is RE $\Rightarrow$ by thm

but we said that RE class is bigger & thus RE cannot be closed under comp!
Thm: L and L’ are RE iff L is recursive.

Pf: $\Leftarrow$ L rec. $\Rightarrow$ $\exists$ TM $T$ for L which never loops $\Rightarrow$ $\exists$ TM $T$ for L, therefore L is RE

$\Rightarrow$ L’ rec. $\Rightarrow$ $\exists$ TM $T_2$ for L’ $\Rightarrow$ L’ is RE

$\Rightarrow$ a TM $T_1$ for L

$\Rightarrow$ $\exists$ TM $T_3$ for L which never loops

$T_1$ on $x$ \hspace{1cm} accept \hspace{0.5cm} iff \hspace{0.1cm} x \in L

T_2$ on $x$ \hspace{1cm} reject \hspace{0.5cm} loop

idea for $T_3$:

run $T_1, T_2$ simultaneously (careful with the tape)

if $T_1$ accepts $\Rightarrow h_a$

if $T_2$ accepts $\Rightarrow h_r$

(no other cases)

$\square$
Noam Chomsky studied grammars as potential models for natural languages. He classified grammars according to these four types:

- **Type 0 Grammars**: *Unrestricted Grammars* (generate RE languages)
- **Type 1 Grammars**: *Context-sensitive (monotone) Grammars* (generate context-sensitive languages)
- **Type 2 Grammars**: *Context-free Grammars* (generate context-free languages)
- **Type 3 Grammars**: *Regular Grammars* (generate regular languages)
Def: An **unrestricted grammar** is a 4-tuple $G=(V,\Sigma,S,P)$ where

- $V$ is a finite set of variables
- $\Sigma$ is a finite set of terminal symbols
- $S \in \Sigma$ is the start symbol
- $P$ is a finite set of productions of the form $\alpha \rightarrow \beta$ where $\alpha \in (V \cup \Sigma)^+$ and $\beta \in (V \cup \Sigma)^*$

($V$ and $\Sigma$ are assumed to be disjoint)
Example: Give an unrestricted grammar for \( \{ a^k b^k c^k \mid k \geq 0 \} \)
Unrestricted Grammars (Type 0) [Section 10.3]

Example: Give an unrestricted grammar for \( \{ a^j \mid j = 2^k, k \geq 0 \} \)
Def: A type 0 grammar $G=(V, \Sigma, S, P)$ is **context-sensitive** if for every production rule $\alpha \rightarrow \beta$ in $P$, $|\alpha| \leq |\beta|$.

Which of our examples of type 0 grammars are context-sensitive?
Context-sensitive Gram. (Type 1)

Def: A type 0 grammar $G=(V,\Sigma,S,P)$ is context-sensitive if for every production rule $\alpha \rightarrow \beta$ in $P$, $|\alpha| \leq |\beta|$.

Lemma: Every context-free language which does not contain $\Lambda$ is context-sensitive.
Context-sensitive Gram. (Type 1)

**Def:** A type 0 grammar $G=(V, \Sigma, S, P)$ is **context-sensitive** if for every production rule $\alpha \rightarrow \beta$ in $P$, $|\alpha| \leq |\beta|$.

**Lemma:** Every context-free language which does not contain $\Lambda$ is context-sensitive.

**Def:** A linear-bounded automaton $A$ is a TM which never rewrites a blank to a non-blank symbol.

**Lemma:** A language $L$ is context-sensitive iff there exists a linear-bounded automaton accepting $L$. 
Regular Grammars (Type 3)

Def: A type 0 grammar $G=(V,\Sigma,S,P)$ is **regular** if every production rule in $P$ is of the form $A \rightarrow \sigma B$ or $A \rightarrow \sigma$, where $A,B \in V$ and $\sigma \in \Sigma$.

Lemma: A language $L$ is regular iff there exists a regular grammar for $L-\{\Lambda\}$. 