

1. Goal : 這個 project 使用 C++ 和現成的 parser, 實作 DPLL 演算法來解決 SAT 問題

2.Feature :

1. 2-literal watching: 完成 assign 和 unassign 更有效率, 它的原理是用 pointer 監看 每個 clause 的 literal , 檢查有沒有 conflict 和 unit clause 產生, 並建立一個 table 去維護這些 pointer。

2. Non-Chronological Backtracking: 利用重複溶解目前的 cut 與 選定的 clause 以及 assign variable of clause 來找出 UIP 和 conflict clause。

3. Random restart: 先用 50 ~ 70 之間的隨機變數生成 restart time (單位: seconds), 當 solver 執行時間超過 restart time, 就 restart 。

4. Clause learning policy: 太長的 learning clause 不是一個很好的選擇, 生成十個 solver, 當 clause 長度超過 $\max(n*i/10, 10)$ 就跳過。

5. Heuristic: 將 parser 的 CNF 用長度排序，
由大到小排序，會有比較好的結果。

3. Summary: 雖然我的 project 能讓 19500 個變數、103887 個 clause 產生結果，但跑不太動助教提供的 par32-1-c.cnf 和 par32-1.cnf，可能跟問題的邏輯複雜度有關，此外將 solver 增加，提升所要學習的 clause 的長度，可以提升解決問題的效率。

4. 心得: 我在 milestone 1 比較有空的時候就實作 solver 大部分的功能，這次主要調整上一次沒有寫好的 heuristic 和 random restart 。