Unit Propagation (1): In a clause

- If a literal is true, the clause is true and can be removed
  - If $a \leftarrow true$, remove all clauses where it is positive
  \[ a \lor b \lor c \]
  - If $a \leftarrow false$, remove all clauses where it is negative
  \[ \neg a \lor b \lor c \]
- If a literal is false, it can be removed from the clause
  - If $a \leftarrow false$, remove it from all clauses where it is positive
  \[ a \lor b \lor c \quad b \lor c \]
  - If $a \leftarrow true$, remove all clauses where it is negative
  \[ \neg a \lor b \lor c \quad b \lor c \]
Unit Propagation (2)

• When a literal in a clause is true, the entire clause is true and can be removed

\[
\begin{align*}
\text{After assignment: } & \quad a \leftarrow \text{false} \\
& \quad \cdots \land C_i \land (\neg a \lor b \lor \neg c) \land C_{i+2} \land \cdots \\
& \equiv \quad \cdots \land C_i \land (\text{true} \lor b \lor \neg c) \land C_{i+2} \land \cdots \\
& \equiv \quad \cdots \land C_i \land \text{true} \land C_{i+2} \land \cdots \\
& \equiv \quad \cdots \land C_i \land C_{i+2} \land \cdots 
\end{align*}
\]
Unit Propagation (3)

• When a literal in a clause is false, the literal may be removed from the clause

\[ \neg a \lor b \lor \neg c \]

After assignment: \( a \leftarrow \text{true} \)

\[ \equiv (\text{false} \lor b \lor \neg c) \]

\[ \equiv (b \lor \neg c) \]
Unit Propagation (4)

• Assignments may result in unit clauses
  \[ a \leftarrow false \text{ and } a \lor b \text{ yield } b \]

• Unit clauses immediately force an assignment
  \[ b \leftarrow true \]

• This can lead to a chain reaction as new assignments ‘propagate’ throughout the clauses
Unit Propagation: Example

- SAT formula with 1,000 variables
- Cycle of implications
  \[ v_1 \rightarrow v_2 \]
  \[ v_2 \rightarrow v_3 \]
  \[ \vdots \]
- A single assignment results in unit propagation to solve the entire problem

\[
\neg v_1 \lor v_2 \\
\land \neg v_2 \lor v_3 \\
\vdots \\
\land \neg v_{999} \lor v_{1000} \\
\land \neg v_{1000} \lor v_1
\]
Unit Resolution

- Unit resolution or disjunctive syllogism

\[
\begin{align*}
& a, \neg a \lor b \\
\Rightarrow & b
\end{align*}
\]

- Unit propagation has similarities to unit resolution
  - Assignments add unit clauses to the formula
  - When a new unit clause is added, it can be resolved with other clauses in the formula and the result added to the formula

\[
(a) \land (\neg a \lor b \lor \neg c) \rightarrow (b \lor \neg c)
\]
Thrashing in Search

• Repeatedly hitting conflict within a subtree of the search tree

• Thrashing is costly
Fail-First Principle (FFP)

• It is better to fail early than to waste time exploring a subtree with no solutions

• Want to handle potential conflicts earlier
Variable Ordering Heuristics

• The order in which variables are assigned
• Heuristic: good performance in general, not guaranteed to be optimal
• Static versus dynamic
  – Static: Assignment order is decided before search and maintained fixed throughout search
  – Dynamic: Variable ordering is adjusted during the course of the search
Activity-Based Heuristics

• Variables are assigned an ‘activity’ value
• Variables involved in a conflict have their activities increased
• Activities exponentially decay
• Select most ‘active’ variable

\[
(\neg a \lor \neg c \lor d) \times \quad Activity(a)+ = 1 \quad Activity(c)+ = 1 \quad Activity(d)+ = 1
\]
Activity-Based Heuristics Example

• UNSAT formula with 1000 variables
• Conflict occurs between $\nu_{999}$ and $\nu_{1000}$

\[ \begin{align*}
&\land (\nu_{999} \lor \nu_{1000}) \\
&\land (\neg \nu_{999} \lor \nu_{1000}) \\
&\land (\nu_{999} \lor \neg \nu_{1000}) \\
&\land (\neg \nu_{999} \lor \neg \nu_{1000})
\end{align*} \]

Lexicographic ordering  Activity-based ordering