UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

“Utilización de los enfoques para preferencias en Answer Set Programming para definir familias de problemas con preferencias.”

T E S I S

PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

PRESENTA:

Diego Rodríguez Villanueva

DIRECTORES DE TESIS:

Dra. Claudia Zepeda Cortés,

M. en C. Verónica Rodríguez López,

HUAJUAPAN DE LEÓN, OAX. OCTUBRE DE 2007

Primeramente gracias a Dios, por permitirme hacer lo que tenía que hacer.

A mis padres, Teófilo y Adelfa, por depositar su confianza en mí y brindarme su apoyo incondicional, lo cuál hizo que pudiera continuar adelante a pesar de los obstáculos. Gracias de todo corazón.

A los cimientos de esta familia, Aurea, Ana María†, Rubén y Reynaldo, porque a su manera están pendientes de mis logros y tropiezos.

A mi hermano Miguel Ángel, como prueba de que si es posible alcanzar las metas establecidas.

A la Dra. Claudia Zepeda Cortés, por su apoyo e insistencia para cumplir juntos este objetivo, por su tiempo y comprensión, infinitas gracias. A la M. C. Verónica Rodríguez López por darme su apoyo cuando la situación tomo otro rumbo al establecido inicialmente.

A los profesores M. C. María Auxilio Medina Nieto, M. C. Wendy Yaneth García Martínez y M. C. Ricardo Ruiz Rodríguez, por su tiempo y colaboración para concluir este trabajo de tesis.

A todos mis compañeros, quienes ayudaron a que la carga de trabajo fuera más ligera. Gracias.

A todos aquellos que conocí durante el andar de este camino y que aportarón su granito de arena. Gracias.

A mis amigas, amigos y demás familia que siempre estuvieron pendientes. Gracias por su interés.

Índice general

Índice de Tablas

Índice de figuras

Capítulo 1

Introducción

La noción de preferencia es común en los contextos en los que están involucradas la decisión y selección, ya que el concepto de preferencia dice que es la acción de determinar una opción cuando se encuentra bajo una situación con varias opciones, donde la opción elegida satisface en forma relativa o absoluta los gustos, deseos, objetivos y/o prioridades de quien realiza la elección, en el momento en que esa opción es determinada [17]. Actualmente el concepto de preferencia se ha desarrollado en diferentes áreas de investigación, mostradas en [7, 15], por ejemplo, en la teoría de decisión matemática, las preferencias (frecuentemente expresadas como utilidades), son usadas para el modelo económico de comportamiento humano. En Inteligencia Artificial, los agentes se apoyan de las preferencias para obtener sus objetivos. En bases de datos, las preferencias ayudan en la reducción de la cantidad de información regresada en las respuestas de las consultas del usuario. En filosofía, las preferencias son usadas para el razonamiento sobre valores, deseos y deberes. En computación, el interés está en la reducción al mínimo de los recursos de cómputo (tiempo, espacio, comunicación, etc.) para realizar una tarea bajo ciertas características.

Además, el campo de aplicación de los problemas reales con preferencias es amplio, [8, 16] citan algunos ejemplos como los siguientes:

1. La configuración de productos, buscando la configuración óptima y factible que satisfaga en un alto porcentaje a los clientes.

2. Búsqueda de información, regresando la cantidad mínima de resultados que satisfagan el criterio de selección de la consulta, teniendo un control respecto a la búsqueda declarativa, describiendo las estrategias de acuerdo a las reglas del negocio.

3. Personalización, dando al usuario la satisfacción de obtener lo más apropiado de acuerdo a sus criterios en la selección de múltiples objetivos.

4. Realizar la propagación de restricciones de forma más rápida, de acuerdo a la activación de los valores preferidos.

5. Ayudan en la elección de alguno de los modelos de solución posibles que existen en el dominio de un problema.

6. Las preferencias también son usadas en el dominio de la planificación, como se muestra en [3, 9, 26], donde las preferencias desempeñan un papel decisivo en la elección de un plan. Es necesario poder evaluar los componentes de cada plan además de que las preferencias de los usuarios son importantes para seleccionar el plan para la ejecución, cuando un problema tiene más de un plan.

Por lo que el interés en el desarrollo de herramientas que trabajen con preferencias para su aplicación a problemas reales es amplio. La herramientas que se tienen actualmente se han desarrollado de forma individual, haciendo referencias y/o retomando ejemplos de otras, sin tener una comparación entre ellas.

Nuestro objetivo central es el tomar algunas herramientas que modelan preferencias y así analizarlas, utilizarlas y compararlas. Las herramientas que se consideraran son denominadas enfoques y están definidas para *answer set programming*,(ASP). La Figura 1.1 muestra el modelo de los pasos que se tienen para solucionar un problema real con preferencias por medio de un enfoque en ASP.

Figura 1.1: Solución de problemas del mundo real con preferencias en ASP.

Del modelo de la Figura 1.1, lo primero es tener la descripción del problema, de tal forma que se tomen en cuenta todas las características que se desean evaluar para encontrar la solución. Después se debe de representar el problema por medio de un programa lógico en ASP, para que se puedan encontrar las soluciones, consideradas los *answer sets* del problema. Esto se puede hacer por medio de los diferentes sistemas que hay para este propósito, como *DLV* o *SMODELS*. Cuando se tienen los *answer sets* del problema, se debe aplicar alguno de los diferentes enfoques de ASP para la representación y solución de problemas con preferencias, los cuales trabajan de acuerdo a criterios de preferencia establecidos para esos enfoques, dando como resultado la solución preferida al problema, la cual es un subconjunto del conjunto de *answer sets* obtenidos anteriormente.

De acuerdo a los pasos del modelo de la figura 1.1 se analizarán algunos enfoques en ASP para modelar preferencias. Aplicados los enfoques en ASP, se comparan los resultados obtenidos para cada enfoque, observando que sea acorde con lo esperado por el sentido común, analizando el tipo de respuesta para el problema y así poder determinar el comportamiento del enfoque.

En ASP algunos de los enfoques para preferencias que existen actualmente son:

1. Programas Lógicos con Disyunción Ordenada de Gerhard Brewka [9].
2. Lenguaje $\mathcal{PP}$ de Enrico Pontelli et al. [26].
3. Reglas para restauración de consistencia de Marcello Balduccini et al. [3].
4. Programas Lógicos con preferencias de Mauricio Osorio et al. [24].
5. Programas en *Answer sets* con optimización de Gerhard Brewka et al. [11].

Como antes se mencionó, cada uno de estos enfoques cuenta con criterios de preferencia particulares para elegir la solución preferida del problema.

La forma en que se desea tener la comparación y el comportamiento de los enfoques es a partir de: 1) presentar de forma uniforme la teoría de los enfoques y, 2) de las características de los problemas con preferencias. Considerando las características de los problemas con preferencias'se propondrá una clasificación.

El documento de tesis está organizado de la siguiente forma. En el capítulo 2 se presenta la información que es la base de este trabajo. En el capítulo 3 se analizan algunos de los enfoques en ASP que trabajan con preferencias, presentándolos de forma uniforme. Los capítulos 2 y 3 forman el marco teórico en este documento. Con el marco teórico analizado, en el capítulo 4 se propone una clasificación para los problemas con preferencias. La clasificación se propone a partir del análisis de los problemas de este tipo en las relaciones y restricciones que puedan tener las preferencias. Con las clases de problemas con preferencias se trata de observar el comportamiento de los enfoques revisados. En el capítulo 5 están las conclusiones obtenidas en el desarrollo de este trabajo. Tambien se presentan las propuestas de trabajos a futuro. Por lo que en los captulos 4 y 5 se tiene la contribución de este trabajo.

El apéndice A presenta una extensión que existe para uno de los enfoques que es revisado en el capítulo 3. Esto con el fin de tener una ampliación en ese enfoque. En el apéndice B se presenta una colección de problemas con preferencias resueltos de acuerdo a los enfoques analizados en el capítulo 3 y a la clasificación del capítulo 4.

Capítulo 2

Fundamentos Teóricos

Este capítulo presenta la información que es la base teórica de este trabajo. La información del capítulo se muestra en dos partes: razonamiento del sentido común y *Answer Set Programming*. En la parte del razonamiento del sentido común, se presenta cómo es considerado y las razones que se tienen para estudiarlo. En la sección de *Answer Set Programming*, se presenta este lenguaje de programación, ubicándolo dentro de los paradigmas de programación y mostrando las definiciones generales con las que trabaja.

2.1. Razonamiento del Sentido Común

El conocimiento del sentido común es aquel conocimiento que tenemos sobre el mundo y de cómo trabaja. El razonamiento del sentido común es el tipo de razonamiento que cada uno desempeña todos los días acerca del mundo sobre el mundo mismo. Este razonamiento es el proceso de tomar información sobre ciertos aspectos de un escenario en el mundo y realizar inferencias sobre otros aspectos de ese escenario basados en el conocimiento del sentido común [20]. Por ejemplo, si sabemos que el cielo está nublado podemos inferir que puede llover, o que si nos gustan las películas de acción y en el cine se exhibe una película de acción, podemos decir que esta película nos va a gustar si la vamos a ver.

El razonamiento del sentido común es esencial para el comportamiento y pensamiento inteligente. Nos permite completar espacios en blanco que se tengan dentro de un escenario, reconstruir partes perdidas del escenario, determinar lo que ha sucedido, y poder predecir qué podría pasar más adelante [20].

Las razones para estudiar el razonamiento del sentido común son tanto la práctica como la científica. La razón práctica consiste en que el automatizar el razonamiento del sentido común tiene muchas aplicaciones, como en interfaces inteligentes para usuarios, donde se utiliza para tener una mejor comunicación entre el sistema y el usuario, también en el procesamiento del lenguaje natural (en la robótica) para tratar de tener una mejor comunicación y que no sea tan predecible como lo es en un proceso secuencial, y más aplicaciones similares. La segunda razón es científica, ya que el razonamiento del sentido común es una capacidad básica de la inteligencia, que apoya otras capacidades de un nivel más alto, es decir, tener la habilidad de entender qué es lo que está sucediendo a partir de hechos pequeños, por más insignificantes que parezcan, y con esto poder determinar una situación más grande. Por ejemplo, de acuerdo a los estudios y publicidad que se le ha hecho a la comida chatarra, se puede determinar el estado de salud que una persona puede tener si consume en exceso este tipo de alimentos por un período largo de tiempo.

Actualmente, algunas de las inquietudes que tienen los investigadores en el área de Inteligencia Artificial son el poder representar el conocimiento y poder realizar el razonamiento del sentido común, aunque estos aspectos del comportamiento humano parecieran tareas muy sencillas y simples, resulta que son elementos difíciles de representar para que puedan ser procesados por una computadora. Por ejemplo, si deseamos diseñar una entidad, (una máquina o un programa) capaz de comportarse de forma inteligente dentro de algún ambiente para realizar cierta tarea, entonces es necesario proveer a esta entidad de conocimiento suficiente sobre este ambiente. Para hacer eso, necesitamos un lenguaje capaz de expresar este conocimiento, junto con una manera exacta y bien entendida de manipular el conjunto de sentencias de ese lenguaje que permitirá que la entidad pueda realizar las inferencias, consultas, y que podamos actualizar la base de conocimiento y el comportamiento deseado de la entidad, de tal forma que, tenga un comportamiento muy próximo a lo que esperamos como inteligente dentro de ese escenario [6].

2.2. *Answer Set Programming*

En esta parte del documento se presenta una introducción al lenguaje *Answer Set Programming* (ASP). Al ser un lenguaje de programación, se ubica con respecto a los paradigmas de programación. Una vez que se tiene ubicado a ASP y el paradigma con el que trabaja, se tienen definiciones generales sobre ASP.

2.2.1. ASP dentro de los Lenguajes de Programación

Dentro de los lenguajes de programación, se pueden identificar varios paradigmas de programación. Lo considerado aquí para los paradigmas de programación es retomado de [2].

Los paradigmas de programación son definidos como una colección de patrones conceptuales que combinados modelan el proceso de diseño para determinar la estructura del programa. Los paradigmas de programación se distribuyen en tres clases, los cuales se muestran en la Figura 2.1. Las tres clases identificadas son: Operacional, Demostrativa y Declarativa. Dentro de las clases que identifican la forma en que se puede solucionar un problema, se tiene otro criterio para clasificar los paradigmas de programación. El criterio es el tipo de paradigma. Los tipos dentro de las clases identifican a los paradigmas de acuerdo a una característica de la clase a parte de la forma en que se solucionan los problemas.

Dentro de los paradigmas (identificados con el cuadro redondeado), se tiene a los diferentes lenguajes de programación, como el ASP identificado por el hexágono. La clasificación mostrada en la Figura 2.1 indica de forma general la jerarquía que se tiene para la clasificación de los paradigmas de programación, aunque actualmente un lenguaje de programación puede combinar diferentes paradigmas.

En la Figura 2.1, se observa que la clase de programación declarativa contiene los lenguajes que construyen programas basados en hechos, reglas, restricciones, ecuaciones, transformaciones y otras propiedades derivadas del conjunto de parámetros o valores que configuran la solución. Los paradigmas que pertenecen a esta clase son: funcional, restringido, lógico, basado en la forma, flujo de datos y de transformación.

Durante un largo período, el paradigma de la programación lógica fue concebido principalmente como un paradigma lógico dentro de la clase para la programación declarativa, y más aún, como una herramienta para la representación y el razonamiento del conocimiento. Recientemente ha llegado a ser evidente que la programación lógica se considera como marco de gran alcance para la representación del conocimiento del sentido común y del razonamiento de este conocimiento.

2.2.2. Definiciones Generales de ASP

En la Figura 2.1 se observa que ASP es un lenguaje de programación lógico. ASP es un lenguaje de clase declarativa útil para la representación del conocimiento. Fue definido en 1987 por M. Gelfond y V. Lifschitz [12]. Actualmente cuenta con fundamentos teóricos sólidos basados en ideas de diversas áreas de la Inteligencia Artificial y la Lógica Matemática. Una de las razones del éxito de ASP es la disponibilidad de

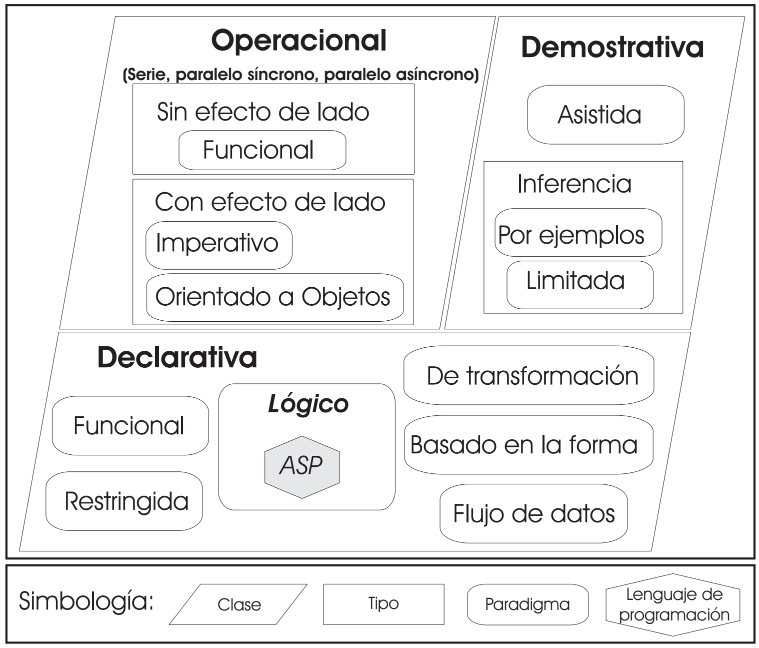

Figura 2.1: Paradigmas de programación.

sistemas eficientes para el cálculo de *answer sets*, como lo son *SMODELS* [1] y *DLV* [2]. El uso de ASP permite describir un problema computacional como un programa lógico cuyos *answer sets* corresponden a las soluciones del problema dado.

Además, ASP es muy conveniente para representar restricciones. Para ilustrar esto, consideremos el problema del Mundo del Tráfico descrito en [1], que en una parte de la descripción del problema nos dice: "Si la velocidad máxima del coche de enfrente es más pequeña que la velocidad máxima del coche que lo sigue, entonces en cierto momento el coche más rápido disminuirá su velocidad a la velocidad máxima del coche de enfrente ya que el camino no permite rebasar", donde la restricción está en la capacidad del camino. Otra ventaja de ASP es que representa excepciones como las descritas en el ejemplo del Mundo del Zoológico en [1], una de ellas dice: "Cada animal está en una posición en un determinado tiempo. Dos animales grandes no pueden ocupar la misma posición,

[1] http://www.tcs.hut.fi/Software/smodels/
[2] http://www.dbai.tuwien.ac.at/proj/dlv/

excepto si uno de ellos está montado sobre el otro", donde la excepción es clara cuando se refiere a la posición de animales grandes. ASP también permite expresar problemas de una manera general, es decir, permite representar familias de problemas.

Se han propuesto diferentes enfoques en ASP para la representación de diferentes problemas que tienen que ver con actualización, argumentación, planificación, preferencias, etc.

En este trabajo, *un programa es interpretado como una teoría proposicional* y la única negación usada es la *negación por default*. Para los lectores no familiarizados con este enfoque, recomendamos [25, 21] como lectura posterior. Nuestra discusión se limita a programas proposicionales utilizando *el lenguaje de la lógica proposicional* de la manera usual, donde se tienen:

- símbolos proposicionales: p, q, $\ldots$,
- conectivos proposicionales: $\wedge$, $\vee$, $\rightarrow$, $\bot$ y
- símbolos auxiliares (,).

Definición 2.1. Una *fórmula proposicional bien formada* está dada por:

- Un símbolo proposicional,
- Si f y g son fórmulas proposicionales bien formadas, entonces también lo son $\neg f$, $f \wedge g$, $f \vee g$, $f \rightarrow g$, $f \rightarrow \bot$.

$\square$

Hacemos énfasis en que la única negación usada en este trabajo es la *negación por default* y es representada por el símbolo $\neg$. Vale la pena mencionar que siempre es posible manejar la otra negación llamada *clásica* o también *negación fuerte*, denotada por $-$, por medio de transformar los átomos con negación clásica [13]. Cada átomo con negación clásica, $-a$, que ocurre en la fórmula es sustituido por un nuevo átomo, a°, y la regla $\neg(a \wedge a^{\circ})$ es agregada. La regla $\neg(a \wedge a^{\circ})$ puede también ser escrita como $(a \wedge a^{\circ}) \rightarrow \perp$ basada en el hecho de que $\neg f$ es solo una abreviación de $f \rightarrow \perp$.

La *firma de un programa* P, denotada como $\mathcal{L}_P$, es el conjunto de átomos que ocurren en P. Una *literal* es cualquier átomo p (una literal positiva) o la negación de un átomo $\neg p$ (una literal negativa). Una *literal negada* es el signo de negación $\neg$ seguido por una literal, i.e. $\neg p$ or $\neg\neg p$. En particular, $f \rightarrow \perp$ es llamada *restricción* y también es denotada como: $\leftarrow f$. Una *teoría regular* o *programa lógico* es un conjunto de fórmulas bien formadas.Un *programa lógico* también puede ser llamado solamente *teoría* o *programa* cuando no surjan ambigüedades.

En algunas definiciones utilizaremos la *lógica Intuicionista* de Heyting, la cual será denotada por *logica* I.

Definición 2.2. [19] Dado un conjunto de átomos M y un programa P, escribiremos $P \vdash_{\mathrm{I}} M$, para abreviar que los átomos de P se satisfacen en M por medio de la *logica* I, es decir: $a \vdash_{\mathrm{I}} M$ para toda $a \in M$. □

Para un conjunto dado de átomos M y un programa P, escribiremos $P \Vdash_{\mathrm{I}} M$ para denotar que:

- $P \vdash_{\mathrm{I}} M$ y
- P es consistente con respecto a la *logicaI* (es decir, que no hay una fórmula A tal que $P \vdash_{\mathrm{I}} A$ y $P \vdash_{\mathrm{I}} \neg A$).

Algunas veces se puede tener $\vdash$ en lugar de $\vdash_{\mathrm{I}}$, sólo cuando no surjan ambigüedades.

Ahora definimos los *answer sets* (o modelos estables) para los programas lógicos. La semántica de *answer sets* fue definida primero en términos de la *reducción Gelfond-Lifschitz* [12] donde se estudia en el contexto de sintaxis dependientes de transformaciones. Aquí se sigue un enfoque alternativo estudiado por Pearce [25] y por Osorio et.al.

[21]. Este enfoque caracteriza los *answer sets* de una teoría proposicional en términos de la lógica intuicionista, presentada en el teorema 2.1. La notación está se basa en [21].

Teorema 2.1. *Sea P una teoría y M un conjunto de átomos. M es un answer set para P si y solo si $P \cup \neg(\mathcal{L}_P \setminus M) \cup \neg\neg M \Vdash_{\mathrm{I}} M$.* □

Como parte de la notación que será utilizada en este trabajo, retomamos la usada en [19], donde se tiene lo siguiente:

- La fórmula $f \leftarrow g$ es otra forma de escribir $g \rightarrow f$.
- La fórmula $(g \leftarrow f) \wedge (f \leftarrow g)$ se puede abreviar como $g \leftrightarrow f$.
- Dado un conjunto de fórmulas F, se define a $\neg F = \{\neg f | f \in F\}$.
- Siguiendo con la noción tradicional de la programación lógica, se puede utilizar *not* indistinto de $\neg$, y también usar a, b indistinto de $a \wedge b$.
- Para un conjunto de fórmulas F, se escribe $literales(F)$ para denotar el conjunto de literales de F.
- Para el conjunto de literales L, $pos(L)$ denota el conjunto de literales positivas en L y $neg(L)$ denota el conjunto de literales negativas en L.
- Dado un programa P. El conjunto de *teoremas de* P, denotado como $th(P)$, es definido como: $th(P) = \{\alpha | \alpha \text{ es una fórmula sobre } \mathcal{L}_P \text{ y } P \vdash \alpha\}$.
- Un conjunto finito de fórmulas P es *consistente* si no hay alguna fórmula A tal que A y $\neg A$ sean teoremas de P .
- Un conjunto finito de fórmulas P es *completo* si, para cualquier fórmula A de P, $P \vdash A$ o $P \vdash \neg A$.

- Un conjunto finito de fórmulas P' es una *extensión* de un conjunto finito de fórmulas P si cada teorema de P es un teorema de P'.

- Dados dos conjuntos X y Y, escribimos $X \subset Y$ para denotar que $X \subseteq Y$ y $X \neq Y$.

Es usual que ASP considere un programa con símbolos del predicado es solamente una abreviatura del programa instanciado, i. e. programa sin variables, denotado como $instanciado(P)$.

En algunos casos, necesitamos modelar algunos problemas usando símbolos del predicado con variables, donde estas variables deben ser instanciadas solamente con un subconjunto del universo de Herbrand. Esta clase de programas se llaman *programas de clases con símbolos de predicado* [5].

2.3. Conclusión

En este capítulo se mostró la información que es la base de lo que se consideró para realizar este documento. Primero se presentó la sección del razonamiento del sentido común. En esta sección se mostro lo que es considerado el razonamiento del sentido común así como las razones que se tienen para estudiarlo. La segunda sección presenta una introducción a ASP, ubicándolo dentro de los paradigmas de programación. También se muestran las definiciones generales con las que trabaja ASP.

Capítulo 3

Enfoques de ASP para Tratar Preferencias

En este capítulo se mostrarán tres de los enfoques que existen para modelar problemas con preferencias en ASP. Los enfoques considerados son: LPOD, PLP y ASOP. Se describirá el enfoque así como su funcionamiento para modelar y resolver problemas con preferencias dentro de una notación uniforme. Al final del capítulo se presentan las diferencias y/o similitudes que puedan tener los enfoques analizados.

3.1. Programas Lógicos con Disyunción Ordenada

En [9], Gerhard Brewka introduce a los Programas Lógicos con Disyunción Ordenada, abreviados como LPODs por su definición en inglés, *Logic Programs with Ordered Disjunction*. La programación lógica con disyunción ordenada es una extensión de la programación lógica bajo la semántica de *answer sets* [12], agregando un operador denominado *disyunción ordenada*, representado por $\times$, y los dos tipos de negación definidos en [14], la negación por default y la negación fuerte. La semántica de los LPODs está basada en una relación de preferencia sobre los *answer sets*.

El operador $\times$ ordena las opciones que se tienen en las reglas de acuerdo a su preferencia. El orden de preferencia se establece de acuerdo a la posición que toma la opción dentro de la regla. De izquierda a derecha, está la opción más preferida hasta la opción menos preferida dentro de la expresión de disyunción ordenada. El operador $\times$ está permitido sólo para aparecer en la cabeza de las reglas de un LPOD.

Definición 3.1. [9] Un LPOD consiste de un conjunto finito de reglas con disyunción ordenada de la forma forma:

$$C_1 \times \cdots \times C_n \leftarrow A_1, \ldots, A_m, not\, B_1, \ldots, not\, B_k. \tag{3.1}$$

Donde los C_i, A_j, B_l son literales instanciadas. □

La regla 3.1 está dividida en dos partes: la parte a la izquierda de $\leftarrow$ es la cabeza de la regla y la parte a la derecha es el cuerpo de la regla. En la cabeza de la regla, las literales $C_1 \ldots C_n$ son las opciones de la regla. Lo que la regla representa es: Si el cuerpo de la regla es verdadero y C_1 es posible, entonces C_1; si C_1 no es posible, entonces C_2; $\ldots$; si ninguno de $C_1, \ldots, C_{n-1}$ es posible entonces C_n.

Algunos casos especiales se derivan de acuerdo a los valores que tomen n, m, k en la regla 3.1. Los programas extendidos son el resultado de que $n = 1$ en todas las reglas. Las restricciones se presentan si $n = 0$, teniendo reglas con forma $\leftarrow$ *cuerpo*. Los hechos se tienen cuando $m = k = 0$, y es usual omitir $\leftarrow$.

A continuación se muestra un ejemplo tomado de [10] que describe un problema cuya solución tiene que ver con preferencias y su representación utiliza reglas con disyunción ordenada.

Ejemplo 3.1. [10] Se tiene que para la elección de un postre usted tiene que elegir entre `helado` o `pastel`, prefiriendo el `helado`. Para acompañarlo usted prefiere tomar `café` sobre tomar `té`. Con la restricción de que no puede elegir `café` para acompañar el `helado`.

El problema anterior se representa con el siguiente LPOD, denotado como P:

$$
\begin{aligned}
r_1 &: \ helado \times pastel. \\
r_2 &: \ cafe \times te. \\
r_3 &: \ \leftarrow helado, cafe.
\end{aligned}
$$

Donde r_1, r_2, r_3 son los identificadores para las reglas. Teniendo que las reglas r_1 y r_2 son hechos y r_3 es una restricción. □

Una vez que el problema está representado por reglas de un LPOD se tiene que encontrar la solución preferida. Para llegar a la solución preferida las reglas se tienen que descomponer en sus opciones. Las opciones de las reglas dan lugar a nuevos programas denominados *split programs*. De los *split programs* se encuentra un conjunto de *answer sets* que son soluciones candidatas a ser la solución preferida del problema. Un *answer set* candidato satisface a las reglas con cierto grado de satisfacción. La elección de la solución preferida se puede hacer por varios criterios sobre los conjuntos de reglas que tienen el mismo grado de satisfacción o bien el grado de satisfacción de la regla. A continuación se describe de forma detallada el proceso para encontrar la solución preferida para un LPOD.

Como se menciona anteriormente, cada regla de la forma 3.1 con $n \geq 1$, tiene n opciones. Alguna de las opciones en la regla puede pertenecer a los *answer sets* candidatos a la solución preferida. Para encontrar los *answer sets* candidatos es necesario tener reglas sin disyunción ordenada. Una regla sin disyunción ordenada se obtiene por las opciones de la regla.

Definición 3.2. [9] Sea r la regla con disyunción ordenada $C_1 \times \cdots \times C_n \leftarrow Cuerpo$. Se define a la k-ésima opción de r para $1 \leq k \leq n$ como:

$$r^k = C_k \leftarrow Cuerpo, not\, C_1, \ldots, not\, C_{k-1}.$$

□

Ejemplo 3.2. Considerando el LPOD P del ejemplo 3.1. Las opciones para la regla r_1 dada como $helado \times pastel$ son:

$$r_1^1: \quad helado \leftarrow .$$
$$r_1^2: \quad pastel \leftarrow not\, helado.$$

y para la regla r_2 dada como $cafe \times te$ las opciones son:

$$r_2^1: \quad cafe \leftarrow .$$
$$r_2^2: \quad te \leftarrow not\, cafe.$$

□

Con las opciones de las reglas de un LPOD, se definen programas libres de disyunción ordenada, los cuales son el resultado de sustituir cada regla del LPOD por una de sus opciones. Estos programas son llamados *split programs*.

Definición 3.3. [9] Sea P un LPOD. Un *split program* P' se obtiene reemplazando cada regla r de P, por una de sus opciones. □

El número de *split programs* que se obtienen de un LPOD depende de las reglas con disyunción ordenada y las opciones que tienen, por tratarse de la sustitución de opciones en cada *split program*.

Ejemplo 3.3. Considerando nuevamente el LPOD del ejemplo 3.1 y las opciones de sus reglas obtenidas en el ejemplo 3.2 se tienen los siguientes 4 *split programs*:

$$
\begin{array}{llll}
P'_1: & helado \leftarrow . & P'_2: & helado \leftarrow . \\
 & cafe \leftarrow . & & te \leftarrow not\, cafe. \\
 & \leftarrow cafe, helado. & & \leftarrow cafe, helado. \\
P'_3: & pastel \leftarrow not\, helado. & P'_4: & pastel \leftarrow not\, helado. \\
 & cafe \leftarrow . & & te \leftarrow not\, cafe. \\
 & \leftarrow cafe, helado. & & \leftarrow cafe, helado.
\end{array}
$$

□

Los *answer sets* para un *split program* de un LPOD corresponden también a los *answer sets* del programa original. Los *answer sets* de un LPOD son considerados como soluciones candidatas a la solución preferida.

Definición 3.4. [9] Dado un LPOD P. Si M es un *answer set* de un *split program* de P, entonces es también un *answer set* de P. □

Ejemplo 3.4. Considerando el LPOD del ejemplo 3.1 y sus *split programs* obtenidos en el ejemplo 3.3. Se tiene que para el *split program* P'_1 no hay un *answer set*. Esto es porque el conjunto $\{helado, cafe\}$ se anula por la restricción $\leftarrow cafe, helado$.

Del *split program* P'_2, se obtiene el *answer set* M_1 formado por $\{helado, te\}$. Para el *split program* P'_3, se tiene el conjunto $\{pastel, cafe\}$ que forma al *answer set* M_2. Finalmente, para el cuarto *split program* P'_4, se tiene el *answer set* M_3 formado por $\{pastel, te\}$.

Entonces los *answer sets* M_1, M_2, M_3 son el conjunto de soluciones candidatas del problema ejemplo 3.1. □

Al ser un *answer set* una solución candidata de un LPOD, está satisface a las reglas del programa. La manera en que un *answer set* satisface a las reglas es denominada como *grado de satisfacción*. El valor del grado de satisfacción se determina de acuerdo a la forma en que la regla se satisface o no por el *answer set*.

Definición 3.5. [9] Sea M un *answer set* de un LPOD P. Entonces M satisface a la regla $C_1 \times \cdots \times C_n \leftarrow A_1, \ldots, A_m, notB_1, \ldots, notB_k$

1. Con grado de satisfacción 1, si $A_l \notin M$, para algún l, ó $B_i \in M$ para algún i.

2. Con grado de satisfacción $j\,(1 \leq j \leq n)$ si todas las $A_l \in M$, todas las $B_i \notin M$ y $j = min\{r | C_r \in M\}$.

El grado de la regla r para M es denotado como $deg_M(r)$. □

Los grados de satisfacción se pueden ver como penalizaciones: El mayor grado es lo menos preferido. Si el cuerpo de la regla no se satisface, esto no es razón para que la regla sea insatisfecha y el mayor grado posible, 1, se asigna. Si la regla se satisface, el valor del grado de satisfacción se determina por la opción más preferida que está en el *answer set*.

Los grados de satisfacción de las reglas de un LPOD P, son la base para definir el criterio de preferencia sobre el conjunto de soluciones candidatas de P. El criterio de preferencia se puede definir a partir del grado de satisfacción o de los conjuntos de reglas que tienen el mismo grado de satisfacción. Los conjuntos de reglas de P que satisfacen al *answer set* M con grado j se denotan como $S^j_M(P)$.

Ejemplo 3.5. Sea el LPOD P del ejemplo 3.1 y los *answer sets*: $M_1 = \{helado, te\}$, $M_2 = \{pastel, cafe\}$ y $M_3 = \{pastel, te\}$, calculados en el ejemplo 3.4. Los grados de satisfacción para las reglas de P de acuerdo a M_1, M_2, M_3 son los siguientes valores:

$$\begin{array}{ll} deg_{M_1}(r_1) = 1 & deg_{M_1}(r_2) = 2 \\ deg_{M_2}(r_1) = 2 & deg_{M_2}(r_2) = 1 \\ deg_{M_3}(r_1) = 2 & deg_{M_3}(r_2) = 2 \end{array}$$

Las reglas por tratarse de hechos tienen el cuerpo vacío, por lo que para determinar el grado de satisfacción se considera el segundo caso de la definición 3.5.

Agrupando las reglas con el mismo grado de satisfación, se tienen los conjuntos:

$$\begin{array}{ll} S^1_{M_1} = \{r_1\} & S^2_{M_1} = \{r_2\} \\ S^1_{M_2} = \{r_2\} & S^2_{M_2} = \{r_1\} \\ S^1_{M_3} = \{\ \} & S^2_{M_3} = \{r_1, r_2\} \end{array}$$

□

El determinar los grados de satisfacción para las reglas de un LPOD ayuda a determinar el criterio de preferencia que hay entre los *answer sets* que se tienen como soluciones candidatas. En la siguiente sección se definen los criterios con los que trabajan los LPODs para determinar la solución preferida de un problema.

3.1.1. Criterios de Preferencia

El enfoque de LPOD, trabaja con tres criterios para elegir la solución preferida: preferencia por inclusión de conjuntos, preferencia por cardinalidad de conjuntos y preferencia por Pareto. Estos criterios ayudan a determinar la solución preferida de acuerdo a los grados de satisfacción de las reglas para los *answer sets* que se tienen como soluciones candidatas al problema.

Preferencia por Inclusión

La solución preferida por inclusión de conjuntos elige entre conjuntos de reglas de dos *answer sets* M_1 y M_2 que son candidatos a solución preferida de un LPOD P. El *answer set* preferido se tiene de comparar dos conjutos $S^k_{M_1}(P)$ y $S^k_{M_2}(P)$ de acuerdo a la relación de si un conjunto es subconjunto del otro o viceversa. El *answer set* preferido es el del conjunto que contiene al otro.

Definición 3.6. [10] Dados dos *answer sets* M_1 y M_2 para un LPOD P. M_1 es la solución preferida de P por inclusión de conjuntos sobre M_2, ($M_1 >_i M_2$), sí y solo sí se tiene un grado k tal que $S^k_{M_2}(P) \subset S^k_{M_1}(P)$, y para toda $j < k$ se tiene que $S^j_{M_2}(P) = S^j_{M_1}(P)$. □

Si dados dos *answer sets* M_1, M_2 no es posible aplicar la definición 3.6, se tiene que ambos conjuntos son igualmente preferidos. Para el caso de que se elija una solución preferida entre todos los *answer sets* de un LPOD, se tiene un *answer set preferido por inclusión.*

Definición 3.7. [10] Un conjunto de literales M es un *answer set preferido por inclusión* de un LPOD P, sí y solo sí M es un *answer set* de P y no hay otro answer set M' de P tal que suceda $M' >_i M$. □

Ejemplo 3.6. Consideremos el LPOD P del ejemplo 3.1, los *answer sets* : $M_1 = \{helado, te\}$, $M_2 = \{pastel, cafe\}$ y $M_3 = \{pastel, te\}$, calculados en el ejemplo 3.4 y los conjuntos de las reglas calculados en el ejemplo 3.5 dados como:

$$\begin{array}{ll} S^1_{M_1} = \{r_1\} & S^2_{M_1} = \{r_2\} \\ S^1_{M_2} = \{r_2\} & S^2_{M_2} = \{r_1\} \\ S^1_{M_3} = \{\,\} & S^2_{M_3} = \{r_1,\, r_2\} \end{array}$$

La solución preferida por inclusión de conjuntos entre los *answer sets* candidatos está dada como se muestra a continuación.

Al comparar los conjuntos de reglas $S^1_{M_1}$ y $S^1_{M_2}$ para los *answer sets* M_1 y M_2 se tiene que no tienen ningún elemento en común, entonces se tiene que ambas son soluciones igualmente preferidas.

Para la comparación entre los conjuntos de reglas de los *answer sets* M_1 y M_3, se tiene que $S^1_{M_3} \subset S^1_{M_1}$ por lo que M_1 es preferido por inclusión sobre M_3.

Para las soluciones candidatas M_2 y M_3, se tiene que M_2 es preferido por inclusión sobre M_3, porque se tiene que $S^1_{M_3} \subset S^1_{M_2}$. Por lo tanto, los *answer sets* M_1 y M_2 corresponden a las soluciones preferidas. □

Preferencia por Cardinalidad

Otro criterio que se tiene para obtener la solución preferida es el criterio por cardinalidad de conjuntos. Este criterio compara la cantidad de reglas en los conjuntos para los grados de satisfacción. El *answer set* que satisface más reglas con cierto grado es preferido sobre el otro.

Definición 3.8. [10] Dados dos *answer set* M_1 y M_2 de un LPOD P. Se dice que M_1 es preferido por cardinalidad de conjuntos sobre M_2, ($M_1 >_c M_2$), si y solo si, para un grado k, se tiene $|S^k_{M_1}(P)| > |S^k_{M_2}(P)|$, y para toda $j < k$ se tiene que $|S^j_{M_1}(P)| \geq |S^j_{M_2}(P)|$. □

Si dados dos *answer sets* M_1, M_2 no es posible lo que se establece en la definición 3.8, M_1 y M_2 se consideran igualmente preferidos. Para el caso de que se elija un solo *answer set* entre todas las soluciones candidatas de un LPOD, se tiene un *answer set preferido por cardinalidad.*

Definición 3.9. [10] Un conjunto de literales M es un *answer set preferido por cardinalidad* de un LPOD P, sí y solo sí, M es un *answer set* de P y no hay otro *answer set* M' de P tal que suceda $M' >_c M$. □

Ejemplo 3.7. Considerando el LPOD P del ejemplo 3.1, los *answer sets* : $M_1 = \{helado, te\}$, $M_2 = \{pastel, cafe\}$ y $M_3 = \{pastel, te\}$, calculados en el ejemplo 3.4 y los conjuntos de las reglas:

$$\begin{array}{ll} S^1_{M_1} = \{r_1\} & S^2_{M_1} = \{r_2\} \\ S^1_{M_2} = \{r_2\} & S^2_{M_2} = \{r_1\} \\ S^1_{M_3} = \{\,\} & S^2_{M_3} = \{r_1, r_2\} \end{array}$$

calculados en el ejemplo 3.5. La solución preferida por cardinalidad de conjuntos entre los *answer sets* candidatos se describe a continuación: al comparar los valores $|S^j_{M_1}|$ y $|S^j_{M_2}|$ para $j = 1, 2$ de las soluciones candidatas M_1 y M_2, se obtiene que son iguales por lo que ambas son soluciones igualmente preferidas.

Para las soluciones candidatas M_2 y M_3, se que tiene que $|S^1_{M_2}| > |S^1_{M_3}|$ por lo que M_2 es preferido por cardinalidad sobre M_3.

Para los *answer sets* candidatos M_1 y M_3, se tiene que $|S^1_{M_1}| > |S^1_{M_3}|$, estableciendo que M_1 es preferido por cardinalidad sobre M_3. Por lo tanto, los conjuntos M_1 y M_2 son las soluciones preferidas. $\square$

Preferencia por Pareto

El tercer criterio que también ayuda a determinar la solución preferida, se denomina preferencia por Pareto. Este criterio elige entre los grados de satisfacción de las reglas de un LPOD dados por dos *answer sets* candidatos. El *answer set* preferido por Pareto se tiene cuando el grado de satisfacción para una regla es menor que el grado de satisfacción para esa regla definido por otro *answer set* y además para el resto de las reglas el grado de satisfacción no es mayor.

Definición 3.10. [10] Dados dos *answer sets* M_1 y M_2 para un LPOD P. Se tiene que M_1 es preferido por Pareto sobre M_2, $(M_1 >_P M_2)$, sí y solo sí para una regla $r \in P$ se tiene que $deg_{M_1}(r) < deg_{M_2}(r)$, y no exista $r' \in P$ que $deg_{M_1}(r') > deg_{M_2}(r')$. $\square$

Si para dos *answer sets* M_1, M_2 no se puede aplicar la definición 3.10, los *answer sets* se consideran igualmente preferidos. Para el caso de que se elija una sola solución preferida entre el conjunto de todos los *answer sets* candidatos de un LPOD, se tiene un *answer set preferido por Pareto.*

Definición 3.11. [10] Un conjunto de literales M es un *answer set preferido por Pareto* de un LPOD P, sí y solo sí M es un *answer set* de P y no hay otro answer set M' de P tal que $M' >_P M$. □

Ejemplo 3.8. Tomando el LPOD P del ejemplo 3.1, los *answer sets* : $M_1 = \{helado, te\}$, $M_2 = \{pastel, cafe\}$ y $M_3 = \{pastel, te\}$, calculados en el ejemplo 3.4 y los grados de satisfacción:

$$deg_{M_1}(r_1) = 1 \qquad deg_{M_1}(r_2) = 2$$
$$deg_{M_2}(r_1) = 2 \qquad deg_{M_2}(r_2) = 1$$
$$deg_{M_3}(r_1) = 2 \qquad deg_{M_3}(r_2) = 2$$

calculados en el ejemplo 3.5. La solución preferida por Pareto entre los *answer sets* candidatos está dada como sigue.

Para elegir entre los conjuntos candidatos M_1 y M_2, se tiene que $deg_{M_1}(r_1) < deg_{M_2}(r_1)$ y $deg_{M_1}(r_2) > deg_{M_2}(r_2)$, por lo que se dice que M_1 y M_2 son soluciones igualmente preferidas por Pareto.

En la elección entre los *answer sets* M_1 y M_3, se tiene que $M_1 >_P M_3$, Esto es porque $deg_{M_1}(r_i) \leq deg_{M_3}(r_i)$ para $i = 1, 2$.

Finalmente, para escoger entre M_2 y M_3, se tiene que $deg_{M_2}(r_i) \leq deg_{M_3}(r_i)$ para $i = 1, 2$. Quedando M_2 como solución preferida por Pareto. Siendo M_1 y M_2 las soluciones preferidas del problema. □

Como se acaba de mostrar los LPODs trabajan con reglas de disyunción ordenada definidas en términos de literales instanciadas. En [23] se presenta una extensión para los LPODs, los programas lógicos con disyunción ordenada extendida, LPEODs (*Logic Programs with Extended Ordered Disjunction*). Esta ampliación define a las reglas con disyunción ordenada en términos de fórmulas proposicionales bien formadas. Ya que se considera que una sintaxis más amplia para estas reglas puede dar algunas ventajas en la representación y comprensión de problemas con preferencias. Esta extensión se presenta en el apéndice A.

3.2. Programación Lógica con Preferencias

En [24], los autores muestran otra alternativa para la representación de problemas con preferencias, los Programas Lógicos con Preferencias, abreviados como PLPs por su definición en inglés, *Preference Logic Programs*. Los PLPs están compuestos por dos partes: (1) el programa lógico que representa la especificación del problema y genera las soluciones del problema, y (2) el programa $Pref$, formado por reglas de preferencia que establecen la preferencia sobre las soluciones. El programa lógico está formado por un conjunto finito de fórmulas bien formadas o reglas de la forma $f \leftarrow g$, donde f, g son fórmulas bien formadas.

Las reglas de preferencia permiten especificar un orden de preferencia entre las soluciones de un problema obtenidas del programa lógico. Las reglas de preferencia son expresadas en términos de fórmulas proposicionales bien formadas y no solamente de literales. El uso de fórmulas proposicionales bien formadas es con el propósito de expresar problemas con preferencias con una teoría más general.

Las reglas de preferencia se identifican porque usan el *operador de preferencia*, representado por el símbolo $*$. El operador de preferencia se usa para conectar una lista ordenada de fórmulas en la cabeza de las reglas de preferencia de los PLPs. Cada fórmula de la cabeza representa una opción con un valor de preferencia sobre algo. El valor de preferencia se asigna de izquierda a derecha, teniendo de la opción más preferida a la menos preferida.

Definición 3.12. [24] Una regla de preferencia es una fórmula de la forma

$$f_1 * \cdots * f_n \stackrel{pr}{\leftarrow} g. \tag{3.2}$$

Donde $f_1, \ldots, f_n, g$ son fórmulas proposicionales bien formadas. Un PLP es un conjunto finito de reglas de preferencia y un conjunto arbitrario de fórmulas bien formadas. □

Las fórmulas $f_1, \ldots, f_n$ son llamadas *opciones de la regla de preferencia*. Los *answer sets* preferidos de un PLP P, son los *answer sets* de P tal que para cada regla de preferencia ocurre lo siguiente: si la primera opción ocurre, en otro caso la segunda opción ocurre, de lo contrario la tercera opción ocurre, y así sucesivamente. Si ninguna de las opciones de cada regla de preferencia ocurre en los *answer sets* de P entonces todos los *answer sets* de P son preferidos. En el caso de que g sea verdadero en la regla de preferencia 3.2, la regla se escribe como $f_1 * \cdots * f_n \stackrel{pr}{\leftarrow}$.

A continuación se describe una modificación de un ejemplo tomado de [4] que tiene que ver con preferencias y su representación utiliza reglas de un PLP.

Ejemplo 3.9. Un show de televisión dirige un juego donde el `primer lugar` recibe un premio de `$200,000.00`, el `segundo lugar` recibe un premio de `$100,000.00`. John busca `participar`, si es posible, en otro caso se `retira`. Si `participa`, busca ganar los `$200,000.00` si es posible, en otro caso ganar los `$100,000.00`.

El problema anterior se representa con el siguiente PLP, denotado como P:

$$\begin{aligned}
r_1 &: \ participar \vee retirarse \leftarrow . \\
r_2 &: \ ganar(200,000{,}00) \vee ganar(100,000{,}00) \leftarrow participar. \\
r_3 &: \ participar * retirarse \stackrel{pr}{\leftarrow} . \\
r_4 &: \ ganar(200,000{,}00) * ganar(100,000{,}00) \stackrel{pr}{\leftarrow} participar.
\end{aligned}$$

Se tiene que los r_i son los identificadores para las reglas. El programa lógico está compuesto por el conjunto de reglas $\{r_1, r_2\}$ y $Pref_P$ por el conjunto $\{r_3, r_4\}$. $\square$

Con las reglas del PLP que representan un problema se busca la solución preferida de un conjunto de soluciones candidatas. Las soluciones generadas por el programa lógico del PLP son las soluciones candidatas. Cada solución candidata tiene un grado de satisfacción que hace que se cumplan las reglas de preferencia en $Pref$. La elección de la solución preferida se hace por medio de un criterio de preferencia sobre los grados de satisfacción establecidos por los *answer sets* para las reglas .

Los *answer sets* de un PLP son también *answer sets* del programa que se tienen sin considerar las reglas de preferencia (véase sección 2.2). Esto queda definido como se muestra a continuación.

Definición 3.13. [24] Dado $Pref_P$ como un conjunto de reglas de preferencia de un PLP P. Sea M un conjunto de átomos. M es un *answer set* de P si y solo si M es un *answer set* de $P \setminus Pref_P$. $\square$

Ejemplo 3.10. Considerando la parte lógica para el PLP P del ejemplo 3.9, es decir, sin la especificación de las preferencias de John. De las reglas r_1, r_2 se obtienen tres *answer sets*, definidos como $M_1 = \{participar, ganar(200,000{,}00)\}$, $M_2 = \{participar, ganar(100,000{,}00\}$ y $M_3 = \{retirarse\}$.

Por lo tanto, los *answer sets* M_1, M_2 y M_3 son soluciones candidatas del problema PLP P del ejemplo 3.9 $\square$

Los *answer sets* candidatos de un PLP satisfacen de alguna manera a las reglas de preferencia. La manera en que una regla de preferencia se satisface por una solución candidata se denomina *grado de satisfacción*. El valor del grado de satisfacción se determina de acuerdo a cómo una regla de preferencia se satisface o no.

Definición 3.14. [24] Sea M un *answer set* de un PLP P. Dada $r = f_1 * \cdots * f_n \overset{pr}{\leftarrow} g$ como una regla de preferencia de P. Se define el grado de satisfacción de r en M, denotada por $deg_M(r)$, como una regla de correspondencia que se define en la siguiente función:

1. Si $M \cup \neg(\mathcal{L}_P \setminus M) \nvdash_{\mathrm{G}_3} g$, el grado de satisfacción es $deg_M(r) = 1$.

2. Si $M \cup \neg(\mathcal{L}_P \setminus M) \vdash_{\mathrm{G}_3} g$, entonces $deg_M(r) = min\,\{i \mid M \cup \neg(\mathcal{L}_P \setminus M) \vdash_{\mathrm{G}_3} f_i\}$.

3. Si $M \cup \neg(\mathcal{L}_P \setminus M) \vdash_{\mathrm{G}_3} g$, y no hay un $1 \leq i \leq n$ tal que $M \cup \neg(\mathcal{L}_P \setminus M) \vdash_{\mathrm{G}_3} f_i$, se tiene que $deg_M(r) = n + 1$

□

Los valores de los grados de satisfacción representan el nivel de preferencia, el valor más pequeño es lo más preferido. El grado de satisfacción en el primer caso es lo más preferido, 1, esto se tiene porque si el cuerpo de la regla no es posible, y no hay razón para que la regla sea insatisfecha. En el segundo caso, se tiene que la regla se cumple y el grado de satisfacción está dado por la posición de la opción en la cabeza de la regla que es la más preferida dentro del *answer set*. Finalmente, en el tercer caso se tienen las reglas que se satisfacen sólo en el cuerpo, pero no es posible una opción de la cabeza de la regla, por lo que el grado de satisfacción está determinado por el número de opciones más uno, $n + 1$, siendo esto lo menos preferido.

Los grados de satisfacción de las reglas de preferencia de un PLP P, son la base para definir el criterio de preferencia sobre las soluciones candidatas de P. El grado de satisfacción de cada regla de preferencia permite definir un conjunto de reglas de preferencia que tienen el mismo grado. Estos conjuntos se utilizan para encontrar el *answer set* preferido de un PLP. Los conjuntos de reglas de preferencia de P que se satisfacen por el *answer set* M con grado j se denotan como $S^j_M(P)$.

Ejemplo 3.11. Retomando el PLP P del ejemplo 3.9 y los *answer sets* candidatos $M_1 = \{participar, ganar(200,000{,}00)\}$, $M_2 = \{participar, ganar(100,000{,}00\}$ y $M_3 = \{retirarse\}$ obtenidos en el ejemplo 3.10. Para las reglas de preferencia de P:

$$r_3: \quad participar * retirarse \overset{pr}{\leftarrow} .$$
$$r_4: \quad ganar(200,000{,}00) * ganar(100,000{,}00) \overset{pr}{\leftarrow} participar.$$

Los grados de satisfacción de las reglas con respecto a los *answer sets* M_1, M_2 y M_3 están definidos como:

$$deg_{M_1}(r_3) = 1 \qquad deg_{M_1}(r_4) = 1$$
$$deg_{M_2}(r_3) = 1 \qquad deg_{M_2}(r_4) = 2$$
$$deg_{M_3}(r_3) = 2 \qquad deg_{M_3}(r_4) = 1$$

Para el valor de $deg_{M_3}(r_4)$ se tiene el primer caso de la definición 3.14 para calcular el grado de satisfacción, esto es, porque M_3 no satisface el cuerpo de la regla r_4. Para el resto de los valores, se utiliza el segundo caso de la definición 3.14 para obtener el grado de satisfacción.

Agrupando las reglas de acuerdo al grado de satisfacción para los *answer sets*, están dados los conjuntos:

$$S^1_{M_1} = \{r_4, r_5\} \qquad S^2_{M_1} = \{\}$$
$$S^1_{M_2} = \{r_4\} \qquad S^2_{M_2} = \{r_5\}$$
$$S^1_{M_3} = \{r_5\} \qquad S^2_{M_3} = \{r_4\}$$

□

Al determinar los grados de satisfacción para las reglas de un PLP, se puede determinar un criterio de preferencia que hay entre los *answer sets* candidatos. En la siguiente sección se definen los criterios con los que trabajan los PLPs para determinar la solución preferida de un problema.

3.2.1. Criterios de Preferencia.

Los PLPs trabajan con dos criterios para determinar la solución preferida: preferencia por inclusión y preferencia por cardinalidad. Los criterios de preferencia para los PLPs están inspirados en los criterios del enfoque LPOD. Los criterios ayudan a determinar la solución preferida de acuerdo a los conjuntos de reglas que tienen el mismo grado de satisfacción para un *answer set* candidato.

Preferencia por Inclusión

Para obtener la solución preferida por el criterio de inclusión de conjuntos, se necesitan comparar dos *answer sets* M_1 y M_2 que son candidatos a solución preferida de un PLP P. El *answer set* preferido resulta de la comparación de los conjutos $S^k_{M_1}(P)$ y $S^k_{M_2}(P)$ para un grado k, encontrando si alguno de ellos es subconjunto del otro. El *answer set* preferido es aquel que en su conjunto de reglas contiene al otro.

Definición 3.15. [24] Dados dos *answer sets* M_1 y M_2 para un PLP P. M_1 es preferido por inclusión de conjuntos sobre M_2, ($M_1 >_i M_2$), si y solo si hay un grado k tal que $S^k_{M_2}(P) \subset S^k_{M_1}(P)$, y para toda $j < k$ se tiene que $S^j_{M_1}(P) = S^j_{M_2}(P)$. □

Si para dos *answer sets* M_1, M_2 no se puede aplicar la definición 3.15, se tiene que M_1 y M_2 son igualmente preferidos. Si del conjunto de *answer sets* de un PLP se tiene a un *answer set* preferido por inclusión, este se denomina *answer set preferido por inclusión.*

Definición 3.16. [24] Un conjunto de literales M es un *answer set preferido por inclusión* de un LPOD P, si y solo si, M es un *answer set* de P y no hay otro *answer set* M' de P tal que $M' >_i M$. □

Ejemplo 3.12. Considerando el PLP P del ejemplo 3.9, los *answer sets* candidatos $M_1 = \{participar, ganar(200,000{,}00)\}$, $M_2 = \{participar, ganar(100,000{,}00\}$ y $M_3 = \{retirarse\}$ obtenidos en el ejemplo 3.10 y los conjuntos de reglas del ejemplo 3.11 dados como:

$$\begin{array}{ll} S^1_{M_1} = \{r_4, r_5\} & S^2_{M_1} = \{\} \\ S^1_{M_2} = \{r_4\} & S^2_{M_2} = \{r_5\} \\ S^1_{M_3} = \{r_5\} & S^2_{M_3} = \{r_4\} \end{array}$$

Al comparar los conjuntos de reglas $S^1_{M_1}$ y $S^1_{M_2}$ para los *answer sets* M_1 y M_2 se tiene que $S^1_{M_1} \subset S^1_{M_2}$, entonces se tiene que $M_1 >_i M_2$.

Para la comparación entre los conjuntos de reglas de los *answer sets* M_1 y M_3, se tiene que $S^1_{M_3} \subset S^1_{M_1}$ por lo que M_1 es preferido por inclusión sobre M_3.

Para las soluciones candidatas M_2 y M_3, se tiene que M_2 y M_3 son igualmente preferidos, porque los conjuntos $S^1_{M_2}$ y $S^1_{M_3}$ no tienen elementos en común.

Por lo tanto, el *answer set* M_1 corresponde a la solución preferida. Teniendo que M_1 es un *answer set preferido por inclusión.*

□

Preferencia por Cardinalidad

El segundo criterio con el que se puede elegir la solución preferida es el de cardinalidad de conjuntos. Este criterio compara la cantidad de elementos en los conjuntos de reglas que tienen el mismo grado de satisfacción de dos *answer sets* candidatos. La solución candidata que satisface más reglas para un grado es preferido sobre el otro.

Definición 3.17. [24] Dados dos *answer sets* candidatos M_1 y M_2 para un PLP P. M_1 es preferido por cardinalidad de conjuntos sobre M_2, ($M_1 >_c M_2$), si y solo si, para algún grado de satisfacción k se tiene que $|S^k_{M_1}(P)| > |S^k_{M_2}(P)|$, y para toda $j < k$ se tiene que $|S^j_{M_1}(P)| = |S^j_{M_2}(P)|$. □

Si se tienen dos *answer sets* M_1, M_2 tal que lo que establece la definición 3.17 no es posible, los *answer sets* candidatos son igualmente preferidos. Al determinar una solución preferida por cardinalidad dentro del conjunto de *answer sets*, se dice que ese conjunto es un *answer set preferido por cardinalidad.*

Definición 3.18. [24] Un conjunto de literales M es un *answer set preferido por cardinalidad* de un LPOD P, si y solo si, M es un *answer set* de P y no hay otro *answer set* M' de P tal que suceda $M' >_c M$. $\square$

Ejemplo 3.13. Sea el PLP P del ejemplo 3.9, los *answer sets* candidatos $M_1 = \{participar, ganar(200,000{,}00)\}$, $M_2 = \{participar, ganar(100,000{,}00\}$ y $M_3 = \{retirarse\}$ del ejemplo 3.10 y los conjuntos de reglas del ejemplo 3.11 dados como:

$$\begin{array}{ll} S^1_{M_1} = \{r_4,\, r_5\} & S^2_{M_1} = \{\} \\ S^1_{M_2} = \{r_4\} & S^2_{M_2} = \{r_5\} \\ S^1_{M_3} = \{r_5\} & S^2_{M_3} = \{r_4\} \end{array}$$

Al comparar los valores $|S^1_{M_1}|$ y $|S^1_{M_2}|$ se tiene que M_1 es preferido sobre M_2.

Para los *answer sets* candidatos M_1 y M_3, se tiene que $|S^1_{M_1}| > |S^1_{M_3}|$, estableciendo que M_1 es preferido por cardinalidad sobre M_3.

Para las soluciones candidatas M_2 y M_3, se que tiene que $|S^1_{M_2}| = |S^1_{M_3}|$ por lo que ambos *answer sets* son igualmente preferidos.

Por lo tanto, el conjunto M_1 es la solución preferida del problema. Teniendo que M_1 es un *answer set preferido por cardinalidad.*

$\square$

3.3. *Answer Sets* con Optimización

En [11], son introducidos los Programas en *Answer Sets* con Optimización, abreviados como ASOPs por su definición en inglés, *Answer Sets Optimization Programs*. Un ASOP está formado por dos partes, la primera es el programa generador P_{gen} que produce los *answer sets*, definiendo el conjunto de soluciones que son candidatas a solución preferida. La segunda parte es el programa de preferencias P_{pref}, que expresa las preferencias para comparar los *answer sets* de P_{gen}. Las reglas de P_{gen} son restricciones fuertes que especifican las condiciones de los *answer sets*. Las reglas de P_{pref} son restricciones débiles que describen las condiciones bajo las cuales un *answer set* es considerado mejor que otro.

Tener por separado la forma de obtener los *answer sets* y la de compararlos tiene dos ventajas importantes [11]:

1. El método de comparar los *answer sets* es independiente del que se tiene para el programa generador. P_{gen} puede ser cualquier tipo de programa lógico (por ejemplo: normal, extendido, disyuntivo), que involucre cardinalidad o peso en los átomos, mientras esté dado por una semántica bien definida para una colección de conjuntos de literales.

2. Las preferencias en P_{pref} (restricciones débiles), se pueden especificar de forma independiente de P_{gen} (restricciones fuertes). Esto hace la elección de preferencias más fácil ya que se encuentran por separado.

Las reglas de P_{pref} utilizan el operador de preferencia, representado por $>$. El operador de preferencia conecta una lista ordenada de opciones en la cabeza de las reglas. El orden de preferencia para las opciones se asigna de izquierda a derecha, teniendo de la más preferida a la menos preferida.

Definición 3.19. [11] Dado A como un conjunto de átomos. El programa de preferencias sobre A es un conjunto finito de reglas de la forma

$$C_1 > \ldots > C_k \leftarrow a_1, \ldots, a_n, not\, b_1, \ldots, not\, b_m. \tag{3.3}$$

donde las $a_i s$, $b_j s$ son literales de átomos en A, y las $C_i s$ son combinaciones booleanas sobre A. □

Las combinaciones booleanas C_i son opciones para las reglas de preferencia. La regla (3.3) se interpreta como: si un *answer set* S contiene $a_1, \ldots, a_n$ y no contiene alguna de las literales $b_1, \ldots, b_m$ entonces C_1 es preferida sobre C_2, C_2 sobre C_3, $\ldots$, C_{k-1} sobre C_k.

Definición 3.20. [11] Sea A un conjunto de átomos. Una *combinación booleana* sobre A es una fórmula constrúida por átomos de A, disyunción, conjunción, negación fuerte ($\neg$) y negación por default (not). Las restricciones que se tienen son: la negación fuerte aparece sólo enfrente de átomos y la negación por default aparece sólo enfrente de literales. □

El uso de las combinaciones booleanas es mejor que sólo utilizar literales dentro de las cabezas para las reglas de preferencia, teniendo con esto una mejor expresividad. Con el uso de la conjunción se puede expresar que ciertas combinaciones son preferidas sobre otras y el uso de la disyunción puede expresar que ciertas opciones son igualmente preferidas.

Definición 3.21. [11] Se tiene a S como un conjunto de literales. La satisfacción de una combinación booleana C en S, ($S \models C$), está definida como:

$$
\begin{array}{ll}
S \models l\,(l\,literal) & si\ y\ solo\ si\ l \in S \\
S \models not\, l\,(l\,literal) & si\ y\ solo\ si\ l \notin S \\
S \models C_1 \vee C_2 & si\ y\ solo\ si\ S \models C_1\ o\ S \models C_2 \\
S \models C_1 \wedge C_2 & si\ y\ solo\ si\ S \models C_1\ y\ S \models C_2
\end{array}
$$

□

Ahora con lo que se ha mencionado, se puede definir la noción de un programa en *answer sets* con optimización.

Definición 3.22. [11] Un ASOP es un par (P_{gen}, P_{pref}), donde P_{gen} es un programa lógico llamado programa generador y P_{pref} es el programa de preferencias. □

Con la representación de un problema como un ASOP, se busca tener una solución preferida sobre un conjunto de soluciones candidatas. El conjunto de soluciones candidatas se forma por los *answer sets* generados de P_{gen} del ASOP. El programa P_{gen} que genera los *answer sets* puede ser de cualquier tipo, sólo se requiere que la semántica esté dada en términos de conjuntos de literales o *answer sets*. Cada *answer set* del conjunto de soluciones candidatas determina un grado de satisfacción para las reglas de P_{pref}. El grado de satisfacción de cada regla con respecto a un *answer set* define un vector de satisfacción. Los vectores de satisfacción para los *answer sets* son comparados para obtener un *answer set* como solución preferida del ASOP.

A continuación se muestra un ejemplo tomado de [11], el cual involucra preferencias, por lo que se modela como un ASOP para determinar la solución preferida.

Ejemplo 3.14. Un menú debe consistir de una `entrada`, el `platillo principal`, un `postre` y una `bebida`. Para la `entrada` se tienen las opciones de `sopa` y `ensalada`. Las opciones disponibles para el `platillo principal` son: `pescado` y `carne`. En el caso de la `carne`, se prefiere el vino `rojo` o la `cerveza` sobre el vino `blanco`. Para el caso del `pescado`, se prefiere el vino `blanco` sobre el vino `rojo` sobre la `cerveza`. Solo hay `helado` y `pay` disponibles para el `postre`. Si la `bebida` es `cerveza`, se prefiere el `pay` sobre el `helado`. Se tiene que por el momento no se tienen disponibles los productos: vino `rojo`, vino `blanco` y `pay`.

El problema descrito se representa con el siguiente ASOP:

$$\begin{aligned}
r_1 &: \quad sopa \vee ensalada \leftarrow entrada.\\
r_2 &: \quad carne \vee pescado \leftarrow principal.\\
r_3 &: \quad pay \vee helado \leftarrow postre.\\
r_4 &: \quad rojo \vee blanco \vee cerveza \leftarrow bebida\\
r_5 &: \quad entrada.\\
r_6 &: \quad principal.\\
r_7 &: \quad postre.\\
r_8 &: \quad bebida.\\
r_9 &: \quad \leftarrow rojo.\\
r_{10} &: \quad \leftarrow blanco.\\
r_{11} &: \quad \leftarrow pay.\\
r_{12} &: \quad blanco > rojo > cerveza \leftarrow pescado.\\
r_{13} &: \quad rojo \vee cerveza > blanco \leftarrow carne.\\
r_{14} &: \quad pay > helado \leftarrow cerveza.
\end{aligned}$$

Los r_i son los identificadores de las reglas del programa. P_{gen} está formado por el conjunto de las reglas de r_1 a r_{11}. De r_5 a r_8 las reglas representan hechos. Las reglas de r_9 a r_{11} son consideradas restricciones. P_{pref} lo forman las reglas de r_{12} a r_{14}. De P_{gen} se obtiene el conjunto de soluciones candidatas formado por los siguientes *answer sets*:

S_1 : $\{bebida, postre, principal, entrada, helado, cerveza, carne, sopa\}$.

S_2 : $\{bebida, postre, principal, entrada, helado, cerveza, carne, ensalada\}$.

S_3 : $\{bebida, postre, principal, entrada, helado, cerveza, pescado, sopa\}$.

S_4 : $\{bebida, postre, principal, entrada, helado, cerveza, pescado, ensalada\}$.

Teniendo que los S_i son los identificadores de los *answer sets*.

□

La manera en que P_{pref} determina la preferencia entre los *answer sets* obtenidos de P_{gen} se da por medio del *vector de satisfacción*. El vector de satisfacción se define a partir de los grados de satisfacción de las reglas de P_{pref} de acuerdo a los *answer sets* de P_{gen}. El grado de satisfacción se determina a partir de cómo la regla de preferencia se satisface por el *answer set*.

Definición 3.23. [11] Sea un ASOP $P = (P_{gen}, P_{pref})$ y S un *answer set* de P. Dada $r = C_1 > \ldots > C_k \leftarrow a_1, \ldots, a_n, not\, b_1, \ldots, not\, b_m$ una regla de P_{pref}. Se define el grado de satisfacción de r en S, denotada por $v_S(r)$, de acuerdo a lo siguiente:

1. Si el cuerpo de r no se satisface en S, esto es, $a_i \notin S$ para algún $i \leq n$, o algún $b_j \in S$ para $j \leq m$; se tiene que $v_S(r) = I$.

2. Si el cuerpo de r se satisface en S y ninguno de los $C_i s$ se satisface en S, entonces $v_S(r) = I$.

3. Si el cuerpo de r se satisface en S y al menos un C_i se satisface en S, el grado de satisfacción está definido por $v_S(r) = min\{i : S \models C_i\}$

□

Los valores de los grados de satisfacción representan el nivel de preferencia, donde el valor más pequeño, 1, es el más preferido. De la definición 3.23, para el primer caso, la regla r es *irrelevante* para S, esto porque la regla no se aplica en S. Para el segundo caso, se tiene que de la lista de preferencias en la regla, ninguna de las opciones está en S; aquí se considera otro tipo de *irrelevancia*. La irrelevancia se tiene porque en el *answer set* existen opciones que no contempla la regla, siendo las opciones de la regla irrelevantes para el *answer set*. Para el último caso, la preferencia expresada en la regla se satisface en algún grado, ya que al menos una C_i aparece en S.

Para la relación entre I y los otros grados de satisfacción, en [11] se exponen dos opciones. La primera, considera a I como incomparable a otros valores, basado en la visión que la "irrelevancia" no se puede comparar con los otros grados de satisfacción. La segunda, supone que las opciones de la regla violan las preferencias por eso existe la irrelevancia, teniendo que I es preferible sobre las opciones de la regla. La opción que se adopta en [11] es la segunda, teniendo el siguiente orden para los grados de satisfacción: $1, I > 2 > \ldots > k$. El valor de I y 1 son considerados *igualmente buenos*, ($1 \geq I$ y $I \geq 1$), y mejor que algún otro valor. Se escribe $x > y$ si x es un grado de satisfacción estrictamente mejor que el grado y.

Los grados de satisfacción de las reglas de P_{pref} para los *answer sets* son la base para definir los vectores de satisfacción y con ellos determinar el *answer set* que será la solución preferida.

Definición 3.24. [11] Dado $P_{pref} = \{r_1, \ldots, r_n\}$ como un programa de preferencias y S un *answer set*. Se tiene que S induce un vector de satisfacción $V_S = (v_s(r_1), \ldots, v_S(r_n))$. □

Ejemplo 3.15. Sea el ASOP del ejemplo 3.14. Teniendo a P_{pref} formado por las reglas:

$$r_{12}: \quad blanco > rojo > cerveza \leftarrow pescado.$$
$$r_{13}: \quad rojo \vee cerveza > blanco \leftarrow carne.$$
$$r_{14}: \quad pay > helado \leftarrow cerveza.$$

y los *answer sets*:

$$S_1: \quad \{bebida, postre, principal, entrada, helado, cerveza, carne, sopa\}.$$
$$S_2: \quad \{bebida, postre, principal, entrada, helado, cerveza, carne, ensalada\}.$$
$$S_3: \quad \{bebida, postre, principal, entrada, helado, cerveza, pescado, sopa\}.$$
$$S_4: \quad \{bebida, postre, principal, entrada, helado, cerveza, pescado, ensalada\}.$$

Los grados de satisfacción para las reglas de P_{pref} con respecto a los *answer sets* de S_1 a S_4 se definen como:

$$v_{S_1}(r_{12}) = I \quad v_{S_1}(r_{13}) = 1 \quad v_{S_1}(r_{14}) = 2$$
$$v_{S_2}(r_{12}) = I \quad v_{S_2}(r_{13}) = 1 \quad v_{S_2}(r_{14}) = 2$$
$$v_{S_3}(r_{12}) = 3 \quad v_{S_3}(r_{13}) = I \quad v_{S_3}(r_{14}) = 2$$
$$v_{S_4}(r_{12}) = 3 \quad v_{S_4}(r_{13}) = I \quad v_{S_4}(r_{14}) = 2$$

Se tiene que todos los valores de I se obtuvieron de acuerdo al primer caso de la definición 3.23. De los grados de satisfacción se forman los siguientes vectores de satisfacción:

$$V_{S_1}: \quad (I, 1, 2).$$
$$V_{S_2}: \quad (I, 1, 2).$$
$$V_{S_3}: \quad (3, I, 2).$$
$$V_{S_4}: \quad (3, I, 2).$$

□

Cuando se tienen los vectores de satisfacción, ya se puede determinar la solución preferida. La solución preferida se obtiene de comparar los vectores de satisfacción de los *answer sets*.

Definición 3.25. [11] Dados dos *answer sets* S_1 y S_2 de un ASOP P. Se escribe $V_{S_1} \geq V_{S_2}$ si $v_{S_1}(r_i) \geq v_{S_2}(r_i)$, para todas las reglas r_i de P_{pref}, en este caso se tiene que S_1 es mejor o igualmente preferido sobre S_2, ($S_1 \geq S_2$). Se escribe $V_{S_1} > V_{S_2}$ si $V_{S_1} \geq V_{S_2}$ y para alguna r_i de P_{pref} se tiene que $v_{S_1}(r_i) > v_{S_2}(r_i)$, entonces se tiene que S_1 es preferido sobre S_2, ($S_1 > S_2$). □

Si de la comparación de los vectores de satisfacción de un ASOP se obtiene un solo *answer set* preferido sobre el resto, se dice que este *answer set* es un modelo óptimo para ese ASOP.

Definición 3.26. [11] Un conjunto de literales S es un modelo óptimo de un ASOP $P = (P_{gen}, P_{pref})$, si S es un *answer set* de P y no hay otro *answer set* S' de P tal que $S' > S$. □

Ejemplo 3.16. Tomando el ASOP y los *answer sets* del ejemplo 3.14:

$$
\begin{aligned}
S_1 &: \{helado, cerveza, carne, sopa\}.\\
S_2 &: \{helado, cerveza, carne, ensalada\}.\\
S_3 &: \{helado, cerveza, pescado, sopa\}.\\
S_4 &: \{helado, cerveza, pescado, ensalada\}.
\end{aligned}
$$

Para los vectores de satisfacción calculados en el ejemplo 3.15:

$$
\begin{aligned}
V_{S_1} &: (I, 1, 2).\\
V_{S_2} &: (I, 1, 2).\\
V_{S_3} &: (3, I, 2).\\
V_{S_4} &: (3, I, 2).
\end{aligned}
$$

Se tiene que $V_{S_1} = V_{S_2}$ y que $V_{S_3} = V_{S_4}$. Al comparar V_{S_1} y V_{S_3}, para r_7 se tiene que $v_{S_1}(r_7) > v_{S_3}(r_7)$ por lo que se prefiere a S_1. Entonces se tiene que S_1 y S_2 son igualmente buenos y los más preferidos para el problema 3.14. $\square$

3.4. Conclusión

En este capítulo se presentaron tres enfoques que modelan problemas con preferencias, estos enfoques fueron: LPOD, PLP y ASOP. La forma en que fueron presentados ayudó para determinar lo que se presenta a continuación.

Los enfoques de LPOD, PLP y ASOP modelan problemas con preferencias. El modelo del problema debe de estar formado con cierta estructura. Las formas de modelar un problema con preferencias con los enfoques analizados se muestra en la Tabla 3.1.

Enfoque	**Forma de las reglas del programa**	
LPOD	Reglas con disyunción ordenada	$C_1 \times \cdots \times C_n \leftarrow A_1, \ldots, A_m, not\, B_1, \ldots, not\, B_k.$
PLP	Reglas disyuntivas Reglas de preferencia	$f_1 \vee \cdots \vee f_n \leftarrow g.$ $f_1 * \cdots * f_n \overset{pr}{\leftarrow} g.$
ASOP	Programa generador Programa de preferencias	$C \leftarrow a_1, \ldots, a_n, not\, b_1, \ldots, not\, b_m.$ $C_1 > \ldots > C_k \leftarrow a_1, \ldots, a_n, not\, b_1, \ldots, not\, B_m.$

Tabla 3.1: Forma de modelar problemas con preferencias.

Como se observa en la Tabla 3.1, el modelado para el enfoque LPOD sólo tiene un conjunto de reglas. Para los enfoques PLP y ASOP el problema se modela en dos partes: la parte que genera las soluciones y la parte donde se especifican las preferencias.

El modelo de LPOD trabaja con literales instanciadas en el cuerpo y la cabeza de la regla. Aunque se tiene una ampliación denominada ELPOD la cual considera fórmulas proposicionales bien formadas para tener una sintaxis más amplia. El enfoque de PLP trabaja con fórmulas proposicionales bien formadas en el cuerpo y la cabeza de la regla. Para el enfoque de ASOP se definen sus reglas como: el cuerpo formado por literales de átomos y la cabeza formada por combinaciones booleanas. Las combinaciones booleanas son similares a las fórmulas bien formadas.

Modelado el problema en cualquiera de los enfoques de LPOD, PLP o ASOP se obtienen las soluciones candidatas a solución preferida. Para determinar la solución preferida, los tres enfoques se apoyan en grados de satisfacción. Los grados de satisfacción determinan el nivel de preferencia. En la Tabla 3.2 se describen los valores que se asignan a los grados de satisfacción para los enfoques analizados.

	S en la regla		
	satisface el cuerpo y la cabeza		**no satisface el cuerpo**
Enfoque	**no se satisface**	**se satisface en un C_i**	
LPOD	No es posible	$min\{r \mid C_r \in S\}$	1
PLP	n+1	$min\,\{i \mid S \cup \neg(\mathcal{L}_S \setminus S) \vdash_{G_3} f_i\}$	1
ASOP	I	$min\{i : S \models C_i\}$	I

Tabla 3.2: Grados de Preferencia para los enfoques.

Lo que muestra la Tabla 3.2 es el valor del grado de satisfacción asignada a una solución candidata S de acuerdo a una regla de preferencia r para un problema con preferencias P.

Los grados de satisfacción dentro de los enfoques tratan de dar un nivel de preferencia a las soluciones candidatas para poder determinar la solución preferida. La Tabla 3.3 muestra los valores que asignan los enfoques a los niveles de preferencia.

Enfoque	**Valores**		
	Más preferidos	**Intermedios**	**Menos preferidos**
LPOD	1	2, 3, ..., $n-1$	n
PLP	1	2, 3, ..., n	$n+1$
ASOP	1, I	2, 3, ..., $n-1$	n

Tabla 3.3: Niveles de Preferencia.

En la Tabla 3.3 se observa que valores son igualmente preferidos para los enfoques. Los enfoques de PLP y ASOP agregan valores. El enfoque PLP agrega el valor $n+1$, el que aparece cuando la regla de preferencia no considera a la solución candidata. Para el enfoque ASOP se tiene el valor I, el cual modela la irrelevancia.

Ya con los valores de los grados de satisfacción para las soluciones candidatas se debe de determinar la solución preferida. La solución preferida se determina por un criterio de preferencia. En la Tabla 3.4 se describen los criterios de preferencia con los que trabajan los enfoques revisados.

Enfoque	**Criterio**
LPOD	Inclusión
	Cardinalidad
	Pareto
PLP	inclusión
	Cardinalidad
ASOP	Comparación vectorial

Tabla 3.4: Criterios de Preferencia.

La Tabla 3.4 muestra las formas de cómo se determina una solución preferida de acuerdo a los enfoques y a los criterios que utiliza.

En el apéndice B se presenta una colección de ejemplos de problemas con preferencias, los cuales son modelados y resueltos por cada uno de los enfoques revisados en este capítulo. Con los ejemplos del apéndice B, se consideran difenretes caracteristicas que se pueden presentar en estos problemas.

Capítulo 4

Problemas con Preferencias

En este capítulo se propone un modelo para los problemas con preferencias. Una vez que se tiene el modelo para los problemas con preferencias, se propone una clasificación para este tipo de problemas. La clasificación propuesta se tiene a partir de las relaciones, dependencias y restricciones que puedan presentar las preferencias.

4.1. Modelado de Problemas con Preferencias

Los problemas con preferencias que se tienen dentro del mundo real se pueden modelar de muchas formas. El modelo que es considerado en estre trabajo se propone a partir de las caracteristicas que se observarón en este tipo de problemas. El modelo propuesto considera un problema con preferencias formado por dos partes: P_{desc} y P_{pref}.

En P_{desc} se describe la naturaleza del problema, la cual está definida por: condiciones, restricciones, hechos, dependencias, relaciones, opciones. De P_{desc} se obtiene un conjunto de soluciones, denominadas *soluciones candidatas*. La parte P_{pref} especifica las preferencias así como sus relaciones, dependencias y restricciones, entendiendo como preferencia a una lista ordenada de opciones. El orden de la lista indica qué opción es preferida sobre otra. El orden considerado para este trabajo es de izquierda a derecha, donde la opción de la izquierda es preferida sobre las que se encuentran a su derecha. Por lo que P_{pref} toma las soluciones candidatas obtenidas de P_{desc} como opciones y determina que opción es preferida sobre las demás. La Figura 4.1 presenta el modelo de un problema con preferencias. De lo anterior se tiene la definición 4.1 para los problemas con preferencias.

Definición 4.1. Un problema con preferencias se define como un par formado por (P_{desc}, P_{pref}), donde P_{desc} describe la naturaleza del problema y P_{pref} especifica las preferencias del problema.

Figura 4.1: Partes de un Problema con Preferencias.

En el ejemplo 4.1 se ilustra un problema con preferencias, teniendo las dos partes del problema de acuerdo al modelo con el que se está trabajando.

Ejemplo 4.1. Para la elección de un postre, Mari tiene las siguientes opciones: `galletas`, `helado` o `pastel`. Para acompañar el postre, Mari debe elegir una de las siguientes opciones de bebida: `café` o `té`.

Ordenando las opciones del postre de acuerdo a sus preferencias, Mari tiene lo siguiente: *galletas* sobre *helado* sobre `pastel`. Para la bebida prefiere el `té` sobre el `café`. El problema de Mari se modela de la siguiente forma:

P_{desc} : Las opciones para el postre son: *galletas*, *helado* o *pastel*.

La opciones para la bebida son: *té* o *café*.

P_{pref} : Para el postre prefiere: *galletas* `sobre` *helado* `sobre` *pastel*.

Para la bebida prefiere: *té* `sobre` *café*.

□

4.2. Clasificación de Problemas con Preferencias

Como parte de la contribución en este trabajo, se propone dividir a los problemas de preferencias en dos grupos. Cada grupo a su vez se divide en clases. Se ha denominado al primer grupo como problemas de preferencias básicas y el segundo grupo como problemas de preferencias compuestas. La Figura 4.2 muestra los grupos y clases de problemas con preferencias que se proponen.

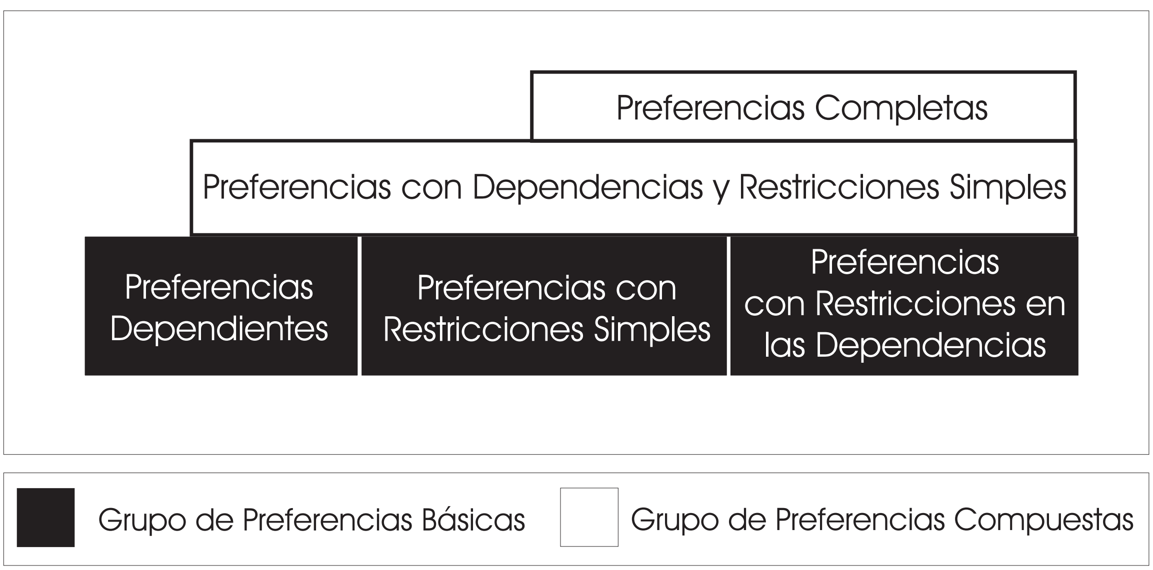

Figura 4.2: Clasificación de Problemas con Preferencias.

La Figura 4.2 indica que las clases de problemas con preferencias están definidos a partir de las clases con un nivel inferior. El grupo de problemas de preferencias compuestas está definido a partir de la combinación de las clases del grupo de problemas de preferencias básicas.

En las siguientes secciones se presenta la definición de las clases de acuerdo a los grupos de problemas de preferencias.

4.2.1. Problemas de Preferencias Básicas

En esta sección se tienen a las clases del grupo de problemas de preferencias básicas para los problemas con preferencias:

- *problemas de preferencias dependientes*,
- *problemas de preferencias con restricciones simples* y
- *problemas de preferencias con restricciones en las dependencias*.

Los problemas de preferencias básicas se consideran así porque no involucran una combinación de clases de problemas con preferencias.

Problemas de Preferencias Dependientes

Esta clase de problemas está definida a partir de relaciones de dependencia que se tienen para las preferencias.

Definición 4.2. Se tiene un problema de *preferencias dependientes* cuando para un problema de preferencias existe al menos una preferencia tiene condiciones.

Los problemas de *preferencias dependientes* por simplicidad pueden ser llamados *problemas pd*. En el ejemplo 4.2 se muestra la parte P_{pref} de un *problema pd*.

Ejemplo 4.2. Retomando la parte P_{pref} del ejemplo 4.1, donde se tienen las siguientes preferencias:

Para el postre prefiere: *galletas* `sobre` *helado* `sobre` *pastel*.

Para la bebida prefiere: *té* `sobre` *café*.

Ahora se agrega una relación de dependencia entre el tipo de `café` que se puede elegir, es decir, que las opciones de: `capuccino, express` o `americano`, dependen de que la opción de `café` sea la elegida como bebida.

Para las opciones que dependen de elegir `café`, Mari prefiere el `capuccino` sobre el `express` sobre el `americano`. Teniendo así la siguiente *preferencia dependiente*:

Si elige *cáfe* prefiere: *capuccino* `sobre` *express* `sobre` *americano*.

De lo anterior se tiene que P_{pref} para el problema queda definido con las siguientes preferencias:

Para el postre prefiere: *galletas* `sobre` *helado* `sobre` *pastel*.

Para la bebida prefiere: *té* `sobre` *café*.

Si elige *café* prefiere: *capuccino* `sobre` *express* `sobre` *americano*.

□

Problemas de Preferencias con Restricciones Simples

Otra clase de problemas de preferencias básicas es la clase de *preferencias con restricciones simples*, la cual agrega restricciones a las preferencias.

Definición 4.3. Se tiene un problema de *preferencias con restricciones simples* cuando para un problema de preferencias, al menos una opción de una preferencia tiene una restricción o más.

Los problemas de *preferencias con restricciones simples* por simplicidad pueden ser llamados *problemas prs*. El ejemplo 4.3 muestra la parte P_{pref} de un *problema prs*.

Ejemplo 4.3. Retomando las preferencias de P_{pref} del ejemplo 4.1 definidas como:

Para el postre prefiere: *galletas* `sobre` *helado* `sobre` *pastel*.

Para la bebida prefiere: *té* `sobre` *café*.

Ahora consideremos que Mari prefiere no elegir `té` y `helado`, porque el sabor de `té` con `helado` no le agrada. Esta restricción involucra una opción de bebida y una opción del postre.

Con esto se tienen una *preferencia con restricciones simples* dada como:

`No` puede tomar *té* con *helado*.

Con lo anterior, el problema de *preferencias con restricciones simples* queda definido en su parte P_{pref} como:

Para el postre prefiere: *galletas* `sobre` *helado* `sobre` *pastel*.

Para la bebida prefiere: *té* `sobre` *café*.

`No` puede tomar *té* con *helado*.

□

Problemas de Preferencias con Restricciones en las Dependencias

La clase de problemas que se define a continuación se forma a partir de las restricciones sobre las opciones de las *preferencias dependientes*.

Definición 4.4. Se tiene un problema de *preferencias con restricciones en las dependencias* cuando para un problema de preferencias, al menos una opción de una *preferencia dependiente* tiene una restricción o más.

Los problemas de *preferencias con restricciones en las dependencias* por simplicidad pueden ser llamados *problemas prd*. Las restricciones de los *problemas prd* se cumplen sólo cuando las opciones de las *preferencias dependientes* para las que están definidas se cumplen. De lo anterior, se tiene que la restricción no sea considerada para algunos casos en el problema. El ejemplo 4.4 muestra a P_{pref} de un *problema prd*.

Ejemplo 4.4. Considerando las preferencias de P_{pref} que se tienen en el problema del ejemplo 4.1 y la siguiente *preferencia dependiente*:

Si elige *helado* prefiere: el sabor *chocolate* `sobre` el sabor *vainilla*.

También se considera que no es posible el postre `helado` de sabor `chocolate` porque no hay suficiente sabor para una porción. Esta es una restricción sobre la opción de sabor `chocolate`, la cual depende de que se elija `helado` como postre. Con lo anterior se tiene la siguiente *preferencia con restricciones en la dependencia*:

Si elige *helado*: `no` puede pedir sabor *chocolate*.

Notesé que la restricción en la dependencia sólo será considerada siempre y cuando el postre elegido sea helado, en otro caso, la restricción no se tomará en cuenta.

De lo anterior, el problema de *preferencias con restricciones en las dependencias* en la parte P_{pref} queda definido como:

Para el postre prefiere: *galletas* `sobre` *helado* `sobre` *pastel*.

Para la bebida prefiere: *té* `sobre` *café*.

Si elige *helado* prefiere: el sabor *chocolate* `sobre` el sabor *vainilla*.

Si elige *helado*: `no` puede pedir sabor *chocolate*.

□

4.2.2. Problemas de Preferencias Compuestas

En esta sección se presentan las clases de problemas del grupo de preferencias compuestas:

- *problemas de preferencias con dependencias y con restricciones simples*,
- *problemas de preferencias completas*.

Las preferencias compuestas se tienen a partir de combinar clases de problemas con preferencias. Las clases de problemas con preferencias que se pueden combinar son las del grupo preferencias básicas. Las combinaciones que se tienen se determinan a partir de la naturaleza de las clases, pues no todas las combinaciones pueden ser posibles. Como resultado se obtiene problemas más grandes de preferencias, en comparación a los problemas de preferencias básicas.

Problemas de Preferencias con Dependencias y con Restricciones Simples

La siguiente clase de preferencias compuestas presenta dos clases básicas: *preferencias dependientes* y *preferencias con restricciones simples*.

Definición 4.5. Se tiene un problema de *preferencias con dependencias y con restricciones simples* cuando para un problema de preferencias hay al menos una *preferencia dependiente* y una o más *preferencias con restricciones simples*.

Los problemas de *preferencias con dependencias y con restricciones simples* por simplicidad pueden ser llamados *problemas pdrs*. Las restricciones presentes en los *problemas pdrs* son independientes de las *preferencias con dependencias*, por lo que las dependencias no son afectadas por las restricciones. El ejemplo 4.5 muestra un *problema pdrs* que ilustra esta clase de preferencias.

Ejemplo 4.5. Considerando P_{pref} que define las *preferencias dependientes* en el ejemplo 4.2, así como la *preferencia con restricciones simples* del ejemplo 4.3, se tiene un problema de *preferencias con dependencias y con restricciones simples* definido en la parte P_{pref} como sigue:

Para el postre prefiere: *galletas* `sobre` *helado* `sobre` *pastel*.

Para la bebida prefiere: *té* `sobre` *café*.

Si elige *café* prefiere: *capuccino* `sobre` *express* `sobre` *americano*.

`No` tomar *té* con *helado*.

Notesé que este problema combina dos clases de problemas con preferencias: *preferencias dependientes* y *preferencias con restricciones simples*. Además que las dependencias no son afectadas por las restricciones del problema. □

Problemas de Preferencias Completas

La última clase de preferencias compuestas considerada se define a partir de combinar tres clases de preferencias básicas.

Definición 4.6. Se tiene un problema de *preferencias completas* cuando para un problema de preferencias hay *preferencias con dependencias y con restricciones simples* y se tiene una o más *preferencias con restricciones en las dependencias*.

Los problemas de *preferencias completas* por simplicidad pueden ser llamados *problemas pc*. El ejemplo 4.6 ilustra un *problema pc*.

Ejemplo 4.6. Considerando a P_{pref} del ejemplo 4.4 y la *preferencia con restricciones simples* del ejemplo 4.3, se tiene un problema con *preferencias completas*, donde P_{pref} está definido por las siguientes preferencias:

Para el postre prefiere: *galletas* `sobre` *helado* `sobre` *pastel*.

Para la bebida prefiere: *té* `sobre` *café*.

Si elige *helado* prefiere: el sabor *chocolate* `sobre` el sabor *vainilla*.

`No` tomar *té* con *helado*.

Si elige *helado*: `no` puede pedir sabor *chocolate*.

El problema incluye las tres clases de preferencias básicas: *preferencias dependientes*, *preferencias con restricciones simples* y *preferencias con restricción en la dependencia*.

□

4.3. Conclusión

En este capítulo se propuso una clasificación para los problemas con preferencias, donde las clases de los problemas con preferencias se dividen en dos grupos. El primer grupo considera clases básicas. El segundo grupo considera combinaciones de las clases básicas, por lo que son denominadas clases compuestas.

La Tabla 4.1 resume la combinación de las clases básicas que forman a las clases de preferencias compuestas. En la Tabla 4.1 se tiene que:

- *prs* identifica a las *preferencias con restricciones simples*,
- *pd* a las *preferencias dependientes* y
- las *preferencias con restricciones en las dependencias* son identificadas como *prd*.

Clase	**pd**	**prs**	**prd**
preferencias con dependencias y con restricciones simples	Si	Si	No
preferencias completas	Si	Si	Si

Tabla 4.1: Clases de problemas con preferencias compuestas.

En el apéndice B se presenta un colección de ejemplos de problemas con preferencias. Los ejemplos se tienen para las clases definidas en este capítulo. Los ejemplos son modelados y resueltos de acuerdo a los enfoques revisados en el capítulo 3. Al final de cada ejemplo se presenta una comparación de los resultados obtenidos con los enfoques, para poder determinar el comportamiento de los enfoques de ASP.

Capítulo 5

Conclusiones y Trabajo a Futuro

En este documento se trabajó con ASP, el cual es un lenguaje declarativo para la representación del conocimiento y el razonamiento del sentido común. Dentro de los problemas que se atacan en el razonamiento del sentido común, se encuentran los problemas con preferencias, y este tipo de problemas son los que se abordaron en este trabajo. Revisamos que los problemas con preferencias se presentan en dos partes: P_{desc} y P_{pref}. En P_{desc} se describe el problema y se obtienen las soluciones candidatas a ser la solución preferida. En P_{pref} se tiene la especificación de las preferencias, que toma las soluciones candidatas y se elige cual es la solución más preferida.

Usando ASP es posible modelar problemas con preferencias y para ello actualmente existen varios enfoques. Los enfoques de ASP para modelar problemas con preferencias revisados en este trabajo son: LPOD, PLP y ASOP. Por lo que parte de la contribución fue presentarlos con una estructura uniforme, lo cual ayudó en la comparación entre ellos. Al comparar los enfoques, se tienen tres puntos a considerar:

1. Forma en que se modela el problema de preferencias.

2. Grados de preferencia.

3. Criterios de preferencia.

Para la estructura del modelado se tiene la Tabla 5.1, que muestra la forma de las reglas, asi como las partes en que se divide.

Enfoque	**Forma de las reglas del programa**	
LPOD	Reglas con disyunción ordenada	$C_1 \times \cdots \times C_n \leftarrow A_1, \ldots, A_m, not\, B_1, \ldots, not\, B_k.$
PLP	Reglas disyuntivas Reglas de preferencia	$f_1 \vee \cdots \vee f_n \leftarrow g.$ $f_1 * \cdots * f_n \overset{pr}{\leftarrow} g.$
ASOP	Programa generador Programa de preferencias	$C \leftarrow a_1, \ldots, a_n, not\, b_1, \ldots, not\, b_m.$ $C_1 > \ldots > C_k \leftarrow a_1, \ldots, a_n, not\, b_1, \ldots, not\, B_m.$

Tabla 5.1: Forma de modelar los de problemas con preferencias.

Para el punto de grados de preferencia se tienen dos aspectos que se consideran: 1) la forma de asignación de los grados de preferencia y 2) los niveles de preferencia que tienen los valores. En la asignación de los grados de preferencia se tienen tres casos, los cuales se muestran en la Tabla 5.2. De acuerdo a los valores de los grados de preferencia

	S en la regla		
	satisface el cuerpo y la cabeza		**no satisface**
Enfoque	**no se satisface**	**se satisface en un C_i**	**el cuerpo**
LPOD	No es posible	$min\{r \mid C_r \in S\}$	1
PLP	n+1	$min\,\{i \mid S \cup \neg(\mathcal{L}_S \setminus S) \vdash_{G_3} f_i\}$	1
ASOP	I	$min\{i : S \models C_i\}$	I

Tabla 5.2: Grados de Preferencia para los enfoques.

se establece el nivel de preferencia, el cual se presenta en la Tabla 5.3, indicando que valor es mejor sobre otro.

Enfoque	**Valores**		
	Más preferidos	**Intermedios**	**Menos preferidos**
LPOD	1	2, 3, ..., $n-1$	n
PLP	1	2, 3, ..., n	$n+1$
ASOP	1, I	2, 3, ..., $n-1$	n

Tabla 5.3: Niveles de Preferencia.

Para el tercer punto en la comparación de los enfoques se tienen los criterios de preferencia. La Tabla 5.4 muestra los criterios con los que están definidos los enfoques.

Enfoque	Criterio
LPOD	Inclusión
	Cardinalidad
	Pareto
PLP	inclusión
	Cardinalidad
ASOP	Comparación vectorial

Tabla 5.4: Criterios de Preferencia.

Después de haber analizado cómo se modelan los problemas con preferencias y tener una comparación para los enfoques de LPOD, PLP y ASOP, se propuso una clasificación para este tipo de problemas. En la clasificación propuesta solo se consideró la parte P_{pref} de los problemas con preferencias, ya que sólo se están considerando las preferencias en este trabajo. Las clases que se están proponiendo están mostradas en la figura 5.1.

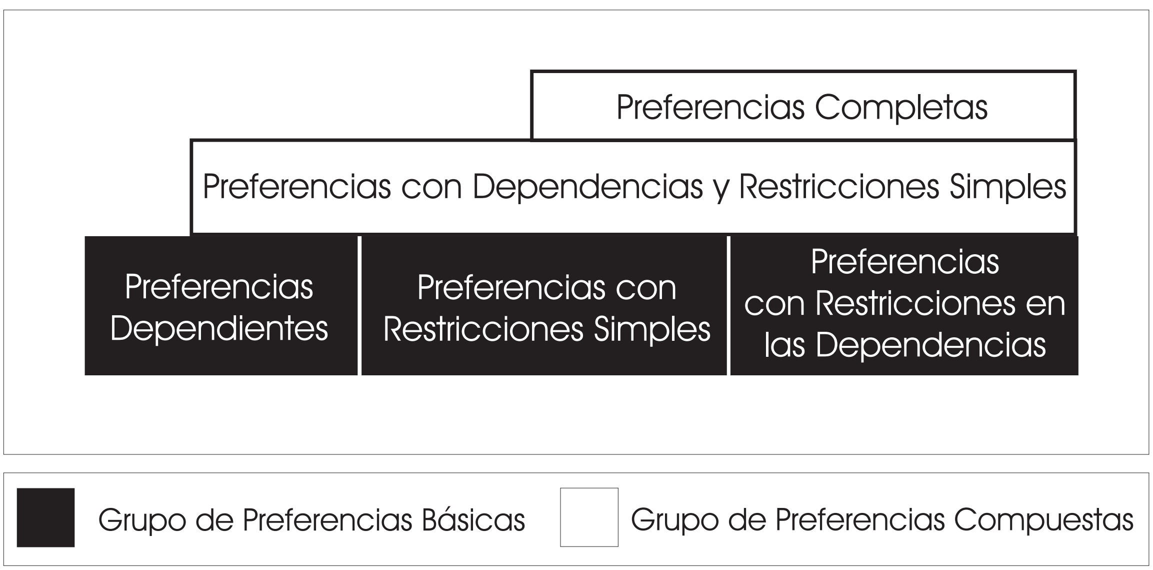

Figura 5.1: Clasificación de Problemas con Preferencias.

Dentro de la clasificación se tienen dos grupos, los cuales consideran cómo están formadas las clases: el grupo de preferencias básicas y el grupo de preferencias compuestas. El grupo de preferencias básicas está formado por las clases: *preferencias dependientes*, *preferencias con restricciones simples* y *preferencias con restricciones en las depen-*

dencias. El grupo de preferencias compuestas está formado por las clases: *preferencias con dependencias y con restricciones simples* y *preferencias completas*. Estas clases se obtienen de las combinaciones posibles de las clases del grupo de preferencias básicas.

Con la clasificación se propuso y resolvio un conjunto de problemas con preferencias, mostrados en el apéndice B. Los problemas se propusieron de tal forma que se consideraron cada una de las clases para los problemas con preferencias. Los problemas se modelaron y resolvieron por cada uno de los enfoques revisados, es decir: LPOD, PLP y ASOP. Al no tener herramientas que encuentran los *answer sets* preferidos para los enfoques de PLP y ASOP, sólo se utilizó *SMODELS* para el cálculo de los *answer sets*. Esto se tiene porque estos enfoques dividen el programa, permitiendo el cálculo de *answer sets* por separado. La solución preferida se obtuvo de forma manual, retomando los pasos descritos para cada enfoque en el capítulo 3. Para LPOD se tiene la herramienta de *PSMODELS*, que es una extensión de *SMODELS* para trabajar con este enfoque. La Tabla 5.5 muestra las herramientas que se tienen para los enfoques de ASP para preferencias.

Enfoque	**Herramienta**
LPOD	PSMODELS
PLP	No existe
ASOP	No existe

Tabla 5.5: Herramientas para los enfoques de ASP para preferencias.

Actualmente no hay herramientas para PLP y ASOP que simplifiquen el trabajo en la cálculo de la solución preferida.

Los resultados obtenidos de los problemas propuestos ayudarón a realizar algunas conjeturas sobre el comportamiento que presentan los enfoques LPOD, PLP y ASOP al modelar y resolver problemas con preferencias. Las conjeturas realizadas tienen que ver con:

- La relación de las soluciones candidatas con las preferencias.
- El orden de las opciones dentro de las preferencias.

Respecto a la relación de las soluciones candidatas con las preferencias, se pueden presentar los siguientes casos:

1. Todas las soluciones candidatas de un problema están relacionadas con las preferencias.
2. Al menos una solución candidata no está relacionada con las preferencias.

Por consecuencia, la solución preferida para cada problema depende de la relación de las soluciones candidatas con las preferencias.

Para el primer caso, se tiene que la solución preferida será la solución candidata que esté formada por las opciones más preferidas. Teniendo la misma solución preferida en cualquiera de los enfoques analizados. Esto se pudo determinar, ya que los ejemplos propuestos están dentro de este conjunto de problemas.

Para el segundo caso, se tienen diferentes comportamientos para los enfoques revisados. Al modelar el problema por el enfoque de LPOD, se tiene que no es posible considerar este caso. Lo anterior se piensa así por la forma en que el enfoque modela el problema, pues sólo se tienen reglas con disyunción ordenada. Para los enfoques de PLP y ASOP, al tener por separado el modelado de las partes del problema con preferencias, se puede presentar al menos una solución candidata que no esté relacionada con las preferencias.

En este conjunto de problemas, al tener una solución candidata que no tenga definidas preferencias, los valores de los grados de satisfacción que le asignan los enfoques de PLP y ASOP tendrían niveles de preferencia opuestos. Para PLP se tendría lo menos preferido, caso contrario para ASOP que determinaría lo más preferido.

Donde el comportamiento exacto de los enfoques LPOD, PLP y ASOP para el conjunto de problemas que generen soluciones candidatas que no tengan relación con las preferencias, queda a consideración para ser retomado en un trabajo a futuro. Esto es porque a pesar del análisis previo a nivel general, no es claro el comportamiento que puede haber dentro de las clases para problemas con preferencias.

Por último, otra conjetura que se tiene es para una característica detectada en los problemas con preferencias. Esta característica se considera como un caso de estudio a futuro, ya que el interés del trabajo fue sobre la clasificación , y al momento de proponer los problemas fue detectada. Esta caracteristica se presenta cuando las opciones de las preferencias son presentadas más de una vez de acuerdo a lo siguiente:

- La forma en que se presentan es con diferente orden de preferencia,
- Las reglas de preferencia son satisfechas por al menos una solución candidata.

Derivado de lo anterior se tiene que las opciones presentes en las reglas de preferencia tienen al menos dos valores para el grado de preferencia. El ejemplo 5.1 ilustra la idea de esta característica de los problemas con preferencias.

Ejemplo 5.1. Supongamos que se tienen las siguientes reglas en la parte de preferencias P_{pref} de un problema con preferencias:

Si se tiene d se prefiere: a `sobre` b `sobre` c.

Si se tiene e se prefiere: b `sobre` c `sobre` a.

y de P_{desc} se han obtenido las siguientes soluciones candidatas:

$$\{a, d, e\}$$
$$\{b, d, e\}$$

donde el caso de estudio trata de determinar el comportamiento en el cálculo de la solución preferida, para cada uno de los enfoques. □

Una vez que se tengan cubiertos los casos especiales propuestos, para determinar el comportamiento de los enfoques de ASP para preferencias, se conocerán las caracteristicas en las que un enfoque puede ser considerado mejor que otro. Con los comportamientos más completos de los enfoques se puede realizar una herramienta que simplifique y/o automatice el trabajo del cálculo de soluciones preferidas.

Apéndice A

LPODs Extendidos

El uso de expresiones jerarquizadas simplifica la tarea de escribir programas lógicos, mejorando su legibilidad ya que permite escribir reglas más cortas y de una manera más natural. Por ejemplo, en [18] se tienen los conectivos paramétricos que son introducidos como una extensión de la semántica de los *answer sets* para usarse en la representación del conocimiento de algunos problemas. Otra alternativa, es el uso de la implicación en el cuerpo de las reglas, lo que proporciona una codificación mas natural y uniforme. En [22] se muestra una aplicación donde la implicación empotrada da una representación natural para un problema real.

En [9], se tienen los programas lógicos con disyunción ordenada, LPODs. Donde las reglas con disyunción ordenada están definidas en términos de literales instanciadas, ahora se definen los programas lógicos con disyunción ordenada extendida, LPEODs (*Logic Programs with Extended Ordered Disjunction*)[23]. Teniendo que las reglas con disyunción ordenada extendida son definidas en términos de fórmulas proposicionales bien formadas. Ya que se considera que una sintaxis más amplia para estas reglas puede dar algunas ventajas en la representación y comprensión de problemas con preferencias.

Definición A.1. [23] Una regla con disyunción ordenada extendida es una fórmula proposicional bien formada como se definió en la Sección 3.1, o una fórmula de la forma:

$$f_1 \times \cdots \times f_n \leftarrow g \tag{A.1}$$

Donde se tiene que $f_1, \ldots, f_n$, g son fórmulas proposicionales bien formadas. Un LPEOD está formado por un conjunto finito de reglas con disyunción ordenada extendida. □

Las fórmulas $f_1 \ldots f_n$ de la ecuación A.1 son las opciones de la regla de un LPEOD. Las opciones forman la cabeza de la regla y la fórmula g es el cuerpo de la regla. La regla quiere decir: si el cuerpo es verdadero y f_1 es posible, entonces f_1; si f_1 no es posible, entonces f_2; $\ldots$; si ninguno de $f_1, \ldots, f_{n-1}$ es posible entonces f_n.

El caso particular donde toda f_i es una literal y g es una conjunción de literales corresponde a los programas con disyunción ordenada según lo presentado por Brewka en [9]. Los LPODs de Brewka utilizan ambos tipos de negación en las reglas: la negación fuerte y la negación por default. Para los LPEODs se considera solamente un tipo de negación, la negación por default, no afectando los resultados dados en [9]. La anterior consideración se tiene porque la negación fuerte se puede simular por un renombramiento de átomos y adición de restricciones [13]. La forma de representar a la negación por default para este enfoque es el símbolo $\neg$.

Otros casos se pueden dar para las reglas de los LPEODs. Si $n = 0$ la regla es una restricción, teniendo a la regla como $\leftarrow g$. Si se tiene que $n = 1$ la regla es para un programa extendido con $f_1 \leftarrow g$. Y si g no aparece en la regla, se tiene un hecho, quedando $f_1 \times \cdots \times f_n$.

A continuación se tiene un ejemplo tomado de [27] que describe un problema que además de que la solución tiene que ver con preferencias, su representación utiliza reglas con disyunción ordenada extendida.

Ejemplo A.1. [27] Una persona debe de viajar de su casa a la escuela durante el `invierno`. Esta persona prefiere viajar en `autobús` y beber `té` mientras viaja sobre transportarse en `bicicleta`. Además, la persona prefiere viajar en `bicicleta` que `caminar` a la escuela. La persona considera que en alguna parte del camino de su casa a la escuela puede haber algún `bloqueo` por la nieve del `invierno` que le impida viajar.

Para el problema anterior se modela el siguiente LPEOD P:

$$r_1 : \quad invierno.$$

$$r_2 : \quad (autobus \wedge te) \times bicicleta \times caminar \leftarrow invierno, \neg bloqueo.$$

Donde r_1 y r_2 son los identificadores para las reglas. Se tiene que la regla r_1 es un hecho. □

Cuando se tiene un problema representado por un LPEOD el objetivo es encontrar la solución preferida de un conjunto de soluciones candidatas. El *answer set* que satisface de la mejor manera a las reglas del LPEOD es la solución preferida. Para determinar el conjunto de soluciones candidatas es necesario descomponer las reglas en sus opciones. Las opciones de las reglas generan un conjunto de programas llamados *split programs*. Los *split programs* tienen *answer sets* que los satisfacen, que son las soluciones candidatas a ser la solución preferida del problema. Para determinar la solución preferida del conjunto de *answer sets* candidatos se calcula el grado de satisfacción para las reglas del programa original de acuerdo a las soluciones candidatas. Con el grado de satisfacción de las reglas y un criterio de elección se determina la solución preferida.

A continuación se presenta la semántica de los LPEODs para encontrar la solución preferida. Esta semántica se apoya de las definiciones presentadas en [9, 10], ya que es un caso especial de este enfoque. La diferencia relevante está en el grado de satisfacción. Ya que el grado de satisfacción para los LPEODs es una generalización de la definición de Brewka para los LPODs.

Como se menciona antes, cada regla de la forma A.1 con $n \geq 1$, tiene n opciones. Algunas de las opciones pueden pertenecer a los *answer sets* candidatos del LPEOD.

Para encontrar los *answer sets* candidatos es necesario tener reglas sin disyunción ordenada. Para que las reglas no tengan disyunción ordenada se tienen que descomponer en sus opciones.

Definición A.2. [23] Sea r la regla con disyunción ordenada dada como $f_1 \times \cdots \times f_n \leftarrow g$. La k-ésima opción de r, para $1 \leq k \leq n$, es definida por:

$$r^k = f_k \leftarrow g, \neg f_1, \ldots, \neg f_{k-1}.$$

□

Ejemplo A.2. Para el LPEOD P del ejemplo A.1. La regla r_1 es un hecho que tiene una opción, por lo que se tiene $r_1^1 = invierno$. Para la regla r_2 dada como $(autobus \wedge te) \times bicicleta \times caminar \leftarrow invierno, \neg bloqueo$ las opciones quedan :

$$r_2^1: \quad (autobus \wedge te) \leftarrow invierno, \neg bloqueo.$$
$$r_2^2: \quad bicicleta \leftarrow invierno, \neg bloqueo, \neg(autobus \wedge te).$$
$$r_2^3: \quad caminar \leftarrow invierno, \neg bloqueo, \neg(autobus \wedge te), \neg bicicleta.$$

□

Al obtener las opciones de las reglas de un LPEOD, se definen programas sin disyunción ordenada. Los programas libres de disyunción ordenada se obtienen de sustituir cada regla por una de sus opciones. Estos programas son llamados *split programs*.

Definición A.3. [23] Sea P un LPEOD. P' es un *split program* de P si se reemplaza cada regla r de P por una de sus opciones. □

El número de *split programs* de un LPEOD depende de las reglas con disyunción ordenada y las opciones que tienen, esto es por ser formados de la sustitución de estas.

Ejemplo A.3. Retomando el LPEOD del ejemplo A.1 y las opciones de su reglas que se muestran en el ejemplo A.2 se tienen 3 *split programs*:

$$
\begin{aligned}
P_1' :\ & invierno.\\
& (autobus \wedge te) \leftarrow invierno, \neg bloqueo.\\
P_2' :\ & invierno\\
& bicicleta \leftarrow invierno, \neg bloqueo, \neg(autobus \wedge te).\\
P_3' :\ & invierno.\\
& caminar \leftarrow invierno, \neg bloqueo, \neg(autobus \wedge te), \neg bicicleta.
\end{aligned}
$$

□

Para cada *split program* de un LPEOD, se obtienen *answer sets* que lo satisfacen. Los *answer sets* de los *split programs* son soluciones candidatas del LPEOD, esto se debe a que han sido generados de ese programa.

Definición A.4. [23] Sea P un LPEOD. Se tiene que M es un *answer set* de P, si M es un *answer set* de un *split program* de P. □

Ejemplo A.4. Sea el LPEOD P del ejemplo A.1 y sus *split programs* calculados en el ejemplo A.3. Para el *split program* P_1' el *answer set* que se obtiene es $M_1 = \{invierno, (autobus \wedge te)\}$. El *split program* P_2' tiene al conjunto $M_2 = \{invierno, bicicleta\}$ como su *answer set*. El *answer set* $M_3 = \{invierno, caminar\}$ se obtiene del *split program* P_3'.

Entonces se tiene que las soluciones candidatas al programa del ejemplo A.1, son los *answer sets* M_1, M_2, M_3. □

Un *answer set* candidato a solución preferida en un LPEOD, satisface las reglas que lo forman. La manera en que una solución candidata satisface a las reglas es denominada como grado de satisfacción. El valor del grado de satisfacción se determina a partir de cómo un *answer set* satisface a la regla.

Definición A.5. [23] Sea M un *answer set* de un LPEOD P, y dada $r = f_1 \times \cdots \times f_n \leftarrow g$ una regla de P. Se define el grado de satisfacción de r con respecto a M, denotado por $deg_M(r)$, como:

1. si $M \cup \neg(\mathcal{L}_P \setminus M) \nvdash_{\mathrm{I}} g$, entonces $deg_M(r) = 1$.

2. si $M \cup \neg(\mathcal{L}_P \setminus M) \vdash_{\mathrm{I}} g$ entonces $deg_M(r) = min_{1 \leq i \leq n} \{i \mid M \cup \neg(\mathcal{L}_P \setminus M) \vdash_{\mathrm{I}} f_i\}$.

□

Los grados de satisfacción se pueden ver como penalizaciones, donde el valor mayor es el menos preferido. Si el cuerpo de la regla no se satisface, no es razón para que la regla sea insatisfecha por lo que el mayor grado posible es asignado. Para el segundo caso, la regla se satisface y el valor del grado de satisfacción toma el valor de la f_i más preferida.

Los grados de satisfacción de las reglas de un LPEOD P, son la base para definir el criterio de preferencia sobre el conjunto de soluciones candidatas de P. El criterio de preferencia se puede definir a partir del grado de satisfacción o de los conjuntos de reglas que tienen el mismo grado de satisfacción. Los conjuntos de reglas de P que satisfacen al *answer set* M con grado j se denotan como $S^j_M(P)$.

Ejemplo A.5. Sea el LPEOD P del ejemplo A.1 y los *answer sets* del ejemplo A.4 dados como: $M_1 = \{invierno, (autobus \wedge te)\}$, $M_2 = \{invierno, bicicleta\}$ y $M_3 = \{invierno, caminar\}$. Los grados de satisfacción para las reglas de P de acuerdo a

M_1, M_2, M_3 son:

$$deg_{M_1}(r_1) = 1 \qquad deg_{M_1}(r_2) = 1$$

$$deg_{M_2}(r_1) = 1 \qquad deg_{M_2}(r_2) = 2$$

$$deg_{M_3}(r_1) = 1 \qquad deg_{M_3}(r_2) = 3$$

Para obtener los valores del grado de satisfacción se considera el segundo caso de la definición A.5. Agrupando las reglas de acuerdo al grado de satisfacción que tienen, se forman los conjuntos:

$$\begin{array}{lll} S^1_{M_1} = \{r_1, r_2\} & S^2_{M_1} = \{\} & S^3_{M_1} = \{\} \\ S^1_{M_2} = \{r_1\} & S^2_{M_2} = \{r_2\} & S^3_{M_2} = \{\} \\ S^1_{M_3} = \{r_1\} & S^2_{M_3} = \{\} & S^3_{M_3} = \{r_2\} \end{array}$$

□

Con la ayuda de los grados de satisfacción para las reglas de un LPEOD se determina el criterio de preferencia que hay entre los *answer sets* que se tienen como soluciones candidatas. En la siguiente sección se definen los criterios con los que trabajan los LPEODs para determinar la solución preferida de un problema.

A.1. Criterios de Preferencia

El enfoque de LPEOD, trabaja con dos criterios para determinar la solución preferida: preferencia por inclusión de conjuntos y preferencia por cardinalidad de conjuntos. Estos criterios ayudan a determinar la solución preferida de acuerdo a los conjuntos de reglas que tienen cierto grado de satisfacción para los *answer sets* que se tienen como soluciones candidatas al problema.

Preferencia por Inclusión

La solución preferida por inclusión de conjuntos elige entre los conjuntos de reglas de dos *answer sets* M_1 y M_2 que son candidatos a solución preferida de un LPEOD P. El *answer set* preferido se tiene de comparar dos conjuntos $S^k_{M_1}(P)$ y $S^k_{M_2}(P)$ de acuerdo a la relación de si un conjunto es subconjunto del otro o viceversa. El *answer set* preferido es el del conjunto que contiene al otro.

Definición A.6. [23] Sean M_1 y M_2 dos *answer sets* de un LPEOD P. M_1 es el conjunto preferido por inclusión sobre M_2, $(M_1 >_i M_2)$, si y solo si, hay un grado k tal que $S^k_{M_2}(P) \subset S^k_{M_1}(P)$, y para toda $j < k$, $S^j_{M_1}(P) = S^j_{M_2}(P)$. □

Si para dos *answer sets* no es posible lo establecido en la definición A.6, se tiene que ambos conjuntos son igualmente preferidos. Cuando se elige a un solo *answer set* del conjunto de soluciones candidatas de un LPEOD, se dice que ese conjunto es un *answer set generalizado preferido por inclusión.*

Definición A.7. [23] Un conjunto de literales M es un *answer set generalizado preferido por inclusión* de un LPOD P, si y solo si M es un *answer set* de P y no hay otro *answer set* M' de P, tal que $M' >_i M$. □

Ejemplo A.6. Considerando el LPOD P del ejemplo A.1, los *answer sets* $M_1 = \{invierno, (autobus \wedge te)\}$,$M_2 = \{invierno, bicicleta\}$ y $M_3 = \{invierno, caminar\}$ calculados en el ejemplo A.4 y los conjuntos de las reglas:

$$\begin{array}{lclcl}
S^1_{M_1} = \{r_1, r_2\} & ; & S^2_{M_1} = \{\} & ; & S^3_{M_1} = \{\} \\
S^1_{M_2} = \{r_1\} & ; & S^2_{M_2} = \{r_2\} & ; & S^3_{M_2} = \{\} \\
S^1_{M_3} = \{r_1\} & ; & S^2_{M_3} = \{\} & ; & S^3_{M_3} = \{r_2\}
\end{array}$$

calculados en el ejemplo A.5. La solución preferida por inclusión de conjuntos entre los *answer sets* candidatos está dada como se describe a continuación.

Al comparar los conjuntos de reglas $S^1_{M_1}$ y $S^1_{M_2}$ para los *answer sets* M_1 y M_2 se tiene que $S^1_{M_2} \subset S^1_{M_1}$. Por lo que se tiene que M_1 es preferido por inclusión sobre M_2.

Para la comparación entre los conjuntos de reglas de los *answer sets* M_1 y M_3, se tiene que $S^1_{M_3} \subset S^1_{M_1}$, por lo que M_1 es preferido por inclusión sobre M_3.

Para las soluciones candidatas M_2 y M_3, se tiene que M_2 es preferido por inclusión sobre M_3, porque se tiene que $S^2_{M_3} \subset S^2_{M_2}$, y para el grado 1 son los mismos conjuntos.

Por lo tanto, la solución preferida para el problema 3.1 es el *answer set* $M_1 = \{invierno, (autobus \wedge te)\}$. Además, se tiene que M_1 es un *answer set* generalizado preferido por inclusión del problema. $\square$

Preferencia por Cardinalidad

Otro criterio que se tiene para obtener la solución preferida es el criterio por cardinalidad de conjuntos. Este criterio compara la cantidad de reglas en los conjuntos para los grados de satisfacción. El *answer set* que satisface más reglas con cierto grado es preferido sobre el otro.

Definición A.8. [23] Sean M_1 y M_2 dos *answer sets* candidatos de un LPEOD P. Se dice que M_1 es preferido por cardinalidad a M_2, $(M >_c N)$, si hay un grado k tal que $\left|S^k_{M_1}(P)\right| > \left|S^k_{M_2}(P)\right|$ y para toda $j < k$ se tiene que $\left|S^j_{M_1}(P)\right| = \left|S^j_{M_2}(P)\right|$. $\square$

Si lo que se presenta en la definición A.8 no determina una solución preferida, se dice que los dos *answer sets* son igualmente preferidos. Si se tiene a un solo *answer set* preferido dentro del conjunto de soluciones candidatas, se dice que es *answer set generalizado preferido por cardinalidad*

Definición A.9. [23] Un conjunto de literales M es un *answer set preferido por cardinalidad* de un LPEOD P, si y solo si M es un *answer set* de P y no hay otro *answer set* M' de P, tal que $M' >_c M$. $\square$

Ejemplo A.7. Para el LPEOD P del ejemplo A.1, considere los *answer sets* : $M_1 = \{invierno, (autobus \wedge te)\}$, $M_2 = \{invierno, bicicleta\}$ y $M_3 = \{invierno, caminar\}$ calculados en el ejemplo A.4 y los conjuntos de las reglas:

$$\begin{array}{lll} S^1_{M_1} = \{r_1, r_2\} & S^2_{M_1} = \{\} & S^3_{M_1} = \{\} \\ S^1_{M_2} = \{r_1\} & S^2_{M_2} = \{r_2\} & S^3_{M_2} = \{\} \\ S^1_{M_3} = \{r_1\} & S^2_{M_3} = \{\} & S^3_{M_3} = \{r_2\} \end{array}$$

calculados en el ejemplo A.5. La solución preferida por cardinalidad de conjuntos entre los *answer sets* candidatos está dada como se describe a continuación.

Al comparar los valores $|S^1_{M_1}|$ y $|S^1_{M_2}|$ de las soluciones candidatas M_1 y M_2, se obtiene que M_1 es preferido por cardinalidad sobre M_2.

Para las soluciones candidatas M_2 y M_3, se que tiene que $|S^2_{M_2}| > |S^2_{M_3}|$ y para el grado inferior son iguales, por lo que M_2 es preferido por cardinalidad sobre M_3.

Para la cardinalidad de los *answer sets* candidatos M_1 y M_3, se tiene que $|S^1_{M_1}| > |S^1_{M_3}|$, estableciendo que M_1 es preferido por cardinalidad sobre M_3.

Por lo tanto, la solución preferida por cardinalidad al problema del ejemplo 3.1 es el conjunto $M_1 = \{invierno, (autobus \wedge te)\}$. Teniendo que M_1 es un *answer set preferido por cardinalidad* del problema. $\square$

Apéndice B

Ejemplos

Este apéndice contiene los problemas ejemplo propuestos para las clases de problemas con preferencias, mostradas en la sección 4.2. Los problemas se modelaron y resolvieron por los enfoques presentados en el capítulo 3.

La forma en que se presentan los problemas ejemplo es de acuerdo al procedimiento mostrado en la figura 1.1 en el capítulo 1. El procedimiento es el siguiente: primero se presenta la descripción de problema, la descripción del problema nos ayuda a modelar dicho problema y de acuerdo a alguno de los enfoques, poder obtener la solución preferida.

En la segunda parte se tiene el modelo de un problema de preferencias a nivel general. De las soluciones candidatas obtenidas se obtiene la solución preferida del problema por medio del sentido común. Con esta solución se tiene un punto de comparación para las soluciones que se obtienen de los enfoques. El punto de comparación se tiene porque la solución por medio del sentido común es lo que se puede esperar como la solución del problema, y los enfoques tratan de representar el funcionamiento del razonamiento del sentido común.

Después se presenta la solución al problema por medio de los enfoques vistos previamente, LPOD (Sección 3.1), PLP (Sección 3.2), ASOP (sección 3.3). El procedimiento de los problemas ejemplo se presentan en forma resumida, tomando en cuenta que los pasos para encontrar la solución no cambian. Para los enfoques PLP (Sección 3.2) y ASOP (sección 3.3) se utilizan las soluciones candidatas obtenidas de la parte descriptiva del problema. Estos *answer sets* son los mismos que se obtienen por medio de *SMODELS*. Para el enfoque LPOD (Sección 3.1), no es posible considerar las soluciones candidatas obtenidas con *SMODELS*, porque el procedimiento para obtenerlas está definido específicamente para el enfoque.

Para los criterios de preferencia en los enfoques, se comparan dos *answer sets*. La notación que se presenta en los problemas ejemplo para esa parte presenta una tabla parecida a la Tabla B.1. Donde la Tabla B.1 representa el resultado de comparar un

	M_1	M_2	M_3
M_1	$=$	$>$	$>$
M_2	$<$	$=$	$=$
M_3	$<$	$=$	$=$

Tabla B.1: Comparación de *answer sets*.

answer set de un renglón con cada *answer set* de una columna. Los operadores utilizados son:

1. El operador *mayor que (>)*, el cual indica que el *answer set* del renglón es preferido sobre el *answer set* de la columna.

2. El operador *menor que (<)*, indicando que el *answer set* de la columna es preferido sobre el *answer set* del renglón.

3. El operador *igual (=)*, que indica que los *answer sets* del renglón y la columna son igualmente preferidos.

El operador *mayor que* se considera opuesto al operador *menor que* y viceversa. El uso del operador *igual* se tiene para dos casos diferentes: el primero se da porque al realizar la comparación entre los *answer sets* los puntos a comparar son iguales, este caso no solo se presenta cuando el *answer set* es el mismo (renglón y columna iguales), pues hay otros casos en que también sucede esto. El segundo caso se presenta cuando no hay nivel de comparación entre los *answer sets*, es decir, de acuerdo a las definiciones no se puede determinar que *answer set* es preferido.

La tabla que se utiliza para la comparación, es una matriz cuadrada, ya que se realiza la comparación de un *answer set* con el conjunto de *answer sets* candidatos, incluyendo el que se está tomando en consideración.

En la Tabla B.1 se tiene una diagonal de operadores *igual*, esto porque los *answer sets* que se comparan son el mismo. También se observa que los resultados de la comparación entre dos *answer sets* se tiene un tipo de matriz triangular superior, pero en vez de tener elementos cero en la parte baja a la diagonal, se tienen los signos opuestos de su elemento correspondiente. Para un elemento (i, j) que está debajo de la diagonal de la matriz, su elemento correspondiente es el elemento (j, i). Lo que se tiene para la Tabla B.1, se aplica para todas las tablas que describen la comparación de *answer sets* en este documento.

B.1. Problemas con Preferencias Dependientes

En las siguientes secciones se presenta un problema con *preferencias dependientes*, (*problema pd*). Se presenta la descripción del problema y a partir de ello, se modela y resuelve de acuerdo a los enfoques analizados. Por último se presenta un concentrado de resultados para comparar lo obtenido de los enfoques.

B.1.1. Descripción del Problema

La descripción del problema con *preferencias dependientes* con el que se trabajará se tiene en el problema B.1.

Problema B.1. Miguel ha decidido aprender a tocar un instrumento musical. Puede escoger entre la familia de `cuerdas` o la familia de los `metales`. Dentro de la familia de las `cuerdas` prefiere la `guitarra` sobre la `mandolina`. Por parte de la familia de los `metales` prefiere el `trombón` sobre la `trompeta`. Miguel dice que los instrumentos de la familia de las `cuerdas` son muy comúnes por lo que prefiere los instrumentos de la familia de los `metales`.

De la descripción en el problema B.1 se tiene la información para modelar el problema. La información se relaciona con identificadores de acuerdo a lo siguiente:

metales:	representa a la familia de metales.
cuerdas:	representa a la familia de cuerdas.
guitarra:	representa al instrumento guitarra.
mandolina:	representa al instrumento mandolina.
trompeta:	representa al instrumento trompeta.
trombon:	representa al instrumento trombón.

Los identificadores ayudarán más adelante dentro de los enfoques, para que las reglas queden en forma sintetizada, siendo fáciles de comprender.

B.1.2. Modelado del Problema

Considerando la descripción para el problema B.1 de Miguel y de acuerdo a la definición 4.1 se tiene el siguiente modelo:

P_{desc} : Las opciones de familias de instrumentos son: *metales* o *cuerdas*.

Las opciones para la familia de *metales* son: *trombon* o *trompeta*.

Las opciones para la familia de *cuerdas* son: *guitarra* o *mandolina*.

P_{pref} : De las familias de instrumentos prefiere: *metales* `sobre` *cuerdas*.

Si elige la familia de *metales* prefiere: *trombon* `sobre` *trompeta*.

Si elige la familia de *cuerdas* prefiere: *guitarra* `sobre` *mandolina*.

B.1.3. Solución por el Sentido Común

Con el modelo de la sección B.1.2 para el problema B.1 de Miguel se tienen las siguientes soluciones:

1. {*metales, trombon*}.

2. {*metales, trompeta*}.

3. {*cuerdas, guitarra*}

4. {*cuerdas, mandolina*}

Al considerar las preferencias de Miguel, se tienen las siguientes soluciones como preferidas:

- {*metales, trombon*}. Esta solución se toma como preferida debido a que la familia de *metales*, como el instrumento *trombon* son las opciones más preferidas para Miguel.

- {*metales, trompeta*}. Esta solución es preferida debido a que la familia de *metales* es preferida sobre la otra familia de instrumentos que se está considerando.

- {*cuerdas, guitarra*}. Esta solución es preferida porque el instrumento *guitarra* es preferido sobre la *mandolina* en la familia de *cuerdas*.

B.1.4. Solución por LPODs

Del modelo presentado en la sección B.1.2 para el problema B.1 y de acuerdo a la definición 3.1 se obtienen las siguientes reglas que modelan el problema como un LPOD:

$$\begin{aligned} r_1: &\quad metales \times cuerdas. \\ r_2: &\quad guitarra \times mandolina \leftarrow cuerdas. \\ r_3: &\quad trombon \times trompeta \leftarrow metales. \end{aligned}$$

Donde las r_i son identificadores de las reglas. Las reglas r_2 y r_3 son dependientes de las opciones que se tienen en la regla r_1.

De acuerdo a la definición 3.4 del enfoque de los LPODs, las soluciones de los *split programs* forman el conjunto de soluciones candidatas del problema modelado en LPOD. Las soluciones candidatas del ejemplo son:

$$\begin{aligned} M_1: &\quad \{metales, trombon\}. \\ M_2: &\quad \{cuerdas, guitarra\}. \\ M_3: &\quad \{metales, trompeta\}. \\ M_4: &\quad \{cuerdas, mandolina\}. \end{aligned}$$

Donde los M_i son los identificadores de los *answer sets* para el conjunto de soluciones candidatas. De los *answer sets* se busca la solución preferida. La solución preferida se obtiene de la comparación de los grados de satisfacción. Los grados de satisfacción de las reglas para los *answer sets*, de acuerdo a la definición 3.5, están dados como:

$$\begin{array}{lll} deg_{M_1}(r_1) = 1 & deg_{M_1}(r_2) = 1 & deg_{M_1}(r_3) = 1 \\ deg_{M_2}(r_1) = 2 & deg_{M_2}(r_2) = 1 & deg_{M_2}(r_3) = 1 \\ deg_{M_3}(r_1) = 1 & deg_{M_3}(r_2) = 1 & deg_{M_3}(r_3) = 2 \\ deg_{M_4}(r_1) = 2 & deg_{M_4}(r_2) = 2 & deg_{M_4}(r_3) = 1 \end{array}$$

Al agrupar las reglas que tienen el mismo grado de satisfacción para los answer sets, los conjuntos están formados como:

$$
\begin{array}{ll}
S^1_{M_1} = \{r_1,\, r_2\ r_3\} & S^2_{M_1} = \{\ \} \\
S^1_{M_2} = \{r_2,\, r_3\} & S^2_{M_2} = \{r_1\} \\
S^1_{M_3} = \{r_1,\, r_2\} & S^2_{M_3} = \{r_3\} \\
S^1_{M_4} = \{r_3\} & S^2_{M_4} = \{r_1,\, r_2\}
\end{array}
$$

Con los valores anteriores se determina la solución preferida a partir de un criterio de preferencia. A continuación se obtiene la solución preferida por los criterios con los que trabajan los LPODs.

Preferencia por Inclusión

De acuerdo al criterio de preferencia por inclusión dado en la definición 3.6 el resultado de comparar los *answer set*s del ejemplo se muestra en la Tabla B.2.

	M_1	M_2	M_3	M_4
M_1	$=$	$>$	$>$	$>$
M_2	$<$	$=$	$=$	$>$
M_3	$<$	$=$	$=$	$=$
M_4	$<$	$<$	$=$	$=$

Tabla B.2: *Problema pd*: LPOD Preferencia por Inclusion

De la información de la Tabla B.2 y por la definición 3.7 se tiene que M_1 es un *answer set* preferido por inclusión, siendo este la solución preferida para el problema.

Preferencia por Cardinalidad

De acuerdo al criterio de preferencia por cardinalidad dado en la definición 3.8 el resultado de comparar los *answer sets* del ejemplo se muestra en la Tabla B.3. De la información de la Tabla B.3 y por la definición 3.9 se tiene que M_1 es un *answer set* preferido por cardinalidad, siendo este la solución preferida para el problema.

	M_1	M_2	M_3	M_4
M_1	$=$	$>$	$>$	$>$
M_2	$<$	$=$	$=$	$>$
M_3	$<$	$=$	$=$	$>$
M_4	$<$	$<$	$<$	$=$

Tabla B.3: *Problema pd*: LPOD Preferencia por Cardinalidad

Preferencia por Pareto

De acuerdo al criterio de preferencia por Pareto dado en la definición 3.10 el resultado de comparar los *answer sets* del ejemplo se muestra en la Tabla B.4.

	M_1	M_2	M_3	M_4
M_1	$=$	$>$	$>$	$>$
M_2	$<$	$=$	$=$	$>$
M_3	$<$	$=$	$=$	$=$
M_4	$<$	$<$	$=$	$=$

Tabla B.4: *Problema pd*: LPOD Preferencia por Pareto

De la información de la Tabla B.4 y por la definición 3.11 se tiene que M_1 es un *answer set* preferido por Pareto, siendo este la solución preferida para el problema.

B.1.5. Solución por PLPs

Del modelo presentado en la sección B.1.2 para el problema B.1 y de acuerdo a la definición 3.12 el problema de Miguel se puede modelar como un PLP formado por las siguientes reglas:

$$\begin{aligned}
r_1 :&\quad metales \vee cuerdas \leftarrow .\\
r_2 :&\quad guitarra \vee mandolina \leftarrow cuerdas.\\
r_3 :&\quad trombon \vee trompeta \leftarrow metales.\\
r_4 :&\quad metales * cuerdas \overset{pr}{\leftarrow} .\\
r_5 :&\quad guitarra * mandolina \overset{pr}{\leftarrow} cuerdas.\\
r_6 :&\quad trombon * trompeta \overset{pr}{\leftarrow} metales.
\end{aligned}$$

Se tiene que los r_i son los identificadores para las reglas. El programa lógico está compuesto por las reglas de r_1 a r_3 y $Pref_P$ por el conjunto $\{r_4, r_5,\ r_6\}$.

Con las reglas del problema modelado en PLP, se busca la solución preferida. Para obtenerla se debe de resolver la parte lógica del problema para encontrar las soluciones candidatas. Las soluciones candidatas son determinadas por lo descrito en la definición 3.13.

Entonces considerando la parte lógica para el PLP. De las reglas r_1 a r_3 se obtienen las siguientes soluciones candidatas:

$$\begin{aligned}
M_1 :&\quad \{metales,\ trombon\}.\\
M_2 :&\quad \{metales,\ trompeta\}.\\
M_3 :&\quad \{cuerdas,\ guitarra\}.\\
M_4 :&\quad \{cuerdas,\ mandolina\}.
\end{aligned}$$

Teniendo a los M_i como los identificadores de los *answer sets* del conjunto de soluciones candidatas. De las soluciones candidatas se busca la solución preferida. La solución preferida se tiene de comparar los grados de satisfacción. Los grados de satisfacción de las reglas para los *answer sets*, de acuerdo a la definición 3.14, se tienen definidos como:

$$
\begin{array}{lll}
deg_{M_1}(r_4) = 1 & deg_{M_1}(r_5) = 1 & deg_{M_1}(r_6) = 1 \\
deg_{M_2}(r_4) = 1 & deg_{M_2}(r_5) = 1 & deg_{M_2}(r_6) = 2 \\
deg_{M_3}(r_4) = 2 & deg_{M_3}(r_5) = 1 & deg_{M_3}(r_6) = 1 \\
deg_{M_4}(r_4) = 2 & deg_{M_4}(r_5) = 2 & deg_{M_4}(r_6) = 1
\end{array}
$$

Con el grado de satisfacción de cada regla de preferencia se tiene un conjunto de reglas de preferencia que tienen el mismo grado. Estos conjuntos están formados como:

$$
\begin{array}{ll}
S^1_{M_1} = \{r_4,\, r_5\ r_6\} & S^2_{M_1} = \{\ \} \\
S^1_{M_2} = \{r_4,\, r_5\} & S^2_{M_2} = \{r_6\} \\
S^1_{M_3} = \{r_5,\, r_6\} & S^2_{M_3} = \{r_4\} \\
S^1_{M_4} = \{r_6\} & S^2_{M_4} = \{r_4,\, r_5\}
\end{array}
$$

Con los conjuntos anteriores se encuentra el *answer set* preferido. El *answer set* preferido se determina a partir de usar un criterio de preferencia. A continuación se obtiene la solución preferida por medio de los criterios con los que trabajan los PLPs.

Preferencia por Inclusión

De acuerdo a lo que se tiene en la definición 3.15 para el criterio de preferencia por inclusión se tiene la Tabla B.5 con los resultados de comparación. De la Tabla B.5 y

	M_1	M_2	M_3	M_4
M_1	$=$	$>$	$>$	$>$
M_2	$<$	$=$	$=$	$=$
M_3	$<$	$=$	$=$	$>$
M_4	$<$	$=$	$<$	$=$

Tabla B.5: *Problema pd*: PLP Preferencia por Inclusión.

por la definición 3.16 se tiene que M_1 es un *answer set* preferido por inclusión, siendo la solución preferida para el problema.

Preferencia por Cardinalidad

De acuerdo a lo que se tiene en la definición 3.17 para el criterio de preferencia por cardinalidad, se tienen los siguientes resultados de comparación, mostrados en la Tabla B.6. De la información de Tabla B.6 y por la definición 3.18 se tiene que M_1 es un

	M_1	M_2	M_3	M_4
M_1	$=$	$>$	$>$	$>$
M_2	$<$	$=$	$=$	$>$
M_3	$<$	$=$	$=$	$>$
M_4	$<$	$<$	$<$	$=$

Tabla B.6: *Problema pd*: PLP Preferencia por Cardinalidad.

answer set preferido por cardinalidad, siendo la solución preferida para el problema.

B.1.6. Solución por ASOPs

Para el problema B.1 modelado en la sección B.1.2 y de acuerdo a las definiciones 3.19 y 3.22 se tienen las siguientes reglas que modelan el problema como un ASOP:

$$
\begin{aligned}
r_1 &: \; metales \vee cuerdas \leftarrow . \\
r_2 &: \; guitarra \vee mandolina \leftarrow cuerdas. \\
r_3 &: \; trombon \vee trompeta \leftarrow metales. \\
r_4 &: \; metales * cuerdas \overset{pr}{\leftarrow} . \\
r_5 &: \; guitarra * mandolina \overset{pr}{\leftarrow} cuerdas. \\
r_6 &: \; trombon * trompeta \overset{pr}{\leftarrow} metales.
\end{aligned}
$$

Teniendo que los r_i son los identificadores para las reglas. P_{gen} está formado por las reglas de r_1 a r_3 y P_{pref} por el conjunto $\{r_4, r_5, r_6\}$.

Con el modelado de un problema como ASOP se busca tener una solución preferida sobre un conjunto de soluciones candidatas. El conjunto de soluciones candidatas se forma por los *answer sets* generados de P_{gen}.

A continuación se tienen las soluciones generadas de P_{gen}:

$$\begin{aligned} S_1 &: \{metales, \; trombon\}. \\ S_2 &: \{metales, \; trompeta\}. \\ S_3 &: \{cuerdas, \; guitarra\}. \\ S_4 &: \{cuerdas, \; mandolina\}. \end{aligned}$$

Teniendo que los S_i son los identificadores de los *answer sets*. Con los *answer sets* obtenidos se determina el grado de satisfacción. De acuerdo a la definición 3.23 se tiene que para los *answer sets*, el grado de satisfacción está dado como:

$$\begin{array}{lll} v_{S_1}(r_4) = 1 & v_{S_1}(r_5) = I & v_{S_1}(r_6) = 1 \\ v_{S_2}(r_4) = 1 & v_{S_2}(r_5) = I & v_{S_2}(r_6) = 2 \\ v_{S_3}(r_4) = 2 & v_{S_3}(r_5) = 1 & v_{S_3}(r_6) = I \\ v_{S_4}(r_4) = 2 & v_{S_4}(r_5) = 2 & v_{S_4}(r_6) = I \end{array}$$

Con el grado de satisfación se forma el vector de satisfacción. De acuerdo a los grados de satisfacción para las reglas y la definición 3.24 se forman los siguientes vectores de satisfacción para los *answer sets*:

$$\begin{aligned} V_{S_1} &: (1, \; I, \; 1). \\ V_{S_2} &: (1, \; I, \; 2). \\ V_{S_3} &: (2, \; 1, \; I). \\ V_{S_4} &: (2, \; 2, \; I). \end{aligned}$$

Con los vectores de satisfacción, se determina la solución preferida. La solución preferida se obtiene de comparar los vectores de satisfacción de los *answer sets* de acuerdo a la definición 3.25 se tienen los resultados mostrados en la Tabla B.7. De la Tabla B.7 y por la definición 3.26 se tiene que S_1 es un modelo óptimo para este ASOP, siendo la solución preferida para el problema.

	S_1	S_2	S_3	S_4
S_1	$=$	$>$	$>$	$>$
S_2	$<$	$=$	$=$	$=$
S_3	$<$	$=$	$=$	$>$
S_4	$<$	$=$	$<$	$=$

Tabla B.7: *Problema pd*: ASOP.

B.1.7. Comparación de Resultados

Las Soluciones candidatas del problema de la sección B.1.1, están definidas como:

1. {*metales, trombon*}.

2. {*metales, trompeta*}.

3. {*cuerdas, guitarra*}.

4. {*cuerdas, mandolina*}.

Con las opciones anteriores al aplicarse cada uno de los enfoques vistos se tiene la Tabla B.8. La solución preferida para este problema es la opción {*metales, trombon* }.

Enfoque	**Criterio**	**Solución Preferida**
Sentido Común		1,2,3
LPOD	Inclusión	1
	Cardinalidad	1
	Pareto	1
PLP	inclusión	1
	Cardinalidad	1
ASOP		1

Tabla B.8: *Problema pd*. Comparación de soluciones preferidas

B.2. Problemas de Preferencias con Restricciones Simples

En las siguientes secciones se presenta un problema de *preferencias con restricciones simples*, (*problema prs*). Se presenta la descripción de un problema y a partir de ello, se modela y resuelve de acuerdo a los enfoques vistos. Por último se presenta un concentrado de resultados para realizar una comparación de los resultados obtenidos.

B.2.1. Descripción del Problema

El problema de *preferencias con restricciones simples* con el que se trabajará esta definido en el problema B.2.

Problema B.2. Omar esta seleccionado para participar en el `concurso de programación`, por lo que tiene que prepararse. Ademas, tiene otros proyectos por entregar como lo son: finalizar su `servicio social` y continuar con su `tesis`. Sabe que para continuar su `tesis` primero debe de terminar su `servicio social`, por lo que prefiere terminar primero el `servicio social`. Como ya ha participado en concursos similares sabe que puede tener una buena participación, por lo que prefiere participar. Sabe que puede aplazar la finalización de su `servicio social`, ya que el `concurso` está próximo, y prefiere `concursar`. Cabe mencionar que Omar prefiere `programar` sobre `redactar` documento alguno.

Del problema B.2 se tiene la información para modelar el problema. La información es relacionada con identificadores de acuerdo a lo siguiente:

concurso:	representa participar en el concurso de programación.
servSoc:	representa realizar el servicio social.
tesis:	representa trabajar en la tesis.
programar:	representa programar.
redactar:	representa redactar documentos.

Los identificadores ayudarán más adelante dentro de los enfoques, para que las reglas queden en forma sintetizada, siendo fáciles de comprender.

B.2.2. Modelado del Problema

Considerando la descripción del problema B.2 para el problema de Omar y de acuerdo a la definición 4.1 se tiene el siguiente modelo del problema:

P_{desc} : Las opciones de proyectos son: *concurso*, *servSoc* o *tesis*.
Las opciones de actividades son: *programar* o *redactar*.

P_{pref} : De las opciones de proyectos prefiere: *concurso* `sobre` *servSoc* `sobre` *tesis*.
De las opciones de actividades prefiere: *programar* `sobre` *redactar*.
Si no realiza el *servSoc* no puede hacer la *tesis*.

B.2.3. Solución por el Sentido Común

Para el problema B.2 de Omar, las soluciones son:

1. *concurso, programar.*

2. *concurso, redactar.*

3. *servSoc, programar.*

4. *servSoc, redactar.*

De acuerdo a las preferencias descritas se tienen las siguientes soluciones preferidas:

- *concurso, programar.* Solución preferida por que tanto *programar* como el *concurso* son las cosas más preferidas por Omar.
- *concurso, redactar.* Esta solución se tiene como preferida debido a que el *concurso* es preferido sobre las demas actividades.
- *servSoc, programar.* Solución preferida por que *programar* para Omar es una opción más preferida.

B.2.4. Solución por LPODs

Del modelo de la sección B.2.2 para el problema B.2 y de acuerdo a la definición 3.1 se obtienen las siguientes reglas que modelan el problema como un LPOD:

$$r_1: \quad concurso \times servSoc \times tesis.$$

$$r_2: \quad programar \times redactar.$$

$$r_3: \quad \leftarrow tesis, \, not \, servSoc.$$

Donde las r_i son los identificadores de las reglas. La regla r_3 es una restricción.

De acuerdo a la definición 3.4 del enfoque de los LPODs, las soluciones de los *split programs* forman el conjunto de soluciones candidatas del LPOD. Las soluciones candidatas del ejemplo son:

$$M_1: \quad \{concurso, programar\}.$$

$$M_2: \quad \{concurso, redactar\}.$$

$$M_3: \quad \{servSoc, programar\}.$$

$$M_4: \quad \{servSoc, redactar\}.$$

Donde los M_i son los identificadores de los *answer sets* para el conjunto de soluciones candidatas. De los *answer sets* se busca la solución preferida. La solución preferida se obtiene de la comparación de los grados de satisfacción. Los grados de satisfacción de las reglas para los *answer sets*, de acuerdo a la definición 3.5, están dados como:

$$
\begin{array}{ll}
deg_{M_1}(r_1) = 1 & deg_{M_1}(r_2) = 1 \\
deg_{M_2}(r_1) = 1 & deg_{M_2}(r_2) = 2 \\
deg_{M_3}(r_1) = 2 & deg_{M_3}(r_2) = 1 \\
deg_{M_4}(r_1) = 2 & deg_{M_4}(r_2) = 2
\end{array}
$$

Al agrupar las reglas que tienen el mismo grado de satisfacción para los *answer sets*, los conjuntos están formados como:

$$
\begin{array}{ll}
S^1_{M_1} = \{r_1, r_2\} & S^2_{M_1} = \{\ \} \\
S^1_{M_2} = \{r_1\} & S^2_{M_2} = \{r_2\} \\
S^1_{M_3} = \{r_2\} & S^2_{M_3} = \{r_1\} \\
S^1_{M_4} = \{\ \} & S^2_{M_4} = \{r_1, r_2\}
\end{array}
$$

Con los valores anteriores se determina la solución preferida a partir de un criterio de preferencia. A continuación se obtiene la solución preferida por los criterios con los que trabajan los LPODs.

Preferencia por Inclusión

De acuerdo al criterio de preferencia por inclusión dado en la definición 3.6 el resultado de comparar los *answer sets* del ejemplo se muestra en la Tabla B.9. De la

	M_1	M_2	M_3	M_4
M_1	$=$	$>$	$>$	$>$
M_2	$<$	$=$	$=$	$>$
M_3	$<$	$=$	$=$	$>$
M_4	$<$	$<$	$<$	$=$

Tabla B.9: *Problema prs*: LPOD Preferencia por Inclusión.

información de la Tabla B.9 y por la definición 3.7 se tiene que M_1 es un *answer set* preferido por inclusión, siendo este la solución preferida para el problema.

Preferencia por Cardinalidad

De acuerdo al criterio de preferencia por cardinalidad dado en la definición 3.8 el resultado de comparar los *answer sets* del ejemplo se muestra en la Tabla B.10.

	M_1	M_2	M_3	M_4
M_1	$=$	$>$	$>$	$>$
M_2	$<$	$=$	$=$	$>$
M_3	$<$	$=$	$=$	$>$
M_4	$<$	$<$	$<$	$=$

Tabla B.10: *Problema prs*: LPOD Preferencia por Cardinalidad.

De la información de la Tabla B.10 y por la definición 3.9 se tiene que M_1 es un *answer set* preferido por cardinalidad, siendo este la solución preferida para el problema.

Preferencia por Pareto

De acuerdo al criterio de preferencia por Pareto dado en la definición 3.10 el resultado de comparar los *answer sets* del ejemplo se muestra en la Tabla B.11. De la

	M_1	M_2	M_3	M_4
M_1	$=$	$>$	$>$	$>$
M_2	$<$	$=$	$=$	$>$
M_3	$<$	$=$	$=$	$>$
M_4	$<$	$<$	$<$	$=$

Tabla B.11: *Problema prs*: LPOD Preferencia por Pareto.

información de la Tabla B.11 y por la definición 3.11 se tiene que M_1 es un *answer set* preferido por Pareto, siendo este la solución preferida para el problema.

B.2.5. Solución por PLPs

Del modelo de la sección B.2.2 para el problema B.2 y de acuerdo a la definición 3.12, el problema se puede modelar como un PLP formado por las siguientes reglas:

$$\begin{aligned}
r_1 &: \; concurso \vee servSoc \vee tesis \leftarrow . \\
r_2 &: \; programar \vee redactar \leftarrow . \\
r_3 &: \; \leftarrow tesis, \, not\, servSoc. \\
r_4 &: \; concurso * servSoc * tesis \stackrel{pr}{\leftarrow} . \\
r_5 &: \; programar * redactar \stackrel{pr}{\leftarrow} .
\end{aligned}$$

Se tiene que los r_i son los identificadores para las reglas. El programa lógico está compuesto por las reglas de r_1 a r_3 y $Pref_P$ por el conjunto $\{r_4, r_5\}$.

Con las reglas del problema modelado en PLP, se busca la solución preferida. Para determinarla se debe de resolver la parte lógica del problema para encontrar la soluciones candidatas. Las soluciones candidatas son determinadas por lo descrito en la definición 3.13.

Entonces considerando la parte lógica para el PLP. De las reglas r_1 a r_4 se obtienen las siguientes soluciones candidatas:

$$\begin{aligned}
M_1 &: \; \{concurso, \, programar\}. \\
M_2 &: \; \{servSoc, \, programar\}. \\
M_3 &: \; \{concurso, \, redactar\}. \\
M_4 &: \; \{servSoc, \, redactar\}.
\end{aligned}$$

Teniendo a los M_i como los identificadores de los *answer sets* del conjunto de soluciones candidatas. De las soluciones candidatas se busca la solución preferida. La solución preferida se tiene de comparar los grados de satisfacción. Los grados de satisfacción de las reglas para los *answer sets*, de acuerdo a la definición 3.14, se tienen definidos como:

$$deg_{M_1}(r_4) = 1 \qquad deg_{M_1}(r_5) = 1$$
$$deg_{M_2}(r_4) = 2 \qquad deg_{M_2}(r_5) = 1$$
$$deg_{M_3}(r_4) = 1 \qquad deg_{M_3}(r_5) = 2$$
$$deg_{M_4}(r_4) = 2 \qquad deg_{M_4}(r_5) = 2$$

Con el grado de satisfacción de cada regla de preferencia se tiene un conjunto de reglas de preferencia que tienen el mismo grado. Estos conjuntos están formados como:

$$S^1_{M_1} = \{r_4, r_5\} \qquad S^2_{M_1} = \{\ \}$$
$$S^1_{M_2} = \{r_5\} \qquad S^2_{M_2} = \{r_4\}$$
$$S^1_{M_3} = \{r_4\} \qquad S^2_{M_3} = \{r_5\}$$
$$S^1_{M_4} = \{\ \} \qquad S^2_{M_4} = \{r_4, r_5\}$$

Con los conjuntos anteriores se encuentra el *answer set* preferido. El *answer set* preferido se determina a partir de usar un criterio de preferencia. A continuación se obtiene la solución preferida por medio de los criterios con los que trabajan los PLPs.

Preferencia por Inclusión

De acuerdo a lo que se tiene en la definición 3.15 para el criterio de preferencia por inclusión se tiene la Tabla B.12 con los resultados de comparación: De la Tabla B.12 y

	M_1	M_2	M_3	M_4
M_1	$=$	$>$	$>$	$>$
M_2	$<$	$=$	$=$	$>$
M_3	$<$	$=$	$=$	$>$
M_4	$<$	$<$	$<$	$=$

Tabla B.12: *Problema prs*: PLP Preferencia por Inclusión.

por la definición 3.16 se tiene que M_1 es un *answer set* preferido por inclusión, siendo la solución preferida para el problema.

Preferencia por Cardinalidad

De acuerdo a lo que se tiene en la definición 3.17 para el criterio de preferencia por cardinalidad, se tienen los siguientes resultados de comparación, mostrados en la Tabla B.13. De la información de Tabla B.13 y por la definición 3.18 se tiene que M_1 es un

	M_1	M_2	M_3	M_4
M_1	$=$	$>$	$>$	$>$
M_2	$<$	$=$	$=$	$>$
M_3	$<$	$=$	$=$	$>$
M_4	$<$	$<$	$<$	$=$

Tabla B.13: *Problema prs*: PLP Preferencia por Cardinalidad.

answer set preferido por cardinalidad, siendo la solución preferida para el problema.

B.2.6. Solución por ASOPs

Para el modelo de la sección B.2.2 del problema B.2 y de acuerdo a las definiciones 3.19 y 3.22 se tienen las siguientes reglas que modelan el problema como un ASOP:

$$r_1: \quad concurso \vee servSoc \vee tesis \leftarrow .$$

$$r_2: \quad programar \vee redactar \leftarrow .$$

$$r_3: \quad \leftarrow tesis, \, not\, servSoc.$$

$$r_4: \quad programar > redactar \leftarrow .$$

$$r_5: \quad concurso > servSoc > tesis \leftarrow .$$

Teniendo que los r_i son los identificadores para las reglas. P_{gen} está formado por las reglas de r_1 a r_3 y P_{pref} por el conjunto $\{r_4, r_5\}$.

Con el modelado de un problema como ASOP se busca tener una solución preferida sobre un conjunto de soluciones candidatas. El conjunto de soluciones candidatas se forma por los *answer sets* generados de P_{gen}.

A continuación se tienen las soluciones generadas de P_{gen}:

$$
\begin{aligned}
M_1: &\quad \{concurso, programar\}.\\
M_2: &\quad \{servSoc, programar\}.\\
M_3: &\quad \{concurso, redactar\}.\\
M_4: &\quad \{servSoc, redactar\}.
\end{aligned}
$$

Teniendo que los S_i son los identificadores de los *answer sets*. Con los *answer sets* obtenidos se determina el grado de satisfacción. De acuerdo a la definición 3.23 se tiene que para los *answer sets*, el grado de satisfacción está dado como:

$$
\begin{array}{ll}
v_{S_1}(r_4) = 1 & v_{S_1}(r_5) = 1\\
v_{S_2}(r_4) = 1 & v_{S_2}(r_5) = 2\\
v_{S_3}(r_4) = 2 & v_{S_3}(r_5) = 1\\
v_{S_4}(r_4) = 2 & v_{S_4}(r_5) = 2
\end{array}
$$

Con el grado de satisfación se forma el vector de satisfacción. De acuerdo a los grados de satisfacción para las reglas y la definición 3.24 se forman los siguientes vectores de satisfacción para los *answer sets*:

$$
\begin{aligned}
V_{S_1}: &\quad (1,\ 1).\\
V_{S_2}: &\quad (1,\ 2).\\
V_{S_3}: &\quad (2,\ 1).\\
V_{S_4}: &\quad (2,\ 2).
\end{aligned}
$$

Con los vectores de satisfacción, se determina la solución preferida. La solución preferida se obtiene de comparar los vectores de satisfacción de los *answer sets* de acuerdo a la definición 3.25 se tienen los resultados mostrados en la Tabla B.14.

De la Tabla B.14 y por la definición 3.26 se tiene que S_1 es un modelo óptimo para este ASOP, siendo la solución preferida para el problema.

	S_1	S_2	S_3	S_4
S_1	=	>	>	>
S_2	<	=	=	>
S_3	<	=	=	>
S_4	<	<	<	=

Tabla B.14: *Problema prs*: ASOP.

B.2.7. Comparación de Resultados

Las Soluciones candidatas del problema de la sección B.2.1, están definidas como:

1. {*concurso, programar*}.

2. {*concurso, redactar*}.

3. {*servSoc, programar*}.

4. {*servSoc, redactar*}.

Con las opciones anteriores al aplicarse cada uno de los enfoques vistos se tiene la Tabla B.15. La solución preferida para este problema es la opción {*concurso, progra-*

Enfoque	**Criterio**	**Solución Preferida**
Sentido Común		1,2,3
LPOD	Inclusión	1
	Cardinalidad	1
	Pareto	1
PLP	inclusión	1
	Cardinalidad	1
ASOP		1

Tabla B.15: *Problema prs*. Comparación de soluciones preferidas.

mar}.

B.3. Problemas de Preferencias con Restricciones en las Dependencias

Las siguientes secciones presentan un problema de *preferencias con restricciones en las dependencias*, (*problema prd*). Se presenta una descripción del problema y a partir de ello, se modela y resuelve de acuerdo a los enfoques vistos. Al final se presenta un concentrado de resultados para realizar una comparación de los resultados obtenidos.

B.3.1. Descripción del Problema

El problema de *preferencias con restricciones en las dependencias* con el que se trabajará en esta clase de problemas se tiene en la descripción B.3.

Problema B.3. Pablo quiere comprar un autoestereo para su Automóvil. Los tipos de autoestereo que está contemplando son los que reproducen discos compactos. Tiene en mente dos modelos de autoestereo: el que reproduce discos compactos de audio normal o `convencionales`, y otro es el que reproduce discos con formato `mp3`. La idea de andar cargando muchos discos no le agrada, por lo que prefiere el que reproduce `mp3`. Para que su equipo de audio quede con buen sonido, requiere comprar un juego de bocinas, para el cual tiene dos opciones, la primera es comprar un juego de `bocinas compatibles` con cualquier estéreo, o comprar las que están en la misma línea que el estéreo que elija y asi obtener mejor fidelidad en el audio. Si compra el estéro reproductor de `mp3` requerira un `adaptador` para las `bocinas compatibles`, por lo que prefiere las `bocinas de línea` sobre las `compatibles`. Si compra el estereo `convencional`, la calidad de audio del disco es buena en cualquier tipo de bocinas por lo que prefiere las `bocinas compatibles` sobre las `de línea`. Por el momento no puede comprar las `bocinas de línea` para el estéreo reproductor `mp3`.

De la descripción B.3 se tiene la información para modelar el problema. La información se relaciona con identificadores de acuerdo a lo siguiente:

estereoMp3:	representa el estéreo reproductor de mp3.
estereoConv:	representa es estéreo convencional.
bocinasLinea:	representa las bocinas de línea.
bocinasComp:	representa las bocinas compatibles.
adaptador:	representa al adaptador que se necesita.

Los identificadores ayudarán más adelante dentro de los enfoques, para que las reglas queden en forma sintetizada, siendo fáciles de comprender.

B.3.2. Modelado del Problema

Considerando la descripción del problema B.3 para el problema de Pablo y de acuerdo a la definición 4.1 el problema se representa con el siguiente modelo:

P_{desc} : Las opciones de autoestereo son: *estereoConv* o *estereoMp3*.
Las opciones del tipo de bocinas son: *bocinalsLinea* o *bocinasComp*.
Si elige el *estereoMp3* y *bocinasComp*: necesita un *adaptador*.

P_{pref} : De las opciones de autoestereo prefiere: *estereoConv* `sobre` *estereoMp3*.
Si elige *estereoMp3* para bocinas prefiere: *bocinasLinea* `sobre` *bocinasComp*.
Si elige *estereoConv* para bocinas prefiere: *bocinasComp* `sobre` *bocinasLinea*.
Si elige el *estereoMp3*: no puede comprar las *bocinasLinea*.

B.3.3. Solución por el Sentido Común

Con el modelo de la sección B.3.2 del problema B.3 de Pablo se tienen las siguientes soluciones:

1. *estereoMp3, bocinasComp, adaptador.*
2. *estereoConv, bocinasLinea.*
3. *estereoConv, bocinasCompatibles.*

Al considerar las preferencias de Pablo se tienen las siguientes soluciones preferidas:

- *estereoMp3, bocinasComp, adaptador.* Esta solución es preferida por el nivel de preferencia que se tiene para el *estereoMp3*.
- *estereoConv, bocinasComp.* Esta solución es preferida por el nivel de preferencia que hay para las *bocinasComp* del *estereoConv*.

B.3.4. Solución por LPODs

De la descripción del problema B.3, del modelo de la sección B.3.2 para el problema de Pablo y de acuerdo a la definición 3.1 se obtienen las siguientes reglas que modelan el problema como un LPOD:

$$\begin{aligned}
r_1 :&\quad estereoMp3 \times estereoConv.\\
r_2 :&\quad bocinasLinea \times bocinasComp \leftarrow estereoMp3.\\
r_3 :&\quad bocinasComp \times bocinasLinea \leftarrow estereoConv.\\
r_4 :&\quad adaptador \leftarrow estereoMp3, bocinasComp.\\
r_5 :&\quad \leftarrow bocinasLinea,\ estereoMp3.
\end{aligned}$$

Donde las r_i son los identificadores de las reglas. Las reglas r_2 y r_3 están relacionadas con las opciones de la regla r_1, r_4 y r_5 son restricciones.

De acuerdo a la definicion 3.4 del enfoque de los LPODs, las soluciones de los *split programs* forman el conjunto de soluciones candidatas del LPOD. Las soluciones candidatas del ejemplo son:

$$M_1: \ \{estereoMp3, \, bocinasComp, \, adaptador\}.$$
$$M_2: \ \{estereoConv, \, bocinasLinea\}.$$
$$M_3: \ \{estereoConv, \, bocinasComp\}.$$

Donde los M_i son los identificadores de los *answer sets* para el conjunto de soluciones candidatas. De los *answer sets* se busca la solución preferida. La solución preferida se obtiene de la comparación de los grados de satisfacción. Los grados de satisfacción de las reglas para los *answer sets*, de acuerdo a la definición 3.5, están dados como:

$$\begin{array}{lll} deg_{M_1}(r_1) = 1 & deg_{M_1}(r_2) = 2 & deg_{M_1}(r_3) = 1 \\ deg_{M_2}(r_1) = 2 & deg_{M_2}(r_2) = 1 & deg_{M_2}(r_3) = 2 \\ deg_{M_3}(r_1) = 2 & deg_{M_3}(r_2) = 1 & deg_{M_3}(r_3) = 1 \end{array}$$

Al agrupar las reglas que tienen el mismo grado de satisfacción para los *answer sets*, los conjuntos están formados como:

$$\begin{array}{ll} S^1_{M_1} = \{r_1, \, r_3\} & S^2_{M_1} = \{r_2\} \\ S^1_{M_2} = \{r_2\} & S^2_{M_2} = \{r_1, \, r_3\} \\ S^1_{M_3} = \{r_2, \, r_3\} & S^2_{M_3} = \{r_1\} \end{array}$$

Con los valores anteriores se determina la solución preferida a partir de un criterio de preferencia. A continuación se obtiene la solución preferida por los criterios con los que trabajan los LPODs.

Preferencia por Inclusión

De acuerdo al criterio de preferencia por inclusión dado en la definición 3.6 el resultado de comparar los *answer set*s del ejemplo se muestra en la Tabla B.16. De la información de la Tabla B.16 se puede ver que no es posible aplicar la definición 3.7 para determinar un *answer set* preferido por inclusión. En este caso se tiene que tanto

	M_1	M_2	M_3
M_1	$=$	$=$	$=$
M_2	$=$	$=$	$<$
M_3	$=$	$>$	$=$

Tabla B.16: *Problema prd*: LPOD Preferencia por Inclusión.

M_1 como M_3 son *answer sets* igualmente preferidos por inclusión. Esta conclusión se obtiene del renglón para M_3.

Preferencia por Cardinalidad

De acuerdo al criterio de preferencia por cardinalidad dado en la definición 3.8 el resultado de comparar los *answer sets* del ejemplo se muestra en la Tabla B.17. De

	M_1	M_2	M_3
M_1	$=$	$>$	$=$
M_2	$<$	$=$	$<$
M_3	$=$	$>$	$=$

Tabla B.17: *Problema prd*: LPOD Preferencia por Cardinalidad.

la información de la Tabla B.17 se observa que los *answer sets* M_1 y M_3 son igualmente preferidos por cardinalidad. Lo anterior se obtiene de ver que los renglones que corresponden a estos *answer sets* son iguales.

Preferencia por Pareto

De acuerdo al criterio de preferencia por Pareto dado en la definición 3.10 el resultado de comparar los *answer sets* del ejemplo se muestra en la Tabla B.18. De la información de la Tabla B.18 se obtiene que M_1 y M_3 son *answer sets* igualmente preferidos por Pareto. Lo anterior se obtiene del renglón de M_3.

	M_1	M_2	M_3
M_1	$=$	$=$	$=$
M_2	$=$	$=$	$<$
M_3	$=$	$>$	$=$

Tabla B.18: *Problema prd*: LPOD Preferencia por Pareto.

B.3.5. Solución por PLPs

Del modelo de la sección B.3.2 para el problema B.3 y de acuerdo a la definición 3.12 el problema se puede modelar como un PLP formado por las siguientes reglas:

$$
\begin{aligned}
r_1 :&\quad estereoMp3 \vee estereoConv \leftarrow .\\
r_2 :&\quad bocinasLinea \vee bocinasComp \leftarrow .\\
r_3 :&\quad adaptador \leftarrow estereoMp3, bocinasComp.\\
r_4 :&\quad \leftarrow bocinasLinea,\ estereoMp3.\\
r_5 :&\quad estereoMp3 * estereoConv \overset{pr}{\leftarrow} .\\
r_6 :&\quad bocinasLinea * bocinasComp \overset{pr}{\leftarrow} estereoMp3.\\
r_7 :&\quad bocinasComp * bocinasLinea \overset{pr}{\leftarrow} estereoConv.
\end{aligned}
$$

Se tiene que los r_i son los identificadores para las reglas. El programa lógico está compuesto por el conjunto de las reglas r_1 a r_4 y $Pref_P$ por el conjunto $\{r_5,\ r_6,\ r_7\}$.

Con las reglas del problema modelado en PLP, se busca la solución preferida. Para obtenerla se debe de resolver la parte lógica del problema para encontrar las soluciones candidatas. Las soluciones candidatas son determinadas por lo descrito en la definición 3.13.

Entonces considerando la parte lógica para el PLP. De las reglas r_1 a r_4 se obtienen las siguientes soluciones candidatas:

$$M_1: \ \{estereoMp3,\, bocinasComp,\, adaptador\}.$$
$$M_2: \ \{estereoConv,\, bocinasLinea\}.$$
$$M_3: \ \{estereoConv,\, bocinasComp\}.$$

Teniendo a los M_i como los identificadores de los *answer sets* del conjunto de soluciones candidatas. De las soluciones candidatas se busca la solución preferida. La solución preferida se tiene de comparar los grados de satisfacción. Los grados de satisfacción de las reglas para los *answer sets*, de acuerdo a la definición 3.14, se tienen definidos como:

$$\begin{array}{lll} deg_{M_1}(r_5) = 1 & deg_{M_1}(r_6) = 2 & deg_{M_1}(r_7) = 1 \\ deg_{M_2}(r_5) = 2 & deg_{M_2}(r_6) = 1 & deg_{M_2}(r_7) = 2 \\ deg_{M_3}(r_5) = 2 & deg_{M_3}(r_6) = 1 & deg_{M_3}(r_7) = 1 \end{array}$$

Con el grado de satisfacción de cada regla de preferencia se tiene un conjunto de reglas de preferencia que tienen el mismo grado. Estos conjuntos están formados como:

$$\begin{array}{ll} S^1_{M_1} = \{r_5,\, r_7\} & S^2_{M_1} = \{r_6\} \\ S^1_{M_2} = \{r_6\} & S^2_{M_2} = \{r_5,\, r_7\} \\ S^1_{M_3} = \{r_6,\, r_7\} & S^2_{M_3} = \{r_5\} \end{array}$$

Con los conjuntos anteriores se encuentra el *answer set* preferido. El *answer set* preferido se determina a partir de usar un criterio de preferencia. A continuación se obtiene la solución preferida por medio de los criterios con los que trabajan los PLPs.

Preferencia por Inclusión

De acuerdo a lo que se tiene en la definición 3.15 para el criterio de preferencia por inclusión se tiene la Tabla B.19 con los resultados de comparación: Del renglón para

	M_1	M_2	M_3
M_1	$=$	$=$	$=$
M_2	$=$	$=$	$<$
M_3	$=$	$>$	$=$

Tabla B.19: *Problema prd*: PLP Preferencia por Inclusión.

M_3 de la Tabla B.19 se tiene que M_1 y M_3 son *answer sets* igualmente preferidos por inclusión.

Preferencia por Cardinalidad

De acuerdo a lo que se tiene en la definición 3.17 para el criterio de preferencia por cardinalidad, se tienen los siguientes resultados de comparación, mostrados en la Tabla B.20. De la información de Tabla B.20 se observa que M_1 y M_3 son *answer sets*

	M_1	M_2	M_3
M_1	$=$	$>$	$=$
M_2	$<$	$=$	$<$
M_3	$=$	$>$	$=$

Tabla B.20: *Problema prd*: PLP Preferencia por Cardinalidad.

igualmente preferidos por cardinalidad.

B.3.6. Solución por ASOPs

Para el problema B.3 asi como el modelo de la sección B.3.2 y de acuerdo a las definiciones 3.19 y 3.22 se tienen las siguientes reglas que modelan el problema como un ASOP:

$$
\begin{aligned}
r_1 &: \ estereoMp3 \vee estereoConv \leftarrow . \\
r_2 &: \ bocinasLinea \vee bocinasComp \leftarrow . \\
r_3 &: \ adaptador \leftarrow estereoMp3, bocinasComp. \\
r_4 &: \ \leftarrow bocinasLinea,\ estereoMp3. \\
r_5 &: \ estereoMp3 > estereoConv \leftarrow . \\
r_6 &: \ bocinasLinea > bocinasComp \leftarrow estereoMp3. \\
r_7 &: \ bocinasComp > bocinasLinea \leftarrow estereoConv.
\end{aligned}
$$

Teniendo que los r_i son los identificadores para las reglas. P_{gen} está formado por las reglas de r_1 a r_4 y P_{pref} por el conjunto $\{r_5,\ r_6,\ r_7\}$.

Con el modelado de un problema como ASOP se busca tener una solución preferida sobre un conjunto de soluciones candidatas. El conjunto de soluciones candidatas se forma por los *answer sets* generados de P_{gen}.

A continuación se tienen las soluciones generadas de P_{gen}:

$$
\begin{aligned}
S_1 &: \ \{estereoMp3,\ bocinasComp,\ adaptador\}. \\
S_2 &: \ \{estereoConv,\ bocinasLinea\}. \\
S_3 &: \ \{estereoConv,\ bocinasComp\}.
\end{aligned}
$$

Teniendo que los S_i son los identificadores de los *answer sets*.

Con los *answer sets* obtenidos se determina el grado de satisfacción. De acuerdo a la definición 3.23 se tiene que para los *answer sets*, el grado de satisfacción está dado como:

$$
\begin{array}{lll}
v_{S_1}(r_5) = 1 & v_{S_1}(r_6) = 2 & v_{S_1}(r_7) = I \\
v_{S_2}(r_5) = 2 & v_{S_2}(r_6) = I & v_{S_2}(r_7) = 2 \\
v_{S_3}(r_5) = 2 & v_{S_3}(r_6) = I & v_{S_3}(r_7) = 1
\end{array}
$$

Con el grado de satisfación se forma el vector de satisfacción. De acuerdo a los grados de satisfacción para las reglas y la definición 3.24 se forman los siguientes vectores de satisfacción para los *answer sets*:

$$V_{S_1}: \quad (1,\, 2,\, I).$$
$$V_{S_2}: \quad (2,\, I,\, 2).$$
$$V_{S_3}: \quad (2,\, I,\, 1).$$

Con los vectores de satisfacción, se determina la solución preferida. La solución preferida se obtiene de comparar los vectores de satisfacción de los *answer sets* de acuerdo a la definición 3.25 se tienen los resultados mostrados en la Tabla B.21.

	S_1	S_2	S_3
S_1	$=$	$=$	$=$
S_2	$=$	$=$	$<$
S_3	$=$	$>$	$=$

Tabla B.21: *Problema prd*: ASOP.

De la Tabla B.21 se tiene que S_1 y S_3 son las soluciones preferidas para el problema resuelto por el enfoque de ASOP.

B.3.7. Comparación de Resultados

Las Soluciones candidatas del problema de la sección B.3.1, están definidas como:

1. {*estereoMp3, bocinasComp, adaptador*}.
2. {*estereoConv, bocinasLinea*}.
3. {*estereoConv, bocinasComp*}.

Con las opciones anteriores al aplicarse cada uno de los enfoques vistos se tiene la Tabla B.22. Para este ejemplo se tienen dos soluciones preferidas: {*estereoMp3, boci-*

Enfoque	**Criterio**	**Solución Preferida**
Sentido Común		1,3
LPOD	Inclusión	1,3
	Cardinalidad	1,3
	Pareto	1,3
PLP	inclusión	1,3
	Cardinalidad	1,3
ASOP		1,3

Tabla B.22: *Problema prd*. Comparación de soluciones preferidas.

nasComp, adaptador} y {*estereoConv, bocinasComp*}.

B.4. Problemas de Preferencias con Dependencias y con Restricciones Simples

En las siguientes secciones se trabaja con un problema de *preferencias con dependencias y con restricciones simples*, (*problema pdrs*). Se presenta la descripción del problema y a partir de ello, se tienen los modelos de acuerdo a los enfoques analizados. Por último se presenta un concentrado de resultados a manera de comparar los resultados obtenidos.

B.4.1. Descripción del Problema

El problema de *preferencias con dependencias y con restricciones simples* con el que se trabajará en esta clase se tiene en el problema B.4.

Problema B.4. Raúl quiere comprar un teléfono celular. Las compañias que tiene para escoger y comprar el celular son las de `Celtel` y la compañía `Celiusa`. Prefiere la más comercial por cuestiones de recarga, que es `Celtel`. Entre los servicios de red existe la `TDMA` y la `GSM`, prefiriendo la `TDMA` por rango de cobertura. La `Mensajería Multimedia` se obtiene por la compaía `Celtel` en su modalidad de `GSM`. Por el momento no hay servicio de telefonía en modo `TDMA` en cualquier compañía.

De la descripción del problema B.4 se tiene la información para modelar el problema. La información se relaciona con identificadores de acuerdo a lo siguiente:

tdma:	representa a la red tdma.
gsm:	representa a la red gsm.
celtel:	representa a la compañía Celtel.
celiusa:	representa a la compañía Celiusa.
multimedia:	representa el servicio de mensajería multimedia.

Los identificadores ayudarán más adelante dentro de los enfoques, para que las reglas queden en forma sintetizada, siendo fáciles de comprender.

B.4.2. Modelado del Problema

Considerando la descripción del problema B.4 para el problema de Raúl y de acuerdo a la definición 4.1 el problema queda modelado como se muestra a continuación:

P_{desc} :	Las opciones de compañia son: *celtel* o *celiusa.*
	Las opciones de servicio son: *tdma* o *gsm.*
	Si elige la compañia *celtel* con el servicio *gsm*: tiene *multimedia.*
P_{pref} :	De las opciones de red prefiere: *tdma* `sobre` *gsm.*
	Si elige el servicio *gsm* prefiere la compañia: *celtel* `sobre` *celiusa.*
	Si elige el servicio *tdm* prefiere la compañia: *celiusa* `sobre` *celtel.*
	no hay servicio *tdma.*

B.4.3. Solución por el Sentido Común

Para el problema de Raúl, con la descripción en el problema B.4 modelado en la sección B.4.2 se tiene las siguientes soluciones:

1. *celtel, gsm, multimedia.*

2. *celiusa, gsm, no tdma.*

De las soluciones al problema al considerar la preferencias se tiene la siguiente solución preferida:

- *celtel, gsm, multimedia.* Esta solución se tiene como preferida debido a la preferencia de compañía que se tiene para ese tipo de red.

B.4.4. Solución por LPODs

De la descripción del problema B.4 para Raúl modelado en la sección B.4.2 y de acuerdo a la definición 3.1 se obtienen las siguientes reglas que modelan el problema como un LPOD:

$$\begin{aligned} r_1 &: \quad tdma \times gsm. \\ r_2 &: \quad celtel \times celiusa \leftarrow gsm. \\ r_3 &: \quad celisua \times celtel \leftarrow tdma. \\ r_4 &: \quad multimedia \leftarrow celtel, gsm. \\ r_5 &: \quad \leftarrow tdma. \end{aligned}$$

Donde las r_i son los identificadores de las reglas. Las opciones de la regla r_1 son las que están relacionadas con las regla r_2 y r_3. La regla r_5 es una restricción.

De acuerdo a la definición 3.4 del enfoque de los LPODs, las soluciones de los *split programs* forman el conjunto de soluciones candidatas del LPOD. Las soluciones candidatas del ejemplo son:

$$M_1: \quad \{celtel,\, gsm,\, multimedia\}.$$
$$M_2: \quad \{celiusa,\, gsm\}.$$

Donde los M_i son los identificadores de los *answer sets* para el conjunto de soluciones candidatas.De los *answer sets* se busca la solución preferida. La solución preferida se obtiene de la comparación de los grados de satisfacción. Los grados de satisfacción de las reglas para los *answer sets*, de acuerdo a la definición 3.5, están dados como:

$$deg_{M_1}(r_1) = 2 \qquad deg_{M_1}(r_2) = 1 \qquad deg_{M_1}(r_3) = 1$$
$$deg_{M_2}(r_1) = 2 \qquad deg_{M_2}(r_2) = 2 \qquad deg_{M_2}(r_3) = 1$$

Al agrupar las reglas que tienen el mismo grado de satisfacción para los answer sets, los conjuntos están formados como:

$$S^1_{M_1} = \{r_2,\, r_3\} \qquad S^2_{M_1} = \{r_1\}$$
$$S^1_{M_2} = \{r_3\} \qquad S^2_{M_2} = \{r_1,\, r_2\}$$

Con los valores anteriores se determina la solución preferida a partir de un criterio de preferencia. A continuación se obtiene la solución preferida por los criterios con los que trabajan los LPODs.

Preferencia por Inclusión

De acuerdo al criterio de preferencia por inclusión dado en la definición 3.6 el resultado de comparar los *answer set*s del ejemplo se muestra en la siguiente Tabla B.23. De la información de la Tabla B.23 y por la definición 3.7 se tiene que M_1 es un *answer set* preferido por inclusión, siendo este la solución preferida para el problema.

	M_1	M_2
M_1	$=$	$>$
M_2	$<$	$=$

Tabla B.23: *Problema pdrs*: LPOD Preferencia por Inclusión.

Preferencia por Cardinalidad

De acuerdo al criterio de preferencia por cardinalidad dado en la definición 3.8 el resultado de comparar los *answer sets* del ejemplo se muestra en la Tabla B.24. De la

	M_1	M_2
M_1	$=$	$>$
M_2	$<$	$=$

Tabla B.24: *Problema pdrs*: LPOD Preferencia por Cardinalidad.

información de la Tabla B.24 y por la definición 3.9 se tiene que M_1 es un *answer set* preferido por cardinalidad, siendo este la solución preferida para el problema.

Preferencia por Pareto

De acuerdo al criterio de preferencia por Pareto dado en la definición 3.10 el resultado de comparar los *answer sets* del ejemplo se muestra en la Tabla B.25. De la

	M_1	M_2
M_1	$=$	$>$
M_2	$<$	$=$

Tabla B.25: *Problema pdrs*: LPOD Preferencia por Pareto.

información de la Tabla B.25 y por la definición 3.11 se tiene que M_1 es un *answer set* preferido por Pareto, siendo este la solución preferida para el problema.

B.4.5. Solución por PLPs

Considerando la descripción del problema B.4, el modelo de la sección B.4.2 para el problema de Raúl y de acuerdo a la definición 3.12 el problema se puede modelar como un PLP formado por las siguientes reglas:

$$
\begin{aligned}
r_1 &: \quad celtel \vee celiusa \leftarrow . \\
r_2 &: \quad tdma \vee gsm \leftarrow . \\
r_3 &: \quad multimedia \leftarrow celtel, gsm. \\
r_4 &: \quad \leftarrow tdma. \\
r_5 &: \quad tdma * gsm \overset{pr}{\leftarrow} . \\
r_6 &: \quad celtel * celiusa \overset{pr}{\leftarrow} gsm. \\
r_7 &: \quad celiusa * celtel \overset{pr}{\leftarrow} tdma.
\end{aligned}
$$

Se tiene que los r_i son los identificadores para las reglas. El programa lógico está compuesto por el conjunto de las reglas r_1 a r_4 y $Pref_P$ por el conjunto $\{r_5,\, r_6,\, r_7\}$.

Con las reglas del problema modelado en PLP, se busca la solución preferida. Para obtenerla se debe de resolver la parte lógica del problema para encontrar las soluciones candidatas. Las soluciones candidatas son determinadas por lo descrito en la definición 3.13.

Entonces considerando la parte lógica para el PLP. De las reglas r_1 a r_4 se obtienen las siguientes soluciones candidatas:

$$
\begin{aligned}
M_1 &: \quad \{celtel,\, gsm,\, multimedia\}. \\
M_2 &: \quad \{celiusa,\, gsm\}.
\end{aligned}
$$

Teniendo a los M_i como los identificadores de los *answer sets* del conjunto de soluciones candidatas. De las soluciones candidatas se busca la solución preferida. La solución preferida se tiene de comparar los grados de satisfacción. Los grados de satisfacción de las reglas para los *answer sets*, de acuerdo a la definición 3.14, se tienen definidos como:

$$
\begin{array}{lll}
deg_{M_1}(r_5) = 2 & deg_{M_1}(r_6) = 1 & deg_{M_1}(r_7) = 1 \\
deg_{M_2}(r_5) = 2 & deg_{M_2}(r_6) = 2 & deg_{M_1}(r_7) = 1
\end{array}
$$

Con el grado de satisfacción de cada regla de preferencia se tiene un conjunto de reglas de preferencia que tienen el mismo grado. Estos conjuntos están formados como:

$$S^1_{M_1} = \{r_6, r_7\} \qquad S^2_{M_1} = \{r_5\}$$
$$S^1_{M_2} = \{r_7\} \qquad S^2_{M_2} = \{r_5, r_6\}$$

Con los conjuntos anteriores se encuentra el *answer set* preferido. El *answer set* preferido se determina a partir de usar un criterio de preferencia. A continuación se obtiene la solución preferida por medio de los criterios con los que trabajan los PLPs.

Preferencia por Inclusión

De acuerdo a lo que se tiene en la definición 3.15 para el criterio de preferencia por inclusión se tiene la Tabla B.26 con los resultados de comparación. De la Tabla B.26 y

	M_1	M_2
M_1	$=$	$>$
M_2	$<$	$=$

Tabla B.26: *Problema pdrs*: PLP Preferencia por Inclusión.

por la definición 3.16 se tiene que M_1 es un *answer set* preferido por inclusión, siendo la solución preferida para el problema.

Preferencia por Cardinalidad

De acuerdo a lo que se tiene en la definición 3.17 para el criterio de preferencia por cardinalidad, se tienen los siguientes resultados de comparación, mostrados en la Tabla B.27. De la información de Tabla B.27 y por la definición 3.18 se tiene que M_1 es un *answer set* preferido por cardinalidad, siendo la solución preferida para el problema.

	M_1	M_2
M_1	$=$	$>$
M_2	$<$	$=$

Tabla B.27: *Problema pdrs*: PLP Preferencia por Cardinalidad.

B.4.6. Solución por ASOPs

Para el problema B.4, el modelo de la sección B.4.2 y de acuerdo a las definiciones 3.19 y 3.22 se tienen las siguientes reglas que modelan el problema como un ASOP:

$$\begin{aligned}
r_1 &: \ celtel \vee celiusa \leftarrow .\\
r_2 &: \ tdma \vee gsm \leftarrow .\\
r_3 &: \ multimedia \leftarrow celtel, gsm.\\
r_4 &: \ \leftarrow tdma.\\
r_5 &: \ tdma > gsm \leftarrow .\\
r_6 &: \ celtel > celiusa \leftarrow gsm.\\
r_7 &: \ celiusa > celtel \leftarrow tdma.
\end{aligned}$$

Teniendo que los r_i son los identificadores para las reglas. P_{gen} está formado por las reglas de r_1 a r_4 y P_{pref} por el conjunto $\{r_5, r_6, r_7\}$.

Con el modelado de un problema como ASOP se busca tener una solución preferida sobre un conjunto de soluciones candidatas. El conjunto de soluciones candidatas se forma por los *answer sets* generados de P_{gen}.

A continuación se tienen las soluciones generadas de P_{gen}:

$$\begin{aligned}
S_1 &: \ \{celtel,\ gsm,\ multimedia\}.\\
S_2 &: \ \{celiusa,\ gsm\}.
\end{aligned}$$

Teniendo que los S_i son los identificadores de los *answer sets*. Con los *answer sets* obtenidos se determina el grado de satisfacción. De acuerdo a la definición 3.23 se tiene que para los *answer sets*, el grado de satisfacción está dado como:

$$v_{S_1}(r_5) = 2 \qquad v_{S_1}(r_6) = 1 \qquad v_{S_1}(r_7) = I$$
$$v_{S_2}(r_5) = 2 \qquad v_{S_2}(r_6) = 2 \qquad v_{S_1}(r_7) = I$$

Con el grado de satisfación se forma el vector de satisfacción. De acuerdo a los grados de satisfacción para las reglas y la definición 3.24 se forman los siguientes vectores de satisfacción para los *answer sets*:

$$V_{S_1}: \quad (2,\, 1,\, I).$$
$$V_{S_2}: \quad (2,\, 2,\, I).$$

Con los vectores de satisfacción, se determina la solución preferida. La solución preferida se obtiene de comparar los vectores de satisfacción de los *answer sets* de acuerdo a la definición 3.25 se tienen los resultados mostrados en la Tabla B.28.

	S_1	S_2
S_1	$=$	$>$
S_2	$<$	$=$

Tabla B.28: *Problema pdrs*: ASOP.

De la Tabla B.28 y por la definición 3.26 se tiene que S_1 es un modelo óptimo para este ASOP, siendo la solución preferida para el problema.

B.4.7. Comparación de Resultados

Las Soluciones candidatas del problema de la sección B.4.1, están definidas como:

1. {*celtel, gsm, multimedia*}.

2. {*celiusa, gsm*}.

Enfoque	Criterio	Solución Preferida
Sentido Común		1
LPOD	Inclusión	1
	Cardinalidad	1
	Pareto	1
PLP	inclusión	1
	Cardinalidad	1
ASOP		1

Tabla B.29: *Problema pdrs*. Comparación de soluciones preferidas.

Con las opciones anteriores al aplicarse cada uno de los enfoques vistos se tiene la Tabla B.29. La solución preferida para este problema es la opción {*celtel, gsm, multimedia, no tdma*}.

B.5. Problemas de Preferencias Completas

Las siguientes secciones presentan un problema de *preferencias completas*, (*problema pc*). Se presenta la descripción del problema y a partir de ello, se modela y resuelve de acuerdo a los enfoques estudiados. Con los resultados se presenta un concentrado para realizar una comparación de lo obtenido.

B.5.1. Descripción del Problema

El problema de *preferencias completas* con el que se trabajará se describe en el problema B.5.

Problema B.5. Andrea está pensando comprar productos para su negocio. Lo que ella puede comprar en este momento son artículos de `temporada` o bien artículos que son `vigentes en cualquier temporada`. Andrea sabe que cualquiera de los artículos que quiera vender son buscados para su compra. Los artículos que puede vender son para el `hogar`, `oficina` o de `uso personal`, siendo ese orden el que prefieren los clientes para comprar artículos de `temporada`. Para el caso de artículos que `no son de temporada` los clientes los prefieren primero de `uso personal`, de `hogar` y por último para `oficina`. Ella vio que los artículos de `temporada` están saliendo `defectuosos`. Los artículos de `temporada` para `oficina` están `escasos`. Por el momento están agotados los artículos `personales` de cualquier línea que quiera, asi como los artículos de `hogar` para la línea de `temporada`.

De la descripción del problema B.5 se tiene la información para modelar el problema. La información se relaciona con identificadores de acuerdo a lo siguiente:

noTemporada:	representa a la vigencia en cualquier temporada.
temporada:	representa a la vegencia en la temporada actual.
hogar:	representa a los artículos del hogar.
oficina:	representa a los artículos de oficina.
personal:	representa a los artículos de uso personal.
defectuoso:	representa que los artículos que están defectuosos.
escaso:	representa que los artículos que están escasos.

Los identificadores ayudarán más adelante dentro de los enfoques, para que las reglas queden en forma sintetizada, siendo fáciles de comprender.

B.5.2. Modelado del Problema

Considerando la descripción del problema B.5 para el problema de Andrea y de acuerdo a la definición 4.1, el problema queda modelado como se muestra a continuación:

P_{desc} : Las opciones de productos son: *hogar*, *oficina* o *personal*.

Las opciones de tipo de producto son: *temporada* o *noTemporada*.

Si elige productos de *temporada*: salen *defectuosos*.

Si elige productos de *temporada* y *oficina*: están *escasos*.

P_{pref} : De las opciones de tipo de producto prefiere: *noTemporada* `sobre` *temporada*.

Si elige productos tipo de *temporada* prefiere: *hogar* `sobre` *oficina* `sobre` *personal*.

Si elige productos tipo de *noTemporada* prefiere: *personal* `sobre` *hogar* `sobre` *oficina*.

No hay productos de uso *personal*.

No hay productos de uso en el *hogar* para la linea de *temporada*.

B.5.3. Solución por el Sentido Común

Con el modelo de la sección B.5.2 para el problema B.5 se tienen como soluciones:

1. {*noTemporada, hogar*}.

2. {*noTemporada, oficina*}.

3. {*temporada, defectuoso, oficina, escaso*}.

Del conjunto de soluciones de acuerdo a las preferencias de Andrea se tienen las siguientes soluciones preferidas:

- *noTemporada, hogar.* Solución preferida por que los artículos que son *noTemporada* son preferidos sobre los artículos de *temporada*.

- *noTemporada, oficina.* Esta solución es preferida porque los artículos de *noTemporada* son preferidos sobre los artículos de *temporada*.

- *temporada, defectuoso, oficina, escaso.* Esta solución es preferida porque los artículos de *oficina* de *temporada* tienen un nivel de preferencia aceptable, y de acuerdo a las restricciones seria lo más preferido sobre los artícules de *temporada*.

B.5.4. Solución por LPODs

De la descripción del problema B.5, el modelo de la sección B.5.2 y de acuerdo a la definición 3.1 se obtienen las siguientes reglas que modelan el problema como un LPOD:

$$
\begin{aligned}
r_1 &: \; noTemporada \times temporada.\\
r_2 &: \; hogar \times oficina \times personal \leftarrow temporada.\\
r_3 &: \; personal \times hogar \times oficina \leftarrow noTemporada.\\
r_4 &: \; \leftarrow personal.\\
r_5 &: \; \leftarrow hogar, temporada.\\
r_6 &: \; defectuoso \leftarrow temporada.\\
r_7 &: \; escaso \leftarrow temporada, oficina.
\end{aligned}
$$

Donde las r_i son los identificadores de las reglas. Teniendo a r_2 y r_3 relacionadas con las opciones de r_1. r_4 es una restricción simple. r_5 restringe la opción de la dependencia de r_2 .

De acuerdo a la definicion 3.4, de las soluciones de los *split programs*, se tiene el conjunto de soluciones candidatas del LPOD. Las soluciones candidatas del ejemplo son:

$$
\begin{aligned}
M_1 &: \; \{noTemporada, hogar\}.\\
M_2 &: \; \{noTemporada, oficina\}.\\
M_3 &: \; \{temporada, defectuoso, oficina, escaso\}.
\end{aligned}
$$

Donde los M_i son los identificadores para los *answer sets* del conjunto de soluciones candidatas. De los *answer sets* se busca la solución preferida. La solución preferida se obtiene de la comparación de los grados de satisfacción. Los grados de satisfacción de las reglas para los *answer sets*, de acuerdo a la definición 3.5, están dados como:

$$deg_{M_1}(r_1) = 1 \qquad deg_{M_1}(r_2) = 1 \qquad deg_{M_1}(r_3) = 2$$
$$deg_{M_2}(r_1) = 1 \qquad deg_{M_2}(r_2) = 1 \qquad deg_{M_2}(r_3) = 3$$
$$deg_{M_3}(r_1) = 2 \qquad deg_{M_3}(r_2) = 2 \qquad deg_{M_3}(r_3) = 1$$

Al agrupar las reglas que tienen el mismo grado de satisfacción para los *answer sets*, los conjuntos están formados como:

$$S^1_{M_1} = \{r_1, r_2\} \qquad S^2_{M_1} = \{r_3\} \qquad S^2_{M_1} = \{\ \}$$
$$S^1_{M_2} = \{r_1, r_2\} \qquad S^2_{M_2} = \{\ \} \qquad S^2_{M_1} = \{r_3\}$$
$$S^1_{M_3} = \{r_3\} \qquad S^2_{M_3} = \{r_1, r_2\} \qquad S^2_{M_1} = \{\ \}$$

Con los valores anteriores se determina la solución preferida a partir de un criterio de preferencia. A continuación se obtiene la solución preferida por los criterios con los que trabajan los LPODs.

Preferencia por Inclusión

De acuerdo al criterio de preferencia por inclusión dado en la definición 3.6 el resultado de comparar los *answer set*s del ejemplo se muestra en la Tabla B.30.

	M_1	M_2	M_3
M_1	$=$	$>$	$=$
M_2	$<$	$=$	$=$
M_3	$=$	$=$	$=$

Tabla B.30: *Problema pc*: LPOD Preferencia por Inclusión.

De la información de la Tabla B.30 y por la definición 3.7 se tiene que M_1 y M_3 son *answer sets* preferidos por inclusión.

Preferencia por Cardinalidad

De acuerdo al criterio de preferencia por cardinalidad dado en la definición 3.8 el resultado de comparar los *answer sets* del ejemplo se muestra en la Tabla B.31.

	M_1	M_2	M_3
M_1	$=$	$>$	$>$
M_2	$<$	$=$	$>$
M_3	$<$	$<$	$=$

Tabla B.31: *Problema pc*: LPOD Preferencia por Cardinalidad.

De la información de la Tabla B.31 y por la definición 3.9 se tiene que M_1 es un *answer set* preferido por cardinalidad, siendo este la solución preferida para el problema.

Preferencia por Pareto

De acuerdo al criterio de preferencia por Pareto dado en la definición 3.10 el resultado de comparar los *answer sets* del ejemplo se muestra en la Tabla B.32. De la

	M_1	M_2	M_3
M_1	$=$	$>$	$=$
M_2	$<$	$=$	$=$
M_3	$=$	$=$	$=$

Tabla B.32: *Problema pc*: LPOD Preferencia por Pareto.

información de la Tabla B.32 y por la definición 3.11 se tiene que M_1 y M_3 son *answer sets* preferidos por Pareto.

B.5.5. Solución por PLPs

De la descripción del problema B.5 de Andrea, el modelo de la sección B.5.2 y de acuerdo a la definición 3.12 el problema se puede modelar como un PLP formado por las siguientes reglas:

$$
\begin{aligned}
r_1 &: \quad hogar \vee oficina \vee personal \leftarrow . \\
r_2 &: \quad noTemporada \vee temporada \leftarrow . \\
r_3 &: \quad defectuoso \leftarrow temporada. \\
r_4 &: \quad escaso \leftarrow temporada, oficina.
\end{aligned}
$$

$$
\begin{aligned}
r_5 &: \quad \leftarrow personal. \\
r_6 &: \quad \leftarrow hogar,\ temporada. \\
r_7 &: \quad hogar * oficina * personal \overset{pr}{\leftarrow} temporada. \\
r_8 &: \quad personal * hogar * oficina \overset{pr}{\leftarrow} noTemporada. \\
r_9 &: \quad noTemporada * temporada \overset{pr}{\leftarrow} .
\end{aligned}
$$

Se tiene que los r_i son los identificadores para las reglas. El programa lógico está compuesto por el conjunto de las reglas r_1 a r_6 y $Pref_P$ por las reglas de r_7 a r_9.

Con las reglas del problema modelado en PLP, se busca la solución preferida. Para obtenerla se debe de resolver la parte lógica del problema para encontrar las soluciones candidatas. Las soluciones candidatas son determinadas por lo descrito en la definición 3.13.

Entonces considerando la parte lógica para el PLP. De las reglas r_1 a r_4 se obtienen las siguientes soluciones candidatas:

$$
\begin{aligned}
M_1 &: \quad \{noTemporada,\ hogar\}. \\
M_2 &: \quad \{noTemporada,\ oficina\}. \\
M_3 &: \quad \{temporada,\ defectuoso,\ oficina,\ escaso\}.
\end{aligned}
$$

Teniendo a los M_i como los identificadores de los *answer sets* del conjunto de soluciones candidatas. De las soluciones candidatas se busca la solución preferida. La solución preferida se tiene de comparar los grados de satisfacción. Los grados de satisfacción de las reglas para los *answer sets*, de acuerdo a la definición 3.14, se tienen definidos como:

$$deg_{M_1}(r_7) = 1 \qquad deg_{M_1}(r_8) = 2 \qquad deg_{M_1}(r_9) = 1$$
$$deg_{M_2}(r_7) = 1 \qquad deg_{M_2}(r_8) = 3 \qquad deg_{M_2}(r_9) = 1$$
$$deg_{M_3}(r_7) = 2 \qquad deg_{M_3}(r_8) = 1 \qquad deg_{M_3}(r_9) = 2$$

Con el grado de satisfacción de cada regla de preferencia se tiene un conjunto de reglas de preferencia que tienen el mismo grado. Estos conjuntos están formados como:

$$S^1_{M_1} = \{r_7, r_9\} \qquad S^2_{M_1} = \{r_8\} \qquad S^3_{M_1} = \{\ \}$$
$$S^1_{M_2} = \{r_7, r_9\} \qquad S^2_{M_2} = \{\ \} \qquad S^3_{M_2} = \{r_8\}$$
$$S^1_{M_3} = \{r_8\} \qquad S^2_{M_3} = \{r_7, r_9\} \qquad S^3_{M_3} = \{\ \}$$

Con los conjuntos anteriores se encuentra el *answer set* preferido. El *answer set* preferido se determina a partir de usar un criterio de preferencia. A continuación se obtiene la solución preferida por medio de los criterios con los que trabajan los PLPs.

Preferencia por Inclusión

De acuerdo a lo que se tiene en la definición 3.15 para el criterio de preferencia por inclusión se tiene la Tabla B.33 con resultados de comparación. De la Tabla B.33 y por

	M_1	M_2	M_3
M_1	$=$	$>$	$=$
M_2	$<$	$=$	$=$
M_3	$=$	$=$	$=$

Tabla B.33: *Problema pc*: PLP Preferencia por Inclusión.

la definición 3.16 se tiene que M_1 y M_3 son *answer sets* preferidos por inclusión.

Preferencia por Cardinalidad

De acuerdo a lo que se tiene en la definición 3.17 para el criterio de preferencia por cardinalidad, se tienen los siguientes resultados de comparación, mostrados en la Tabla B.34.

	M_1	M_2	M_3
M_1	$=$	$>$	$>$
M_2	$<$	$=$	$>$
M_3	$<$	$<$	$=$

Tabla B.34: *Problema pc*: PLP Preferencia por Cardinalidad.

De la información de Tabla B.34 y por la definición 3.18 se tiene que M_1 es un *answer set* preferido por cardinalidad, siendo la solución preferida para el problema.

B.5.6. Solución por ASOPs

Para el problema B.5 de Andrea, el modelo de la sección B.5.2 y de acuerdo a las definiciones 3.19 y 3.22 se tienen las siguientes reglas que modelan el problema como un ASOP:

$$\begin{aligned}
r_1 &: \ hogar \vee oficina \vee personal \leftarrow . \\
r_2 &: \ noTemporada \vee temporada \leftarrow . \\
r_3 &: \ defectuoso \leftarrow temporada. \\
r_4 &: \ escaso \leftarrow temporada, oficina. \\
r_5 &: \ \leftarrow personal. \\
r_6 &: \ \leftarrow hogar, temporada. \\
r_7 &: \ hogar > oficina > personal \leftarrow temporada. \\
r_8 &: \ personal > hogar > oficina \leftarrow noTemporada. \\
r_9 &: \ noTemporada > temporada \leftarrow .
\end{aligned}$$

Teniendo que los r_i son los identificadores para las reglas. P_{gen} está formado por las reglas de r_1 a r_6 y P_{pref} por el conjunto $\{r_7, r_8, r_9\}$.

Con el modelado de un problema como ASOP se busca tener una solución preferida sobre un conjunto de soluciones candidatas. El conjunto de soluciones candidatas se forma por los *answer sets* generados de P_{gen}.

A continuación se tienen las soluciones generadas de P_{gen}:

$$\begin{aligned} S_1 &: \ \{noTemporada, hogar\}. \\ S_2 &: \ \{noTemporada, oficina\}. \\ S_3 &: \ \{temporada, defectuoso, oficina, escaso\}. \end{aligned}$$

Teniendo que los S_i son los identificadores de los *answer sets*.

Con los *answer sets* obtenidos se determina el grado de satisfacción. De acuerdo a la definición 3.23 se tiene que para los *answer sets*, el grado de satisfacción está dado como:

$$\begin{array}{lll} v_{S_1}(r_7) = I & v_{S_1}(r_8) = 2 & v_{S_1}(r_9) = 1 \\ v_{S_2}(r_7) = I & v_{S_2}(r_8) = 3 & v_{S_2}(r_9) = 1 \\ v_{S_3}(r_7) = 2 & v_{S_3}(r_8) = I & v_{S_3}(r_9) = 2 \end{array}$$

Con el grado de satisfación se forma el vector de satisfacción. De acuerdo a los grados de satisfacción para las reglas y la definición 3.24 se forman los siguientes vectores de satisfacción para los *answer sets*:

$$\begin{aligned} V_{S_1} &: \ (I, 2, 1). \\ V_{S_2} &: \ (I, 3, 1). \\ V_{S_3} &: \ (2, I, 2). \end{aligned}$$

Con los vectores de satisfacción, se determina la solución preferida. La solución preferida se obtiene de comparar los vectores de satisfacción de los *answer sets* de acuerdo a la definición 3.25 se tienen los resultados mostrados en la Tabla B.35.

	S_1	S_2	S_3
S_1	$=$	$>$	$=$
S_2	$<$	$=$	$=$
S_3	$=$	$=$	$=$

Tabla B.35: *Problema pc*: ASOP.

De la Tabla B.35 y por la definición 3.26 se tiene que S_1 y S_3 son las soluciones preferidas para el problema.

B.5.7. Comparación de Resultados

Las Soluciones candidatas del problema de la sección B.5.1, están definidas como:

1. {*noTemporada, hogar*}.

2. {*noTemporada, oficina*}.

3. {*temporada, defectuoso, oficina, escaso*}.

Con las opciones anteriores al aplicarse cada uno de los enfoques vistos se tiene la Tabla B.36. La solución preferida para este problema es la opción {*noTemporada,*

Enfoque	**Criterio**	**Solución Preferida**
Sentido Común		1, 2, 3
LPOD	Inclusión	1,3
	Cardinalidad	1
	Pareto	1,3
PLP	inclusión	1, 3
	Cardinalidad	1
ASOP		1,3

Tabla B.36: *Problema pc*. Comparación de soluciones preferidas.

hogar}.

Bibliografía

[1] Varol Akman, Selim Erdogan, Joohyung Lee, Vladimir Lifschitz, y Hudson Turner. Representing the zoo world and the traffic world in the language of the casual calculator. *Artificial Intelligence*, 2004.

[2] Allen L. Ambler, Maragaret M. Burnett, y Betsy A. Zimmerman. Operational versus definitional: A perspective on programming paradigms. *Computer*, 25(9):28–43, Septiembre 1992.

[3] Marcello Balduccini y Michael Gelfond. Logic programs with consistency-restoring rules. En Patrick Doherty, John McCarthy, y Mary-Anne Williams, editores, *Working Papers of the 2003 AAAI Spring Symposium on Logical Formalization of Commonsense Reasoning*, páginas 9–18. AAAI Press, Menlo Park, California, 2003.

[4] Marcello Balduccini y Veena S. Mellarkod. A-prolog with cr-rules and ordered disjunction. En *International Conference on Intelligent Sensing and Information Processing*, 2004.

[5] Chitta Baral. *Knowledge Representation, reasoning and declarative problem solving with Answer Sets*. Cambridge University Press, Cambridge, 2003.

[6] Chitta Baral y Michael Gelfond. Logic programming and knowledge representation. *Journal of Logic Programming*, 1994.

[7] Gianni Bosi, Ronen I. Brafman, Jan Chomicki, y Werner Kiešling. 04271 abstracts collection – preferences: Specification, inference, applications. En Gianni Bosi, Ronen I. Brafman, Jan Chomicki, y Werner Kiešling, editores, *Preferences: Specification, Inference, Applications*, número 04271 en Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2006.

[8] Ronen I. Brafman y Carmel Domshlak. Reasoning and constrained optimization with cp-networks (tutorial). En *9th International Conference on Principles and Practice of Constraints Programming*, Kinsale, Ireland, 2003.

[9] Gerhard Brewka. Logic programming with ordered disjunction. En *18th National Conference on Artificial Intelligence*, páginas 100–105, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[10] Gerhard Brewka, Ilkka Niemelä, y Tommi Syrjänen. Implementing ordered disjunction using answer set solvers for normal programs. En *JELIA '02: Proceedings of the European Conference on Logics in Artificial Intelligence*, páginas 444–455, London, UK, 2002. Springer-Verlag.

[11] Gerhard Brewka, Ilkka Niemelä, y Mirosław Truszczyński. Answer set optimization. En *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, páginas 867–872. Morgan Kaufmann Publishers, August 2003.

[12] Michael Gelfond y Vladimir Lifschitz. The stable model semantics for logic programming. En Robert A. Kowalski y Kenneth Bowen, editores, *Proceedings of the Fifth International Conference on Logic Programming*, páginas 1070–1080, Cambridge, Massachusetts, 1988. The MIT Press.

[13] Michael Gelfond y Vladimir Lifschitz. Logic Program with Classical Negation. En David H. D. Warren y Peter Szeredi, editores, *Proceedings of the 7th Int. Conf. on Logic Programming*, páginas 579–597, Jerusalem, Israel, Junio 1990. MIT.

[14] Michael Gelfond y Vladimir Lifschitz. Classical negation in logic programs and disjunctive databases. volumen 9, páginas 365–386, 1991.

[15] Kannan Govindarajan, Bharat Jayaraman, y Surya Mantha. Preference logic programming. En *International Conference on Logic Programming*, páginas 731–745, 1995.

[16] Ulrich Junker. Preference-based problem solving for constraint programming. En Gianni Bosi, Ronen I. Brafman, Jan Chomicki, y Werner Kiešling, editores, *Preferences: Specification, Inference, Applications*, número 04271 en Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2006. <http://drops.dagstuhl.de/opus/volltexte/2006/399> [fecha de cita: 2006-01-01].

[17] Jérôme Lang. Logical representation of preference and nonmonotonic reasoning. En *10th International Workshop on Non-Monotonic Reasoning NMR2004*, Whister BC, Canada, 2004.

[18] Nicola Leone y Simona Perri. Parametric Connectives in Disjunctive Logic Programming. En *ASP03 Answer Set Programming: Advances in Theory and Implementation*, Messina, Sicily, Septiembre 2003.

[19] Elliott Mendelson. *Introduction to Mathematical Logic.* Wadsworth and Brooks/Cole Advanced Books & Software, Belmont, CA, tercera edición, 1987.

[20] Erik T. Mueller. *Commonsense Reasoning.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006.

[21] Mauricio Osorio, Juan Antonio Navarro, y José Arrazola. Applications of Intuitionistic Logic in Answer Set Programming. *Theory and Practice of Logic Programming (TPLP)*, 4(3):325–354, 2004.

[22] Mauricio Osorio y Magdalena Ortiz. Embedded Implications and Minimality in ASP. En Michael Hanus Ulrich Geske Dietmar Seipel y Oskar Bartenstein, editores, *15th International Conference on Applications of Declarative Programming and Knowledge Management. INAP 2004*, Postdam, Germany, Marzo 2004.

[23] Mauricio Osorio, Magdalena Ortiz, y Matilde Hernandez. Generalized Ordered Disjunctions and its Applicatons. Sin publicar. http://mail.udlap.mx/∼is103378/research/pubs/iclp/genOrdDisj.pdf, 2004.

[24] Mauricio Osorio y Claudia Zepeda. Answer set general theories and preferences. En *MICAI*, páginas 59–69, 2006.

[25] David Pearce. Stable Inference as Intuitionistic Validity. *Logic Programming*, 38(1):79–91, 1999.

[26] Tran Cao Son y Enrico Pontelli. Planning with preferences using logic programming. *Theory Pract. Log. Program.*, 6(5):559–607, 2006.

[27] Claudia Zepeda. *Evacuation Planning using Answer Sets.* Tesis Doctoral, Universidad de las Americas, Puebla and Institut National des Sciences Appliquées de Lyon, 2005.