1.6 Splitting into Horn Clauses

- A *Horn clause* is a clause with at most one positive literal.
- They are typically denoted as implications: $P_1, \ldots, P_n \rightarrow Q$.
  (In general we can write $P_1, \ldots, P_n \rightarrow Q_1, \ldots, Q_m$ for $\neg P_1 \vee \ldots \vee \neg P_n \vee Q_1 \vee \ldots \vee Q_m$.)
- Compared to arbitrary clause sets, Horn clause sets enjoy further properties:
  - Horn clause sets have unique minimal models.
  - Checking satisfiability is often of lower complexity.

Propositional Horn Clause SAT is in $\mathbf{P}$

boolean HornSAT(literal set M, Horn clause set N) {
    if (all clauses in N are supported by M) return true;
    elsif (a negative clause in N is not supported by M) return false;
    elsif (N contains clause $P_1, \ldots, P_n \rightarrow Q$ where
            $\{P_1, \ldots, P_n\} \subseteq M$ and $Q \notin M$)
        return HornSAT($M \cup \{Q\}$, N);
}

A clause $P_1, \ldots, P_n \rightarrow Q_1, \ldots, Q_m$ is *supported* by M if $\{P_1, \ldots, P_n\} \not\subseteq M$ or some $Q_i \in M$. A *negative* clause consists of negative literals only.

Initially, HornSAT is called with an empty literal set M.

Lemma 1.13 *Let N be a set of propositional Horn clauses.Then:*

(1) HornSAT($\emptyset$, N)=true iff N is satisfiable

(2) HornSAT is in $\mathbf{P}$

Proof. (1) (Idea) For example, by induction on the number of positive literals in N.

(2) (Scetch) For each recursive call M contains one more positive literal. Thus HornSAT terminates after at most n recursive calls, where n is the number of propositional variables in N. □

SplitHornSAT

boolean SplitHornSAT(clause set N) {
    if (N is Horn)
g   return HornSAT($\emptyset$,N);
    else {
        select non Horn clause $P_1, \ldots, P_n \rightarrow Q_1, \ldots, Q_m$ from N;
        $N' = N \setminus \{P_1, \ldots, P_n \rightarrow Q_1, \ldots, Q_m\}$;
        if (SplitHornSAT($N' \cup \{P_1, \ldots, P_n \rightarrow Q_1\}$)) return true;
        else return
          SplitHornSAT($N' \cup \{\rightarrow Q_2, \ldots, Q_m\} \cup \bigcup_i \{\rightarrow P_i\} \cup \{Q_1 \rightarrow\}$);
    }
}

Lemma 1.14 *Let N be a set of propositional clauses. Then:*

(1) SplitHornSAT(N)=true iff N is satisfiable

(2) SplitHornSAT(N) terminates

Proof. (1) (Idea) Show that N is satisfiable iff $N' \cup \{P_1, \ldots, P_n \rightarrow Q_1\}$ is satisfiable or $N' \cup \{\rightarrow Q_2, \ldots, Q_m\} \cup \bigcup_i \{\rightarrow P_i\} \cup \{Q_1 \rightarrow\}$ is satisfiable for some clause $P_1, \ldots, P_n \rightarrow Q_1, \ldots, Q_m$ from N.

(2) (Idea) Each recursive call reduces the number of positive literals in non Horn clauses. □

1.7 Other Calculi

OBDDs (Ordered Binary Decision Diagrams):

Minimized graph representation of decision trees, based on a fixed ordering on propositional variables,

see script of the Computational Logic course,

see Chapter 6.1/6.2 of Michael Huth and Mark Ryan: *Logic in Computer Science: Modelling and Reasoning about Systems*, Cambridge Univ. Press, 2000.

FRAIGs (Fully Reduced And-Inverter Graphs)

Minimized graph representation of boolean circuits.

1.8 Example: SUDOKU

	1	2	3	4	5	6	7	8	9
1								1	
2	4								
3		2							
4					5		4		7
5			8				3		
6			1		9				
7	3			4			2		
8		5		1					
9				8		6			

Idea: $p_{i,j}^d$=true iff the value of square i, j is d

For example: $p_{3,5}^8 = true$

Coding SUDOKU by propositional clauses

- Concrete values result in units: $p_{i,j}^d$
- For every value, column we generate: $\neg p_{i,j}^d \vee \neg p_{i,j+k}^d$
  Accordingly for all rows and 3×3 boxes
- For every square we generate: $p_{i,j}^1 \vee \ldots \vee p_{i,j}^9$
- For every two different values, square we generate: $\neg p_{i,j}^d \vee \neg p_{i,j}^{d'}$
- For every value, column we generate: $p_{i,0}^d \vee \ldots \vee p_{i,9}^d$
  Accordingly for all rows and 3×3 boxes

Constraint Propagation is Unit Propagation

	1	2	3	4	5	6	7	8	9
1								1	
2	4								
3		2							
4					5		4		7
5			8				3		
6			1		9				
7	3			4	**7**		2		
8		5		1					
9				8		6			

From $\neg p_{1,7}^3 \vee \neg p_{5,7}^3$ and $p_{1,7}^3$ we obtain by unit propagating $\neg p_{5,7}^3$ and further from $p_{5,7}^1 \vee p_{5,7}^2 \vee p_{5,7}^3 \vee p_{5,7}^4 \vee \ldots \vee p_{5,7}^9$ we get $p_{5,7}^1 \vee p_{5,7}^2 \vee p_{5,7}^4 \vee \ldots \vee p_{5,7}^9$.