E.A.6.6 (Edge Colouring)

1.1 Modellazione

Dato un grafo non diretto $G = (V, E)$ siano

- $\mathcal{C} = \{1, 2, 3\}$
- $X = \left\{X^c_{u,v} \mid (u, v) \in E \land c \in \mathcal{C} \land u < v\right\}$ l'insieme di variabili t.c.
  - $X^c_{u,v}$ è vera se l'arco $(u, v) \in E$ ha colore c

$$\phi = \phi_{\text{almeno\_un\_colore\_per\_arco}} \land \phi_{\text{al\_più\_un\_colore\_per\_arco}} \land \phi_{\text{triangoli}}$$

$$\phi_{\text{almeno\_un\_colore\_per\_arco}} = \bigwedge_{\substack{(u,v)\in E \\ u<v}} \left(\bigvee_{c\in\mathcal{C}} X^c_{u,v} \right)$$

$$\phi_{\text{al\_più\_un\_colore\_per\_arco}} = \bigwedge_{\substack{(u,v)\in E \\ c_1,c_2\in\mathcal{C} \\ u<v \\ c_1<c_2}} \left(X^{c_1}_{u,v} \to \neg X^{c_2}_{u,v} \right)$$

$$\phi_{\text{triangoli}} = \bigwedge_{\substack{u,v,w\in V \\ (u,v),(v,w),(u,w)\in E \\ c\in\mathcal{C} \\ u<v<w}} \left(X^c_{u,v} \land X^c_{v,w} \to \neg X^c_{u,w} \right)$$

1.2 Istanziazione

1.2.1 Variabili

Dato $G = (V, E)$ come in Figura 1.1 si ha

- $X = \Big\{$
  $X^1_{\mathrm{A,E}}, X^2_{\mathrm{A,E}}, X^3_{\mathrm{A,E}}, X^1_{\mathrm{A,H}}, X^2_{\mathrm{A,H}}, X^3_{\mathrm{A,H}},$
  $X^1_{\mathrm{A,I}}, X^2_{\mathrm{A,I}}, X^3_{\mathrm{A,I}}, X^1_{\mathrm{A,S}}, X^2_{\mathrm{A,S}}, X^3_{\mathrm{A,S}},$
  $X^1_{\mathrm{B,C}}, X^2_{\mathrm{B,C}}, X^3_{\mathrm{B,C}}, X^1_{\mathrm{B,\ G2}}, X^2_{\mathrm{B,\ G2}}, X^3_{\mathrm{B,\ G2}},$
  $X^1_{\mathrm{B,I}}, X^2_{\mathrm{B,I}}, X^3_{\mathrm{B,I}}, X^1_{\mathrm{B,J}}, X^2_{\mathrm{B,J}}, X^3_{\mathrm{B,J}},$
  $X^1_{\mathrm{B,S}}, X^2_{\mathrm{B,S}}, X^3_{\mathrm{B,S}}, X^1_{\mathrm{C,D}}, X^2_{\mathrm{C,D}}, X^3_{\mathrm{C,D}},$
  $X^1_{\mathrm{C,\ G2}}, X^2_{\mathrm{C,\ G2}}, X^3_{\mathrm{C,\ G2}}, X^1_{\mathrm{C,S}}, X^2_{\mathrm{C,S}}, X^3_{\mathrm{C,S}},$
  $X^1_{\mathrm{D,E}}, X^2_{\mathrm{D,E}}, X^3_{\mathrm{D,E}}, X^1_{\mathrm{D,S}}, X^2_{\mathrm{D,S}}, X^3_{\mathrm{D,S}},$
  $X^1_{\mathrm{E,\ G1}}, X^2_{\mathrm{E,\ G1}}, X^3_{\mathrm{E,\ G1}}, X^1_{\mathrm{E,H}}, X^2_{\mathrm{E,H}}, X^3_{\mathrm{E,H}},$
  $X^1_{\mathrm{G1,H}}, X^2_{\mathrm{G1,H}}, X^3_{\mathrm{G1,H}}, X^1_{\mathrm{G2,J}}, X^2_{\mathrm{G2,J}}, X^3_{\mathrm{G2,J}},$
  $X^1_{\mathrm{H,I}}, X^2_{\mathrm{H,I}}, X^3_{\mathrm{H,I}},$
  $\Big\}$

1.2.2 Vincoli

$$
\begin{aligned}
\varphi_{\text{almeno\_un\_colore\_per\_arco}} = \Big\{ & \\
& (X^1_{\mathrm{A,E}} \vee X^2_{\mathrm{A,E}} \vee X^3_{\mathrm{A,E}}) \wedge (X^1_{\mathrm{A,H}} \vee X^2_{\mathrm{A,H}} \vee X^3_{\mathrm{A,H}}) \wedge \\
& (X^1_{\mathrm{A,I}} \vee X^2_{\mathrm{A,I}} \vee X^3_{\mathrm{A,I}}) \wedge (X^1_{\mathrm{A,S}} \vee X^2_{\mathrm{A,S}} \vee X^3_{\mathrm{A,S}}) \wedge \\
& (X^1_{\mathrm{B,C}} \vee X^2_{\mathrm{B,C}} \vee X^3_{\mathrm{B,C}}) \wedge (X^1_{\mathrm{B,\ G2}} \vee X^2_{\mathrm{B,\ G2}} \vee X^3_{\mathrm{B,\ G2}}) \wedge \\
& (X^1_{\mathrm{B,I}} \vee X^2_{\mathrm{B,I}} \vee X^3_{\mathrm{B,I}}) \wedge (X^1_{\mathrm{B,J}} \vee X^2_{\mathrm{B,J}} \vee X^3_{\mathrm{B,J}}) \wedge \\
& (X^1_{\mathrm{B,S}} \vee X^2_{\mathrm{B,S}} \vee X^3_{\mathrm{B,S}}) \wedge (X^1_{\mathrm{C,D}} \vee X^2_{\mathrm{C,D}} \vee X^3_{\mathrm{C,D}}) \wedge \\
& (X^1_{\mathrm{C,\ G2}} \vee X^2_{\mathrm{C,\ G2}} \vee X^3_{\mathrm{C,\ G2}}) \wedge (X^1_{\mathrm{C,S}} \vee X^2_{\mathrm{C,S}} \vee X^3_{\mathrm{C,S}}) \wedge \\
& (X^1_{\mathrm{D,E}} \vee X^2_{\mathrm{D,E}} \vee X^3_{\mathrm{D,E}}) \wedge (X^1_{\mathrm{D,S}} \vee X^2_{\mathrm{D,S}} \vee X^3_{\mathrm{D,S}}) \wedge \\
& (X^1_{\mathrm{E,\ G1}} \vee X^2_{\mathrm{E,\ G1}} \vee X^3_{\mathrm{E,\ G1}}) \wedge (X^1_{\mathrm{E,H}} \vee X^2_{\mathrm{E,H}} \vee X^3_{\mathrm{E,H}}) \wedge \\
& (X^1_{\mathrm{G1,H}} \vee X^2_{\mathrm{G1,H}} \vee X^3_{\mathrm{G1,H}}) \wedge (X^1_{\mathrm{G2,J}} \vee X^2_{\mathrm{G2,J}} \vee X^3_{\mathrm{G2,J}}) \wedge \\
& (X^1_{\mathrm{H,I}} \vee X^2_{\mathrm{H,I}} \vee X^3_{\mathrm{H,I}}) \\
\Big\} &
\end{aligned}
$$

$$
\begin{aligned}
&\varphi_{\text{al\_più\_un\_colore\_per\_arco}} = \Big\{ \\
&\quad \left(\neg X^1_{\text{A,E}} \vee \neg X^2_{\text{A,E}}\right) \wedge \left(\neg X^1_{\text{A,E}} \vee \neg X^3_{\text{A,E}}\right) \wedge \left(\neg X^2_{\text{A,E}} \vee \neg X^3_{\text{A,E}}\right) \wedge \\
&\quad \left(\neg X^1_{\text{A,H}} \vee \neg X^2_{\text{A,H}}\right) \wedge \left(\neg X^1_{\text{A,H}} \vee \neg X^3_{\text{A,H}}\right) \wedge \left(\neg X^2_{\text{A,H}} \vee \neg X^3_{\text{A,H}}\right) \wedge \\
&\quad \left(\neg X^1_{\text{A,I}} \vee \neg X^2_{\text{A,I}}\right) \wedge \left(\neg X^1_{\text{A,I}} \vee \neg X^3_{\text{A,I}}\right) \wedge \left(\neg X^2_{\text{A,I}} \vee \neg X^3_{\text{A,I}}\right) \wedge \\
&\quad \left(\neg X^1_{\text{A,S}} \vee \neg X^2_{\text{A,S}}\right) \wedge \left(\neg X^1_{\text{A,S}} \vee \neg X^3_{\text{A,S}}\right) \wedge \left(\neg X^2_{\text{A,S}} \vee \neg X^3_{\text{A,S}}\right) \wedge \\
&\quad \left(\neg X^1_{\text{B,C}} \vee \neg X^2_{\text{B,C}}\right) \wedge \left(\neg X^1_{\text{B,C}} \vee \neg X^3_{\text{B,C}}\right) \wedge \left(\neg X^2_{\text{B,C}} \vee \neg X^3_{\text{B,C}}\right) \wedge \\
&\quad \left(\neg X^1_{\text{B, G2}} \vee \neg X^2_{\text{B, G2}}\right) \wedge \left(\neg X^1_{\text{B, G2}} \vee \neg X^3_{\text{B, G2}}\right) \wedge \left(\neg X^2_{\text{B, G2}} \vee \neg X^3_{\text{B, G2}}\right) \wedge \\
&\quad \left(\neg X^1_{\text{B,I}} \vee \neg X^2_{\text{B,I}}\right) \wedge \left(\neg X^1_{\text{B,I}} \vee \neg X^3_{\text{B,I}}\right) \wedge \left(\neg X^2_{\text{B,I}} \vee \neg X^3_{\text{B,I}}\right) \wedge \\
&\quad \left(\neg X^1_{\text{B,J}} \vee \neg X^2_{\text{B,J}}\right) \wedge \left(\neg X^1_{\text{B,J}} \vee \neg X^3_{\text{B,J}}\right) \wedge \left(\neg X^2_{\text{B,J}} \vee \neg X^3_{\text{B,J}}\right) \wedge \\
&\quad \left(\neg X^1_{\text{B,S}} \vee \neg X^2_{\text{B,S}}\right) \wedge \left(\neg X^1_{\text{B,S}} \vee \neg X^3_{\text{B,S}}\right) \wedge \left(\neg X^2_{\text{B,S}} \vee \neg X^3_{\text{B,S}}\right) \wedge \\
&\quad \left(\neg X^1_{\text{C,D}} \vee \neg X^2_{\text{C,D}}\right) \wedge \left(\neg X^1_{\text{C,D}} \vee \neg X^3_{\text{C,D}}\right) \wedge \left(\neg X^2_{\text{C,D}} \vee \neg X^3_{\text{C,D}}\right) \wedge \\
&\quad \left(\neg X^1_{\text{C, G2}} \vee \neg X^2_{\text{C, G2}}\right) \wedge \left(\neg X^1_{\text{C, G2}} \vee \neg X^3_{\text{C, G2}}\right) \wedge \left(\neg X^2_{\text{C, G2}} \vee \neg X^3_{\text{C, G2}}\right) \wedge \\
&\quad \left(\neg X^1_{\text{C,S}} \vee \neg X^2_{\text{C,S}}\right) \wedge \left(\neg X^1_{\text{C,S}} \vee \neg X^3_{\text{C,S}}\right) \wedge \left(\neg X^2_{\text{C,S}} \vee \neg X^3_{\text{C,S}}\right) \wedge \\
&\quad \left(\neg X^1_{\text{D,E}} \vee \neg X^2_{\text{D,E}}\right) \wedge \left(\neg X^1_{\text{D,E}} \vee \neg X^3_{\text{D,E}}\right) \wedge \left(\neg X^2_{\text{D,E}} \vee \neg X^3_{\text{D,E}}\right) \wedge \\
&\quad \left(\neg X^1_{\text{D,S}} \vee \neg X^2_{\text{D,S}}\right) \wedge \left(\neg X^1_{\text{D,S}} \vee \neg X^3_{\text{D,S}}\right) \wedge \left(\neg X^2_{\text{D,S}} \vee \neg X^3_{\text{D,S}}\right) \wedge \\
&\quad \left(\neg X^1_{\text{E, G1}} \vee \neg X^2_{\text{E, G1}}\right) \wedge \left(\neg X^1_{\text{E, G1}} \vee \neg X^3_{\text{E, G1}}\right) \wedge \left(\neg X^2_{\text{E, G1}} \vee \neg X^3_{\text{E, G1}}\right) \wedge \\
&\quad \left(\neg X^1_{\text{E,H}} \vee \neg X^2_{\text{E,H}}\right) \wedge \left(\neg X^1_{\text{E,H}} \vee \neg X^3_{\text{E,H}}\right) \wedge \left(\neg X^2_{\text{E,H}} \vee \neg X^3_{\text{E,H}}\right) \wedge \\
&\quad \left(\neg X^1_{\text{G1,H}} \vee \neg X^2_{\text{G1,H}}\right) \wedge \left(\neg X^1_{\text{G1,H}} \vee \neg X^3_{\text{G1,H}}\right) \wedge \left(\neg X^2_{\text{G1,H}} \vee \neg X^3_{\text{G1,H}}\right) \wedge \\
&\quad \left(\neg X^1_{\text{G2,J}} \vee \neg X^2_{\text{G2,J}}\right) \wedge \left(\neg X^1_{\text{G2,J}} \vee \neg X^3_{\text{G2,J}}\right) \wedge \left(\neg X^2_{\text{G2,J}} \vee \neg X^3_{\text{G2,J}}\right) \wedge \\
&\quad \left(\neg X^1_{\text{H,I}} \vee \neg X^2_{\text{H,I}}\right) \wedge \left(\neg X^1_{\text{H,I}} \vee \neg X^3_{\text{H,I}}\right) \wedge \left(\neg X^2_{\text{H,I}} \vee \neg X^3_{\text{H,I}}\right) \\
&\Big\}
\end{aligned}
$$

$$
\begin{aligned}
&\varphi_{\text{triangoli}} = \Big\{ \\
&\quad \left(\neg X^1_{\text{A,E}} \vee \neg X^1_{\text{E,H}} \vee \neg X^1_{\text{A,H}}\right) \wedge \left(\neg X^2_{\text{A,E}} \vee \neg X^2_{\text{E,H}} \vee \neg X^2_{\text{A,H}}\right) \wedge \\
&\quad \left(\neg X^3_{\text{A,E}} \vee \neg X^3_{\text{E,H}} \vee \neg X^3_{\text{A,H}}\right) \wedge \left(\neg X^1_{\text{A,H}} \vee \neg X^1_{\text{H,I}} \vee \neg X^1_{\text{A,I}}\right) \wedge \\
&\quad \left(\neg X^2_{\text{A,H}} \vee \neg X^2_{\text{H,I}} \vee \neg X^2_{\text{A,I}}\right) \wedge \left(\neg X^3_{\text{A,H}} \vee \neg X^3_{\text{H,I}} \vee \neg X^3_{\text{A,I}}\right) \wedge \\
&\quad \left(\neg X^1_{\text{B,C}} \vee \neg X^1_{\text{C, G2}} \vee \neg X^1_{\text{B, G2}}\right) \wedge \left(\neg X^2_{\text{B,C}} \vee \neg X^2_{\text{C, G2}} \vee \neg X^2_{\text{B, G2}}\right) \wedge
\end{aligned}
$$

$$
\begin{aligned}
&\left(\neg X^{3}_{\mathrm{B,C}} \vee \neg X^{3}_{\mathrm{C,\ G2}} \vee \neg X^{3}_{\mathrm{B,\ G2}}\right) \wedge \left(\neg X^{1}_{\mathrm{B,C}} \vee \neg X^{1}_{\mathrm{C,S}} \vee \neg X^{1}_{\mathrm{B,S}}\right) \wedge \\
&\left(\neg X^{2}_{\mathrm{B,C}} \vee \neg X^{2}_{\mathrm{C,S}} \vee \neg X^{2}_{\mathrm{B,S}}\right) \wedge \left(\neg X^{3}_{\mathrm{B,C}} \vee \neg X^{3}_{\mathrm{C,S}} \vee \neg X^{3}_{\mathrm{B,S}}\right) \wedge \\
&\left(\neg X^{1}_{\mathrm{B,\ G2}} \vee \neg X^{1}_{\mathrm{G2,J}} \vee \neg X^{1}_{\mathrm{B,J}}\right) \wedge \left(\neg X^{2}_{\mathrm{B,\ G2}} \vee \neg X^{2}_{\mathrm{G2,J}} \vee \neg X^{2}_{\mathrm{B,J}}\right) \wedge \\
&\left(\neg X^{3}_{\mathrm{B,\ G2}} \vee \neg X^{3}_{\mathrm{G2,J}} \vee \neg X^{3}_{\mathrm{B,J}}\right) \wedge \left(\neg X^{1}_{\mathrm{C,D}} \vee \neg X^{1}_{\mathrm{D,S}} \vee \neg X^{1}_{\mathrm{C,S}}\right) \wedge \\
&\left(\neg X^{2}_{\mathrm{C,D}} \vee \neg X^{2}_{\mathrm{D,S}} \vee \neg X^{2}_{\mathrm{C,S}}\right) \wedge \left(\neg X^{3}_{\mathrm{C,D}} \vee \neg X^{3}_{\mathrm{D,S}} \vee \neg X^{3}_{\mathrm{C,S}}\right) \wedge \\
&\left(\neg X^{1}_{\mathrm{E,\ G1}} \vee \neg X^{1}_{\mathrm{G1,H}} \vee \neg X^{1}_{\mathrm{E,H}}\right) \wedge \left(\neg X^{2}_{\mathrm{E,\ G1}} \vee \neg X^{2}_{\mathrm{G1,H}} \vee \neg X^{2}_{\mathrm{E,H}}\right) \wedge \\
&\left(\neg X^{3}_{\mathrm{E,\ G1}} \vee \neg X^{3}_{\mathrm{G1,H}} \vee \neg X^{3}_{\mathrm{E,H}}\right) \wedge \\
\}
\end{aligned}
$$

1.3 Codifica in SATCodec

1.3.1 EdgeColouringToSAT

```
import it.uniroma1.di.tmancini.utils.*;
import it.uniroma1.di.tmancini.teaching.ai.SATCodec.*;
import java.util.*;

public class EdgeColouringToSAT {
  public static void main(String args[]) {

    var AEdges = new ArrayList<Integer>();
    AEdges.add(4);
    AEdges.add(7);
    AEdges.add(8);
    AEdges.add(10);

    var BEdges = new ArrayList<Integer>();
    BEdges.add(2);
    BEdges.add(6);
    BEdges.add(8);
    BEdges.add(9);
    BEdges.add(10);

    var CEdges = new ArrayList<Integer>();
    CEdges.add(3);
    CEdges.add(6);
    CEdges.add(10);

    var DEdges = new ArrayList<Integer>();
    DEdges.add(4);
    DEdges.add(10);

    var EEdges = new ArrayList<Integer>();
    EEdges.add(5);
    EEdges.add(7);

    var G1Edges = new ArrayList<Integer>();
    G1Edges.add(7);

    var G2Edges = new ArrayList<Integer>();
    G2Edges.add(9);

    var HEdges = new ArrayList<Integer>();
    HEdges.add(8);

    var IEdges = new ArrayList<Integer>();
    var JEdges = new ArrayList<Integer>();
    var SEdges = new ArrayList<Integer>();

    var edges = new ArrayList<ArrayList<Integer>>();
    edges.add(AEdges);
    edges.add(BEdges);
```

```
    edges.add(CEdges);
    edges.add(DEdges);
    edges.add(EEdges);
    edges.add(G1Edges);
    edges.add(G2Edges);
    edges.add(HEdges);
    edges.add(IEdges);
    edges.add(JEdges);
    edges.add(SEdges);

    // A = 0
    // B = 1
    // C = 2
    // D = 3
    // E = 4
    // G1 = 5
    // G2 = 6
    // H = 7
    // I = 8
    // J = 9
    // S = 10

    var colorsRange = new IntRange("colors", 1, 3);
    var edgesRange = new IntRange("edges", 0, 10);

    var encoder = new SATEncoder("EdgeColuring", "edge-
colouring.cnf");
    encoder.defineFamilyOfVariables("X", edgesRange, edgesRange,
colorsRange);

    // Almeno un colore per arco
    for (Integer u : edgesRange.values()) {
      for (Integer v : edges.get(u)) {
        if (u < v) {
          for (Integer c : colorsRange.values()) {
            encoder.addToClause("X", u, v, c);
          }
          encoder.endClause();
        }
      }
    }

    // Al più un colore per arco
    for (Integer u : edgesRange.values()) {
      for (Integer v : edges.get(u)) {
        if (u < v) {
          for (Integer c1 : colorsRange.values()) {
            for (int c2 = c1 + 1; c2 ≤ 3; c2++) {
              encoder.addNegToClause("X", u, v, c1);
              encoder.addNegToClause("X", u, v, c2);
              encoder.endClause();
            }
          }
```

```
        }
      }
    }

    // Triangoli
    for (Integer u : edgesRange.values()) {
      for (Integer v : edgesRange.values()) {
        for (Integer w : edgesRange.values()) {
          if (u < v && v < w && edges.get(u).contains(v) &&
edges.get(v).contains(w)
              && edges.get(u).contains(w)) {
            for (Integer c : colorsRange.values()) {
              encoder.addNegToClause("X", u, v, c);
              encoder.addNegToClause("X", v, w, c);
              encoder.addNegToClause("X", u, w, c);
              encoder.endClause();
            }
          }
        }
      }
    }

    encoder.end();
  }
}
```

1.3.2 SATToEdgeColouring

```
import it.uniroma1.di.tmancini.utils.*;
import it.uniroma1.di.tmancini.teaching.ai.SATCodec.*;
import java.util.*;

public class SATToEdgeColouring {
  public static void main(String args[]) throws
java.io.IOException, java.io.FileNotFoundException {
    String[] variables = { "A", "B", "C", "D", "E", "G1", "G2",
"H", "I", "J", "S" };

    var decoder = new SATModelDecoder(args);
    decoder.run();

    int maxVar = decoder.getMaxVar();
    for (int i = 1; i ≤ maxVar; i++) {
      Boolean v_i = decoder.getModelValue(i);
      if (v_i == null || !v_i)
        continue;

      SATModelDecoder.Var variable = decoder.decodeVariable(i);
      int u = variable.getIndices().get(0);
      int v = variable.getIndices().get(1);
      int c = variable.getIndices().get(2);
      System.out.println(variables[u] + " → " + variables[v] +
" : " + c);
```

```
    }
    System.out.println("-- STOP HERE");
  }
}
```

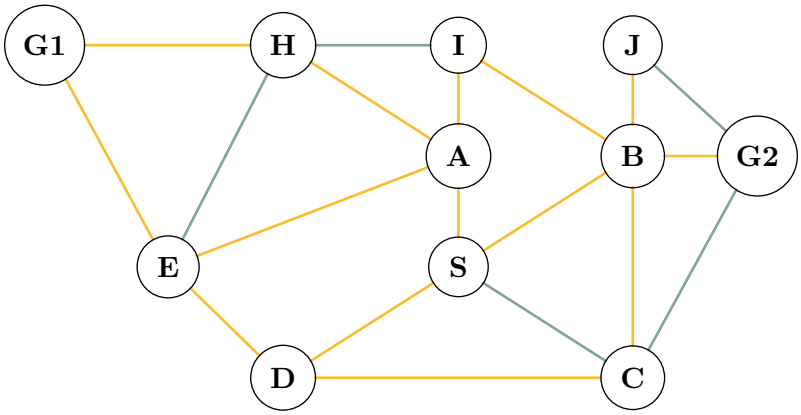

Figura 1: soluzione generata da `picosat`