

---

# 요리용 코딩언어 개발

---

2025 전기 졸업과제 중간보고서



## Team Cook

정보컴퓨터공학부 202055570 윤선재

정보컴퓨터공학부 201824598 채문석

정보컴퓨터공학부 202055619 팜민두웅

지도교수: 조환규

## 목차

1. 과제 개요.....	2
I. 연구 배경 .....	2
II. 선행 연구 .....	2
III. 기대 효과.....	5
2. 과제 구성.....	7
I. 과제 목표 .....	7
II. 기존 요구 사항 및 수정 사항.....	8
III. 설계 상세화 .....	9
IV. 개발 환경.....	10
V. 사용 기술 .....	10
VI. 예상 구조.....	11
3. 수행 계획.....	12
I. 갱신된 개발 일정 .....	12
II. 구성원별 진척도 .....	12
4. 과제 수행 내역 및 중간 결과 .....	13
5. 진행 상황.....	14
6. 참고 문헌.....	15

# 1.과제 개요

## I. 연구 배경

요리 레시피는 본디 자연어로 작성되는데, 이는 인간이 이해하기는 쉽지만 많은 모호성과 생략된 정보를 내포하고 있어 일관된 해석을 하기 어려운 구조를 지니고 있다. 이를 해결하기 위해 지난 20년간 도메인 특화 언어(DSL)로 레시피를 형식화하는 방법이 활발히 연구되고 있다.

최근 인공지능 및 컴퓨터 비전 기술의 발전으로 널리 개발되고 있는 로봇을 이용한 자동화 조리는 많은 관심을 받고 있다. 이와 더불어 자동화 조리에 사용되는 요리 레시피에 대한 관심도 커지고 있다. 기존의 자연어 기반 요리 레시피의 경우 특유의 모호성과 불명확성 등이 기계 해석에 있어서 약점으로 작용하기 때문에 이를 해결하는 방법으로 새로운 요리 레시피의 표현법이 주목받고 있다.

본 보고서는 이러한 최신 연구 및 상용 사례를 중심으로 요리 레시피의 코드화, 그리고 AI·로보틱스 기반 조리 자동화 시스템의 동향을 살펴본 후, 이를 개선하는 방법을 제시하려고 한다.

## II. 선행 연구

### 1. 도메인 특화 언어(DSL)와 레시피 코드화 연구

요리 과정을 컴퓨터가 이해할 수 있는 도메인 특화 언어(DSL)로 표현하려는 시도는 여러 차례 있었다. 이러한 DSL은 레시피의 재료, 조리 단계, 조리 기구 등을 형식화하여 알고리즘처럼 기술하고자 한다. 기존에 시도된 구현 사례의 특징 및 한계는 다음과 같다.

#### 1.1 레시피ML(Recipe Markup Language)

2000년경 제안된 XML 기반 포맷으로, 코드명 DESSERT(Document Encoding and Structuring Specification for Electronic Recipe Transfer)로도 불린다. XML 태그를 통해 레시피의 재료 목록과 단계별 지시사항을 마크업 함으로써, 서로 다른 단위 환산을 자동화하거나 재료를

체계적으로 표현하려는 목적이 있었다[1]. 하지만 RecipeML은 산업 표준으로 널리 채택되지는 못했고, 다양한 레시피 포맷이 혼재하는 등의 한계가 있다.

## 1.2 Corel (Cooking Recipe Language)

2021년 네덜란드 흐로닝언대(University of Groningen)의 연구로 개발된 요리 레시피 DSL이다. Rascal 언어 워크벤치를 이용하여 구현되었으며, 레시피의 구조를 도메인 분석(FODA 기법)으로 모델링하여 설계되었다. Corel은 재료에 대한 이해와 영양 성분 계산 등의 기능을 포함하고 있어 레시피로부터 영양 라벨을 자동 생성할 수 있다. 실제 요리 동작의 모든 측면을 기술하기에는 한계가 있지만, 재료, 분량, 단계 등을 형식화하여 레시피 지식을 컴퓨터가 처리할 수 있음을 보였다는 의의가 있다[2].

## 1.3 Cooklang

2020년대에 커뮤니티 주도로 만들어진 오픈소스 레시피 마크업 언어이다. 일반 텍스트로 작성된 레시피에 특수 문자를 넣어 재료, 용기, 조리 시간 등의 정보를 기계가 파싱할 수 있게 하였다. 예를 들어 2킬로그램의 감자는 '@감자{2kg}'의 형태로 표현한다. 또한, 각 줄을 단계로 구분하는 등 단순한 문법을 제공하며[3], 개발 도구와 모바일 앱 등의 생태계를 갖추어 개인 레시피 관리나 장보기 리스트 연계 등에 활용되고 있다.

항목	RecipeML	Corel	Cooklang
제안 시기	2000년경	2021년	2020년대
표현 방식	XML 마크업	Rascal 기반 DSL	특수 기호가 포함된 일반 텍스트
구현 목적	단위 변환, 재료 체계화	레시피 지식 구조화, 영양 계산	레시피 관리, 앱 연동
표현력	중간 (재료, 단계 위주)	높음 (재료, 영양소, 조리 단계 등)	낮음 (기본 정보 중심)
기계 해석 가능성	있음 (XML 기반 구조화)	높음 (도메인 모델링 기반)	제한적 (단순한 문법 구조)
활용 사례	제한적 (표준화 실패)	연구용 도구로 사용됨	모바일 앱, 도구 등 존재
한계점	보급 부족, 포맷 난립	실제 조리 절차 표현 미흡	표현력 부족

표 1. 요리 레시피 DSL 목록

## 2. 자동 조리 로봇의 연구 및 개발 동향

인공지능 및 컴퓨터 비전 기술의 발전으로 자동 조리 로봇은 비약적으로 발전했다. 이는 상용 시스템과 연구 프로토타입 모두 해당되며, 이러한 기술에는 다음과 같은 것들이 있다.

### 2.1 BakeBot

2010년대 초 MIT에서 진행한 프로젝트로 PR2 로봇을 이용하여 쿠키 반죽 만들기 등 베이킹 작업을 수행하며, 자연어 레시피를 로봇 명령으로 해석하는 실험을 하였다[4].

### 2.2 Moley Robotic Kitchen

2017년대에 개발된 대표적 사례로, 주방 전체를 하나의 로봇 셰프로 구성하여 로봇 팔이 조리 동작을 수행하도록 설계됐다. 셰프의 움직임을 그대로 모방하는 것을 목표로 하였으며, 구획된 공간에서 조리 후 자체 세척까지 수행하는 컨셉을 제시했다. 하지만 복잡한 기술과 높은 비용 때문에 일반 가정에 보급되지는 않았다[5].

### 2.3 Bot Chef

삼성 등 대기업에서 CES 2020 등에서 공개한 사례로, 두 개의 로봇 팔이 조리대에서 작업을 수행하며, 인간과 협동하여 보조적인 역할을 할 수 있는 주방 로봇 팔로 주목받았으나, 실제 상용 제품으로 출시되지는 않았다[6].

### 2.4 YORI 시스템

UCLA 연구팀 등이 제안한 모듈형 자동 주방 시스템으로, 협동 로봇(cobot)을 이용하여 조리대, 도구 배치 등을 유연하게 변경하며, 다양한 요리를 구현하는 연구가 진행됐다[7].





항목	BakeBot	Moley Robotic Kitchen	Bot Chef	YORI 시스템
				
개발 시기/ 기관	2010년대 초 / MIT	2017년경 / Moley Robotics	2020년경 / 삼성 등	2020년대 / UCLA 등
조리 범위	베이킹 중심 (반죽 등)	전체 요리 (셰프 동작 모방)	조리 보조 중심	조리 보조 중심
로봇 유형	PR2 이족형 로봇	전용 로봇 팔 2개 + 주방 통합 시스템	로봇 팔 2개 (협동형)	협동 로봇(cobot) 기반
레시피 해석 방식	자연어 레시피 → 명령 해석	사전 녹화된 셰프 동작 모방	명령 기반 조작, 비전 활용 가능	DSL 등 연구 가능성 있음
자동화 수준	반자동	완전 자동	반자동, 인간 협동형	유연한 자동화 구성 가능
상용화 여부	연구용	상용화 시도 (비싼 가격)	전시용 시연, 미출시	연구용 프로토타입
기술적 한계	인식 정확도, 기능 제한	고비용, 설치 공간 제한	제품화 미진, 실용성 부족	구성 복잡성, 시스템 통합 문제

표2. 자동 조리 시스템 목록[8, 9, 10, 11]

### III. 기대 효과

#### 1. 레시피의 표준화 및 일관된 해석 제공

제안된 요리용 코딩언어는 사람마다 다르게 해석되는 자연어 기반 레시피의 문제점을 해결하고, 레시피의 정확한 구조화와 해석 가능성을 제공한다. 이를 통해 누구나 동일한 방식으로 조리법을 이해하고 재현할 수 있게 된다.

#### 2. 자동 조리 시스템과의 통합 용이성

요리 과정을 기초 조리 동작의 조합으로 표현하여 기계 해석이 용이하도록 하고, 조리 로봇에 의한 반복 가능하고 일관된 자동 조리가 가능하도록 한다. 이를 통해 레시피의 개발과 이를 자동 조리 시스템에 적용하는 일을 한번에 수행할 수 있게 된다.

### **3. 조리 교육 및 시뮬레이션 활용 가능성**

레시피의 정형화된 표현은 조리 학습 시스템, 시뮬레이터 등에 쉽게 적용될 수 있어, 교육 도구로서의 활용 가치가 크다. 특히 언어 학습과 시뮬레이션이 연동될 경우, 실제 조리 경험 없이도 조리 과정을 습득할 수 있게 된다.

## 2.과제 구성

### 1. 과제 목표

본 과제는 조리 동작의 표현을 중점으로 한 요리 레시피 DSL을 설계하고, 이를 실제로 사용할 수 있는 환경을 구현하는 것을 목표로 한다. 이를 통해 요리 레시피 기술의 새로운 방법을 제시함으로써 다음과 같은 성과를 지향한다.

#### 1. 요리 과정의 형식화

- A. 조리 단계, 재료, 도구, 시간 등 레시피를 구성하는 요소들을 명확하게 정의할 수 있도록 한다.
- B. 재료 표현 등에서 사용되는 단위를 SI 단위계 기반으로 표준화한다.

#### 2. 요리 과정의 일관된 해석 구조 구현

- A. 재료 용량이나 조리 시간 등을 표준 단위로 표현하여 지역이나 문화 등에 따른 해석의 차이를 방지한다.
- B. 요리 과정을 기초 조리 동작의 조합으로 표현하여 모호성을 최소화하고, 일관된 해석이 가능하도록 한다.

#### 3. 인간 친화적 출력 기능 제공

- A. 코드 형태의 레시피를 사람이 쉽게 이해할 수 있도록 자연어 형식으로 출력할 수 있는 기능을 포함한다.

#### 4. 레시피 작성을 위한 편의 기능 제공

- A. 레시피 오류 검사 기능을 통해 작성된 레시피 코드의 오류 등을 쉽게 파악하고 수정할 수 있도록 한다.
- B. 필요한 도구 자동 분석 기능을 통해 작성된 레시피에서 필요한 도구를 자동으로 출력할 수 있도록 한다.



## II. 기존 요구 사항 및 수정 사항

### 1 기존 요구 사항

#### 1.1 SI 단위계로 표현된 재료 용량

기존 레시피의 '적당량', '한 줌', '티스푼' 같은 표현은 사람마다 해석이 다르며 자동화 조리에서 큰 혼란을 야기할 수 있다. 따라서 본 언어는 재료의 용량을 그램(g), 밀리리터(mL) 등 SI 단위계로 통일하여 표현한다.

#### 1.2 사람이 이해하기 쉬운 형태의 레시피 출력 기능

개발된 언어는 기계가 해석할 수 있도록 구조화되지만, 동시에 사람이 읽기 쉬운 형태로 변환 가능한 문법 구조를 제공한다. 예를 들어, 원본 코드에서 단계별 동작을 요약하거나 자연어 형식으로 출력할 수 있는 기능을 함께 제공한다.

#### 1.3 자동 요리 로봇에 적용할 수 있는 구조

언어 설계는 로봇 동작으로 변환 가능한 명령어 체계를 지니며, 조리 도구 및 동작을 명시적으로 표현할 수 있도록 한다. 이를 위해 각 조리 단계는 행동, 대상, 도구 등으로 구조화한다.

### 2 수정된 요구 사항

#### 2.1 재료 용량 및 조리 시간 표현법

기본적으로 그램(g), 밀리리터(mL) 등 SI 단위계를 사용하여 표현하지만, 예외적으로 재료를 삶을 때 필요한 물의 양이나 개인의 기호에 따라 양을 조절할 수 있는 부분 등 명확한 용량의 표기가 어려운 부분을 표현할 수 있도록 한다. 또한, 조리 과정에서 기본적으로 시간을 기준으로 조리 시간을 표현하지만, 일부 경우 '물이

끓으면'과 같이 명확한 시간의 제시가 어렵고 조건으로 제시하는 것이 나은 경우가 존재하기에 이를 별도로 표현할 수 있게 한다.

## 2.2 요리 레시피 작성 특화 기능

기존 계획의 자동 요리 로봇에 적용할 수 있는 구조는 실제 테스트를 하기 어렵다는 문제점을 지니고 있으며, 과제의 규모와 실용성을 고려할 때 졸업 과제로 진행하기 부적절하다고 판단하였다. 그 대신 요리 레시피의 기술에 초점을 맞춰 선행 기술 조사에서 있었던 기존 언어들의 문제점과 이를 개선하는 부분에 주목하였고, 이에 대한 해결책으로 레시피 오류 검사와 필요한 도구 자동 분석 등의 기능을 추가하여 주요 목적이었던 자연어 기반의 레시피 출력과 이를 보조하는 기능들에 집중하도록 수정하였다.

# III. 설계 상세화

## 1. 레시피 오류 검사

일반적인 범용 프로그래밍 언어들은 문법 오류나 오타 등을 검사하는 기능이 존재한다. 이러한 기능은 사용자가 작성한 프로그램의 문제점들을 쉽게 파악하고, 이를 수정할 수 있도록 하여 프로그램의 실행에 있어 생길 수 있는 문제들을 해결할 수 있도록 한다. 본 언어는 레시피 기술에 특화된 DSL임을 이용하여 위의 기능을 확장한 새로운 기능을 제공한다. 레시피 오류 검사는 작성된 레시피의 재료와 조리 과정을 분석하여 사용되지 않은 재료나 조리 동작의 누락 여부 등을 탐색하고, 이를 사용자가 발견해서 수정할 수 있도록 문제가 생긴 부분을 경고하는 오류 메시지를 제공한다.

## 2. 도구 자동 분석

조리 동작에는 조리 도구가 필요하며, 하나의 레시피에서 같은 조리 도구가 여러 번 반복해서 사용될 수 있다. 이는 필요한 조리 도구를 직접

기술하는 것보다 레시피의 조리 과정을 분석하여 필요한 조리 도구를 자동으로 구하는 것이 이점을 가진다는 것을 의미한다. 그래서 이러한 기능을 제공하여 별도의 작성 없이 자동으로 필요한 도구를 출력할 수 있도록 한다.

### **3. 자연어 레시피 출력**

코드 형태의 레시피는 간결성과 명확성에서 이점을 가지지만 사람이 읽고 이해하는데 어렵다는 단점을 지닌다. 이를 해결하기 위해 코드를 분석해서 사람이 쉽게 이해할 수 있는 자연어 형식으로 출력한다.

## **IV. 개발 환경**

**Docker 기반 Ubuntu 25.04 컨테이너**

## **V. 사용 기술**

1. **Make** (빌드 자동화 도구)
2. **flex** (어휘 분석기 생성기)
3. **Bison** (구문 분석기 생성기)
4. **GCC** (C/C++ 컴파일러)

## VI. 예상 구조

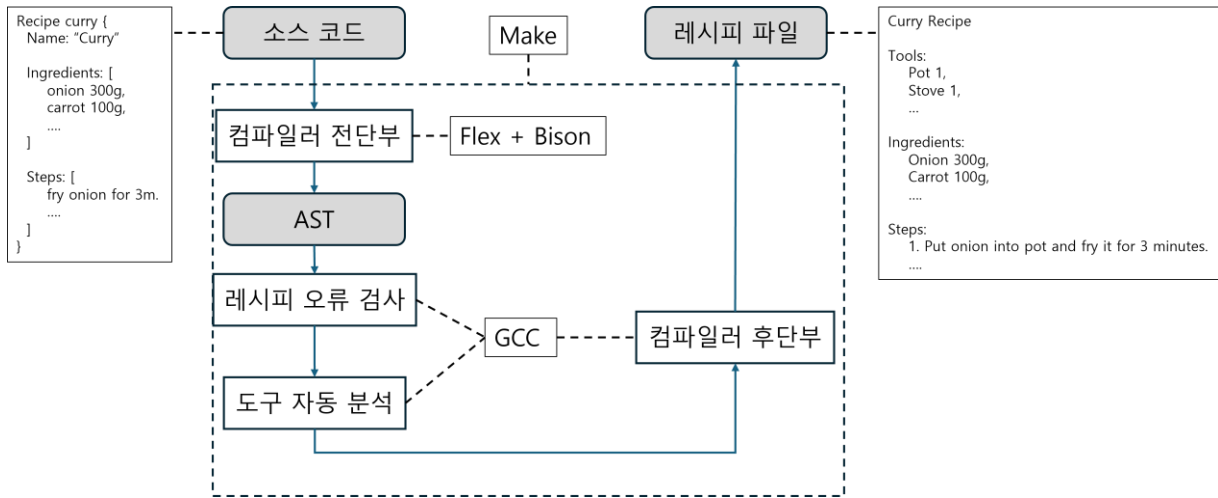


그림1. 프로그램의 전체 구조

예상 결과물	내용	형태
레시피 DSL 문법 정의서	요리 레시피를 기술할 수 있는 DSL의 문법 정의	PDF / 문서 파일
컴파일러 전단부	DSL의 소스 코드를 추상 구문 트리 (AST)로 변환하는 어휘·구문·의미 분석기 구현	C/C++ 코드 (flex, Bison)
레시피 오류 검사	레시피의 오류를 검사하고 경고하는 기능	C/C++ 코드
도구 자동 분석	레시피에서 필요한 도구를 분석하고, 이를 출력하는 기능	C/C++ 코드
컴파일러 후단부	DSL의 AST를 자연어 레시피로 변환하는 코드 생성기 구현	C/C++ 코드
테스트 레시피 파일	DSL로 작성된 예시 레시피	소스 코드 파일
자연어 출력 결과	DSL로 작성된 레시피가 자연어로 변환된 결과	텍스트 파일 / 출력 예시
문서화 및 사용자 가이드	DSL 문법 사용법, 컴파일 및 실행 방법 정리	PDF / README

표 3. 예상 결과물

### 3.수행 계획

#### I. 갱신된 개발 일정

구분	작업	6월		7월		8월		9월		10월	
		12	34	12	34	12	34	12	34	12	34
개발	언어 설계										
	컴파일러 전단부										
	레시피 오류 검사										
	도구 자동 분석										
	컴파일러 후단부										
테스트	레시피 파일 작성										
	유닛 테스트										
	통합 테스트										
마무리	보고서 작성										

표4. 갱신된 개발 일정

#### II. 구성원별 진척도

이름	담당	진척도
윤선재	언어 설계	○
	컴파일러 전단부	△
	레시피 오류 검사	X
	도구 자동 분석	X
채문석	사전 연구 조사	○
	컴파일러 후단부	△
	테스트 작업	X
팜민두웅	레시피 조사	△
	레시피 파일 작성	X

○ : 완료 | △ : 진행 중 | X : 예정

표5. 구성원별 진척도

## 4.과제 수행 내역 및 중간 결과

### 1 언어 설계

```
1  program = { recipe_definition | ingredient_definition | action_definition };
2
3  recipe_definition = 'Recipe', recipe_identifier, '{',
4                      'Name', ':', name,
5                      'Ingredients', ':', '[', ingredient, { ingredient, ',' }, ']',
6                      'Steps', ':', '[', action, { action, ',' }, ']',
7                      '}' ;
8
9  ingredient_definition = 'Ingredient', ingredient_identifier, '{',
10                         'Name', ':', name,
11                         'Unit', ':', unit,
12                         'Actions', ':', '[', action_identifier, { action_identifier, ',' }, ']',
13                         '}' ;
14
15  action_definition = 'Action', action_identifier, '{',
16                     'Name', ':', name,
17                     '}' ;
18
19  ingredient = ingredient_identifier, [ amount ] ;
20
21  amount = number, [ unit ] ;
22
23  unit = 'g' | 'mL' ;
24
25  action = action_identifier, { ingredient_identifier }, 'for', time
26          | action_identifier, { ingredient_identifier }, 'until', condition ;
27
28  time = [ number ], [ 'h' ], [ number ], [ 'm' ], [ number ], [ 's' ] ;
29
30  recipe_identifier = identifier ;
31
32  ingredient_identifier = identifier ;
33
34  action_identifier = identifier ;
35
36  identifier = string ;
37
38  name = "'", string, "'";
39
40  string = { letter | ' ' | '_' } ;
41
42  letter = 'A' | 'B' | 'C' | 'D' | 'E' | 'F' | 'G'
43          | 'H' | 'I' | 'J' | 'K' | 'L' | 'M' | 'N'
44          | 'O' | 'P' | 'Q' | 'R' | 'S' | 'T' | 'U'
45          | 'V' | 'W' | 'X' | 'Y' | 'Z' | 'a' | 'b'
46          | 'c' | 'd' | 'e' | 'f' | 'g' | 'h' | 'i'
47          | 'j' | 'k' | 'l' | 'm' | 'n' | 'o' | 'p'
48          | 'q' | 'r' | 's' | 't' | 'u' | 'v' | 'w'
49          | 'x' | 'y' | 'z' ;
50
51  number = { digit } ;
52
53  digit = '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9' ;
```

그림2. ISO-14977 EBNF로 표현한 언어의 EBNF

## 5.진행 상황

### I. 잘된 부분

기존 언어 설계는 선행 연구와 크게 다르지 않은 구조로 이루어져 차별점을 내세우기 어려웠고, 기능에 비해 구현이 지나치게 복잡하여 실질적인 성과를 내기 어려웠다. 이를 개선하여 언어의 핵심 기능과 선행 연구와의 차별점에 집중하였고, 이것이 성공적으로 진행되어 확실한 장점을 지니게 되었다.

### II. 계획대로 되지 않은 부분

기존에 설계한 언어가 컴파일러 구현에 있어 문제점을 가지고 있었고, 이를 해결하는 동시에 언어의 설계를 수정하는 과정에서 예상보다 많은 시간이 소요되었다. 이로 인해 예정보다 계획이 지연되었다. 또한 예상보다 촉박한 일정과 구성원들의 참여도 부족으로 프로젝트의 방향성을 잡는 부분에서 어려움이 있었다.

## 6.참고 문헌

- [1] FORMATDATA. RecipeML. Available at: <http://www.formatdata.com/recipe/ml/>.
- [2] ROORDA, Auke. Corel: A DSL for Cooking Recipes. 2021. PhD Thesis.
- [3] COOKLANG. Cooklang Specification. Available at: <https://cooklang.org/docs/spec/>.
- [4] Bollini, M., Tellex, S., Thompson, T., Roy, N., & Rus, D. (2013). Interpreting and executing recipes with a cooking robot. In J. Desai, G. Dudek, O. Khatib, & V. Kumar (Eds.), *Experimental Robotics* (pp. 481–495). Springer International Publishing.
- [5] DESIGNBOOM. Moley robotic kitchen: world's first automated kitchen prepares gourmet meals. Available at: <https://www.designboom.com/technology/moley-robotic-kitchen-chef-12-07-2015/>.
- [6] SAMSUNG NEWSROOM. Get a glimpse of the next-generation innovations on display at Samsung's technology showcase. Available at: <https://news.samsung.com/global/get-a-glimpse-of-the-next-generation-innovations-on-display-at-samsungs-technology-showcase>.
- [7] Noh, D., Nam, H., Gillespie, K., Liu, Y., & Hong, D. (2024). YORI: Autonomous Cooking System Utilizing a Modular Robotic Kitchen and a Dual-Arm Proprioceptive Manipulator.
- [8] <https://danielarus.csail.mit.edu/index.php/2015/10/robots-and-cooking-bakebot/>
- [9] <https://robbreport.com/gear/electronics/moley-robotics-robot-kitchen-uk-for-sale-1234590791/>
- [10] <https://news.samsung.com/global/get-a-glimpse-of-the-next-generation-innovations-on-display-at-samsungs-technology-showcase>
- [11] <https://spectrum.ieee.org/romela-cooking-robot>