Towards Finding Relational Redescriptions

Esther Galbrun   Angelika Kimmig

Helsinki Institute for Information Technology
Department of Computer Science, University of Helsinki

Departement Computerwetenschappen, KU Leuven

Outline

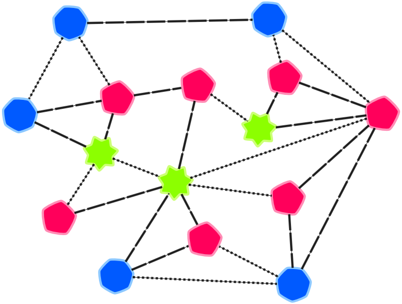
Relational Redescription Mining

Query Mining

Experiments

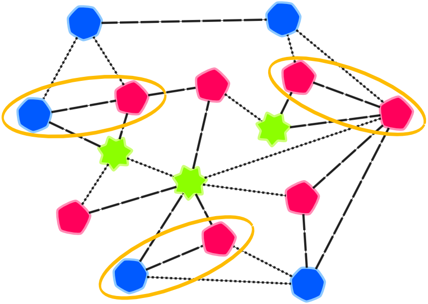
Conclusion

Definition



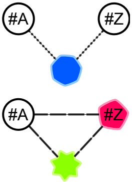
Dataset A network with node and egde attributes

Definition



Dataset A network with node and egde attributes

Task Find structurally different patterns covering (almost) the same tuples of nodes.



Aims

- Find node tuples with similar connection patterns
- Discover sets of related attributes
- View the same objects under different perspectives

Related Work

- Redescription Mining with propositional features [Ramakrishnan04],
- Inductive Logic Programming,
  - Progol [Muggleton95],
  - Aleph [Srinivasan07],
  - path-finding [Richards92, Ong05],

Goal

Describe connections between tuples of nodes

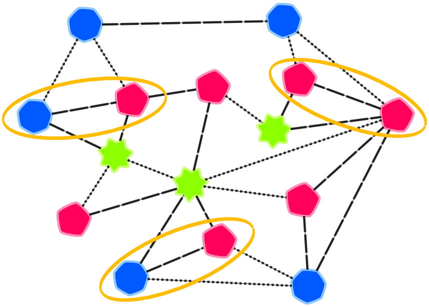
Goal

Describe connections between tuples of nodes

vs

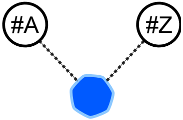
Characterize individual nodes using surrounding relations.

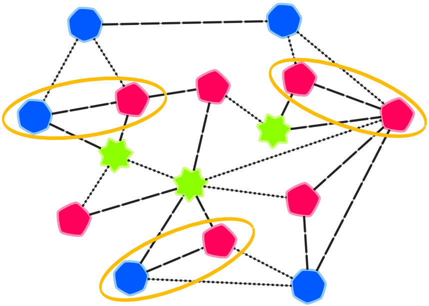
Alternating Scheme



Alternating Scheme

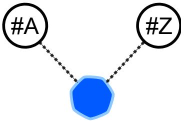
1. Fix a pattern to obtain examples

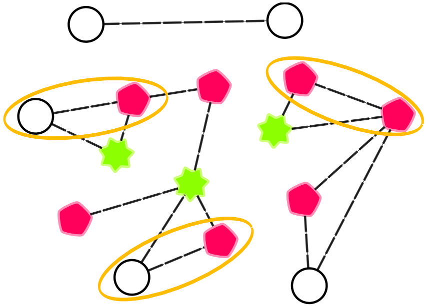




Alternating Scheme

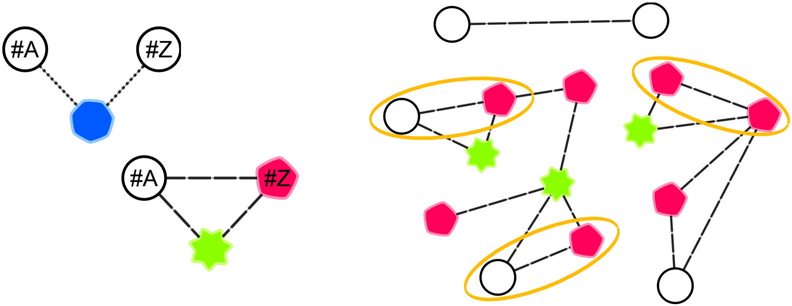
1. Fix a pattern to obtain examples
2. Consider remaining attributes





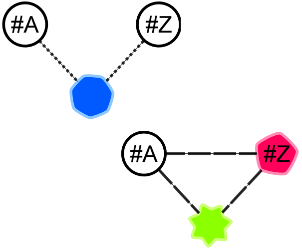
Alternating Scheme

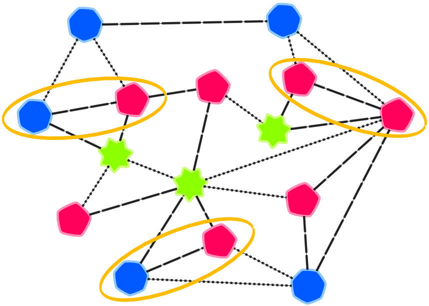
1. Fix a pattern to obtain examples
2. Consider remaining attributes
3. Find a matching pattern



Alternating Scheme

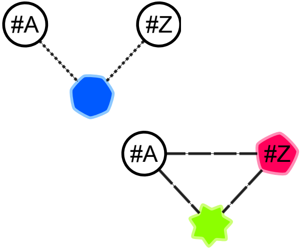
1. Fix a pattern to obtain examples
2. Consider remaining attributes
3. Find a matching pattern
4. Swap roles and iterate

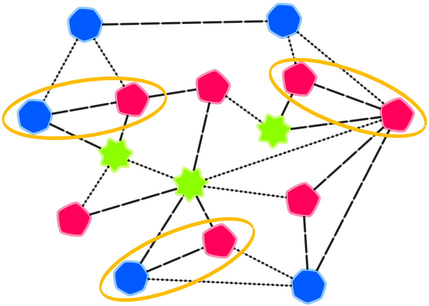




Alternating Scheme

1. Fix a pattern to obtain examples
2. Consider remaining attributes
3. **Find a matching pattern**
4. Swap roles and iterate

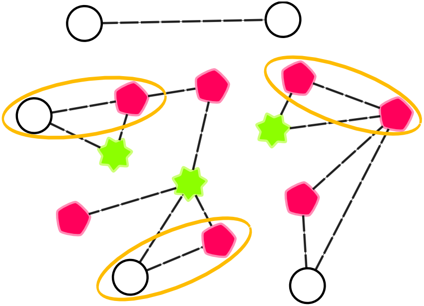




Subproblem: Query mining

✓ Given a set of examples
 and a subset of attributes

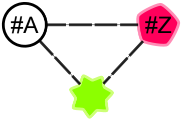
- **Find a matching pattern**

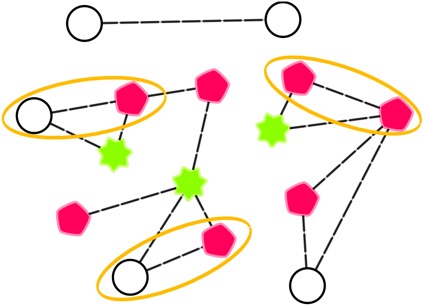


Subproblem: Query mining

✓ Given a set of examples
  and a subset of attributes

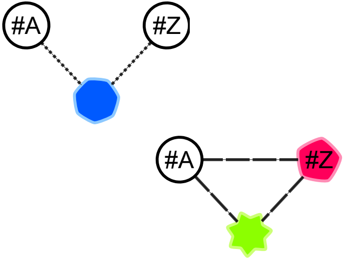
- **Find a matching pattern**

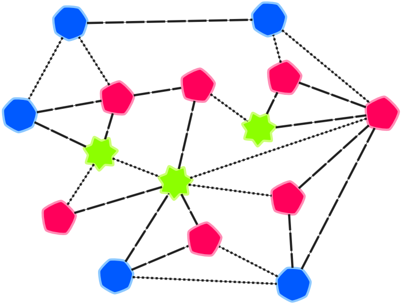




Requirements

- Connected patterns
- Complex patterns
- Efficient mining





Stepwise Approach

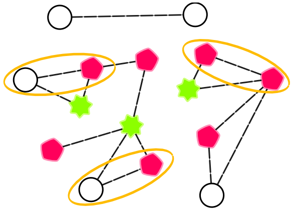
1. Enumerate connecting paths
   and mine frequent path patterns
2. Build graph patterns from path patterns
3. Select a subset of graph patterns

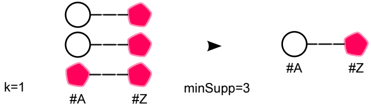
1. Find path patterns

Starting with paths of length $k = 1$

1. Enumerate connecting paths
2. Mine frequent path patterns
3. Increase k by one and iterate

Until all examples are connected
or k exceeds a chosen threshold



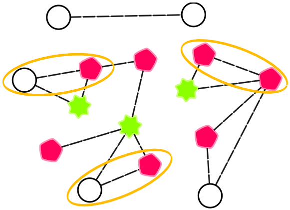


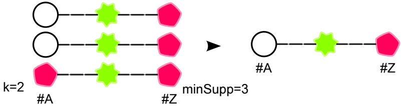
1. Find path patterns

Starting with paths of length $k = 1$

1. Enumerate connecting paths
2. Mine frequent path patterns
3. Increase k by one and iterate

Until all examples are connected
or k exceeds a chosen threshold



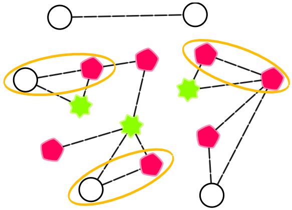


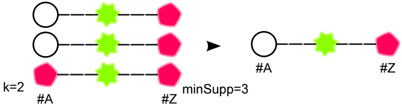
1. Find path patterns

Starting with paths of length $k = 1$

1. Enumerate connecting paths
2. **Mine frequent path patterns**
3. Increase k by one and iterate

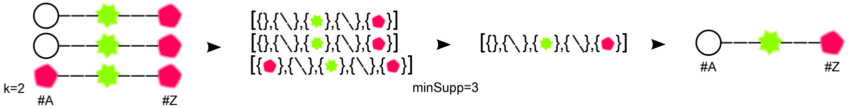
Until all examples are connected
or k exceeds a chosen threshold





1. Find path patterns

- Align all connecting paths for a fixed length *k*,
  - represented as sequences,
  - one set for each node/edge
- Frequent sequence mining
  - special gap constraints
  - constraint-based mining using `FIM_CP`

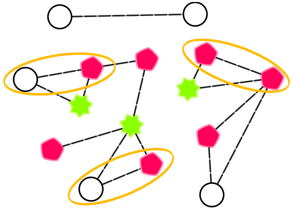


1. Find path patterns

Starting with paths of length $k = 1$

1. Enumerate connecting paths
2. Mine frequent path patterns
3. Increase k by one and iterate

Until all examples are connected
or k exceeds a chosen threshold

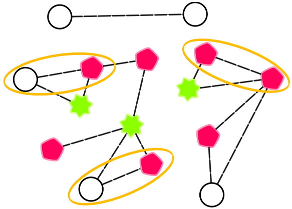


1. Find path patterns

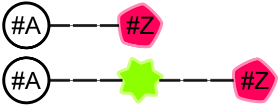
Starting with paths of length $k = 1$

1. Enumerate connecting paths
2. Mine frequent path patterns
3. Increase k by one and iterate

Until all examples are connected
or k exceeds a chosen threshold

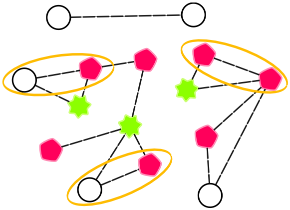


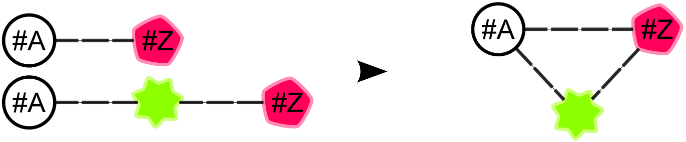
Outcome a set of frequent path patterns



2. Build graph patterns

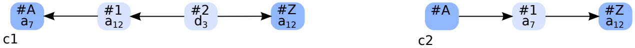
- ✓ Given a set of path patterns
  and of examples
- Combine paths
  to build graph patterns



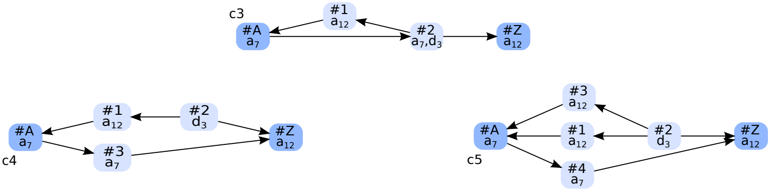


2. Build graph patterns

- How to combine path patterns?

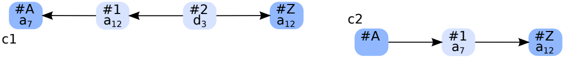


- Infinitely many combinations

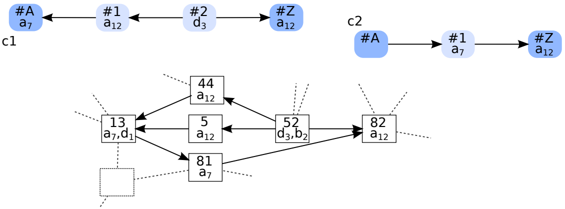


- But few occuring in the data

2. Build graph patterns

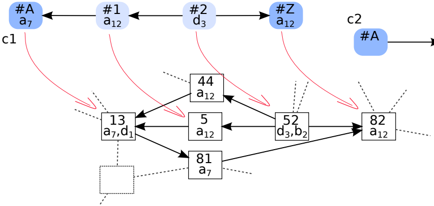


2. Build graph patterns



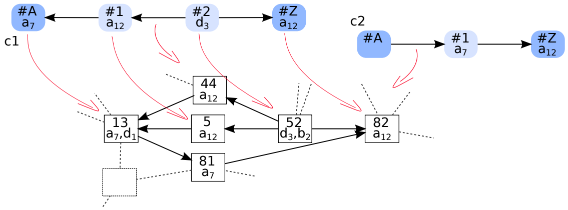
- Combination based on the data

2. Build graph patterns



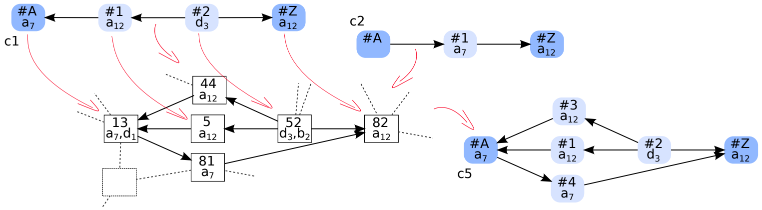
- Combination based on the data
- For each supporting example
  - Map pattern path onto data nodes

2. Build graph patterns



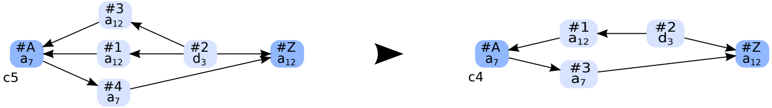
- Combination based on the data
- For each supporting example
  - Map pattern path onto data nodes

2. Build graph patterns



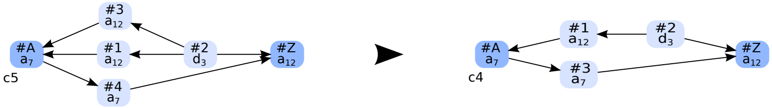
- Combination based on the data
- For each supporting example
  - Map pattern path onto data nodes
  - Generate *maximal* combination graph

2. Build graph patterns



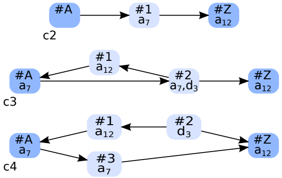
- Reduce *maximal* patterns to their *singular*,
  remove duplicate nodes,
  with identical labels and indentical neighbors

2. Build graph patterns



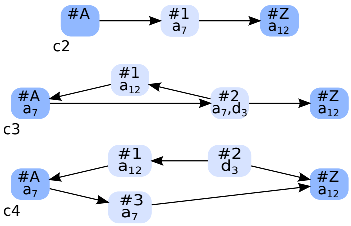
- Reduce *maximal* patterns to their *singular*,
  remove duplicate nodes,
  with identical labels and indentical neighbors

Outcome a set of *singular* graph patterns



3. Select graph patterns

- ✓ Given a set of graph patterns
  and of examples
- Select a good cover

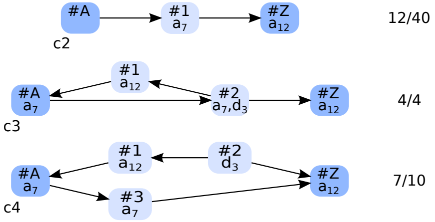


3. Select graph patterns

- Determine for each graph pattern

  $P =$ Number of covered examples

  $N =$ Number of other covered node pairs
- Greedy selection of pattern based on ratio $P/(P+N)$



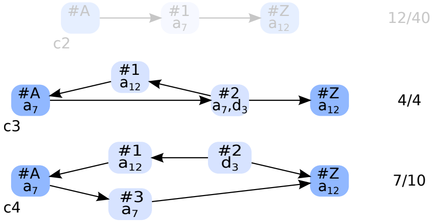
3. Select graph patterns

- Determine for each graph pattern

  $P =$ Number of covered examples

  $N =$ Number of other covered node pairs

- Greedy selection of pattern based on ratio $P/(P+N)$



3. Select graph patterns

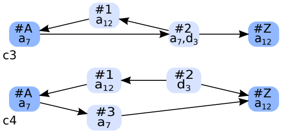
- Determine for each graph pattern

  $P =$ Number of covered examples

  $N =$ Number of other covered node pairs

- Greedy selection of pattern based on ratio $P/(P+N)$

Outcome a small set of graph patterns best matching the examples



Stepwise Approach

1. Enumerate connecting paths
   and mine frequent path patterns
2. Build graph patterns from path patterns
3. Select a subset of graph patterns

Stepwise Approach

1. Enumerate connecting paths
   and mine frequent path patterns
2. Build graph patterns from path patterns
3. Select a subset of graph patterns

- In practice, steps 2. and 3. are interleaved,
  to avoid costly mappings of unpromizing patterns.

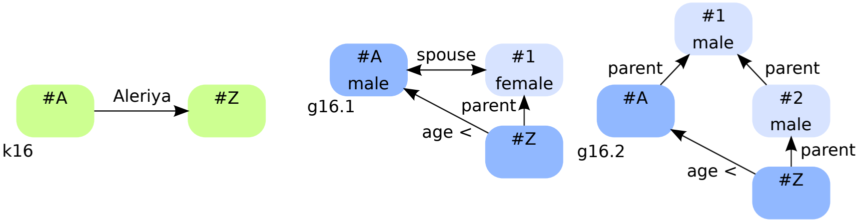
Experiments: Alyawarra Kinship

Data Alyawarra Ethnographic Database

- **nodes** individuals of an Australian indigenous community
- **relations** kinship and genealogical relations

Goal Find genealogical patterns to explain kinship terminology

Experiments: Alyawarra Kinship



"Son, male speaker" "Brother's child"

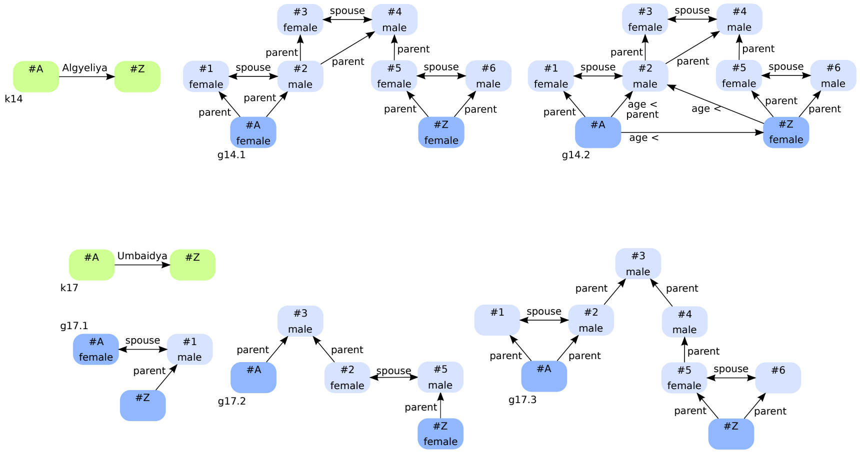
Experiments: Alyawarra Kinship

	Kinship relation	$\lvert\mathscr{O}^+\rvert$	$\lvert\mathscr{O}_k\rvert$	$\lvert\text{supp}^+\rvert$	$\lvert\text{supp}^-\rvert$	Prec.	Rec.	Jacc.	$\lvert P\rvert$	Time (s)
(1)	Arengiya	228	15	0	0	0	0	0	0	1.60
(2)	Anyainya	489	243	123	4	0.968	0.251	0.249	3	40.45
(3)	Aidmeniya	231	113	30	4	0.882	0.129	0.127	4	21.55
(4)	Aburliya	379	59	24	7	0.774	0.063	0.062	3	3.44
(5)	Adardiya	493	91	21	1	0.954	0.042	0.042	2	2.64
(6)	Agngiya	508	199	138	2	0.985	0.271	0.27	3	56.02
(7)	Aweniya	453	231	127	12	0.913	0.28	0.273	5	67.43
(8)	Amaidya	817	92	92	1	0.989	0.112	0.112	2	1.85
(9)	Abmarliya	805	172	79	7	0.918	0.098	0.097	3	19.90
(10)	Awaadya	462	49	43	1	0.977	0.093	0.092	2	4.39
(11)	Anguriya	505	43	37	2	0.948	0.073	0.072	2	4.01
(12)	Adiadya	739	83	72	4	0.947	0.097	0.096	5	19.39
(13)	Angeliya	299	220	40	9	0.816	0.133	0.129	5	260.80
(14)	Algyeliya	447	205	36	4	0.9	0.08	0.079	2	180.09
(15)	Adniadya	43	30	9	3	0.75	0.209	0.195	1	55.51
(16)	Aleriya	943	384	277	26	0.914	0.293	0.285	5	153.23
(17)	Umbaidya	1256	364	276	7	0.975	0.219	0.218	3	163.26
(18)	Anowadya	392	61	61	3	0.953	0.155	0.154	2	0.55
(19)	Muriya	569	181	20	0	1	0.035	0.035	4	30.51
(20)	Agenduriya	13	9	0	0	0	0	0	0	18.68
(21)	Amburniya	272	118	94	19	0.831	0.345	0.323	7	8.19
(22)	Andungiya	142	58	20	8	0.714	0.14	0.133	3	3.88
(23)	Aneriya	193	85	0	0	0	0	0	0	10.56
(26)	Undyaidya	6	3	0	0	0	0	0	0	1.03

Experiments: Alyawarra Kinship

	Kinship relation	$\lvert\mathscr{O}^+\rvert$	$\lvert\mathscr{O}_k\rvert$	$\lvert\text{supp}^+\rvert$	$\lvert\text{supp}^-\rvert$	Prec.	Rec.	Jacc.	$\lvert P\rvert$	Time (s)
(1)	Arengiya	228	15	0	0	0	0	0	0	1.60
(2)	Anyainya	489	243	123	4	0.968	0.251	0.249	3	40.45
(3)	Aidmeniya	231	113	30	4	0.882	0.129	0.127	4	21.55
(4)	Aburliya	379	59	24	7	0.774	0.063	0.062	3	3.44
(5)	Adardiya	493	91	21	1	0.954	0.042	0.042	2	2.64
(6)	Agngiya	508	199	138	2	0.985	0.271	0.27	3	56.02
(7)	Aweniya	453	231	127	12	0.913	0.28	0.273	5	67.43
(8)	Amaidya	817	92	92	1	0.989	0.112	0.112	2	1.85
(9)	Abmarliya	805	172	79	7	0.918	0.098	0.097	3	19.90
(10)	Awaadya	462	49	43	1	0.977	0.093	0.092	2	4.39
(11)	Anguriya	505	43	37	2	0.948	0.073	0.072	2	4.01
(12)	Adiadya	739	83	72	4	0.947	0.097	0.096	5	19.39
(13)	Angeliya	299	220	40	9	0.816	0.133	0.129	5	260.80
(14)	Algyeliya	447	205	36	4	0.9	0.08	0.079	2	180.09
(15)	Adniadya	43	30	9	3	0.75	0.209	0.195	1	55.51
(16)	Aleriya	943	384	277	26	0.914	0.293	0.285	5	153.23
(17)	Umbaidya	1256	364	276	7	0.975	0.219	0.218	3	163.26
(18)	Anowadya	392	61	61	3	0.953	0.155	0.154	2	0.55
(19)	Muriya	569	181	20	0	1	0.035	0.035	4	30.51
(20)	Agenduriya	13	9	0	0	0	0	0	0	18.68
(21)	Amburniya	272	118	94	19	0.831	0.345	0.323	7	8.19
(22)	Andungiya	142	58	20	8	0.714	0.14	0.133	3	3.88
(23)	Aneriya	193	85	0	0	0	0	0	0	10.56
(26)	Undyaidya	6	3	0	0	0	0	0	0	1.03

Experiments: Alyawarra Kinship



Conclusion

- Relational Redescription Mining problem definition
- Query miner as a first step towards a solution
- Experiments on the Alyawarra kinship problem

Conclusion

- Relational Redescription Mining problem definition
- Query miner as a first step towards a solution
- Experiments on the Alyawarra kinship problem

To be continued...

- Complete RRM algorithm
- Experiments with various datasets

Conclusion

- Relational Redescription Mining problem definition
- Query miner as a first step towards a solution
- Experiments on the Alyawarra kinship problem

To be continued...

- Complete RRM algorithm
- Experiments with various datasets

Thank you ...

Esther Galbrun
esther.galbrun@cs.helsinki.fi

Department of Computer Science
University of Helsinki
Finland

Helsinki Institute for Information Technology
Helsinki
Finland

Angelika Kimmig
angelika.kimmig@cs.kuleuven.be

Katholieke Universiteit Leuven
Leuven
Belgium

University of Maryland
College Park
USA