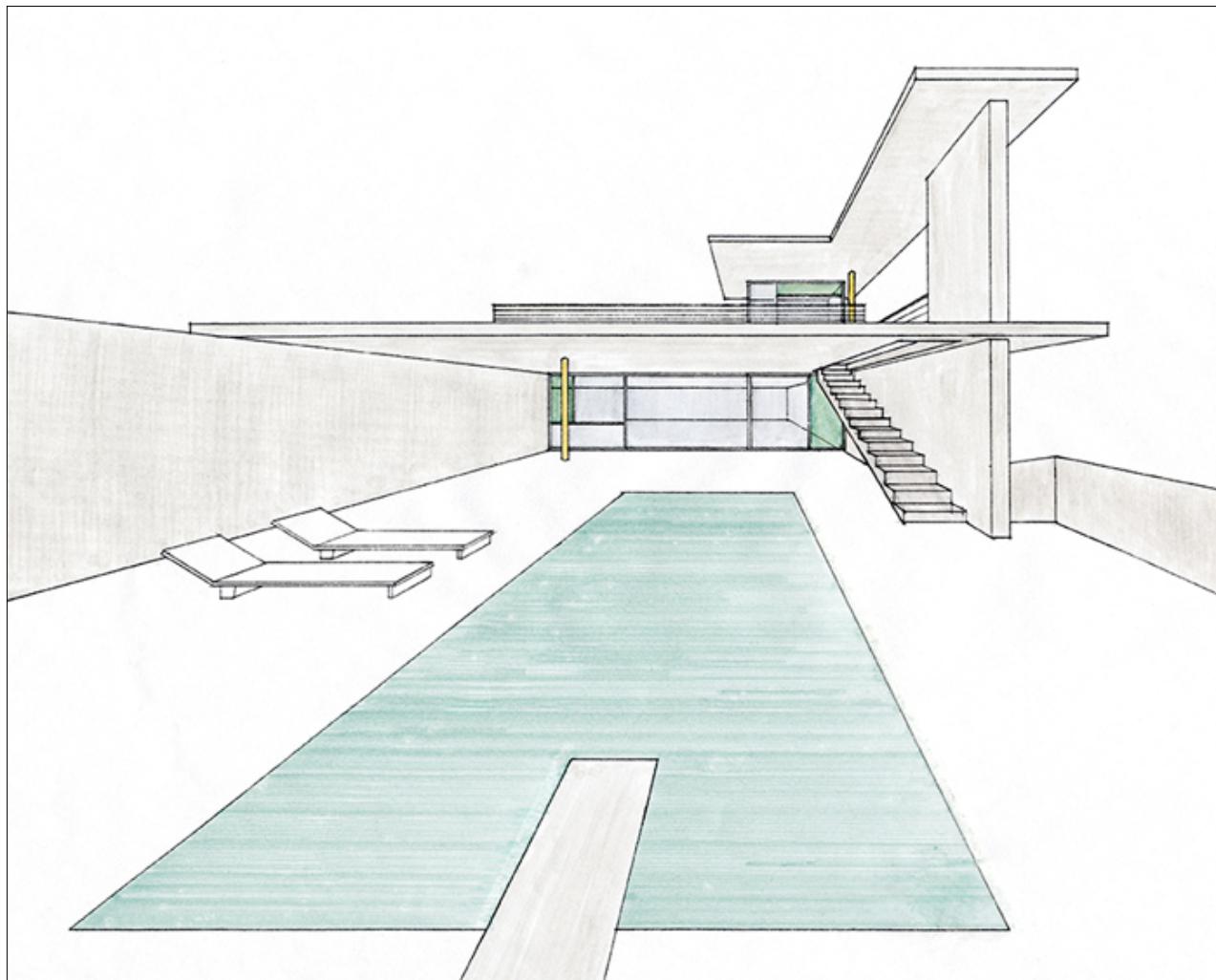


---

## GYOSHIL PROGRAM



## 교실(Gyoshil) 프로그램 소개서

- 교실 웹사이트 주소 : [www.gyoshil.com](http://www.gyoshil.com)
  - 담당 교수 프로필 주소 : <https://www.marksantolucito.com/>
  - 프로그램 설명회 신청 링크 : <https://forms.gle/5eCTqBrvA6UeUeeb6>
  - 카카오톡 문의 채널 : <https://open.kakao.com/o/sxIffYfg>
  - 한국 매니저 연락처 : 010 - 4964 - 2334
-

## GYOSHIL PROGRAM

### 교실 프로그램 소개 및 목적

교실 프로그램은 미국 상위권 대학/대학원/보딩스쿨 유학 준비생을 위한 교육 프로그램으로, **6주 동안의 zoom 이론 강의와 학술지 작성 실습 프로젝트**로 구성되어 있습니다.

현재 아이비리그 콜롬비아 대학교(예일대 박사)에서 컴퓨터 공학 강의를 하고 있는 마크 교수가 직접 운영하며, 3명의 아이비리그 출신 졸업생들이 조교로 참여합니다. 최대 6명의 학생이 참여하는 프라이빗한 인공지능 교육 프로그램이고, 이론 수업을 바탕으로 학생들이 팀을 이뤄 학술지를 작성하여 세계적인 컨퍼런스에 제출하는 것을 목표로 합니다.

프로그램 과정에서 컴퓨터 공학 지식, 영어, 학술지 작성 능력 향상은 물론, 현 아이비리그 교수 및 조교들과의 네트워크를 구축하여 미국 유학을 위한 실질적인 다양한 도움을 받을 수 있습니다.

### 교실 프로그램 교수 및 조교 소개

#### 담당 교수 : **Mark Santolucito (마크 교수님)**

현 컬럼비아 대학교 컴퓨터 공학과 조교수  
Assistant Professor of Computer Science at  
Barnard College, Columbia University

예일대학교 컴퓨터 공학과 박사  
Phd in Computer Science working on  
Program synthesis and Computer Music at Yale

자세한 교수 소개 및 연구 관련 자료는  
<https://www.marksantolucito.com/> 에서 확인 가능합니다.



Raven Rothkopf (조교)

컬럼비아 졸업,  
UCSD 컴퓨터 공학 박사 과정 (예정)



Leo Qiao (조교)

컬럼비아 졸업,  
컬럼비아 석사 재학중



Hayeon Kim (조교)

컬럼비아 졸업

## 교실 프로그램이 제공할 수 있는 가치

### 1. 현 아이비리그 교수의 실제 영어 인공지능 강의

: 교실 프로그램은 유학 준비생을 위한 프로그램으로서, 대학/대학원/보딩스쿨 수업을 수강 준비가 된 학생이라는 전제가 있기 때문에, 모든 강의는 영어 강의로 진행됩니다. (한국어가 가능한 조교도 함께할 계획입니다)

### 2. 소규모 프라이빗 강의

: 프로그램별 최대 인원은 6명입니다. 참여 학생 모두가 학술지(리서치 페이퍼) 작업을 마칠 수 있도록, 강의를 진행하는 마크 교수님 뿐만 아니라, 아이비리그 출신 조교 2명과 한국인 조교 1명이 적극적으로 보조하게 됩니다.

### 3. 학술지 작성 및 세계적인 컨퍼런스 제출 (with. 아이비리그 교수 및 조교)

: 강의 수강 기간 동안 교수와 학생들이 함께 팀을 이뤄, 학술지를 작성하고 세계적인 컨퍼런스(ex. AAAI, ICMl, ICLR, NeurIPS, LCAI) 중 하나의 컨퍼런스에 제출하는 것을 목표로 합니다. 학술지 등재는 심사위원의 둘이기 때문에 등재를 보장할 수 없으나, 학술지 작성 경험은 아이비리그 입학에 긍정적인 영향을 미칩니다. 실제 2023년 펜실베니아 대학 입학생 중 3분의 1 이상은 연구 논문 작성 경험이 있습니다.

### 4. 아이비리그 교수, 학생과의 네트워크 형성 (Lifelong Network)

: 프로그램을 성실히 참여한다면, 교수, 조교, 학생 사이에는 네트워크가 형성됩니다. 이러한 네트워크는 미국 대학 입학을 위한 조언으로 이어지고, 다양한 실질적인 도움을 제공할 수 있는 자산이 됩니다. 실질적인 도움에 대한 세부 내용은 설명회에서 공유드릴 예정입니다.

## 마크 교수가 고등학생과 함께 작성한 학술지 사례 소개

마크 교수는 고등학생들과 함께 학술지를 작성하고, 세계적인 컨퍼런스인 AAAI(The Association for the Advancement of Artificial Intelligence)에 제출한 바 있습니다. 참여 팀원 중 고등학생은 **Kairo Morton**과 **Elven Shum**이 있었고,

**Kairo Morton** 학생은 MIT 대학에, **Elven Shum** 학생은 Yale 대학에 진학했습니다.

### Grammar Filtering For Syntax-Guided Synthesis

Kairo Morton,<sup>1</sup> William Hallahan,<sup>2</sup> Elven Shum,<sup>3</sup> Ruzica Piskac,<sup>2</sup> Mark Santolucito,<sup>2</sup>  
<sup>1</sup>George School, <sup>2</sup>Yale University, <sup>3</sup>Deerfield Academy  
mortonk@georgeschool.org, william.hallahan@yale.edu, eshum20@deerfield.edu, ruzica.piskac@yale.edu, mark.santolucito@yale.edu

#### Abstract

Programming-by-example (PBE) is a synthesis paradigm that allows users to generate functions by simply providing input-output examples. While a promising interaction paradigm, synthesis is still too slow for real-time interaction and more widespread adoption. Existing approaches to PBE synthesis have used automated reasoning tools, such as SMT solvers, as well as works applying machine learning techniques. At first, automated reasoning approach had highly domain specific knowledge of programming languages. On the other hand, the machine learning approaches utilize the fact that when working with program code, it is possible to generate arbitrarily large training datasets. In this work, we propose a system for using machine learning in tandem with automated reasoning techniques to solve Syntax Guided Synthesis (SyGuS) style PBE problems. By preprocessing SyGuS PBE problems with a neural network, we can use a data driven approach to reduce the size of the search space, then allow automated reasoning-based solvers to more quickly find a solution analytically. Our system is able to run atop existing SyGuS PBE synthesis tools, decreasing the rate of the winner of the 2019 SyGuS Competition for the PBE Strings track by 47.65% to outperform all of the competing tools.

#### Introduction

The term “program synthesis” refers to automatically generating code to satisfy some specification. That specification

The idea of automated code synthesis is an area of research with a long history (cf. the Church synthesis problem (Church 1963)). However, due to the problem’s undecidability and high computational complexity for decidable fragments, for almost 50 years the research in program synthesis was mainly focused on addressing theoretical questions and the state of synthesized programs was relatively small. However, the state of affairs has drastically changed in the last decade. By leveraging advances in automated reasoning and formal methods, there has been a renewed interest in software synthesis. The research in program synthesis has recently focused on developing efficient algorithms and tools, and synthesis has even been used in industrial software (Gulwani 2011). Today, machine learning plays a vital role in modern software synthesis and there are numerous tools and startups that rely on machine learning and big data to automatically generate code (cod 2019; Balog et al. 2017).

With numerous synthesis tools and formats being developed, it was difficult to empirically evaluate and compare existing synthesis tools. The Syntax Guided Synthesis (SyGuS) format language (Alur et al. 2013; Raghothaman and Updya 2019) was introduced in an effort to standardize the specification format of program synthesis, including PBE synthesis problems. The SyGuS language specifies synthesis problems through two components – a set of constraints (eg

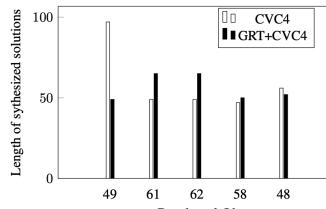


Figure 3: When the GRT+CVC4 found a different solution than CVC4, it was on average shorter than the solution found with the full grammar.

these cases there was little room for improvement even with GRT+CVC4.

Figure 4 shows the exact running times with both the full and reduced grammars from the benchmarks with the 30 largest running times with the full grammar. These are the benchmarks for which the synthesis times and size of the solution diverge most meaningfully, however all other data is available in the supplementary material for this paper. Figure 4 also shows  $|P|$  and  $|P^*|$ , the sizes of the programs found by the CVC4 and GRT+CVC4, respectively. We define size of a program as the number of nodes in the abstract syntax tree of the program. In terms of the grammar  $G$ , this is the number of terminals (including duplicates) that were composed to create the program.

In Figure 2, we present a visual comparison of the results for the 20 functions that took CVC4 the longest, while still

id	file	$T_{CVC4}^S$	$T_{GRT+CVC4}^S$	$ P $	$ P^* $
34	lastname-small.sl	1.80	1.84	4	4
35	bikes-long.sl	1.97	1.76	3	3
36	bikes-long-repeat.sl	2.08	1.71	3	3
37	lastname.sl	2.31	1.83	4	4
38	phone-6-short.sl	3.23	1.22	11	11
39	phone-7-short.sl	3.26	1.26	11	11
40	initials-long-repeat.sl	3.33	2.54	7	7
41	phone-5-short.sl	3.72	1.51	9	9
42	phone-7.sl	4.57	2.03	11	11
43	phone-8.sl	4.72	2.17	11	11
44	phone-6.sl	4.85	1.97	11	11
45	phone-5.sl	4.88	2.20	11	11
46	phone-9-short.sl	4.88	4.73	52	52
47	phone-10-short.sl	8.81	8.28	49	49
48	phone-9.sl	12.08	4.86	56	52
49	phone-10.sl	31.23	8.49	97	49
50	lastname-long.sl	32.40	25.49	4	4
51	lastname-long-repeat.sl	32.49	24.92	4	4
52	phone-6-long-repeat.sl	83.59	25.31	11	11
53	phone-5-long-repeat.sl	84.77	33.68	11	11
54	phone-7-long.sl	87.83	26.15	11	11
55	phone-7-long-repeat.sl	89.13	26.23	11	11
56	phone-5-long.sl	90.81	30.01	11	11
57	phone-8-long-repeat.sl	91.04	35.64	11	11
58	phone-9-long-repeat.sl	91.19	77.02	47	50
59	phone-6-long.sl	98.15	24.75	11	11
60	phone-8-long.sl	108.06	29.94	11	11
61	phone-10-long-repeat.sl	149.53	129.43	49	65
62	phone-10-long.sl	153.32	133.22	49	65
63	initials-long.sl	TO	TO	-	-
64	phone-9-long.sl	TO	3516.21	-	49

Figure 4: Synthesis results over the 30 longest running benchmarks from SyGuS Competition’s PBE Strings track.

## 고등학생 함께 작성한 실제 학술지

### 주제 : Grammar Filtering For Syntax-Guided Synthesis

(<https://www.marksantolucito.com/papers/aaai2020.pdf>)

## 프로그램 개요

### 1. 프로그램 대상 및 설명

: 올 여름 진행되는 첫번째 교실 프로그램은 **컴퓨터에 대한 배경지식이 없어도, 누구나 들을 수 있는 교양 과정입니다.** AI(인공지능) 기술에 대한 포괄적인 개념에서 시작해, AI 기술을 활용한 김소월 시인의 시, 한복 디자인, 가야금 음악 만들기를 함께 만들고, 수업 내용을 바탕으로 AI 기술과 관련된 리서치 페이퍼를 작성하게 됩니다.

강의는 **매주 2회(1회 수업 1.5h), 총 12회차로** 구성되어 있고, 이와 별개로 학술지 작성을 위한 학생과 조교 간의 **1:1 멘토링은 30분 씩 4회, 총 2시간**이 진행될 예정입니다.

프로그램	프로그램 주제	참여 대상	기간
교양 과정	AI 기술을 적용한 한국 문화 표현 (김소월 시, 한복 디자인, 가야금 음악)	비전공자, 누구나	7/16 - 8/22 - 주 2회 수업 (1회 수업 1.5h) - 1:1 조교 멘토링 (2h)

### 2. 프로그램 세부 커리큘럼

회차	주요 강의 내용	비고
1	Orientation : 프로그램 세부 일정, 리서치 페이퍼 주제 공유	
2-3	Fairness and Bias : AI 기술의 정당성과 편향성 (인공지능 알고리즘 설계를 하는 과정에서 생기는 편향성 문제들)  - Cross cultural limitations : 문화적 차이에서 오는 한계 - Related work : 주제와 관련된 내용 실습	
	LLM (text) : 거대언어모델 (ex. Chat GPT, LLAMA)	
4-5	- overview of how they work etc : 거대언어모델의 작동 방법 교육 - 김소월 poetry : 김소월 시인의 시를 이용한 실습	
6-7	Image Generation : AI 기술을 활용한 이미지 창작  - fashion and AI : 패션과 AI 기술의 접목 - hanbok generation : AI 기술을 활용한 한복 디자인 실습	뉴욕 패션대학교 교수 특강 (Fashion Institute of Technology, FIT)
8-9	Audio : 음악과 AI 기술 접목  - MusicGen from Meta : 메타(페이스북)에서 제공하는 뮤직젠 교육 - gayageum music : AI 기술을 활용한 가야금 음악 만들기 실습	
10	How to write a research paper : 리서치 페이퍼 작성 교육	조교와 함께 진행
11	Non Neural Net Generative AI : 인공 신경망을 사용하지 않은 AI 기술에 대한 학습, AI의 역사	
12	Presentation day	주제는 1회차 수업에서 논의할 예정

### 3. 프로그램 가격 : 4,500달러 (5월 30일까지 등록 시 얼리버드 가격, 이후에는 5,000달러로 변경)

## **프로그램 신청방법 및 신청기간**

### **1. 프로그램 신청방법**

: 교실 프로그램 참여 신청은 교실 홈페이지 ([www.gyoshil.com](http://www.gyoshil.com))에서 가능합니다.

선착순 선발이 아닌, 접수한 신청서를 전체 검토한 후, 6월 7일에 최종 선발자를 일괄적으로 발표할 예정입니다.

### **2. 프로그램 신청일정**

1) 교실 프로그램 설명회 접수 : ~ 5/10 (설문폼 접수를 통해 참여 신청 가능합니다)

2) 교실 프로그램 설명회 : 5/11 토요일 오후 3시 30분 (설명회는 비대면 줌 강의로 진행됩니다)

3) 프로그램 신청서 접수 마감 : 5/31 금요일 자정

4) 프로그램 최종 선발자 발표 : 6/7 금요일 오후 2시

이후, 프로그램 참가 비용 입금 확인 후, 수강생들을 위한 단톡방으로 초대될 예정입니다.

## **프로그램 관련 문의**

: 문의는 카카오톡(<https://open.kakao.com/o/sxlffYfg>) 또는 담당자 연락(010-4964-2334)을 통해 가능합니다.

: 이외, 마크 교수님과 비대면 1:1 미팅이 필요한 경우, 일정 협의 후 유료(15분 단위)로 진행이 가능합니다.

단, 프로그램 선발이 확정되면, 1:1 미팅에 납부한 비용은 100% 환불됩니다.

