Theory of Computation and Compilers

University Computer Science · Vocabulary in context

Word	Example
theory of computation and compilers	**Theory of computation and compilers** studies what machines can do and how code is translated.
deterministic finite automaton	A **deterministic finite automaton** reads a string and ends in an accepting or rejecting state.
pumping lemma	The **pumping lemma** shows some languages are not regular.
context-free grammars	**Context-free grammars** can describe matching brackets.
derivation	A **derivation** builds a string step by step from the grammar.
parse tree	A **parse tree** shows how a sentence fits the grammar.
ambiguity /ˌæmbɪˈgjuːɪti/	**Ambiguity** means one string has two different parse trees.
Turing machine	A **Turing machine** reads and writes symbols on a long tape.
halting problem	The **halting problem** cannot be solved by any program.
diagonalisation /daɪˌægənəlaɪˈzeɪʃn/	**Diagonalisation** builds something that differs from every item in a list.
tokens /ˈtəʊkənz/	The lexer splits code into **tokens**, such as names and numbers.

exercises.lol exercises.com.cn

Word	Example
context-free languages	Some **context-free languages** need a stack to recognise them.
symbol table /ˈsɪmbl ˈteɪbl/	The **symbol table** stores each label and its address.
intermediate representation	The compiler turns code into an **intermediate representation** before machine code.

exercises.lol exercises.com.cn