**AI\_Assignment03 Report**

**1분반 20191244 김현승**

1. **standardize\_variables**

standardize\_variables 함수는 rule의 antecedents 또는 consequent에 있는 표준화되지 않은 변수(“something” 또는 “someone”)를 고유한 변수명으로 치환해주는 함수이다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

이 함수는 원본 데이터 파일의 rule의 내용을 수정하면 안 되므로 copy 모듈의 deepcopy 함수를 이용해 복사본을 생성한다. 만들어진 변수 명을 저장할 variables 리스트를 초기화하고, 고유한 변수 명 뒤에 숫자를 지정해주기 위해 numVar 변수를 선언한다.

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

각 rule에 대해 for문을 수행하는데, rule의 구조가 명제의 리스트로 구성된 antecedents와 한 개의 명제로 이루어진 consequent로 구성되어 있으므로 각 antecedents의 명제, consequent의 명제에 대해 반복문을 돌며 “something”이나 “someone”이 포함되어 있는지 찾는다. 만약 포함되어 있다면 해당 인덱스에 f-string을 이용해 문자열을 포맷팅하여 순서대로 x0000, x0001과 같은 변수명을 부여한다. 또, flag를 설정해주어 반복문이 끝나기 전 flag가 True인 경우 다음 변수명 뒤 숫자를 조정해준다.

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

이후 생성된 변수의 수만큼 반복문을 돌며 사용된 변수명을 variables 리스트에 추가한 후 standardized\_rules와 variables를 리턴한다.

1. **unify**

unify 함수는 query와 datum에 대해 적절한 변수 치환으로 두 명제를 동일하게 만들어주는 함수이다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

우선, 인자로 받은 query와 datum의 내용이 수정되면 안 되므로 deepcopy 함수를 통해 각각의 복사본을 생성한다. 그리고 치환한 변수의 내용을 저장할 subs 딕셔너리를 초기화한다. 만약 unify할 두 명제의 길이가 다르면 unify를 할 수 없으므로 (None, None)을 리턴하고, 만약 두 명제의 참/거짓이 다른 경우도 불가하므로 (None, None)을 리턴한다.

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

unify\_flag가 False가 될 때까지 while문을 돌며 query와 datum의 각 literal들을 비교하는 작업을 한다. 만약 어느 literal이 다른데 둘 다 상수인 경우 unify가 불가한 경우이므로 (None, None)을 리턴한다. 서로 다른 두 리터럴이 하나가 변수, 하나는 상수인 경우 unify\_flag를 True로 설정하고 최대한 query를 datum에 맞추어 변수를 변경한다. 만약 query 쪽이 변수, datum 쪽이 상수인 경우에는 datum을 query에 맞추게 된다. 만약 for문 내의 모든 조건문을 통과하지 못 한 경우 더 이상 unify할 부분이 없다는 뜻이므로 while문을 탈출한다.

1. **apply**

apply 함수는 여러 goal로 이루어진 goal set을 특정한 rule에 적용하는 함수이다. goal이 rule의 consequent와 unify 될 수 있으면 unify한 후 해당 goal을 unify된 rule의 antecedents로 치환해 새로운 goal set을 만들게 된다. 먼저 apply 함수를 구현하기 위해 명제의 변수를 치환하는 substitution 함수를 구현했다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

unify 함수에서 얻은 subs를 proposition에 적용하는 함수이다. deepcopy를 이용해 원본 명제의 수정을 방지했고, subs 딕셔너리의 key에 포함된 변수를 찾은 경우 치환하게 된다.

텍스트, 폰트, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명

apply 함수도 마찬가지로 deepcopy를 사용해 원본의 수정을 방지한다. application과 goalsets를 초기화 한 후 각 application의 index를 나타내어줄 변수 i를 선언했다.

텍스트, 스크린샷, 디스플레이, 폰트이(가) 표시된 사진

자동 생성된 설명

goals의 각 goal에 대해 rule의 consequent와 unify 할 수 있는지를 확인한다. 만약 unify가 된다면, application 리스트에 새로운 딕셔너리를 추가해 consequent를 unification의 결과로 설정한다. (goal과 동일한 상태일 것이다.) 다음, goals의 복사본을 goalsets에 추가하고 현재 적용된 goal을 제거한다. 이후 substitution 함수를 이용해서 unification에서 적용된 치환을 사용해 rule의 antecedent를 치환한 후 제거된 goal 대신에 goalsets에 추가하게 된다. 마지막으로 반복문이 돌기 전 index를 하나 추가해주었다.

1. **backward\_chain**

backward\_chain 함수는 주어진 query를 rule에 적용해 rule의 antecedent를 다시 증명하는 것을 반복함으로써 명제를 증명하는 함수이다.

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명

먼저, backward\_chain 함수를 구현하기 위해 스택을 사용하였다. 스택에는 goal이 저장되고, 매 반복마다 스택에서 goal을 꺼내 rule에 적용시킨다. 적용시킨 rule의 antecedent를 다시 goal 스택에 추가하고, 이를 반복하면 적용될 rule을 찾지 못하거나, goal 스택이 비게 될 것이다. 적용 가능한 rule을 찾지 못한 경우 해당 목표를 증명할 수 없으므로(CWA) None을 리턴하고, goal 스택이 빌 때까지 계속 적용을 마쳤다면 성공적으로 증명이 완료된 것이므로 proof를 리턴한다. 매번 rule을 적용할 때마다 그 rule을 proof에 추가한다.

1. **실행 결과**

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

코드를 실행했을 때, 주어진 샘플 데이터의 세 world에 대해 모두 올바른 정답을 도출한다.