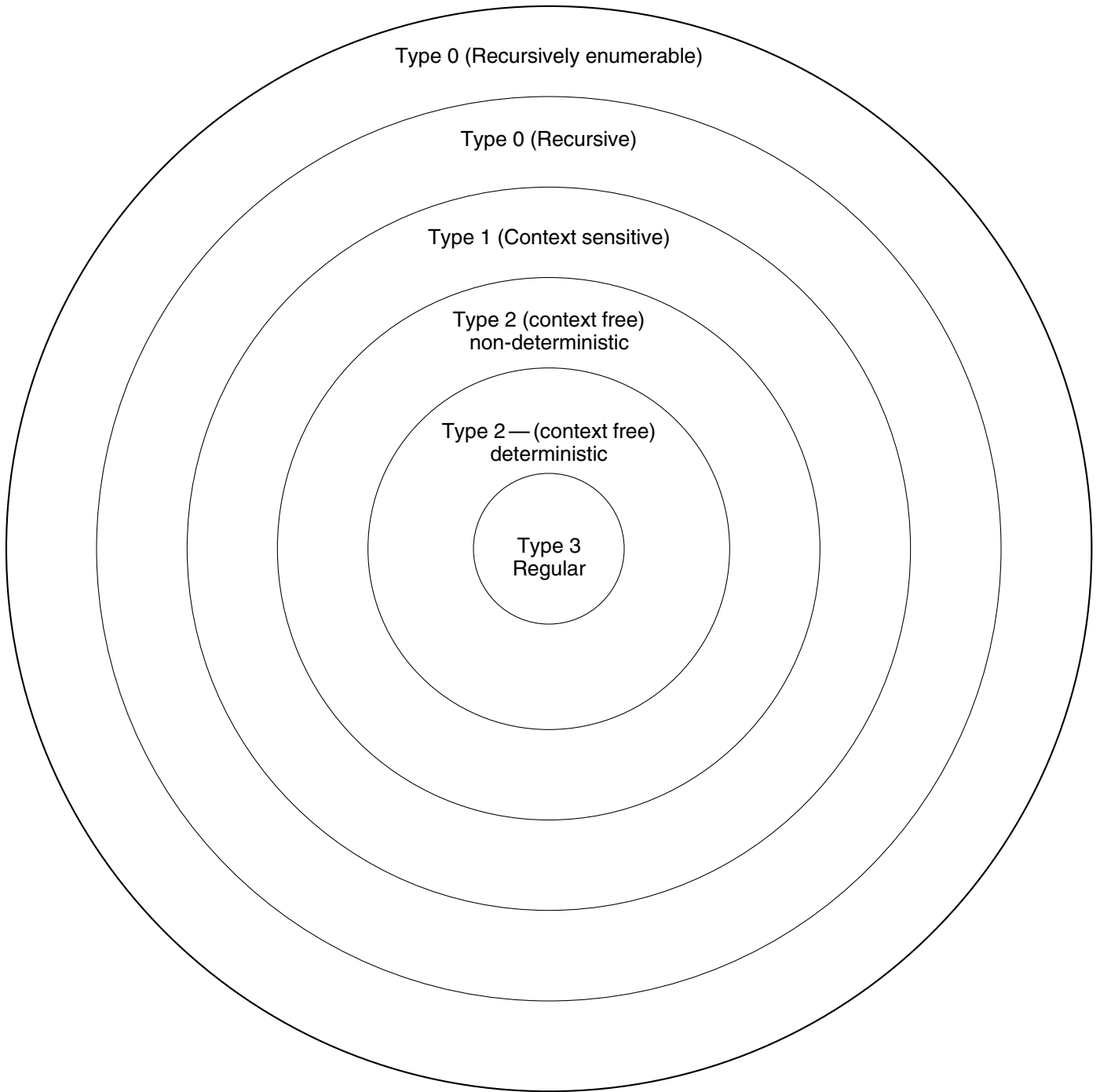


# THE CHOMSKY HIERARCHY

- 'Not effectively computable'



## Language, Grammar and Automaton Types

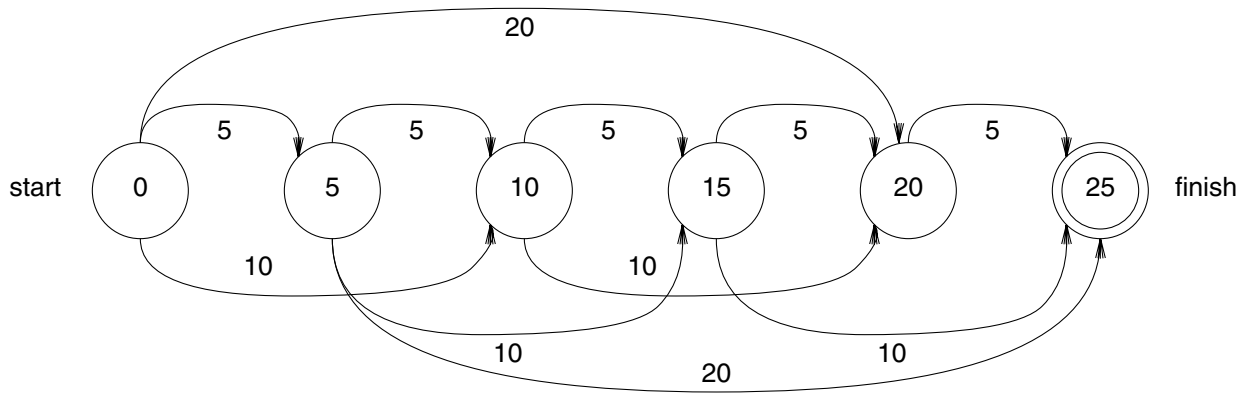
Chomsky Type	Language Type	Grammar Restriction	Automaton Type (Memory?)
Type 3	Regular	LHS Single variable RHS either (a) a single word, e.g. $S \rightarrow x$ (b) or, a single word plus single variable e.g. $S \rightarrow aX$	Finite state automata  (No memory)
Type 2	Context Free	LHS single variable RHS can be anything e.g. $S \rightarrow aXYbh$	Pushdown stack automata (Stack memory)
Type 1	Context Sensitive	e.g. $aZ \rightarrow Y$ i.e. "Z goes to Y provided a is on the left"  RHS never shorter than LHS	Linear bounded Automata (Bounded RAM)
Type 0	Recursive	No restrictions but productions terminate	Turing machines (Unlimited RAM)
Type 0	Recursively Enumerable	No restrictions but may loop for ever	Turing machines (Unlimited RAM)

### Notes:

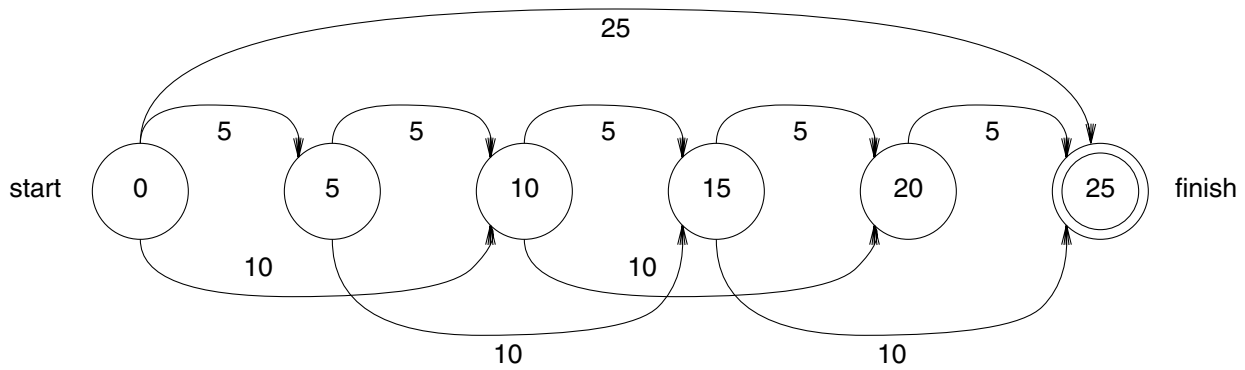
\* In Chomsky notation capital letters (e.g. S, X, Y) represent non-terminal (variable) symbols. Lower case letters e.g. x, y, z represent terminal (word) symbols

\* Note that a limited form of left/right recursion is allowed in Type 3 rules e.g.  $X \rightarrow aX$  (or  $X \rightarrow Xa$ ).

\*Type 2 rules allow more than one terminal symbol on the RHS e.g.  $S \rightarrow aXYb$  as well as allowing 'self-embedding' recursion e.g.  $X \rightarrow aXa \mid a$



**Finite State Automaton (FSA) for accepting a 25p car park charge (UK version)**



**Finite State Automaton (FSA) for accepting a 25¢ car park charge (US version)**