GENERACIÓ D'HORARIS D'INSTITUT AMB OPERACIONS LÒGIQUES

Ismael El Habri
Tutors: Dr. Josep Suy i Dr. Jordi Coll

Universitat de Girona

10 de setembre del 2019

Table of Contents

1. Introducció
   - Marc de treball
   - Objectius
   - Metodologia
   - Planificació
   - Pressuposts
2. Implementació
   - Parser
   - Model
   - Restriccions
3. Conclusions i Resultats
   - Resultats
   - Conclusions

Introducció

Confecció d'horaris, problema recurrent amb el que es troben els instituts ⇒ High School TimeTabling problem (HSTT)

- Alta combinatòria i complexitat, és un problema NP-Complet.
- Repartir events i recursos de manera viable i tenint en compte preferències del professorat.
- Diferents països, diferents necessitats ⇒ més complexitat!

Marc de treball

- Grup de recerca de Lògica i Programació
- API SMT desenvolupada pel Dr. Jordi Coll. API per a la codificació de problemes SAT, SMT o MaxSAT, actuant com a interfície per a diferents solvers. En aquest treball s'utilitzarà el Yices 2. També té implementades les diferents implementacions de múltiples restriccions globals.
- Com a punt de partida s'ha utilitzat el treball realitzat el 2015 per en Cristòfor Nogueira. Mentre ell ha utilitzat BitVectors i MaxSAT, en aquest treball s'utilitzarà SMT. Així s'han aconseguit uns resultats superiors pel que fa el temps d'execució.

Objectius

- Aprofundir sobre el tema.
  - Problema de generació d'horaris d'institut.
  - Problemes de satisfacció de restriccions(CSP).
  - Tècniques per resoldre problemes CSP com ara SAT i extensions.
- Crear un generador.

Metodologia

- Estudi del treball previ i estat de l'art
- Entregues periòdiques
- Prototipatge

Planificació

1. Estudi del problema i la seva duresa.
2. Disseny i implementació
    1. Parser i estructura de dades
    2. Codificació model i restriccions
    3. Tractament de la solució
3. Estudi dels resultats

Pressuposts

	€/h	Hores	Cost
Programador	14	260	2520

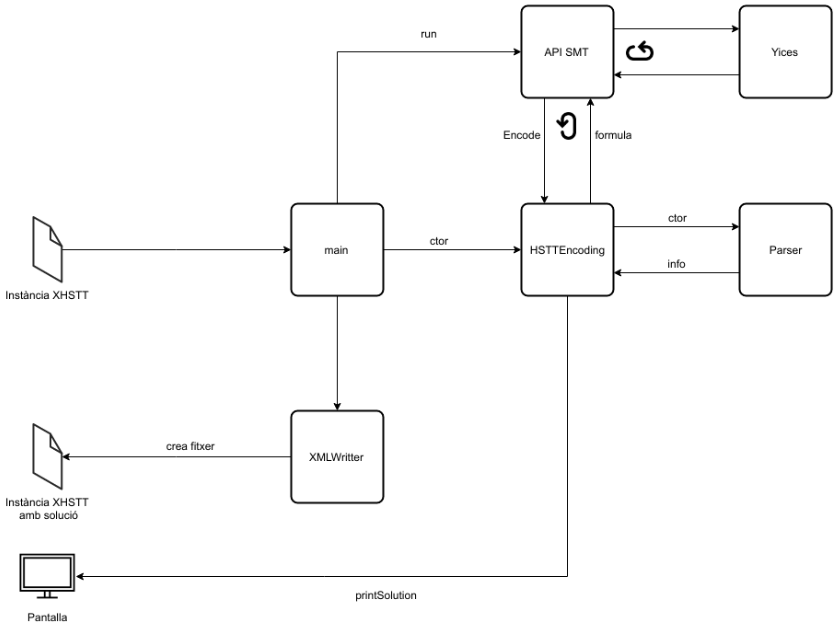
	Cost Total	Hores	€/h
Ordinador Principal	828	240	3.45
Ordinador Portatil Secundari	200	20	10
Total	1028	260	3.95

	Hores
Josep Suy	30
Jordi Coll	20

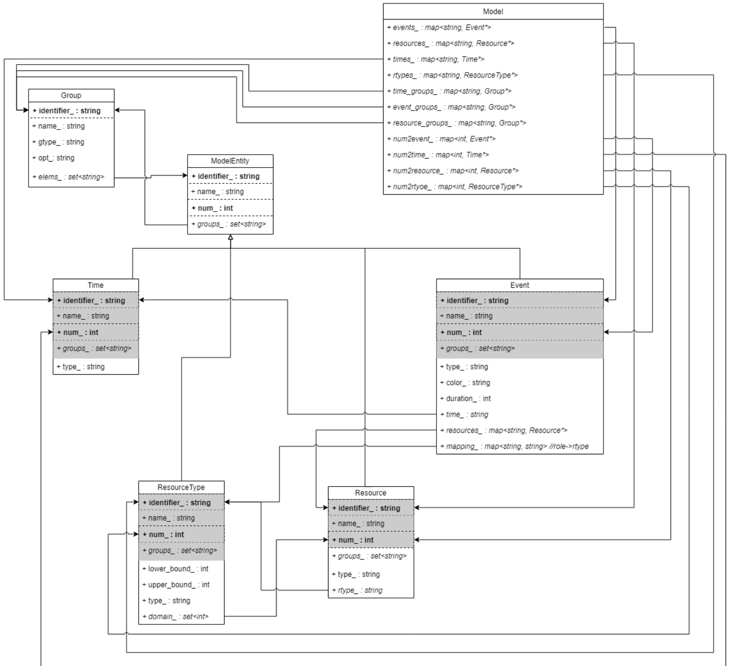
Table of Contents

1. Introducció
   - Marc de treball
   - Objectius
   - Metodologia
   - Planificació
   - Pressuposts
2. Implementació
   - Parser
   - Model
   - Restriccions
3. Conclusions i Resultats
   - Resultats
   - Conclusions

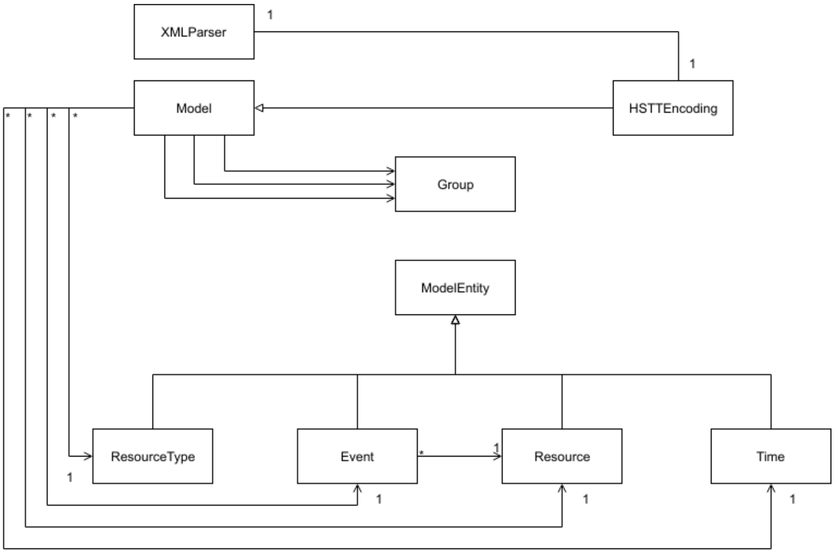
Implementació: Esquema



Implementació: Model de dades



Implementació: Model d'objectes



Parser

```xml
<Resources>
    <ResourceTypes>
        <ResourceType Id="Room">
            <Name>Room</Name>
        </ResourceType>
    </ResourceTypes>
    <ResourceGroups>
        <ResourceGroup Id="Rooms">
            <Name>Rooms</Name>
            <ResourceType Reference="Room"/>
        </ResourceGroup>
    </ResourceGroups>
    <Resource Id="Room1">
        <Name>Room1</Name>
        <ResourceType Reference="Room"/>
        <ResourceGroups>
            <ResourceGroup Reference="Rooms"/>
        </ResourceGroups>
    </Resource>
</Resources>
```

Instància XHSTT $\Rightarrow$ dades + restriccions

Model

- $Xt_{0,0} \ldots Xt_{|Events|-1,|Times|-1}$
  Cada variable ens indica si en un espai de temps, es dona lloc l'event corresponent.
- $Xs_{0,0} \ldots Xs_{|Events|-1,|Times|-1}$
  Cada variable ens indica si en un espai de temps, comença l'event corresponent.
- $Xd_{0,1,0} \ldots Xd_{|Events|-1,event.duration,|Times|-1}$
  Cada variable ens indica si comença una lliçó de la durada i en l'espai de temps que representa la variable.

Clàusules de Channeling

- Si un event comença a una hora determinada, llavors té una duració:

$$\forall e \in 0...|Events| - 1\ \forall i \in 0...|Times| - 1$$
$$exactly\_one(Xs_{e,i} \rightarrow \{Xd_{e,1...e.duration,i}\})$$

- Si un event té lloc a t però no a t-1, és que comença:

$$\forall e \in 0...|Events| - 1\ \forall i \in 0...|Times| - 1$$
$$(Xt_{e,i} \wedge \neg Xt_{i-1}) \rightarrow Xs_i$$

- Si un event comença amb duració d, llavors té lloc en d hores consecutives:

$$\forall e \in 0...|Events| - 1\ \forall din1...e.duration\ \forall i \in 0...|Times| - 1$$
$$\forall j \in i...i + d - 1$$
$$Xd_{e,d,i} \rightarrow Xt_{e,j}$$

Restriccions

- Assign Times Constraint

$$\forall e \in Events\ exactly\_k(\{Xt_{e,0} ... Xt_{e,|Times|-1}\}, e.duration)$$

- Split Events Constraint

$$\forall e \in Events\ at\_most\_k(\{Xs_{e,0} ... Xs_{e,|Times|-1}\}, MaximumAmount)$$
$$\forall e \in Events\ at\_least\_k(\{Xs_{e,0} ... Xs_{e,|Times|-1}\}, MinimumAmount)$$
$$\forall e \in Events\ \forall d \notin MinimumDuration ... MaximumDuration \land d \in 1 ... e.duration$$
$$\forall t \in 0 ... |Times| - 1 \quad \neg Xd_{e,d,t}$$

- Distribute Split Constraint

$$\forall e \in Events\ at\_most\_k(\{Xd_{e,d,0} .... Xd_{e,d,|Times|-1}\}, max) \qquad si\ \ max < \frac{e.duration}{d}$$
$$\forall e \in Events\ at\_most_k(\{Xd_{e,d,0} .... Xd_{e,d,|Times|-1}\}, min) \qquad si\ \ min > 0$$

Restriccions

- Prefer Times Constraint

$$\forall e \in Events\ \forall t \in Times \wedge t \notin Ta$$
$$(\neg Xd_{e,d,t})$$

- Spread Events Constraint

$$\forall e \in Events\ \forall g \in Tg$$
$$at\_most\_k(\{Xs_{e,t} | t \leftarrow g\}, max)$$
$$\forall e \in Events\ \forall g \in Tg$$
$$at\_least\_k(\{Xs_{e,t} | t \leftarrow g\}, max)$$

- Avoid Clashes Constraint

$$\forall r \in Resources\ \forall t \in Times$$
$$at\_most\_one(\{Xt_{e,t} | e \leftarrow E_r\})$$

- Avoid Unavailable Times Constraint

$$\forall r \in Resources\ \forall t \in T\ \forall e \in E_r \quad (\neg Xt_{e,t})$$

Restriccions

- Limit Idle Times Constraint
  Per a cada recurs r es fan les clàusules següents:

$$\forall g \in Tg \ \forall t \in g \ \forall e \in E_r$$
$$(\neg Idle_i \vee \neg Xt_{e,t}) \qquad si \quad (B_t \neq \emptyset$$
$$\forall g \in Tg \ \forall t \in g$$
$$(\neg Idle_t \vee \{\forall b \in B_t \ \forall e \in E_r \ Xt_{e,b}\}) \qquad s$$
$$\forall g \in Tg \ \forall t \in g$$
$$(\neg Idle_t \vee \{\forall a \in A_t \ \forall e \in E_r \ Xt_{e,a}\}) \qquad s$$
$$\forall g \in Tg \ \forall t \in g \ \forall b \in B_t \ \forall a \in A_t$$
$$\forall e1 \in E_r \ \forall e_2 \in E_r \ \forall e_3 \in E_r$$
$$(Xt_{e_1,t} \vee \neg Xt_{e_2,b} \vee Xt_{e_3,a} \vee Idle_t) \qquad si \quad (B_t \neq \emptyset$$

Un cop definides i lligades les variables auxiliars l'únic que queda és, per cada recurs, imposar les restriccions de cardinalitat:

$$\forall r \in Resources$$

Restriccions

- Cluster Busy Times Constraint
  Per a cada recurs r es fan les clàusules següents:

$$\forall g \in Tg \quad (\neg Busy_g \vee (\forall e \in E_r \ \forall t \in g \quad Xt_{e,t}))$$

$$\forall g \in Tg \ \forall t \in g \ \forall e \in E_r \quad (\neg Xt_{e,t} \vee Bsuy_g)$$

Un cop definides i lligades les variables auxiliars l'únic que queda és, per cada recurs, imposar les restriccions de cardinalitat:

$$\forall r \in Resources$$
$$at\_most\_k(\{Busy_g | g \leftarrow Tg\}, max)$$
$$at\_least\_k(\{Busy_g | g \leftarrow Tg\}, min)$$

Table of Contents

1. Introducció
    - Marc de treball
    - Objectius
    - Metodologia
    - Planificació
    - Pressuposts
2. Implementació
    - Parser
    - Model
    - Restriccions
3. Conclusions i Resultats
    - Resultats
    - Conclusions

Resultats resolució

-

Resultats optimització