C289043 이소연

인공지능 Problem Set #3

Problem #1:

다음은 클로즈의 집합(set of clauses) 이다.

(1) ~P ∨ ~Q ∨ R

(2) ~P ∨ Q ∨ ~R

(3) ~Q ∨ ~R ∨ ~W

(4) Q ∨ R

(5) Q ∨ W

(6) P ∨ ~Q

(7) ~Q ∨ W

효율적인 DPLL 알고리즘을 적용하여, 위 논리표현 전부를 참으로 만드는 모델(진리 값: truth assignment)이 있는지, 혹은 없는지를 판단해 보라.

Initial set of clauses S0:

(1) ~P ∨ ~Q ∨ R

(2) ~P ∨ Q ∨ ~R

(3) ~Q ∨ ~R ∨ ~W

(4) Q ∨ R

(5) Q ∨ W

(6) P ∨ ~Q

(7) ~Q ∨ W

Initial valuation V0: All atoms unbound. Sequence of calls.

1. Call dp1(ATOMS,S0,V0)

P, Q, R, W 모두 긍, 부정이 혼합되어 등장 => 순수 리터럴 없음.

No pure literals, no Singleton clauses.

Try V[P] := TRUE; V1 is the valuation V1[P]=True.

Delete clauses 6, delete ~P from 1, 2

S1:

(1) ~Q ∨ R

(2) Q ∨ ~R

(3) ~Q ∨ ~R ∨ ~W

(4) Q ∨ R

(5) Q ∨ W

(7) ~Q ∨ W

2. Call dp1(ATOMS,S1,V1)

No pure literals, no Singleton clauses.

Try V[Q] := TRUE; V2 is the valuation V2[P]=True, V2[Q] = TRUE.

Delete clauses 2,4,5 , delete ~Q from 1, 3,7

S2:

(1) R

(3) ~R ∨ ~W

(7) W

3. Call dp1(ATOMS,S2,V2)

1 is a singleton clause with literal R; V[R] = TRUE;

V3 is the valuation V3[P]=TRUE, V3[Q]=TRUE, V3[R]=TRUE

Delete clause 1, delete ~R from clause 3.
New set of clauses S3:

(3) ~W

(4) W

⇨ 모순 발생

4. Call dp1(ATOMS,S3,V3)

Try V[Q]=FALSE;

V4[P]=TRUE, V4[Q]=FALSE

Delete clauses 1,3,7 , delete Q from 2,4,5

S4:

(2) ~R

(4) R

(5) W

⇨ Singleton clause 끼리 모순 발생 (2,4)

5. Call dp1(ATOMS,S4,V4)

Try V[P]=FALSE;

V5[P]=FASLE;.

Delete clauses 1,2, delete P from 6

S5:

(3) ~Q ∨ ~R ∨ ~W

(4) Q ∨ R

(5) Q ∨ W

(6) ~Q

(7) ~Q ∨ W

6. Call dp1(ATOMS,S5,V5)

6 is a singleton clause with literal ~Q

V[Q]=FLASE

V6[P]=FASLE, V6[Q]=FALSE;

Delete clauses 3,6,7, delete Q from 4,5

S6:

(4) R

(5) W

7. Call dp1(ATOMS,S6,V6)

4 is a singleton clause with literal R

V[R]=TRUE

V7[P]=FASLE, V7[Q]=FALSE, V7[R]=TRUE;

Delete clauses 4

S7:

(5) W

8.

7. Call dp1(ATOMS,S7,V7)

5 is a singleton clause with literal W

V[W]=TRUE

V8[P]=FASLE, V8[Q]=FALSE, V8[R]=TRUE, V8[W]=TRUE

Delete clauses 5

S8: empty set of clauses.

➔ V[P]=FASLE, V[Q]=FALSE, V[R]=TRUE, V[W]=TRUE 일 때 SAT 하다.

Problem #2:

A. 다음의 논리식을 CNF 형태로 변형하라. (여기서,"~" 은 negation 의미)

P <=> (Q ∧ ~R).

W => P.

R <=> S.

S => P.

P => (~(Q ∨ W) ∨ S).

1. ~P v Q
2. ~P v ~R
3. ~Q v R v P
4. ~W v P
5. ~R v S
6. ~S v R
7. ~S v P
8. ~P v ~Q v S
9. ~P v ~W v S

B. 효율적인 DPLL 알고리즘을 적용하여 satisfiable 한지 아닌지를 판단하라.

S0 :

(1)~P v Q

(2)~P v ~R

(3)-~Q v R v P

(4)~W v P

(5)~R v S

(6)~S v R

(7)~S v P

(8)~P v ~Q v S

(9)~P v ~W v S

Step 1. Call dp1(ATOMS,S0,V0)

~W is a pure literal. (W never appears) V1[W]=FALSE.

Delete clauses 4, 9

S1:

(1)~P v Q

(2)~P v ~R

(3)-~Q v R v P

(5)~R v S

(6)~S v R

(7)~S v P

(8)~P v ~Q v S

No pure literals, no singleton clauses.
Try V[P] := TRUE; V2 is the valuation V2[P] = TRUE, V2[W] = FALSE. Call propagate(P,S1,V2):
Delete clauses 3,7 , delete ~P from 1,2,8
New set of clauses S2:

(1) Q

(2) ~R

(5) ~R v S

(6) ~S v R

(8) ~Q v S

Step 2. Call dp1(ATOMS, S2, V2).
1 is a singleton clause with literal Q

V[Q]=TRUE

V3[P] = TRUE, V3[Q]=TRUE , V3[W] = FALSE ;

Delete clause 1, delete ~Q from 8
New set of clauses S3:

(2) ~R

(5) ~R v S

(6) ~S v R

(8) S

3. Call dp1(ATOMS, S3, V3).

2 is a singleton clause with literal ~R

V[R]=FALSE;

~ V4[P] = TRUE, V4[Q]=TRUE , V4[R]=FALSE, V4[W] = FALSE ;

Delete clause 2,5 , delete R from 5
New set of clauses S4:

(6) ~S

(8) S

=> singleton clauses 간 모순 발생 => UNSAT

4. Call dp1(ATOMS, S4, V4)

TRY V[P]=FALSE;

V5[P] = FALSE, V5[W] = FALSE.

Delete clause 1,2,8 , delete P from 3,7

S5:

(3) ~Q v R

(5) ~R v S

(6) ~S v R

(7) ~S

5. Call dp1(ATOMS, S5, V5)

7 is a singleton clause with literal ~S

V[S] = FALSE;

V3 is the valuation V6[P] = FALSE, V6[S]=FALSE, V6[W] = FALSE.

Delete clause 6,7 , delete S from clause 5
S6:

(3) ~Q v R

(5) ~R

6. Call dp1(ATOMS, S6, V6)

5 is a singleton clause with literal ~R

V[R] = FALSE;

V3 is the valuation V7[P] = FALSE, V7[R]=FALSE, V7[S]=FALSE, V7[W] = FALSE.

Delete clause 5 , delete R from clause 3
S7:

(3) ~Q

7. Call dp1(ATOMS, S7, V7)

3 is a singleton clause with literal ~Q

V[Q] = FALSE;

V3 is the valuation V8[P] = FALSE, V8[Q]=FALSE, V8[R]=FALSE, V8[S]=FALSE, V8[W] = FALSE

Delete clause 3
S8 : empty set of clauses.

➔ V[P] = FALSE, V[Q]=FALSE, V[R]=FALSE, V[S]=FALSE, V[W] = FALSE 할 때 SAT 하다.

Problem #3:

KB 에 있는 아래 룰들을 사용하여 ~P1,2 가 TRUE 임을 증명해 보자.

(여기서, ~ 은 negation; ⇔ 는 iff 의미하며 두개의 룰로 나누어도 됨.)

Rule 1: ~P1,1

Rule 2: B1,1 ⇔ P1,2 ∨ P2,1

Rule 3: B2,1 ⇔P1,1 ∨ P2,2 ∨ P3,1

Rule 4: ~B1,1

Rule 5: B2,1

A.Forward chaining 방법으로 "~P1,2 가 TRUE" 임을 추론해 보라.

1. R4 에서 B1,1 은 거짓
2. R2 에서

B1,1 => P1,2 v P2,1
~B1,1 => ~P1,2 ^ ~P2,1

B1,1 이 거짓이므로 P1,2 와 P2,1 둘 다 거짓

➔ ~P1,2 는 참

B. Backward chaining 방법으로 "~P1,2 가 TRUE" 임을 추론해 보라.

목표 = **~P1,2 가 TRUE (=P1,2 가 거짓)**

1. R2 에서 B1,1 이 거짓이면 P1,2 와 P2,1 모두 거짓

> P1,2 가 거짓이어야 하므로 B1,1 과 P2,1 모두 거짓이어야 함.

2. R4 에서 B1,1 이 거짓인 것이 확인됨

➔ P1,2 가 거짓인 것을 확정할 수 있음

➔ ~P1,2 는 참

C. Resolution 방법으로 "~P1,2 가 TRUE" 임을 추론해 보라.

2, 4 결합 ➔ ~P1,2 v ~P2,1

P1,2 가 참이라면 Rule 2,4 에 모순이 발생한다.

따라서 P1,2 는 거짓이다.

➔ ~P1,2 는 참