Universitatea POLITEHNICA din București
Facultatea de Electronică, Telecomunicații și Tehnologia Informației
Universitatea Babeș-Bolyai
Facultatea de Matematică și Informatică

Romagna: o abordare modernă asupra uneltelor de analiză conceptuală a datelor

Lucrare de licență

Prezentată ca cerință parțială pentru obținerea
titlului de *Inginer*
în domeniul *Calculatoare și tehnologia informației*
programul de studii *Ingineria informației*

Conducător științific
Christian Săcarea

Absolvent
Mihai Chereji

Anul 2014

Declaraţie de onestitate academică

Prin prezenta declar că lucrarea cu titlul *Romagna: o abordare modernă asupra uneltelor de analiză conceptuală a datelor*, prezentată în cadrul Facultăţii de Electronică, Telecomunicaţii şi Tehnologia Informaţiei a Universităţii "Politehnica" din Bucureşti ca cerinţă parţială pentru obţinerea titlului de *Inginer* în domeniul Inginerie Electronică şi Telecomunicaţii/ Calculatoare şi Tehnologia Informaţiei, programul de studii *Ingineria informației* este scrisă de mine şi nu a mai fost prezentată niciodată la o facultate sau instituţie de învăţământ superior din ţară sau străinătate. Declar că toate sursele utilizate, inclusiv cele de pe Internet, sunt indicate în lucrare, ca referinţe bibliografice. Fragmentele de text din alte surse, reproduse exact, chiar şi în traducere proprie din altă limbă, sunt scrise între ghilimele şi fac referinţă la sursă. Reformularea în cuvinte proprii a textelor scrise de către alţi autori face referinţă la sursă. Înţeleg că plagiatul constituie infracţiune şi se sancţionează conform legilor în vigoare. Declar că toate rezultatele simulărilor, experimentelor şi măsurătorilor pe care le prezint ca fiind făcute de mine, precum şi metodele prin care au fost obţinute, sunt reale şi provin din respectivele simulări, experimente şi măsurători. Înţeleg că falsificarea datelor şi rezultatelor constituie fraudă şi se sancţionează conform regulamentelor în vigoare.

Bucureşti, Iulie 2014.
Absolvent: Mihai Chereji

.....

Cuprins

Lista figurilor

Lista tabelelor

Lista acronimelor

FCA = Formal Concept Analysis (Analiza Conceptuală Formală) HA = Harap Alb
PLL = Păsări-Lăți-Lungilă
Sp = Spânul

Introducere

Big-data și machine learning sunt domenii foarte căutate în zilele noastre, devenind chiar "buzzwords". Majoritatea se bazează pe algoritmi simplii, de adunare și prelucrare automată a datelor. Analiza conceptuală formală oferă o alternativă...

- Importanța explorării conceptelor
- Lacunele softului existent
- Intențiile aplicației

Capitolul 1
Analiza conceptuală formală

1.1 Introducere

Analiza conceptuală formală este o metodă de sistematizare a datelor în **concepte**, definite la modul larg ca mulțimi de obiecte care împărtășesc anumite atribute sau proprietăți. Este o reinterpretare a teoriei clasică a laticelor, dezvoltată în principal în anii '30, axată către partea practică. Conceptul a fost introdus în lucrarea seminală a lui Robert Wille din 1982 [Wil82], iar termenul a fost introdus în 1984 de același autor. În ultimele decenii, domeniul a atras multe contribuții și și-a dovedit utilitatea în domenii cum ar fi analiza și vizualizarea datelor, managementul informației.

1.2 Concepte matematice de bază

Întrucât scopul lucrării este de a descrie aplicația practică, ne vom limita la a descrie conceptele de care avem nevoie pentru a înțelege domeniul.

1.2.1 Mulțimi ordonate, latice, latice complete

Definiție 1. *O mulțime M este ordonată dacă se poate aplica asupra sa o relație R care îndeplinește următoarele condiții:*

Reflexivitate xRx

Antisimetrie $xRy, x \neq y \Rightarrow yRx$ *e fals*

Tranzitivitate $xRy, yRz \Rightarrow xRz$

$\forall x, y, z \in M$. *Relația R se numește o* ***relație de ordine****.*

Cel mai simplu exemplu, intuitiv exemplu este mulțimea numerelor reale $\mathbb{R}$, alături de relația $\leq$. Notăm o mulțime ordonată cu relația $\leq$ cu $(M, \leq)$.

Definiție 2. *Un element y al mulțimii M este* ***vecinul superior*** *al lui x dacă $x < y$ și nu există nici un z astfel încât $x < z < y$. În mod invers, x este* ***vecinul inferior*** *al lui y.*

Putem nota relația de vecinătate astfel: $x \prec y, y \succ x$.

Definiție 3. *Două elemente ale unei mulțimi ordonate sunt* ***comparabile*** *dacă $x \leq y$ sau $y \leq x$ (adică relația $\leq$ se aplică asupra lor). Altfel sunt* ***incomparabile****. Un* ***lanț*** *este o submulțime în care oricare două elemente sunt comparabile. Un* ***anti-lanț*** *este o submulțime în care oricare două elemente sunt incomparabile.*

Definiție 4. *Fie $(M, \leq)$ o mulțime ordonată, și N o submulțime a sa. Înțelegem prin* ***limita inferioară*** *a mulțimii N un element i astfel încât $\forall a \in N, i \leq a$. În mod invers,* ***limita superioară*** *a mulțimii s este definită prin $\forall a \in N, s \geq a$. Putem nota mulțimea tuturor*

limitelor inferioare a grupului N cu I. Elementul cel mai mare din această mulțime este numit ***minorantul*** *mulțimii N. Invers, cel mai mic element din mulțimea limitelor superioare este numit* ***majorantul*** *mulțimii N.*

Minorantul se poate nota cu $\wedge N$ sau $\inf N$ (de la **infimum**), iar majorantul cu $\vee N$ sau $\sup N$ (de la supremum).

Definiție 5. *O mulțime ordonată M este numită o* ***latice*** *dacă $\forall x, y \in M, \exists x \vee y, \exists x \wedge y$. În alte cuvinte, o mulțime ordonată este o latice dacă pentru orice 2 elemente ale mulțimii există majorant și minorant. O latice este* ***completă*** *dacă pentru orice submulțime a ei există majorant și minorant.*

Orice latice completă are un element superior, numit **elementul unitate**, și un element inferior, numit **elementul zero**.

Definiție 6. *Conform [CR04] O* ***conexiune Galois*** *este compusă din două mulțimi ordonate, M, N și două funcții γ, ψ astfel ca $\gamma : M \to N, \psi : N \to M$, dacă și numai dacă:*

1. $m_1 \leq m_2 \Rightarrow \gamma m_1 \geq \gamma m_2$
2. $n_1 \leq n_2 \Rightarrow \psi n_1 \geq \psi m_2$
3. $m \leq \psi\gamma m, n \leq \gamma\psi n$

sau, echivalent, $m \leq \psi n \Leftrightarrow n \leq \gamma m$

1.2.2 Context, concept, ierarhie de concepte

Definiție 7. *În cadrul analizei conceptuale, un* ***context*** *$K = (G, M, I)$ este format din 2 mulțimi, G și M, și o relație binară I între acestea. Mulțimea G reprezintă obiecte, iar M atribute.*

Literele provin din limba germană, în care conceptele au fost descrise inițial, de la Gegenstände și MerKmale, respectiv. Relația I e numită **relația de incidență**, iar gIm poate fi citit ca "obiectul g este descris de atributul m", sau "atributul m descrie obiectul g".

Exemplu 1. *Preluăm următorul exemplu din [CR04], un context (foarte redus)al animalelor vertebrate.*

		respiră în apă (a)	zboară (b)	are cioc (c)	are mâini (d)	are schelet (e)	are aripi (f)	trăiește în apă (g)	naște pui vii (h)	produce lumină (i)
1	Liliac		×			×	×		×	
2	Vultur		×	×		×	×			
3	Maimuță				×	×			×	
4	Pește papagal	×		×		×		×		
5	Pinguin			×		×	×	×		
6	Rechin	×				×		×		
7	Pește lanternă	×				×		×		×

Tabela 1.1: Un context al animalelor vertebrate. Sursa: CDA [CR04]

Pentru $A \subseteq G$, definim $A' = \{m \in M | gIm, \forall g \in A\}$.

În mod asemănător, pentru $B \subseteq M$, $B' = \{g \in G | gIm, \forall m \in B\}$.

În cuvinte, A' este mulțimea tuturor atributelor (din contextul la care ne raportăm) care descriu toate obiectele din A.

Definiție 8. *Un **concept** al contextului (G, M, I) este definit de $A \subseteq G$, $B \subseteq M$, unde $A' = B$ și $B' = A$.*

În engleză, mulțimea A (a tuturor obiecte descrise de atributele conceptului) este numită **extent** (extindere), iar B (atributele care descriu toate obiectele conceptului) **intent**(obiectiv).

Fie (G, M, I) un context, și A, A_1, A_2 submulțimi ale lui G, iar B, B_1, B_2 submulțimi de-ale lui M. Conform [GW97], atunci:

1. $A_1 \subseteq A_2 \Rightarrow A_1' \supseteq A_2'$
2. $A \subseteq A'''$
3. $A' = A'''$
4. $A \subseteq B' \Longleftrightarrow B \subseteq A' \Longleftrightarrow A \times B \subseteq I$

Proprietăți echivalente se observă imediat și pentru B, B_1, B_2.

Având în vedere că $' : \mathcal{P}(G) \to \mathcal{P}(M)$ și $B' : M \to G$, cei doi operatori pot fi combinați pentru a crea A'' și G'', care au ca domeniu mulțimea submulțimilor G și M respectiv.

Se observă pornind de la proprietățile enumerate mai sus că cele două funcții de derivare descriu o conexiune Galois între mulțimile submulțimilor pentru obiecte ($\mathcal{P}(G)$) și ($\mathcal{P}(M)$)

În exemplul de mai sus, $\{2, 4\}'' = \{2, 4, 5\}$, $\{d, h\}'' = \{d, e, h\}$.

Câteva proprietăți de remarcat ale conceptelor, așa cum sunt definite:

- Nu orice submulțime de obiecte definește extinderea unui concept. Din cele descrise mai sus, rezultă că e necesar ca $A = A''$ pentru A să fie extinderea unui concept.

  Ca exemplu, în tabelul 1.1, $\{6\}$ nu definește un concept, deoarece $\{6\}'' = \{4, 6, 7\}$, adică toate atributele care descriu rechinul în contextul nostru descriu de-asemenea și peștele lanternă, și peștele papagal.

- Intersecția oricâtor extinderi (sau obiective) de concepte are ca rezultat întotdeauna o altă extindere (respectiv obiectiv).

- În urma reuniunii lor, pe de altă parte, rareori rezultă o altă extindere.

- Mulțimea conceptelor unui context este o mulțime ordonată, dacă definim o relație de ordine în felul următor:

  Definiție 9. *Fie (A_1, B_1) și (A_2, B_2) concepte ale contextului $K = (G, M, I)$. Spunem că (A_2, B_2) este un **subconcept** al lui (A_1, B_1) (notat $(A_2, B_2) \leq (A_1, B_1)$ dacă $A_2 \subseteq A_1$. Astfel, (A_1, B_1) este **supraconceptul** lui (A_2, B_2)*

Teoremă 1. ***Teorema de bază a laticelor de concepte*** *- Fie un context (G, M, I), și o mulțime ordonată $\mathcal{C}(G, M, I; \leq)$ se numește laticea de concepte a contextului, care are majorantul și minorantul descrise de:*

$$\bigwedge_{t \in T} (A_t, B_t) = \left(\bigcap_{t \in T} A_t, \left(\bigcup_{t \in T} B_t \right)'' \right)$$

$$\bigvee_{t \in T} (A_t, B_t) = \left(\left(\bigcup_{t \in T} A_t \right)'', \bigcap_{t \in T} B_t \right)$$

- Diagrame Hasse
- Reducerea și clarificarea contextelor
- Rezolvarea contextelor cu valori multiple

1.3 Algoritmi relevați

1.4 Utilizări practice

Capitolul 2
Starea actuală

Există multe programe pentru diferite aspecte ale analizei conceptuale formale. O listă mai dezvoltată, care adună majoritatea programelor disponibile poate fi găsită la [Pri07].

Mai jos vom discuta doar câteva programe care au influențat dezvoltarea Romagnei, sau sunt relevante din alte motive.

2.1 Navigatoare de concepte

2.1.1 Toscana

Toscana[VW95] a fost lansat în 1995, și a fost unul din cele mai folosite unelte de explorare a laticelor de concepte pentru următorii ani.

Toscana este astăzi probabil cel mai bine ținută minte ca precursorul programului ToscanaJ, folosit și astăzi, prezentat mai jos.

În ciuda eforturilor noastre, unealta nu a fost găsită pentru descărcare.

2.1.2 Toscanaj

ToscanaJ([Dev14a]) este "moștenitorul direct" al lui Toscana, lucru evidențiat și de nume (J-ul vine de la Java).

După cum spune chiar site-ul programului [Dev14b]: "E o unealtă de vizualizare pentru scheme conceptuale foarte avansată, care reușește să afișeze informație interogată dintr-o bază de date în diagrame de latice, sau direct din structuri de date luate din memorie."

2.1.2.1 Funcționalități

Din nou, citând site-ul programului[Dev14b], prezentăm câteva funcționalități ale programului:

- Afișarea diagramelor simple și imbricate.
- Culoarea unui nod reprezintă mărimea contingentului obiectelor (poate fi modificat să reprezinte cuprinsul), deasemenea mărimea nodului poate fi folosită pentru același tip de informație.
- Mulțimea de obiecte de interes poate fi filtrată printr-un dublu click asupra nodurilor
- Nodurile din diagramă pot fi selectate pentru a fi scoase în evidență, pentru a ajuta citirea /n.t. diagramei].
- Diagramele pot fi exportate ca SVG, PNG și JPEG. Informații adiționale despre cum diagrama a fost obținută sunt exportate ca fișiere text separate, prin memoria temporară a calculatorului *(clipboard)*, sau direct în fișierul SVG (ca elementul `<desc>`).

- Etichetele nodurilor pot avea conținut diferit, folosindu-se de fragmentele de SQL specifice datelor
- Vizualizări adiționale a bazei de date pot fi deschise din diagramă, de exemplu folosind șabloane HTML în care rezultatele interogărilor sunt afișate.
- Interfața de vizualizare a bazei de date a fost gândită ca o interfață pentru pluginuri pentru a ușura extinderea ToscanaJ pentru scopuri specifice.
- Descrieri HTML pot fi atașate schemei, diagramelor și atributelor
- Vederile bazelor de date pot fi folosite pentru atribute, de exemplu pentru a interoga un URL din baza de date care e mai apoi deschis într-un navigator extern.

Având în vedere că scopul programului Romagna este de a oferi o alternativă programului Toscana(J), aceste funcționalități se vor regăsi și în Romagna, alături de altele, descrise în capitolul 3.1.

Mai jos prezentăm două capturi de ecran care prezintă programul ToscanaJ.

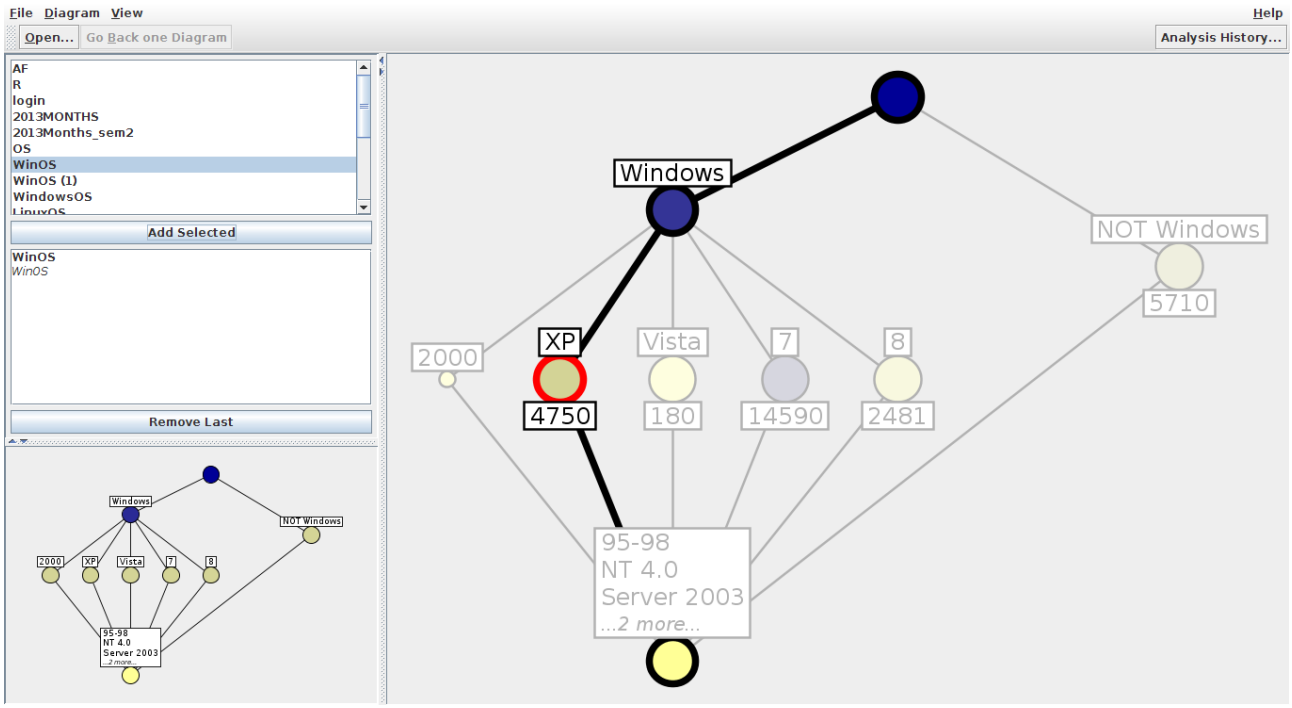

Figura 2.1: ToscanaJ, afișând o latice simplă generată din date de acces ale unui site web

Interfața ToscanaJ este punctul de pornire pentru Romagna. Vom încerca să păstrăm anumite elemente comune, pentru a ajuta utilizatorii obișnuiți, dar vom face, bine-înțeles schimbări și adaptări, mai ales luând în considerare diferențele convențiilor dintre interfețele aplicațiilor web, și cele desktop.

ToscanaJ este, asemenea Romagna, doar un navigator de concepte. Modelarea conceptelor din date este realizată cu ajutorul altor programe din suita din care face parte și ToscanaJ.

Elba este un editor pentru schemele conceptuale, legat de baze de date.

Siena este un editor pentru scheme conceptuale, foarte asemănător cu **Elba**, diferența fiind că Siena nu are nevoie de o legătură cu o bază de date, putând descrie concepte simple direct în program.

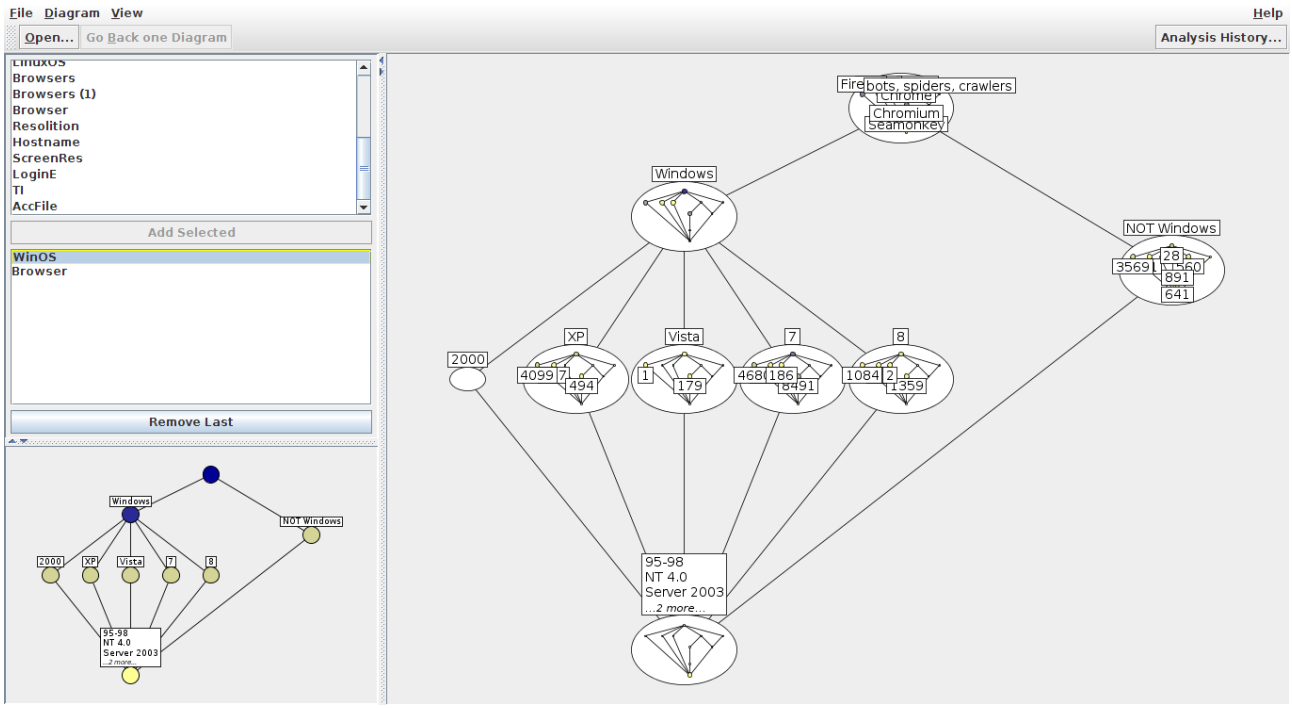

Figura 2.2: ToscanaJ, afișând aceeași latice ca în 2.1, de data aceasta imbricată cu altă diagramă

2.1.3 GaloisExplorer

GaloisExplorer [tea09] este un program mai recent, cu ultima actualizare în 2009.

Are o intefață realizată în QT, ceea ce îi permite să funcționeze pe mai multe platforme. O abordare interesantă, dar în cele din urmă doar cu valoare estetică, este prezentarea laticelor în 3d (2.3).

Din păcate, folosește anumite librării 3d (coin3d[**?**]) care nu sunt imediat disponibile și care necesită compilare manuală.

Pentru mulți utilizatori care nu au cunoștințe avansate de calculatore, aceste impedimente sunt motiv suficient pentru a abandona această instalare.

2.2 Software conex

În această secțiune vom prezenta câteva alte programe remarcabile în domeniul FCA, sau care au avut un impact indirect asupra dezvoltării Romagna:

FCAStone dezvoltat de Uta Priss, e un program care urmărește obținerea inter-operabilității între diferite programe pentru FCA (numele vine de la Piatra *[n.t. Stone = Piatră]* Rosetta) și convertește contexte în latice de concepte, și latice de concepte în formate grafice.

Conexp dezvoltat de Serhiy A. Yevtushenko, este o unealtă scrisă în Java, folosită în principal în scopuri educaționale (dar nu numai), are numeroase funcționalități de modelare a contextelor, dintre care amintim:

- calculează numărul de concepte formale dintr-un context dat
- calculează laticea de concepte
- permite explorarea atributelor
- calculează mulțimea de implicații Duquenne-Guiges

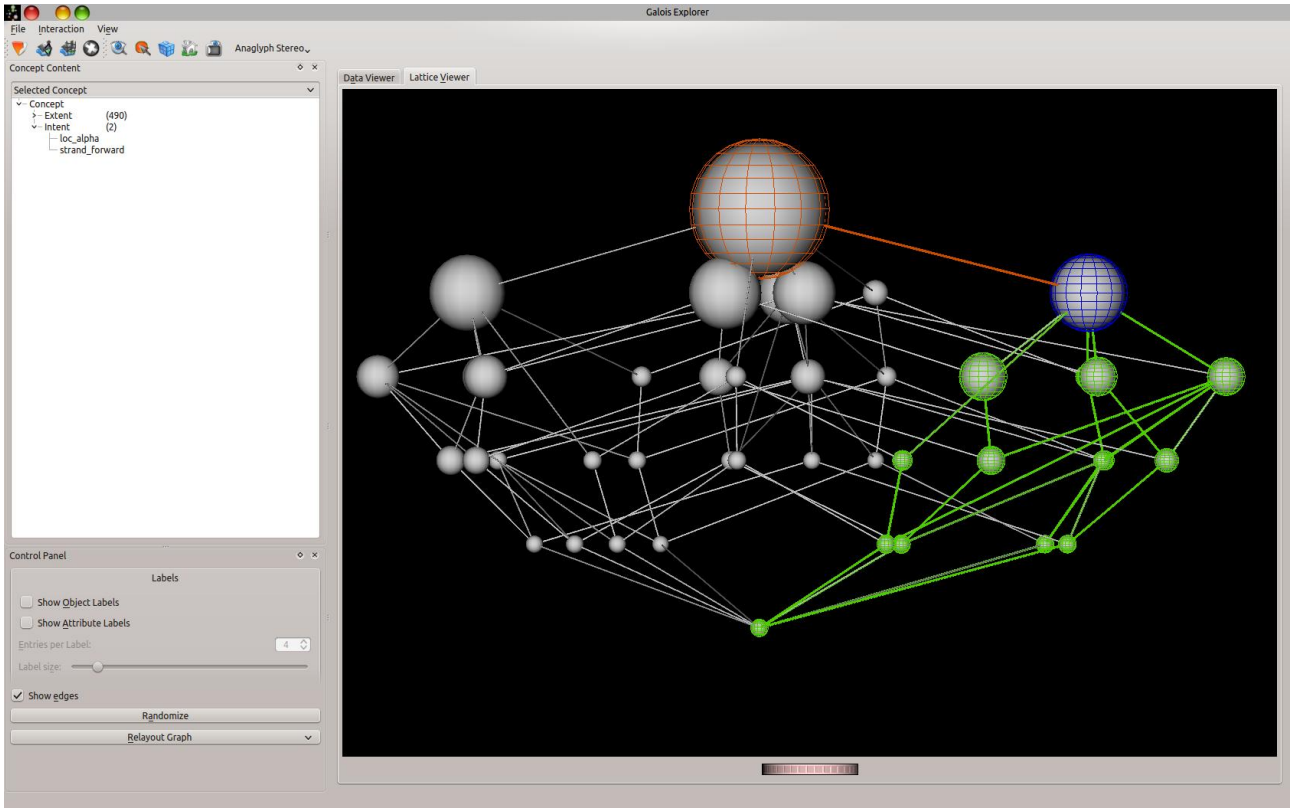

Figura 2.3: GaloisExplorer, afișând o diagramă în 3d. Sursa: site-ul proiectului [tea14]

- calculează regulile de asociații

LaTeXfor FCA [Gan14] Plugin pentru LaTeX, scris de Bernhard Ganter, oferă câteva "environment"-uri noi, folosite și în această lucrare, care permite aranjarea ușoară în pagina a contextelor și a laticilor derivate din acestea.

Capitolul 3
Romagna

Romagna este o aplicație dezvoltată începând cu anul 2014, pornită la Universitatea Babeș-Bolyai ca o alternativă folosindu-se de tehnologiile web pentru un navigator de concepte modern.

3.1 Raționament

3.1.1 Ușurință de utilizare

3.1.2 Avantajele distribuirii aplicațiilor web

3.1.3 Folosirea web-ului, păstrarea confidențialității datelor

3.2 Structură

3.3 Tehnologii

3.3.1 CoffeeScript

3.3.2 Ember.js

3.3.3 d3.js

3.3.3.1 svg

3.3.4 sql.js

3.3.4.1 emscripten

Emscripten e super mișto pentru că te lasă să compilezi programe scrise în C/++ în JavaScript... TO BE CONTINUED

3.4 Dezvoltare

3.4.1 Proces

3.4.2 Probleme întâmpinate

3.4.2.1 MySQL versus sql.js

3.4.2.2 SVG IS FUCKED

3.4.3 Viitor

Capitolul 4
Concluzii

Bibliografie

[CR04] Claudio Carpineto and Giovanni Romano. *Concept Data Analysis: Theory and Applications*. John Wiley & Sons, 2004.

[Dev14a] ToscanaJ Developers. Toscanaj homepage. `http://toscanaj.sourceforge.net/`, 2014. [Online; accesat la 15-Iunie-2014].

[Dev14b] ToscanaJ Developers. Toscanaj homepage. `http://toscanaj.sourceforge.net/toscanaJ/index.html`, 2014. [Online; accesat la 15-Iunie-2014].

[Gan14] Bernhard Ganter. Latex for fca. `http://www.math.tu-dresden.de/~ganter/fca/`, 2014. [Online; accesat la 15-Iunie-2014].

[GW97] Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edition, 1997.

[Pri07] Uta Priss. Formal concept analysis homepage. `http://www.upriss.org.uk/fca/fcasoftware.html`, 2007. [Online; accesat la 15-Iunie-2014].

[tea09] GaloisExplorer team. Galoisexplorer: Project web hosting - open source software. `http://galoisexplorer.sourceforge.net/`, 2009. [Online; accesat la 15-Iunie-2014].

[tea14] GaloisExplorer team. Galoisexplorer — free science & engineering software downloads at sourceforge.net. `http://sourceforge.net/projects/galoisexplorer/`, 2014. [Online; accesat la 15-Iunie-2014].

[VW95] Frank Vogt and Rudolf Wille. Toscana — a graphical tool for analyzing and exploring data. In Roberto Tamassia and IoannisG. Tollis, editors, *Graph Drawing*, volume 894 of *Lecture Notes in Computer Science*, pages 226–233. Springer Berlin Heidelberg, 1995.

[Wil82] Rudolf Wille. Restructuring lattice theory: an approach based on hierarchies of concepts. In Ivan Rival, editor, *Ordered Sets*, pages 445–470, Dordrecht/Boston, 1982. Reidel.