Wang’s Algorithm

Automated Theorem Proving

Hao Wang 1960

Key Question and Idea

Is it possible to automatically prove logical statements?

Use manipulations to transform sequent into desirable form, where it easy to determine if statement is true or false!

$$(p \to q) \land (\neg r \to \neg q) \vdash p \to r$$

Operators

- $\vdash$ sequent (|=)
- $\neg$ negation (!)
- $\wedge$ conjunction (&)
- $\vee$ disjunction (v)
- $\rightarrow$ implication (->)
- $\leftrightarrow$ equivalence (<->)

Wang’s algorithm

- Rule 1: Replace negation on the left or the right.
- Rule 2: Replace conjunction by commas on the left.
- Rule 3: Replace disjunction by commas on the right.
- Rule 4: Branch disjunction on the left.
- Rule 5: Branch conjunction on the right.
- Rule 6: Replace implication on the left or the right.
- Rule 7: Replace equivalence on the left or the right.

Check statement for Tautology if no rule applies!

Rule 1

Replace negation on the left or the right.

$$\neg p, q \vdash r$$

$$\Downarrow$$

$$q \vdash p, r$$

Rule 2 and Rule 3

Replace conjunction by commas on the left.
Replace disjunction by commas on the right.

$$p \land q \vdash p \lor q$$

$$\Downarrow$$

$$p, q \vdash p, q$$

Rule 4

Branch disjunction on the left.

$$p \vee q \vdash p \wedge q$$

$$\Downarrow \qquad\qquad\qquad \Downarrow$$

$$p \vdash p \wedge q \qquad\qquad\qquad q \vdash p \wedge q$$

Rule 5

Branch conjunction on the right.

$$p \vee q \vdash p \wedge q$$

$$\Downarrow \qquad\qquad\qquad\qquad \Downarrow$$

$$p \vee q \vdash p \qquad\qquad\qquad p \vee q \vdash q$$

Rule 6 and Rule 7

Replace implication on the left or the right.
Replace equivalence on the left or the right.

$$p \leftrightarrow q \vdash p \rightarrow q$$

$$\Downarrow$$

$$(p \rightarrow q) \wedge (q \rightarrow p) \vdash \neg p \vee q$$

Tautology

"… is a formula or assertion that is true in every possible interpretation."

If there are no more manipulations possible, compare the lists on the right and the left.

$$p, q, r \vdash p, q, r$$

$$p \wedge q \wedge r \ \subset \ p \vee q \vee r$$

$$(s, t \vdash p, q)$$

To be, or not to be, ...

$\vdash \text{tobe} \lor \neg\text{tobe}$

```
?- prove([[]],[[tobe v !tobe]]).

Statement: [] |= [tobe v !tobe]

Attempting proof!

*Rule 3                                  [] |= [tobe,!tobe]
*Rule 1R                                 [tobe] |= [tobe]
*Tautology?                              [tobe] |= [tobe]
*True.

Result: The given theorem is true.
```

```prolog
% Main call.
prove([],[]).
prove([L|P],[R|A]):-
  nl, ansi_format([bold],'Statement: ',[]), write(L), write(' |= '), write
  ansi_format([bold],'Attempting proof!',[]), nl, nl,
  wang(L,R),
  prove(P,A).

% Procedure of Wang to prove theorem.
wang(L,R):-
  rules(L,R,[],0), nl,
  ansi_format([bold,fg(green)], 'Result: The given theorem is true.',[]), nl, nl;
  ansi_format([bold, fg(red)],'Proof failed for current step!',[]), nl, nl,
  ansi_format([bold,fg(red)], 'Result: The given theorem is false.',[]), nl, nl.

% Move negations from left to right.
rules(L,R,S,T):-
  member(!X,L),
  delete(L,!X,Ld),
  append(S,[[['*Rule 1L                 '],[Ld,' |= ',[X|R]],T]],Sa),
  rules(Ld,[X|R],Sa,T).

% Move negations from right to left.
rules(L,R,S,T):-
  member(!X,R),
  delete(R,!X,Rd),
  append(S,[[['*Rule 1R                 '],[[X|L],' |= ',Rd],T]],Sa),
  rules([X|L],Rd,Sa,T).
```

```prolog
% Main call.
prove([],[]):-nl.
prove([L|P],[R|A]):-
  nl, write(L), write(' |= '), write(R), nl,
  wang(L,R),
  prove(P,A).

% Procedure of Wang to prove theorem.
wang(L,R):-
  rules(L,R),
  write('TRUE'), nl.
wang(_,_):-
  write('FALSE'), nl.
```

```prolog
% Replace equivalence on the right.
rules(L,R):-
  member(X <-> Y,R),
  delete(R,X <-> Y,Rd),
  rules(L,[(X -> Y) & (Y -> X)|Rd]).

% Finally compare both sides.
rules(L,R):-
  member(X,L),
  member(X,R).
```

```
?- prove([[(p & q) & r]],[[p & (q & r)]]).

Statement: [(p&q)&r] |= [p&q&r]

Attempting proof!

*Rule 2                                  [p&q,r] |= [p&q&r]
*Rule 2                                  [p,q,r] |= [p&q&r]
*Rule 5a - Branch Level 0                [p,q,r] |= [p]
     *Tautology?                         [p,q,r] |= [p]
     *True.
*Rule 5b - Branch Level 0                [p,q,r] |= [q&r]
     *Rule 5a - Branch Level 1           [p,q,r] |= [q]
          *Tautology?                    [p,q,r] |= [q]
          *True.
     *Rule 5b - Branch Level 1           [p,q,r] |= [r]
          *Tautology?                    [p,q,r] |= [r]
          *True.

Result: The given theorem is true.
```

```
Statement: [(p->q)&(!r->!q)] |= [p->r]

Attempting proof!

*Rule 2                                        [p->q,!r->!q] |= [p->r]
*Rule 6L                                       [!p v q,!r->!q] |= [p->r]
*Rule 4a - Branch Level 0                      [!p,!r->!q] |= [p->r]
     *Rule 1L                                  [!r->!q] |= [p,p->r]
     *Rule 6L                                  [!!r v !q] |= [p,p->r]
     *Rule 4a - Branch Level 1                 [!!r] |= [p,p->r]
          *Rule 1L                             [] |= [!r,p,p->r]
          *Rule 1R                             [r] |= [p,p->r]
          *Rule 6R                             [r] |= [!p v r,p]
          *Rule 3                              [r] |= [!p,r,p]
          *Rule 1R                             [p,r] |= [r,p]
          *Tautology?                          [p,r] |= [r,p]
          *True.
     *Rule 4b - Branch Level 1                 [!q] |= [p,p->r]
          *Rule 1L                             [] |= [q,p,p->r]
          *Rule 6R                             [] |= [!p v r,q,p]
          *Rule 3                              [] |= [!p,r,q,p]
          *Rule 1R                             [p] |= [r,q,p]
          *Tautology?                          [p] |= [r,q,p]
          *True.
```

```
*Rule 4b - Branch Level 0                    [q,!r->!q] |= [p->r]
     *Rule 6L                                [!!r v !q,q] |= [p->r]
     *Rule 4a - Branch Level 1               [!!r,q] |= [p->r]
          *Rule 1L                           [q] |= [!r,p->r]
          *Rule 1R                           [r,q] |= [p->r]
          *Rule 6R                           [r,q] |= [!p v r]
          *Rule 3                            [r,q] |= [!p,r]
          *Rule 1R                           [p,r,q] |= [r]
          *Tautology?                        [p,r,q] |= [r]
          *True.
     *Rule 4b - Branch Level 1               [!q,q] |= [p->r]
          *Rule 1L                           [q] |= [q,p->r]
          *Rule 6R                           [q] |= [!p v r,q]
          *Rule 3                            [q] |= [!p,r,q]
          *Rule 1R                           [p,q] |= [r,q]
          *Tautology?                        [p,q] |= [r,q]
          *True.

Result: The given theorem is true.
```

- Hao Wang, Toward Mechanical Mathematics, IBM Journal of Research and Development, volume 4, 1960.
- Stuart Russell and Peter Nerving. 2009. Artificial Intelligence: A Modern Approach (3rd edition). Prentice Hall Press, Upper Saddle River, NJ, USA. (Chapters 8, 9 and 12)
- Štěpán, Jan. "Propositional calculus proving methods in Prolog." Acta Universitatis Palackianae Olomucensis. Facultas Rerum Naturalium. Mathematica 29.1 (1990): 301-321.