# Propositional Horn clauses

## Data structures

Atoms $P_1, \ldots, P_n \quad \mapsto \quad \{1, \ldots, n\}$

neg-occ-list(A): list of all clauses in which $A$ occurs negatively

pos-occ-list(A): list of all clauses in which $A$ occurs positively

| Clause: | $P_1$ | $P_2$ | $\ldots$ | $P_n$ | counter |
|---|---|---|---|---|---|
| | neg | neg | | pos | ↑ |
| | | ↑ | | | number of literals |

first-active-literal (fal): first literal not marked as deleted.

| atom status: | | |
|---|---|---|
| | pos | (deduced as positive unit clause) |
| | neg | (deduced as negative unit clause) |
| | nounit | (otherwise) |

# Variable-free Horn clauses

**Input:** Set $N$ of Horn formulae

Step 1. Collect unit clauses; check if complementary pairs exist

**forall** $C \in N$ **do**

**if** is-unit(C) **then begin** const. time

L := first-active-literal(C) const. time

**if** state(atom(L)) = nounit **then** state(atom(L)) = sign(L) const. time

push(atom(L), stack)

**else if** state(atom(L)) $\neq$ sign(L) **then return false**

# Variable-free Horn clauses

2. Process the unit clauses in the stack

**while** stack $\neq \emptyset$ **do**

**begin** A := top(stack); pop(stack)

**if** state(A) = pos **then** delete-literal-list := neg-oc-list(A) O(# neg-oc-list)

**else** delete-literal-list := pos-oc-list(A) O(# pos-oc-list)

**endif**

**for all** C in delete-literal-list **do**

**if** state(A) = pos **then** delete-literal(A,C) const. time + nfal - ofal

**if** state(A) = neg **then** delete-literal($\neg$ A,C) const. time + nfal - ofal

**if** unit(C) **then** L1 := first-active-literal(C) const. time

**if** state(atom(L1)) = nounit **then** state(atom(L1)) = sign(L1),
L1 $\rightarrow$ stack

**elseif** state(atom(L1)) $\neq$ sign(L1) **then** return false

**endif**

**end**