

프로그램 합성 기법을 이용한 효율적인 능동 모델 학습 방법

소프트웨어재난연구센터 2025 겨울 정기 워크샵
김요엘, 최윤자

능동 모델 학습 (Active Model Learning)

Learner

Execution trace

Speed	Pedal	Break
0	true	false
5	true	false
10	false	true
5	false	false
0	.	.

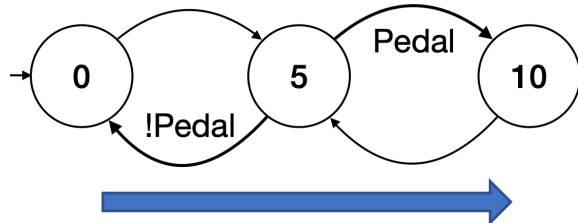
Predicate synthesis

5 → 10: (Pedal == true)
5 → 0 : (Pedal == false)

←
Missing/wrong behavior

Speed	Pedal	Break
10	true	false
20	.	.

Learned model



Teacher

Reactive program

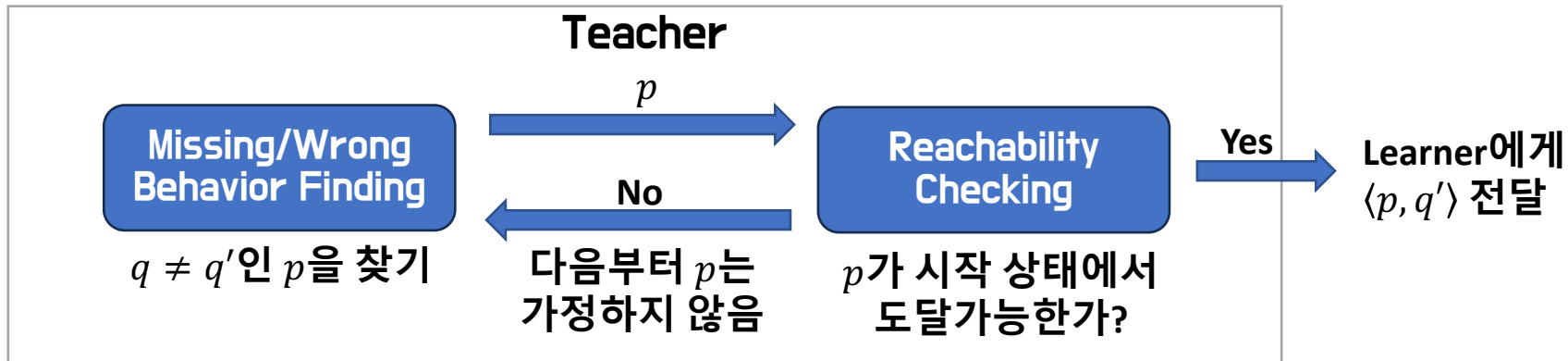
```
while (true) {  
  ...  
  ...  
}
```

Model verification

- Speed 10에서 항상 Speed 5로 감소하는가?
- Speed 5에서 Pedal을 밟으면 정말로 Speed 10이 되는가?

Teaching이 어려운 이유

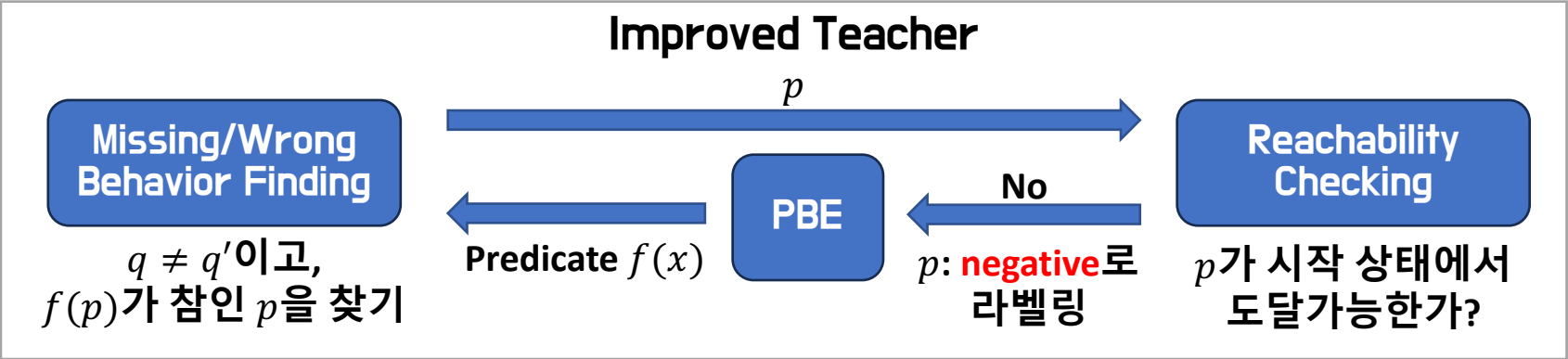
- 임의의 프로그램 상태 p 을 가정:
 - q : 학습된 모델에서 상태 p 의 다음 상태.
 - q' : 프로그램에서 상태 p 가정하에 loop iteration을 한 번 수행한 후의 상태.



- EX) 엘리베이터 컨트롤러 프로그램:
 - 주 상태: INIT, IDLE, RUN, STOP, EMERGENCY
 - 상태 변수: floor (1~4층), timer (0~9), direction (up, down, stay), button[12]
 - 모든 가능한 프로그램 상태 조합: $5 \times 4 \times 10 \times 3 \times 2^{12} = 2,457,600$ 개

해결책: 도달 불가능한 상태를 일반화하기

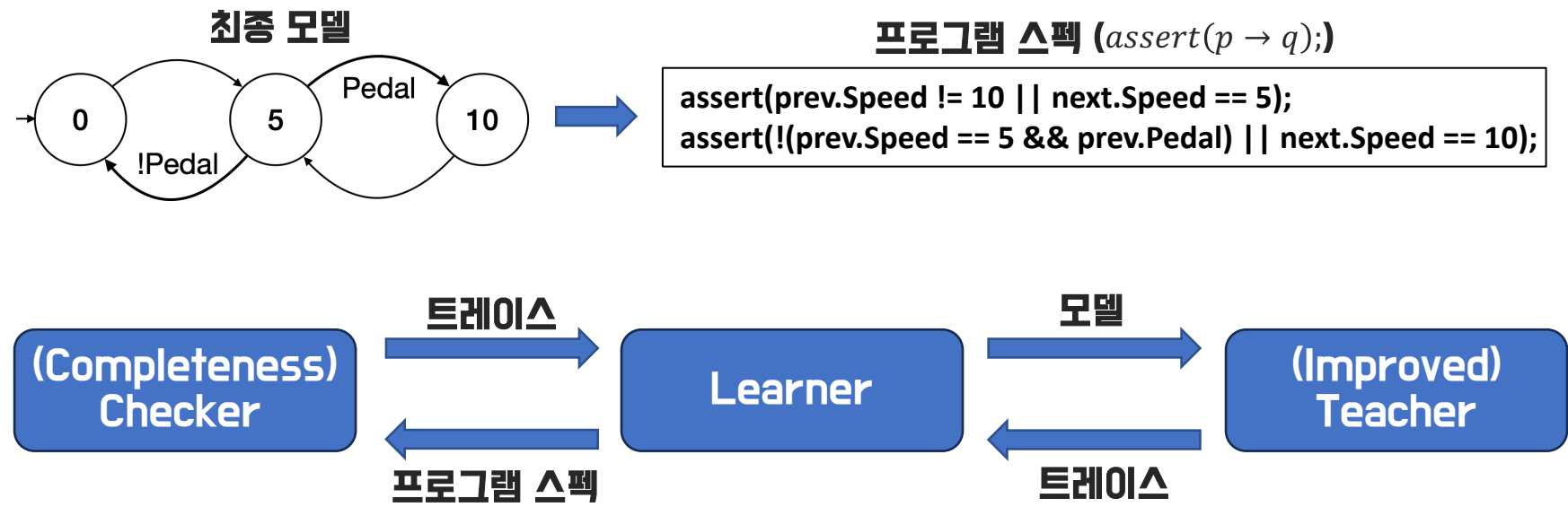
- 도달 가능한 상태는 **positive**로, 도달 불가능한 상태는 **negative**로 라벨링하여 프로그램 합성기 (PBE)에게 전달하여 predicate 합성.
- 합성된 predicate를 만족하는 프로그램 상태만 선택하도록 유도.



- 엘리베이터 컨트롤러 프로그램 적용 결과:
 - 기존 방식: 24시간이 경과해도 learner에게 아무 것도 전달하지 못함 (첫번째 모델에서 중단).
 - 해결책 적용 방식: 약 1시간 만에 5번째 모델까지 학습 성공.

해결책의 단점

- 전체 프로그램 상태 중 일부만 확인하기 때문에, 최종 모델이 완전하지 않을 수 있음.
- 따라서, 최종 모델을 프로그램 스펙 (specification)으로 변환하고, 해당 스펙을 위반하는 반례를 찾아 학습을 재개하는 방식을 적용해야 함.



	Improved Teacher		Existing Teacher	
Benchmark	Time (s)	# of unreachable states	Time (s)	# of unreachable states
WasOn	21.4	3	19792.7	9141
InDiscPresent	2513.4	118	10696.9	3354
Intersection	13.3	2	156.0	280
ATM	182.2	62	196.5	213
GreenLight	13.6	2	97.1	200
Overall	431.6	25	262.0	152
MovingAvg	49.5	11	219.8	91
AbortLogic	13.5	1	76.8	66
Simulink	21.1	2	40.0	33
Redundant	47.2	13	62.4	25
InMotion	235.1	5	137.5	7
DelayLine	18.2	3	23.1	5

도달 불가능한 상태 가정을 평균 1130.5회에서 20.6회로 줄임으로써,
약 8.9배 더 빠른 모델 학습 달성

결론

- 프로그램 합성 기법을 이용하여 teacher가 고려해야 하는 프로그램 상태의 수를 상당량 줄임으로써, 능동 모델 학습의 성능을 높일 수 있음.
- 그러나, 제안한 기법의 단점으로 인해 오히려 학습 시간이 증가하는 경우가 존재함. 또한, 도달 불가능한 상태의 일반화 (PBE)의 오버헤드 역시 무시할 수 없음.