

# Introduction to Airline (가제)

조승현

November 22, 2014

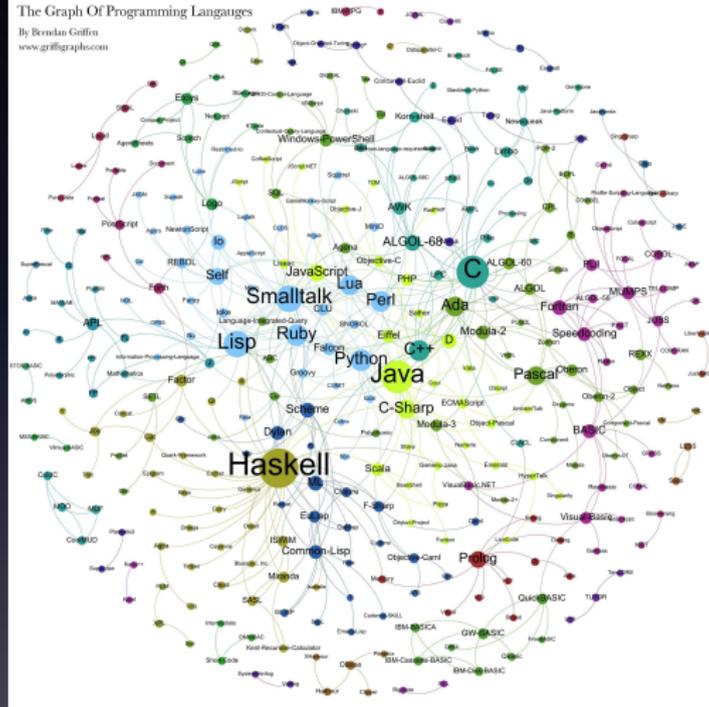
# 프로그래밍 언어

...What we see everyday

# 프로그래밍 언어

The Graph Of Programming Languages

By Brendan Griffin  
www.griffgraphs.com



# 프로그래밍 언어의 역사

- 최초의 프로그램 (1843)
- 프로그래밍 언어에 대한 연구 시작 (1930s)
- 튜링 머신 (1936)
- 폰 노이만 구조 (1945)
- ...

# 기타 등등

- Syntax & Semantics
- Turing Completeness
- 절차형 언어, 객체지향 언어, 함수형 언어, 난해한 언어 ...

# 오늘의 주제



# 오늘의 주제



# 오늘의 주제

00:00 IB103 To Madrid  
00:30 IB107 To Frankfurt  
01:00 LH113 From Madrid  
01:30 LH115 From Frankfurt  
01:40 AF332 To Seoul  
02:00 OZ501 From Seoul

# Airline (가제)

- 비행 시각표를 소재로 한 난해한 프로그래밍 언어
- 현재 버전 0.1, 정수 입출력 및 연산, 분기가 가능

# 문법

- 한 문장의 구성: 시각, 편명, 출발지 또는 목적지
- 시각: 각 문장의 위치를 나타내는 값
- 편명: 연산의 종류를 나타내는 값, 첫 글자가 연산의 종류를 결정
- 출발지 또는 목적지: 값을 저장하거나 불러올 위치

# 연산의 종류

- Input

- ▶ 입력
- ▶ 편명이 홀수이면 정수, 짝수이면 문자를 입력받음
- ▶ IB103 To Madrid: 정수를 입력받아 Madrid에 저장

- Output

- ▶ 출력
- ▶ 편명이 홀수이면 정수, 짝수이면 숫자에 해당하는 문자를 출력
- ▶ 0Z501 From Seoul: Seoul에 저장된 정수를 출력

# 연산의 종류

- Load
  - ▶ 연산을 위해 저장된 값을 스택으로 불러옴
  - ▶ LH113 From Madrid: Madrid에 저장된 값을 스택으로 불러옴
- Store
  - ▶ 스택 최상위값을 저장
  - ▶ SQ305 To Singapore: 스택의 값을 Singapore에 저장

# 연산의 종류

- Add, Divide, Multiply
  - ▶ 스택의 상위 두 값의 합/몫/곱을 저장
  - ▶ AF332 To Seoul: 스택의 상위 두 값의 합을 Seoul에 저장
- Negate
  - ▶ 스택 최상위값의 음수를 저장
  - ▶ NZ333 To Auckland: 스택 최상위값의 음수를 Auckland에 저장

# 연산의 종류

- Zeroize

- ▶ 저장된 값을 0으로 초기화
- ▶ ZE101 To Jeju: Jeju에 0을 저장

- Unit

- ▶ 저장된 값을 1로 초기화
- ▶ UA154 To Honolulu: Honolulu에 1을 저장

# 연산의 종류

- Branch
  - ▶ 스택 최상위값이 0일 경우 이동
  - ▶ BA130 From London: 스택 최상위값이 0일 경우 01:30 문장으로 이동
- Compare
  - ▶ 스택 최상위값이 두 번째 값보다 크면 1, 작으면 -1, 같으면 0을 저장
  - ▶ CA105 To Shanghai: 비교 결과를 Shanghai에 저장

# 예제 프로그램

00:00 IB103 To Madrid

정수를 입력받아 Madrid에 저장

00:30 IB107 To Frankfurt

정수를 입력받아 Frankfurt에 저장

01:00 LH113 From Madrid

Madrid에 저장된 값을 스택으로 가져옴

01:30 LH115 From Frankfurt

Frankfurt에 저장된 값을 스택으로 가져옴

01:40 AF332 To Seoul

스택에서 두 값을 꺼내 합을 Seoul에 저장

02:00 OZ501 From Seoul

Seoul의 값을 정수로 출력

# 생각할 점

- Turing-complete?
  - ▶ 조건 분기문이 있고, 메모리의 임의 위치의 값을 바꾸는 기능이 있으면 튜링 완전함이 알려져 있음
  - ▶ 문장의 조합을 통해 임의 위치에 임의의 값을 할당할 수 있고, 조건에 따라 이동하는 문장이 설계되어 있으므로 튜링 완전
- Syntactic Sugar
  - ▶ 소개한 연산만으로 모든 프로그램을 구현하는 것이 가능함
  - ▶ 하지만 프로그램 작성의 효율을 위해 적절한 설탕 구조가 있으면 좋을 듯

# 계획

- 적절한 이름을 정한다.
- 문법을 완성한다.
- 컴파일러 혹은 그에 상응하는 프로그램을 만든다.
- 알려진 문제에 대한 여러 가지 예제 코드를 만든다.
- Google Code Jam 2015에 이를 이용한 코드를 제출하여 언어로 공인받는다.(!)
- 일단 디펜스를 해야 하니 나중에..

끝

질문?