Generación y simplificación automática de especificaciones de procesos de producción

Autor: Lucio Nardelli

Director: Hernán ($\mathcal{Z}$eta) Ponce de León

Departamento de Ciencias de la Computación
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Universidad Nacional de Rosario

Junio, 2016

Contenidos

Motivación

Motivación

- Hoy en día existe un acceso masivo a sistemas informáticos.

Motivación

- Hoy en día existe un acceso masivo a sistemas informáticos.
- Esto genera inmensas cantidades de información.

Motivación

- Hoy en día existe un acceso masivo a sistemas informáticos.
- Esto genera inmensas cantidades de información.
- Dependencia total de estos sistemas, es esencial mantenerlos eficientes y seguros.

Motivación

- Hoy en día existe un acceso masivo a sistemas informáticos.
- Esto genera inmensas cantidades de información.
- Dependencia total de estos sistemas, es esencial mantenerlos eficientes y seguros.
- ¿Cómo se puede asegurar esto?

Motivación

- Hoy en día existe un acceso masivo a sistemas informáticos.
- Esto genera inmensas cantidades de información.
- Dependencia total de estos sistemas, es esencial mantenerlos eficientes y seguros.
- ¿Cómo se puede asegurar esto?
- Utilizaremos métodos de representación formal.

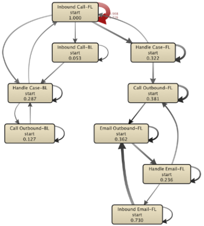
Motivación

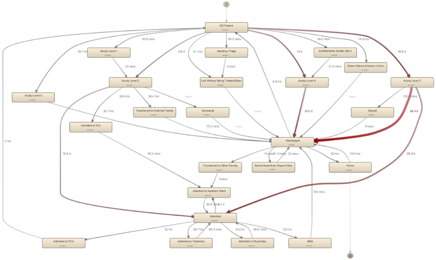
Modelos formales

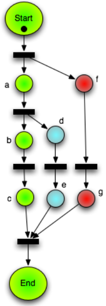
Utilizar un modelo gráfico, fácil de entender pero formal para que permitir su análisis.

Motivación
Modelos formales

Utilizar un modelo gráfico, fácil de entender pero formal para que permitir su análisis.

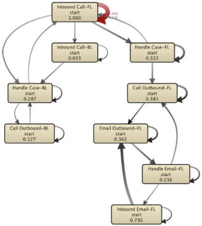


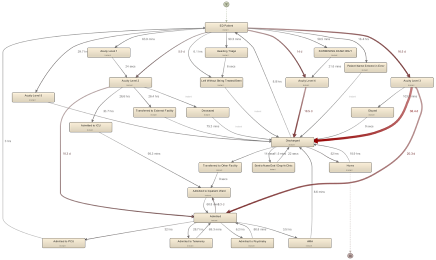


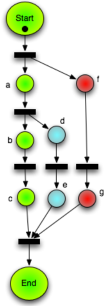


Motivación
Modelos formales

Utilizar un modelo gráfico, fácil de entender pero formal para que permitir su análisis.







El problema es que estos modelos formales, ¡rara vez existen!

Motivación

Descubrimiento de procesos

- Para obtener el modelo recurrimos al *descubrimiento de procesos*.

Motivación

Descubrimiento de procesos

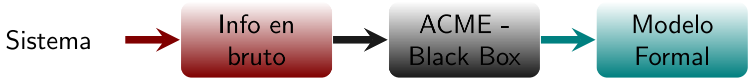
- Para obtener el modelo recurrimos al *descubrimiento de procesos*.
- Es una técnica de aprendizaje automatizado.

Motivación
Descubrimiento de procesos

- Para obtener el modelo recurrimos al *descubrimiento de procesos*.
- Es una técnica de aprendizaje automatizado.
- Transforma un registro de acciones de un sistema en un modelo formal.

Motivación
Descubrimiento de procesos

- Para obtener el modelo recurrimos al *descubrimiento de procesos*.
- Es una técnica de aprendizaje automatizado.
- Transforma un registro de acciones de un sistema en un modelo formal.



Motivación

Mejora de modelos

- Problema con las técnicas de descubrimiento: modelos *spaghetti*.

Motivación

Mejora de modelos

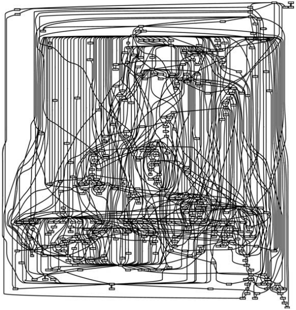
- Problema con las técnicas de descubrimiento: modelos *spaghetti*.

Los spaghetti pueden ser una muy buena idea para una cena romántica, pero no son una buena idea para un modelo formal...

Motivación

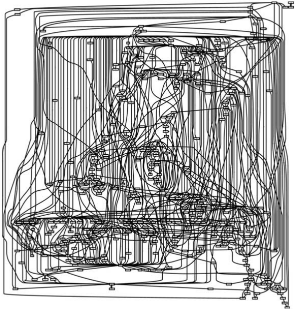
Mejora de modelos

- Problema con las técnicas de descubrimiento: modelos *spaghetti*.



Motivación
Mejora de modelos

- Problema con las técnicas de descubrimiento: modelos *spaghetti*.



¿Quién podría entender algo de un modelo como ese?

Motivación
Mejora de modelos

- Para mejorarlos se aplican técnicas de *mejora de modelos*

Motivación
Mejora de modelos

- Para mejorarlos se aplican técnicas de *mejora de modelos*
- Puede lograrse considerando las trazas más frecuentes o bien información de comportamientos que *no* deben ocurrir.

Motivación
Objetivo general

- Obtener un modelo formal a partir del registro de actividades.

Motivación
Objetivo general

- Obtener un modelo formal a partir del registro de actividades.
- Simplificar el modelo.

Motivación

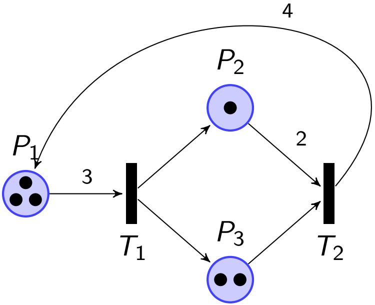
Objetivo general

- Obtener un modelo formal a partir del registro de actividades.
- Simplificar el modelo.
- Automatizar los puntos anteriores.

Contenidos

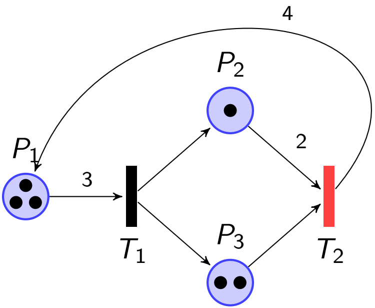
Redes de Petri

Ejemplo de la evolución de una red de Petri



Redes de Petri

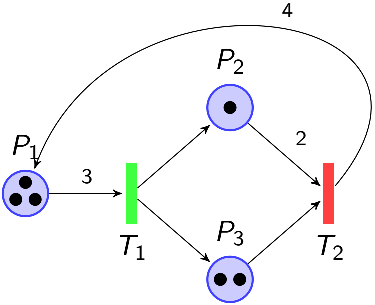
Ejemplo de la evolución de una red de Petri



$M_0(p_1) = 3$
$M_0(p_2) = 1$
$M_0(p_3) = 2$

Redes de Petri

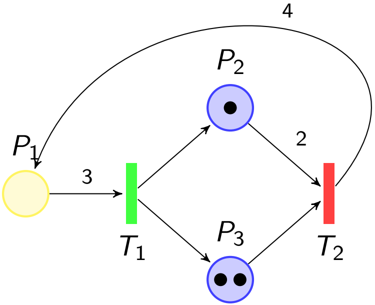
Ejemplo de la evolución de una red de Petri



$M_0(p_1) = 3$
$M_0(p_2) = 1$
$M_0(p_3) = 2$

Redes de Petri

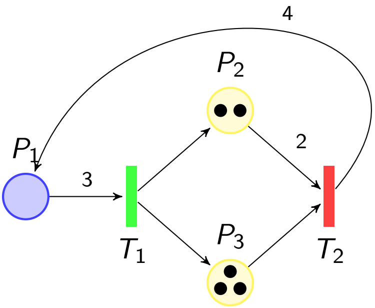
Ejemplo de la evolución de una red de Petri



$M_0(p_1) = 3$
$M_0(p_2) = 1$
$M_0(p_3) = 2$

Redes de Petri

Ejemplo de la evolución de una red de Petri



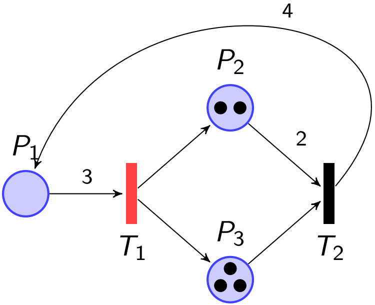
$$M_0(p_1) = 3$$
$$M_0(p_2) = 1$$
$$M_0(p_3) = 2$$

T1

$$M_1(p_1) = M_0(p_1) - 3 = 0$$
$$M_1(p_2) = M_0(p_2) + 1 = 2$$
$$M_1(p_3) = M_0(p_3) + 1 = 3$$

Redes de Petri

Ejemplo de la evolución de una red de Petri



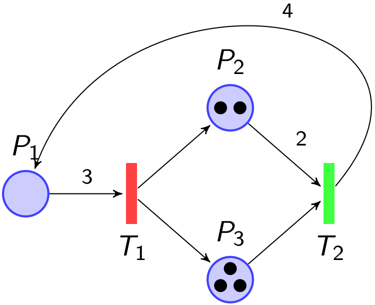
$$M_0(p_1) = 3$$
$$M_0(p_2) = 1$$
$$M_0(p_3) = 2$$

$$\downarrow \text{T1}$$

$$M_1(p_1) = M_0(p_1) - 3 = 0$$
$$M_1(p_2) = M_0(p_2) + 1 = 2$$
$$M_1(p_3) = M_0(p_3) + 1 = 3$$

Redes de Petri

Ejemplo de la evolución de una red de Petri



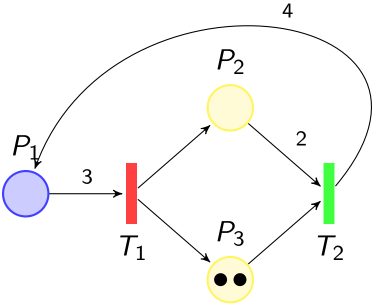
$M_0(p_1) = 3$
$M_0(p_2) = 1$
$M_0(p_3) = 2$

T1

$M_1(p_1) = M_0(p_1) - 3 = 0$
$M_1(p_2) = M_0(p_2) + 1 = 2$
$M_1(p_3) = M_0(p_3) + 1 = 3$

Redes de Petri

Ejemplo de la evolución de una red de Petri



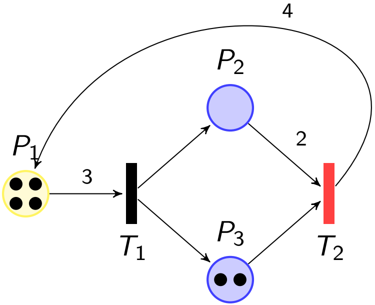
$$M_0(p_1) = 3$$
$$M_0(p_2) = 1$$
$$M_0(p_3) = 2$$

$$\Big\downarrow \text{T1}$$

$$M_1(p_1) = M_0(p_1) - 3 = 0$$
$$M_1(p_2) = M_0(p_2) + 1 = 2$$
$$M_1(p_3) = M_0(p_3) + 1 = 3$$

Redes de Petri

Ejemplo de la evolución de una red de Petri



$$\begin{aligned} M_0(p_1) &= 3 \\ M_0(p_2) &= 1 \\ M_0(p_3) &= 2 \end{aligned}$$

$$\Big\downarrow \text{T1}$$

$$\begin{aligned} M_1(p_1) &= M_0(p_1) - 3 = 0 \\ M_1(p_2) &= M_0(p_2) + 1 = 2 \\ M_1(p_3) &= M_0(p_3) + 1 = 3 \end{aligned}$$

$$\Big\downarrow \text{T2}$$

$$\begin{aligned} M_2(p_1) &= M_1(p_1) + 4 = M_0(p_1) - 3 + 4 = 4 \\ M_2(p_2) &= M_1(p_2) - 2 = M_0(p_2) + 1 - 2 = 0 \\ M_2(p_3) &= M_1(p_2) - 1 = M_0(p_3) + 1 - 1 = 2 \end{aligned}$$

Redes de Petri

Evolución de una red de Petri

- Los markings pueden definirse de manera incremental como

$$M'(p) = M(p) - F(p,t) + F(t,p).$$

Redes de Petri

Evolución de una red de Petri

- Los markings pueden definirse de manera incremental como

$$M'(p) = M(p) - F(p,t) + F(t,p).$$

- Más general, ante una sucesión de eventos:

$$M(p) = M_0(p) + \sum_{x_i} F(x_i, p) \cdot \widehat{\sigma}(x_i) - \sum_{y_i} F(p, y_i) \cdot \widehat{\sigma}(y_i).$$

Redes de Petri

Evolución de una red de Petri

- Los markings pueden definirse de manera incremental como

$$M'(p) = M(p) - F(p,t) + F(t,p).$$

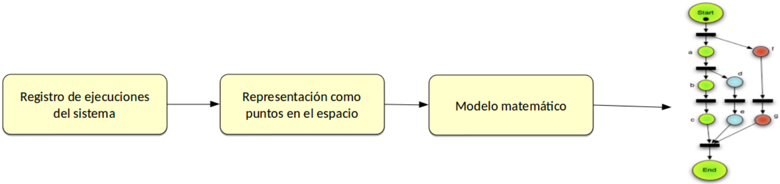
- Más general, ante una sucesión de eventos:

$$M(p) = M_0(p) + \sum_{x_i} F(x_i, p) \cdot \widehat{\sigma}(x_i) - \sum_{y_i} F(p, y_i) \cdot \widehat{\sigma}(y_i).$$

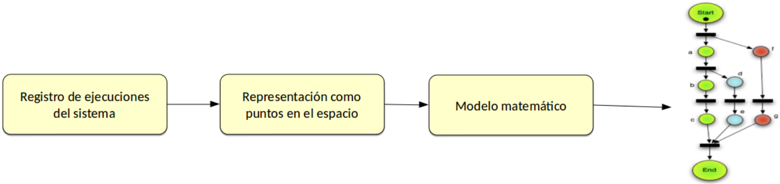
- Utilizando notación matricial, puede definirse para todos los places de una red:

$$M = M_0 + A \cdot \widehat{\sigma}.$$

Enfoque del trabajo



Enfoque del trabajo



- El registro de ejecuciones es generado por el sistema en los *logs de eventos*.
- La representación como puntos en el espacio corresponde al *conjunto de vectores Parikh* del log.

De logs y vectores Parikh
Conceptos

- Un *log de eventos* es un conjunto de trazas o sucesión ordenada de actividades relevantes de un sistema.

De logs y vectores Parikh

Conceptos

- Un *log de eventos* es un conjunto de trazas o sucesión ordenada de actividades relevantes de un sistema.
- Dada una traza $\sigma = \sigma_1 \cdot \sigma_2 \cdot \ldots \cdot \sigma_k$ sobre un alfabeto $T = \{t_1, t_2, \ldots, t_n\}$, el *vector Parikh* corresponde a la cantidad de ocurrencias de cada acción t_i en σ.

De logs y vectores Parikh

Conceptos

- Un *log de eventos* es un conjunto de trazas o sucesión ordenada de actividades relevantes de un sistema.
- Dada una traza $\sigma = \sigma_1 \cdot \sigma_2 \cdot \ldots \cdot \sigma_k$ sobre un alfabeto $T = \{t_1, t_2, \ldots, t_n\}$, el *vector Parikh* corresponde a la cantidad de ocurrencias de cada acción t_i en σ.

Ejemplo

Para las trazas $\sigma_1 = t_1 \cdot t_2 \cdot t_1 \cdot t_2 \cdot t_1 \cdot t_1 \cdot t_3$ y $\sigma_2 = t_4 \cdot t_4$, sobre el alfabeto $T = \{t_1, t_2, t_3, t_4\}$ los vectores de Parikh de cada traza vienen dados por las tuplas $(4, 2, 1, 0)$ y $(0, 0, 0, 2)$ respectivamente.

De logs y vectores Parikh

Conceptos

- Un *log de eventos* es un conjunto de trazas o sucesión ordenada de actividades relevantes de un sistema.
- Dada una traza $\sigma = \sigma_1 \cdot \sigma_2 \cdot \ldots \cdot \sigma_k$ sobre un alfabeto $T = \{t_1, t_2, \ldots, t_n\}$, el *vector Parikh* corresponde a la cantidad de ocurrencias de cada acción t_i en σ.

Ejemplo

Para las trazas $\sigma_1 = t_1 \cdot t_2 \cdot t_1 \cdot t_2 \cdot t_1 \cdot t_1 \cdot t_3$ y $\sigma_2 = t_4 \cdot t_4$, sobre el alfabeto $T = \{t_1, t_2, t_3, t_4\}$ los vectores de Parikh de cada traza vienen dados por las tuplas $(4, 2, 1, 0)$ y $(0, 0, 0, 2)$ respectivamente.

Entonces, teníamos un log de eventos y ahora podemos convertirlo en tuplas de enteros...

De logs y vectores Parikh

Conceptos

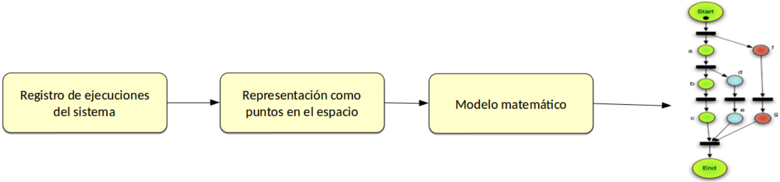
- Un *log de eventos* es un conjunto de trazas o sucesión ordenada de actividades relevantes de un sistema.
- Dada una traza $\sigma = \sigma_1 \cdot \sigma_2 \cdot \ldots \cdot \sigma_k$ sobre un alfabeto $T = \{t_1, t_2, \ldots, t_n\}$, el *vector Parikh* corresponde a la cantidad de ocurrencias de cada acción t_i en σ.

Ejemplo

Para las trazas $\sigma_1 = t_1 \cdot t_2 \cdot t_1 \cdot t_2 \cdot t_1 \cdot t_1 \cdot t_3$ y $\sigma_2 = t_4 \cdot t_4$, sobre el alfabeto $T = \{t_1, t_2, t_3, t_4\}$ los vectores de Parikh de cada traza vienen dados por las tuplas $(4, 2, 1, 0)$ y $(0, 0, 0, 2)$ respectivamente.

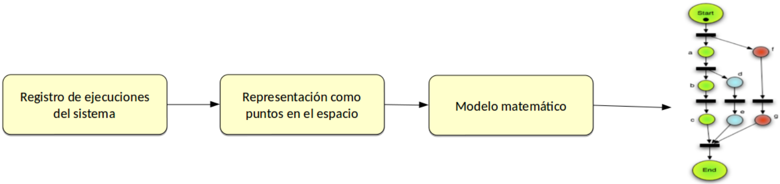
Entonces, teníamos un log de eventos y ahora podemos convertirlo en tuplas de enteros...

Enfoque del trabajo



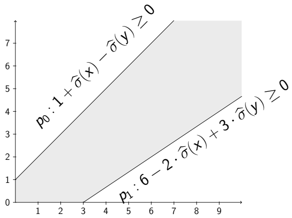
- El registro de ejecuciones es generado por el sistema en los *logs de eventos*.
- La representación como puntos en el espacio corresponde al conjunto de vectores Parikh del log.

Enfoque del trabajo

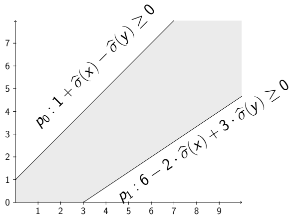


- El registro de ejecuciones es generado por el sistema en los *logs de eventos*.
- La representación como puntos en el espacio corresponde al conjunto de vectores Parikh del log.
- El modelo matemático corresponde a un poliedro convexo que contenga el conjunto de vectores Parikh.

Dominios numéricos abstractos
Poliedros convexos

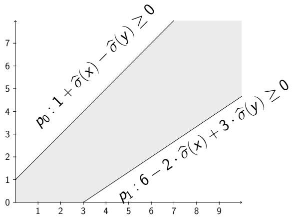


Dominios numéricos abstractos
Poliedros convexos



- Los *semi-espacios* se representan mediante una inecuación lineal de la forma $a_1x_1 + a_2x_2 + \cdots + a_nx_n \geq b$.

Dominios numéricos abstractos
Poliedros convexos

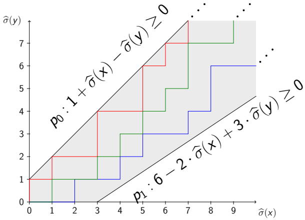


- Los *semi-espacios* se representan mediante una inecuación lineal de la forma $a_1x_1 + a_2x_2 + \cdots + a_nx_n \geq b$.
- Un poliedro convexo $\mathcal{P}$ puede representarse como una intersección de un conjunto de k hiper-espacios

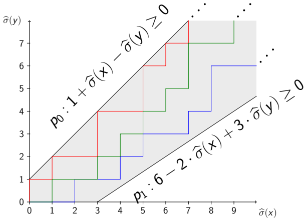
$$\mathcal{P} = \{x \in \mathbb{R}^n \mid A \cdot x + b \geq 0\}$$

donde $A \in \mathbb{R}^{k \times n}$ y $b \in \mathbb{R}^k$.

Descubrimiento de procesos
¡Juntemos todo!



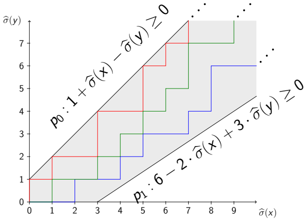
Descubrimiento de procesos
¡Juntemos todo!



$$\mathcal{P} = \left\{ \begin{bmatrix} \widehat{\sigma}(x) \\ \widehat{\sigma}(y) \end{bmatrix} \;\middle|\; \begin{array}{c} 1 + \widehat{\sigma}(x) - \widehat{\sigma}(y) \geq 0 \\ 6 - 2 \cdot \widehat{\sigma}(x) + 3 \cdot \widehat{\sigma}(y) \geq 0 \end{array} \right\}$$

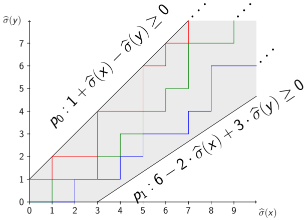
Descubrimiento de procesos

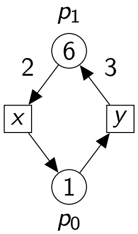
¡Juntemos todo!



$$\mathcal{P} = \left\{ \begin{bmatrix} \widehat{\sigma}(x) \\ \widehat{\sigma}(y) \end{bmatrix} \;\middle|\; \begin{array}{c} 1 + \widehat{\sigma}(x) - \widehat{\sigma}(y) \geq 0 \\ 6 - 2 \cdot \widehat{\sigma}(x) + 3 \cdot \widehat{\sigma}(y) \geq 0 \end{array} \right\} \qquad M = \begin{bmatrix} 1 \\ 6 \end{bmatrix} + \begin{bmatrix} 1 & -1 \\ -2 & 3 \end{bmatrix} \cdot \begin{bmatrix} \widehat{\sigma}(x) \\ \widehat{\sigma}(y) \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Descubrimiento de procesos
¡Juntemos todo!





$$\mathcal{P} = \left\{ \begin{bmatrix} \widehat{\sigma}(x) \\ \widehat{\sigma}(y) \end{bmatrix} \;\middle|\; \begin{array}{c} 1 + \widehat{\sigma}(x) - \widehat{\sigma}(y) \geq 0 \\ 6 - 2 \cdot \widehat{\sigma}(x) + 3 \cdot \widehat{\sigma}(y) \geq 0 \end{array} \right\} \qquad M = \begin{bmatrix} 1 \\ 6 \end{bmatrix} + \begin{bmatrix} 1 & -1 \\ -2 & 3 \end{bmatrix} \cdot \begin{bmatrix} \widehat{\sigma}(x) \\ \widehat{\sigma}(y) \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Contenidos

Descubrimiento de procesos

Simplificación

- Modelos excesivamente complicados.
- Necesidad de simplificar.

$$
\begin{array}{rcrcrcrcc}
120 & + & 64 \cdot x_1 & + & 33 \cdot x_2 & - & 10 \cdot x_3 & \geq & 0 \\
-40 & + & 14 \cdot x_1 & - & 62 \cdot x_2 & + & 8 \cdot x_3 & \geq & 0 \\
-83 & + & 20 \cdot x_1 & + & 46 \cdot x_2 & + & 40 \cdot x_3 & \geq & 0
\end{array}
$$

Descubrimiento de procesos
Simplificación

- Modelos excesivamente complicados.
- Necesidad de simplificar.

$$\begin{array}{ccccccccc} ??? & + & ??? \cdot x_1 & + & ??? \cdot x_2 & + & ??? \cdot x_3 & \geq & 0 \\ ??? & + & ??? \cdot x_1 & + & ??? \cdot x_2 & + & ??? \cdot x_3 & \geq & 0 \\ ??? & + & ??? \cdot x_1 & + & ??? \cdot x_2 & + & ??? \cdot x_3 & \geq & 0 \end{array}$$

Descubrimiento de procesos
Simplificación

- Modelos excesivamente complicados.
- Necesidad de simplificar.

$$\begin{array}{ccccccccc}
??? & + & ??? \cdot x_1 & + & ??? \cdot x_2 & + & ??? \cdot x_3 & \geq & 0 \\
??? & + & ??? \cdot x_1 & + & ??? \cdot x_2 & + & ??? \cdot x_3 & \geq & 0 \\
??? & + & ??? \cdot x_1 & + & ??? \cdot x_2 & + & ??? \cdot x_3 & \geq & 0
\end{array}$$

- Buscamos nuevos coeficientes para el sistema de inecuaciones:
  - **MIN**imización: Los nuevos coeficientes deben ser más "simples".
  - **P**reservación **T**razas: Cada solución del sistema original tiene que continuar siendo una solución del nuevo sistema.

Descubrimiento de procesos
Simplificación

Dado un modelo de la forma:

$$
\begin{array}{ccccccccc}
\alpha_{1,0} & + & \alpha_{1,1} \cdot x_1 & + & \ldots & + & \alpha_{1,n} \cdot x_n & \geq & 0 \\
\alpha_{2,0} & + & \alpha_{2,1} \cdot x_1 & + & \ldots & + & \alpha_{2,n} \cdot x_n & \geq & 0 \\
\ldots & & & & & & \ldots & & \\
\alpha_{m,0} & + & \alpha_{m,1} \cdot x_1 & + & \ldots & + & \alpha_{m,n} \cdot x_n & \geq & 0
\end{array}
$$

Descubrimiento de procesos
Simplificación

Dado un modelo de la forma:

$$\begin{array}{ccccccccc} \alpha_{1,0} & + & \alpha_{1,1} \cdot x_1 & + & \dots & + & \alpha_{1,n} \cdot x_n & \geq & 0 \\ \alpha_{2,0} & + & \alpha_{2,1} \cdot x_1 & + & \dots & + & \alpha_{2,n} \cdot x_n & \geq & 0 \\ \dots & & & & & & \dots & & \\ \alpha_{m,0} & + & \alpha_{m,1} \cdot x_1 & + & \dots & + & \alpha_{m,n} \cdot x_n & \geq & 0 \end{array}$$

Se buscan nuevos coeficientes $\beta_{1,0}, \beta_{1,1}, \dots, \beta_{m,n}$ tal que:

$$|\beta_{i,j}| \leq |\alpha_{i,j}|. \tag{MIN}$$

$$\bigwedge_{i=1}^{m} (\alpha_{i,0} + \sum_{j=1}^{n} \alpha_{i,j} \cdot x_j) \geq 0 \Rightarrow \bigwedge_{i=1}^{m} (\beta_{i,0} + \sum_{j=1}^{n} \beta_{i,j} \cdot x_j) \geq 0. \tag{PT}$$

Descubrimiento de procesos
Simplificación

Dado un modelo de la forma:

$$\begin{array}{ccccccccc}
\alpha_{1,0} & + & \alpha_{1,1} \cdot x_1 & + & \dots & + & \alpha_{1,n} \cdot x_n & \geq & 0 \\
\alpha_{2,0} & + & \alpha_{2,1} \cdot x_1 & + & \dots & + & \alpha_{2,n} \cdot x_n & \geq & 0 \\
\dots & & & & & & \dots & & \\
\alpha_{m,0} & + & \alpha_{m,1} \cdot x_1 & + & \dots & + & \alpha_{m,n} \cdot x_n & \geq & 0
\end{array}$$

Se buscan nuevos coeficientes $\beta_{1,0}, \beta_{1,1}, \dots, \beta_{m,n}$ tal que:

$$|\beta_{i,j}| \leq |\alpha_{i,j}|. \qquad \text{(MIN)}$$

$$\bigwedge_{i=1}^{m} (\alpha_{i,0} + \sum_{j=1}^{n} \alpha_{i,j} \cdot x_j) \geq 0 \Rightarrow \bigwedge_{i=1}^{m} (\beta_{i,0} + \sum_{j=1}^{n} \beta_{i,j} \cdot x_j) \geq 0. \qquad \text{(PT)}$$

- Para realizar el proceso de simplificación se utiliza *SMT-Solver*.
- Los SMT (*Satisfability modulo theories*) permiten definir un sistema de restricciones y encontrar una solución.

Descubrimiento de procesos

- Finalmente conseguimos el modelo simplificado.

Descubrimiento de procesos

- Finalmente conseguimos el modelo simplificado.

¡Qué bien!

Descubrimiento de procesos

- Finalmente conseguimos el modelo simplificado.
- Simplificar agregó nuevos puntos y por lo tanto comportamientos potencialmente indeseados.

Descubrimiento de procesos

- Finalmente conseguimos el modelo simplificado.
- Simplificar agregó nuevos puntos y por lo tanto comportamientos potencialmente indeseados.

¡Qué mal!

Descubrimiento de procesos

- Finalmente conseguimos el modelo simplificado.
- Simplificar agregó nuevos puntos y por lo tanto comportamientos potencialmente indeseados.
- Si contamos con información negativa podemos mejorarlo.

Descubrimiento de procesos

- Finalmente conseguimos el modelo simplificado.
- Simplificar agregó nuevos puntos y por lo tanto comportamientos potencialmente indeseados.
- Si contamos con información negativa podemos mejorarlo.

¡Qué bien!

Descubrimiento de procesos

- Finalmente conseguimos el modelo simplificado.
- Simplificar agregó nuevos puntos y por lo tanto comportamientos potencialmente indeseados.
- Si contamos con información negativa podemos mejorarlo.
- La información negativa representa conocimiento experto sobre comportamiento que el sistema no debe admitir.

Descubrimiento de procesos

- Finalmente conseguimos el modelo simplificado.
- Simplificar agregó nuevos puntos y por lo tanto comportamientos potencialmente indeseados.
- Si contamos con información negativa podemos mejorarlo.
- La información negativa representa conocimiento experto sobre comportamiento que el sistema no debe admitir.

...

Algoritmo de descubrimiento de procesos

Algoritmo completo de descubrimiento y simplificación supervisado

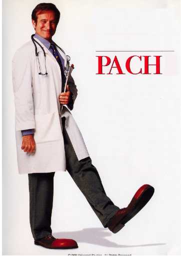
Entrada: trazas positivas $\mathcal{L}^+$ y trazas negativas $\mathcal{L}_-$
Salida: una red de Petri N donde $\forall\, \sigma \in \mathcal{L}^+ : \sigma \in L(N)$ y $\forall\, \sigma \in \mathcal{L}_- : \sigma \notin L(N)$

procedure $\mathrm{DISCOVER}(\mathcal{L}^+, \mathcal{L}_-)$
    $pp = \textsc{Parikh}(\mathcal{L}^+)$
    $np = \textsc{Parikh}(\mathcal{L}_-)$
    $H = \textsc{ConvexHull}(pp)$
    $H_{smt} = \textsc{ShiftRotate}(H, np)$
    $N = \textsc{Hull2Net}(H_{smt})$
    return N
end procedure

Descubrimiento de procesos

PacH

- Desarrollo herramienta PacH en Python.
- Diferentes parámetros de configuración.
- Posibilita ser usada como herramienta de post procesamiento.
- Código disponible en GitHub: `/lucionardelli/PacH`.



PACH
Resultados Experimentales

Benchmark	Poliedro	SMT Pos.	SMT Neg.
A(32)	13129	10324	6489
A(42)	7813	4209	5009
CONFDIMB	33	29	29
CYCLES(5)	105	102	105
DOCUMENTFLOW	56	50	52
INCIDENT	406	216	292
RECEIPT	588	412	462
TELECOM	840	592	688

Resultados de complejidad de los modelos obtenidos mediante PACH.

Contenidos

Conclusión

- Desarrollo de una herramienta que implementa minería de procesos de manera supervisada.
- Uso de SMT-Solver como herramienta de simplificación.
- La solución propuesta es independiente del algoritmo de descubrimiento.
- Buenos resultados experimentales.

Trabajo futuro

- Realizar pruebas con información negativa proporcionada por conocimiento experto.
- Descomposición en subproblemas ante información negativa dentro del poliedro positivo.
- Combinar con otras técnicas de simplificación.
- Obtención de poliedro de manera eficiente mediante SMT-Solver.

¿Preguntas?

¿Preguntas?

¡Gracias!