

Veri Tabanlarında Bilgi Keşfine Formel Bir Yaklaşım: Kısım II- Eşleştirme Sorgularının Biçimsel Kavram Analizi ile Modellenmesi

*A Principled Approach to the Knowledge Discovery in
Databases: Part II- Modelling Association Queries by
Formal Concept Analysis*

Hayri Sever* ve Buket Oğuz**

Öz

Eşleştirme kurallarını modelleyebilmek için biçimsel kavram analizinden faydalanılmıştır. Biçimsel kavram analizi evreni, nesnelere ve özelliklerden oluşan topolojik bir yapı olarak görür. Nesnelere ve özellikler arasındaki ilişkiyi kullanarak kavram adı verilen birimi tanımlar. Bir kavram bir grup nesne ve özellikten oluşur. Kavramda yer alan özellikler, kavramda yer alan nesne grubu tarafından taşınan ortak özelliklerin en büyük kümesidir. Benzer biçimde nesnelere, kavramda yer alan tüm özellikleri taşıyan en büyük nesne kümesidir. Biçimsel kavram analizi kavramlar arasındaki ilişkileri incelemek ve kavram yapılarını kurmak için matematiğe dayalı biçimsel araç ve teknikleri kullanır. Bu çalışmada, eşleştirme kuralı çıkarımı ve biçimsel kavram analizi arasında bir bağlantı önerilmiş ve geliştirilmiştir. Bir eşleştirme sorgusu ile bulunan bağımlılıkların kavram yapısından elde edilebileceği gösterilmiştir. Biçimsel kavram analizi çerçevesi, eşleştirme kuralı çıkarımını ele alabilecek biçimde genişletilmiştir. Bu genişletmeyi yapabilmek için

*Doç. Dr.; Başkent Üniversitesi, Bilgisayar Mühendisliği Bölümü 06530 Bağlıca-Ankara (sever@baskent.edu.tr).

**Y. Bilgisayar Müh.; AYESAŞ, ODTÜ-Teknokent 06531 Ankara (buketo@ayesas.com.tr).

eşleştirme sorgularının özel bir biçimi olan market sepeti problemi kullanılmıştır. Bu model, eşleştirme kuralı çıkarımı algoritmalarının karmaşıklıklarını ele alabilmek için bir temel sağlamıştır. Ayrıca, model diğer veri madenciliği problemleri için birleştirilmiş bir çerçeve oluşturmaya yardımcı olabilir.

Anahtar sözcükler: *Biçimsel kavram analizi, Eşleştirme sorguları, Bağımlılık ilişkileri, Kavram yapıları.*

Abstract

In this study we utilize formal concept analysis to model association rules. Formal concept analysis provides a topological structure for a universe of objects and attributes. By exploiting the relationship between objects and attributes, formal concept analysis then introduces an entity called a concept. A concept is a set of attributes and objects. The attributes are maximally possessed by the set of objects and similarly the objects are the maximal set which all possess the set of attributes. Formal concept analysis deals with formal mathematical tools and techniques to develop and analyze relationship between concepts and to develop concept structures. We propose and develop a connection between association rule mining and formal concept analysis. We show that dependencies found by an association query can be derived from a concept structure. We have extended formal concept analysis framework to the association rule mining. We use analysis of market-basket problem, a specific case of association rule mining, to achieve this extension. This extension provides a natural basis for complexity analysis of the association rule mining. This extension can also help in developing a unified framework for common data mining problems.

Keywords: *Formal concept analysis, Association query, Dependency relationships, Concept structures.*

Giriş

Son yirmi yıldır, veriyi toplama ve saklama kapasitesindeki çok ani büyüme şahit olmaktayız. Öyle ki, bir bilgisayarın işleyebileceği veriden daha fazlası üretilmektedir. Gerçekte bu durum, dünyadaki bilgi miktarının her 20 ayda bir ikiye katlandığı varsayımı ile uygunluk arz etmektedir. Veri biriktirilmesi ile eş zamanlı olarak onu yorumlamadaki ve özümsemedeki beceriksizliğimiz, özdevimli ve akıllı veri tabanı analizi için, yeni nesil araçlarına ve tekniklerine olan ihtiyacı doğurdu. Sonuç olarak, büyük gerçek-dünya veri tabanlarından değerli, ilginç ve önceden bilinmeyen bilgiyi keşfetme (veya çıkarma) problemi ile eşleştirilen pratik uygulamalar ve olası çözümlerin kuramsal zorlukları nedeni ile, veri tabanlarında bilgi keşfi (VTBK) önemli ve aktif bir araştırma alanına evrimleşti. Veri tabanı sistemleri, makine öğrenimi, akıllı bilgi sistemleri, istatistik ve uzman sistemler gibi birbirleri ile yakından ilişkili alanlarca VTBK'nın birçok yönü incelendi. Çalışmanın ilk kısmında, VTBK'yı süreç esaslı bakış açısı ile incelemiş ve özelinde, eşleştirme sorguları için çözüm çerçevesi sunmuştu (Sever ve Oğuz, 2002). Bu makalede ise, biçimsel kavram analizi aracılığı ile eşleştirme kuralları modelleneyecektir.

Biçimsel Kavram Analizi (BKA) son yıllarda ilgi çeken araştırma konuları arasında yer almaktadır, Rudolf Wille (1982) tarafından 1980'li yıllarda kafes teorisinin genişletilmesiyle ortaya çıkmıştır. Kavram, matematiksel nosyon olarak kökünü biçimsel mantıktan almaktadır. Bununla birlikte kavram çeşitli disiplinlerde genel bir mekanizma olarak karşımıza çıkmıştır. Bu genel tanım *kapsam (extent)* ve *içerik (intent)* olmak üzere iki türlü yapılabilir. İçerik kavramın özelliklerini, kapsam ise kavramda yer alan nesnelere verir. Nesnelere taşıdıkları özelliklere göre gruplanmasına *kavramlaştırma* denir. Bilgisayar makineleri uzayında, örneğin, kişisel bilgisayarlar kavramı makine kapasitesi özelliğine göre elde edilir. BKA, kavramları verilen bir bağlam içinde tanımlar ve kavramlar arası kesin ilişkiyi, bağlama karşılık gelen kafes yapısını kullanarak inceler. Biçimsel olarak bağlam, nesnelere (G), özellikler (M) ve nesnelere ile özellikler arasındaki ilişkiden oluşan (I) üçlü cebirsel bir yapıyla (G,M,I) ifade edilir (Wille, 1982; Davey ve Priestley, 1990).

Literatürde eşleştirme kuralı çıkarımı problemine ilişkin yapılan çalışmaların çoğu etkin bir algoritma geliştirmeye çalışmış, problemi matematiksel bir temele oturtmamıştır (Agrawal, Imielinski ve Swami, 1993; Agrawal ve

Srikant, 1994; Park, Chen ve Yu, 1995; Savasere, Omiecinski ve Navathe, 1995). Bu makalede, eşleştirme kuralı çıkarımı problemi için biçimsel bir çerçevede ifade edilerek, problemin matematiksel temeli tanımlanmıştır. Eşleştirme sorgusu probleminin kuramsal temeli biçimsel kavram analizi konusuna dayandırılmıştır. Eşleştirme sorgusu sonucu bulunan bağımlılıkların kavram yapısı kullanılarak elde edilebileceği kuramsal olarak gösterilmiştir.

Makale kapsamında eşleştirme sorguları, eşleştirme sorgularının özel bir biçimi olan market sepeti problemi kullanılarak, BKA ile modellenmiştir. Market sepeti problemi, müşterilerin satın alma alışkanlıklarını analiz etmek için kullanılır. Burada, girdi olarak süpermarketlerde oluşan satış hareketleri kullanılır. Tipik bir satış hareketinde, hareket numarası, ürün kodu, ürün miktarı ve fiyatı bilgileri yer alır. Bu çalışmada, satış hareketi ilişkisinden kafes yapısına dönüştürme kuralları tanımlanmıştır. Kafes yapısındaki kavramlar, satış hareketi verilerinin hareket numaralarına göre gruplandırılmasıyla oluşturulan uzayda, ürünler arasındaki kapsama ilişkisinden yola çıkılarak tanımlanmıştır. Bu ilginç yaklaşımla oluşturulan kavramlar (ürün grupları) arasındaki olası (tam olmayan) ilişkileri incelemede, güvenilirlik ve destek parametreleri kullanılmıştır. Eşleştirme kuralı üretimine taban teşkil eden bu parametrelerin hesaplanmasına yönelik kurallar ortaya konmuş ve doğrulanmıştır. Ayrıca, arama uzayını sınırlandıran kurallar oluşturulmuş ve bu kuralların herhangi bir eşleştirme modeline indirgenebileceği hipotezi araştırılmıştır ki, bu yönden ele alınan çalışmamızın eşleştirme modeline sezgisel yaklaşımlar yerine biçimsel bir çerçeve kazandıracığı savındayız.

Tanımlanan kuralları içeren algoritmalar biçimsel kavram analizi çerçevesinde ele alınarak gerçekleştirilmiştir. Bu algoritmalar, eşleştirme kurallarını verilen parametrelere göre çıkarmaktadır. Geliştirilen BKA modelinde, artık eşleştirmeleri sık geçen öge kümelerinin görüşü kullanılarak budamak mümkündür. Bu işlem, dolaylı olarak daraltılmış arama uzayındaki ilginç örüntüleri bulmamıza imkân vermektedir.

Makalede önce biçimsel kavram analizi tartışılmakta ve BKA notasyonu bileşik ve/veya tanımlanamaz kavramları kapsayacak şekilde genişletilmektedir. Sonraki bölümde, eşleştirme kurallarının kafese yerleştirilmesi ve görüşler aracılığı ile kavramlar arası ilişkilerin nasıl incelenebileceği tartışılmaktadır. Ek-1'de ise, kafes için matematiksel ön tanımlar verilmektedir.

1. Biçimsel Kavram Analizi (BKA)

BKA¹ dünyayı nesnelere ve niteliklere olarak modelleyen bir bilgi gösterim biçimidir. Kavramın ne olduğu sorusu matematikçilerden ziyade filozofları ilgilendirmiştir. Gerçekte felsefenin kavram tanımı, yapılacak olan kavramın biçimsel tanımına temel sağlar. Bir kavram, kavrama ait içerik ve kapsam değerleriyle tanımlanır. Kapsam, kavramın tanımlı olduğu bağlamdaki tüm nesnelere içerir. Örneğin okuyucular, canlı insan kavramının kapsamıdır. İçerik ise nesnelere paylaştıkları nitelikleri içerir. Tüm canlı insanlar “nefes alabilme” özelliğine sahiptir. Genellikle, bir kavrama ait olan tüm nesnelere ve bu nesnelere taşıdıkları özellikleri sıralamak güçtür. Bu yüzden nesne ve nitelik sayısının sabit olduğu belirli bir bağlamda çalışmak doğaldır.

Örnek 1. (Güneş Sistemi Bağlamı): Çizelge 1’de güneş sistemimizdeki gezegenler ve bu gezegenlerin bazı nitelikleri gösterilmektedir. Güneş sistemi bağlamında, nesnelere gezegenleri; nitelikler ise gezegenlerin büyüklüğünü (küçük, orta, büyük), gezegenin uydusunun var olup olmadığını (var, yok) ve güneşe uzaklığını (yakın, uzak) ifade etmektedir.

Güneş sistemi bağlamından kavram elde edebilmek için şu yol izlenmelidir: Bir nesne alınır (örneğin Dünya), seçilen nesnenin taşıdığı özellikler kümesi (örneğin *B* kümesi), *B* kümesindeki niteliklere taşıyan nesnelere kümesi (örneğin *A* kümesi) belirlenir. Böylece *A*, *B* kümeleri (*A, B*) kavramını oluşturur. Bu işlem başlangıçta bir nesne seçmek yerine bir grup nesne seçilerek de yapılabilir. Benzer biçimde başlangıçta bir nitelik veya nitelik kümesi seçilip, seçilen nitelik/nitelik kümesini taşıyan nesnelere de belirlenilir.

$B = \{\text{büyüklük-küçük, uzaklık-yakın, uydu-var}\}$

$A = \{\text{Dünya, Mars}\}$

¹BKA ile ilgili formel tanımlar ve temel kuram Ek-1’de verilmiştir. Bundan dolayı kullanılan jargonun tanımları için, okuyucu gerek görürse Ek-1’e göz atabilir. Bu bölümde, BKA, daha akıcı bir dille ve koşan-örnek çerçevesinde ifade edilecektir.

	Büyüklik			Güneşe Olan Uzaklık		Uydu	
	Küçük	Orta	Büyük	Yakın	Uzak	Var	Yok
Merkür	X			X			X
Venüs	X			X			X
Dünya	X			X		X	
Mars	X			X		X	
Jüpiter			X		X	X	
Satürn			X		X	X	
Uranüs		X			X	X	
Neptün		X			X	X	
Pluton	X				X	X	

Çizelge 1: Güneş Sistemi Bağlamı

Bu bağlamdan elde edilebilecek diğer kavramlar şöyledir:

{Merkür, Venüs}, {büyüklik-küçük, uzaklık-yakın, uydu-yok}

{Jüpiter, Satürn}, {büyüklik-büyük, uzaklık-uzak, uydu-var}

{Uranüs, Neptün}, {büyüklik-orta, uzaklık-uzak, uydu-var}

{Pluton}, {büyüklik-küçük, uzaklık-uzak, uydu-var}

Yukarıdaki örnekten de anlaşılacağı gibi, bağlamı, üç çoklu oluşturur: Nesnelere kümesi (G), nitelikler kümesi (M) ve bu iki küme arasındaki ilişki (I). Kavram ise, niteliklerce tanımlanan özellikleri ortak olarak paylaşan maksimal nesnelere kümesi; ya da başka bir deyişle, nesnelerce ortaklaşa belirtilen maksimal nitelikler kümesi olarak tanımlanabilir.

Tanım 0: Bağlam, nesnelere kümesi (G), nitelikler kümesi (M) ve bu iki kümenin kartezyen çarpımında tanımlanmış I ikili ilişkisinden ($I \subseteq G \times M$) oluşan (G, M, I) cebirsel üçlüsüyle ifade edilir. g nesnesi ve m niteliği için, $(g, m) \in I$ gösterimi yerine glm gösterimi kullanılır ve glm "g nesnesi m niteliğini taşır" anlamına gelir.

Tanım 1: (G, M, I) bağlamında, A kümesi bağlamda yer alan nesnelerin bir alt kümesi ($A \subseteq G$) ve B kümesi bağlamda yer alan niteliklerin bir alt kümesi ($B \subseteq M$) olmak üzere,

$$A' = \{m \in M \mid (\forall g \in A)g \text{ Im}\},$$

$$B' = \{g \in G \mid (\forall m \in B)g \text{ Im}\},$$

A' , A kümesinde yer alan tüm nesnelerin taşıdıkları maksimal ortak özellikler kümesidir. B' ise, B kümesinde yer alan tüm nitelikleri taşıyan maksimal nesnelere kümesidir.

Tanım 2: (G, M, I) bağlamında kavram, $A \subseteq G$ ve $B \subseteq M$ olmak üzere $A=B'$ ve $B=A'$ koşulunu sağlayan (A, B) ikilisi ile tanımlanır. (A, B) kavramında A kümesi (A, B) kavramının *kapsamı*, B kümesine ise *içeriği* denir.

(G, M, I) bağlamından elde edilebilecek tüm kavramlar kümesi $B(G, M, I)$ ile gösterilsin. $B(G, M, I)$ üzerinde bir sıralama ilişkisi şu şekilde tanımlanır: (A_1, B_1) ve (A_2, B_2) kavramları $B(G, M, I)$ yer alan iki kavram olsun.

$(A_1, B_1) \leq (A_2, B_2)$ ancak ve ancak $A_1 \subseteq A_2$ (eşdeğer biçimde $B_2 \subseteq B_1$). Bu durumda (A_1, B_1) kavramı (A_2, B_2) kavramının alt kavramı veya (A_2, B_2) kavramı (A_1, B_1) kavramının üst kavramıdır denir. G ve M kapalı kümeleri arasındaki dönüşümlerin anti-isomorfik² olduğu görülmektedir. Öteki deyişle, kavramda yer alan nesne sayısı arttıkça, nitelik sayısı azalır (büyük kapsama sahip kavramlar küçük içeriğe sahip olur). Alt kavram/üst kavram kısmi sıralamasına dayanarak kavram sıradüzeni oluşturulur. $\langle B(G, M, I); \leq \rangle$ poseti (ya da kısmi sıralı küme, bkz. Ek 1) bir tam kafestir. (G, M, I) bağlamının kavram kafesi olarak adlandırılır.

Teorem 1: Kavram kafesleri için temel teorem:

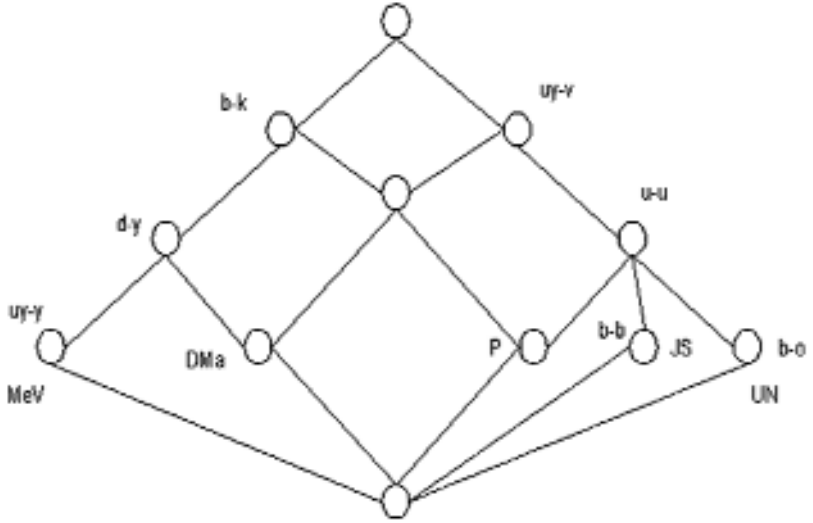
(G, M, I) bağlamında $\langle B(G, M, I); \leq \rangle$ aşağıda tanımlı supremum (*join*) ve infimum (*meet*) eşitliklerini gerçekleyen tam bir kafestir.

$$\bigvee_{\mathcal{K}} (A_i, B_i) = ((\bigcup_{\mathcal{K}} A_i)', \bigcap_{\mathcal{K}} B_i)$$

$$\bigwedge_{\mathcal{K}} (A_i, B_i) = (\bigcap_{\mathcal{K}} A_i, (\bigcup_{\mathcal{K}} B_i)')$$

²Anti-isomorfik terimi eş yapılı olmayan anlamındadır.

Bu teorem bir bağlamdan üretilen kavramların bir kafes oluşturduğunu ve kafeste yer alan bir kavramın supremum ve infimumu yukarıda verilen eşitlik ile hesaplanacağını söyler. BKA'nın cebirsel alt yapısı kafes kuramına dayanır. BKA, kavram adı verilen elemanları arasındaki sıralama bağıntısını kullanarak kafes elde eder. Elde edilen kafes nitelikler arasındaki tüm ilişkileri ifade edebilecek güce sahiptir. Güneş sistemi bağlamından türetilen kafes Şekil 1'de verilmiştir.



Şekil 1: Çizelge 1'deki Güneş Sistemi Bağlamı için Kavram Kafesi³

³Güneş Sistemi Bağlamından elde edilen kafeste, her düğüm kenarında düğümle ilişkin etiketler yer almaktadır. Çizimin karmaşıklığının artmaması için etiketler kısaltılmıştır. Büyük harfler gezegenlerin ilk harfini, küçük harfler ise nitelik adının ilk harfi-nitelik değerini temsil etmektedir. Örneğin D dünya b-b ise büyüklük büyük anlamına gelmektedir.

Wille'nin yaptığı kavram tanımı Şekil 2'deki gibi genişletilebilir. Burada, atomik kavramı Wille'nin tanımına tekabül eder (bkz. Tanım 2). (G, M, I) bağlamında G ve M kümelerinin her alt kümesi bir kavrama denk gelmez. Kavram tanımlanmasına yol açan alt kümelere *olurlu kümeler* adı verilir. G kümesinin bir A alt kümesinin herhangi bir kavramın kapsamı olabilmesi için $A' = A$ eşitliğini gerçeklemesi gerekir. Bu durumda A kümesi (A, A') biricik kavramının (*unique concept*) kapsamı olur. Benzer biçimde, M kümesinin bir B alt kümesinin herhangi bir kavramın içeriği olabilmesi için $B' = B$ eşitliğini gerçeklemesi gerekir. Bu durumda B kümesi (B', B) biricik kavramının içeriği olur.

Tanımlanabilir/Bileşik Kavramlar

Tanımlanabilir kavramlar, Wille'nin kavram tanımının genelleştirilmiş biçimidir. Gerçek hayatta kullanılan pek çok uygulama bileşik kavramların kullanımını gerektirir.

(G, M, I) bağlamında bir kavram tanımlayabilecek her özellik kümesini $(B_j \subseteq M)$, S_j mantıksal değişkeni ile ifade edilsin. Örneğin $B_j = \{a, b, c\}$ ise $S_j = (a \wedge b \wedge c)$ veya basitçe $S_j = (a, b, c)$ olarak gösterilir. M özellikler kümesi üzerinde kavram tanımına yol açabilecek olurlu kümelerin tamamı M ile gösterilsin.

$$M = \{S_j \mid S_j = S(B_j), B_j \subseteq M \text{ olurlu kümeleri} \}$$

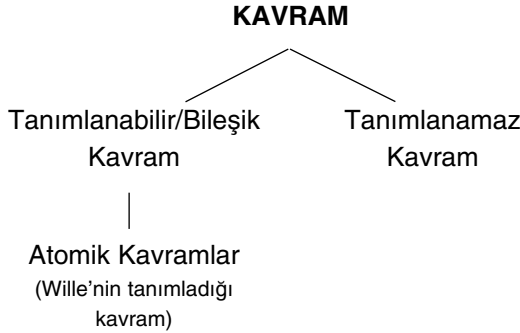
S_j ve S_k değişkenleri eğer her ikisi de birbirini kapsamıyorsa görelî asal değişkenlerdir. $F \subseteq M$ kümesi şu şekilde tanımlanır:

$$S = \{ \{ S_1, S_2, \dots, S_k \} \mid S_i \in M, 1 \leq i \leq k, \text{ ve } k \geq 1 \}$$

Tanım 3: (G, M, I) bağlamında tanımlanabilir/bileşik kavram $A \subseteq G$ ve $F \subseteq M$ ve F kümesi içinde yer alan değişkenler görelî asal olmak üzere koşulunu sağlayan (A, F) ikilisi ile tanımlanır. (A, F) bileşik kavramında A kümesi (A, F) bileşik kavramının *kapsamı*, F kümesine ise *içeriği* denir.

Tanımlanamaz Kavramlar

Görüntü ve belge erişimi gibi pek çok uygulamada, nesnelere taşıdıkları özellikler kesin değildir. Örneğin, aynı belge kümesi içinde bir kullanıcı ile ilgili olan belgelerle diğer bir kullanıcı ile ilgili olan belgeler kümesi, her iki kullanıcı aynı konular hakkında araştırma yapıyor olsa bile tamamen örtüşmeyebilir. Bu yüzden kullanıcı olurlu olmayan bir nesnelere kümesinin özelliklerini sorgulamak isteyebilir. Yani, kullanıcının sorgusuna mevcut kavram sıradüzeninde cevap vermek mümkün olmayabilir. Bu durumda bir kavram çifti elde edilemez, buna tanımlanamaz kavram adı verilir. Tanımlanamaz kavram terimi ilk olarak kaba küme teorisi kullanılarak ifade edilmiştir⁴.



Şekil 2: Kavram Çeşitleri

2. Kafes Kuramına Dayalı Olarak Eşleştirme Kurallarının Üretilmesi

Eşleştirme sorguları, bir ilişkide bir niteliğin aldığı değerler arasındaki bağımlılıkları, anahtarlar yer almayan diğer niteliklere göre gruplama yapılmış verileri kullanarak bulur. Eşleştirme sorgularının özel bir durumu olan market-sepet ilişkisi analizi, örneğin bir firmada, müşterilerin satın alma eğilimi hakkında bilgi verebilir.

⁴ Kaba küme teorisi Pawlak tarafından ortaya atılmış, veri bağımlılıkları, özellik seçimi ve tümevarımsal sınıflama gibi pek çok alanda kullanılmıştır (Pawlak, 1984).

Örnek 2: Ürünler ile satış hareketleri arasında, satın alınan ürün miktarını da taşıyan bir çoğa bir çok $r(T,P,Q)$ ilişkisi tanımlansın. Bu ilişkide yer alan nitelikler sırasıyla şöyledir: T , hareket numarası; P , ürün kodu; Q , miktar. r ilişkisinde anahtar nitelikler T , P nitelikleridir.

$$r = \{ \begin{array}{l} <100,1,3>, <100,2,1>, <100,4,5>, <200,1,1>, <200,4,1>, \\ <200,3,1>, <300,1,1>, <300,2,1>, <300,4,4>, <300,5,2>, \\ <400,2,3>, <400,5,3>, <400,6,2>, <400,7,2>, <400,8,7>, \\ <500,1,6>, <500,2,3>, <500,7,4>, <600,2,1>, <600,4,1>, \\ <600,7,2>, <600,1,3> \end{array} \}$$

Birlikte satın alınan ürünler hangileridir sorgusu, r ilişkisinde T niteliğine göre gruplama yapılarak, P niteliği seçilerek bulunur. Elde edilen bu yeni ilişki r_1 olsun. r_1 ilişkisi `SELECT P' FROM r GROUP BY T AS r1(P')` deyimi ile sağlanır. Burada, SQL'in bir ilişkide çok değerli niteliklere izin verdiği varsayılmıştır.⁵ Buna göre:

$$r_1(P') = \{ <1,2,4> <1,4,3> <1,2,4,5> <2,5,6,7,8> <1,2,7> <1,2,4,7> \}.$$

Bu biçimdeki kümeler market-sepet verisidir. $\mathfrak{S}(P)$, P ürünler kümesinin güç kümesini (ya da ürün ailesini) tanımlasın. Market sepeti probleminde amaç, verilen minimum *güvenirlilik* (c) ve *destek* (s) kriterlerini sağlayan ilginç eşleştirmeleri, $R \subseteq \mathfrak{S}(P) \times \mathfrak{S}(P)$, bulmaktır. x ve $y \in \mathfrak{S}(P)$ olmak üzere $\langle xRy, c, s \rangle$ ifadesinde y , x 'e c güvenirlilik değeri, ve s destek değeri ile bağımlıdır (ilişkilidir). $(x, y) \in R$ olmak üzere güvenirlilik değeri y ürünün x ürününün bulunduğu hareketlerde yer alma olasılığıdır; $c = Pr(y | x)$ ile ifade edilebilir. Destek değeri ise x ve y ürününün aynı hareket tutanağında yer alma olasılığıdır; $s = Pr(x \cup y)$ 'dir.

⁵ Eşleştirme ve ardışık sorgular gibi bilgi keşfi sorgularını yanıtlayacak biçimde SQL'in genişletilmesi Meo, Psaila ve Ceri (1996) tarafından ele alınmıştır.

Güvenlik ve destek değerleri hesabı yukarıda tanımlanmış $r_1(P')$ ilişkisi kullanılarak $x=\{1\}$, $y=\{2\}$ ürünleri için şöyle hesaplanır: $Pr(x)$, x kümesi içinde yer alan tüm ürünlerin herhangi bir sepet varlığında bulunma olasılığı olsun. Buna göre $c(xRy)=Pr(2|1)=4/5=0.8$ ve $s(xRy)=Pr(1\cup 2)=4/6=0.66$ ⁶. Hareket numarasına göre gruplandırılmış market sepeti verisi Çizelge 2'de verilmiştir.

Market sepeti problemi, tutanakları 1'ler ve 0'lardan oluşan bir Boolean ilişkiye dönüştürülebilir (Meo, Psaila ve Ceri, 1996). Ürün sayısı n , $r_1(P')$ sepet ilişkisindeki tutanak sayısı m , $r_1(P')$ ilişkisi $dom(P)$ ile gösterilsin. Aşağıda verilen dönüştürüm işlemi kullanılarak market sepeti problemi Boolean bir ilişkiye çevrilebilir.

$$f : X \in \mathfrak{S}(P) \rightarrow \langle B_1, B_2, \dots, B_n \rangle \text{ öyleki } B_i = \begin{cases} i \in X \Rightarrow 1 \\ i \notin X \Rightarrow 0 \end{cases} \quad (3.1)$$

Hareket No	Ürün No	Miktar
100	1	3
	2	1
	4	5
200	1	1
	4	1
	3	1
300	1	1
	2	1
	4	4
	5	2
400	2	3
	5	3
	6	2
	7	2
	8	7
500	1	6
	2	3
	7	4
600	1	3
	2	1
	4	1
	7	2

Çizelge 2: Hareket Numarasına göre Gruplandırılmış Sepet İlişkisi

⁶ Küme gösterimi tek elemanlı kümeler için kaldırılmıştır. Bu yüzden $Pr(\{1\}|\{2\})$ ifadesi $Pr(1|2)$ ifadesine indirgenmiştir. Ayrıca, olasılık hesaplamasına teşkil eden 'olay' açık olarak tanımlanmasa bile, onun $\mathfrak{S}(P)$ küme ailesinin herhangi bir elemanına karşılık geldiğine dikkat edilmelidir.

Kısım 2'de tanımlı f fonksiyonu, n adet niteliğe sahip ve her bir niteliğin değeri eğer i ürünü X' in bir elemanı ise 1, değilse 0 olacak biçimde bir tutanak döndürür. Örneğin, $r_1(P')$ ilişkisinde yer alan $\langle 1,2,4 \rangle$ tutanağına f fonksiyonu uygulanırsa $f(\langle 1,2,4 \rangle) = \langle 1,1,0,1,0,0,0,0 \rangle$ sonucu elde edilir. Sonuç olarak f fonksiyonu $r_1(P')$ ilişkisini $r_2(B_1, B_2, \dots, B_n)$ ilişkisine çevirir. Elde edilen r_2 ilişkisi n adet nitelikten ve m adet tutanaktan oluşur.

Hareket No.	Ürün Kodu							
	1	2	3	4	5	6	7	8
100	X	X		X				
200	X		X	X				
300	X	X		X	X			
400		X			X	X	X	X
500	X	X					X	
600	X	X		X			X	

Çizelge 3: Market-Sepet İlişkisi için Bağlam

Bu biçimde eşleştirme problemi, verilen ilişkideki 1'ler arasındaki eşleştirmeleri bulma problemine dönüşür (Meo, Psaila ve Ceri, 1996) ve Boolean eşleştirme problemi adını alır. Görüldüğü gibi market-sepet ilişkisi, satış hareketleri nesnel kümesi (G), ürünler nitelikler kümesi (M) ve ürünün ilgili harekette satın alınıp/alınmadığı verisi I ilişkisi olmak üzere bir (G, M, I) bağlamı tanımlamak mümkündür. Çizelge 3'de satış hareketlerine ve ürün kodlarına göre oluşturulmuş bir bağlam örneği verilmiştir. Örnek 2'de verilen market-sepet verisi için $G = \{100, 200, 300, 400, 500, 600\}$ kümesi hareket numaralarını, $M = \{1, 2, 3, 4, 5, 6, 7, 8\}$ kümesi, her biri nitelik ismi olarak öngörülen ürün kodlarını içerir.

Genel olarak $i \neq j$ için $X_i R X_j$ ikili ilişkisi içinde yer alan $X_i, X_j \in \mathcal{S}(P)$ ler arasındaki (B_i 'ler) tüm eşleştirmelerin üretilmesi ile ilgilenilirse, r_2 bağlamındaki kavram kafesi kullanılabilir. Bu bağlama karşılık gelen kavram kafesi Şekil 3'te verilmiştir. Bu çizgedeki bağlantı önemli/önemsiz tüm eşleştirmeleri verir.⁷ Market-sepet ilişkisinden kavram kafesi elde edebilmek için aşağıda verilen tanımlar yapılmalıdır.

⁷Literatürde yapılan diğer çalışmalarda arama uzayı kontrollü olarak azaltılmaktadır. Bu makalenin yaklaşımı ise arama uzayını azaltmadan eşleştirme kurallarını üretmektedir.

Verilen bir bağlamda kavram kafesi $C = \langle V, E \rangle$ ⁸ yönsüz çizgesiyle modellenir. Çizgenin her bir düğümü nitelikler kümesinden oluşur. Ayrıca her bir düğümle geçiş sıklığı bilgisi eşleştirilir ki, bu bilgi basitçe düğümde yer alan niteliklerin, ilgili bağlam için oluşturulan sepet ilişkisindeki geçiş sıklığıdır. $v \in V$ düğümündeki nitelikleri döndüren bir $a: V \rightarrow \mathfrak{S}(P)$ fonksiyonu olsun. V düğümleri üzerinde J dizin kümesi tanımlansın. Ayrıca, her bir düğümle sıklık bilgisini $fr: V \rightarrow N^+$ fonksiyonu aracı ile ilişkilendirilsin. Tanımlanan fr fonksiyonu, verilen bir düğüm için Sepet ilişkisindeki niteliklerin sıklığını döndürür⁹. Çizgede bulunan her bir kenar iki düğüm arasındaki eşleştirmeyi verir. $(v_i, v_j) \in E$ ise v_i ve v_j düğümlerinin nitelikleri arasında bir R eşleştirmesi vardır. C çizgesi üzerinde R eşleştirme ilişkisi yansımali, geçişli, karşı simetrik bir kısmi sıra verir¹⁰ ($i, j \in J(a(v_i)Ra(v_j))$ ($a(v_i) R a(v_j)$)).

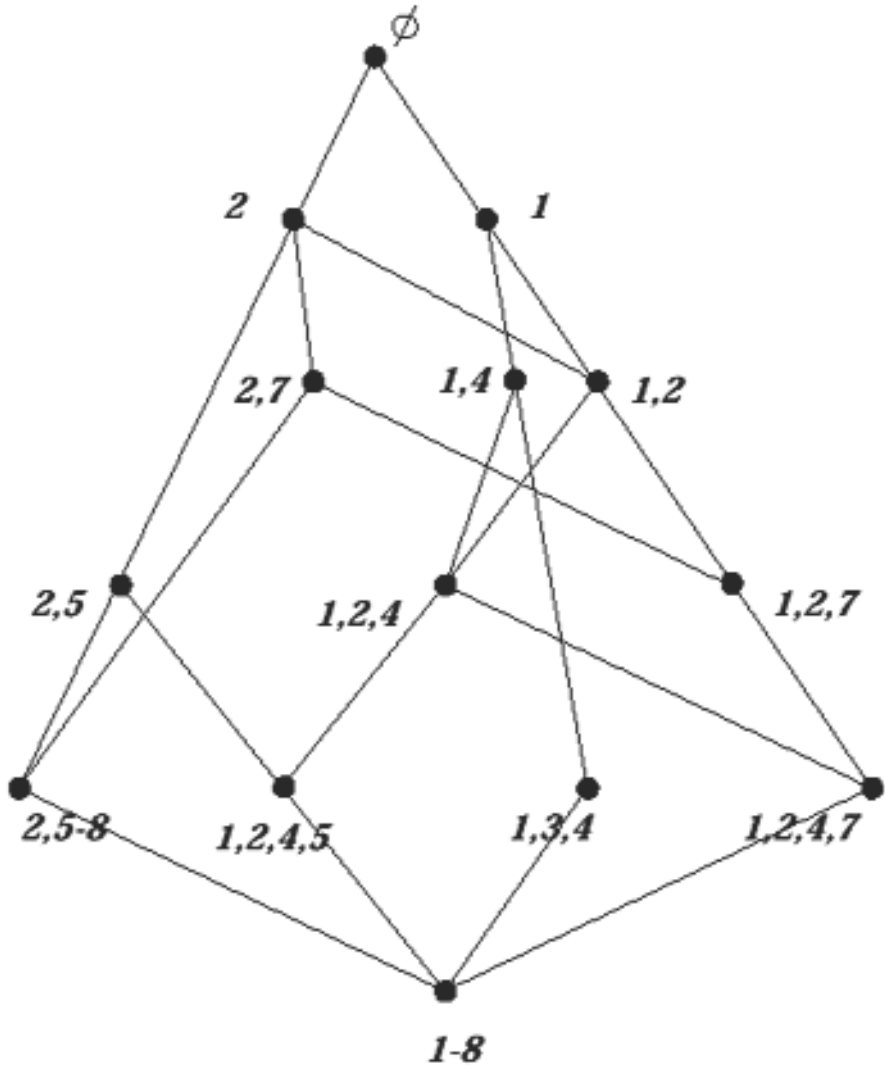
2.1. Kavram Kafesinin Oluşturulması

Verilen bağlamda $C = \langle V, E \rangle$ kavram kafesini oluşturmak için öncelikle V , E kümelerinin bulunması gerekir. Kavram kafesi oluşturma algoritması, v_i ile v_j arasında bir kenar varsa ya $a(v_i) \subset a(v_j)$ ya da $a(v_j) \subset a(v_i)$ fikrine dayanır. $C = \langle V, E \rangle$ çizgesinin tam kafes olması özelliği kullanılarak V düğümler kümesini elde etmek için; $r_1(P')$ ilişkisindeki her bir tutanak için bir düğüm yaratılır, tam kafes oluşana dek düğümler eklenir.

⁸ V :Düğümleri, E :Kenarları temsil etmektedir.

⁹ $\alpha = a(v_i)$ işlevi kullanılarak i . düğümde yer alan nitelikler kümesi elde edilir. Aynı biçimde $fr(v_j)$ kullanılarak düğümün sepet ilişkisinde geçiş sıklığı elde edilir. v_j düğümün geçiş sıklığı değeri için kimi zaman $fr(v_j)$ kimi zaman $fr(\alpha)$ kullanılacaktır.

¹⁰ R ilişkisi kısmi sıralı kümedir (Bkz. Ek-1).



Şekil 3: Market-Sepet Bağlamı için Kavram Kafesi

Şekil 4'te C çizgesinde yer alan düğümleri bulmak için kullanılan yaklaşım algoritmik olarak ifade edilmiştir.

```

DUGUMBUL( $r_1(P')$ ,  $V(C)$ ,  $J$ ) YORDAMI;
DISYAPILAR;
 $r_1(P')$ , GİRİŞ;
/*  $r_1(P')$  ilişkisindeki tutanaklar */
 $V(C)$ , ÇIKIŞ;
/*  $C$  çizgesini oluşturan düğümler kümesi */
 $J$  TAMSAYI, ÇIKIŞ;
/*  $V(C)$  üzerinde sınırları  $1-V(C)$  kümesinin eleman sayısına kadar değişen dizin */
ICYAPILAR;
KOMUTLAR;
1.  $r_1(P')$  ilişkisindeki tutanakları  $V(C)$  kümesine ekle;
2.  $\emptyset$ ,  $\{1,2,\dots,n\}$  kümelerini  $V(C)$  kümesine ekle;
3.  $J := |V(C)| + 2$ ;
4. ONCE YINELE, (  $V(C)$  kümesinde değişiklik yoksa ) ÇIK
5. EGER ( $i \neq j$ ) VE ( $a(v_i) \cap a(v_j) \notin V(C)$ ) İSE [
6.    $V(C) := V(C) \cup a(v_i) \cap a(v_j)$ 
7.    $J := J + 1$ ;
8. ]
DUGUMBUL BITTI

```

Şekil 4: Çizgedeki Düğümlerin Bulunması için Algoritma Kesiti

Örnek market-sepet verisi üzerinde Şekil 4'teki algoritma işletilirse Şekil 3'teki kavram kafesi elde edilir. İlk satırda $r_1(P')$ ilişkisindeki her bir tutanak çizgeye düğüm olarak eklenir. İkinci satırda \emptyset ve tüm ürün kodlarını içeren $\{1,2,\dots,8\}$ kümesi çizgeye düğüm olarak eklenir. Bu durumda çizgede şu düğümler yer alır: $\{1,2,4\}$, $\{1,4,3\}$, $\{1,2,4,5\}$, $\{2,5,6,7,8\}$, $\{1,2,7\}$, $\{1,2,4,7\}$, \emptyset , $\{1,2,\dots,8\}$. Dördüncü satır işletildiğinde $\{1,4\}$, $\{1,2\}$, $\{2\}$, $\{1\}$, $\{2,5\}$, $\{2,7\}$ düğümleri kafese eklenir. Algoritmanın işletimi sonlandığında $V(C)$ kümesinde, kavram kafesinde yer alan bütün düğümler bulunur.

DUGUMBUL algoritmasının algoritma karmaşıklığı şu biçimdedir: İlk satırda $O(m)^{11}$, ikinci satır $O(1)$ 'dir. Bu algoritmanın en iyi çalışma zamanı $r_1(P')$ ilişkisinde yer alan tutanaklar doğrusal bir sıra oluşturduğunda ($a(v_i) \subset a(v_{i+1})$ $1 < i < m-1$) görülür. Bu durumda dördüncü satır $O(m^2)$ adım gerektirir.

¹¹ m , $r_1(P')$ ilişkisinde yer alan tutanak sayısı, n nitelik sayısıdır.

Bu yüzden en iyi durumda algoritma karmaşıklığı $O(m^2)$ 'dir. En kötü çalışma zamanı $r_1(P')$ ilişkisinde yer alan tutanaklar ($i \neq j$ için $v_i, v_j \in V(C)$, $a(v_i) \cup a(v_j) = \{1, 2, \dots, n\}$) koşulunu sağlayan $m = \binom{n}{m-1}$ tutanaklardan oluşursa ortaya çıkar ve çalışma zamanı $\sum_{i=1}^n \binom{n}{i} = 2^n$ olur. Bu yüzden en kötü çalışma zamanı algoritma karmaşıklığı $O(2^n)$ 'dir.

Bir çizgedeki kenarlar kümesi $E(C)$ 'yi oluşturmak için, $V(C)$ 'de birbirleriyle doğrudan ilişkisi olan düğüm çiftleri bulunur. Eğer $a(v_i) \subset a(v_j)$ ve düğümler arasında $a(v_i) \subset a(v_k) \subset a(v_j)$ şartını sağlayan bir $v_k \in V(C)$ düğümü yoksa v_i ile v_j arasında bir kenar vardır. Buna kapsama özelliği denir. Eğer v_i, v_j kapsama özelliğini taşıyorsa v_i düğümü v_j düğümünü tam anlamıyla kapsar denir (tersi de söylenebilir). $E(C)$ 'de yer alan tüm elemanlar, $V(C)$ 'de kapsama özelliğini sağlayan tüm düğüm çiftleridir.

Şekil 5'de C çizgesindeki alan kenarları bulmak için kullanılan yaklaşım algoritmik olarak ifade edilmiştir.

KENARBUL algoritmasının algoritma karmaşıklığının $O(I^3)$ olduğu kolayca görülebilir. Örnek market-sepet verisi üzerinde, DUGUMBUL algoritması sonucunda elde edilen düğümler kümesi, kenar bulma algoritmasına girdi olarak verilirse Şekil 3'te verilen kenarlar bulunur.

Oluşturulan kavram kafesinin eşleştirme sorularını yanıtlayabilmesi için her bir düğümün geçiş sıklığı değeri hesaplanmalıdır. $w(a(v_i))$ işlevi $a(v_i)$ nitelikler kümesini kapsayan ve $K(a(v_i))$ ise $a(v_i)$ nitelikler kümesine özdeş (kapsayan ve aynı zamanda kapsanan) hareketlerin sayısını versin. $fr(a(v_i))$ işlevi aşağıdaki gibi hesaplanır.

```

KENARBUL(V(C)) YORDAMI;
DISYAPILAR;
V(C) , GİRİŞ;
/* C çizgesini oluşturan düğümler kümesi */
ICYAPILAR;
I, TAMSAYI;
/* C çizgesinde yer alan düğümlerin sayısı */
KOMUTLAR;
1. I := |V(C)|;
2. E(C) := ∅;
3. SAYARAK YINELE (i:=1, i <= I, i:=i+1)
4.   SAYARAK YINELE (j:=1, j <= I, j:=j+1) [
5.     kenarekle=YANLIS;
6.     EGER ( i ≠ j ) VE ( a(vi) ⊂ a(vj) ) İSE [
7.       kenarekle=DOGRU;
8.       SAYARAK YINELE (k:=1, kenarekle VE k<=I, k+1)
9.         EGER ( i ≠ j ) VE ( k ≠ j ) VE a(vi) ⊂ a(vk) ⊂ a(vj)
10.          kenarekle=YANLIS;
11.        ]
12.       EGER kenarekle İSE
13.         E(C):=E(C) ∪ { ( vi, vj ) }
14.      ]
KENARBUL BITTI

```

Şekil 5: Çizgedeki Kenarların Bulunması için Algoritma Kesiti

1. Eğer $v_j \in V r_1(P')$ ilişkisinde bir tutanak ise, $fr(a(v_j)) = K(a(v_j)) + v_j$ tarafından kapsanan v_j düğümler sayısıdır.

2. Eğer $v_j \in V r_1(P')$ ilişkisinde bir tutanak ise, $fr(a(v_j)) = K(a(v_j)) + v_j$ tarafından kapsanan v_j düğümler sayısıdır.

Örnek olarak Şekil 3'te yer alan düğümlerin bazılarının frekans değerleri hesaplanmıştır.

- $fr(\{1,2,7\})=2$, $fr(\{1,2,4\})=3$
- $fr(\{1,2\})=w(\{1,2,7\})+w(\{1,2,4\})-w(\{1,2,7\}) \cap w(\{1,2,4\})=3+2-1=4$
- $fr(\{1\})=w(\{1,4\})+w(\{1,2\})-w(\{1,4\}) \cap w(\{1,2\})=4+4-3=5$

2.2. Kavram Kafesinin Eşleştirme Sorgusu Kistaslarının Bulunmasında Kullanımı

α ve β , P alanında yer alan iki küme olsun; $c(\alpha R\beta)$ güvenlik değeri, sepet ilişkisi için kurulan kavram kafesi kullanılarak, aşağıda tanımlı kuralların uygulanması ile elde edilebilir.

1. $i, j \in J$ için $\alpha = a(v_i)$, $\beta = a(v_j)$ iken $\alpha R\beta \in E(C)$ ise $c(\alpha R\beta) = \text{fr}(\alpha \cup \beta) / \text{fr}(\alpha)^{12}$ dir. Eğer $\beta \subseteq \alpha$ ise $c(\alpha R\beta) = 1$ olacaktır. Öteki deyişle α, β kümeleri çizgede birer düğümse ve aralarında bir kenar mevcutsa güvenlik değeri her iki kümenin bileşiminin frekans değerinin α kümesinin frekans değerine oranıdır.

2. $i, j \in J$ için $\alpha = a(v_i)$, $\beta = a(v_j)$ iken $\alpha R\beta \notin E(C)$ ise $c(\alpha R\beta)$ değeri v_i , v_j düğümlerinin supremum düğümü¹³ aracılığı ile hesaplanır. $a^{-1}(v_i)$ ve $a^{-1}(v_j)$ düğümlerinin supremum düğümü v_m , $\gamma = a(v_m)$ olsun. Bu durumda $c(\alpha R\beta) = c(\alpha R\gamma) * c(\gamma R\beta)$ olarak hesaplanır. $\beta \subseteq \gamma$ olduğundan $c(\gamma R\beta) = 1$ dir. Bu yüzden $c(\alpha R\beta) = c(\alpha R\gamma)$ değeridir.

3. $i, j \in J$ için $\alpha \subseteq a(v_i)$, $\beta \subseteq a(v_j)$ iken eğer $\alpha \subseteq a(v)$ ve $\beta \not\subseteq a(v)$ koşulunu sağlayan bir $v \in V(C)$ düğümü yoksa veya $a(v_i) \subseteq a(v_k) \subseteq a(v_j)$ koşulunu sağlayan bir v_k düğümü yoksa (v_i den v_j 'ye herhangi bir düğüm aracılığıyla bir kenar gitmiyorsa) C çizgesinde β daima α ile birlikte yer alır. Bu durumda $c(\alpha R\beta) = 1$ 'dir.

4. $i, j \in J$ için $\alpha \subset a(v_i)$, $\beta = a(v_j)$ iken $\alpha \subset a(v_k) \subset a(v_j)$ koşulunu sağlayan bir $v_k \in V(C)$ düğümü yoksa $c(\alpha R\beta) = c(a(v_i) R\beta)$ 'dir.

5. $i, j \in J$ için $\alpha = a(v_i)$, $\beta \subset a(v_j)$ iken $\beta \subset a(v_k) \subset a(v_j)$ koşulunu sağlayan bir $v_k \in V(C)$ düğümü yoksa $c(\alpha R\beta) = c(\alpha R a(v_j))$ 'dir.

6. $i, j \in J$ için $\alpha \subset a(v_i)$, $\beta \subset a(v_j)$ iken $\beta \subset a(v_k) \subset a(v_j)$ koşulunu sağlayan bir $v_k \in V(C)$ düğümü yoksa $\beta \subset a(v_m) \subset a(v_j)$ koşulunu sağlayan bir $v_m \in V(C)$ $c(\alpha R\beta) = c(a(v_i) R a(v_j))$ 'dir.

¹² $\text{fr}(\alpha) = 0$ ise $c(\alpha R\beta)$ değeri 0 olarak kabul edilecektir.

¹³ Verilen iki düğümün en yakın ortak atasına (aşağıdan yukarı) supremum düğüm denir (Bkz. EK-1).

7. Eğer $a \subseteq a(v_j)$ olan hiç bir $v_j \in (V(C) - a^{-1}(\{1, 2, \dots, n\}))$ yoksa, $c(\alpha R \beta) = c(\beta R \alpha) = 0$; çünkü $fr(\beta)$ 'nin sıklık değeri ne olursa olsun $fr(\alpha \cup \beta) = 0$ 'dir. Bu durumda tanımlanamayan hesaplamaya imkan vermemek için $c(\alpha R \beta) = 0$ varsayımı yapılmıştır.

Önsav 1 : $\gamma \in a(v_j)$ ve iken $\gamma \subset a(v_k) \subset a(v_j)$ koşulunu sağlayan bir $v_k \in V(C)$ düğümü yoksa $fr(\gamma) = fr(a(v_j))$ 'dir.

Önsav 2 : $i, j, m \in J$ için $\alpha = a(v_i)$, $\beta = a(v_j)$, $v_m = \text{supremum}\{v_i, v_j\}$, $\gamma = a(v_m)$ olsun. $c(\alpha R \beta) = fr(\gamma)/fr(\alpha)$ 'dir.

$i, j \in J$ için $\alpha \subseteq a(v_i)$, $\beta \subseteq a(v_j)$ olsun. $\alpha \subset a(v_k) \subset a(v_i)$ koşulunu sağlayan bir $v_k \in V(C)$ düğümü ve $\beta \subset a(v_m) \subset a(v_j)$ koşulunu sağlayan bir $v_m \in V(C)$ düğümleri elde edilemesin. Bu takdirde, $\alpha R \beta$ eşleştirmesinin destek değerini bulmak için aşağıdaki kurallar kullanılır.

1. v_i ve v_j düğümleri arasında direkt bağlantı var ise C çizgesinde eşleştirme hazır olarak bulunmaktadır. Bu durumda $s(\alpha R \beta) = fr(\alpha \cup \beta)/fr(\emptyset)$ 'dir.

2. C çizgesinde eşleştirme hazır olarak mevcut değilse $i, j, m \in J$ için v_i ve v_j düğümlerinin supremum düğümü v_m destek değerini hesaplamada kullanılır. Bu durumda $s(\alpha R \beta) = fr(a(v_m))/fr(\emptyset)$ 'dir. Birinci kuralın tanımladığımız kuralın özel bir hali olduğunu görmek mümkündür.

3. $i, j \in J$ için $\alpha \subset a(v_i)$, $\beta = a(v_j)$ iken $\alpha \subset a(v_k) \subset a(v_i)$ koşulunu sağlayan bir $v_k \in V(C)$ düğümü yoksa $s(\alpha R \beta) = s(a(v_i) R \beta)$ 'dir.

4. $i, j \in J$ için $\alpha = a(v_i)$, $\beta \subset a(v_j)$ iken $\beta \subset a(v_k) \subset a(v_j)$ koşulunu sağlayan bir $v_k \in V(C)$ düğümü yoksa $s(\alpha R \beta) = s(\alpha R a(v_j))$ 'dir.

5. $i, j \in J$ için $\alpha \subset a(v_i)$, $\beta \subset a(v_j)$ iken $\alpha \subset a(v_k) \subset a(v_i)$ koşulunu sağlayan bir $v_k \in V(C)$ düğümü yoksa ve $\beta \subset a(v_m) \subset a(v_j)$ koşulunu sağlayan bir $v_m \in V(C)$ $s(\alpha R \beta) = s(a(v_i) R a(v_j))$ 'dir.

6. Eğer $\alpha \subseteq a(v_i)$ olan hiç bir $v_i \in (V(C)-a^{-1}(\{1,2,\dots,n\}))$ yoksa, $s(\alpha R \beta) = s(\beta R \alpha) = 0$; çünkü $fr(\alpha), fr(\beta)$ 'nin sıklık değeri ne olursa olsun $fr(\alpha \cup \beta) = 0$ 'dır.

Önsav 3: $i, j, m \in J$ için $\alpha = a(v_i), \beta = a(v_j), v_m = \text{supremum}\{v_i, v_j\}, \gamma = a(v_m)$ olsun. $s(\alpha R \beta) = fr(\gamma)/fr(\emptyset)$ 'dir.

Şekil 3'teki sepet ilişkisi için oluşturulan kavram kafesi aracılığı ile tanımlanan güvenlik ve destek kurallarının her birinin nasıl kullanıldığını örnekleyen durumlar incelenecektir.

Örnek 3: $2R\{1,2\}$ ilişkisinin güvenlik ve destek değerlerini hesaplayalım. $\{2\}$ ve $\{1,2\}$ kümeleri C çizgesinde birer düğüm ve aralarında bir kenar bulunduğundan birinci kural kullanılarak güvenlik değeri hesaplanır. $c(2R\{1,2\}) = fr(\{1,2\})/fr(2) = 4/5$, $s(2R\{1,2\}) = fr(\{1,2\})/fr(\emptyset) = 4/6$ 'dır.

Örnek 4: $1R2$ ilişkisinin güvenlik ve destek değerlerini hesaplayalım. $\{2\}$ ve $\{1\}$ kümeleri C çizgesinde birer düğümdür, ancak bu iki düğüm arasında bir kenar yoktur. Bu yüzden ikinci kural kullanarak hesaplamalar yapılır. Buna göre 1 ve 2 düğümünün supremum düğümü $\{1,2\}$ düğümüdür. O halde $c(1R2) = c(1R\{1,2\}) = 4/5$, $s(1R\{1,2\}) = 4/6$ 'dır.

Örnek 5: $3R1$ ilişkisinde 3 numaralı ürün 1 numaralı ürün ile birlikte bulunduğundan üç numaralı kural kullanılarak değerler hesaplanır. C çizgesinde 3 numaralı ürünü içeren ancak 1 numaralı ürünü içermeyen bir düğüm yoktur. $c(3R1) = 1$ olur. $s(3R1) = s(\{1,3,4\}R\{1\}) = 1/6$ 'dır.

Örnek 6: $\{2,4\}R\{2,5\}$ ilişkisi incelendiğinde $\{2,4\}$ kümesinin $\{1,2,4\}$ düğümünün alt kümesi olduğu ve çizgede $\{2,4\} \subset a(v_i) \subset \{1,2,4\}$ koşulunu sağlayan bir v_i düğümü bulunmadığından dördüncü kural uygulanır. $c(\{2,4\}R\{2,5\}) = c(\{1,2,4\}R\{2,5\}) = 1/3$, $s(\{2,4\}R\{2,5\}) = s(\{1,2,4\}R\{2,5\}) = 1/6$.

Örnek 7: $\{2,5\}R\{2,4\}$ ilişkisi incelendiğinde güvenlik değeri hesabı için beşinci kuralın uygulanacağı görülür. $c(\{2,5\}R\{2,4\}) = c(\{2,5\}R\{1,2,4\}) = 1/2$. Destek değeri hesabı için dördüncü kural uygulanır. Buna göre $s(\{2,5\}R\{2,4\}) = s(\{2,5\}R\{1,2,4\}) = 1/6$.

Örnek 8: $5R\{2,4\}$ incelendiğinde altıncı güvenlik ve beşinci destek kuralının uygulanacağı görülür. $c(5R\{2,4\}) = c(\{2,5\}R\{1,2,4\}) = 1/2$, $s(5R\{2,4\}) = s(\{2,5\}R\{1,2,4\})=1/6$.

Örnek 9: $\{3,5\}R\{2,4\}$ ilişkisi incelendiğinde $\{3,5\}$ kümesinin $\{1,2,...8\}$ düğümü dışında hiçbir düğümün alt kümesi olmadığı görülür. Bu durumda yedinci güvenlik altıncı destek kuralı uygulanır. $c(\{3,5\}R\{2,4\})=0$, $s(\{3,5\}R\{2,4\})=0$.

$\alpha, \beta \in \mathfrak{I}(P)$ olmak üzere $\langle xRy, c, s \rangle$ değerlerini hesaplayan GUVENLIK-DESTEKHESAPLA algoritması Şekil 6'da verilmiştir. Bu algoritmanın en iyi çalışma zamanı $a^{-1}(\alpha)$ ve $a^{-1}(\beta) \in E(C)$ olduğu durumdur. Bu durumda algoritma karmaşıklığı $O(l)$ 'dir. Herhangi iki düğümün supremum düğümünü bulmak $O(l)$ gerektirir. Bu yüzden en kötü çalışma zamanı algoritma karmaşıklığı $O(l^2)$ 'dir.

Bu bölümde, analitik sonuçların doğruluğunun testi için bir veri kümesinin kafesteki görüş noktasından yararlanılmıştır. Verilen bir kümenin kafesteki görüş açısı, o kümeyi kapsayan en küçük düğümün çizgenin iki uç düğümüne, \emptyset ve tüm ürünlerin yer aldığı düğümler, kadar giden tüm kenar ve düğümlerin oluşturduğu alt çizgedir. Bu gerçek, çizge bir kafes olduğu için, çizge içindeki herhangi bir düğüm için geçerlidir; yani, küresel infimum ve supremum düğümleri kafeste mutlaka yer alır.

```

GUVENLIKDESTEKHESAPLA (V(C)) YORDAMI;
DISYAPILAR;
V(C), GIRIS;
/* C çizgesini oluştursun düğümler kümesi */
E(C), GIRIS;
/* C çizgesindeki kenarlar kümesi */
c(aR,β), ÇIKIŞ;
/* c(aR,β) eşleştirmesinin güvenlik değeri */
s(aR,β), ÇIKIŞ;
/* s(aR,β) eşleştirmesinin destek değeri */
ICYAPILAR;
KOMUTLAR;
1. EGER (a-1(α),a-1(β)) ∈ E(C) İSE [
2.   c(aR,β) =  $\frac{P(\alpha \vee \beta)}{P(\alpha)}$ ; s(aR,β) =  $\frac{P(\alpha \vee \beta)}{P(\beta)}$ ;]
3. DEĞİLSE [
4.   v1 = DUGUMARA(α,V(C));
5.   v2 = DUGUMARA(β,V(C));
6.   EGER (v1,v2) ∈ E(C) İSE [
7.     c(aR,β) =  $\frac{P(\alpha \vee \beta)}{P(\alpha)}$ ; s(aR,β) =  $\frac{P(\alpha \vee \beta)}{P(\beta)}$ ;]
8.   DEĞİLSE [
9.     vmin = supremum(v1,v2);
10.    γ = g(vmin);
11.    c(aR,β) =  $\frac{P(\gamma)}{P(\alpha)}$ ; s(aR,β) =  $\frac{P(\gamma)}{P(\beta)}$ ;
12.  ]
13. ]
GUVENLIKDESTEKHESAPLA BITTI

```

Şekil 6: Verilen Bir Eşleştirmenin Güvenlik ve Destek Değerlerinin Hesaplanması

Şekil 6'da DUGUMARA ile gösterilen, C çizgesindeki ilgili düğümü döndüren yordam Şekil 7'de algoritmik olarak ifade edilmiştir.

```

DUGUMARA (V(C)) YORDAMI;
DISYAPILAR;
V(C), GİRİŞ;
/* C çizgesini oluşturan düğümler kümesi */
α, GİRİŞ;
/* P alanında tanımlı nitelik kümesi */
V, ÇIKIŞ;
/* v C çizgesinde bir düğüm ( v ∈ V(C) ) */
İÇYAPILAR;
KOMUTLAR;
1. SAYARAK YINELE (i=1, i < [V(C)], i=i+1)
2.   EGER α(vi) ≥ α İSE [
3.     devam=DOGRU;
4.     SAYARAK YINELE (j = 1, devam VE j < [V(C)], j=i+1)
5.       EGER (i≠1) VE αca(vi) < α(vi) İSE
6.         devam=YANLIŞ;
7.     EGER DEVAM İSE [
8.       v = vi;
9.     DON ]]
DUGUMARA BITTI;

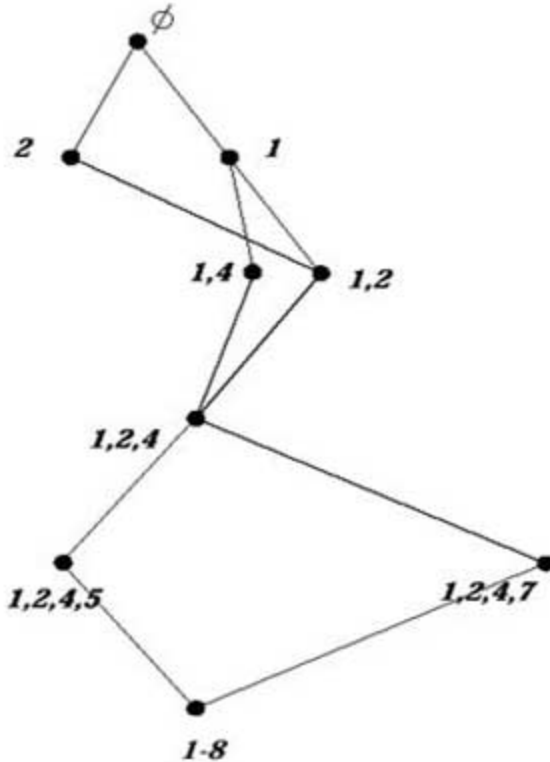
```

Şekil 7: [DUGUMARA] Çizgede Verilen Ürün Kümesine En Yakın Düğümü Bulan Algoritma Kesiti

Şekil 8'de {1,2,4} kümesi için görüş açısı verilmiştir. Bir kümenin görüş açısını bulabilmek için GORUS algoritması (bkz. Şekil 9) geliştirilmiştir. Bu algorithmada, öncelikle ilgili kümeyi kapsayan en küçük düğüm (kafesteki her bir düğümün bir veri/ürün kümesine tekabül ettiğini göz önünde bulundurunuz) bulunur ve tüm infimum ve supremum düğümleri bu düğümden çıkan kenarlar yardımıyla bir kuyruğa atılır. Kuyruk kullanılarak, bu düğüm ile infimum düğümleri arasındaki eşleştirme kuralları çıkarılır.

Sonuç

Makale kapsamında BKA kullanılarak eşleştirme kuralı çıkarım problemi için bir çerçeve geliştirilmiştir. Bu çerçeve eşleştirme kuralı çıkarımını matematiksel bir tabana oturtması açısından yeni bir yaklaşımdır. Geliştirilen çerçeve ile eşleştirme sorgularında bulunan bağımlılıkların, BKA ile elde edilebileceği gösterilmiştir. Eşleştirme kurallarının karşılaması gereken minimum destek ve güvenilirlik eşik ölçütlerinin kavram kafesi kullanılarak hesaplanabileceği analitik olarak gösterilmiştir. Bunun yanı sıra, yöntem, eşleştirme sorgularının uygulanmasına ilişkin zaman/yer karmaşıklığı hesabının yapılmasına olanak tanımıştır.



Şekil 8: {1,2,4} Kümesi için Görüş Açısı

Ek 1: Kafes için Matematiksel Ön Tanımlar

Tanım 1.0 [Kısmi Sıralı Kümeler]: Bir A kümesi üzerindeki bir \leq bağıntısı yansımali, antisimetrik ve geçişli ise, bu durumda \leq 'ye *kısmi sıralama* ve A kümesine de *kısmi sıralı küme* (Partially Ordered SET -POSET) denir. POSET \leq bağıntısı A kümesi üzerinde kısmi sıra olmak üzere (A, \leq) ikilisi ile gösterilir.

GORUS (V(C), E(C), α) YORDAMI;
 DISYAPILAR;
 V(C), GİRİŞ;
 /* C çizgesini oluşturan düğümler kümesi */
 α , GİRİŞ;
 /* P alanında tanımlı nitelik kümesi */
 v , DUGUM;
 /* $v \in C$ çizgesinde bir düğüm ($v \in V(C)$ ve $\alpha \subseteq v$) */
 k , KUYRUK;
 /* v düğümünün tüm infimum ve supremum düğümlerinin atıldığı kuyruk */
 on, arka TAMSAYI;
 /* kuyruk üzerinde göstergeler*/
 ICYAPILAR;
 KOMUTLAR;
 1. $v=DUGUMARA(V(C), \alpha)$
 2. v düğümün tüm infimum ve supremum düğümlerini E(C) aracılığıyla kuyruğa at
 3. Kuyrukta Yer alan Tüm Düğümler İçin
 4. GUVENLIKDESTEKHESAPLA(V(C), v , kuyrukta yer alan düğüm)
 GORUS BITTI;

Şekil 9: Görüş Algoritması

(a,b) kısmi sıralama bağıntısında yer alıyor ise $a \leq b$ ile gösterilir. A kümesi üzerinde kısmi sıralama ilişkisi aşağıdaki özellikleri taşır:

Yansımali : Her $a \in A$ için $a \leq a$ ise, \leq yansımali dır.
 Antisimetrik: $a, b \in A$, $a \leq b$ ve $b \leq a$ iken $a=b$ ise \leq antisimetriktir.
 Geçişli: $a, b, c \in A$, $a \leq b$ ve $b \leq c$ iken $a \leq c$ ise \leq geçişlidir.

Örnek 1.0: Gerçel sayılar kümesi üzerinde (veya gerçel sayılar kümesinin herhangi bir alt kümesinde) \leq bağıntısı bir kısmi sıralama bağıntısıdır. Çünkü, her $a \in A$ için $a \leq a$ (yansıma), $a=b$, eğer $a \leq b$ ve $b \leq a$ 'dir (antisimetri) ve $a \leq b$ ve $b \leq c$ iken $a \leq c$ 'dir (geçişlilik).

Tanım 1.1: (A, \leq) bir POSET olsun, A kümesinde yer alan a ve b öğeleri eğer $a \leq b$ veya $b \leq a$, \leq bağıntısında yer alıyorsa, a ve b karşılaştırılabilir öğelerdir, denir.

Tanım 1.2: (A, \leq) POSET'inde her $a, b \in A$ için $a \leq b$ veya $b \leq a$, \leq bağıntısında yer alıyorsa (A, \leq) POSET'ine *tamamen sıralı küme* denir.

Tanım 1.3: A bir POSET ve $S \subseteq A$ olsun. Her $s \in S$ için $s \leq a$ koşulunu sağlayan bir $a \in S$ elmanı varsa, a 'ya *maksimal öge*, her $s \in S$ için $a \leq s$ koşulunu sağlayan bir $a \in S$ elemanı varsa, a 'ya *minimal öge* denir.

Tanım 1.4: (A, \leq) POSET'inde $a \in A$ ögesi, her $b \in A$ için $b \leq a$ özelliğini gerçekleştiriyor ise, a ögesine (A, \leq) POSET'inin *en büyük ögesi* denir ve T ile gösterilir. Benzer biçimde, her $b \in A$ için $a \leq b$ ise, a ögesine (A, \leq) POSET'inin *en küçük ögesi* denir ve \perp ile gösterilir. Bir POSET'in en fazla bir en büyük ve bir en küçük ögesi vardır.

Tanım 1.5: (A, \leq) bir POSET ve $S \subseteq A$ olsun. Her $s \in S$ için $s \leq a$ koşulunu sağlayan bir $a \in A$ ögesi varsa, a 'ya S kümesinin *üst sınırı* denir. Her $s \in S$ için $a \leq s$ koşulunu sağlayan bir $a \in A$ ögesi varsa, a 'ya S kümesinin *alt sınırı* denir. S kümesinin tüm üst sınırları S^u ve tüm alt sınırları S^l kümeleri ile gösterilir.

$$S^u = \{ a \in A \mid (\forall s \in S) s \leq a \} \text{ ve } S^l = \{ a \in A \mid (\forall s \in S) s \geq a \}$$

Tanım 1.6: (A, \leq) bir POSET ve $S \subseteq A$ olsun. Eğer S^u kümesinin en az bir ögesi varsa, bu öge x olsun, x ögesine S kümesinin *en küçük üst sınırı* (*least upper bound, lub*) denir. Öteki deyişle, x ögesinin S kümesinin en küçük üst sınırı olabilmesi için şu koşulları taşıması gerekir.

- x , S kümesinin bir üst sınırı olmalıdır ($x \in S^u$).
- S kümesinin diğer tüm üst sınırları için $x \leq y$ olmalıdır.

Böylece, en küçük üst sınır bir kısmi sıralı kümenin bir S alt kümesi için, S nin tüm üst sınırlarından oluşan alt kümenin en küçük ögesidir.

Eğer S' kümesinin en az bir ögesi varsa, bu öge x olsun, x ögesine S kümesinin en büyük alt sınırı (*greatest lower bound, glb*) denir. Öteki deyişle, x ögesinin S kümesinin en büyük alt sınırı olabilmesi için şu koşulları taşıması gerekir.

- x , S kümesinin bir alt sınırı olmalıdır ($x \in S'$).
- S kümesinin diğer tüm alt sınırları için $y \leq x$ olmalıdır.

Diğer bir deyişle, en büyük alt sınır bir kısmi sıralı kümenin bir S alt kümesi için, S nin tüm alt sınırlarından oluşan alt kümenin en büyük ögesidir.

S kümesinin en küçük üst sınırına S kümesinin *supremumu* denir ve *sup* S ile gösterilir, en büyük alt sınırına S kümesinin *infimumu* denir ve *inf* S ile gösterilir. İki ögenin supremumu bu iki ögeden daha büyük olan en küçük ögedir (eğer böyle bir öge varsa bir tanedir). Benzer şekilde, iki ögenin infimumu bu iki ögeden daha küçük olan en büyük ögedir.

S kümesinin A kümesine eşit olduğu ve S kümesinin boş küme olduğu iki uç durum incelenmelidir. $S=A$ olsun. Bu durumda eğer A kümesinin en büyük ögesi varsa S kümesinin supremumu A kümesinin en büyük ögesidir. ($\text{sup } S = T$) eğer A kümesinin en büyük ögesi yoksa $\text{sup } S = \emptyset$ olur. Benzer biçimde $S = \emptyset$ ise eğer A kümesinin en küçük ögesi varsa $\text{sup } \emptyset = \perp$, eğer A kümesinin en büyük ögesi varsa $\text{inf } \emptyset = T$ olur.

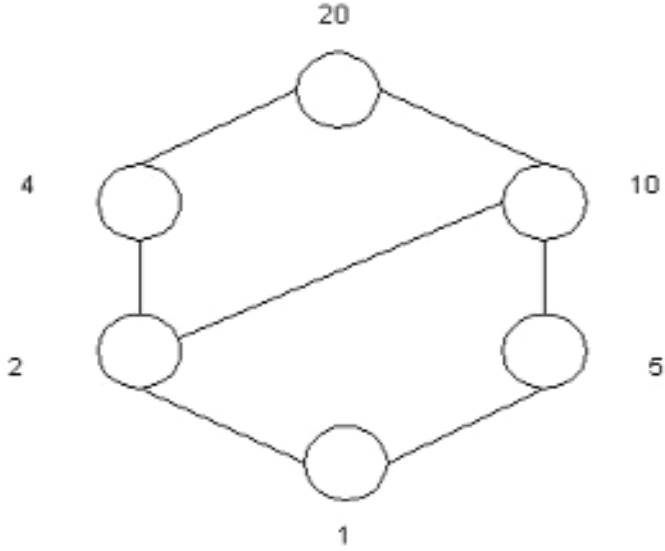
Supremum ve infimum ögelerinin bir diğer gösterim biçimi şöyledir. Eğer $\text{sup}\{x,y\}$ mevcutsa yerine $x \vee y$ (" x birleşim y " biçiminde okunur), eğer $\text{inf}\{x,y\}$ mevcutsa yerine $x \wedge y$ (" x kesişim y " biçiminde okunur) kullanılır. S kümesi için supremum ve infimum değerleri mevcutsa $\vee S$ ve $\wedge S$ kullanılır.

Tanım 1.7 [kafes]: A boş olmayan bir POSET olsun.

- Tüm $x, y \in A$ çiftleri için $x \vee y$ ve $x \wedge y$ mevcutsa, A POSET'i bir kafestir.
- Tüm $S \subseteq A$ için $\vee S$ ve $\wedge S$ mevcutsa, A tam kafes olur.

A kümesi bir kafes ise $\langle A; \wedge, \vee \rangle$ cebirsel yapıya sahiptir ve \vee ve \wedge A kümesinde tanımlı işleçlerdir.

Örnek 1.1: N pozitif doğal sayıların bir elemanı olsun ($N \in \mathbb{Z}^+$). N sayısının bölünenleri kümesi, D_N ile gösterilsin, bölünebilme bağıntısı altında bir kafestir. Şekil 1-1'de $N=20$ için D_N kafesi görülmektedir.



Şekil 1.1: $N=20$ için, Bölünebilme Bağıntısı Altında $D_{20} = \{1, 2, 5, 10, 20\}$ Kafesi

Örnek 1.2: Herhangi bir S kümesinin öz alt kümeleri ailesinin \subseteq bağıntısı ile oluşturduğu $(\mathfrak{A}(S), \subseteq)$ POSET'i tam kafestir. $(\mathfrak{A}(S), \subseteq)$ POSET'i aşağıdaki koşulları sağlar.

$\mathfrak{A}(S)$ POSET'ini L ile gösterelim. $\langle L, \subseteq \rangle$ üzerinde $x \vee y = x \cup y$ ve $x \wedge y = x \cap y$ biçimindedir. $\langle L, \subseteq \rangle$ tam bir kafestir.

Kaynakça

- Agrawal, R., Imielinski, T. ve Swami, A. (1993). Mining association rules between sets of items in large databases. *ACM SIGMOD Record, Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*. 22 (2): 207-216.
- Agrawal, R. ve Srikant, R. (1994). Fast algorithms for mining association rules in large databases. *20th International Conference on Very Large Databases, VLDB'94, September 12-15, Santiago, Chile*, içinde s. 487-499.
- Davey, B.A. ve Priestley, H.A. (1990). *Introduction to lattices and orders*. Cambridge: Cambridge University Press.
- Meo, R., Psaila, G. ve Ceri, S. (1996). A new SQL-like operator for mining association rules. *22nd International Conference on Very Large Databases, VLDB'96, September 3-6, Mumbai (Bombay), India*, içinde s. 1-12.
- Park, J. S., Chen, M.S. ve Yu, P.S. (1995). An effective hash-based algorithm for mining association rules. *Proceedings of the ACM SIGMOD International Conference on Management of Data, May 22-25, San Jose, California*, içinde s. 175-186.
- Pawlak, Z. (1984). Rough classification. *International Journal of Man-Machine Studies*, 31(1): 63-103.
- Savasere, A., Omiecinski, E. ve Navathe, S. (1995). An efficient algorithm for mining association rules in large databases. *21st International Conference on Very Large Databases, VLDB'95, September 11-15, Zurich, Switzerland*, içinde s. 432-444.
- Sever, H. ve Oğuz, B. (2002). Veri tabanlarında bilgi keşfine formal bir yaklaşım: Kısım 1- Eşleştirme sorguları ve algoritmalar. *Bilgi Dünyası*, 3(2): 173-204.
- Wille, R. (1982). Restructuring lattice theory: An approach based on hierarchies on concepts. I. Rival (ed.). *Ordered Sets* içinde (s. 445-470). Dordrecht-Boston: D. Reidel Publishing Company.

TEŞEKKÜR

Bu çalışma TÜBİTAK-EEEAG tarafından 1999E03 nolu ve Kavramsal Analizinin Eşleştirme Sorgularına Uygulanması adlı Araştırma Projesi tarafından desteklenmektedir.