De la información al Conocimiento

Aplicaciones basadas en implicaciones y computación paralela

TESIS DOCTORAL

Fernando Benito Picazo

Dpto. Lenguajes y Ciencias de la Computación

Universidad de Málaga

Directores: Dr. Manuel Enciso, Dr. Carlos Rossi

Índice

Preámbulo

- Ingeniero en Informática, MSc. Ingeniería del SW e IA
- Doctorando a tiempo parcial (2014-2018)
- 3 publicaciones JCR (Q2), 4 congresos internacionales
- Docencia universitaria (UMA, US)
- Cursos, seminarios
- Revisor, Knowledge-Based Systems, factor impacto: 4.529
- CIO, International Boarding Solutions
- Grupos de Investigación: SICUMA, GIMAC, IMUS

Índice

Introducción

- La gestión de la información es uno de los pilares esenciales de la Ingeniería Informática.
- El trabajo desarrollado se ha centrado en hacer aportaciones dentro de dos campos de conocimiento:
  - Bases de datos
  - Sistemas de recomendación (SRs)
- Base teórica: Análisis Formal de Conceptos (FCA).
  - Implicaciones

Análisis Formal de Conceptos

FCA es una teoría matemática y una metodología que permite derivar una jerarquía de conceptos a partir de una colección de objetos, sus atributos y las relaciones entre ellos [23].

Análisis Formal de Conceptos

FCA se ha aplicado con éxito en diferentes áreas:

- Biología celular [20]
- Ingeniería del software [40, 29]
- Medicina [45]
- Genética [53]
- ...

Análisis Formal de Conceptos

FCA agrupa conjuntos de objetos y sus atributos en forma de tablas, denominadas contextos formales, que representan las relaciones entre esos objetos y atributos.

Análisis Formal de Conceptos

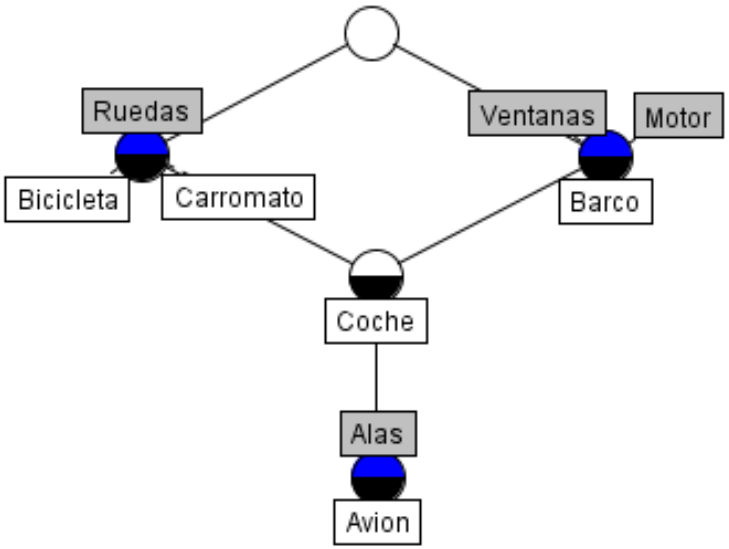
FCA agrupa conjuntos de objetos y sus atributos en forma de tablas, denominadas contextos formales, que representan las relaciones entre esos objetos y atributos.

	Alas	Motor	Ventanas	Ruedas
Bicicleta				✓
Carromato				✓
Barco		✓	✓	
Coche		✓	✓	✓
Avión	✓	✓	✓	✓

Análisis Formal de Conceptos

A partir de ahí, se generan dos herramientas básicas para representar el conocimiento:

Retículo de conceptos



Conjunto de implicaciones

Motor $\rightarrow$ Ventanas
Ventanas $\rightarrow$ Motor
Alas $\rightarrow$ Motor Ventanas Ruedas

Implicaciones

Básicamente, las implicaciones pueden considerarse como reglas del tipo *si-entonces*, que representan un concepto muy intuitivo:

Cuando se verifica una premisa, entonces se cumple una conclusión

Implicaciones

Básicamente, las implicaciones pueden considerarse como reglas del tipo *si-entonces*, que representan un concepto muy intuitivo:

Cuando se verifica una premisa, entonces se cumple una conclusión

- FCA: Implicaciones
- Teoría relacional: Dependencias funcionales (DFs)

Implicaciones

Trabajar con conjuntos de implicaciones permite utilizar técnicas de razonamiento automático basadas en la lógica.

Objetivo principal: a partir de los conjuntos de implicaciones, aplicar mecanismos lógicos para realizar un tratamiento eficiente de la información.

1. Claves minimales
2. Generadores minimales
3. Sistemas de recomendación conversacionales

Concepto de Clave

El concepto de clave es fundamental en cualquier modelo de datos, incluyendo el modelo de datos relacional de Codd [10].

Está compuesta por un subconjunto de atributos que permite identificar cada fila de una tabla, impidiendo que exista más de una fila con la misma información.

Puede representarse por medio de dependencias funcionales (DFs).

Concepto de Dependencia Funcional

Las DFs especifican una relación entre dos subconjuntos de atributos, e.g. A y B, representada como $A \rightarrow B$, que asegura que para cualesquiera dos tuplas de una tabla de datos, si los valores de sus atributos de A coinciden, entonces también han de coincidir los de B.

Ejemplo Básico

Título	Año	País	Director	Nacionalidad	Actor
Pulp Fiction	1994	USA	Q. Tarantino	USA	John Travolta
Pulp Fiction	1994	USA	Q. Tarantino	USA	Uma Thurman
Pulp Fiction	1994	USA	Q. Tarantino	USA	Samuel Jackson
King Kong	2005	NZ	P. Jackson	NZ	Naomi Watts
King Kong	2005	NZ	P. Jackson	NZ	Jack Black
King Kong	1976	USA	De Laurentiis	IT	Jessica Lange
King Kong	1976	USA	De Laurentiis	IT	Jeff Bridges
Django Unchained	2012	USA	Q. Tarantino	USA	Jamie Foxx
Django Unchained	2012	USA	Q. Tarantino	USA	Samuel Jackson
Blade Runner	1982	USA	R. Scott	UK	Harrison Ford
Blade Runner	2017	USA	D. Villeneuve	CAN	Harrison Ford

Ejemplo Básico

Título	Año	País	Director	Nacionalidad	Actor
Pulp Fiction	1994	USA	Q. Tarantino	USA	John Travolta
Pulp Fiction	1994	USA	Q. Tarantino	USA	Uma Thurman
Pulp Fiction	1994	USA	Q. Tarantino	USA	Samuel Jackson
King Kong	2005	NZ	P. Jackson	NZ	Naomi Watts
King Kong	2005	NZ	P. Jackson	NZ	Jack Black
King Kong	1976	USA	De Laurentiis	IT	Jessica Lange
King Kong	1976	USA	De Laurentiis	IT	Jeff Bridges
Django Unchained	2012	USA	Q. Tarantino	USA	Jamie Foxx
Django Unchained	2012	USA	Q. Tarantino	USA	Samuel Jackson
Blade Runner	1982	USA	R. Scott	UK	Harrison Ford
Blade Runner	2017	USA	D. Villeneuve	CAN	Harrison Ford

$$\text{DFs} = \{Director \rightarrow Nacionalidad;\ Titulo, Año \rightarrow Pais, Director\}$$

Ejemplo Básico

Título	Año	País	Director	Nacionalidad	Actor
Pulp Fiction	1994	USA	Q. Tarantino	USA	John Travolta
Pulp Fiction	1994	USA	Q. Tarantino	USA	Uma Thurman
Pulp Fiction	1994	USA	Q. Tarantino	USA	Samuel Jackson
King Kong	2005	NZ	P. Jackson	NZ	Naomi Watts
King Kong	2005	NZ	P. Jackson	NZ	Jack Black
King Kong	1976	USA	De Laurentiis	IT	Jessica Lange
King Kong	1976	USA	De Laurentiis	IT	Jeff Bridges
Django Unchained	2012	USA	Q. Tarantino	USA	Jamie Foxx
Django Unchained	2012	USA	Q. Tarantino	USA	Samuel Jackson
Blade Runner	1982	USA	R. Scott	UK	Harrison Ford
Blade Runner	2017	USA	D. Villeneuve	CAN	Harrison Ford

$$\text{DFs} = \{\mathit{Director} \rightarrow \mathit{Nacionalidad};\ \mathit{Titulo}, \mathit{Año} \rightarrow \mathit{Pais}, \mathit{Director}\}$$

Ejemplo Básico

Título	Año	País	Director	Nacionalidad	Actor
Pulp Fiction	1994	USA	Q. Tarantino	USA	John Travolta
Pulp Fiction	1994	USA	Q. Tarantino	USA	Uma Thurman
Pulp Fiction	1994	USA	Q. Tarantino	USA	Samuel Jackson
King Kong	2005	NZ	P. Jackson	NZ	Naomi Watts
King Kong	2005	NZ	P. Jackson	NZ	Jack Black
King Kong	1976	USA	De Laurentiis	IT	Jessica Lange
King Kong	1976	USA	De Laurentiis	IT	Jeff Bridges
Django Unchained	2012	USA	Q. Tarantino	USA	Jamie Foxx
Django Unchained	2012	USA	Q. Tarantino	USA	Samuel Jackson
Blade Runner	1982	USA	R. Scott	UK	Harrison Ford
Blade Runner	2017	USA	D. Villeneuve	CAN	Harrison Ford

Clave $= \{Año, Actor\}$

El Problema de la Búsqueda de Claves

Consiste en encontrar todos los subconjuntos de atributos que componen una clave minimal a partir de un conjunto de DFs.
Es un campo de estudio con décadas de antigüedad [47, 21].
Problema complejo para el que no existe un algoritmo que lo resuelva en tiempo polinómico. Es NP-completo [31, 59].

El Problema de la Búsqueda de Claves

Existen numerosos trabajos en la literatura sobre el problema de la búsqueda de claves minimales:

- [46] presentaron un algoritmo usando grafos con atributos.
- [61] utiliza mapas de Karnaugh para calcular todas las claves.
- [50, 56] ...

El Problema de la Búsqueda de Claves

En [13], los autores muestran cómo el problema de las claves minimales en las bases de datos tiene su análogo en FCA, donde el papel de las DFs se trata como implicaciones de atributos.

Se presentó desde un punto de vista lógico empleando un sistema axiomático, que los autores denominaron SL_{FD} (*Simplification Logic for Functional Dependencies*) [12].

Algoritmos para el Cálculo de Claves

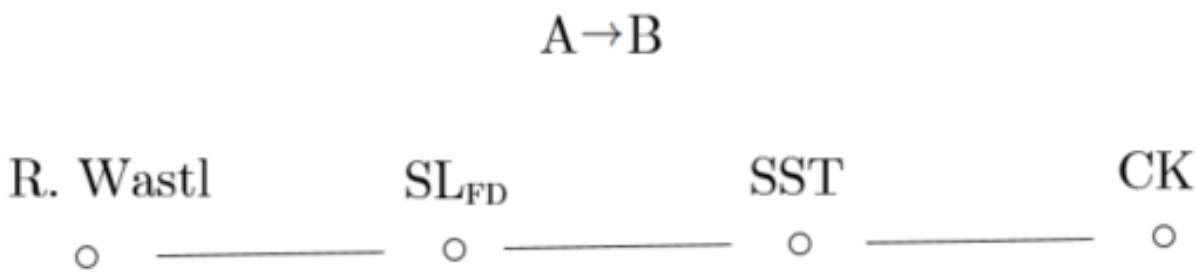
El objetivo de esta parte de la tesis se centra en los algoritmos de búsqueda de claves basados en la lógica, y más específicamente, en aquellos que utilizan el paradigma de tableaux [38, 44] como sistema de inferencia.

Algoritmos para el Cálculo de Claves

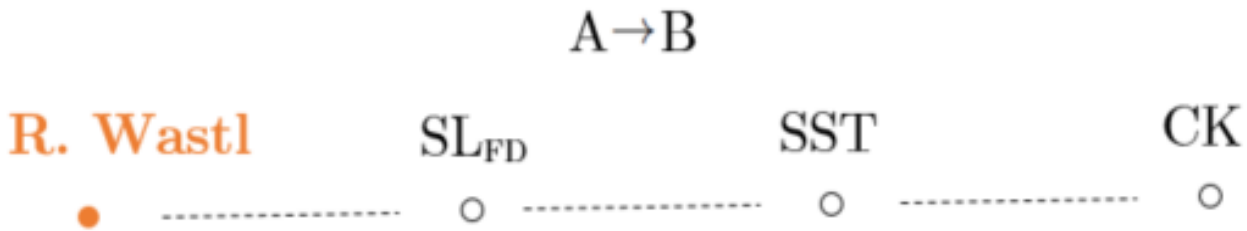
El objetivo de esta parte de la tesis se centra en los algoritmos de búsqueda de claves basados en la lógica, y más específicamente, en aquellos que utilizan el paradigma de tableaux [38, 44] como sistema de inferencia.

- Árbol de búsqueda
- Raiz, ramas, hojas
- Reglas de inferencia
- Versatilidad y comparabilidad

Algoritmos para el Cálculo de Claves



Algoritmos para el Cálculo de Claves



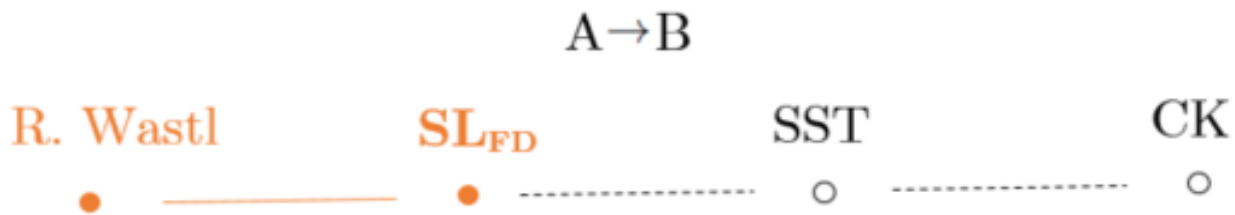
Sistema de inferencia $\mathbb{K}$ [55]

Utiliza dos reglas de inferencia:

- $\mathbb{K}1$ para construir la raiz del arbol
- $\mathbb{K}2$ para construir las sucesivas ramas

Cuando ya no se puede aplicar $\mathbb{K}2$, se alcanzan las hojas, que representan las claves.

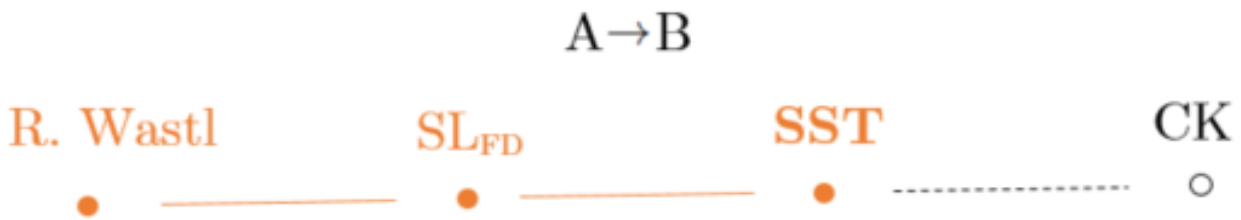
Algoritmos para el Cálculo de Claves



SL_{FD} [11]

Utiliza sistema de inferencia basado en la lógica SL_{FD}.
El espacio de búsqueda generado sobrepasa las capacidades computacionales.

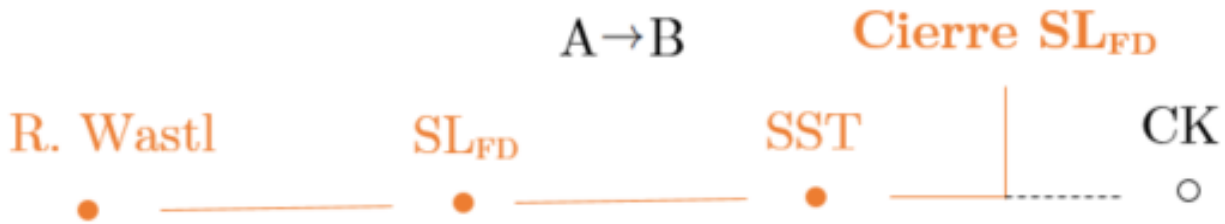
Algoritmos para el Cálculo de Claves



Strong Simplification Tableaux (SST) [13]

Incluye el test de minimalidad para evitar la apertura de ramas adicionales.
Amplio estudio en [4].

Algoritmos para el Cálculo de Claves

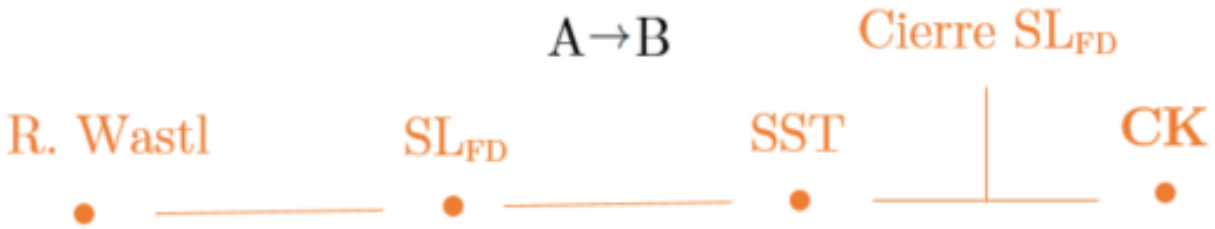


Cierre SL_{FD} [37]

Nuevo operador de cierre que proporciona:

- El conjunto de atributos derivados del cierre.
- Un subconjunto del conjunto original de implicaciones.

Algoritmos para el Cálculo de Claves



Closure Keys (CK)

Aportación de esta tesis.

Eficiente mecanismo de poda para mejorar el rendimiento de sus antecesores.

Implementación secuencial y paralela.

Algoritmos para el Cálculo de Claves

SUPERCOMPUTACIÓN

Supercomputación

1. [5], paralelización del método de Wastl y el algoritmo de claves SL_{FD}.
2. [4], paralelización del método SST.
3. **[6]**, diseño e implementación secuencial y paralela del método CK.

Supercomputación

1. [5], paralelización del método de Wastl y el algoritmo de claves SL_{FD}.
2. [4], paralelización del método SST.
3. **[6]**, diseño e implementación secuencial y paralela del método CK.

Concepto de Generador Minimal

Una forma de representar en FCA el conocimiento es el retículo de conceptos.

Concepto de Generador Minimal

Una forma de representar en FCA el conocimiento es el retículo de conceptos.

↓

Los conjuntos cerrados son la base para la generación del retículo de conceptos.

Concepto de Generador Minimal

Una forma de representar en FCA el conocimiento es el retículo de conceptos.

↓

Los conjuntos cerrados son la base para la generación del retículo de conceptos.

↓

Generadores minimales como representaciones canónicas de cada conjunto cerrado.

[23]

Relevancia

Los generadores minimales son esenciales para obtener una representación completa del conocimiento en FCA [41, 42].

Esta parte de la tesis ha consistido en el estudio y diseño de métodos para la enumeración de todos los conjuntos cerrados y generadores minimales a partir del conjunto de implicaciones.

Métodos para la Enumeración de Generadores

Los métodos desarrollados en esta tesis son una evolución del presentado en [14] que los autores denominaron MinGen.

Se utilizó la lógica SL_{FD} como medio para encontrar todos los generadores minimales a partir de un conjunto de implicaciones.

Claves vs Generadores

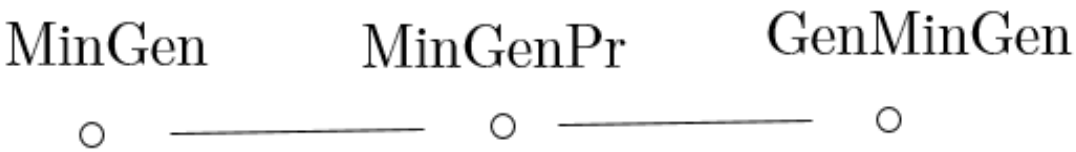
Similitudes

- Lógica SL_{FD}
- Cierre SL_{FD}
- Tableaux

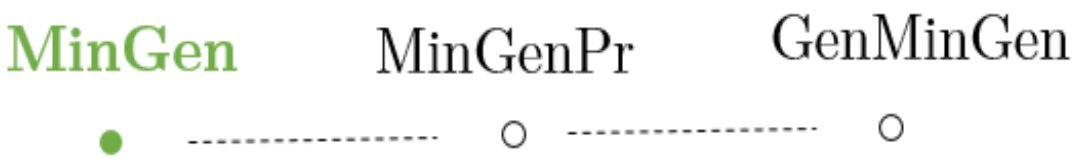
Diferencias

- Soluciones en las hojas vs soluciones parciales en cada nodo
- Paralelización

Métodos para la Enumeración de Generadores



Métodos para la Enumeración de Generadores



Método original, MinGen [14]

- Sistema de inferencia basado en la lógica SL_{FD}.
- Árbol de búsqueda estilo Tableaux.

Métodos para la Enumeración de Generadores

MinGenPr

- Nuevo mecanismo de poda basada en el test de inclusión de conjuntos.
- Comprobación en el mismo nivel del árbol.
- Disminuye redundancias.

Métodos para la Enumeración de Generadores

GenMinGen

- Extiende el mecanismo de poda no sólo al mismo nivel del árbol sino también a las soluciones parciales obtenidas hasta el momento.
- Reduce aún más las redundancias.

Generadores y Paralelismo

[7]

Definición de Sistema de Recomendación

Básicamente, un sistema de recomendación (SR) es un sistema para proponer sugerencias acordes a las preferencias del usuario entre varias alternativas (e.g. películas [26], libros [15], etc.).

Trayectoria y Relevancia

- Campo de estudio en continua evolución [27].
- Importante campo de investigación [18].
- Elemento imprecindible para sólidos entornos comerciales a nivel mundial (Amazon [30], LinkedIn [43], Facebook [51].

Recomendaciones y FCA

Es una relación existente en la literatura desde hace años.

- En [17], los autores utilizan FCA para agrupar elementos y usuarios en conceptos para posteriormente, realizar recomendaciones colaborativas según la afinidad con los elementos vecinos.
- En [62] se propone un SR personalizado basado en el retículo de conceptos para descubrir información valiosa de acuerdo con los requisitos del usuario.
- ...

Tipos de SRs

Existen numerosos tipos de SRs atendiendo a cómo se generan las recomendaciones:

- Filtrado Colaborativo [35] que basan su funcionamiento en valoraciones.
- Basados en contenido [60] que proporcionan resultados que tengan características similares.
- Basados en conocimiento [34] utilizan un método de razonamiento.

La mejor alternativa consiste en generar SRs híbridos.

SRs Conversacionales

Los SRs más importantes para esta tesis son los denominados conversacionales [24].

Utilizan un sistema de interacción con el usuario para guiar el proceso de recomendación.

Problemas de los SRs

- Arranque en frío.
- Escasez.
- Ataques maliciosos.
- Privacidad.
- ... [49].

La Maldición de la Dimensión

Aparece cuando es necesario trabajar sobre *datasets* con un alto número de características [48].

De forma intuitiva:
Cuando hay pocas columnas de datos, los algoritmos de tratamiento de la información (aprendizaje automático, *clustering*, clasificación, etc.) suelen tener un buen comportamiento, sin embargo, a medida que la cantidad de datos aumenta, se hace más difícil recomendar con precisión.

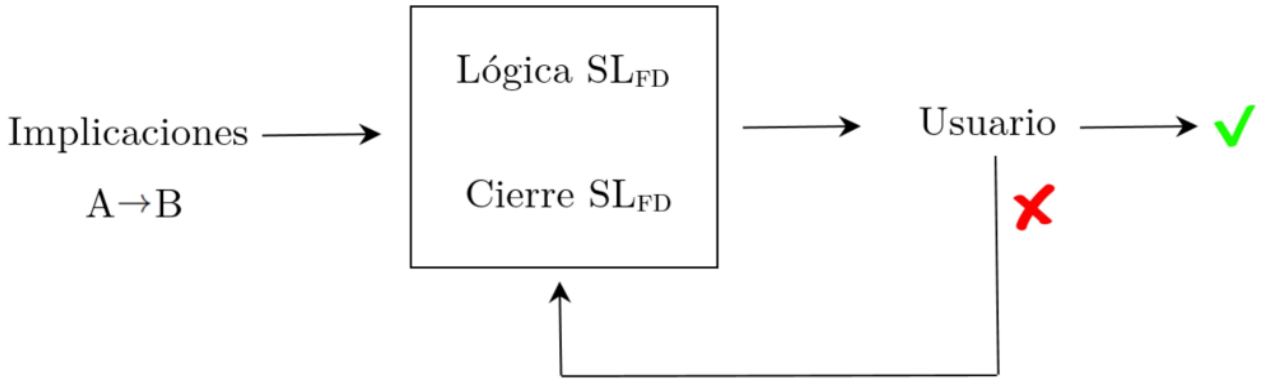
Propuesta Desarrollada

El objetivo principal de esta parte de la tesis ha sido abordar este problema a través de un proceso de selección de atributos por parte del usuario mediante un SR híbrido:

- Basado en contenido
- Basado en conocimiento
- Conversacional

Propuesta Desarrollada

Partiendo del conjunto de implicaciones, utiliza la lógica SL_{FD} y el cierre SL_{FD} como motor para facilitar y acelerar la recomendación.



Propuesta Desarrollada

Objetivos alcanzados:

- Aliviar la sobrecarga de información con la que ha de tratar el usuario del SR.
- Reducir el número de pasos necesarios en el diálogo para obtener una recomendación.

Índice

Contexto Formal

Definición

Un contexto formal es una tripleta $K = (G, M, I)$ que consiste en dos conjuntos no vacíos, G y M, y una relación binaria I entre ellos. Los elementos de G se llaman objetos del contexto, y los elementos de M se llaman atributos del contexto. Para $g \in G$ y $m \in M$, escribimos $< g, m > \in I$ o gIm si el objeto g posee el atributo m.

Ejemplo

Fuente: [22]	Latino Ame.	Europa	Canadá	Asia	Oriente	África	México	Caribe	EE.UU
Air Canada	✓	✓	✓	✓	✓		✓	✓	✓
Air New Zealand		✓		✓					✓
All Nippon Airlines		✓		✓					✓
Ansett Australia				✓					
Austrian Airlines		✓	✓	✓	✓	✓			✓
British Midland		✓							
Lufthansa	✓	✓	✓	✓	✓	✓	✓		✓
Mexicana	✓		✓				✓	✓	✓
Scandinavian Airlines	✓	✓		✓		✓			✓
Singapore Airlines		✓	✓	✓	✓	✓			✓
Thai Airways Int.	✓	✓		✓				✓	✓
Unites Airlines	✓	✓	✓	✓			✓	✓	✓
VARIG	✓	✓		✓		✓	✓		✓

Operadores de Derivación

Definición

Dado un contexto formal $K = (G, M, I)$, se definen los operadores de derivación:

$$(\ )' : 2^G \rightarrow 2^M \qquad (\ )' : 2^M \rightarrow 2^G$$

$$A' = \{m \in M \mid g\ I\ m \quad \forall g \in A\} \qquad B' = \{g \in G \mid g\ I\ m \quad \forall m \in B\}$$

Ejemplo

$$\{Mexicana\}' = \{LatinoAme.,\ Canada,\ Mexico,\ Caribe,\ EE.UU\}$$

Operador de Cierre

Definición

Sea $K = (G, M, I)$ un contexto formal, entonces el operador $()'' : 2^M \rightarrow 2^M$ es un operador de cierre, es decir, satisface las siguientes propiedades:

- Idempotente: $X'''' = X'' \quad \forall X \in 2^M$
- Monótona: $X \subseteq Y \rightarrow X'' \subseteq Y'' \quad \forall X, Y \in 2^M$
- Extensiva: $X \subseteq X'' \quad \forall X \in 2^M$

Ejemplo

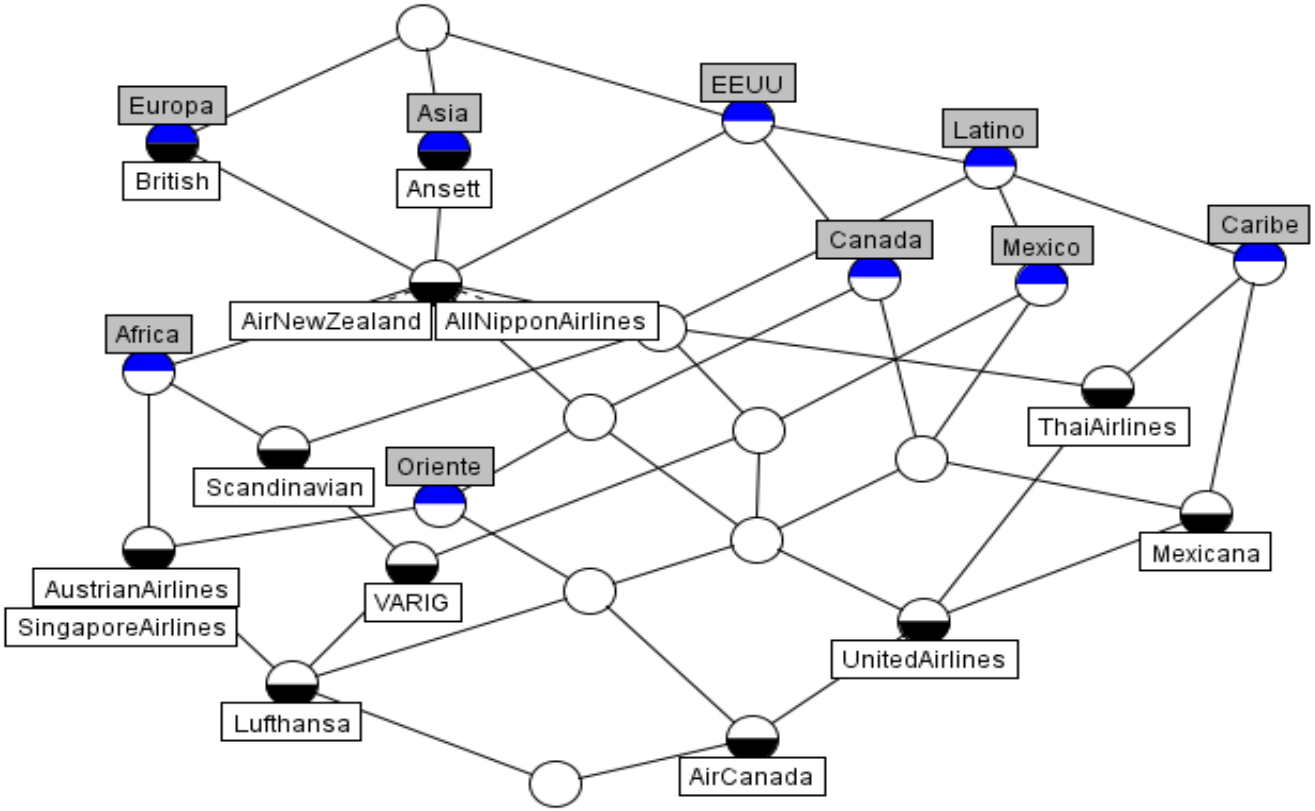
$Z = \{Latinoamerica, Europa, Canada, Asia, Oriente\ Medio, Africa, Mexico, Estados\ Unidos\}$.
Ya que $Z'' = Z$.

Concepto Formal

Definición

Sea $K = (G, M, I)$ un contexto formal y $A \subseteq G$, $B \subseteq M$. El par (A, B) se denomina concepto formal si $A' = B$ y $B' = A$.
El conjunto de objetos A se denomina extensión del concepto (A, B) mientras que el conjunto de atributos B será la intensión del concepto.

Retículo de Conceptos



Implicaciones

- Equivalente al retículo de conceptos
- Eje fundamental de esta tesis
- FCA, BB.DD

Componentes de una Lógica

- Lenguaje
- Semántica
- Sistema axiomatico
- Método de razonamiento automático

Lenguaje y Semántica

Definición

Dado un conjunto M finito de símbolos (denominados atributos) no vacío, el lenguaje sobre M se define como:

$$\mathcal{L}_M = \{A \to B \mid A, B \subseteq M\}$$

Definición

Sea $K = (G, M, I)$ y sea $A \to B \in \mathcal{L}_M$. El contexto K es un modelo para $A \to B$ si $B \subseteq A''$. Se denota por $K \models A \to B$.

$K \models Caribe \to Latinoamerica$
$K \not\models Caribe \to Mexico$

BB.DD Relacionales

- Modelo de base de datos relacional, E. F. Codd, 1970 [9].
- Datos organizada en forma de tablas bidimensionales.
- Una base de datos consiste en una o más relaciones [19] (esquema y cuerpo).

BB.DD Relacionales

- Modelo de base de datos relacional, E. F. Codd, 1970 [9].
- Datos organizada en forma de tablas bidimensionales.
- Una base de datos consiste en una o más relaciones [19] (esquema y cuerpo).

Definición

Un esquema E para una relación R se define como un conjunto fijo de pares *(atributo:dominio)*.

Definición

Un cuerpo C para una relación R se define como un conjunto de tuplas de pares *(atributo:valor)*.

Dependencias Funcionales

Diseño defectuoso de la BB.DD relacional

↓

Anomalías [54]

↓

Descomposición de relacionales (normalización)

↓

Dependencias funcionales (Formas normales)

Dependencias Funcionales

- Misma sintaxis que implicaciones en FCA.
- Diferente semántica.

Dependencias Funcionales

- Misma sintaxis que implicaciones en FCA.
- Diferente semántica.

Definición

Una dependencia funcional $X \rightarrow Y$ se cumple en una tabla R si y sólo si para cada dos tuplas de R, si sus valores en X coinciden, entonces también coinciden sus valores en Y.

Implicaciones y DFs: diferente semántica

	a	b	c
g_1	1	0	1
g_2	1	1	1
g_3	0	0	0
g_4	0	1	1

Se verifica la implicación a $\rightarrow$ c.

No se verifica la DF a $\rightarrow$ c, debido a los valores de g_3 y g_4

Axiomas de Armstrong

Primer sistema axiomático descrito para tratar sistemas de implicaciones utilizando la lógica [1].
Esquema de axioma, denominado *Reflexivo*:

$$\texttt{[Ref]}\ \frac{}{AB \rightarrow A}$$

y dos reglas de inferencia, *Aumentativa* y *Transitiva* que se definen como:

$$\texttt{[Aug]}\ \frac{A \rightarrow B}{AC \rightarrow BC} \qquad \texttt{[Tran]}\ \frac{A \rightarrow B,\ B \rightarrow C}{A \rightarrow C}$$

Noción de *Deducción* ($\vdash$)

Se dice que una implicación $A \rightarrow B$ se deriva sintácticamente (o se deduce) de un sistema de implicaciones Σ, y se denota por $\Sigma \vdash A \rightarrow B$, si existe una secuencia de implicaciones $\sigma_1, \ldots, \sigma_n \in \mathcal{L}_M$ tal que $\sigma_n = A \rightarrow B$ y, para todo $1 \leq i \leq n$, la implicación σ_i satisface una de las siguientes condiciones:

- σ_i es un axioma, es decir, verifica el esquema `[Ref]`.
- $\sigma_i \in \Sigma$.
- σ_i se obtiene a partir de implicaciones pertenecientes a $\{\sigma_j \mid 1 \leq j < i\}$ aplicando las reglas de inferencia del sistema axiomático.

Axiomas de Armstrong

Sistema axiomático correcto y completo tanto para implicaciones como DFs.

No es adecuado para el razonamiento automático debido a su fuerte dependencia a la transitividad.

Lógica de Simplificación, SL_{FD}

- No toma el paradigma de la transitividad como centro sino que se guía por la idea de simplificar el conjunto de implicaciones mediante la eliminación de atributos redundantes de manera eficiente [12].
- Abrió la puerta al desarrollo de métodos de razonamiento automatizados.
- Motivación principal para el desarrollo de esta tesis.

Sistema Axiomático, SL_{FD}

SL_{FD} tiene el siguiente esquema de axioma:

$$\texttt{[Ref]} \quad \frac{}{AB \to A}$$

Y reglas de inferencia, denominadas *fragmentación*, *composición* y *simplificación* respectivamente.

$$\texttt{[Frag]} \ \frac{A \to BC}{A \to B} \qquad \texttt{[Comp]} \ \frac{A \to B,\ C \to D}{AC \to BD}$$

$$\texttt{[Simp]} \ \text{Si } A \subseteq C, A \cap B = \varnothing, \ \frac{A \to B,\ C \to D}{A(C \setminus B) \to D}$$

Lógica de Simplificación, SL_{FD}

(También) Sistema axiomático correcto y completo tanto para implicaciones como DFs [12].

Las reglas de inferencia pueden considerarse reglas de equivalencia.

Núcleo principal para el desarrollo de métodos automáticos para diversas aplicaciones.

Problema de la Implicación

Dado un conjunto de implicaciones Σ, determinar mediante la lógica si a partir de ese conjunto se puede inferir la validez de una nueva implicación dada $\Sigma \vdash A \rightarrow B$ [2, 36, 33].

Cierre de Atributos

Dado un conjunto $X \subseteq M$, se denomina cierre de X sobre Σ (notado X_Σ^+) como el mayor subconjunto de M tal que $\Sigma \vdash X \rightarrow X_\Sigma^+$.

Se utiliza el cierre sintáctico a modo de demostrador automático, pues debido a la definición de cerrado anterior: $\Sigma \vdash A \rightarrow B$ si y sólo si $B \subseteq A_\Sigma^+$

Cierre Clásico de Maier [32]

1. Recibe un conjunto de atributos $X \subseteq M$ y un conjunto de implicaciones $\Sigma \subseteq \mathcal{L}_M$.
2. Incluye nuevos atributos de forma iterativa mediante la relación de subconjunto.
3. La complejidad del problema del cierre es $O(|A|\,|\Sigma|)$ [39].

Algoritmo del Cierre de Maier

Algoritmo 2.1: Cierre clásico

Entrada: Σ, A
Salida: A^+_Σ

```
1  inicio
2      A+Σ := A
3      repetir
4          A' := A+Σ
5          para cada X → Y ∈ Σ hacer
6              si (X ⊆ A+Σ) y (Y ⊈ A+Σ) entonces
7                  A+Σ := A+Σ ∪ {Y}
8      hasta A+Σ = A'
9      devolver A+Σ
```

Cierre SL_{FD}

- Presentado por los autores en [37].
- Novedad: la salida no es sólo el conjunto cerrado sino también un conjunto de implicaciones que puede ser interpretado como el conocimiento que resta en el sistema.
- Evita el coste de extraer el nuevo conjunto de implicaciones tras cada aplicación.
- Explotado ampliamente en los resultados obtenidos a lo largo de esta tesis doctoral.
- Se basa en el Teorema de la deducción y un conjunto de equivalencias.

Cierre SL_{FD}

Teorema de la deducción

Sea $A \to B \in \mathcal{L}_M$ y $\Sigma \subseteq \mathcal{L}_M$. Entonces,

$$\Sigma \vdash A \to B \quad \text{si y sólo si} \quad \{\varnothing \to A\} \cup \Sigma \vdash \{\varnothing \to B\}$$

Cierre SL_{FD}

Teorema de la deducción

Sea $A \to B \in \mathcal{L}_M$ y $\Sigma \subseteq \mathcal{L}_M$. Entonces,

$$\Sigma \vdash A \to B \quad \text{si y sólo si} \quad \{\varnothing \to A\} \cup \Sigma \vdash \{\varnothing \to B\}$$

Proposición

Sean $A, B, C, D \subseteq M$. Se verifican las siguientes equivalencias:

1. $\{A \to B\} \equiv \{A \to B \setminus A\}$
2. $\{A \to B, A \to C\} \equiv \{A \to B \cup C\}$
3. $\{A \to B, C \to D\} \equiv \{A \to B, C \setminus B \to D \setminus B\}$ siendo $A \cap B = \varnothing$ y $A \subseteq C$

Cierre SL_{FD}

Mediante el Teorema de la deducción y las equivalencias anteriores, se puede pasar de las reglas de inferencia del sistema axiomático clásico de Armstrong a un sistema que puede ser automatizable.

Así aparece el cierre SL_{FD}, que los autores denominaron `Cls` [37].

- **Eq. I**: Si $B \subseteq A$ entonces $\{\emptyset \rightarrow A, B \rightarrow C\} \equiv \{\emptyset \rightarrow A \cup C\}$.
- **Eq. II**: Si $C \subseteq A$ entonces $\{\emptyset \rightarrow A, B \rightarrow C\} \equiv \{\emptyset \rightarrow A\}$.
- **Eq. III**: En otro caso $\{\emptyset \rightarrow A, B \rightarrow C\} \equiv \{\emptyset \rightarrow A, B \setminus A \rightarrow C \setminus A\}$.

Algoritmo Cls

Algoritmo 2.2: Cierre Cls

Entrada: Σ, A
Salida: A^+_Σ, Σ'

```
1  inicio
2      A+Σ := A
3      Σ' := Σ
4      repetir
5          A' := A+Σ
6          para cada B → C ∈ Σ hacer
7              si B ⊆ A+Σ entonces
8                  (Eq. I)
9                  A+Σ := A+Σ ∪ {B}
10                 Σ' := Σ' ∖ B → C
11             si no si C ⊆ A+Σ entonces
12                 (Eq. II)
13                 Σ' := Σ' ∖ B → C
14             si no si (B ∩ A+Σ ≠ ∅) o (C ∩ A+Σ ≠ ∅) entonces
15                 (Eq. III)
16                 Σ' := Σ' ∪ {B ∖ A+Σ → C ∖ A+Σ}
17     hasta A+Σ = A'
18     devolver A+Σ, Σ'
```

Índice

CLAVES MINIMALES

Sumario

- Implicaciones, lógica SL_{FD}, cierre SL_{FD}
- Nuevo algoritmo CK, superando al anterior SST
- Implementación secuencial y paralela en entorno de supercomputación

Algoritmo SST

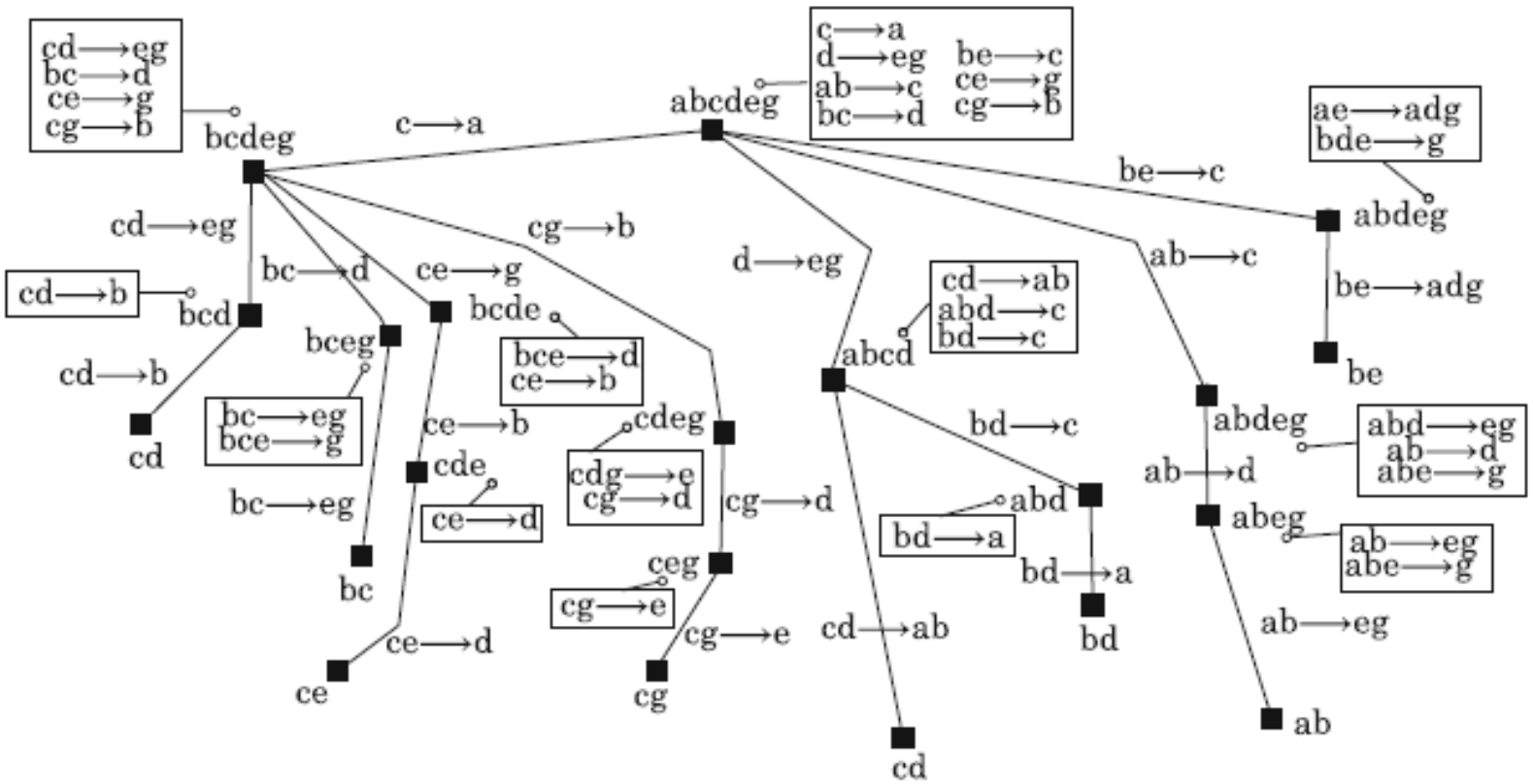


Figura 1: Ejemplo de tableaux utilizando el método SST.

Algoritmo CK

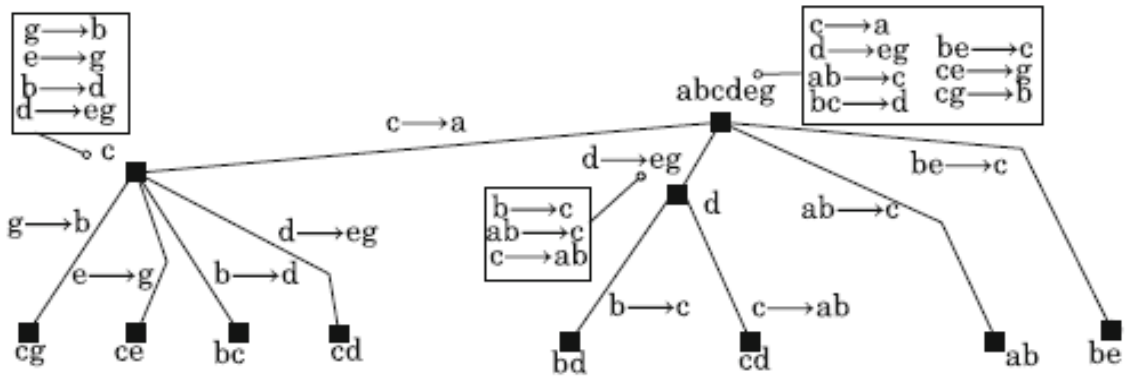


Figura 2: Ejemplo de tableaux utilizando el método CK.

El tamaño del árbol se reduce de 21 a 11 nodos

Paralelismo

Los métodos de búsqueda de claves pueden producir unos resultados de dimensiones inadmisibles para las máquinas actuales (miles de millones de nodos y tiempos de ejecución de días [3, 5].

Aprovechar el diseño del Tableaux para dividir el problema original en instancias atómicas que puedan resolverse por separado en un tiempo razonable.

Implementación Paralela

1. **Etapa de división (secuencial).** Ejecuta el método de búsqueda de claves, en vez de construir el árbol completo, se detendrá en un cierto nivel determinado según un valor de corte generando un conjunto de sub-problemas; uno por nodo del árbol en ese nivel. El resultado de esta etapa es un conjunto de problemas reducidos.
2. **Etapa de resolución (paralela).** Ejecuta simultánemente el algoritmo de búsqueda de claves sobre cada uno de los sub-problemas generados en la etapa anterior y, finalmente, combina las soluciones para obtener todas las claves minimales.

Implementación Paralela

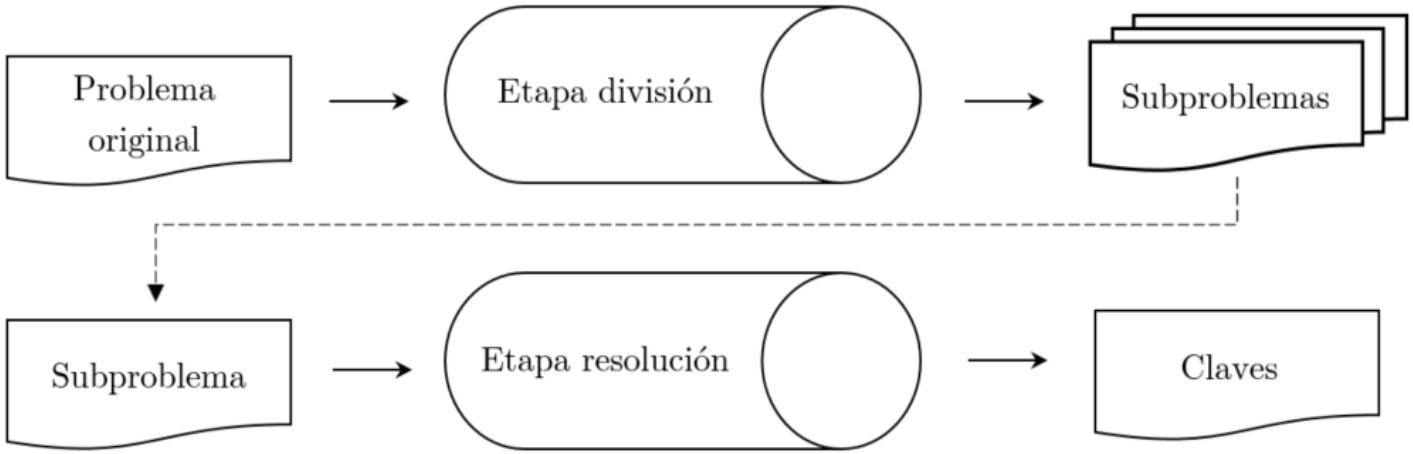


Figura 3: Esquema general de implementación paralela.

Implementación Paralela

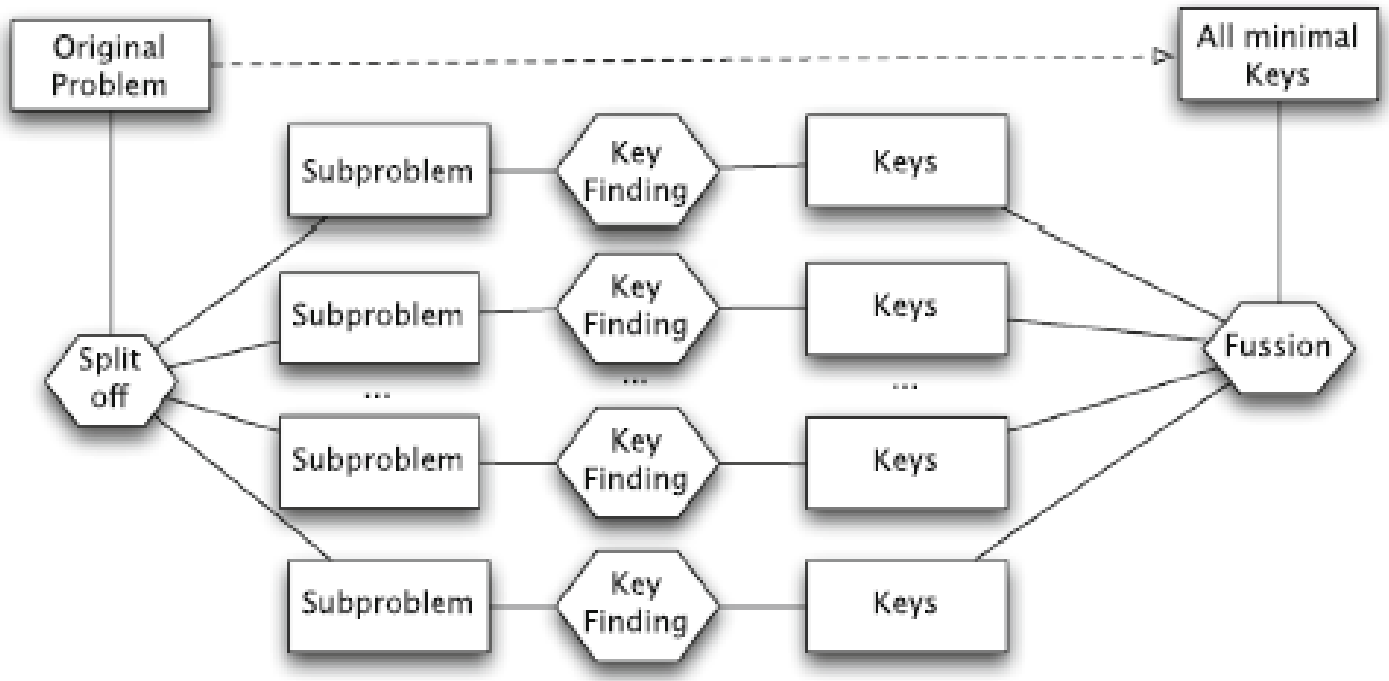


Figura 4: Algoritmo paralelo para claves minimales según el paradigma *MapReduce* [16].

Valor de parada *(BOV, Break-Off Value)*

Es el valor que decide en qué momento el algoritmo paralelo pasa de la etapa secuencial a la paralela. Es realmente difícil aproximar cuál es el mejor valor en cada experimento.

Valor de parada *(BOV, Break-Off Value)*

Es el valor que decide en qué momento el algoritmo paralelo pasa de la etapa secuencial a la paralela. Es realmente difícil aproximar cuál es el mejor valor en cada experimento.

	BOV	
	Cercano a la raiz	Lejano a la raiz
Tiempo división	↓	↑
No. de subproblemas	↓	↑
Paralelismo	↓	↑

Experimentos

Se realizan dos bloques principales de experimentos:

	Atributos	Implicaciones	BOV	Núcleos
Experimento A	100	100	90	32
Experimento B	150	150	140	32

Experimentos

Se realizan dos bloques principales de experimentos:

	Atributos	Implicaciones	BOV	Núcleos
Experimento A	100	100	90	32
Experimento B	150	150	140	32

Estos números van más allá de las capacidades de máquinas convencionales [11], e incluso mejoran sustancialmente los resultados dados en [5], donde ya se aplicaron técnicas paralelas.

Experimentos

Para la comparación de resultados se utilizan dos parámetros fundamentales:

- Tiempos de ejecución
- Número de nodos del árbol

Asimismo, debido a la naturaleza intrínseca del tiempo de ejecución, cada experimento se ha repetido 50 veces para poder obtener valores medios fiables.

Arquitectura

La arquitectura y recursos de supercomputación utilizados son:

- 32 nodos cluster SL230, 32 núcleos
- 64GB de memoria RAM
- Red Infiniband FDR

Facilitados por el Centro de Supercomputación y Bioinnovación de la Universidad de Málaga[1].

[1] www.scbi.uma.es

Resultados Experimento A

Problema & Método	Subp	División$_t$(s)	Total$_t$(s)	Nodos	Ratio
100100-1-*SST*	14	0	1	33	33
100100-1-*CK*	0	0	0	15	15
100100-2-*SST*	1.354	36	105	25.621	244
100100-2-*CK*	212	4	15	12.715	847
100100-3-*SST*	8.602	183	644	192.574	299
100100-3-*CK*	1.286	37	99	94.255	952
100100-4-*SST*	400	7	26	1.704	65
100100-4-*CK*	15	1	2	751	375
100100-5-*SST*	39	0	2	119	59
100100-5-*CK*	0	0	1	42	42
100100-6-*SST*	1.808	37	123	7.856	63
100100-6-*CK*	115	4	9	3.698	410
100100-7-*SST*	6.167	182	489	275.429	563
100100-7-*CK*	1.378	24	90	118.884	1.320
100100-8-*SST*	5.104	146	415	182.167	438
100100-8-*CK*	1.014	19	68	81.632	1.200
100100-9-*SST*	314	11	25	868	34
100100-9-*CK*	0	1	1	341	341

Resultados Experimento B

Problema & Método	Subp	División$_t$(s)	Total$_t$(s)	Nodos	Ratio
150150-1-*SST*	165	6	14	911	65
150150-1-*CK*	11	2	3	374	124
150150-2-*SST*	2.949	229	394	116.517	295
150150-2-*CK*	347	25	44	54.375	1.235
150150-3-*SST*	12.968	1.049	1.716	157.947	92
150150-3-*CK*	822	125	165	68.531	415
150150-4-*SST*	5.352	581	885	55.211	62
150150-4-*CK*	344	48	65	25.477	391
150150-5-*SST*	5.361	211	484	32.377	66
150150-5-*CK*	168	27	36	12.522	347
150150-6-*SST*	771	72	155	17.298	111
150150-6-*CK*	79	7	11	8.110	737
150150-7-*SST*	9.473	638	1.252	576.912	460
150150-7-*CK*	1.754	97	187	262.621	1.404
150150-8-*SST*	5.466	424	857	510.627	595
150150-8-*CK*	966	57	104	257.267	2.473
150150-EXTRA-*SST*	31.401	2.950	30.983	21.404.732	690
150150-EXTRA-*CK*	8.049	354	1.320	10.614.386	8.041

Resultados Experimento B

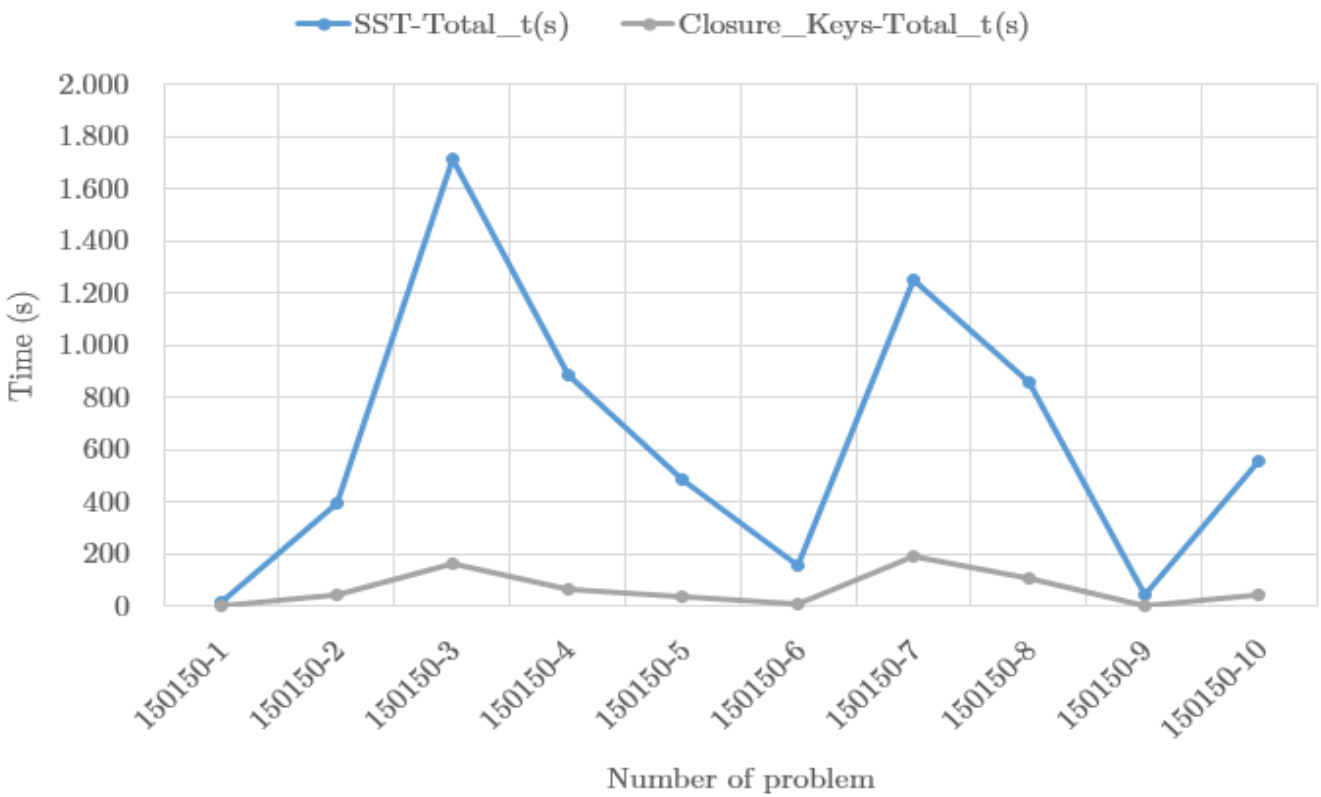


Figura 5: Tiempos de ejecución de los métodos paralelos aplicados a problemas grandes.

Resultados Experimento B

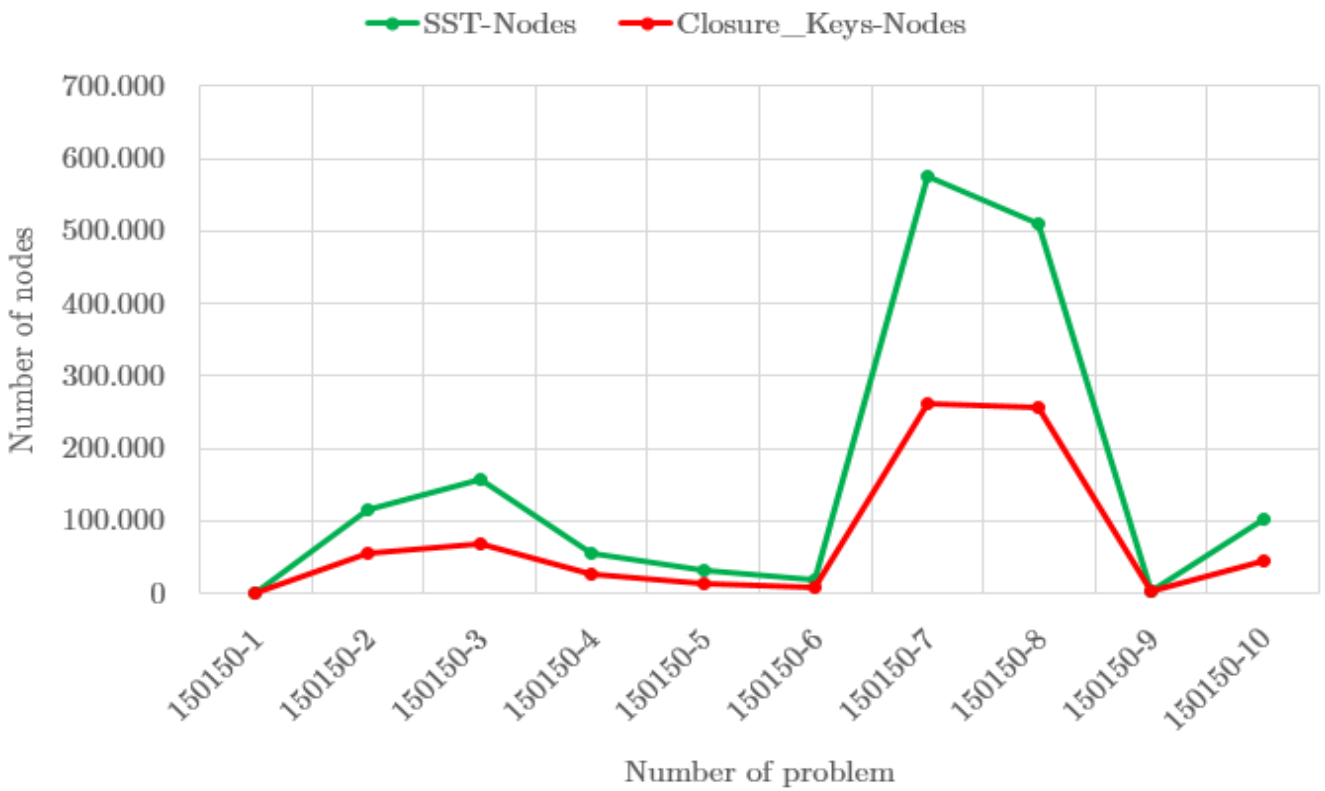


Figura 6: Número de nodos del árbol de los métodos paralelos aplicados a problemas grandes.

GENERADORES MINIMALES

Sumario

- Implicaciones, lógica SL_{FD}, cierre SL_{FD}
- Diseño, implementación y comparación de los métodos MinGenPr y GenMinGen a partir de MinGen
- Implementación secuencial y paralela en entorno de supercomputación, método MinGenPar
- Pruebas de optimización

Ejemplo

Sea el conjunto de implicaciones inicial:

$$\Sigma = \{a \rightarrow c, bc \rightarrow d, c \rightarrow ae, d \rightarrow e\}$$

Se va a aplicar los métodos de enumeración de generadores minimales (MinGen, MinGenPr, GenMinGen) sobre el ejemplo.

Ejemplo

Al aplicar el método MinGen sobre Σ se obtiene el siguiente resultado:

X	$\varnothing$	b	e	be	ace	$acde$	$abcde$	de	bde
$mg_\Sigma(X)$	$\varnothing$	b	e	be	a	ad	ab	d	bd
					c	cd	bc		

Y el árbol de búsqueda que se va generando se puede ver en la siguiente figura.

Método MinGen [14]

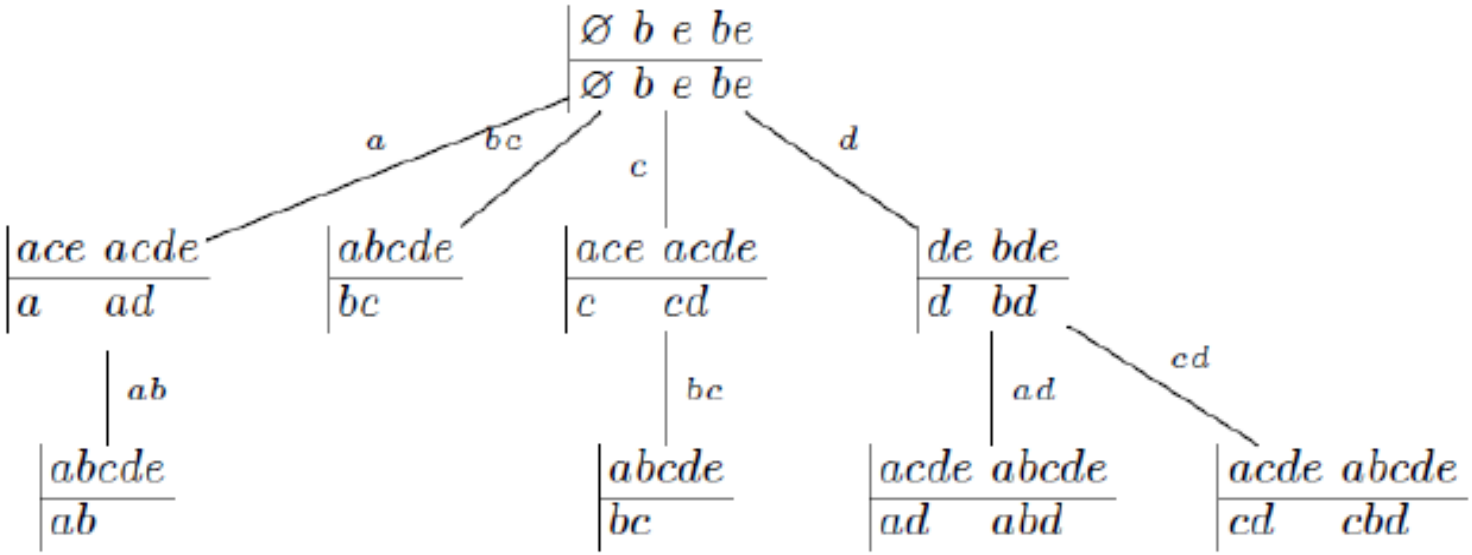


Figura 7: Árbol de búsqueda obtenido como resultado de aplicar el método MinGen sobre Σ.

Traza de Ejecución I

MinGen($abcde, \varnothing, \varnothing, \{a \to c, bc \to d, c \to ae, d \to e\}$):

$$\text{Cls}(\varnothing, \Sigma) = (\varnothing, \{a \to c, bc \to d, c \to ae, d \to e\})$$

X	$\varnothing$	b	e	be
$mg_{\Sigma}(X)$	$\varnothing$	b	e	be

1 MinGen($abcde, a, a, \{a \to c, bc \to d, c \to ae, d \to e\}$):

$$\text{Cls}(a, \{a \to c, bc \to d, c \to ae, d \to e\}) = (ace, \{b \to d\})$$

X	ace	$acde$
$mg_{\Sigma_1}(X)$	a	ad

1.1 MinGen($bd, ab, abce, \{b \to d\}$):

$$\text{Cls}(abce, \{b \to d\}) = (abcde, \varnothing)$$

X	$abcde$
$mg_{\Sigma_{1.1}}(X)$	ab

Traza de Ejecución II

Volvemos al nivel 1: $mg_{\Sigma_1} := mg_{\Sigma_1} \sqcup mg_{\Sigma_{1.1}}$

X	ace	$acde$	$abcde$
$mg_{\Sigma_1}(X)$	a	ad	ab

Volvemos al nivel raiz: $mg_{\Sigma} := mg_{\Sigma} \sqcup mg_{\Sigma_1}$

X	$\varnothing$	b	e	be	ace	$acde$	$abcde$
$mg_{\Sigma}(X)$	$\varnothing$	b	e	be	a	ad	ab

Ejemplo

Después de desplegar todo el árbol se obtiene el resultado que se adelantó:

X	$\varnothing$	b	e	be	ace	$acde$	$abcde$	de	bde
$mg_\Sigma(X)$	$\varnothing$	b	e	be	a	ad	ab	d	bd
					c	cd	bc		

Método MinGenPr

Realiza una poda mediante test de inclusión en los nodos de un mismo nivel del árbol.

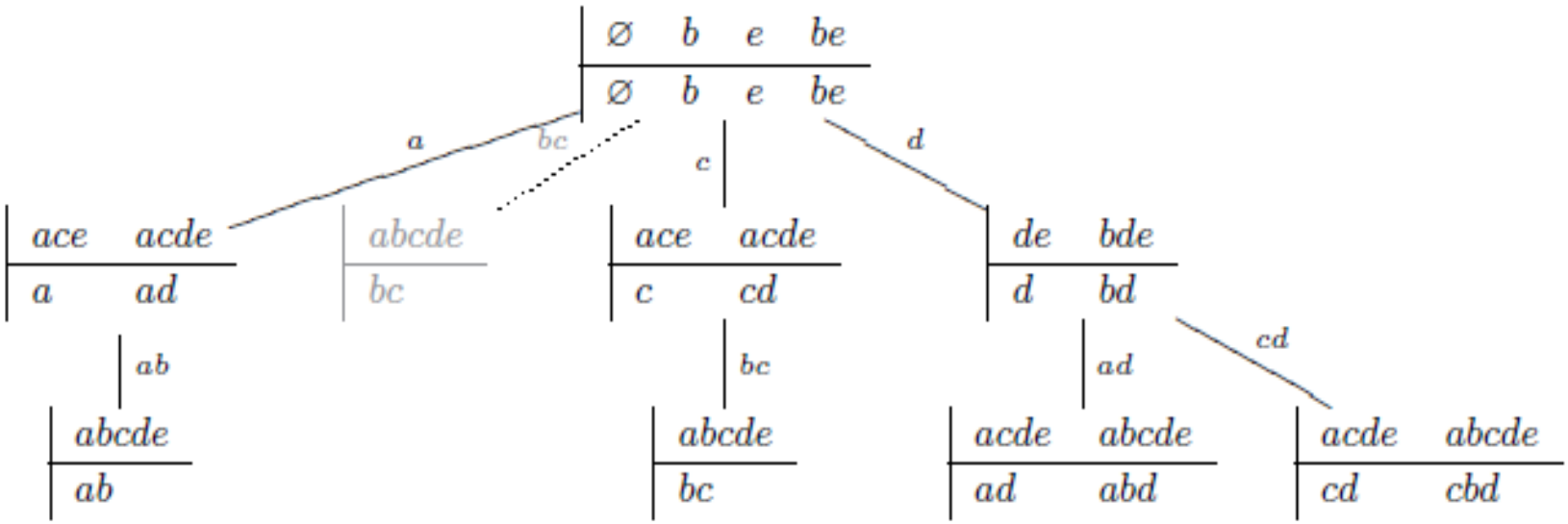


Figura 8: Árbol de búsqueda obtenido como resultado de aplicar el método MinGenPr sobre Σ.

Método GenMinGen

Generaliza el mecanismo de poda a todos los resultados parciales obtenidos hasta el momento.

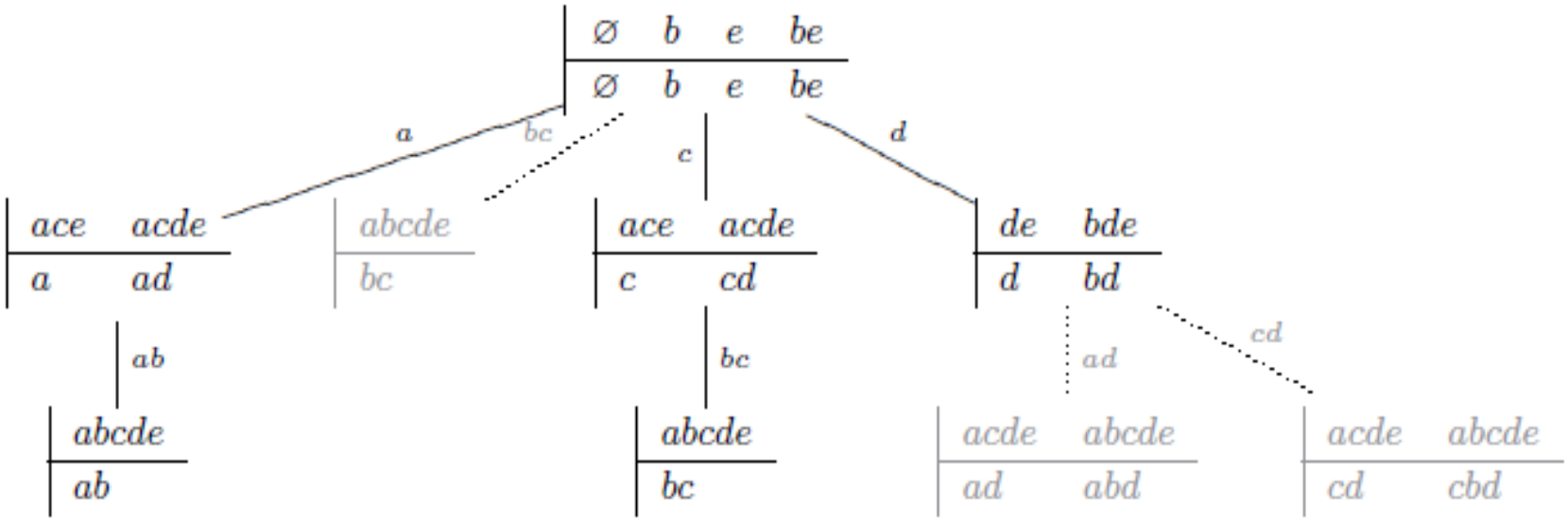


Figura 9: Árbol de búsqueda obtenido como resultado de aplicar el método GenMinGen sobre Σ.

Experimentos

Para las pruebas de los métodos de enumeración de generadores minimales, se llevan a cabo dos bloques de experimentos diferentes:

1. Secuenciales
2. Paralelos

Experimentos Secuenciales

El primer paso consiste en la comparación de resultados entre los diferentes métodos, utilizando conjuntos de datos que puedan gestionarse de forma local.

Para ello, se han utilizado:

- 5 ficheros con datos sintéticos. 50 impl. y 50 atrib. (claves)
- Datos reales (MovieLens 10M, 249 impl. y 19 atrib.)

Experimentos Secuenciales

Problema y Método	Total_t(s)	Nodos	MinGens
sequential-1-MinGen	489	35.062	1.437
sequential-1-MinGenPr	108	7.734	1.437
sequential-1-GenMinGen	97	6.714	1.437
sequential-2-MinGen	53	32.408	271
sequential-2-MinGenPr	8	4.670	271
sequential-2-GenMinGen	7	3.922	271
sequential-3-MinGen	41	7.518	688
sequential-3-MinGenPr	9	1.642	688
sequential-3-GenMinGen	9	1.611	688

...

Experimentos Secuenciales

Problema y Método	Total$_t$(s)	Nodos	MinGens
	...		
sequential-4-MinGen	693	52.067	1.444
sequential-4-MinGenPr	179	12.802	1.444
sequential-4-GenMinGen	148	11.014	1.444
sequential-5-MinGen	10.647	496.521	2.941
sequential-5-MinGenPr	1.897	72.470	2.941
sequential-5-GenMinGen	1.071	42.957	2.941
MovieLens10M-MinGen	980	254.170	2.681
MovieLens10M-MinGenPr	210	19.187	2.681
MovieLens10M-GenMinGen	198	18.926	2.681

Justificación MinGenPar

Al igual que en el caso de la búsqueda de claves minimales, cuando se quiere realizar una enumeración de generadores minimales a partir de un conjunto de implicaciones de un tamaño considerable, las máquinas convencionales no son suficiente.

Por tanto, además del diseño de los nuevos métodos y su implementación secuencial, se ha realizado una implementación paralela del método MinGenPr, denominada MinGenPar.

Justificación MinGenPar

- Se utiliza la misma estrategia de computación paralela (MapReduce) que en el caso de claves minimales
- MinGenPr mejora a MinGen...
- ...pero GenMinGen es superior a MinGenPr, ¿entonces?

Arquitectura

La arquitectura y recursos de supercomputación utilizados son:

- 32 nodos cluster SL230, 32 núcleos
- 64GB de memoria RAM
- Red Infiniband FDR
- ---
- 7 nodos cluster DL980, 80 núcleos
- 2TB de memoria RAM
- Red Infiniband QDR

Facilitados por el Centro de Supercomputación y Bioinnovación de la Universidad de Málaga[2].

[2] www.scbi.uma.es

Experimentos Paralelos

Se han utilizado 10 ficheros con información sintética que contienen 150 implicaciones y 150 atributos (3 veces más que en los experimentos secuenciales).

Los experimentos realizados para la versión paralela son:

1. Comparación de la versión secuencial MinGenPr con la versión paralela MinGenPar
2. Cálculo del valor de corte
3. Estimación del número de núcleos
4. Prueba sobre información real

Experimentos Paralelos I

Tabla 1: Comparación entre las versiones secuencial (MinGenPr) y paralela (MinGenPar) aplicado a problemas de gran tamaño.

Prob	Sec_t(s)	Subp	$División_t$(s)	$Paralelo_t$(s)	$Total_t$(s)	Nodos	MinGens
#1	43	11	3	1	4	374	216
#2	17.352	347	220	55	275	54.375	6.273
#3	33.338	822	2.338	251	2.589	68.531	6.529
#4	4.612	344	350	97	447	25.477	2.478
#5	1.585	168	432	30	462	12.522	1.159
#6	1.653	79	35	7	42	8.110	1.436
#7	107.238	1.754	958	242	1.200	262.621	9.113
#8	61.381	966	253	188	441	257.267	5.538
#9	372	24	7	2	9	1.726	683
#10	7.484	277	186	65	251	45.962	2.969

Experimentos Paralelos II

Tabla 2: Experimentos utilizando diferentes valores de *BOV* con la implementación paralela *MinGenPar* sobre problemas de gran tamaño.

Problema	Subp	División$_t$(s)	Paralelo$_t$(s)	Total$_t$(s)	BOV
#1	0	44	-	44	66.67 %
	0	41	-	41	86.67 %
	11	3	1	4	93.33 %
#2	0	18.533	-	18.533	66.67 %
	885	8.532	58	8.590	86.67 %
	347	220	55	275	93.33 %
#3	0	33.377	-	33.377	66.67 %
	0	33.285	-	33.285	86.67 %
	822	2.338	251	2.589	93.33 %

...

Experimentos Paralelos II

Tabla 3: Continuación...

Problema	Subp	División$_t$(s)	Paralelo$_t$(s)	Total$_t$(s)	BOV
			...		
#7	0	167.451	-	167.451	66.67 %
	5.412	96.433	295	96.728	86.67 %
	1.754	958	242	1.200	93.33 %
#8	0	75.060	-	75.060	66.67 %
	5.344	41.404	375	41.779	86.67 %
	966	253	188	441	93.33 %
#9	0	82	-	82	66.67 %
	24	42	2	44	86.67 %
	24	7	2	9	93.33 %

Experimentos Paralelos III

Tabla 4: Tiempos de ejecución (en segundos) de los resultados obtenidos incrementando el número de cores para la ejecución en paralelo.

Problema	Número de cores				
	16	32	48	64	80
#1	5	4	3	3	3
#2	310	275	199	190	197
#3	2.742	2.589	2.122	2.078	2.075
#4	512	447	444	440	446
#5	598	462	416	445	442
#6	50	42	34	35	34
#7	1.457	1.200	1.078	1.010	1.012
#8	499	441	432	430	425
#9	9	9	7	7	7
#10	267	251	196	194	195

Experimentos Paralelos IV

Se ha elegido elegido *Mushroom Data Set*[3] accesible desde el sitio web de la Universidad de California, Irvine (UCI)[4].

Tabla 5: Algoritmo MinGenPar aplicado sobre información real.

	Atrib	Implicaciones	BOV	Cores		
	126	1.587	1.481	64		
Problema	Subp	División$_t$(s)	Paralelo$_t$(s)	Total$_t$(s)	Nodos	MinGens
mushrooms	224	152	9	161	81.363	17.127

[3] https://archive.ics.uci.edu/ml/datasets/mushroom
[4] http://archive.ics.uci.edu/ml/

SISTEMAS DE RECOMENDACIÓN

Sumario

- Implicaciones, lógica SL_{FD}, cierre SL_{FD}
- Sistema de recomendación híbrido
  - Basado en contenido
  - Basado en conocimiento
  - Conversacional
- Problema de la maldición de la dimensionalidad

SR basado en contenido

El sistema necesita información de base sobre la que poder trabajar...

Tabla 6: Extracto del *dataset* Auto MPG Data Set de la UCI

Nombre	MPG	Cilindrada	Potencia	Peso	Aceleración
Chevrolet Monte Carlo	15.0	8	150.0	3761	9.5
Buick Estate Wagon	14.0	8	225.0	3086	10.0
Toyota Corolla Mark II	24.0	4	95.0	2372	15.0
Plymouth Duster	22.0	6	95.0	2833	15.5
...					

SR basado en contenido

...pero además, esa información ha de presentarse mediante una representación binaria.

Tabla 7: Extracto del *dataset* Auto MPG Data Set de la UCI

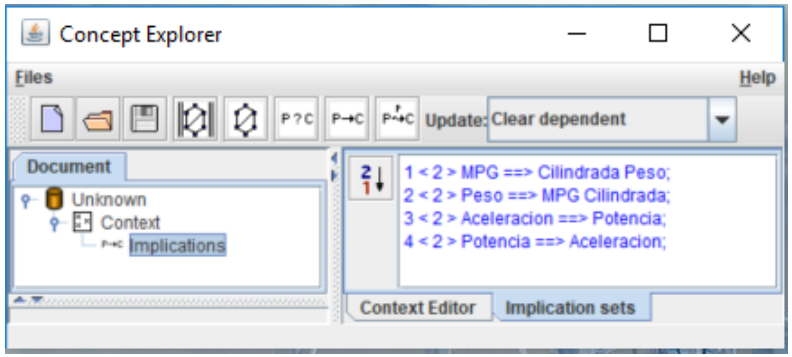
Nombre	MPG	Cilindrada	Potencia	Peso	Aceleración
Chevrolet Monte Carlo	15.0	8	150.0	3761	9.5
Buick Estate Wagon	14.0	8	225.0	3086	10.0
Toyota Corolla Mark II	24.0	4	95.0	2372	15.0
Plymouth Duster	22.0	6	95.0	2833	15.5
...					
Chevrolet Monte Carlo	1	1	0	1	0
Buick Estate Wagon	1	1	0	1	0
Toyota Corolla Mark II	0	0	1	0	1
Plymouth Duster	0	1	1	0	1
...					

SR basado en conocimiento

De la información de entrada, debe obtenerse el conjunto de implicaciones, para lo cual existen varios trabajos (y sus respectivas implementaciones) en la literatura que se encargan de ello [28, 57, 58].

SR basado en conocimiento

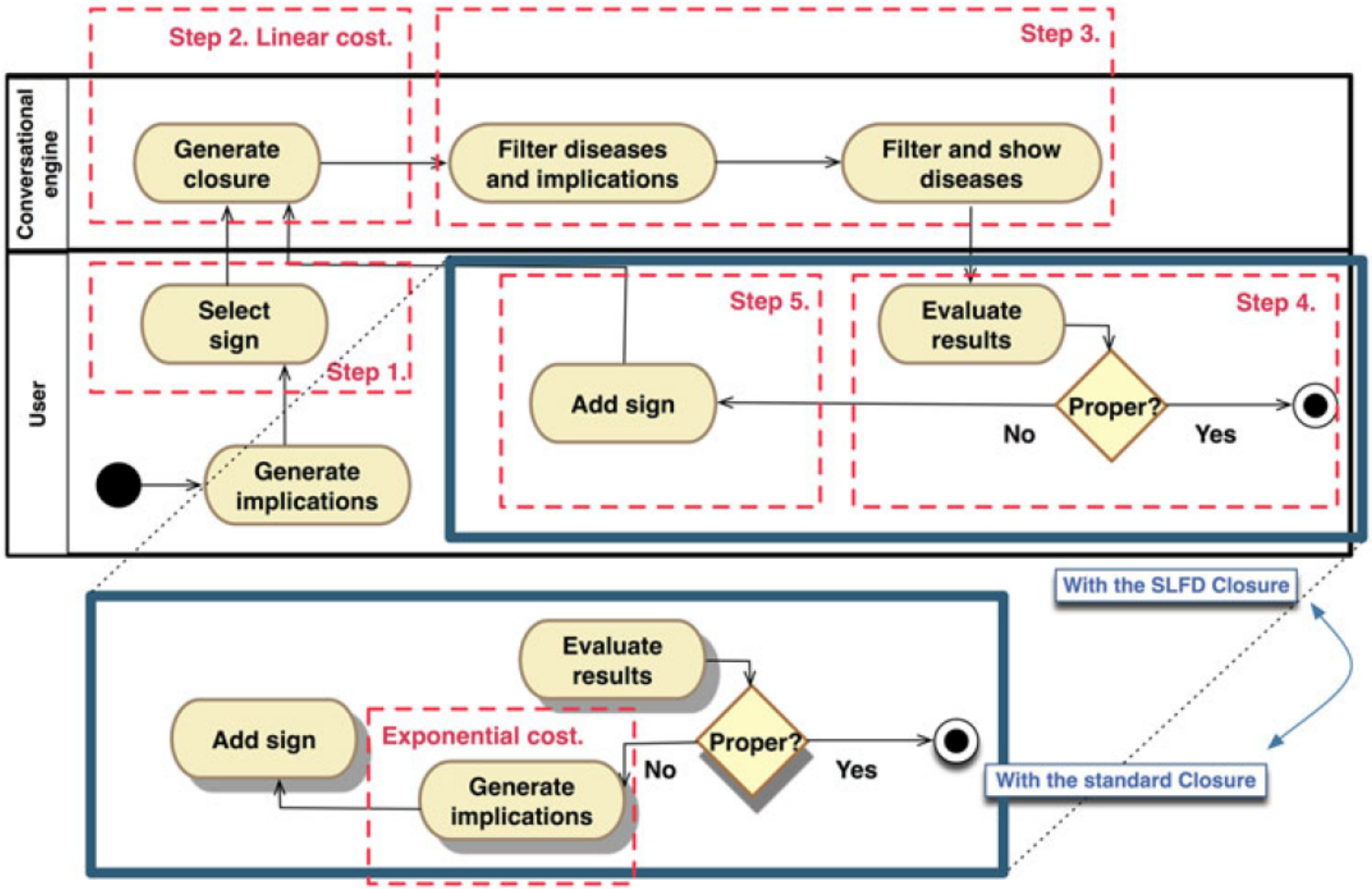
De la información de entrada, debe obtenerse el conjunto de implicaciones, para lo cual existen varios trabajos (y sus respectivas implementaciones) en la literatura que se encargan de ello [28, 57, 58].



SR Conversacional

1. La interacción del usuario con el sistema comienza cuando el usuario elige un atributo con el que comenzar el diálogo.
2. A continuación, el proceso entra en el algoritmo del cierre SL_{FD} para calcular el cierre del conjunto de atributos y al mismo tiempo, el conjunto de implicaciones reducido.
3. Una vez el cierre SL_{FD} termina, se muestra un primer resultado.
4. El usuario decide si terminar el diálogo en caso de estar satisfecho o bien, seguir interactuando mediante la elección de un nuevo atributo.

SR Conversacional



Métricas de Evaluación

Todo sistema debe someterse a una serie de evaluaciones que confirmen su viabilidad y utilidad.
En el caso concreto de los SRs son muchas las métricas que pueden aplicarse, sin embargo, no todas son adecuadas; es preciso utilizar aquéllas cuya naturaleza case con el SR objeto de evaluación [25].

Métricas de Evaluación

Todo sistema debe someterse a una serie de evaluaciones que confirmen su viabilidad y utilidad.
En el caso concreto de los SRs son muchas las métricas que pueden aplicarse, sin embargo, no todas son adecuadas; es preciso utilizar aquéllas cuya naturaleza case con el SR objeto de evaluación [25].

Métrica	Idoneidad
Precision	×
Recall	×
MAE, RMSE	×
Número de pasos	✓
...	

Métricas de Evaluación

Número de pasos (N)

Representa el número de pasos que se dan en el diálogo, y en consecuencia, el número de atributos seleccionados por el usuario [a]

$$N = |\text{Atributos seleccionados}|$$

donde $|A|$ representa el cardinal de A.

[a] Esto se debe a que se ha restringido a uno el número de atributos seleccionables en cada paso.

Métricas de Evaluación

Velocidad de poda en el paso i (S_i)

Representa el porcentaje de atributos que el sistema reduce a lo largo de la conversación y de forma acumulativa de un paso al siguiente.

$$S_i = \frac{|\text{Atributos del cierre}|_i - i}{|\text{M}|}$$

Siendo $i = 1, ..., N$, y M el conjunto global de atributos.

Reducción global de atributos (P)

Representa la reducción global de atributos que ha realizado el sistema al terminar el diálogo.

$$P = S_N$$

Experimentos SRs

Cada experimento realizado consiste en un test que simula 100 diálogos diferentes siguiendo al pie de la letra el esquema presentado anteriormente.
La elección de los atributos se realiza de forma aleatoria:

- Garantiza la honestidad de los resultados.
- Puede oscurecer virtudes del sistema.

Se presentan 3 experimentos principales: **MovieLens, Hoteles Costa del Sol, POIs[5] Mundial**.

[5] En inglés: *Points of Interest*, puntos de interés.

Experimentos SRs I *(MovieLens)*

MovieLens[6] es un proyecto desarrollado por el equipo de investigación GroupLens[7] del Departamento de Ciencias de la Computación e Ingeniería de la Universidad de Minnesota especializado en SRs. Almacena información sobre películas (título, género, duración, año, elenco...) y cuenta con miles de usuarios registrados.

Constituye un referente a la hora de probar el funcionamiento de SRs [26].

[6] http://movielens.org
[7] https://grouplens.org

Experimentos SRs I *(MovieLens)*

MovieLens10M

elementos ↑, atributos ↓, implicaciones ↓

Título	Acción	Comedia	Crimen	Drama	Romance	...
Little City (1998)		✓			✓	
Driver, The (1978)	✓		✓			
Father of the Bride (1950)		✓				
Bio-Dome (1996)		✓				
Fast Runner, The (2001)				✓		
Overboard (1987)		✓			✓	
Get Rich or Die Tryin' (2005)	✓		✓	✓		
...						

- 10.681 películas.
- 19 géneros.
- 245 implicaciones.

Experimentos SRs I *(MovieLens)*

Supongamos que el usuario está buscando una película de acción, con experiencia IMAX[8] y con algunos toques de misterio.

Iter.	Selección	Cierre	Atribs.	Implics.	Items
1	Acción	{Acción, Thriller, Aventura}	16	121	1.473

[8] https://www.imax.com

Experimentos SRs I *(MovieLens)*

Supongamos que el usuario está buscando una película de acción, con experiencia IMAX[8] y con algunos toques de misterio.

Iter.	Selección	Cierre	Atribs.	Implics.	Items
1	Acción	{Acción, Thriller, Aventura}	16	121	1.473
2	IMAX	{Acción, Thriller, Aventura, Sci-Fi, IMAX, Comedia, Fantasía}	12	43	108

[8] https://www.imax.com

Experimentos SRs I *(MovieLens)*

Supongamos que el usuario está buscando una película de acción, con experiencia IMAX[8] y con algunos toques de misterio.

Iter.	Selección	Cierre	Atribs.	Implics.	Items
1	Acción	{Acción, Thriller, Aventura}	16	121	1.473
2	IMAX	{Acción, Thriller, Aventura, Sci-Fi, IMAX, Comedia, Fantasía}	12	43	108
3	Misterio	{Acción, Thriller, Aventura, Sci-Fi, IMAX, Comedia, Fantasía, Misterio Crimen, Romance, Cine-Negro}	8	1	6

[8] https://www.imax.com

Experimentos SRs I *(MovieLens)*

Metas alcanzadas

- Incluso al tratar con un *dataset* muy grande como MovieLens10M, 3 pasos son suficientes para que el usuario obtenga una recomendación.
- La acción del cierre alivia la sobrecarga de información.
- La reducción simultánea de atributos e implicaciones acelera el rendimiento.

Experimentos SRs I *(MovieLens)*

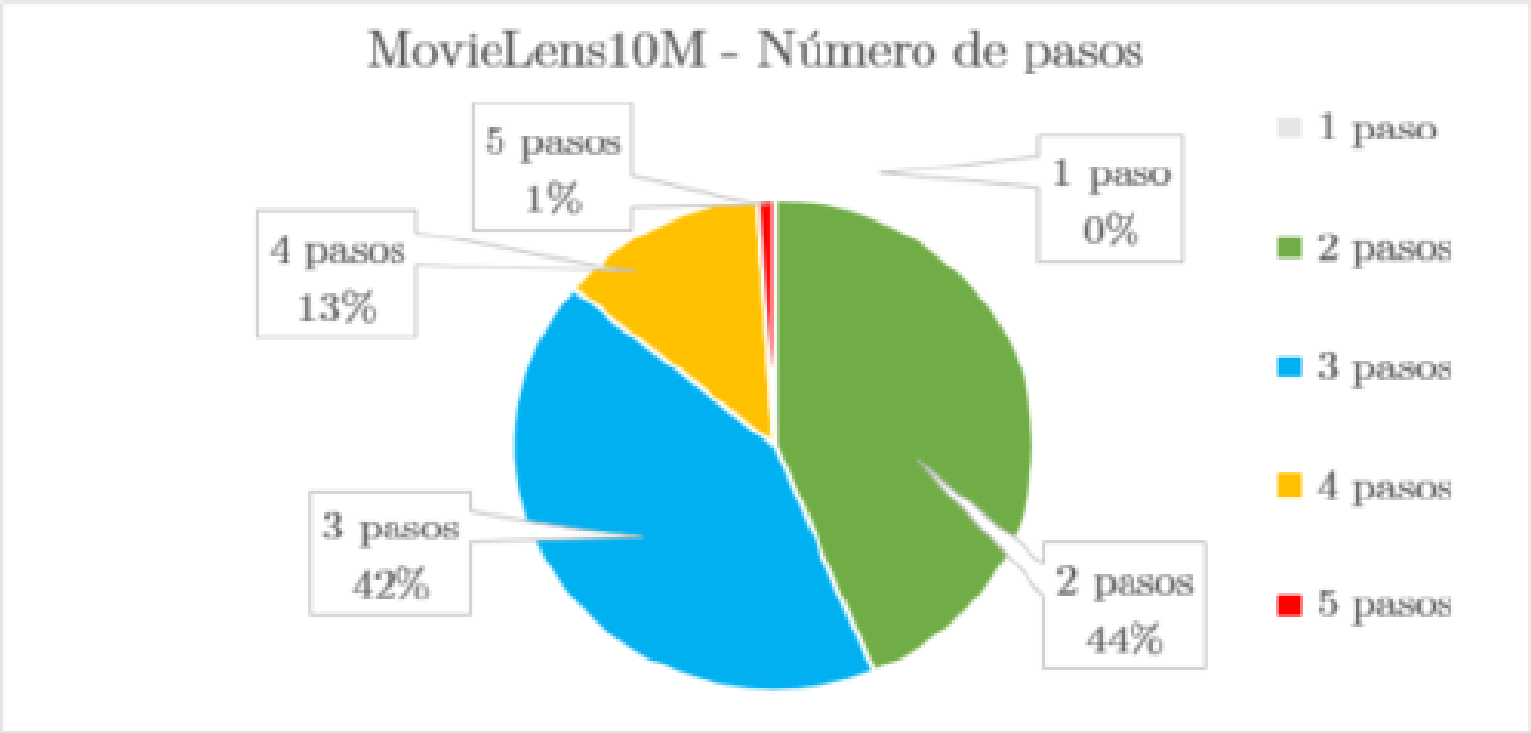


Figura 10: MovieLens10M. Resultado del número de pasos.

Experimentos SRs I *(MovieLens)*

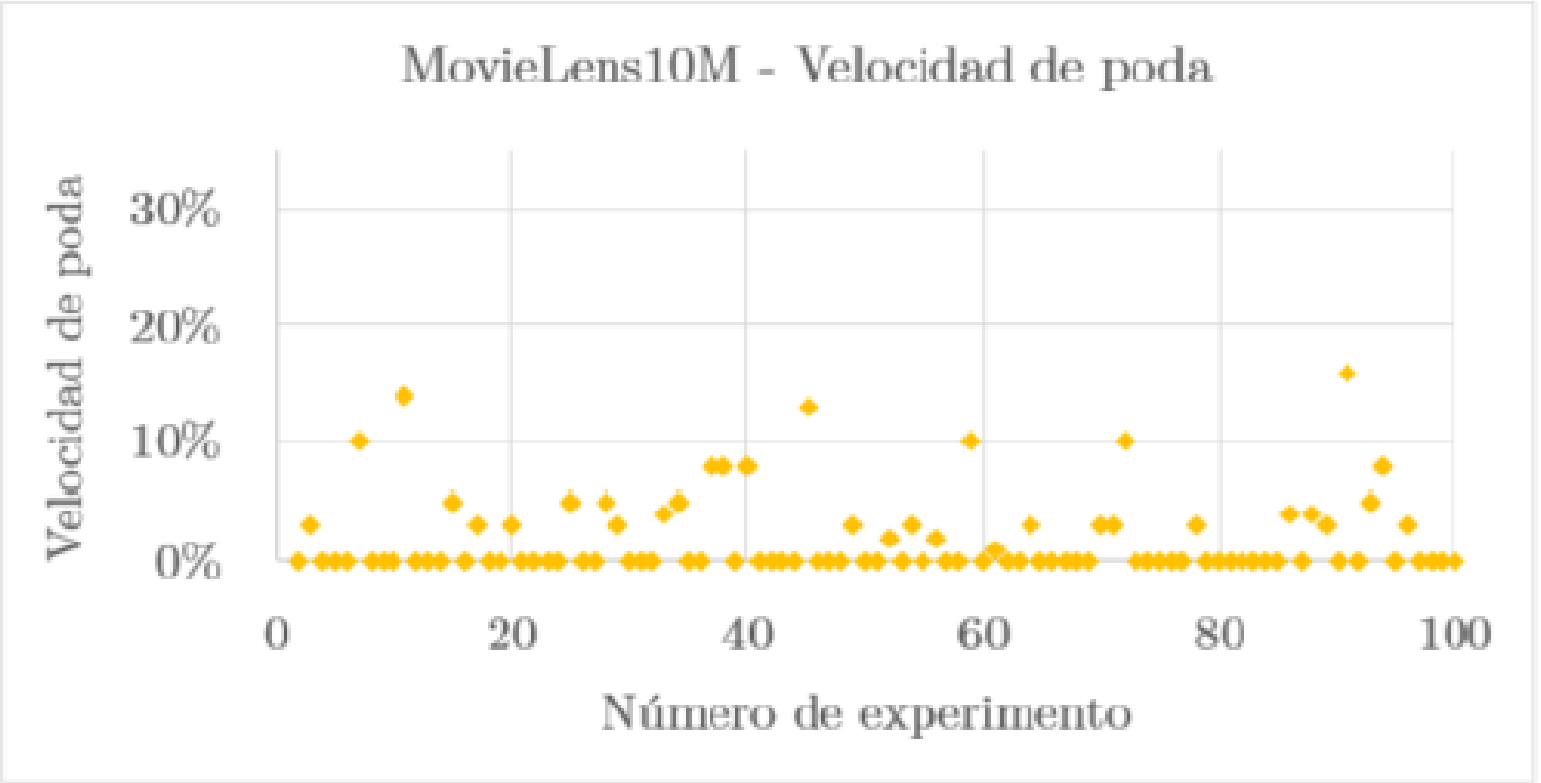


Figura 11: MovieLens10M. Resultado de la velocidad de poda.

Experimentos SRs I *(MovieLens)*

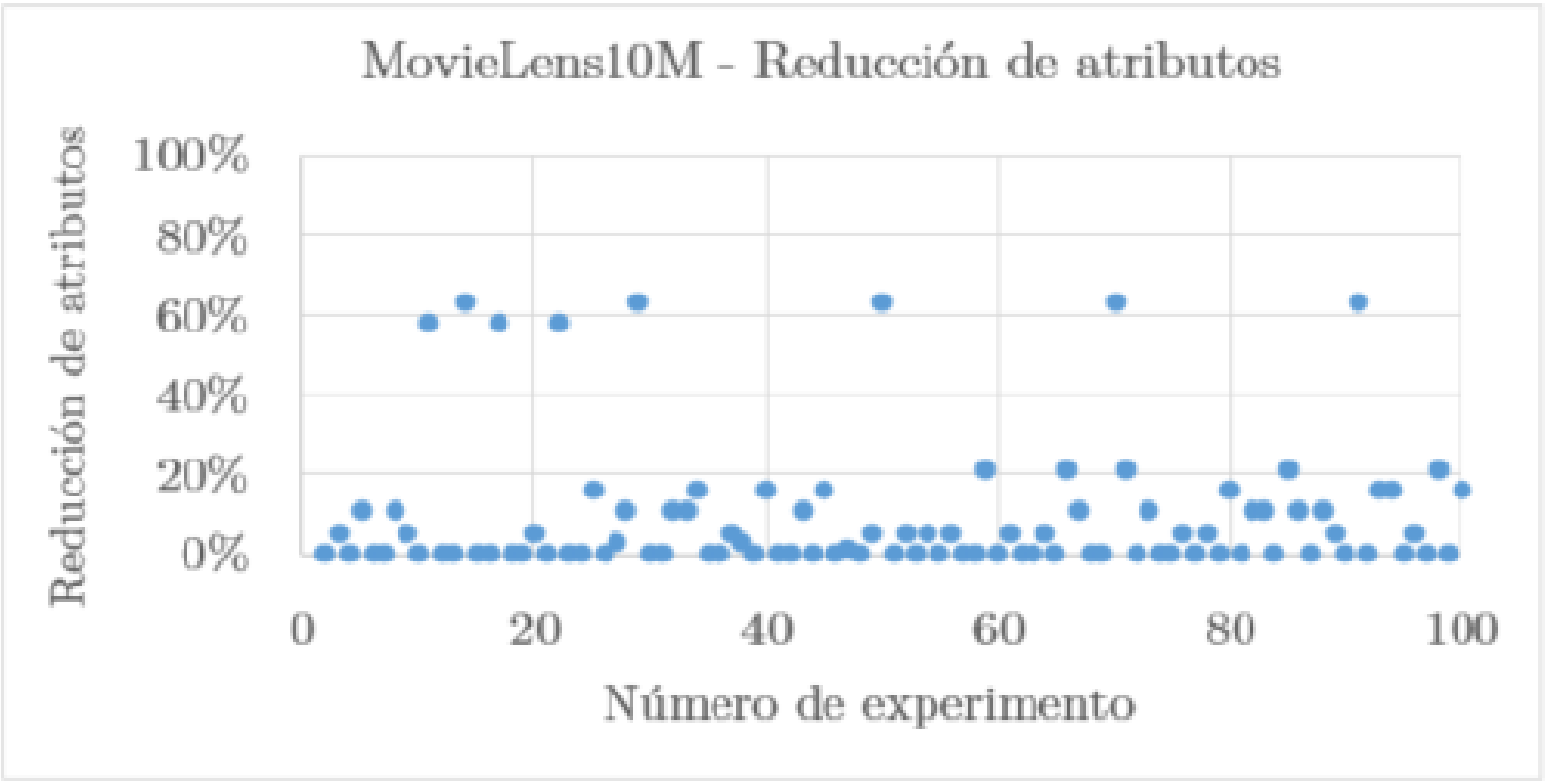


Figura 12: MovieLens10M. Resultado de la reducción de atributos.

Experimentos SRs I *(MovieLens)*

Sumario

- En el 86 % de los casos, 2 ó 3 pasos son suficientes.
- La velocidad de poda a cada paso se sitúa entre el 2 % y el 10 % de los atributos.
- La reducción global está entre el 5 % y el 20 % del total de atributos, alcanzando en ocasiones picos del 60 %.
- Estos resultados superan con creces estudios previos donde, incluso con un *dataset* más pequeño, el número de pasos necesarios es mayor [52].

Experimentos SRs II *(Hoteles Costa del Sol)*

Hoteles Costa del Sol[9]

elementos ↓, atributos ↑, implicaciones ↑

- 361 hoteles.
- 37 atributos (AC, Gimnasio, Wi-Fi, Spa, Aparcamiento, ...).
- 1.507 implicaciones.

[9]Información real obtenida del portal web www.andalucia.org mediante la técnica de *webscrapping*.

Experimentos SRs II *(Hoteles Costa del Sol)*

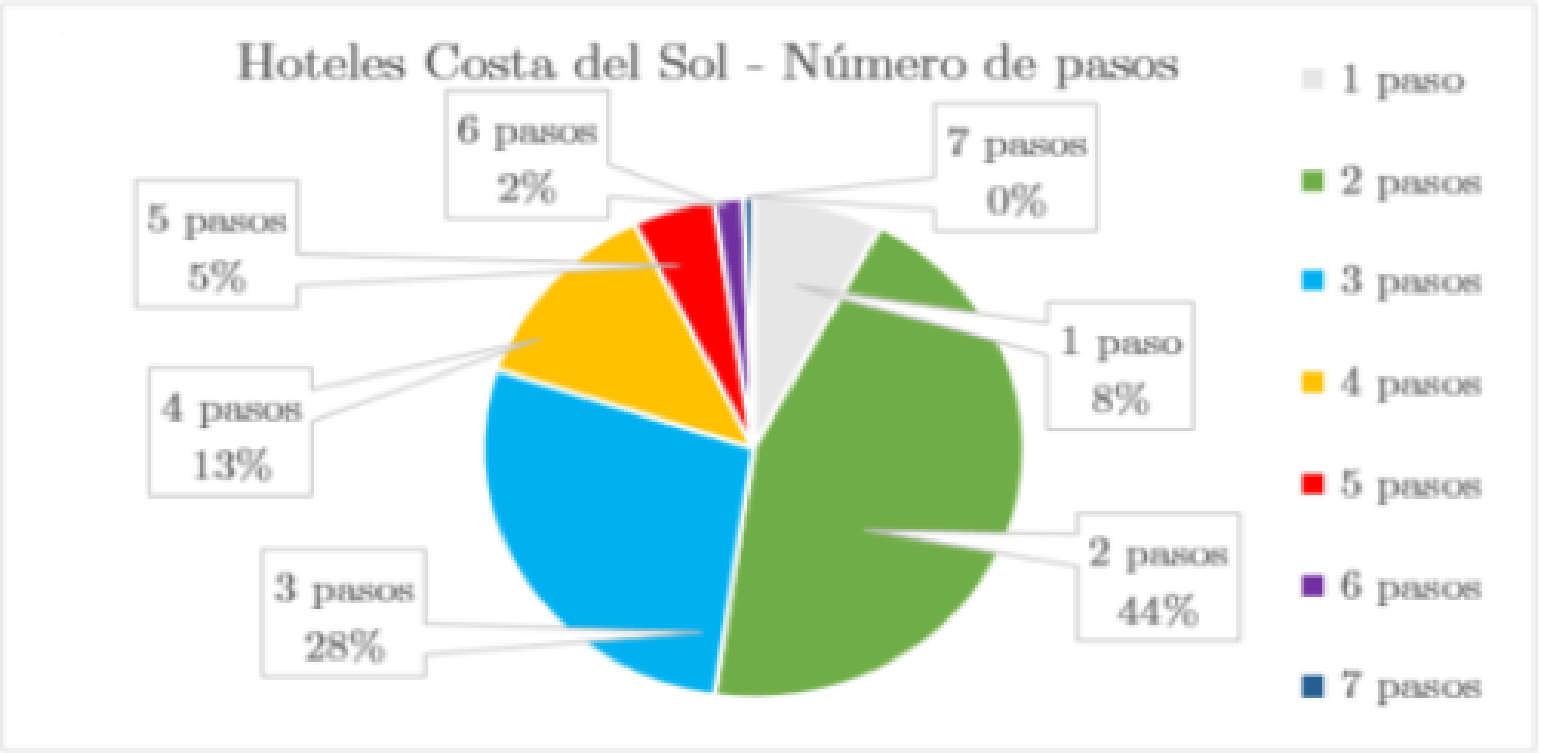


Figura 13: Hoteles Costa del Sol. Resultado del número de pasos.

Experimentos SRs II *(Hoteles Costa del Sol)*

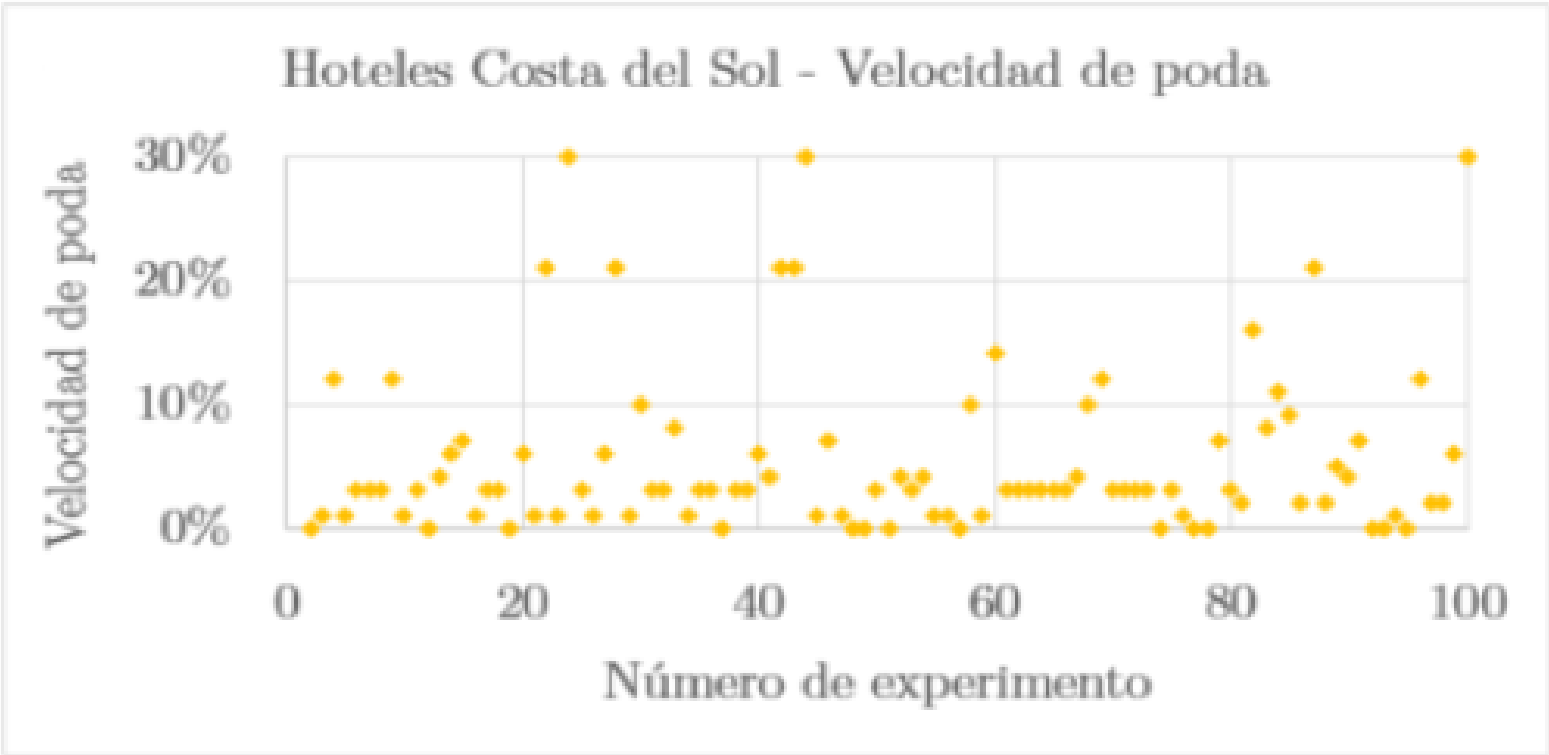


Figura 14: Hoteles Costa del Sol. Resultado de la velocidad de poda.

Experimentos SRs II *(Hoteles Costa del Sol)*

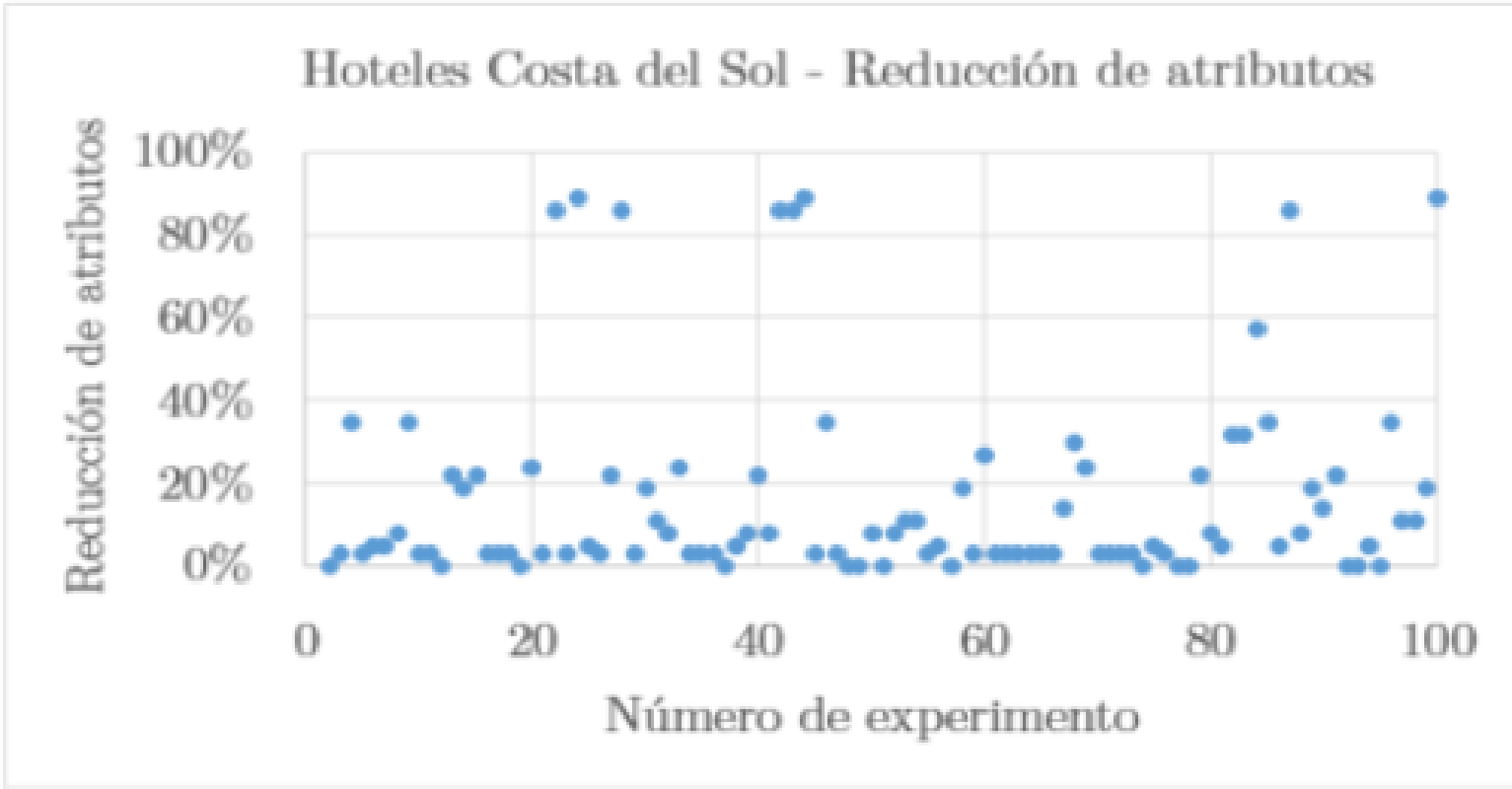


Figura 15: Hoteles Costa del Sol. Resultado de la reducción de atributos.

Experimentos SRs III *(POIs Mundial)*

POIs Mundial[10]

elementos ↑, atributos ↑, implicaciones →

- 17.400 puntos de interés (Torre Eiffel, Estatua de la Libertad, Alhambra, Big Ben, Plaza Roja, Coliseo, ...).
- 115 atributos (Parques, Iglesias, Fuentes, Estadios, Momumentos, Templos, Playas...).
- 675 implicaciones.

[10] Información real obtenida del portal web www.tripadvisor.com mediante la técnica de *webscrapping*.

Experimentos SRs III *(POIs Mundial)*

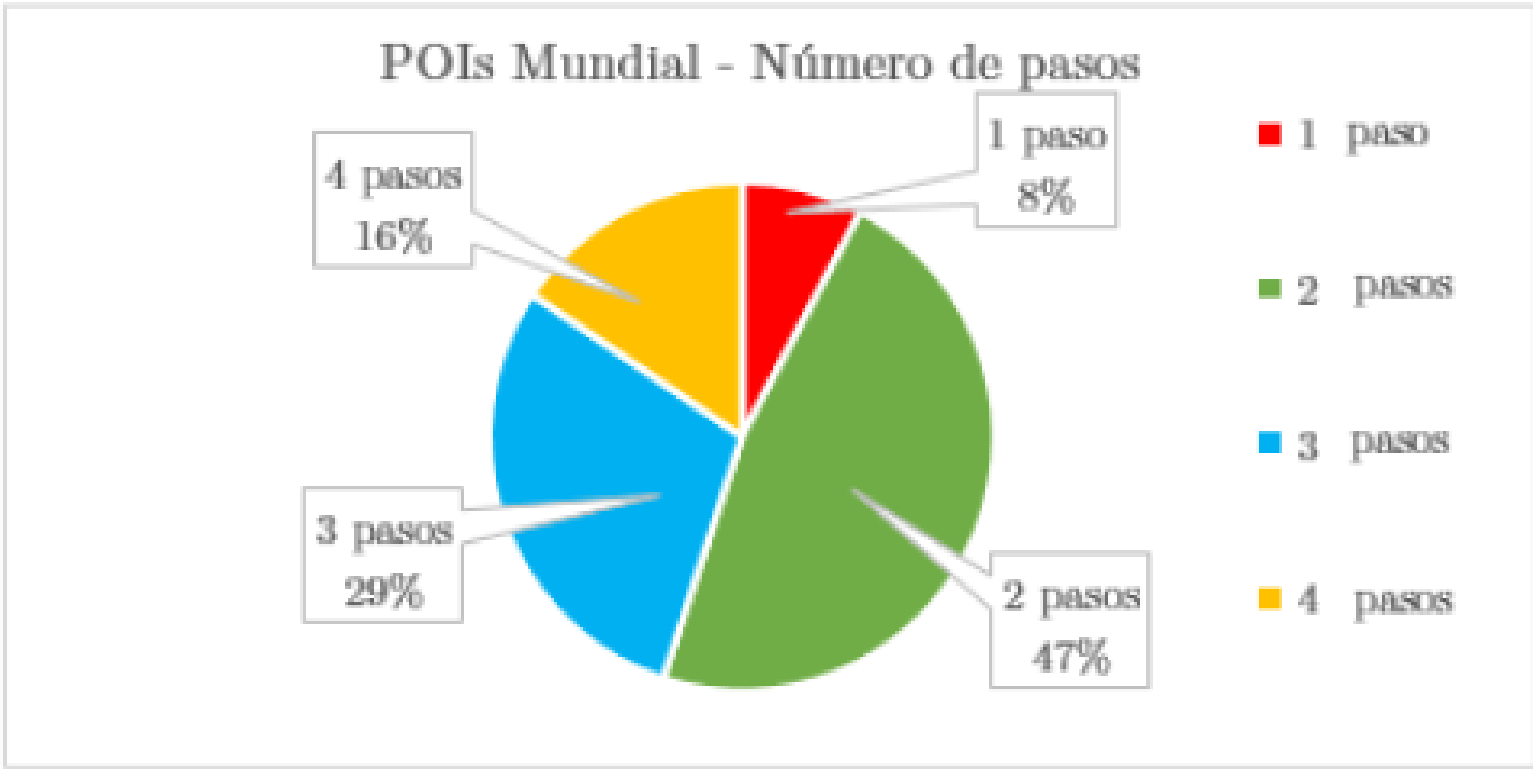


Figura 16: POIs Mundial. Resultado del número de pasos.

Experimentos SRs III *(POIs Mundial)*

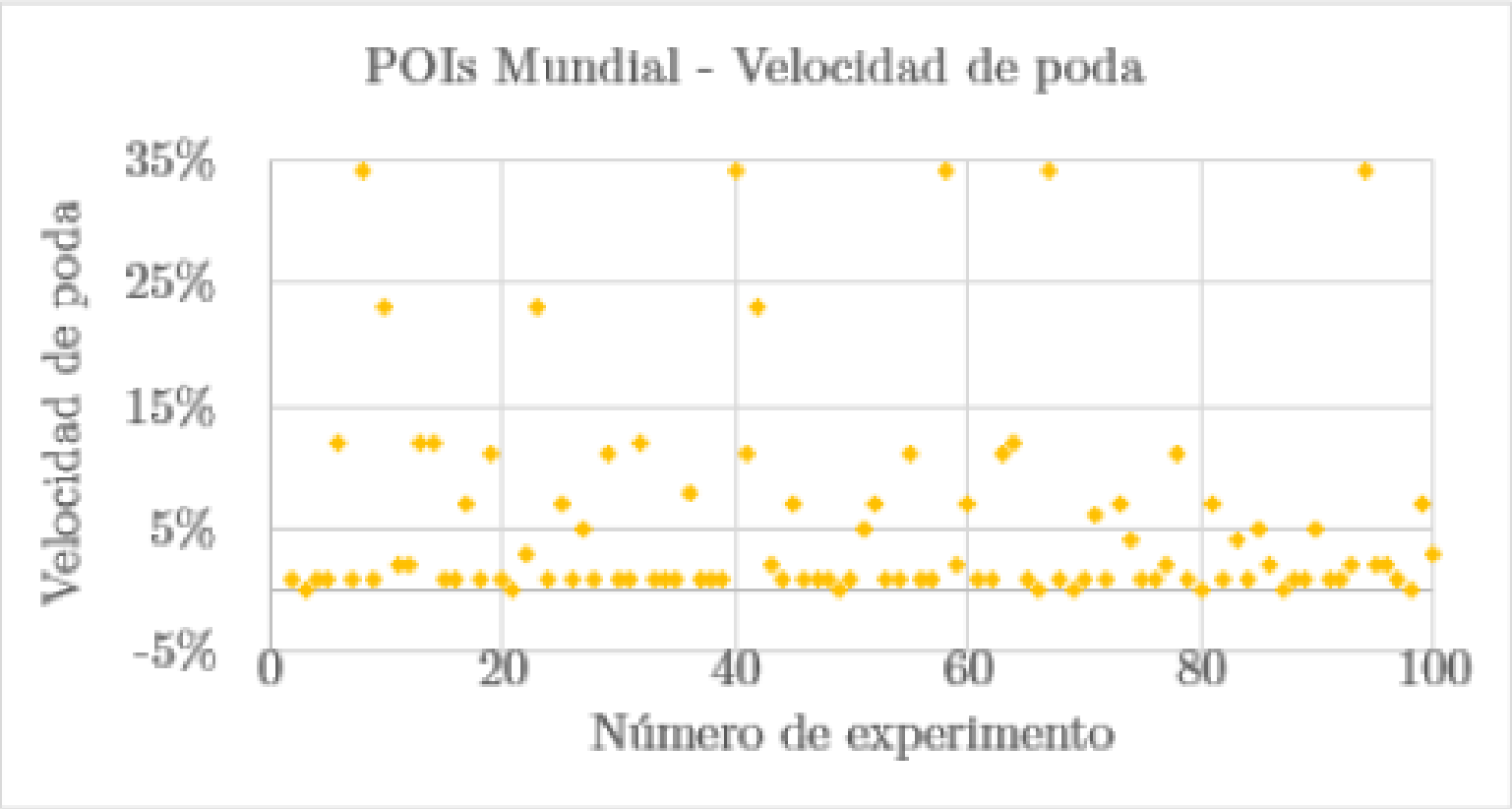


Figura 17: POIs Mundial. Resultado de la velocidad de poda.

Experimentos SRs III *(POIs Mundial)*

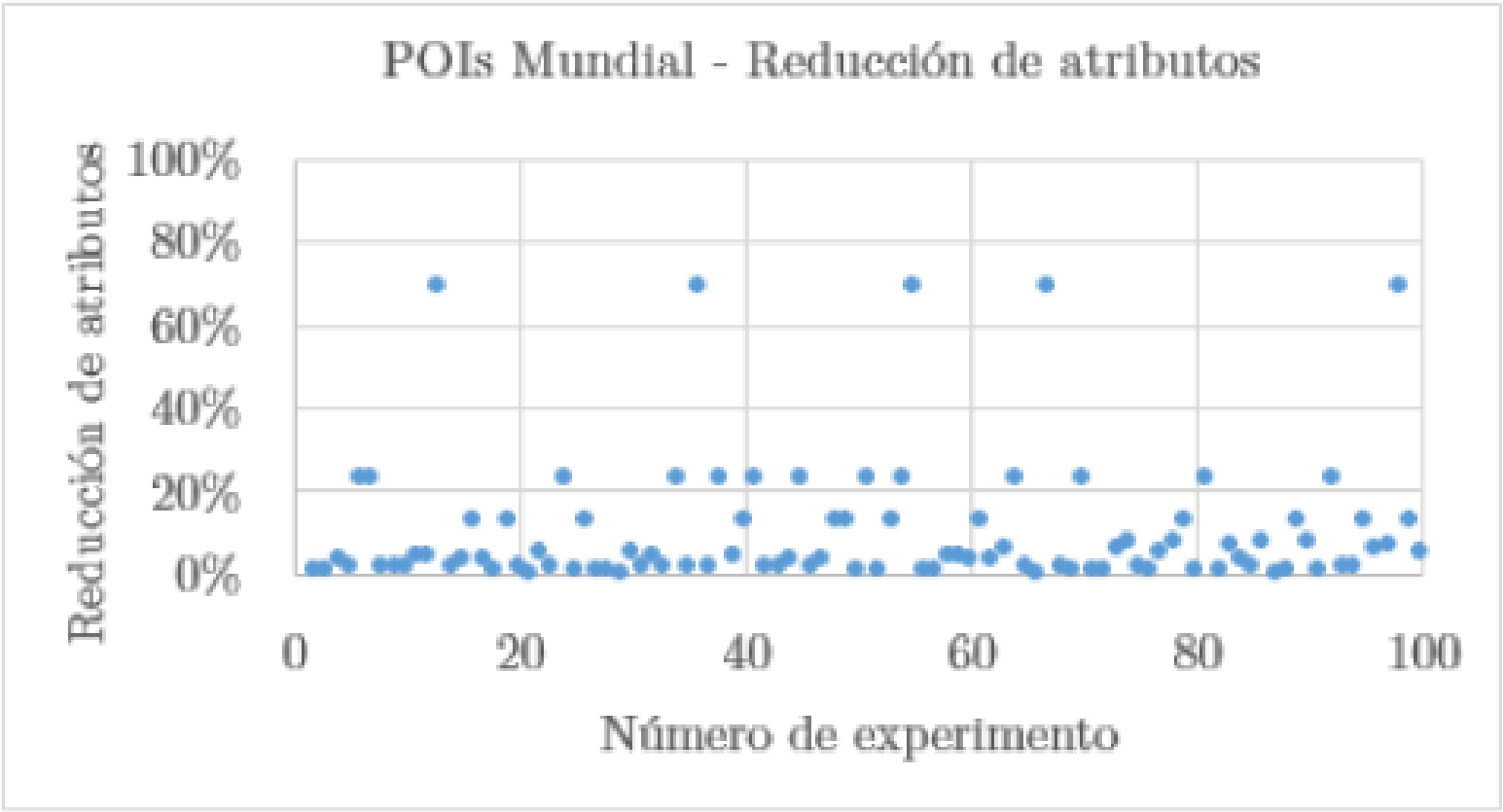


Figura 18: POIs Mundial. Resultado de la reducción de atributos.

Experimentos SRs IV *(Enfermedades & Fenotipos)*

Enfermedades & Fenotipos[11].

elementos $\rightarrow$, atributos $\uparrow$, implicaciones $\uparrow\uparrow\uparrow$

[8]

- 446 anomalías.
- 100 fenotipos.
- 6.468 implicaciones.

[11] Fuente: Online Mendelian Inheritance in Man (www.omim.org) y Phenotype Ontology Consortium (www.human-phenotype-ontology.org)

Índice

Conclusiones Generales

- Una sólida teoría basada en la Lógica y las Matemáticas concede la base para la creación de métodos automatizados para el desarrollo de aplicaciones de ingeniería
- Conjunto de implicaciones, lógica SL_{FD} y cierre SL_{FD}
- Existe una gran cantidad de información implícita en los datos
- De la teoría a la práctica
- Recursos de supercomputación

Conclusiones Claves Minimales

- Nuevo método de búsqueda de clave minimales, CK
- Mejora las aproximaciones anteriores
- Implementación paralela
- Evaluación
- Información real

Conclusiones Generadores Minimales

- Evolución del método MinGen original a MinGenPr y GenMinGen
- Implementación paralela, MinGenPar
- Evaluación
- Información real

Conclusiones Sistemas de Recomendación

- Implicaciones y SRs
- SR híbrido
- Problema de la dimensionalidad
- Supera aportaciones anteriores
- Información real

Trabajos Futuros

- Claves y generadores minimales
  - Caracterizar el problema
  - Optimizar el BOV
  - Número de procesadores
  - Otras formas de paralelismo, e.g. *software*
- Sistemas de recomendación
  - Ponderar la influencia de las características del dataset
  - Explicaciones

Referencias

Armstrong, W. W.

Dependency structures of data base relationships.

In *Proceedings of the International Federation for Information Processing Congress* (1974), pp. 580–583.

Beeri, C., and Vardi, M. Y.

The implication problem for data dependencies.

In *Automata, Languages and Programming* (Berlin, Heidelberg, 1981), S. Even and O. Kariv, Eds., Springer Berlin Heidelberg, pp. 73–85.

Benito-Picazo, F.

Minimal Key-Par. Una versi?n paralela de los algoritmos de b?squeda de claves minimales basados en Tableaux.

Dpto. Lenguajes y Ciencias de la Computación, Universidad de