```cpp
// include strongly connected components

struct cnf2_t {
  int n; // size of variables
  struct literal_t {
    int index;
    bool neg;
  };
  struct clause_t {
    literal_t x, y;
  };
  vector<clause_t> L;
};

class SAT2 {
private:
  int n;
  bool fail_flag;
  vvi sccs;
  vi scc_gid;
  vector<bool> result;
  int inv(int id) {
    return (id + n) % (n * 2);
  }
public:
  SAT2(cnf2_t CNF) {
    vvi lst(n * 2);
    Loop(i, CNF.L.size()) {
      lst[CNF.L[i].x.index + (CNF.L[i].x.neg ? 0 : n)].push_back(CNF.L[i].y.index + (CNF.L[i].y.neg ? n : 0));
      lst[CNF.L[i].y.index + (CNF.L[i].y.neg ? 0 : n)].push_back(CNF.L[i].x.index + (CNF.L[i].x.neg ? n : 0));
    }
    Strongly_Connected_Components *scc = new Strongly_Connected_Components(lst);
    sccs = scc->get_sccs();
    scc_gid = scc->get_scc_gid();
    fail_flag = false;
    result.resize(n);
    Loop(i, n) {
      if (scc_gid[i] > scc_gid[inv(i)]) result[i] = true;
      else if (scc_gid[i] < scc_gid[inv(i)]) result[i] = false;
      else {
        result.clear();
        fail_flag = true;
        return;
      }
    }
    return;
  }
  bool is_satisfiable() {
    return !fail_flag;
  }
  vector<bool> get_result() {
    return result;
  }
};
```