

주의사항

- TW Exercise Files 폴더를 다른 곳에 복사해서 저장할 것.
 - 예: 바탕화면

1. Tarski's World 사용법 및 단순명제

Tarski's World 사용법 1

- You try it (p. 24)
 - Tarski's World 첫 실행
 - 단순명제 해석 방법 소개
1. Wittgenstein's World와 Wittgenstein's Sentences 실행
 2. 단순명제들의 진리값 확인
 - 우측 상단에 위치한 T/F 버튼 이용
 - 월드 부분을 수정한 후 진리값의 변화 확인
 - 예: Adjoins 명제가 모두 False임. 일부 True가 되도록 수정할 것.
 - Large 블록은 서로 붙어있을 수 없음을 확인
 - 예: Between의 의미가 일상에서와 다름
 - 동일한 줄, 열, 또는 대각선 상에 위치해야 함.
 3. 타스키 월드 사용법 및 단순명제 각각의 의미 확인
 4. 주석(코멘트)은 세미콜론(";")으로 시작하며 실행과는 무관

단순명제

- 술어predicate와 적절한 수의 인자들을 활용하여 만든 명제
 - LeftOf(a, b)
 - Between(a, b, c) 등등
- 인자들의 순서가 중요함
 - LeftOf(a, b)와 LeftOf(b, a)는 의미가 다름.
- 술어가 일반적으로 맨 앞에 위치함.
 - 예외: 등호(=) 술어는 두 개의 인자 사이에 위치
 - $a = b$
 - Wittenstein's Sentences에 추가해서 확인해볼 것

연습문제

- p.25: 1.2
- p.26: 1.4

2. 주장과 반례

Tarski's World 사용법 2

- You try it(p. 64)
 - 논리추론 연습
1. Bill's Argument 열기
 2. 마지막 명제가 첫 세 개의 명제로부터 추론되는지 여부 판단하기
 - 즉, 첫 세 개의 명제가 참이라고 가정할 때, 마지막 명제도 참이되는지 여부 판단
 - 먼저 머리로 판단하기
 3. 새로운 월드(New World) 창을 생성하여 실험하기
 - 네 개의 블록을 생성한 후 각각에 a, b, c, d 이름 명명하기
 - 주의사항
 - 하나의 블록에 여러 개의 이름 명명 가능
 - 하지만 동일한 이름을 여러 개의 블록에는 사용 못함
 - 여기서는 네 개의 블록이 각각 다른 이름 갖도록 함
 4. 네 개의 블록들의 위치를 임의로 변경한 후 세 개의 전제들이 모두 참이되도록 만들 것
 5. 세 개의 전제가 모두 참인데 결론은 거짓이 되는 월드가 존재하는지 계속해서 실험할 것
 - 만약 그런 월드를 찾는다면 World Counterexample 1이란 파일명으로 저장할 것

용어 설명

- Premise: 전제
- Conclusion: 결론
- Argument: 주장

주장과 반례

- 주장이란?
 - 전제가 참이면 결론도 참이라고 말하는 것
 - 전제는 1개 이상의 명제들로 구성됨
 - 결론은 1개의 명제로 구성됨
- "전제로부터 결론이 따라온다"란?
 - 전제에 포함된 모든 명제들이 참이면 결론도 무조건 참이되는 경우
- "전제로부터 결론이 반드시 따라오는 것은 아니다"란?
 - 전제가 모두 참이지만 결론은 거짓인 경우
- 반례
 - 전제가 모두 참이지만 결론이 거짓이 되는 경우를 의미
 - 여기서는 Tarski's World에서 구현된 어떤 월드를 의미함.
- 어떤 주장이 거짓임을 보이는 과정
 - 전제가 모두 참이지만, 결론은 거짓인 월드 생성
- 어떤 주장이 참임을 보이는 과정
 - 보다 어려움
 - 어떤 월드에서든, 전제가 모두 참이면 결론도 반드시 참이다라는 것을 입증할 수 있어야 함.
 - 즉, 어떤 월드에서든 전제가 모두 참일 수 없거나, 만약에 모두 참이라면 결론도 참이다라는 것을 보여야 함.
 - 그런데 어떻게 모든 월드를 대상으로 삼을 수 있나?
 - 매우 어려운 질문임
 - 괴델의 완전성 정리와 깊이 연관됨.
 - 정형증명 방식을 활용하기도 함, 예를 들어, Fitch, Coq 등 컴퓨터 소프트웨어 활용

연습문제

- p. 66: 2.24 ~ 2.27

3. 논리연산자

- 논리연산자를 이용하여 보다 복잡한 명제를 생성한다.
- "부정" 연산자
- "그리고" 연산자
- "또는" 연산자

3.1 부정 연산자: "¬"

- 형식: $\neg P$
- "... 가 아니다", "... 하지 않다" 등 어떤 명제의 부정을 표현할 때 사용
- P가 명제일 때 $\neg P$ 의 진리값은 P의 진리값을 부정하면 된다.
- 진리값을 부정하는 방식은 다음과 같다.

P	$\neg P$
True	False
False	True

예제

- p.69: You try it

1. Wittgenstein's World 열기
2. 새로운 sentence 파일 열기
3. 아래 명제의 진리값 추정하기

$\neg\neg\neg\neg\neg\neg\text{Between}(e, d, f)$

4. 추정값이 맞는지 확인하기 위해 게임 실행
 - 우측 상단에 있는 Play Game 버튼 누름
 - 게임요령
 - 먼저 진리값의 추정치 선택
 - 선택된 결과에 따라 다음 게임이 진행됨
 - 단계마다 컴퓨터의 요구사항에 맞추어 여러 선택을 해야 함.
 - 컴퓨터 또는 사용자가 이길 때까지 게임 진행됨.
 - 이후에 게임에 대한 보다 자세한 설명 예정

연속적으로 사용된 부정 연산자 다루기

- 기본 원칙: 부정의 부정은 긍정이다!
- 연속적으로 두 번 사용된 부정은 삭제 가능
- 결국 부정이 연속적으로 짝수 개수만큼 사용된 것은 삭제 가능
- 예: $\neg\neg\neg\neg\neg\neg\text{Between}(e, d, f)$ 와 $\neg\text{Between}(e, d, f)$ 은 동일한 진리값 가짐

연습문제

- Exc 3.3 (P. 70)

3.2 "그리고" 연산자: " \wedge "

- 형식: $P \wedge Q$
- 두 개의 명제가 동시에 참이라고 주장할 때 사용
- 두 개의 명제 P와 Q가 동시에 참이 되려면 P도 참이고, Q도 참이면 된다.
- $P \wedge Q$ 의 진리값은 아래 표에 따라 결정된다.

P	Q	$P \wedge Q$
True	True	True
True	False	False
False	True	False
False	False	False

예제

- You trye it(P.72)
1. Claire's World 열기
 2. 새로운 명제 파일 열기
 3. 아래 문장의 진리값 추정하기
 $\neg \text{Cube}(a) \wedge \neg \text{Cube}(b) \wedge \neg \text{Cube}(c)$
 4. 추정값이 맞는지 확인하기 위해 게임 실행
 - 추정값이 맞아도 게임에 질 수 있음
 - 게임진행이 이상할 경우 Back 버튼을 눌러 이전 단계로 되돌릴 수 있음
 - Tarski's World는 절대로 실수하지 않음

연습문제

- Exc. 3.7 (P. 74)

3.3 "또는" 연산자: " \vee "

- 형식: $P \vee Q$
- 두 개의 명제 중에 적어도 하나가 참이라고 주장할 때 사용
- 두 개의 명제 P와 Q 중에 적어도 하나가 참이 되려면 P가 참이거나 Q가 참이면 된다.
 - 주의: 둘 다 참이어도 된다.
- $P \vee Q$ 의 진리값은 아래 표에 따라 결정된다.

P	Q	$P \vee Q$
True	True	True
True	False	True
False	True	True
False	False	False

예제

- You trye it(P.76)
1. Ackermann's World 열기
 2. 새로운 명제 파일 열기
 3. 아래 문장의 진리값 추정하기
$$\text{Cube}(c) \vee \neg(\text{Cube}(a) \vee \text{Cube}(b))$$
 4. 추정값이 맞는지 확인하기 위해 게임 실행

연습문제

- Exc. 3.10 (P. 77)

4. 게임 관련

- 게임 규칙표

형태	사용자 추정값	선택 당사자	다음 단계
$P \vee Q$	True	사용자	P, Q 중에 True인 거 선택 후 게임 진행.
	False	—	P, Q 각각에 대해 False 추정값 확인 게임 진행하 둘 중에 하나가 True이면 사용자가 지도록 유도
$P \wedge Q$	True	—	P, Q 각각에 대해 True 추정값 확인 게임 진행하 둘 중에 하나가 False이면 사용자가 지도록 유도
	False	사용자	P, Q 중에 False인 거 선택 후 게임 진행
$\neg P$	아무거나	—	$\neg P$ 를 P로 대체 후 게임 진행.

- p.79, Exc 3.11
- Kleene's World and Sentences

5. 괄호의 중요성

- 복잡한 명제를 다룰 때 괄호에 주의해야 한다.
- 예제: 아래 두 명제는 서로 다른 의미를 가진다.

$$P \vee (Q \wedge R) \quad \text{vs} \quad (P \vee Q) \wedge R$$

예제

- Exc 3.14 (P.82)
- $\neg(\text{Small}(a) \vee \text{Small}(b))$ 는 참이지만 $\neg\text{Small}(a) \vee \text{Small}(b)$ 는 거짓이 되는 월드를 생성하라.

예제

- Exc 3.15 (P.82)
- $\text{Cube}(a) \wedge (\text{Cube}(b) \vee \text{Cube}(c))$ 는 참이지만 $(\text{Cube}(a) \wedge (\text{Cube}(b))) \vee \text{Cube}(c)$ 는 거짓이 되는 월드를 생성하라.

6 조건 연산자: " \rightarrow "

- 형식: $P \rightarrow Q$
- "P가 참이면 Q도 참이다" 라고 주장할 때 사용
- $P \wedge Q$ 의 진리값은 $\neg P \vee Q$ 의 진리값과 동일.

P	Q	$\neg P \vee Q$	$P \rightarrow Q$
True	True	True	True
True	False	False	False
False	True	True	True
False	False	True	True

- Tarski's World에서는 $P \wedge Q$ 를 바로 $\neg P \vee Q$ 로 변환해서 다룸.
- 동치연산자(" \leftrightarrow ")는 조건 연산자를 이용하여 정의됨. 즉, $P \leftrightarrow Q$ 는 $(P \rightarrow Q) \wedge (Q \rightarrow P)$ 의 줄임말임.

예제

- Exc 7.14(P.189)
 1. Euler's Sentences 열기
 2. 새로운 월드 열기
 3. Euler's Sentences의 명제들이 모두 참이되도록 하는 월드 생성하기
 - 함께 구현하기

연습문제

- Exc. 7.16 (P. 189)

조별 발표 게임

- 연습문제 풀기