

MathWorks Compiler Course – Day 2

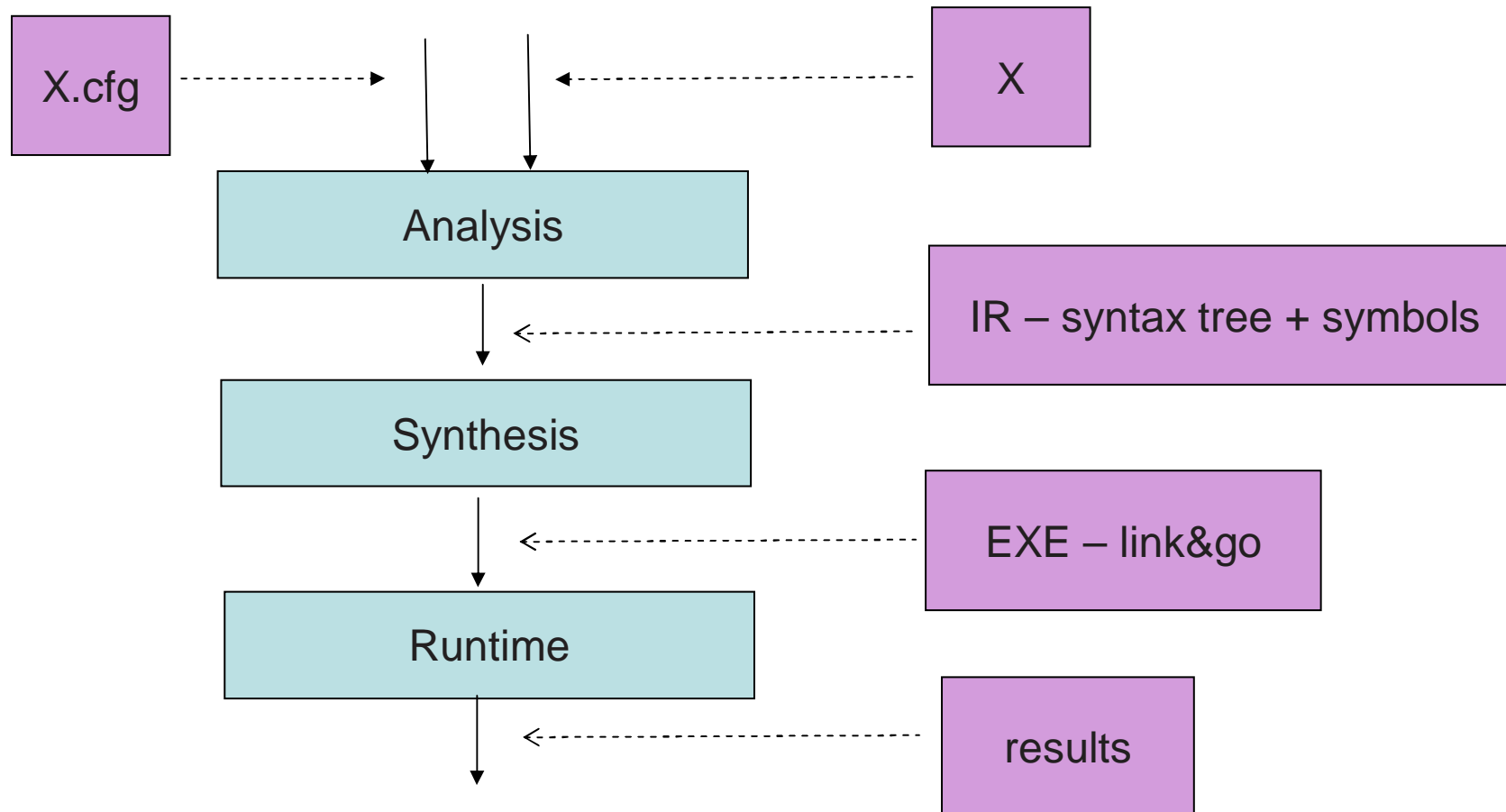
- MATLAB objects using nested functions

MathWorks Compiler Course – Day 2

- MATLAB objects using nested functions
- The structure of xcom

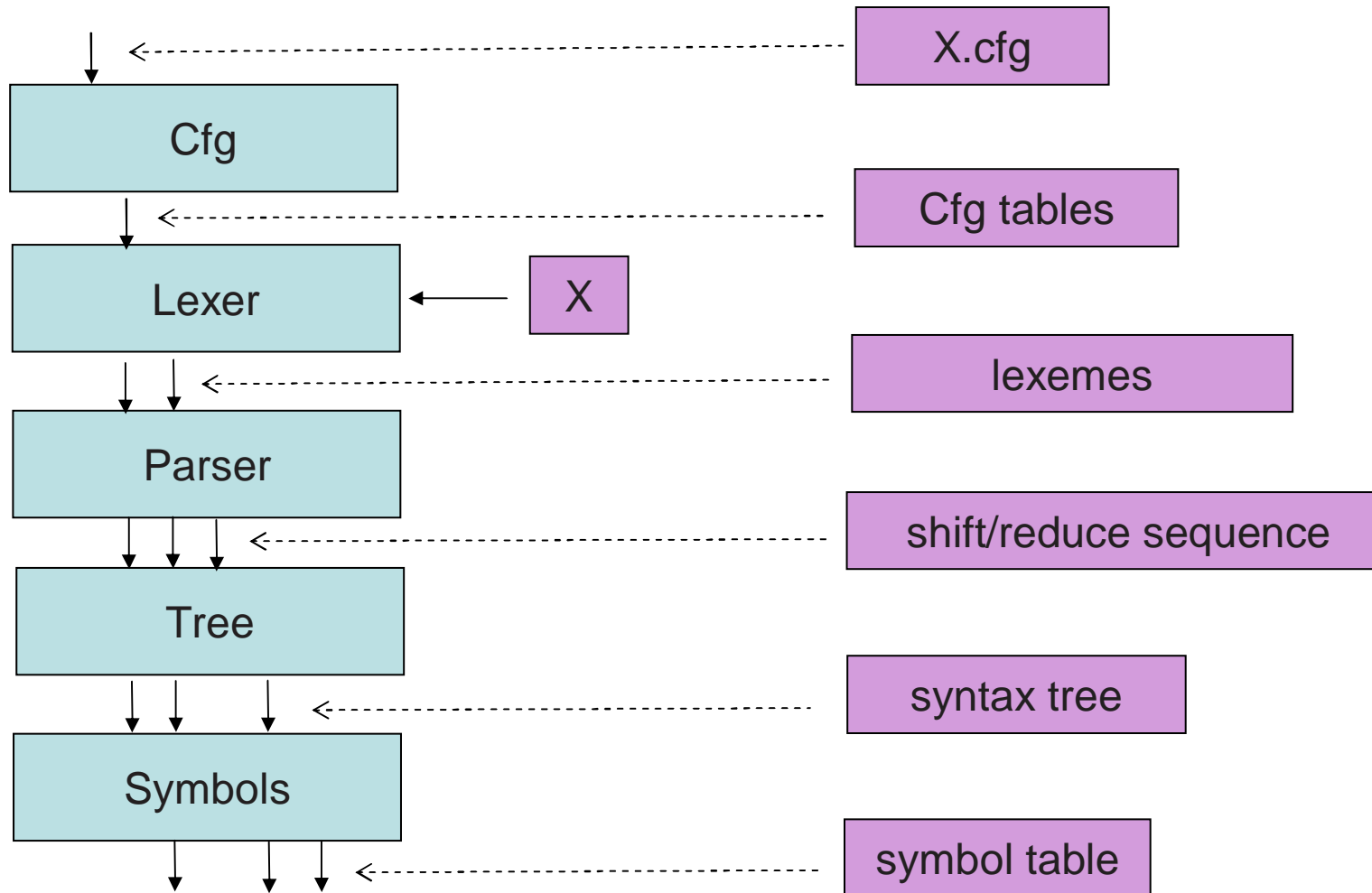
MathWorks Compiler Course – Day 2

- The structure of xcom



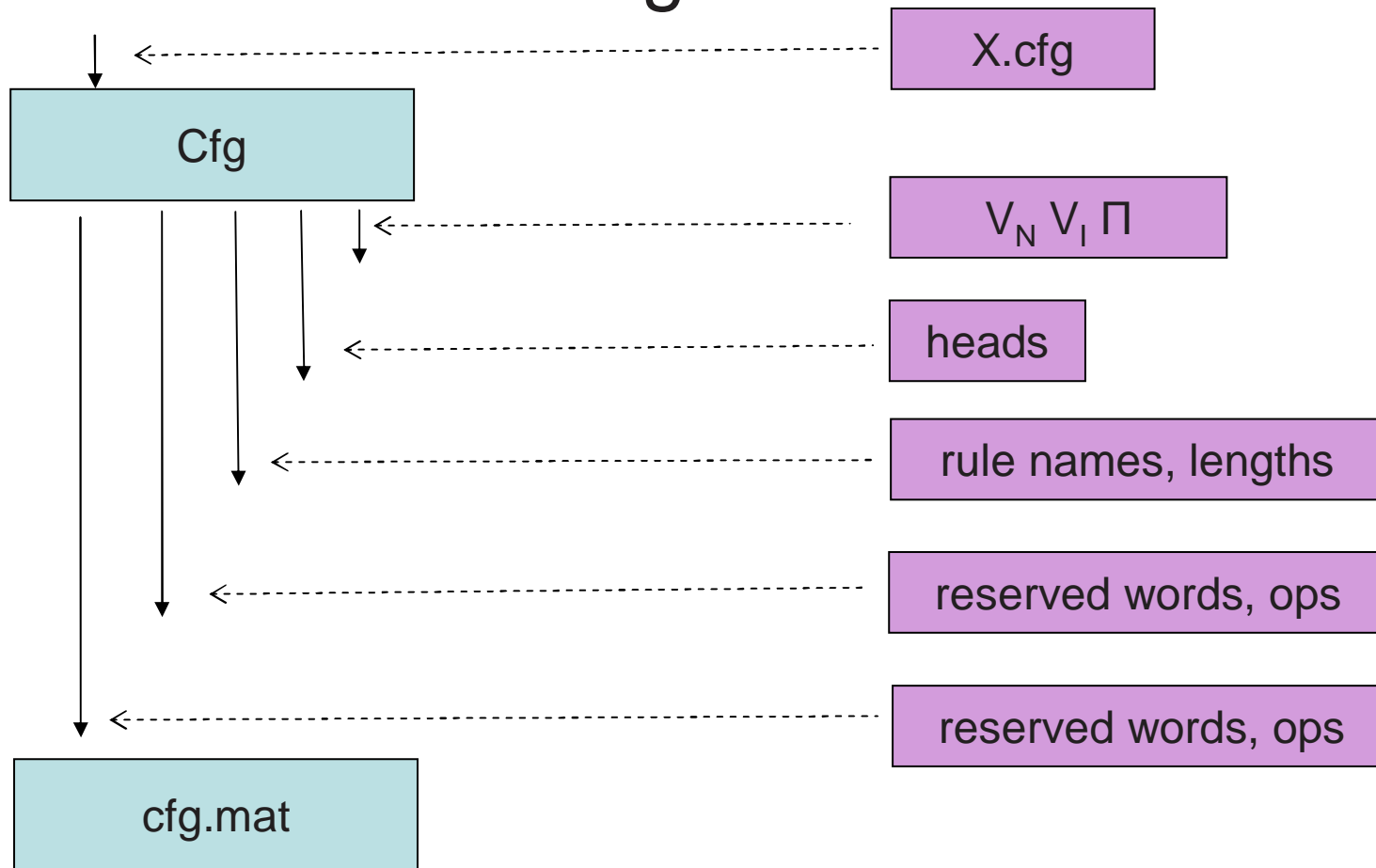
MathWorks Compiler Course – Day 2

- The structure of Analysis



MathWorks Compiler Course – Day 2

- The structure of Cfg



MathWorks Compiler Course – Day 2

- Notation supporting grammars

MathWorks Compiler Course – Day 2

- Notation supporting grammars
 - Definitions

MathWorks Compiler Course – Day 2

- Notation supporting grammars
 - Definitions
 - Greek alphabet

MathWorks Compiler Course – Day 2

- Notation supporting grammars
 - Definitions
 - Greek alphabet
 - Propositional and Predicate Logic

MathWorks Compiler Course – Day 2

- Notation supporting grammars
 - Definitions
 - Greek alphabet
 - Propositional and Predicate Logic
 - Sets

MathWorks Compiler Course – Day 2

- Notation supporting grammars
 - Definitions
 - Greek alphabet
 - Propositional and Predicate Logic
 - Sets
 - Ordered pairs and tuples

MathWorks Compiler Course – Day 2

- Notation supporting grammars
 - Definitions
 - Greek alphabet
 - Propositional and Predicate Logic
 - Sets
 - Ordered pairs and tuples
 - Sequences

MathWorks Compiler Course – Day 2

- Notation supporting grammars
 - Definitions
 - Greek alphabet
 - Propositional and Predicate Logic
 - Sets
 - Ordered pairs and tuples
 - Sequences
 - Relations

MathWorks Compiler Course – Day 2

- Notation supporting grammars
- Context-free Grammars

MathWorks Compiler Course – Day 2

- Notation supporting grammars
- Context-free Grammars
 - Phrase structure

MathWorks Compiler Course – Day 2

- Notation supporting grammars
- Context-free Grammars
 - Phrase structure
 - Reduction rules

MathWorks Compiler Course – Day 2

- Notation supporting grammars
- Context-free Grammars
 - Phrase structure
 - Reduction rules
 - Doing reductions

MathWorks Compiler Course – Day 2

- Notation supporting grammars
- Context-free Grammars
 - Phrase structure
 - Reduction rules
 - Doing reductions
 - Syntax tree

MathWorks Compiler Course – Day 2

- Notation supporting grammars
- Context-free Grammars
 - Phrase structure
 - Reduction rules
 - Doing reductions
 - Syntax tree
 - Everyday notation for CFGs

MathWorks Compiler Course – Day 2

- Notation supporting grammars
- Context-free Grammars
 - Reduction rules
 - Doing reductions
 - Syntax tree
 - Everyday notation for CFGs
 - Rule names, numbers

MathWorks Compiler Course – Day 2

- Notation supporting grammars
- Context-free Grammars
 - Doing reductions
 - Syntax tree
 - Everyday notation for CFGs
 - Rule names, numbers
 - Formal definition of CFG

MathWorks Compiler Course – Day 2

- Notation supporting grammars
- Context-free Grammars
 - Doing reductions
 - Syntax tree
 - Everyday notation for CFGs
 - Rule names, numbers
 - Formal definition of CFG
 - Derive CFG from Π alone

MathWorks Compiler Course – Day 2

- Notation supporting grammars
- Context-free Grammars
 - Syntax tree
 - Everyday notation for CFGs
 - Rule names, numbers
 - Formal definition of CFG
 - Derive CFG from Π alone
 - Formal definition of language

MathWorks Compiler Course – Day 2

- Notation supporting grammars
- Context-free Grammars
 - Everyday notation for CFGs
 - Rule names, numbers
 - Formal definition of CFG
 - Derive CFG from Π alone
 - Formal definition of language
 - Left to right parsing

MathWorks Compiler Course – Day 2

- Notation supporting grammars
- Context-free Grammars
 - Rule names, numbers
 - Formal definition of CFG
 - Derive CFG from Π alone
 - Formal definition of language
 - Left to right parsing
 - Restrictions on CFGs

MathWorks Compiler Course – Day 2

- Notation supporting grammars
- Context-free Grammars
 - Formal definition of CFG
 - Derive CFG from Π alone
 - Formal definition of language
 - Left to right parsing
 - Restrictions on CFGs
 - Transforming CFGs

MathWorks Compiler Course – Day 2

- Notation supporting grammars
- Context-free Grammars
 - Derive CFG from Π alone
 - Formal definition of language
 - Left to right parsing
 - Restrictions on CFGs
 - Transforming CFGs
 - Free-form CFGs

MathWorks Compiler Course – Day 2

- Notation supporting grammars
- Context-free Grammars
 - Formal definition of language
 - Left to right parsing
 - Restrictions on CFGs
 - Transforming CFGs
 - Free-form CFGs
 - A grammar-grammar

MathWorks Compiler Course – Day 2

- Notation supporting grammars
- Context-free Grammars
- Regular Expressions

MathWorks Compiler Course – Day 2

- Notation supporting grammars
- Context-free Grammars
- Regular Expressions
- Finite Automata

MathWorks Compiler Course – Day 2

- Notation supporting grammars
- Context-free Grammars
- Regular Expressions
- Finite Automata
 - State-transition diagrams

MathWorks Compiler Course – Day 2

- Notation supporting grammars
- Context-free Grammars
- Regular Expressions
- Finite Automata (FA)
 - State-transition diagrams
 - CFG for FA

MathWorks Compiler Course – Day 2

- Notation supporting grammars
- Context-free Grammars
- Regular Expressions
- Finite Automata (FA)
 - State-transition diagrams
 - CFG for FA
 - Deterministic (DFA)

MathWorks Compiler Course – Day 2

- Notation supporting grammars
- Context-free Grammars
- Regular Expressions
- Finite Automata (FA)
 - State-transition diagrams
 - CFG for FA
 - Deterministic (DFA)
 - Nondeterministic (NFA)

MathWorks Compiler Course – Day 2

- Context-free Grammars
- Regular Expressions
- Finite Automata (FA)
 - State-transition diagrams
 - CFG for FA
 - Deterministic (DFA)
 - Nondeterministic (NFA)
 - NFA to DFA

MathWorks Compiler Course – Day 2

- Regular Expressions
- Finite Automata (FA)
 - State-transition diagrams
 - CFG for FA
 - Deterministic (DFA)
 - Nondeterministic (NFA)
 - NFA to DFA
- Regular Expression Grammars (REG)

MathWorks Compiler Course – Day 2

- Finite Automata (FA)
 - State-transition diagrams
 - CFG for FA
 - Deterministic (DFA)
 - Nondeterministic (NFA)
 - NFA to DFA
- Regular Expression Grammars (REG)
 - REG grammar-grammar

MathWorks Compiler Course – Day 2

- Finite Automata (FA)
- Regular Expression Grammars (REG)
 - REG grammar-grammar
 - Rewriting REGs

MathWorks Compiler Course – Day 2

- Finite Automata (FA)
- Regular Expression Grammars (REG)
 - REG grammar-grammar
 - Rewriting REGs
- I/O Grammars (IOG)

MathWorks Compiler Course – Day 2

- Finite Automata (FA)
- Regular Expression Grammars (REG)
 - REG grammar-grammar
 - Rewriting REGs
- I/O Grammars (IOG)
 - Formal Definition of IOG

MathWorks Compiler Course – Day 2

- Regular Expression Grammars (REG)
 - REG grammar-grammar
 - Rewriting REGs
- I/O Grammars (IOG)
 - Formal Definition of IOG
 - IOG grammar-grammar

MathWorks Compiler Course – Day 2

- Regular Expression Grammars (REG)
 - REG grammar-grammar
 - Rewriting REGs
- I/O Grammars (IOG)
 - Formal Definition of IOG
 - IOG grammar-grammar
 - Rewriting IOGs

MathWorks Compiler Course – Day 2

- Regular Expression Grammars (REG)
 - REG grammar-grammar
 - Rewriting REGs
- I/O Grammars (IOG)
 - Formal Definition of IOG
 - IOG grammar-grammar
 - Rewriting IOGs
- Grammars for X, C and Java