2.2. Complexity of Natural Languages

context free grammars(type 2): rules of the form $X \rightarrow \alpha$ where X is a single non-terminal symbol, α are nonempty sequence of symbols.

(right) regular grammars(type 3): rules of the form $X \rightarrow a$ and $X \rightarrow aN$ where X and N are nonterminal symbols, and a is a terminal symbol.

(left) regular grammars(type 3): rules of the form $X \rightarrow a$ and $X \rightarrow Na$ where X and N are nonterminal symbols, and a is a terminal symbol.

2.2. Complexity of Natural Languages

The Chomsky Hierarchy

- **unrestricted** or **type-0** grammars, generate the *recursively* enumerable languages, automata equals *Turing machines*
- **context-sensitive** grammars, generate the *context-sensitive* languages, automata equals *Linear Bounded Automata*
- **context-free** grammars generate the *context-free* languages. automata equals *Pushdown Automata*
- **regular** grammars, generate the *regular* languages, automata equals *Finite-State Automata*

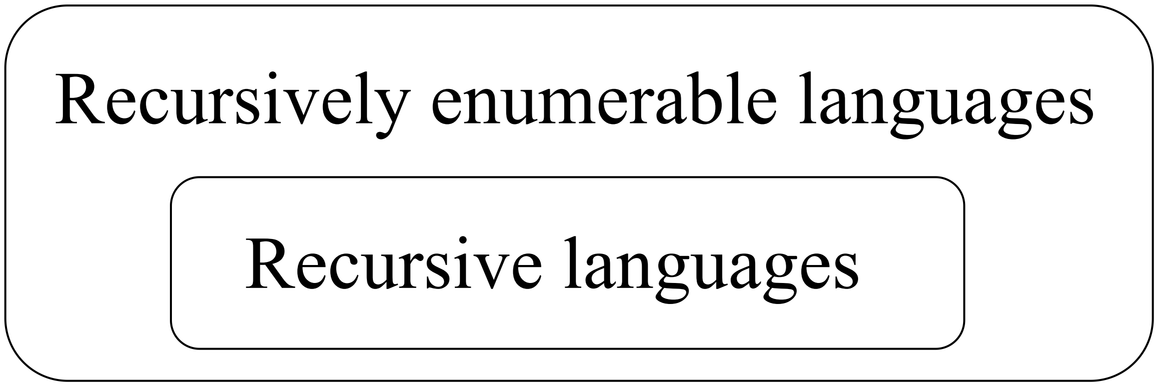
2.2. Complexity of Natural Languages

* A language is *recursively enumerable* if there exists a Turing machine that accepts every string of the language, and does not accept strings that are not in the language.

"Does not accept" is *not* the same as "reject" -- the Turing machine could go into an infinite loop instead, and never get around to either accepting *or* rejecting the string.
The languages generated by unrestricted grammars are precisely the recursively enumerable languages.

* A language is *recursive* if there exists a Turing machine that accepts every string of the language and rejects every string (over the same alphabet) that is not in the language.

2.2. Complexity of Natural Languages



Decidable language (definition)

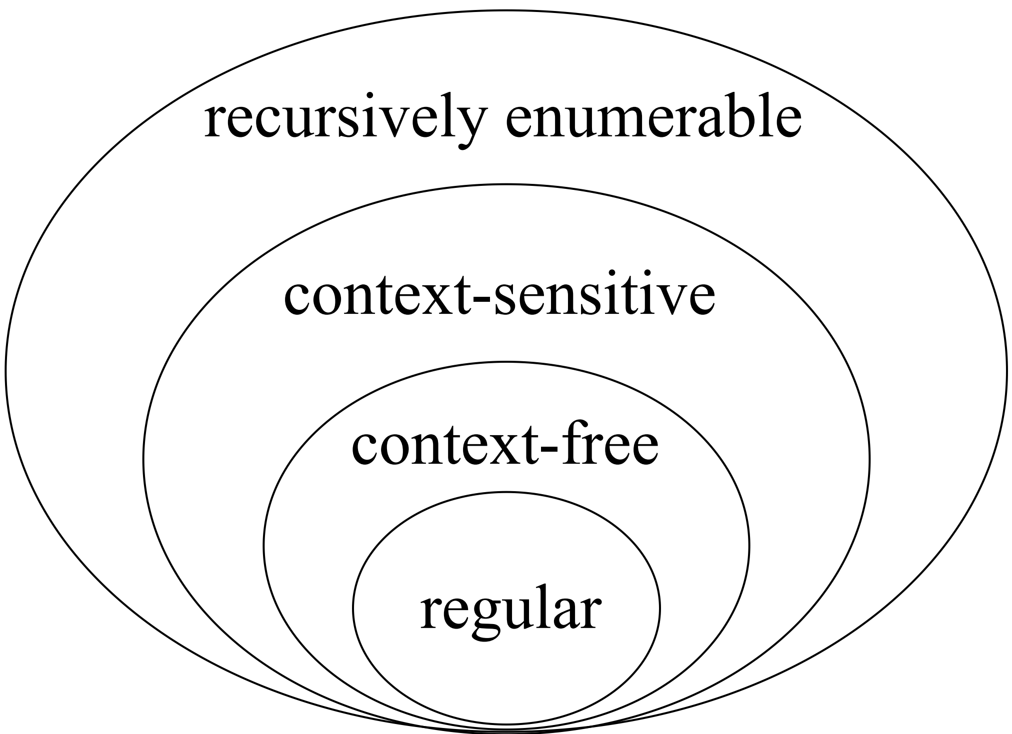
Definition: A language for which membership can be decided by an *algorithm* that halts on all inputs in a finite number of steps --- equivalently, can be recognized by a *Turing machine* that halts for all inputs.

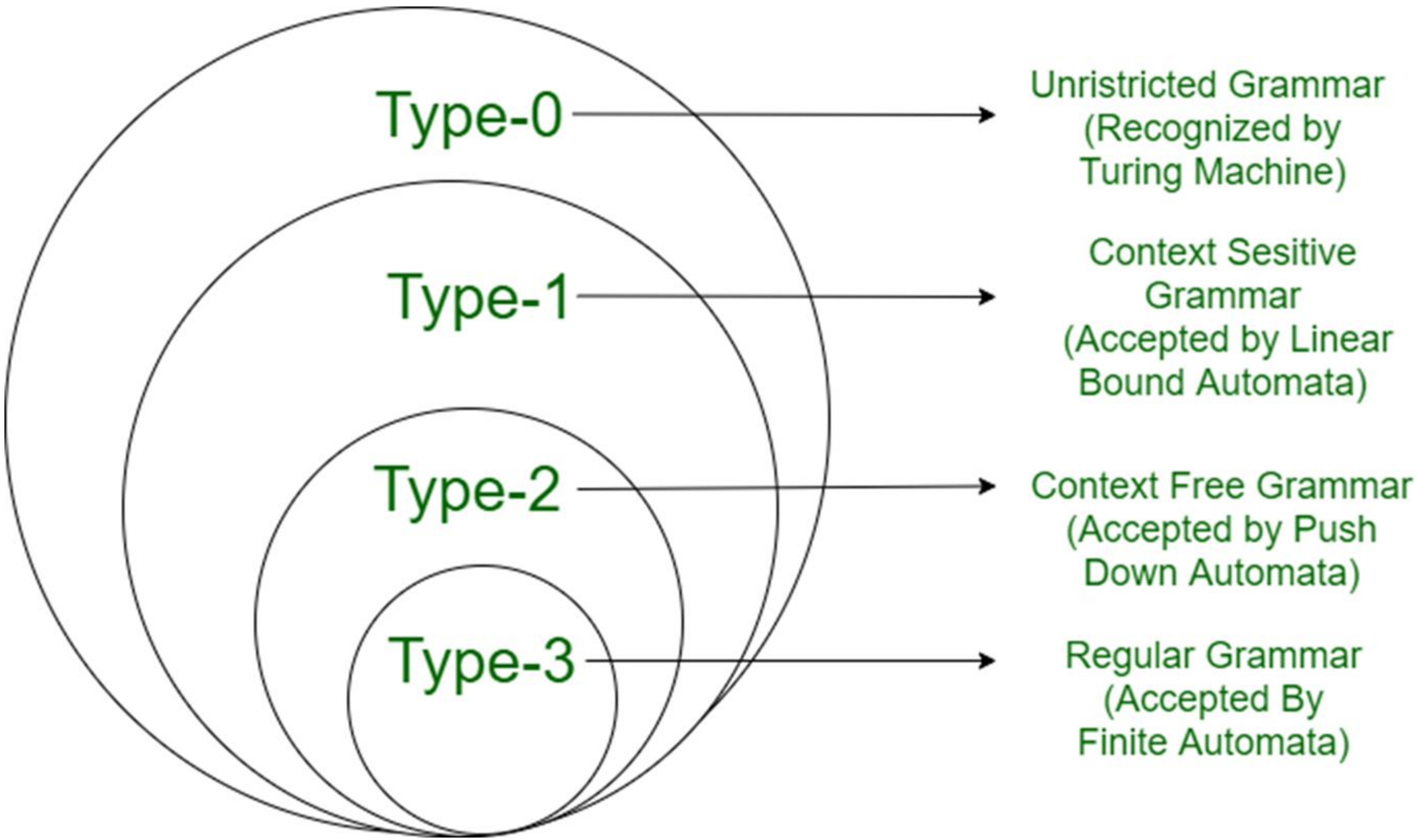
2.2. Complexity of Natural Languages

Generative capacity of grammars

Any grammar G that is a type n (> 0) grammar is also a type n-1 grammar.

Any language that is a type n (> 0) language is also a type n-1 language.




Type-0
Type-1
Type-2
Type-3
Unristricted Grammar
(Recognized by
Turing Machine)
Context Sesitive
Grammar
(Accepted by Linear
Bound Automata)
Context Free Grammar
(Accepted by Push
Down Automata)
Regular Grammar
(Accepted By
Finite Automata)