Laboratori LI. Primavera 2024

- Professor: Antoni Lozano
- C/E: antoni.lozano @ upc.edu
- Despatx: 233, edifici Ω

Informació general del curs

- Pàgina: `https://www.cs.upc.edu/~li/`
  Pràctiques de laboratori, apunts, exàmens
  Novetat $\longrightarrow$ exàmens resolts de laboratori (2021–2023)
- Racó: avisos i lliurament de pràctiques

Pràctiques

- Hi ha 6 pràctiques i hi dediquem 2 dies per pràctica:
  - P1: SAT solver
  - P2: Introducció a Prolog
  - P3: Codificació en SAT
  - P4: Optimització en SAT
  - P5: Prolog avançat
  - P6: Constrained Logic Programming
- Cal entendre-les i també fer-les!

Parcials de laboratori

Hi ha 2 parcials de laboratori:

1. Parcial 1: 3/4/24, 15:30 –17:30
2. Parcial 2: 4/6/24, 11:30 –14:30

Pràctica 1: SAT solver

Lògica en la Informàtica

FIB

Antoni Lozano

Q2 2023–2024

Objectiu

Aquesta primera pràctica té com a objectius:

- enfrontar-se a un problema NP-complet com SAT
- adquirir uns coneixements bàsics sobre els SAT solvers

Per fer-ho, caldrà millorar l'eficiència d'un SAT solver de joguina que es proporciona.

Conceptes bàsics

- **SAT**: decidir si una fórmula de lògica proposicional té un model
- **Lògica proposicional**
  - Sintaxi. Les fórmules es construeixen amb símbols de predicat

    $$P = \{p, q, r, \dots\}$$

    i les connectives

    $$\wedge \text{ (conjunció) }, \vee \text{ (disjunció) }, \neg \text{ (negació)}$$

  - Semàntica. Una interpretació és una funció $I : P \longrightarrow \{0, 1\}$.
    Diem que I satisfà una fórmula F ($I \models F$) si $\text{eval}_I(F) = 1$.

Conceptes bàsics

- **SAT**: decidir si una fórmula de lògica proposicional té un model
- **Lògica proposicional**
  - Sintaxi. Les fórmules es construeixen amb símbols de predicat

    $$P = \{p, q, r, \dots\}$$

    i les connectives

    $$\wedge \text{ (conjunció) }, \vee \text{ (disjunció) }, \neg \text{ (negació)}$$

  - Semàntica. Una interpretació és una funció $I : P \longrightarrow \{0, 1\}$.
    Diem que I satisfà una fórmula F ($I \models F$) si $\mathrm{eval}_I(F) = 1$.

Conceptes bàsics

- Exemple de fórmula F: $p \wedge ((q \vee \neg r) \wedge ((\neg p \vee r) \wedge \neg q))$
- Exemple d'interpretació: $I(p) = 0, I(q) = 1, I(r) = 0$
- $$\begin{aligned} \mathrm{eval}_I(F) &= 0 \wedge ((1 \vee \neg 0) \wedge ((\neg 0 \vee 0) \wedge \neg 1)) \\ &= 0 \wedge ((1 \vee 1) \wedge ((1 \vee 0) \wedge 0) \\ &= 0 \wedge (1 \wedge (1 \wedge 0)) \\ &= 0 \wedge (1 \wedge 0) \\ &= 0 \wedge 0 \\ &= 0 \end{aligned}$$
- Per tant, F és falsa amb la interpretació I

Conceptes bàsics

- Exemple de fórmula F: $p \land ((q \lor \neg r) \land ((\neg p \lor r) \land \neg q))$
- Exemple d'interpretació: $I(p) = 0, I(q) = 1, I(r) = 0$
- $$\begin{aligned} \text{eval}_I(F) &= 0 \land ((1 \lor \neg 0) \land ((\neg 0 \lor 0) \land \neg 1)) \\ &= 0 \land ((1 \lor 1) \land ((1 \lor 0) \land 0) \\ &= 0 \land (1 \land (1 \land 0)) \\ &= 0 \land (1 \land 0) \\ &= 0 \land 0 \\ &= 0 \end{aligned}$$
- Per tant, F és falsa amb la interpretació I

Conceptes bàsics

- Exemple de fórmula F: $p \land ((q \lor \neg r) \land ((\neg p \lor r) \land \neg q))$
- Exemple d'interpretació: $I(p) = 0, I(q) = 1, I(r) = 0$
- $\text{eval}_I(F) = 0 \land ((1 \lor \neg 0) \land ((\neg 0 \lor 0) \land \neg 1))$

$$
\begin{aligned}
&= 0 \land ((1 \lor 1) \land ((1 \lor 0) \land 0) \\
&= 0 \land (1 \land (1 \land 0)) \\
&= 0 \land (1 \land 0) \\
&= 0 \land 0 \\
&= 0
\end{aligned}
$$

- Per tant, F és falsa amb la interpretació I

Conceptes bàsics

- Exemple de fórmula F: $p \land ((q \lor \neg r) \land ((\neg p \lor r) \land \neg q))$
- Exemple d'interpretació: $I(p) = 0, I(q) = 1, I(r) = 0$
- $$\begin{aligned} \text{eval}_I(F) &= 0 \land ((1 \lor \neg 0) \land ((\neg 0 \lor 0) \land \neg 1)) \\ &= 0 \land ((1 \lor 1) \land ((1 \lor 0) \land 0) \\ &= 0 \land (1 \land (1 \land 0)) \\ &= 0 \land (1 \land 0) \\ &= 0 \land 0 \\ &= 0 \end{aligned}$$
- Per tant, F és falsa amb la interpretació I

Conceptes bàsics

- Exemple de fórmula F: $p \land ((q \lor \neg r) \land ((\neg p \lor r) \land \neg q))$
- Exemple d'interpretació: $I(p) = 0, I(q) = 1, I(r) = 0$
- $$\begin{aligned} \mathrm{eval}_I(F) &= 0 \land ((1 \lor \neg 0) \land ((\neg 0 \lor 0) \land \neg 1)) \\ &= 0 \land ((1 \lor 1) \land ((1 \lor 0) \land 0) \\ &= 0 \land (1 \land (1 \land 0)) \\ &= 0 \land (1 \land 0) \\ &= 0 \land 0 \\ &= 0 \end{aligned}$$
- Per tant, F és falsa amb la interpretació I

Conceptes bàsics

- Exemple de fórmula F: $p \wedge ((q \vee \neg r) \wedge ((\neg p \vee r) \wedge \neg q))$
- Exemple d'interpretació: $I(p) = 0, I(q) = 1, I(r) = 0$
- $$\begin{aligned} \mathrm{eval}_I(F) &= 0 \wedge ((1 \vee \neg 0) \wedge ((\neg 0 \vee 0) \wedge \neg 1)) \\ &= 0 \wedge ((1 \vee 1) \wedge ((1 \vee 0) \wedge 0) \\ &= 0 \wedge (1 \wedge (1 \wedge 0)) \\ &= 0 \wedge (1 \wedge 0) \\ &= 0 \wedge 0 \\ &= 0 \end{aligned}$$
- Per tant, F és falsa amb la interpretació I

Conceptes bàsics

- Exemple de fórmula F: $p \land ((q \lor \neg r) \land ((\neg p \lor r) \land \neg q))$
- Exemple d'interpretació: $I(p) = 0, I(q) = 1, I(r) = 0$
- $$\begin{aligned} \mathrm{eval}_I(F) &= 0 \land ((1 \lor \neg 0) \land ((\neg 0 \lor 0) \land \neg 1)) \\ &= 0 \land ((1 \lor 1) \land ((1 \lor 0) \land 0) \\ &= 0 \land (1 \land (1 \land 0)) \\ &= 0 \land (1 \land 0) \\ &= 0 \land 0 \\ &= 0 \end{aligned}$$
- Per tant, F és falsa amb la interpretació I

Conceptes bàsics

- Exemple de fórmula F: $p \wedge ((q \vee \neg r) \wedge ((\neg p \vee r) \wedge \neg q))$
- Exemple d'interpretació: $I(p) = 0, I(q) = 1, I(r) = 0$
- $$\begin{aligned} \mathrm{eval}_I(F) &= 0 \wedge ((1 \vee \neg 0) \wedge ((\neg 0 \vee 0) \wedge \neg 1)) \\ &= 0 \wedge ((1 \vee 1) \wedge ((1 \vee 0) \wedge 0) \\ &= 0 \wedge (1 \wedge (1 \wedge 0)) \\ &= 0 \wedge (1 \wedge 0) \\ &= 0 \wedge 0 \\ &= 0 \end{aligned}$$
- Per tant, F és falsa amb la interpretació I

Conceptes bàsics

- Exemple de fórmula F: $p \wedge ((q \vee \neg r) \wedge ((\neg p \vee r) \wedge \neg q))$
- Exemple d'interpretació: $I(p) = 0, I(q) = 1, I(r) = 0$
- $$\begin{aligned} \mathrm{eval}_I(F) &= 0 \wedge ((1 \vee \neg 0) \wedge ((\neg 0 \vee 0) \wedge \neg 1)) \\ &= 0 \wedge ((1 \vee 1) \wedge ((1 \vee 0) \wedge 0) \\ &= 0 \wedge (1 \wedge (1 \wedge 0)) \\ &= 0 \wedge (1 \wedge 0) \\ &= 0 \wedge 0 \\ &= 0 \end{aligned}$$
- Per tant, F és falsa amb la interpretació I

Conceptes bàsics

- Totes les interpretacions de $F = p \wedge ((q \vee \neg r) \wedge ((\neg p \vee r) \wedge \neg q))$:

p	q	r	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Conceptes bàsics

- Si hi ha n símbols, quantes interpretacions hi ha?
- Una interpretació que satisfà una fórmula se'n diu model.
- F és insatisfactible, no té cap model.
- Quants models tenen les fórmules següents?

$$G = x_1 \wedge x_2 \wedge \cdots \wedge x_n$$

$$H = ((\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2)) \vee x_3 \vee \cdots \vee x_n$$

Conceptes bàsics

- Si hi ha n símbols, quantes interpretacions hi ha?
- Una interpretació que satisfà una fórmula se'n diu model.
- F és insatisfactible, no té cap model.
- Quants models tenen les fórmules següents?

$$G = x_1 \wedge x_2 \wedge \cdots \wedge x_n$$

$$H = ((\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2)) \vee x_3 \vee \cdots \vee x_n$$

Conceptes bàsics

- Si hi ha n símbols, quantes interpretacions hi ha?
- Una interpretació que satisfà una fórmula se'n diu model.
- F és insatisfactible, no té cap model.
- Quants models tenen les fórmules següents?

$$G = x_1 \wedge x_2 \wedge \cdots \wedge x_n$$

$$H = ((\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2)) \vee x_3 \vee \cdots \vee x_n$$

Conceptes bàsics

- Si hi ha n símbols, quantes interpretacions hi ha?
- Una interpretació que satisfà una fórmula se'n diu model.
- F és insatisfactible, no té cap model.
- Quants models tenen les fórmules següents?

$$G = x_1 \wedge x_2 \wedge \cdots \wedge x_n$$

$$H = ((\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2)) \vee x_3 \vee \cdots \vee x_n$$

SAT solvers

Els SAT solvers consideren les fórmules d'entrada en forma normal conjuntiva (CNF):

- Un literal és un símbol de predicat afirmat o negat
- Una clàusula és una disjunció de literals
- Una fórmula booleana en CNF és una conjunció de clàusules

SAT solvers

El format d’entrada per al nostre SAT solver és:

```
p cnf 300 1275
-255 15 257  0
239 1 115  0
100 -118 -152  0
-32 93 20  0
272 41 -269  0
-220 -251 -53  0
-9 254 129  0
-38 -71 -127  0
-52 113 186  0
-175 -281 -291  0
```

variables

clàusules

literal

$\neg x_{220}$

SAT solvers

Algorismes per decidir la satisfactibilitat:

- Cerca exhaustiva: enumerar totes les interpretacions i, per a cadascuna, avaluar la fórmula. Avaluar la fórmula donada la interpretació és lineal, però tenim 2^n interpretacions.
- CDCL-based SAT Solvers (Conflict-Driven Clause-Learning backtracking/backjumping algorithm), fets per Intel, Microsoft...

Exemple

Veure la presentació **backtrackingForSAT.pdf**.

SAT solvers

Algorismes per decidir la satisfactibilitat:

- Cerca exhaustiva: enumerar totes les interpretacions i, per a cadascuna, avaluar la fórmula. Avaluar la fórmula donada la interpretació és lineal, però tenim 2^n interpretacions.
- CDCL-based SAT Solvers (Conflict-Driven Clause-Learning backtracking/backjumping algorithm), fets per Intel, Microsoft...

Exemple

Veure la presentació **backtrackingForSAT.pdf**.

SAT solvers

Algorismes per decidir la satisfactibilitat:

- Cerca exhaustiva: enumerar totes les interpretacions i, per a cadascuna, avaluar la fórmula. Avaluar la fórmula donada la interpretació és lineal, però tenim 2^n interpretacions.
- CDCL-based SAT Solvers (Conflict-Driven Clause-Learning backtracking/backjumping algorithm), fets per Intel, Microsoft...

Exemple

Veure la presentació **backtrackingForSAT.pdf**.

SAT solvers

En aquest curs:

1. Triarem un problema P qualsevol
2. El codificarem com una fórmula proposicional F_P
3. Cridarem a un SAT solver amb F_P com a entrada
4. Convertirem el model que ens doni el SAT solver en la solució de P

Main de SAT-alumnes.cpp

```cpp
int main(){
  readClauses(); // reads numVars, numClauses and clauses
  model.resize(numVars+1,UNDEF);
  indexOfNextLitToPropagate = 0;
  decisionLevel = 0;

  // Take care of initial unit clauses, if any
  for (uint i = 0; i < numClauses; ++i)
    if (clauses[i].size() == 1) {
      int lit = clauses[i][0];
      int val = currentValueInModel(lit);
      if (val == FALSE) {cout << "UNSATISFIABLE" << endl; return 10;}
      else if (val == UNDEF) setLiteralToTrue(lit);
    }

  // DPLL algorithm
  while (true) {
    while ( propagateGivesConflict() ) {
      if ( decisionLevel == 0) { cout << "UNSATISFIABLE" << endl; return 10; }
      backtrack();
    }
    int decisionLit = getNextDecisionLiteral();
    if (decisionLit == 0) { checkmodel(); cout << "SATISFIABLE" << endl; return 20; }
    // start new decision level:
    modelStack.push_back(0);  // push mark indicating new DL
    ++indexOfNextLitToPropagate;
    ++decisionLevel;
    setLiteralToTrue(decisionLit);    // now push decisionLit on top of the mark
  }
}
```

SAT solver SAT-alumnes.cpp

Feina a fer en aquesta pràctica:

1. Entendre la implementació
2. Millorar `propagateGivesConflict()` (la unit propagation) fent servir **OccurLists**, llistes on s'indica en quines clàusules hi apareix negat cada literal.
3. Millorar `getNextDecisionLiteral()` perquè decideixi primer sobre variables que poden tenir més impacte. Per exemple, aquelles que
    - apareguin més sovint en les clàusules inicials
    - provoquin més conflictes
    - compleixin altres criteris
      (més propagacions, més backtracking)

Millores

Algunes heurístiques per `getNextDecisionLiteral()`:

1. Per cada variable, tenir un comptador d'aparicions en les clàusules inicials. Triar la variable indefinida amb el comptador més alt.
2. Tenir un comptador per variable x que compti quantes vegades ha aparegut x en un conflicte:
    - Cada cop que es detecta un conflicte en una clàusula, s'incrementen els comptadors de les variables de la clàusula.
    - Donar més pes als conflictes recents. Per exemple, dividir tots els comptadors per 2 cada N conflictes. (Proveu valors alts de N com 50.000, 100.000,...)