E.A.6.7 (HC-VIP)

1.1 Modellazione

Dati i parametri $I, V, G = (I, \text{bus})$ t.c.
- $casa \in I$
- $V \subseteq I / \{casa\}$
- $bus \subseteq I \times I$
- $|V| \leq \frac{|I|}{2}$

Si definiscono le seguenti variabili:
- $T = |I|$ è il numero di *bus* da prendere in un percorso che parte da *casa*, visita tutti gli indirizzi esattamente una volta e ritorna a *casa*
  - servono $|I| - 1$ *bus* per visitare tutti gli indirizzi esattamente una volta più 1 *bus* per tornare a casa a fine giornata
- $\mathcal{T} = \{1, ..., T\}$
- $\mathcal{I} = \{1, ..., |I|\}$ l'insieme di identificatori per cui esiste una funzione id biettiva t.c.

$$\text{id} : \mathcal{I} \to I \quad \text{e} \quad \text{id}(1) = casa \tag{1}$$

- $\mathcal{V} = \{i \mid i \in \mathcal{I} \land \exists v \; v \in V \land \text{id}(i) = v\}$
- $X = \left\{X^t_{i,j} \mid (i,j) \in bus \land t \in \mathcal{T}\right\}$ l'insieme di variabili dove
  - $X^t_{i,j}$ è vera se l'arco $(i,j) \in bus$ è stato percorso al t-esimo passo

$$\begin{aligned}\phi = \; & \phi_{\text{almeno\_un\_arco\_per\_passo}} \land \\ & \phi_{\text{al\_più\_un\_arco\_per\_passo}} \land \\ & \phi_{\text{almeno\_un\_arco\_per\_indirizzo}} \land \\ & \phi_{\text{al\_più\_un\_arco\_per\_indirizzo}} \land \\ & \phi_{\text{clienti\_VIP\_nella\_prima\_metà}} \land \\ & \phi_{\text{partenza\_da\_casa}} \land \\ & \phi_{\text{arrivo\_a\_casa}} \land \\ & \phi_{\text{percorso\_valido}}\end{aligned} \tag{2}$$

$$\begin{aligned}\phi_{\text{almeno\_un\_arco\_per\_passo}} &= \bigwedge_{t \in \mathcal{T}} \bigvee_{(i,j) \in \, bus} X^t_{i,j} \\ \phi_{\text{al\_più\_un\_arco\_per\_passo}} &= \bigwedge_{\substack{t \in \mathcal{T} \\ (i_1,j_1),(i_2,j_2) \in \, bus \\ (i_1,j_1) < (i_2,j_2)}} X^t_{i_1,j_1} \to \neg X^t_{i_2,j_2}\end{aligned} \tag{3}$$

$$
\begin{aligned}
\phi_{\text{almeno\_un\_arco\_per\_indirizzo}} &= \bigwedge_{j \in \mathcal{I}} \bigvee_{\substack{t \in \mathcal{T} \\ (i,j) \in \, bus}} X^t_{i,j} \\
\phi_{\text{al\_più\_un\_arco\_per\_indirizzo}} &= \bigwedge_{\substack{t_1, t_2 \in \mathcal{T} \\ (i_1,j),(i_2,j) \in \, bus \\ t_1 \leq t_2 \\ i_1 < i_2}} X^{t_1}_{i_1,j} \to \neg X^{t_2}_{i_2,j} \\
\phi_{\text{clienti\_VIP\_nella\_prima\_metà}} &= \bigwedge_{v \in \mathcal{V}} \bigvee_{\substack{t \in \mathcal{T} \\ t \leq \lceil \frac{T}{2} \rceil \\ (i,v) \in \, bus}} X^t_{i,v} \\
\phi_{\text{partenza\_da\_casa}} &= \bigvee_{\substack{i \in \mathcal{I} \setminus \{1\} \\ (1,i) \in \, bus}} X^1_{1,i} \\
\phi_{\text{arrivo\_a\_casa}} &= \bigvee_{\substack{i \in \mathcal{I} \setminus \{1\} \\ (i,1) \in \, bus}} X^T_{i,1} \\
\phi_{\text{percorso\_valido}} &= \bigwedge_{\substack{t \in \mathcal{T} \setminus \{T\} \\ (i,j) \in \, bus}} X^t_{i,j} \to \bigvee_{(j,k) \in \, bus} X^{t+1}_{j,k}
\end{aligned}
\tag{3}
$$

1.2 Istanziazione

1.2.1 Variabili

- $I = \{casa, i_1, i_2, i_3, i_4, i_5\}$
- $V = \{i_1\}$
- $bus = \{(casa, i_1), (casa, i_2), (i_1, i_3), (i_1, i_4), (i_2, casa), (i_3, casa), (i_3, i_4), (i_4, i_1), (i_4, i_2)\}$
- $\mathcal{T} = \{1, 2, 3, 4, 5\}$
- $\mathcal{I} = \{1, 2, 3, 4, 5\}$
- $\mathcal{V} = \{2\}$
- $X = \Big\{$

$$
\begin{gathered}
X^1_{1,2} X^1_{1,3} X^1_{2,4} X^1_{2,5} X^1_{3,1} \\
X^1_{3,5} X^1_{4,1} X^1_{4,5} X^1_{5,2} X^1_{5,3} \\
X^2_{1,2} X^2_{1,3} X^2_{2,4} X^2_{2,5} X^2_{3,1} \\
X^2_{3,5} X^2_{4,1} X^2_{4,5} X^2_{5,2} X^2_{5,3} \\
X^3_{1,2} X^3_{1,3} X^3_{2,4} X^3_{2,5} X^3_{3,1} \\
X^3_{3,5} X^3_{4,1} X^3_{4,5} X^3_{5,2} X^3_{5,3} \\
X^4_{1,2} X^4_{1,3} X^4_{2,4} X^4_{2,5} X^4_{3,1} \\
X^4_{3,5} X^4_{4,1} X^4_{4,5} X^4_{5,2} X^4_{5,3} \\
X^5_{1,2} X^5_{1,3} X^5_{2,4} X^5_{2,5} X^5_{3,1} \\
X^5_{3,5} X^5_{4,1} X^5_{4,5} X^5_{5,2} X^5_{5,3}
\end{gathered}
$$

$\Big\}$

1.2.2 Vincoli

La codifica che ho generato ha 293 clausole anche per un problema così piccolo… non vale la pena elencarle tutte, metto un vincolo d'esempio.

$$
\begin{aligned}
&\phi_{\text{al\_più\_un\_arco\_per\_passo}} = \Big\{ \\
&\left(X^1_{1,2} \to \neg X^1_{1,3}\right) \land \left(X^1_{1,2} \to \neg X^1_{2,4}\right) \land \left(X^1_{1,2} \to \neg X^1_{2,5}\right) \land \left(X^1_{1,2} \to \neg X^1_{3,1}\right) \land \left(X^1_{1,2} \to \neg X^1_{3,5}\right) \land \\
&\left(X^1_{1,2} \to \neg X^1_{4,1}\right) \land \left(X^1_{1,2} \to \neg X^1_{4,5}\right) \land \left(X^1_{1,2} \to \neg X^1_{5,2}\right) \land \left(X^1_{1,2} \to \neg X^1_{5,3}\right) \land \left(X^1_{1,3} \to \neg X^1_{2,4}\right) \land \\
&\left(X^1_{1,3} \to \neg X^1_{2,5}\right) \land \left(X^1_{1,3} \to \neg X^1_{3,1}\right) \land \left(X^1_{1,3} \to \neg X^1_{3,5}\right) \land \left(X^1_{1,3} \to \neg X^1_{4,1}\right) \land \left(X^1_{1,3} \to \neg X^1_{4,5}\right) \land \quad (4) \\
&\left(X^1_{1,3} \to \neg X^1_{5,2}\right) \land \left(X^1_{1,3} \to \neg X^1_{5,3}\right) \land \left(X^1_{2,4} \to \neg X^1_{2,5}\right) \land \left(X^1_{2,4} \to \neg X^1_{3,1}\right) \land \left(X^1_{2,4} \to \neg X^1_{3,5}\right) \land \\
&\left(X^1_{2,4} \to \neg X^1_{4,1}\right) \land \left(X^1_{2,4} \to \neg X^1_{4,5}\right) \land \left(X^1_{2,4} \to \neg X^1_{5,2}\right) \land \left(X^1_{2,4} \to \neg X^1_{5,3}\right) \land \left(X^1_{2,5} \to \neg X^1_{3,1}\right) \land \\
&\left(X^1_{2,5} \to \neg X^1_{3,5}\right) \land \left(X^1_{2,5} \to \neg X^1_{4,1}\right) \land \left(X^1_{2,5} \to \neg X^1_{4,5}\right) \land \left(X^1_{2,5} \to \neg X^1_{5,2}\right) \land \left(X^1_{2,5} \to \neg X^1_{5,3}\right) \land
\end{aligned}
$$

$$
\begin{aligned}
&\left(X^1_{3,1} \to \neg X^1_{3,5}\right) \wedge \left(X^1_{3,1} \to \neg X^1_{4,1}\right) \wedge \left(X^1_{3,1} \to \neg X^1_{4,5}\right) \wedge \left(X^1_{3,1} \to \neg X^1_{5,2}\right) \wedge \left(X^1_{3,1} \to \neg X^1_{5,3}\right) \wedge \\
&\left(X^1_{3,5} \to \neg X^1_{4,1}\right) \wedge \left(X^1_{3,5} \to \neg X^1_{4,5}\right) \wedge \left(X^1_{3,5} \to \neg X^1_{5,2}\right) \wedge \left(X^1_{3,5} \to \neg X^1_{5,3}\right) \wedge \left(X^1_{4,1} \to \neg X^1_{4,5}\right) \wedge \\
&\left(X^1_{4,1} \to \neg X^1_{5,2}\right) \wedge \left(X^1_{4,1} \to \neg X^1_{5,3}\right) \wedge \left(X^1_{4,5} \to \neg X^1_{5,2}\right) \wedge \left(X^1_{4,5} \to \neg X^1_{5,3}\right) \wedge \left(X^1_{5,2} \to \neg X^1_{5,3}\right) \wedge \\
&\left(X^2_{1,2} \to \neg X^2_{1,3}\right) \wedge \left(X^2_{1,2} \to \neg X^2_{2,4}\right) \wedge \left(X^2_{1,2} \to \neg X^2_{2,5}\right) \wedge \left(X^2_{1,2} \to \neg X^2_{3,1}\right) \wedge \left(X^2_{1,2} \to \neg X^2_{3,5}\right) \wedge \\
&\left(X^2_{1,2} \to \neg X^2_{4,1}\right) \wedge \left(X^2_{1,2} \to \neg X^2_{4,5}\right) \wedge \left(X^2_{1,2} \to \neg X^2_{5,2}\right) \wedge \left(X^2_{1,2} \to \neg X^2_{5,3}\right) \wedge \left(X^2_{1,3} \to \neg X^2_{2,4}\right) \wedge \\
&\left(X^2_{1,3} \to \neg X^2_{2,5}\right) \wedge \left(X^2_{1,3} \to \neg X^2_{3,1}\right) \wedge \left(X^2_{1,3} \to \neg X^2_{3,5}\right) \wedge \left(X^2_{1,3} \to \neg X^2_{4,1}\right) \wedge \left(X^2_{1,3} \to \neg X^2_{4,5}\right) \wedge \\
&\left(X^2_{1,3} \to \neg X^2_{5,2}\right) \wedge \left(X^2_{1,3} \to \neg X^2_{5,3}\right) \wedge \left(X^2_{2,4} \to \neg X^2_{2,5}\right) \wedge \left(X^2_{2,4} \to \neg X^2_{3,1}\right) \wedge \left(X^2_{2,4} \to \neg X^2_{3,5}\right) \wedge \\
&\left(X^2_{2,4} \to \neg X^2_{4,1}\right) \wedge \left(X^2_{2,4} \to \neg X^2_{4,5}\right) \wedge \left(X^2_{2,4} \to \neg X^2_{5,2}\right) \wedge \left(X^2_{2,4} \to \neg X^2_{5,3}\right) \wedge \left(X^2_{2,5} \to \neg X^2_{3,1}\right) \wedge \\
&\left(X^2_{2,5} \to \neg X^2_{3,5}\right) \wedge \left(X^2_{2,5} \to \neg X^2_{4,1}\right) \wedge \left(X^2_{2,5} \to \neg X^2_{4,5}\right) \wedge \left(X^2_{2,5} \to \neg X^2_{5,2}\right) \wedge \left(X^2_{2,5} \to \neg X^2_{5,3}\right) \wedge \\
&\left(X^2_{3,1} \to \neg X^2_{3,5}\right) \wedge \left(X^2_{3,1} \to \neg X^2_{4,1}\right) \wedge \left(X^2_{3,1} \to \neg X^2_{4,5}\right) \wedge \left(X^2_{3,1} \to \neg X^2_{5,2}\right) \wedge \left(X^2_{3,1} \to \neg X^2_{5,3}\right) \wedge \\
&\left(X^2_{3,5} \to \neg X^2_{4,1}\right) \wedge \left(X^2_{3,5} \to \neg X^2_{4,5}\right) \wedge \left(X^2_{3,5} \to \neg X^2_{5,2}\right) \wedge \left(X^2_{3,5} \to \neg X^2_{5,3}\right) \wedge \left(X^2_{4,1} \to \neg X^2_{4,5}\right) \wedge \\
&\left(X^2_{4,1} \to \neg X^2_{5,2}\right) \wedge \left(X^2_{4,1} \to \neg X^2_{5,3}\right) \wedge \left(X^2_{4,5} \to \neg X^2_{5,2}\right) \wedge \left(X^2_{4,5} \to \neg X^2_{5,3}\right) \wedge \left(X^2_{5,2} \to \neg X^2_{5,3}\right) \wedge \\
&\left(X^3_{1,2} \to \neg X^3_{1,3}\right) \wedge \left(X^3_{1,2} \to \neg X^3_{2,4}\right) \wedge \left(X^3_{1,2} \to \neg X^3_{2,5}\right) \wedge \left(X^3_{1,2} \to \neg X^3_{3,1}\right) \wedge \left(X^3_{1,2} \to \neg X^3_{3,5}\right) \wedge \\
&\left(X^3_{1,2} \to \neg X^3_{4,1}\right) \wedge \left(X^3_{1,2} \to \neg X^3_{4,5}\right) \wedge \left(X^3_{1,2} \to \neg X^3_{5,2}\right) \wedge \left(X^3_{1,2} \to \neg X^3_{5,3}\right) \wedge \left(X^3_{1,3} \to \neg X^3_{2,4}\right) \wedge \\
&\left(X^3_{1,3} \to \neg X^3_{2,5}\right) \wedge \left(X^3_{1,3} \to \neg X^3_{3,1}\right) \wedge \left(X^3_{1,3} \to \neg X^3_{3,5}\right) \wedge \left(X^3_{1,3} \to \neg X^3_{4,1}\right) \wedge \left(X^3_{1,3} \to \neg X^3_{4,5}\right) \wedge \\
&\left(X^3_{1,3} \to \neg X^3_{5,2}\right) \wedge \left(X^3_{1,3} \to \neg X^3_{5,3}\right) \wedge \left(X^3_{2,4} \to \neg X^3_{2,5}\right) \wedge \left(X^3_{2,4} \to \neg X^3_{3,1}\right) \wedge \left(X^3_{2,4} \to \neg X^3_{3,5}\right) \wedge \\
&\left(X^3_{2,4} \to \neg X^3_{4,1}\right) \wedge \left(X^3_{2,4} \to \neg X^3_{4,5}\right) \wedge \left(X^3_{2,4} \to \neg X^3_{5,2}\right) \wedge \left(X^3_{2,4} \to \neg X^3_{5,3}\right) \wedge \left(X^3_{2,5} \to \neg X^3_{3,1}\right) \wedge \\
&\left(X^3_{2,5} \to \neg X^3_{3,5}\right) \wedge \left(X^3_{2,5} \to \neg X^3_{4,1}\right) \wedge \left(X^3_{2,5} \to \neg X^3_{4,5}\right) \wedge \left(X^3_{2,5} \to \neg X^3_{5,2}\right) \wedge \left(X^3_{2,5} \to \neg X^3_{5,3}\right) \wedge \\
&\left(X^3_{3,1} \to \neg X^3_{3,5}\right) \wedge \left(X^3_{3,1} \to \neg X^3_{4,1}\right) \wedge \left(X^3_{3,1} \to \neg X^3_{4,5}\right) \wedge \left(X^3_{3,1} \to \neg X^3_{5,2}\right) \wedge \left(X^3_{3,1} \to \neg X^3_{5,3}\right) \wedge \\
&\left(X^3_{3,5} \to \neg X^3_{4,1}\right) \wedge \left(X^3_{3,5} \to \neg X^3_{4,5}\right) \wedge \left(X^3_{3,5} \to \neg X^3_{5,2}\right) \wedge \left(X^3_{3,5} \to \neg X^3_{5,3}\right) \wedge \left(X^3_{4,1} \to \neg X^3_{4,5}\right) \wedge \\
&\left(X^3_{4,1} \to \neg X^3_{5,2}\right) \wedge \left(X^3_{4,1} \to \neg X^3_{5,3}\right) \wedge \left(X^3_{4,5} \to \neg X^3_{5,2}\right) \wedge \left(X^3_{4,5} \to \neg X^3_{5,3}\right) \wedge \left(X^3_{5,2} \to \neg X^3_{5,3}\right) \wedge \\
&\left(X^4_{1,2} \to \neg X^4_{1,3}\right) \wedge \left(X^4_{1,2} \to \neg X^4_{2,4}\right) \wedge \left(X^4_{1,2} \to \neg X^4_{2,5}\right) \wedge \left(X^4_{1,2} \to \neg X^4_{3,1}\right) \wedge \left(X^4_{1,2} \to \neg X^4_{3,5}\right) \wedge \\
&\left(X^4_{1,2} \to \neg X^4_{4,1}\right) \wedge \left(X^4_{1,2} \to \neg X^4_{4,5}\right) \wedge \left(X^4_{1,2} \to \neg X^4_{5,2}\right) \wedge \left(X^4_{1,2} \to \neg X^4_{5,3}\right) \wedge \left(X^4_{1,3} \to \neg X^4_{2,4}\right) \wedge \\
&\left(X^4_{1,3} \to \neg X^4_{2,5}\right) \wedge \left(X^4_{1,3} \to \neg X^4_{3,1}\right) \wedge \left(X^4_{1,3} \to \neg X^4_{3,5}\right) \wedge \left(X^4_{1,3} \to \neg X^4_{4,1}\right) \wedge \left(X^4_{1,3} \to \neg X^4_{4,5}\right) \wedge \\
&\left(X^4_{1,3} \to \neg X^4_{5,2}\right) \wedge \left(X^4_{1,3} \to \neg X^4_{5,3}\right) \wedge \left(X^4_{2,4} \to \neg X^4_{2,5}\right) \wedge \left(X^4_{2,4} \to \neg X^4_{3,1}\right) \wedge \left(X^4_{2,4} \to \neg X^4_{3,5}\right) \wedge \\
&\left(X^4_{2,4} \to \neg X^4_{4,1}\right) \wedge \left(X^4_{2,4} \to \neg X^4_{4,5}\right) \wedge \left(X^4_{2,4} \to \neg X^4_{5,2}\right) \wedge \left(X^4_{2,4} \to \neg X^4_{5,3}\right) \wedge \left(X^4_{2,5} \to \neg X^4_{3,1}\right) \wedge \\
&\left(X^4_{2,5} \to \neg X^4_{3,5}\right) \wedge \left(X^4_{2,5} \to \neg X^4_{4,1}\right) \wedge \left(X^4_{2,5} \to \neg X^4_{4,5}\right) \wedge \left(X^4_{2,5} \to \neg X^4_{5,2}\right) \wedge \left(X^4_{2,5} \to \neg X^4_{5,3}\right) \wedge \\
&\left(X^4_{3,1} \to \neg X^4_{3,5}\right) \wedge \left(X^4_{3,1} \to \neg X^4_{4,1}\right) \wedge \left(X^4_{3,1} \to \neg X^4_{4,5}\right) \wedge \left(X^4_{3,1} \to \neg X^4_{5,2}\right) \wedge \left(X^4_{3,1} \to \neg X^4_{5,3}\right) \wedge \\
&\left(X^4_{3,5} \to \neg X^4_{4,1}\right) \wedge \left(X^4_{3,5} \to \neg X^4_{4,5}\right) \wedge \left(X^4_{3,5} \to \neg X^4_{5,2}\right) \wedge \left(X^4_{3,5} \to \neg X^4_{5,3}\right) \wedge \left(X^4_{4,1} \to \neg X^4_{4,5}\right) \wedge
\end{aligned} \tag{4}
$$

$$\left(X^{4}_{4,1} \to \neg X^{4}_{5,2}\right) \wedge \left(X^{4}_{4,1} \to \neg X^{4}_{5,3}\right) \wedge \left(X^{4}_{4,5} \to \neg X^{4}_{5,2}\right) \wedge \left(X^{4}_{4,5} \to \neg X^{4}_{5,3}\right) \wedge \left(X^{4}_{5,2} \to \neg X^{4}_{5,3}\right) \wedge$$

$$\left(X^{5}_{1,2} \to \neg X^{5}_{1,3}\right) \wedge \left(X^{5}_{1,2} \to \neg X^{5}_{2,4}\right) \wedge \left(X^{5}_{1,2} \to \neg X^{5}_{2,5}\right) \wedge \left(X^{5}_{1,2} \to \neg X^{5}_{3,1}\right) \wedge \left(X^{5}_{1,2} \to \neg X^{5}_{3,5}\right) \wedge$$

$$\left(X^{5}_{1,2} \to \neg X^{5}_{4,1}\right) \wedge \left(X^{5}_{1,2} \to \neg X^{5}_{4,5}\right) \wedge \left(X^{5}_{1,2} \to \neg X^{5}_{5,2}\right) \wedge \left(X^{5}_{1,2} \to \neg X^{5}_{5,3}\right) \wedge \left(X^{5}_{1,3} \to \neg X^{5}_{2,4}\right) \wedge$$

$$\left(X^{5}_{1,3} \to \neg X^{5}_{2,5}\right) \wedge \left(X^{5}_{1,3} \to \neg X^{5}_{3,1}\right) \wedge \left(X^{5}_{1,3} \to \neg X^{5}_{3,5}\right) \wedge \left(X^{5}_{1,3} \to \neg X^{5}_{4,1}\right) \wedge \left(X^{5}_{1,3} \to \neg X^{5}_{4,5}\right) \wedge$$

$$\left(X^{5}_{1,3} \to \neg X^{5}_{5,2}\right) \wedge \left(X^{5}_{1,3} \to \neg X^{5}_{5,3}\right) \wedge \left(X^{5}_{2,4} \to \neg X^{5}_{2,5}\right) \wedge \left(X^{5}_{2,4} \to \neg X^{5}_{3,1}\right) \wedge \left(X^{5}_{2,4} \to \neg X^{5}_{3,5}\right) \wedge$$

$$\left(X^{5}_{2,4} \to \neg X^{5}_{4,1}\right) \wedge \left(X^{5}_{2,4} \to \neg X^{5}_{4,5}\right) \wedge \left(X^{5}_{2,4} \to \neg X^{5}_{5,2}\right) \wedge \left(X^{5}_{2,4} \to \neg X^{5}_{5,3}\right) \wedge \left(X^{5}_{2,5} \to \neg X^{5}_{3,1}\right) \wedge \tag{4}$$

$$\left(X^{5}_{2,5} \to \neg X^{5}_{3,5}\right) \wedge \left(X^{5}_{2,5} \to \neg X^{5}_{4,1}\right) \wedge \left(X^{5}_{2,5} \to \neg X^{5}_{4,5}\right) \wedge \left(X^{5}_{2,5} \to \neg X^{5}_{5,2}\right) \wedge \left(X^{5}_{2,5} \to \neg X^{5}_{5,3}\right) \wedge$$

$$\left(X^{5}_{3,1} \to \neg X^{5}_{3,5}\right) \wedge \left(X^{5}_{3,1} \to \neg X^{5}_{4,1}\right) \wedge \left(X^{5}_{3,1} \to \neg X^{5}_{4,5}\right) \wedge \left(X^{5}_{3,1} \to \neg X^{5}_{5,2}\right) \wedge \left(X^{5}_{3,1} \to \neg X^{5}_{5,3}\right) \wedge$$

$$\left(X^{5}_{3,5} \to \neg X^{5}_{4,1}\right) \wedge \left(X^{5}_{3,5} \to \neg X^{5}_{4,5}\right) \wedge \left(X^{5}_{3,5} \to \neg X^{5}_{5,2}\right) \wedge \left(X^{5}_{3,5} \to \neg X^{5}_{5,3}\right) \wedge \left(X^{5}_{4,1} \to \neg X^{5}_{4,5}\right) \wedge$$

$$\left(X^{5}_{4,1} \to \neg X^{5}_{5,2}\right) \wedge \left(X^{5}_{4,1} \to \neg X^{5}_{5,3}\right) \wedge \left(X^{5}_{4,5} \to \neg X^{5}_{5,2}\right) \wedge \left(X^{5}_{4,5} \to \neg X^{5}_{5,3}\right) \wedge \left(X^{5}_{5,2} \to \neg X^{5}_{5,3}\right) \wedge$$

$$\}$$

1.3 Codifica SATCodec (a questo giro in `Rust`)

```rust
use std::collections::BTreeSet;

use computer_braining::framework::sat_codec::*;
use serde::Serialize;

#[derive(PartialEq, Eq, PartialOrd, Ord, Hash, Debug, Serialize)]
struct X(usize, usize, usize);

fn main() {
    use Literal::Neg;

    let addresses_cardinality = 5;

    let addresses = 1..=addresses_cardinality;
    let steps = 1..=addresses_cardinality;
    let vips = BTreeSet::from([2]);

    #[rustfmt::skip]
    // Already sorted
    let buses: Vec<(usize, usize)> = vec![
        (1, 2), (1, 3), (2, 4), (2, 5), (3, 1), (3, 5), (4, 1),
(4, 5), (5, 2), (5, 3)
    ];

    let mut encoder = Encoder::new();

    // Almeno un arco per passo
    for t in steps.clone() {
        let mut c = encoder.clause_builder();
        for &(i, j) in buses.iter() {
            c.add(X(t, i, j));
        }
        encoder = c.end()
    }

    // Al più un arco per passo
    for t in steps.clone() {
        for (index, &(i1, j1)) in buses.iter().enumerate() {
            for &(i2, j2) in buses.iter().skip(index + 2) {
                let mut c = encoder.clause_builder();
                c.add(Neg(X(t, i1, j1)));
                c.add(Neg(X(t, i2, j2)));
                encoder = c.end();
            }
        }
    }

    // Almeno un arco per indirizzo
    for j in addresses.clone() {
        let mut c = encoder.clause_builder();
        for t in steps.clone() {
```

```
            for &(i, k) in buses.iter() {
                if k == j {
                    c.add(X(t, i, j));
                }
            }
        }
        encoder = c.end();
    }

    // Al più un arco per indirizzo
    for t1 in steps.clone() {
        for t2 in t1 + 1..=*steps.end() {
            for j in addresses.clone() {
                for i1 in addresses.clone() {
                    for i2 in i1 + 1..=*addresses.end() {
                        if buses.contains(&(i1, j)) &&
buses.contains(&(i2, j)) {
                            let mut c = encoder.clause_builder();
                            c.add(Neg(X(t1, i1, j)));
                            c.add(Neg(X(t2, i2, j)));
                            encoder = c.end();
                        }
                    }
                }
            }
        }
    }

    // Clienti VIP nella prima metà
    for &v in vips.iter() {
        let mut c = encoder.clause_builder();
        for t in 1..=steps.end().div_ceil(2) {
            for i in addresses.clone() {
                if buses.contains(&(i, v)) {
                    c.add(X(t, i, v));
                }
            }
        }
        encoder = c.end();
    }

    // Partenza da casa
    let mut c = encoder.clause_builder();
    for i in 2..=*addresses.end() {
        if buses.contains(&(1, i)) {
            c.add(X(1, 1, i));
        }
    }
    encoder = c.end();

    // Arrivo a casa
    let mut c = encoder.clause_builder();
    for i in 2..=*addresses.end() {
```

```
        if buses.contains(&(i, 1)) {
            c.add(X(*steps.end(), i, 1));
        }
    }
    encoder = c.end();

    // Percorso valido
    for t in *steps.start()..=*steps.end() - 1 {
        for &(i, j) in buses.iter() {
            let mut c = encoder.clause_builder();
            c.add(Neg(X(t, i, j)));
            for k in addresses.clone() {
                if buses.contains(&(j, k)) {
                    c.add(X(t + 1, j, k))
                }
            }
            encoder = c.end();
        }
    }

    encoder.end();
}
```

1.3.1 ...

Devo ancora implementare il decodificatore. La decodifica l'ho fatta a mano a questo giro, e negli esempi che ho provato funziona perfettamente.