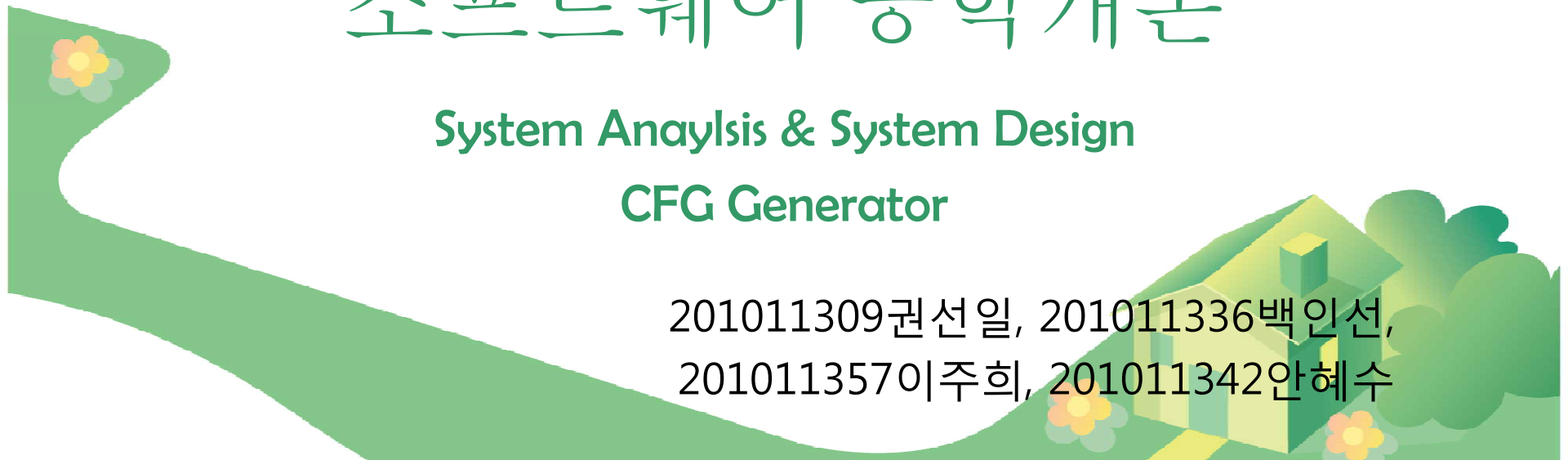


# 소프트웨어 공학개론

System Analysis & System Design  
CFG Generator

201011309권선일, 201011336백인선,  
201011357이주희, 201011342안혜수



# Statement of Purpose

1. CFG가 만들어지기 시작하면, 화면에 시작한다는 **Message**를 출력해준다.
2. CFG가 만들어지면, 화면에 **Success Message**만 보여주고, CFG는 **File**에 저장한다.
3. CFG를 만드는데 실패하면 **Error Message**를 보여주고, 모든 **Process**를 종료한다.
4. 무조건 **C source code**만을 다룬다.(\*.c)
5. **Main() Function** 내부만 CFG로 만든다.
6. '{}'와 같은 **Block**이 없거나, 입력된 **File** 경로가 올바르지 않은 경우, CFG를 만들어 없으므로, **CFG Generator**가 에러로 처리되어 종료된다.

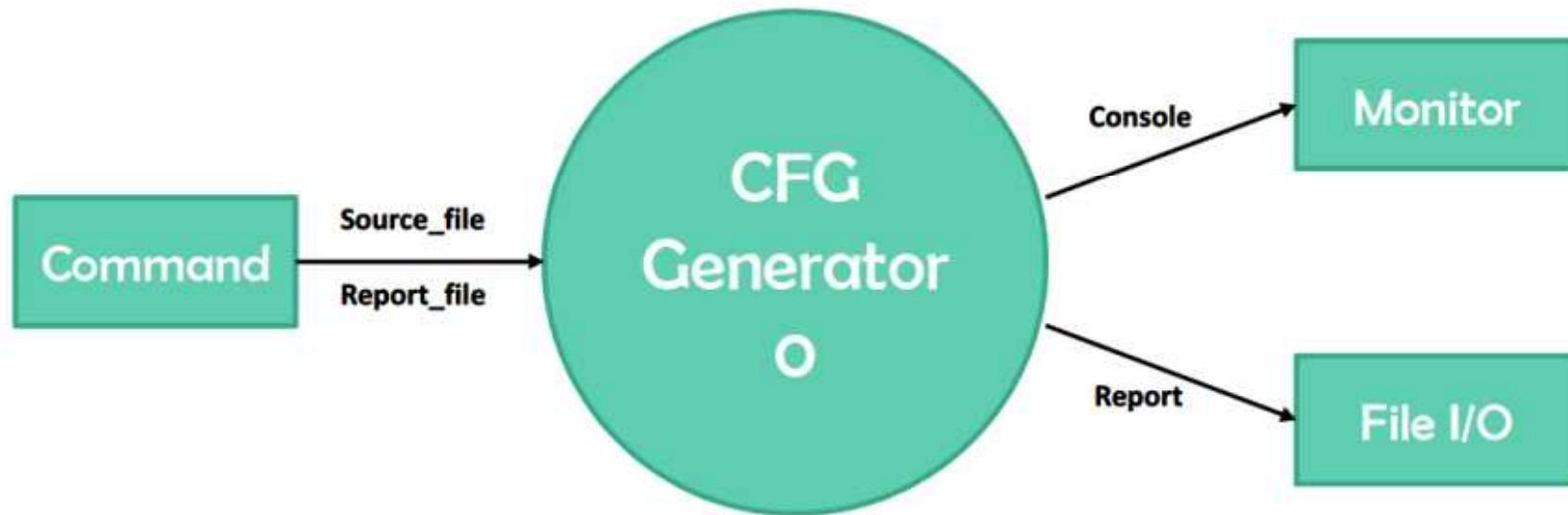
# Statement of Purpose

7. 전체 **Compile Error**는 고려하지 않고, **CFG**를 만드는데 필요한 경우의 **Error**만 고려한다.('{ }'와 같은 **Block**이 하나 모자라거나, **Main()**이 없는 경우)
8. **CFG**가 만들어지면, **File**에 **CFG**의 **BB List**와 **Edge List**를 한 줄씩 저장한다.
9. **CFG**의 저장방식은 **BB List**는 '[ { (Source) , [ (In-Edge) ], [ (Out-Edge) ] } , ..., { ... } ]]', **Edge List**는 '[ { (Start BB), (End BB) } , ..., { ... } ]'이다.
10. **Help Message**는 **C Source Code**의 **File** 경로가 틀렸을 경우 출력된다.
  - **Help Message**는 해당 형태로 출력된다.  
`./cg <source code filepath(*.c)> <report filepath(*.txt)>`

# Statement of Purpose

- **Statement**는 이전 세미콜론(;)부터 다음 세미콜론(;)을 만나기 전까지를 나타낸다.
- **Basic Block**은 **CFG**를 이루는 기본 단위이다. **BB** 내부에 데이터는 **CFG** 자체가 될 수도 있고, 하나의 **Statement**가 될 수도 있다.
- **Control Flow Graph**(이하 **CFG**)는 **Basic Block**들을 **Edge**로 연결시켜놓은 **Graph**를 나타낸다. 이 **CFG**는 **BB** 내부에 또 다른 **CFG**를 갖고 있을 수 있다.

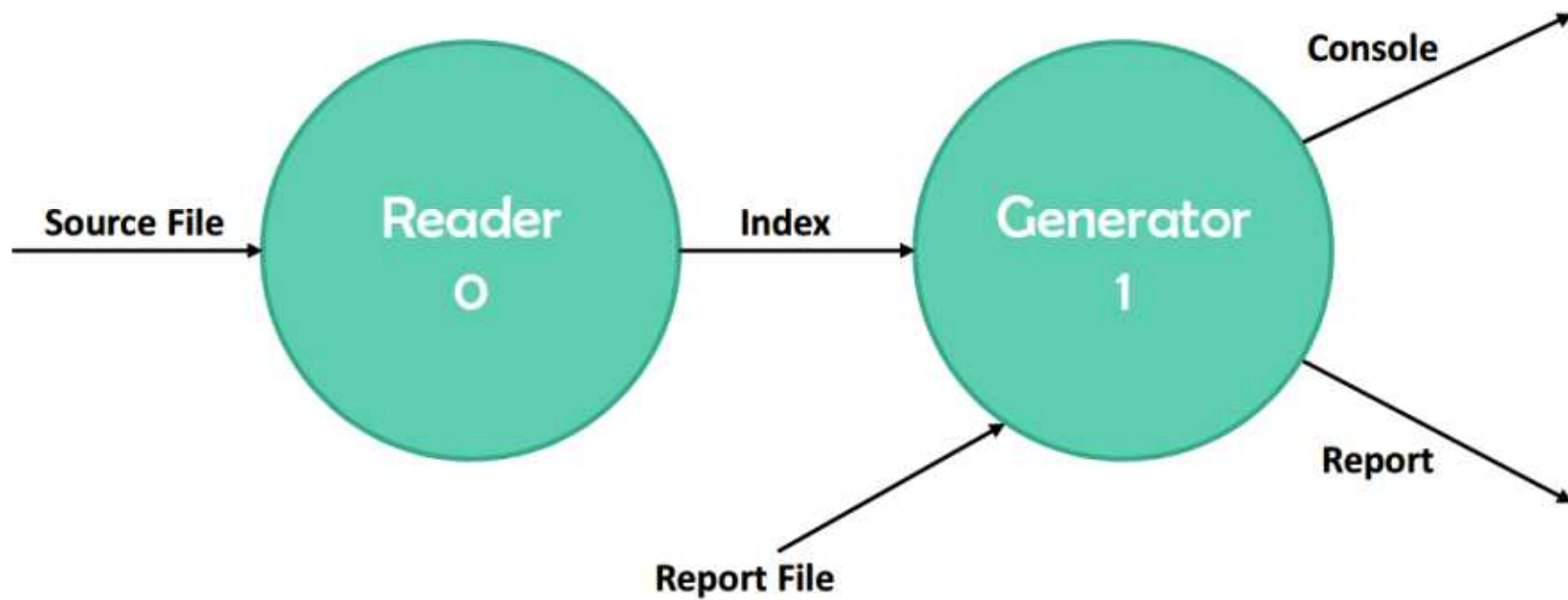
# System Context Diagram



# Event List

Input/Output Event	Description	Format/Type
Source_File	C Source Code의 File 경로	문자열(*.c)
Report_File	완성된 CFG가 출력되는 File 경로	문자열(*.txt)
Console	Success/Start/Error Message	Console에 출력되는 문자열
Report	CFG의 Basic Block들과 Edge들을 출력하는 문자열	File에 출력되는 문자열

# DFD – Level 0



# Level 0 - Data Dictionary

Input/Output Event	Description	Format/Type
Index	C Source Code를 Generator에서 읽기 쉽게 바꿔준다. Index는 typeNum, source, BlockNum, In/Out EdgeNum 을 가지고 있다	Type과 Source를 가지고 있는 구조체 형태이다. (1) TypeNum 1-1-1 if : type = 1.11 1-1-2 else if : type = 1.12 1-1-3 else : type = 1.13 1-2 switch : type = 1.2 1-3 for : type = 1.3 1-4 while : type = 1.4 1-5 일반statement = type = 1.0 (2) Source : 해당 index의 source (3) BlockNum : 해당 Index Block의 Number (4) In/OutEdgeNum : 해당 Index의 Block In/Out Edge Number



# Level 0 - Specification

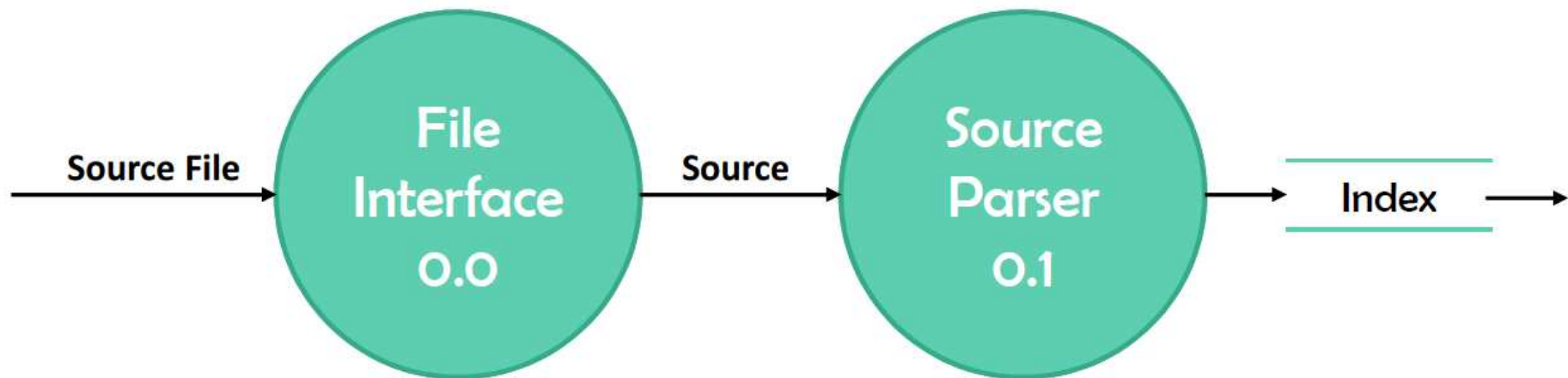
Name	Reader
Reference Number	0
Input	Source_File
Output	Index(+Data Structure)
Description	Source File을 받아서, Parsing하여 Index로 돌려준다. 만약 Source File의 경로가 잘못되어 있을 경우 (해당 파일이 존재하지 않거나, 파일의 형식이 *.c가 아닐 경우), Index에 NULL을 넣는다.

# Level 0 - Specification

Name	Generator
Reference Number	1
Input	Index(+Data Structure)
Output	Message, Report
Description	Index를 받아 Basic Block을 만들어가며, CFG를 만들어 Report_File에 쓴다. 또한 경우에 따라서 Console Message(Start, Success, Fail)를 출력한다.

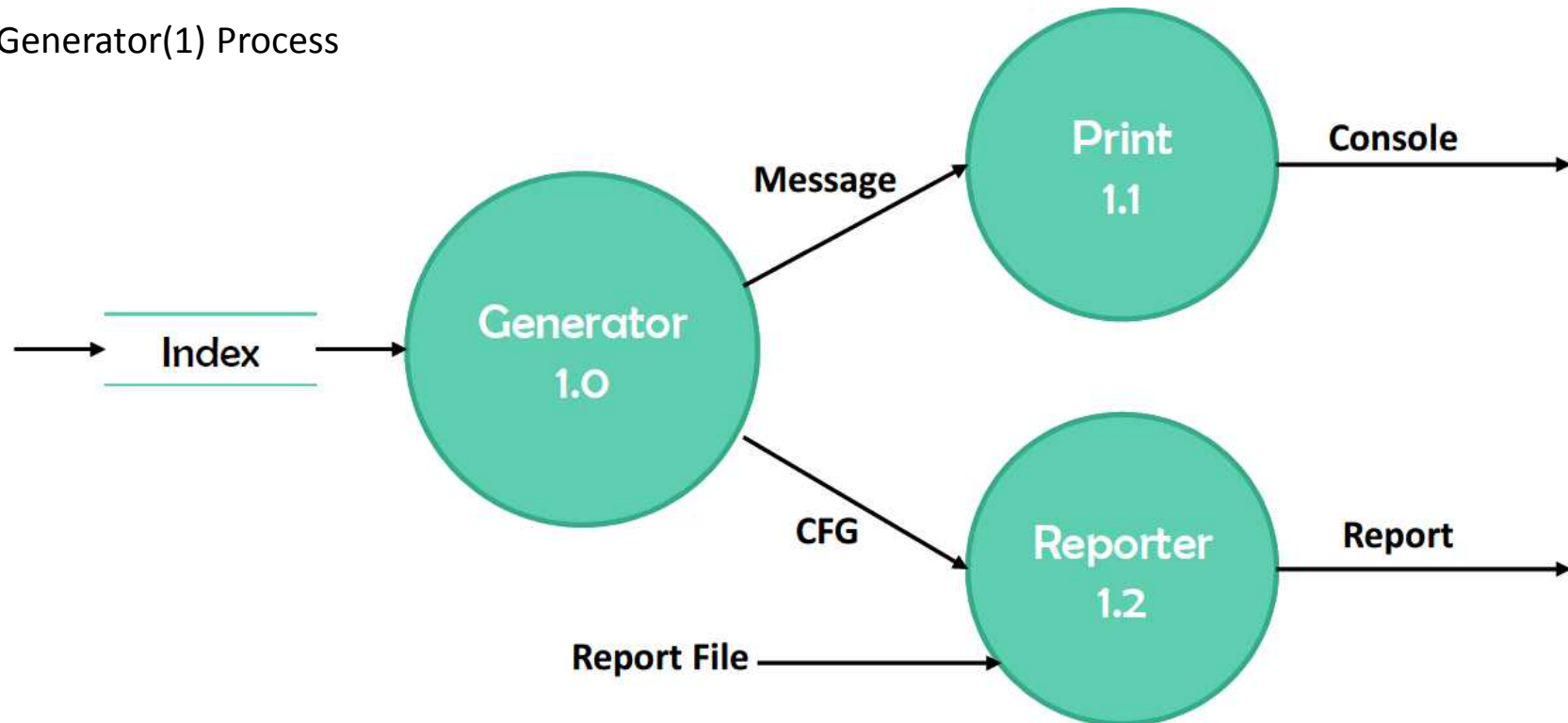
# DFD – Level 1

- Reader Process



# DFD – Level 1

- Generator(1) Process



# Level 1 - Data Dictionary

Input/Output Event	Description	Format/Type
Source	Source_File을 쪽 읽어서, 문자열 형태로 나타내어진다.	문자열
Message	Console에 표시할 Message를 나타낸다.	문자열
CFG	Generator에서 Index를 읽어서 만들어낸 Graph	문자열(Graph)

# Level 1 - Specification

Name	File Interface
Reference Number	0.0
Input	Source_File
Output	Source
Description	Source_File의 파일을 다 읽어서, 문자열인 Source로 넘겨준다. 만약 경로 잘못되어 있을 경우, Source에 null이 들어가게 된다.

# Level 1 - Specification

Name	Source Parser
Reference Number	0.1
Input	Source
Output	Index
Description	Source를 읽어 들여서, Index로 변환한다. 이때, Source가 null일 경우, Structure Index 전체가 NULL이 된다.

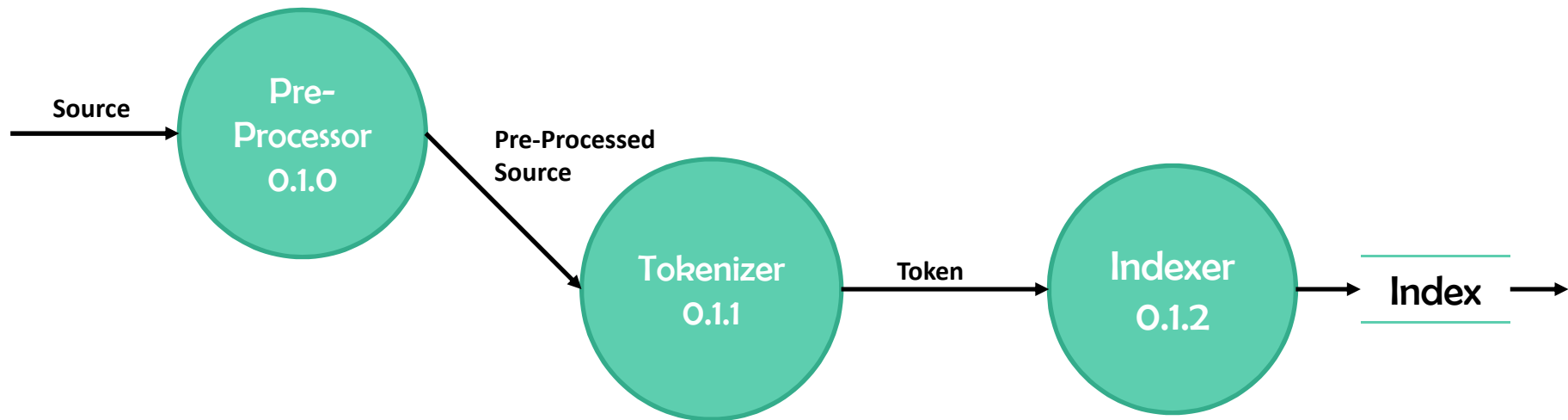
# Level 1 - Specification

Name	Generator
Reference Number	1.0
Input	Index
Output	Message, CFG
Description	Index를 읽어서 CFG로 변환한다. Message : "Start" - CFG를 만들기 시작할 때, "Success" - CFG생성이 성공적으로 끝났을 때, "Fail" - Index의 Source가 잘못 되어 있을 때({}의 쌍이 맞지 않을 때 등), index가 NULL일 때, index가 부적절할 때



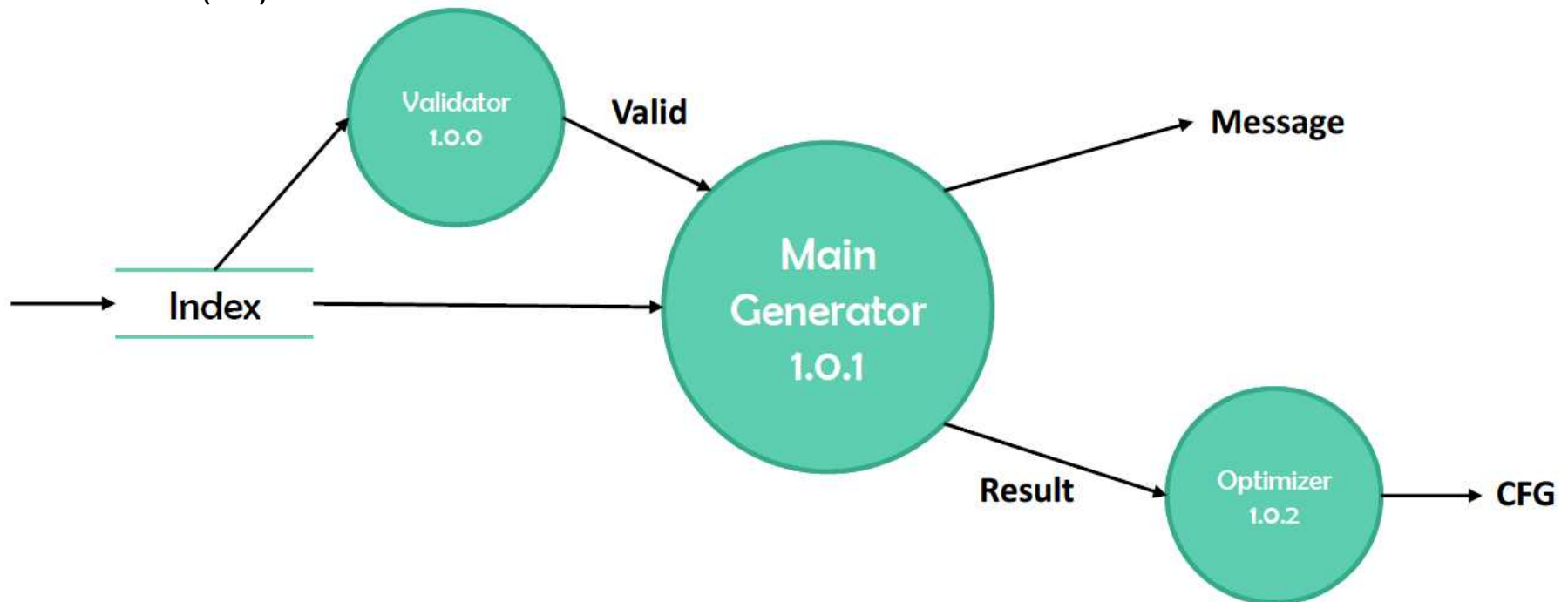
# DFD – Level 2

- Source Parser Process



# DFD – Level 2

- Generator(1.0) Process



## Level 2 - Data Dictionary

Input/Output Event	Description	Format/Type
Pre-processed Source	주석과 빈칸들이 지워진 <b>C Source Code</b> 이다.	문자열
Token	{, }, ;로 나눈 문자열 조각의 리스트이다. 만약, null일 경우, 파일을 읽기가 실패 한경우다.	문자열(+Data Structure)
Valid	해당 <b>Index</b> 가 정확한지 <b>Check</b> 하여 저장된다	True/False
Result	<b>Generator</b> 에서 각 조건들로부터 만들어지는 조각 <b>CFG</b> 이다. <b>Optimizer</b> 로 모여서 <b>CFG</b> 로 만들어진다.	조각 <b>CFG</b>

# Level 2 - Specification

Name	Pre-Processor
Reference Number	0.1.0
Input	Source
Output	Pre-processed Source
Description	Source를 받아서, 주석과 공백, 엔터를 지운 후, Pre-processed Source에 넣어준다. Source가 null인 경우(파일이 존재하지 않거나, 내용이 없는 경우) output은 NULL이다.

# Level 2 - Specification

Name	Tokenizer
Reference Number	0.1.1
Input	Pre-processed Source
Output	Token(+Data Structure)
Description	Pre-processed Source를 '{', '}', ';'로 나누어 토큰화한 결과를 Token에 저장한다. 만약, Pre-processed Source가 null이 넘어올 경우, Token에 null을 집어넣어, 예러처리한다.

# Level 2 - Specification

Name	Indexer
Reference Number	0.1.2
Input	Token(+Data Structure)
Output	Index
Description	Token을 처리하여, Index로 변환하는 작업을 말한다.

# Level 2 - Specification

Name	Validator
Reference Number	1.0.0
Input	Index
Output	Valid
Description	<p>Index들을 한번씩 읽어 들여, 해당 Index가 올바른 것인지 검사한다. 올바르지 않다면 False를, 올바르다면 True를 넘겨준다.</p> <ul style="list-style-type: none"><li>-False를 넘겨주는 경우 : TypeNum == NULL    Source == NULL    BlockNum == NULL    InEdgeNum == NULL    OutEdgeNum == NULL</li><li>- True를 넘겨주는 경우 : False가 아닐 경우</li></ul>

# Level 2 - Specification

Name	Main Generator
Reference Number	1.0.1
Input	Valid, Index
Output	Message, Result
Description	Valid가 True라면, Message에 "Start"를 출력하고, CFG생성을 시작한다. Valid가 False이면, Message에 "Fail"를 출력한다. 그리고 각 Parsing에서의 결과물인 Result는 Optimizer에 모여 합쳐져서 CFG로 만들어진다.

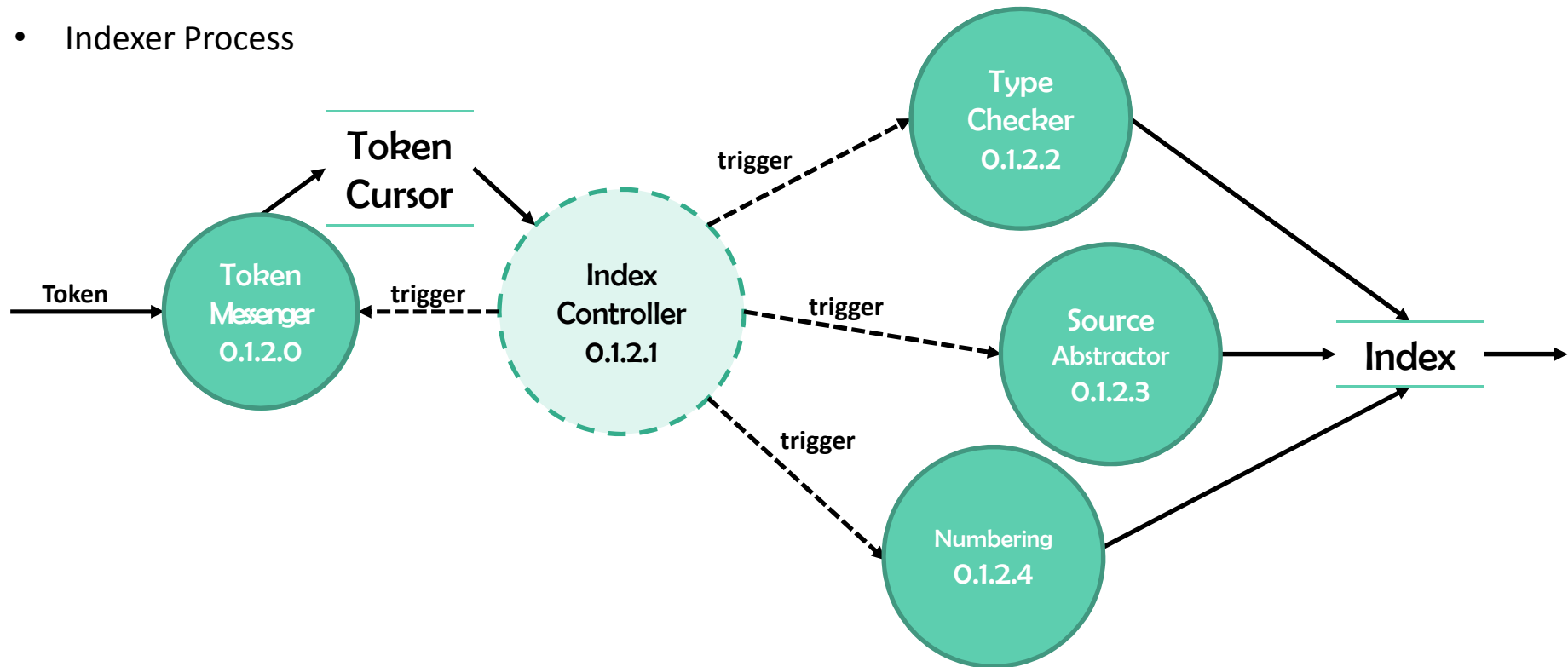


## Level 2 - Specification

Name	Optimizer
Reference Number	1.0.2
Input	Result
Output	CFG
Description	모든 <b>Result</b> 가 넘어오면, 기존 <b>CFG</b> 에 합친다. <b>CFG</b> 가 완성되면 최적화시켜 <b>CFG</b> 를 돌려준다.

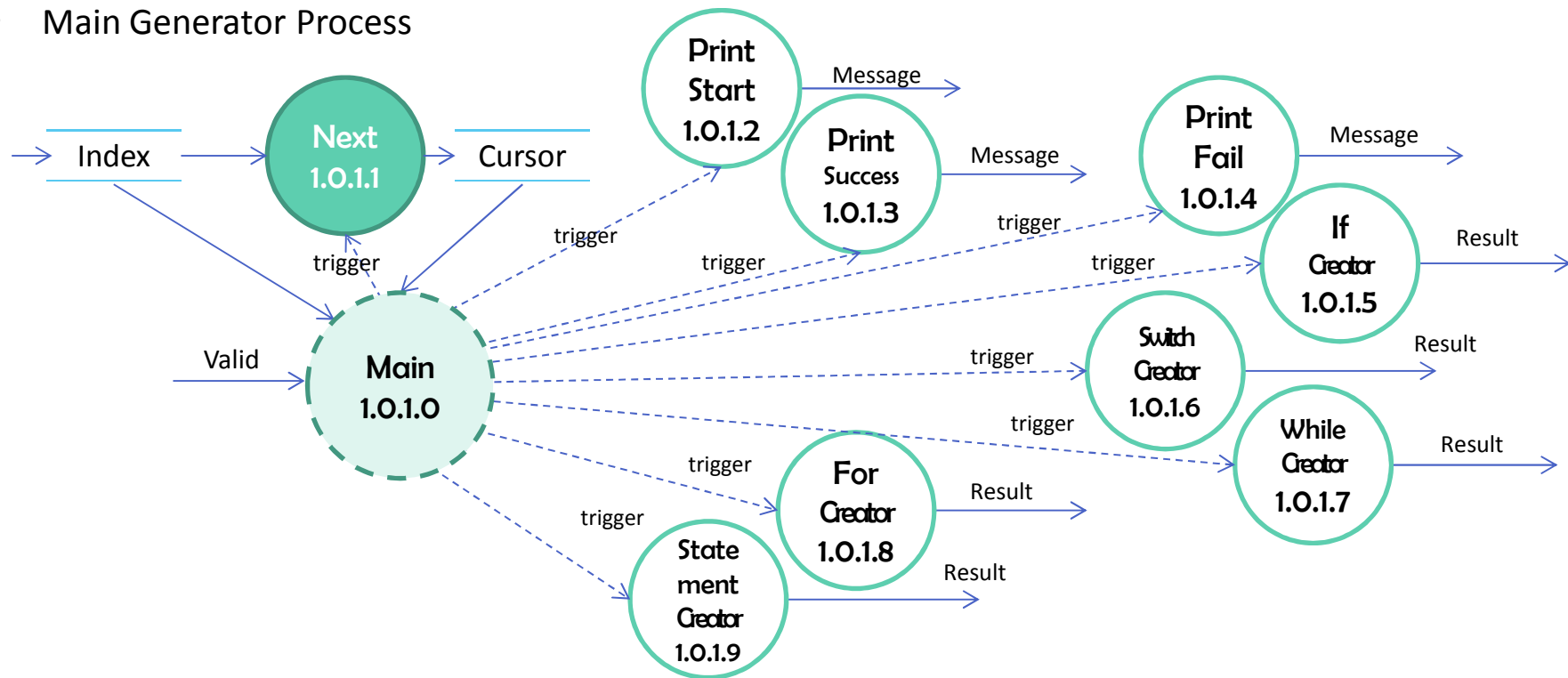
# DFD – Level 3

- Indexer Process



# DFD – Level 3

- Main Generator Process



## Level 3 - Data Dictionary

Input/Output Event	Description	Format/Type
Token Cursor	Token 하나를 가르키는 포인터.	'Err', 'Start', 'End' / 포인터 OR 상수
Cursor	Index를 List 형태로 하나씩 앞으로 옮기며 반환하는 Pointer	Pointer

# Level 3 - Specification

Name	Token Messenger
Reference Number	0.1.2.0
Input	Token(+Data Structure)
Output	Token Cursor
Description	Token을 받아, 컨트롤러에서 Trigger할 때 마다, 해당하는 Token Cursor를 넘겨준다. 만약 Token이 null일 경우, Token Cursor에는 'Err'이라는 상수가 들어가게 된다. Token Cursor의 맨 앞과 마지막에는 'Start'와 'End'라는 상수가 들어간다.

# Level 3 - Specification

Name	Index Controller
Reference Number	0.1.2.1
Input	Token Cursor
Output	trigger
Description	Token Cursor를 받아서, 해당 Token Cursor에 해당하는 작업을 한다.

# Level 3 - Specification

Name	TypeChecker
Reference Number	0.1.2.2
Input	Trigger
Output	Index(+Data Structure)
Description	If, else if, else / switch / while / for과 같은 분기문과 일반 statement를 구분하여 index 내의 type을 결정한다.

# Level 3 - Specification

Name	Source Abtractor
Reference Number	0.1.2.3
Input	Trigger
Output	Index
Description	Token화 된 구문에서 CFG의 내용에 해당하는 Source를 추출하여 Index내의 Source에 넣는다.



# Level 3 - Specification

Name	Numbering
Reference Number	0.1.2.4
Input	Trigger
Output	Index(+Data Structure)
Description	CFG를 구성할 Basic Block과 그 Block의 in-edge, out-edge에 각각 Number를 부여하여 Index에 넣는다.

# Level 3 - Specification

Name	Main
Reference Number	1.0.1.0
Input	Index, Cursor, Valid
Output	Trigger
Description	<p>Cursor의 TypeNum을 확인하고, 각 조건마다 해당하는 Creator를 Trigger한다.</p> <p>또한, Valid가 False일 경우, Print Fail을 Trigger하고, 가장 처음 넘겨지는 Valid가 True일 경우, Print Start를 Trigger하고, 마지막 넘겨지는 Valid가 True일 경우(모든 코드가 CFG로 변환 된 경우, 정상종료가 가능한 경우) Print Success를 Trigger한다.</p>

# Level 3 - Specification

Name	Next
Reference Number	1.0.1.1
Input	Index
Output	Cursor
Description	해당 Index를 받아서 한번 호출될 때마다 다음으로 넘어가며 해당 위치의 Cursor를 돌려준다.

# Level 3 - Specification

Name	Print Success
Reference Number	1.0.1.2
Input	Trigger
Output	Message
Description	Trigger되면 Message에 "Success"를 넣어준다.

Name	Print Fail
Reference Number	1.0.1.3
Input	Trigger
Output	Message
Description	Trigger되면 Message에 "Fail"를 넣어준다.

# Level 3 - Specification

Name	Print Start
Reference Number	1.0.1.4
Input	Trigger
Output	Message
Description	Trigger되면 Message에 "Start"를 넣어준다.

# Level 3 - Specification

Name	If Creator
Reference Number	1.0.1.5
Input	Trigger
Output	Result
Description	Trigger되면 If구문에 맞는 CFG를 구성한다.

Name	Switch Creator
Reference Number	1.0.1.6
Input	Trigger
Output	Result
Description	Trigger되면 Switch구문에 맞는 CFG를 구성한다.

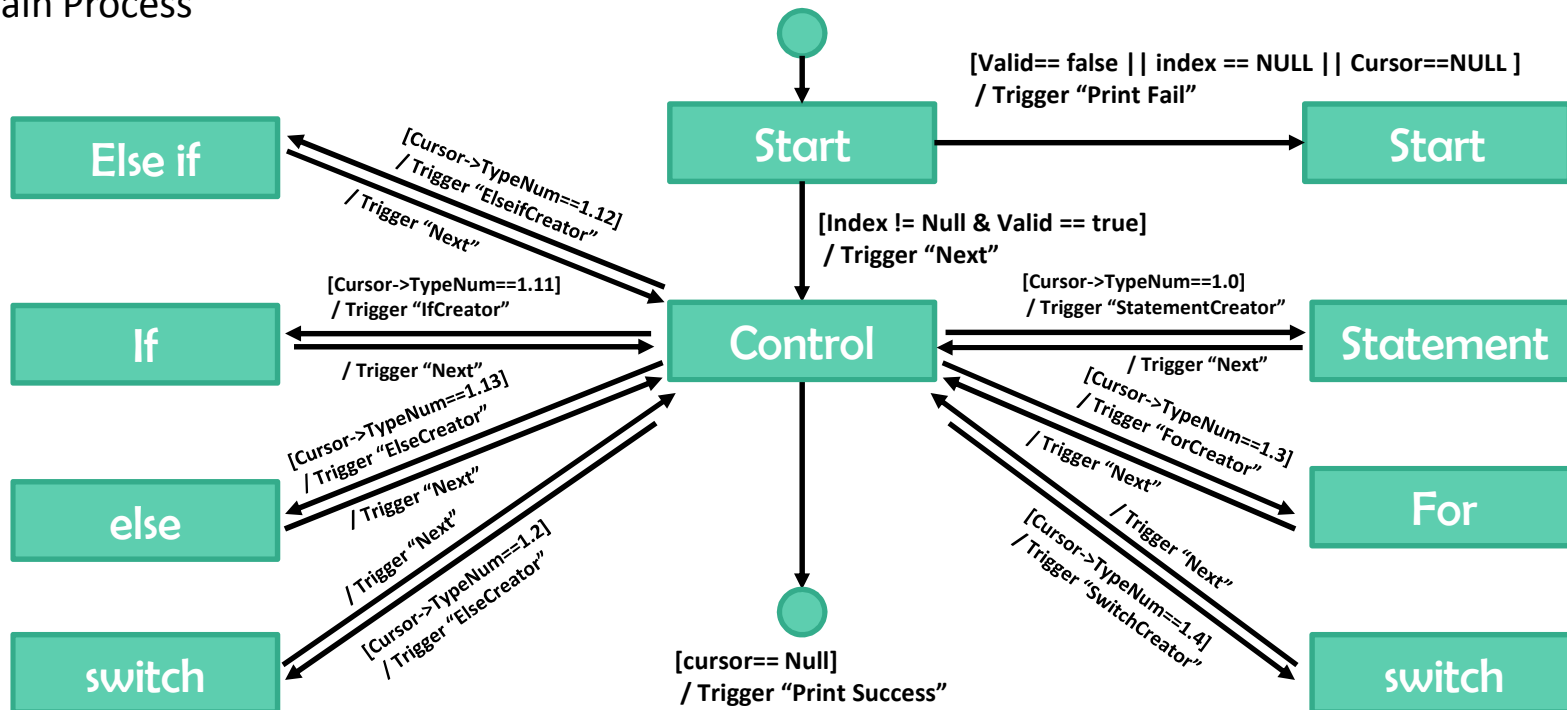
# Level 3 - Specification

Name	While Creator
Reference Number	1.0.1.7
Input	Trigger
Output	Result
Description	Trigger되면 While구문에 맞는 CFG를 구성한다.

Name	For Creator
Reference Number	1.0.1.8
Input	Trigger
Output	Result
Description	Trigger되면 For구문에 맞는 CFG를 구성한다.

# FSM – Level 4

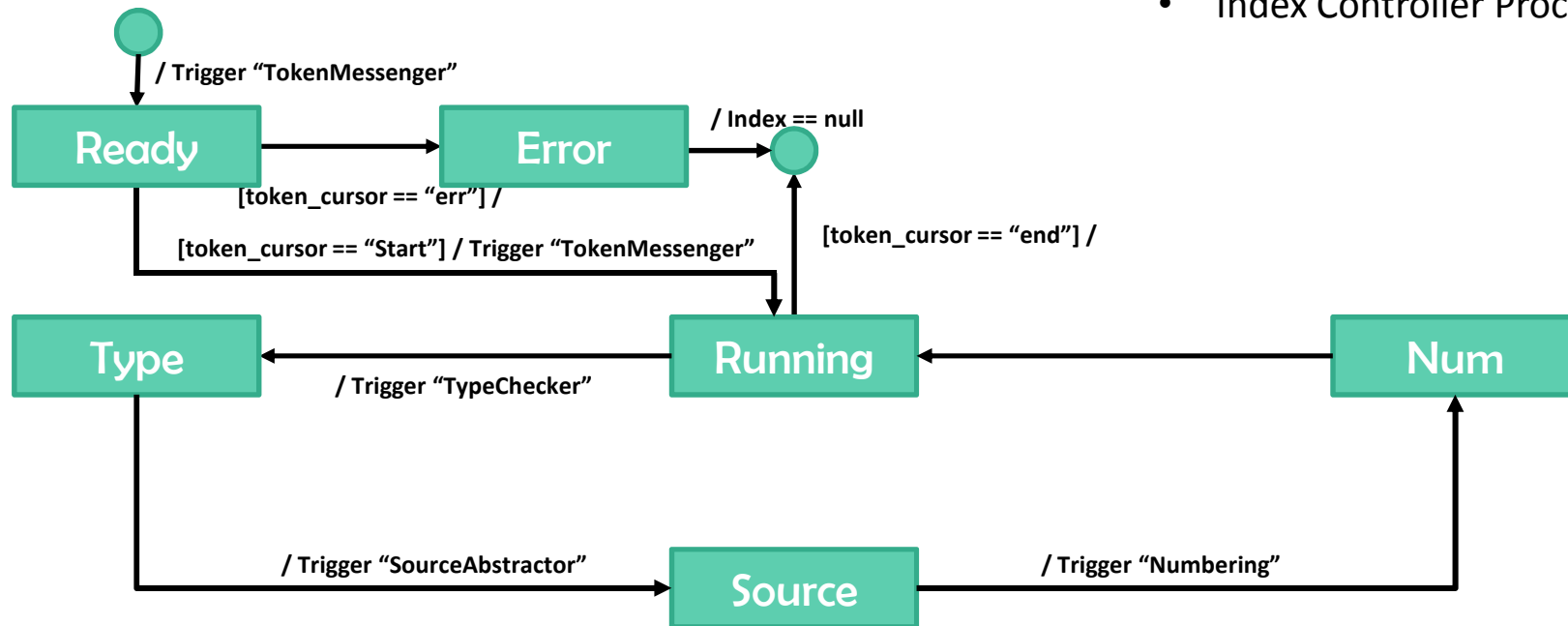
- Main Process



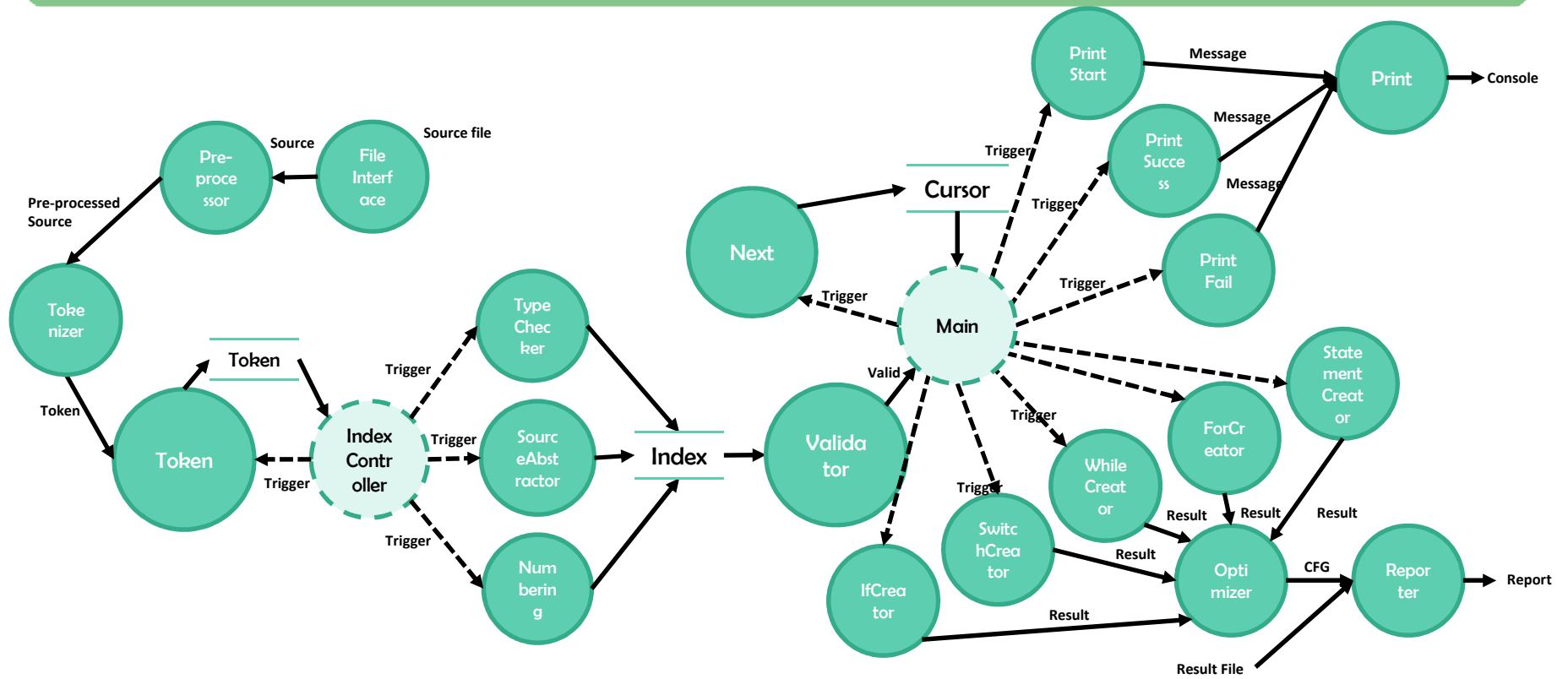


# FSM – Level 4

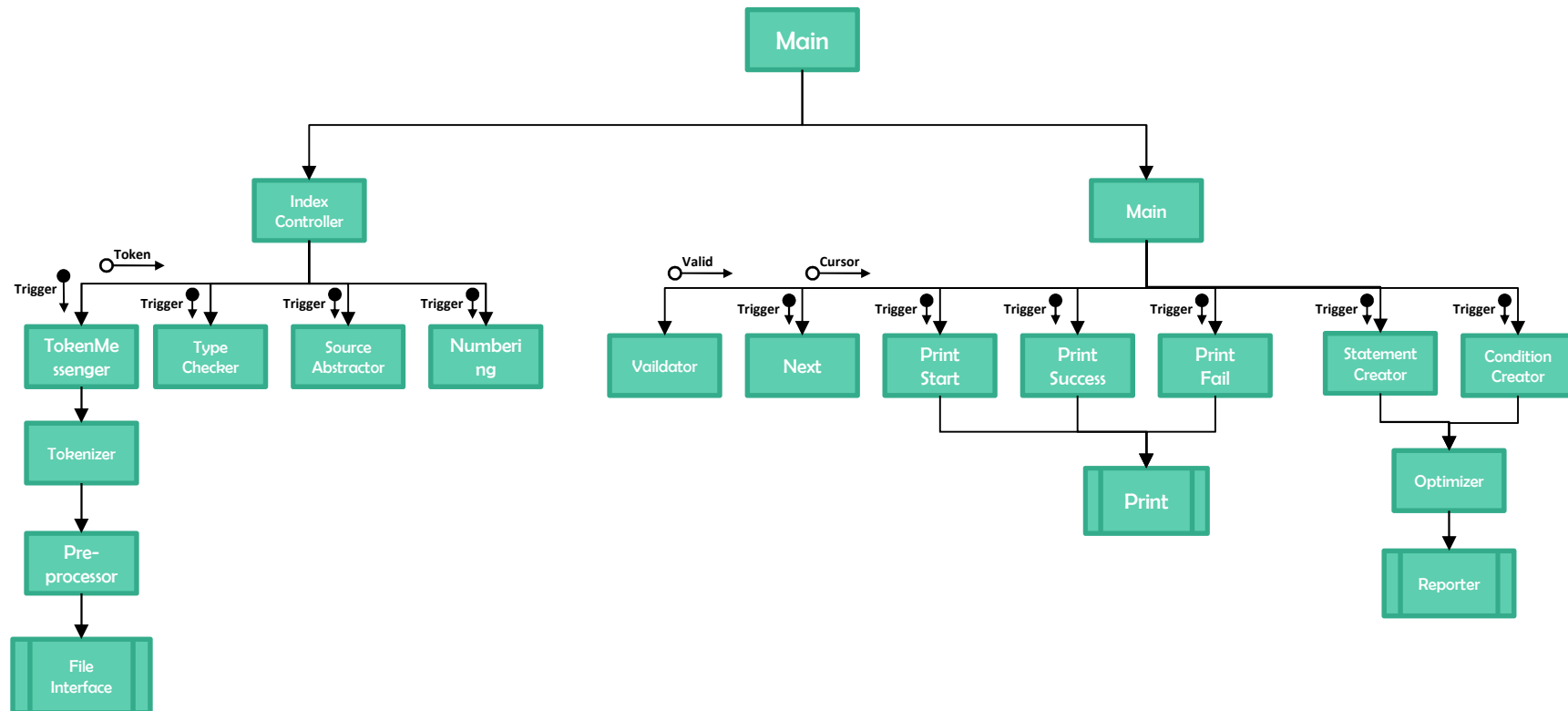
- Index Controller Process



# DFD



# Structured Charts



**Thank You!!**

