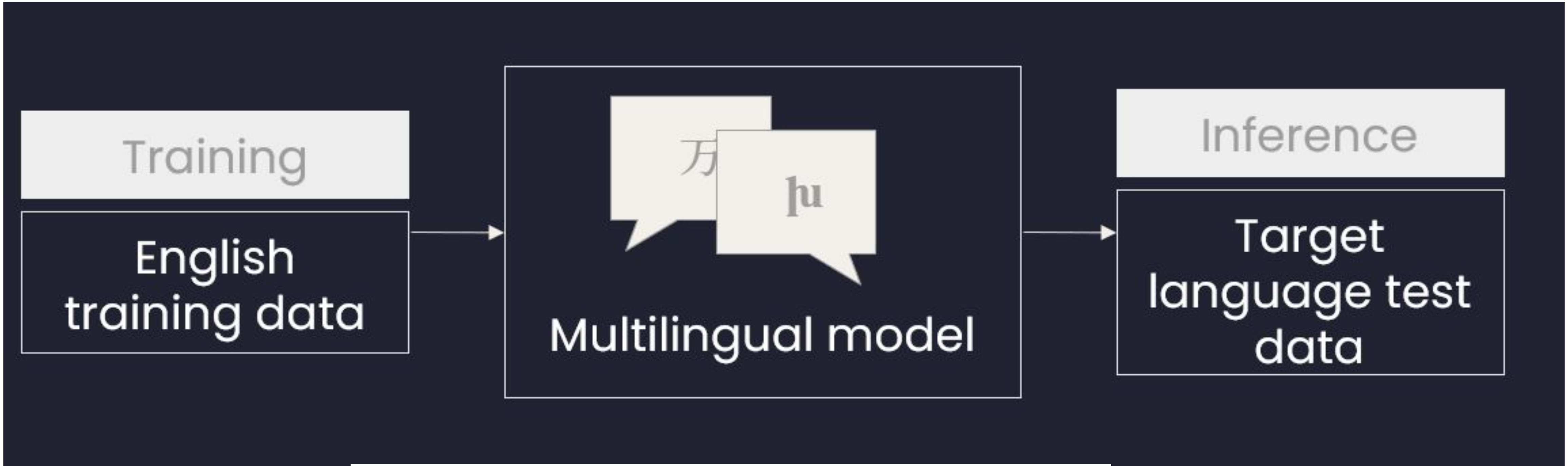
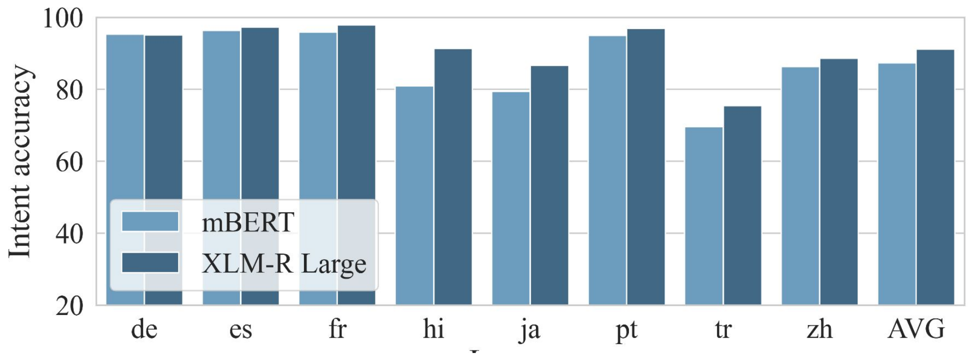
Multimodality

Seung-won Hwang
Professor
Department of CSE, Seoul National University

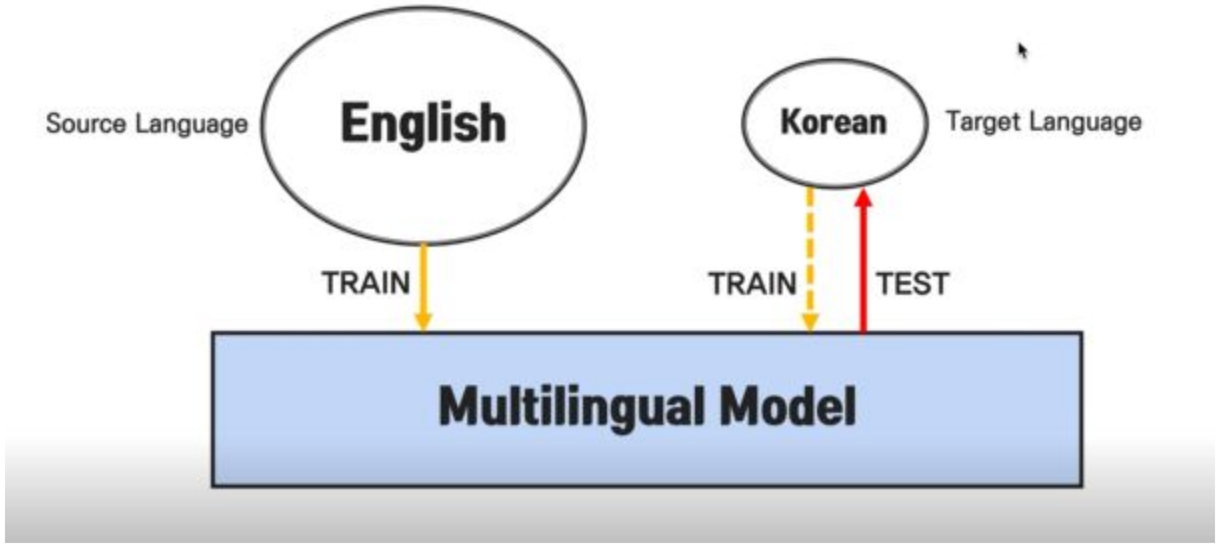
Cross-lingual Transfer #1: zero-shot



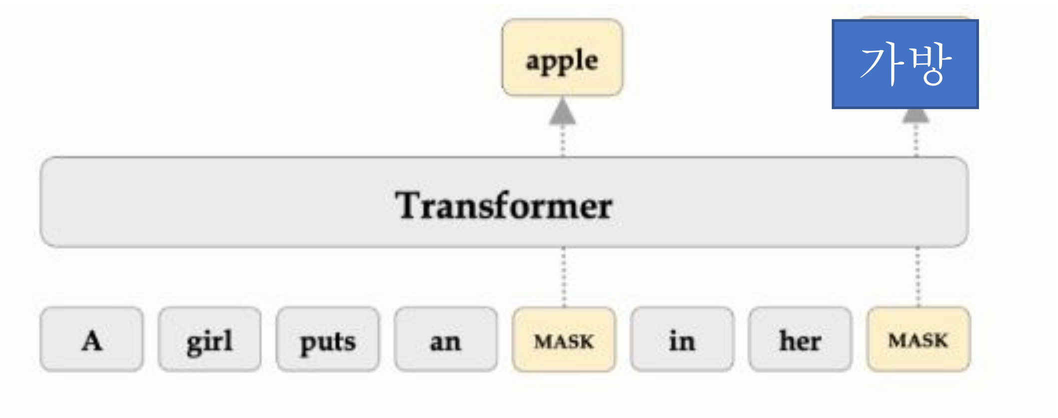


#2: Between two languages

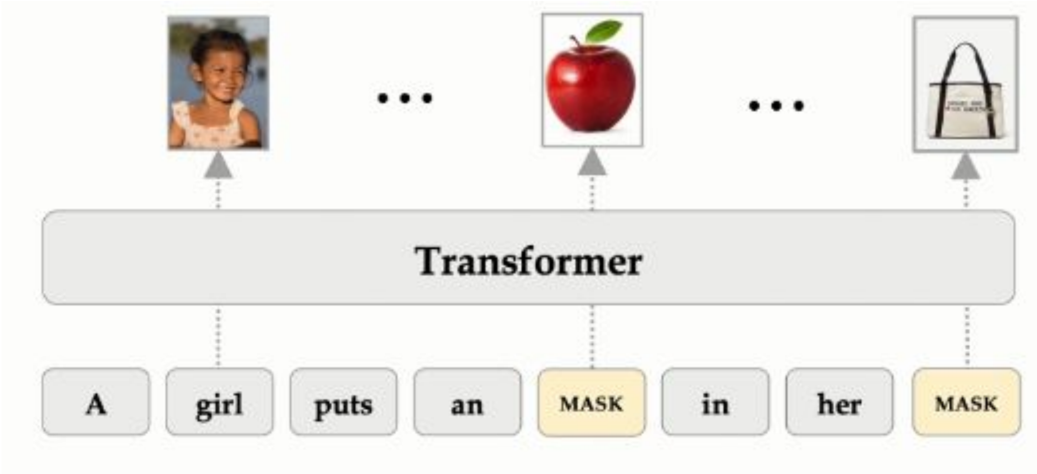
- Curse of multilinguality: Performs poorly on low-resourced
- Can we choose good source language to transfer from? => presentation



Multimodality vs Multilinguality



Monolingual



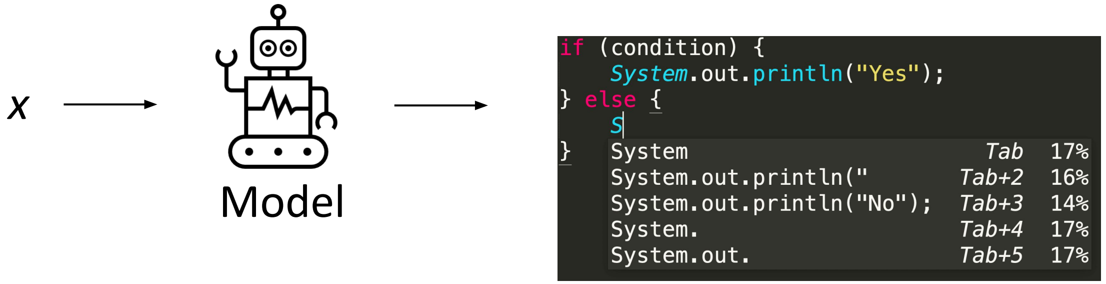
Parallel: En-Ko, Image-caption

Motivation: Code Intelligence

- 100s of millions of repositories of code+text
- Motivating multimodal representation
  - Code-code: Find/generate related code during development
  - Text-code: Generate code by natural language, summarize code into text

Partial code

Textual description



Resources

Category	Task	Dataset Name	Language	Train/Dev/Test Size	Baselines	Task definition
Code-Code	Clone Detection	BigCloneBench	Java	900K/416K/416K	CodeBERT	Predict semantic equivalence for a pair of codes.
		POJ-104	C/C++	32K/8K/12K		Retrieve semantically similar codes.
	Defect Detection	Devign	C	21k/2.7k/2.7k		Identify whether a function is vulnerable.
	Cloze Test	CT-all	Python, Java, PHP, JavaScript, Ruby, Go	-/-/176k		Tokens to be predicted come from the entire vocab.
		CT-max/min	Python, Java, PHP, JavaScript, Ruby, Go	-/-/2.6k		Tokens to be predicted come from {max, min}.
	Code Completion	PY150	Python	100k/5k/50k	CodeGPT	Predict following tokens given contexts of codes.
		GitHub Java Corpus	Java	13k/7k/8k		
	Code Repair	Bugs2Fix	Java	98K/12K/12K	Encoder-Decoder	Automatically refine codes by fixing bugs.
	Code Translation	CodeTrans	Java-C#	10K/0.5K/1K		Translate the codes from one programming language to another programming language.
Text-Code	NL Code Search	CodeSearchNet, AdvTest	Python	251K/9.6K/19K	CodeBERT	Given a natural language query as input, find semantically similar codes.
		CodeSearchNet, WebQueryTest	Python	251K/9.6K/1k		Given a pair of natural language and code, predict whether they are relevant or not.
	Text-to-Code Generation	CONCODE	Java	100K/2K/2K	CodeGPT	Given a natural language docstring/comment as input, generate a code.
Code-Text	Code Summarization	CodeSearchNet	Python, Java, PHP, JavaScript, Ruby, Go	908K/45K/53K	Encoder-Decoder	Given a code, generate its natural language docstring/comment.
Text-Text	Documentation Translation	Microsoft Docs	English-Latvian/Danish/Norwegian/Chinese	156K/4K/4K		Translate code documentation between human languages (e.g. En-Zh), intended to test low-resource multi-lingual translation.

Limitation of MLM Objective for Source Code

- **Source Code** is more structured compared to **Natural Language**.
- Representing/Learning Source Code as a s**eries of Text Token is not viable**.
- **Code Semantics** may not be properly represented

Various Components in Source Code Compilation

- Lexer - Takes in series of characters and converts then into a Lexical Token.
- Parser - Converts Lexical Tokens into Syntax Trees, by incubating structure in them.
- Translator - Translates AST to lower level Code.
- Optimizer - Optimization of the given piece of Lower Language Code(Three Address Code).
- Compiler - Converts Optimized Code into Binary instruction(Machine Code)

Various Representations of Code:

1. Raw Text Tokens - Human Readable Version
2. Abstract Syntax Tree - Tree data structure which captures, structure of the given code.
3. Data Flow Graph - Captures the Data Interaction/Transfer in a given code. It includes variables and hardcoded values.
4. Control Flow Graph - Each node is a statement, captures the probable control flow from each statement.
5. Executional Flow Graph - Each node is a statement, captures the exact transfer of Control which executing the code.

Dynamic Representation
Static Representation

Prelude - CodeBERT

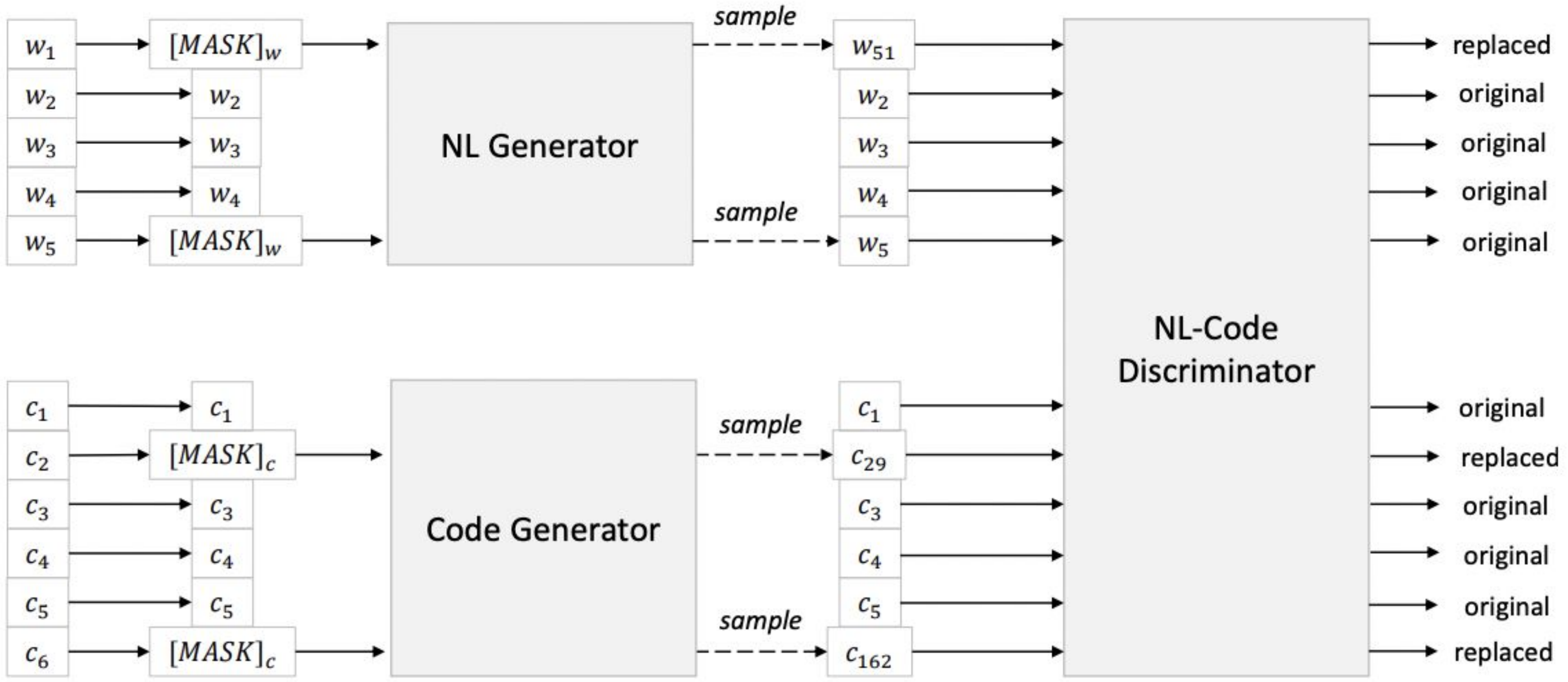
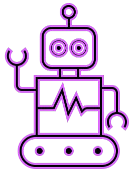


Figure 2: An illustration about the replaced token detection objective. Both NL and code generators are language models, which generate plausible tokens for masked positions based on surrounding contexts. NL-Code discriminator is the targeted pre-trained model, which is trained via detecting plausible alternatives tokens sampled from NL and PL generators. NL-Code discriminator is used for producing general-purpose representations in the fine-tuning step. Both NL and code generators are thrown out in the fine-tuning step.

Example: Copilot

Prompt: Natural language description (+ Input-output

```
def solution(lst):
    """Given a non-empty list of integers, return the sum of all of the odd elements
    that are in even positions.

    Examples
    solution([5, 8, 7, 1]) ==>12
    solution([3, 3, 3, 3, 3]) ==>9
    solution([30, 13, 24, 321]) ==>0
    """
```



Copilot

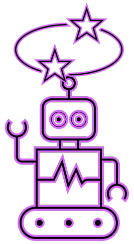
```
def solution(lst):
    """Given a non-empty list of integers, return the sum of all of the odd elements
    that are in even positions.

    Examples
    solution([5, 8, 7, 1]) ==>12
    solution([3, 3, 3, 3, 3]) ==>9
    solution([30, 13, 24, 321]) ==>0
    """
    return sum(lst[i] for i in range(0,len(lst)) if i % 2 == 0 and lst[i] % 2 == 1)
```

Programming language

Example: Copilot (fail case)

Prompt: Natural language

```python
def do_work(x, y, z, w):
    """ Add 3 to y, then subtract 4
    from both x and w. Return the
    product of the four numbers. """
```



Copilot

```python
def do_work(x, y, z, w):
    """ Add 3 to y, then subtract 4
    from both x and w. Return the
    product of the four numbers. """
    t = y + 3
    u = x - 4
    v = z * w
    return v
```

Incorrect Programming language

Our Research Question

- Current annotation pairs up code-text as a sequence pair
- But there are other ways to explain (z) how code works
  - Abstract syntax tree (AST)
  - Data flow graph (DFG)
  - Pseudo code
- Instead of annotating (x,y), enriching annotation into (x,y,z) may robustify training
- We had success in related NLP problems!

Dissecting DFG

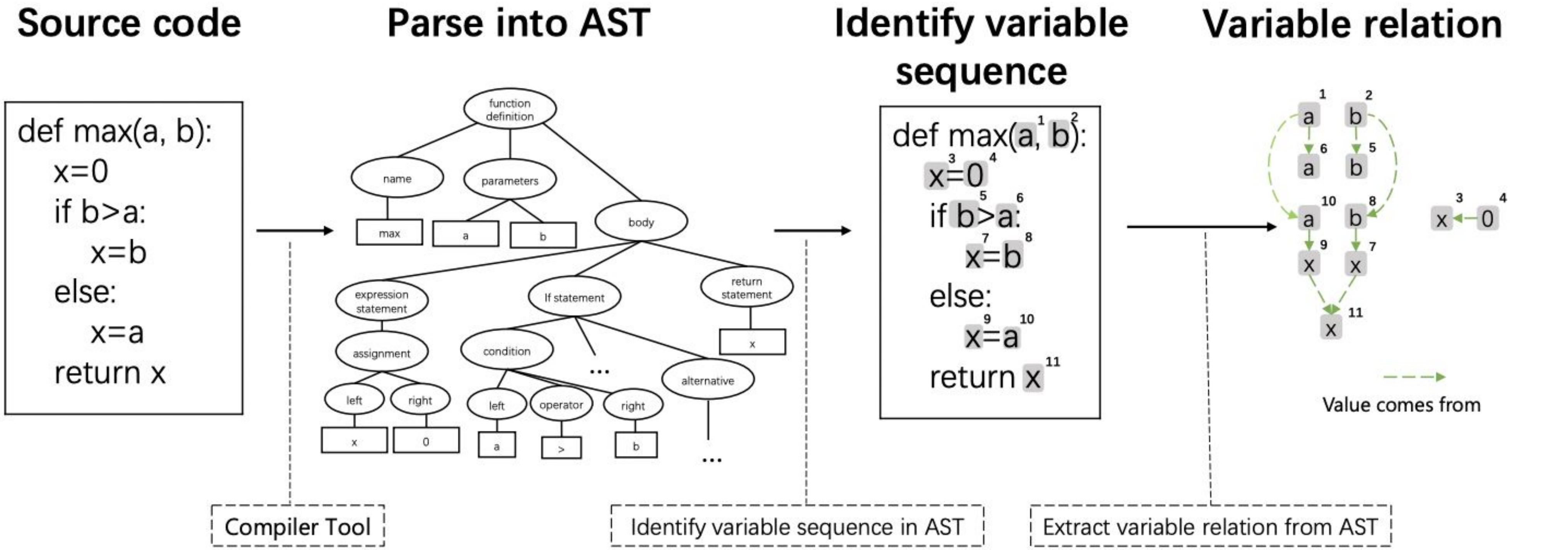


Figure 1: The procedure of extracting data flow given a source code. The graph in the rightmost is data flow that represents the relation of "where-the-value-comes-from" between variables.