Graph Matching for Context Recognition

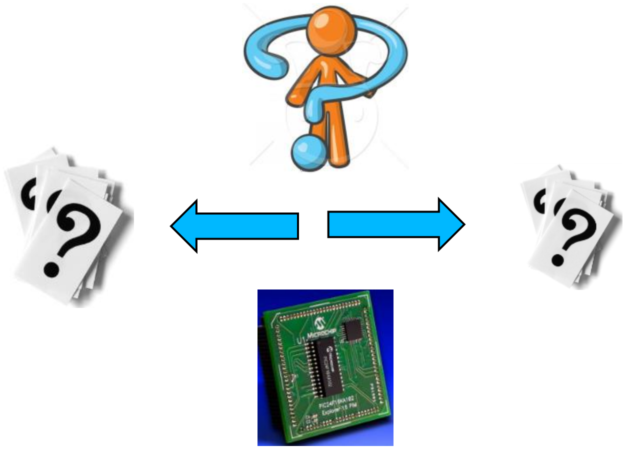
Adrian Dobrescu and Andrei Olaru

Outline

- Introduction
- Context Matching for Ambient Intelligence
- Related work
- Context Graphs and Patterns
- Adapting graph matching algorithms
- Integrating Regular Expressions
- Experiments and Results
- Conclusions

Introduction

- Ambient Intelligence (AmI) – a pervasive electronic environment to help people in their daily tasks



- A simple scenario: *Emily has to go to the store. Her personal agent detects that she is near the door and she forgot to take the key and the wallet before leaving.*

Context Matching for AmI

- Context – any information that characterize the situation of entities, relevant to the interaction of the users with an application. [Dey, 2001]
- Agents are able to produce context information and to exchage context information between them.
- Question: how to work with context information in a domain-independent manner?
  - Represent context as a graph
  - Represent situations as graph patterns – recognize situation, infere information, choose pro-active action
- Problem: matching patterns against the current context

Related Work

Context-Awareness in Ambient Intelligence

- AmI applications usually focused on specific environments
  - Smart homes, aspects of AAL, museum assistance
- Infrastructures for the processing of context information (Hong, Harter, Feng) favor a single direction of information transfer
- Context representation
  - models varying from tuples to logical representations
  - ontologies
  - Henricksen – rule-based reasoning using associations

Related Work

Graph Matching Algorithms

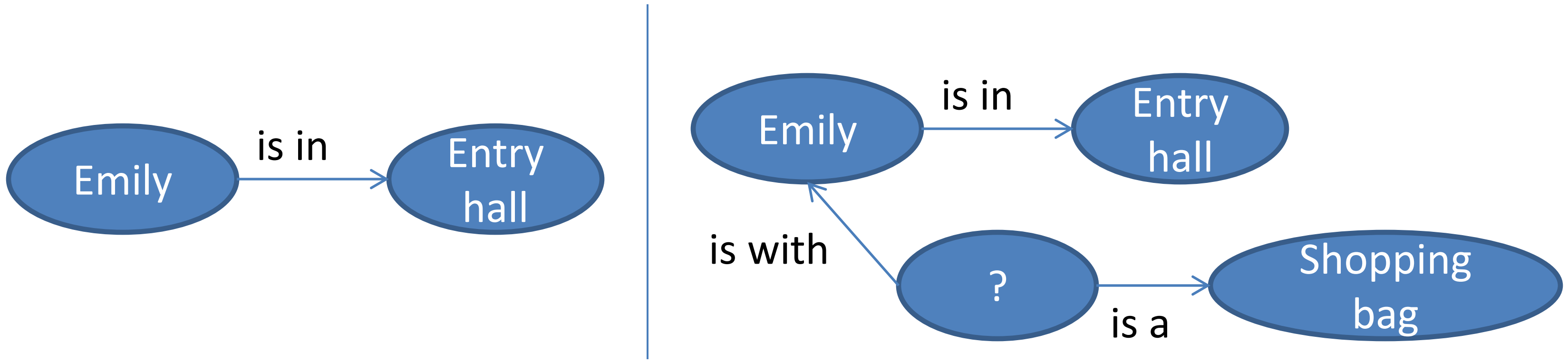
- Graph matching: NP-complete problem
  - Gaining interest at the end of 70s, now interesting again thanks to increased processing power
    - Used in image processing and pattern recognition
  - Exact and inexact graph matching algorithms
- Graphs are a representation that is flexible and adequate for dynamic changes of context
  - Graphs are used in semantic networks, concept maps, conceptual graphs

Context Graphs and Patterns

- Each agent has a context graph

  CG_A = (V, E), V ∈ Concepts, E = {(from, to, value), value ∈ Relations}

  – Concepts and Relations are strings and URIs
- Patterns = {(G_s^P) | s ∈ PatternNames, with G_s^P a graph pattern}
  – Can have question marks instead of nodes
  – Edges can be labeled regular expressions



Adapting Graph Matching Algorithms

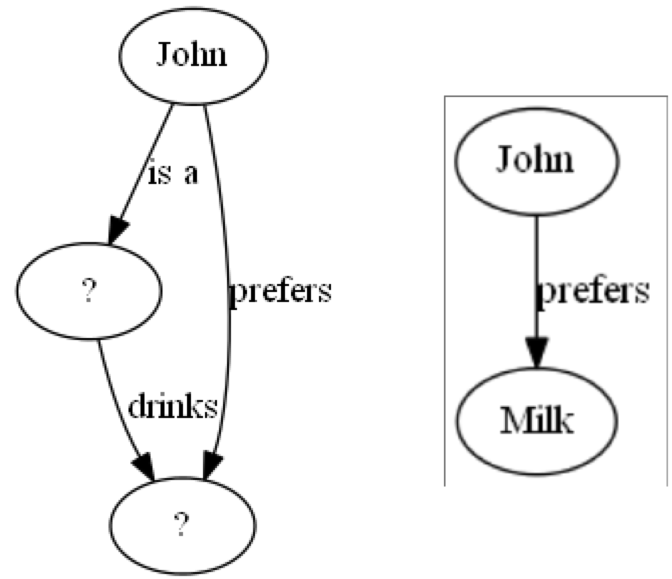
General comments

- Algorithms that we have studied:
  - McGregor, Larrosa, Akkoyunlu, Bron-Kerbosch, Balas-Yu, Koch, Durand-Pasari
- Nodes labeled with “?” can match any other label
- NP-hard problem
- Difficulties: undirected graphs, exact matchings, numerical graphs
- Algorithms: branch and bound, maximal clique finding, constraint satisfaction

Adapting Graph Matching Algorithms

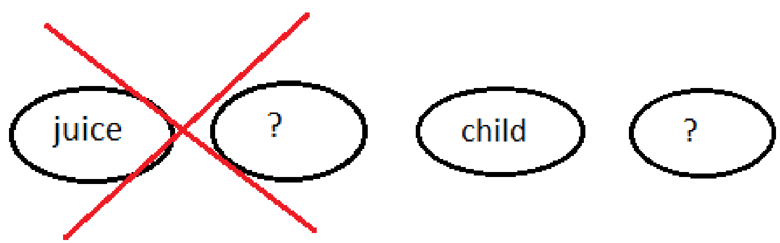
The McGregor algorithm [McGregor, 1982]

- Algorithm:
  - initial state - void
  - (v1, v2) or (v1, ?)
    - V1 has same label as v2
  - Test of possible extension
  - Maximal solution is saved
- Features:
  - Easy to implement
  - Provides partial solutions
  - Large execution time



Pattern

Partial solution



Adapting Graph Matching Algorithms

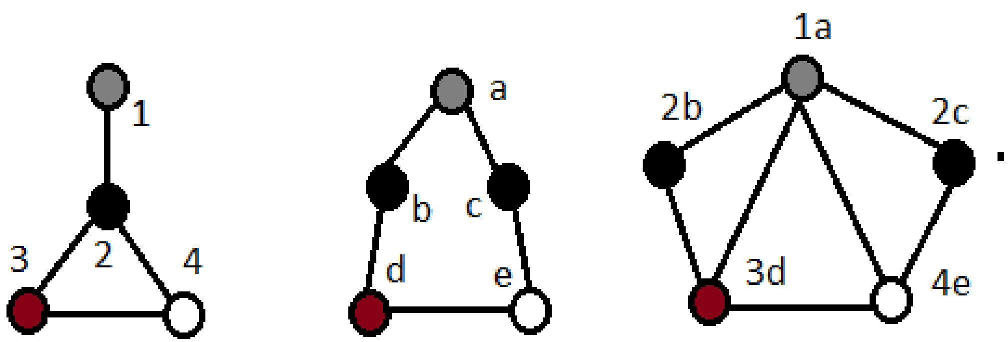
The Larrosa algorithm [Larrosa & Valiente, 2002]

- The fastest algorithm in our research
- It is only finding exact matches
  - It cannot solve the context-matching problem
- Constraint satisfaction problem, looking for connectivity with all the neighbours
- The values inside the domains are efficiently erased.
- Larrosa < full prediction < partial prediction

Adapting Graph Matching Algorithms

The Bron-Kerbosch algorithm [Bron & Kerbosch, 1973]

- Modular product -> associations graph



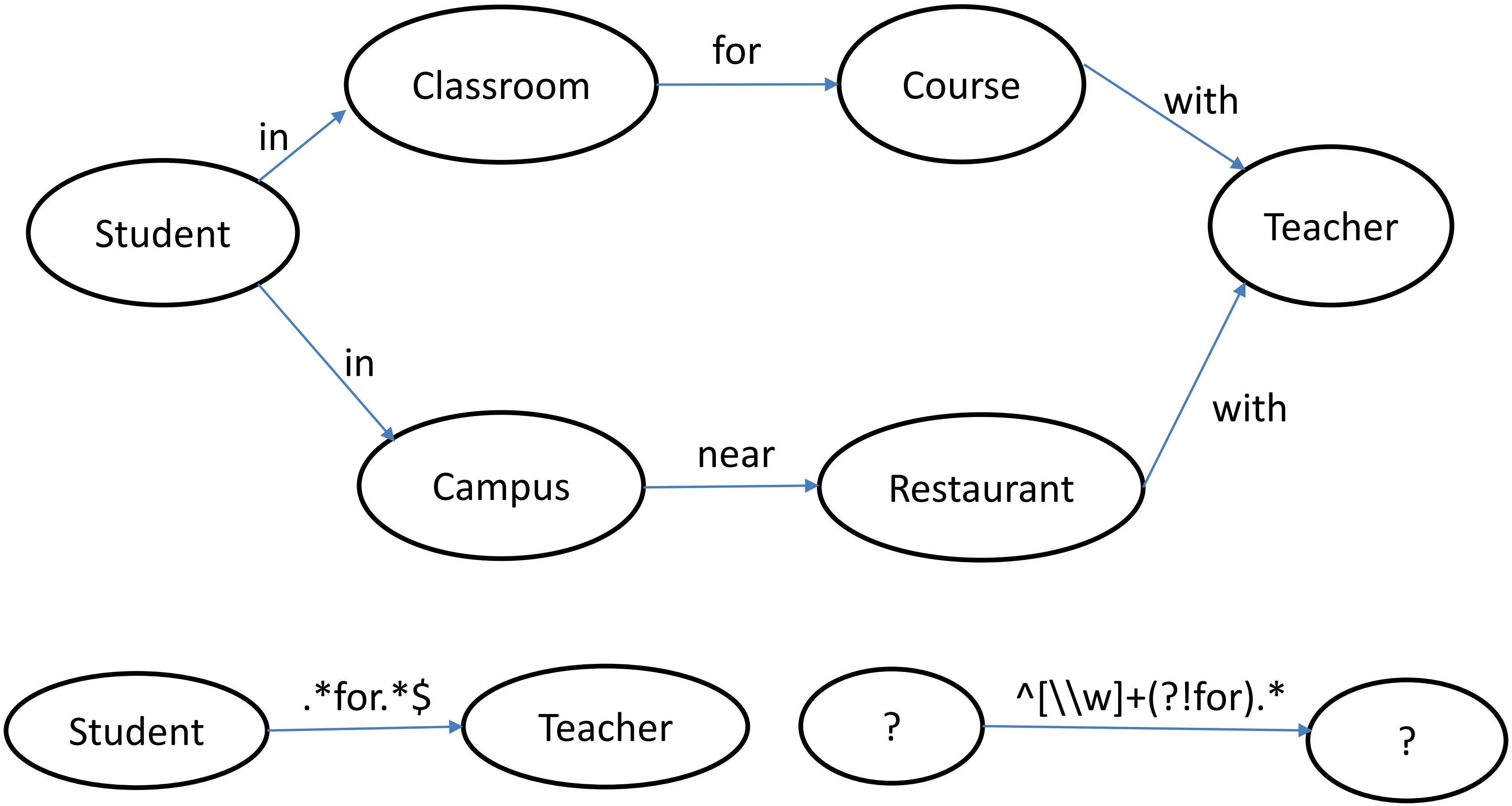
- Unknown nodes are expanded into all possible solutions
- Convert original graph to an unordered graph
- Finding the maximal clique = finding the maximal common subgraph
- Check if the solutions are valid on the original graph

Adapting Graph Matching Algorithms

Other clique algorithms

- Akkoyunlu [Akkoyunlu, 1973]
  - An improved version of the Bron-Kerbosch algorithm, considering vertex ordering.
- Durand-Pasari [Durand et al, 1999]
  - suplimentary validations in each step
- Koch [Koch, 2001]
  - Modular product built on edges
- Balas-Yu [Balas & Yu, 1986]
  - Heuristics for state-space prunning

Integrating regular expressions

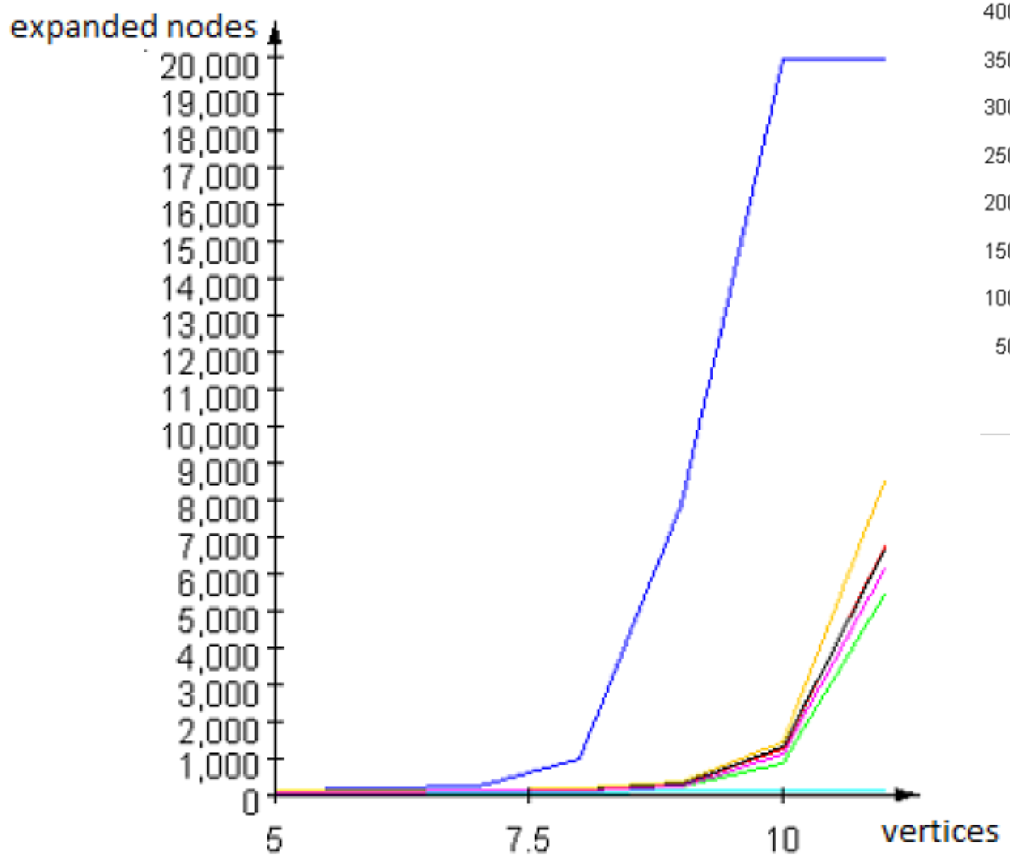


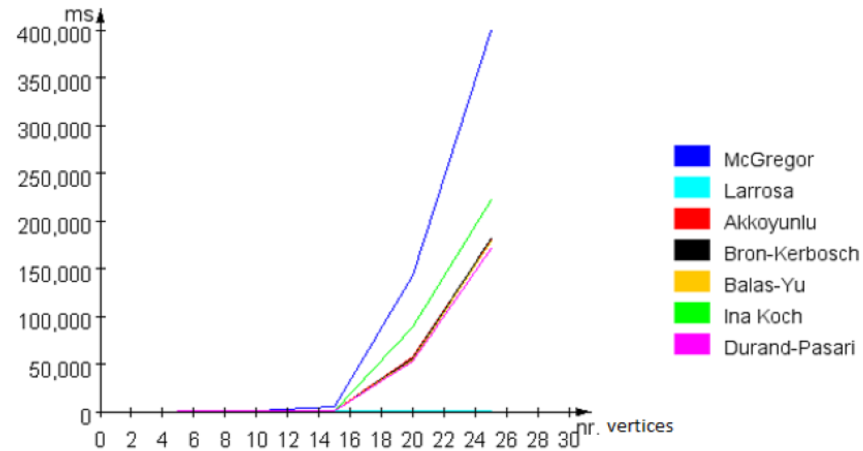
Experiments and Results

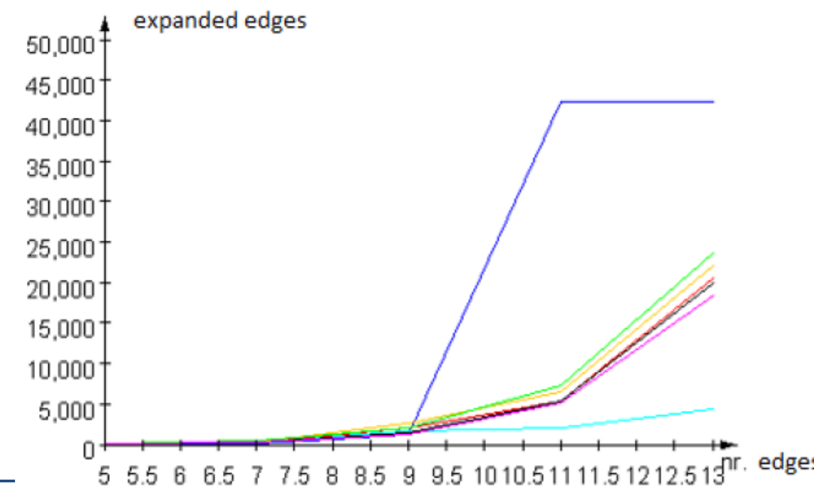
- An optimized backtracking procedure on smaller graphs is better than finding heuristics on bigger graphs.
- The McGregor algorithm can be a suitable solution for simple agents with small contexts.
- Clique algorithms are hard to implement and they require many transformations for the context graph problem.

Experiments and Results

Performance graphs







Experiments and Results

Numerical results

- Expanded edges

Edges	McGregor	Larrosa	Akkoyunlu	Bron-Kerbosch	Balas-Yu	Koch	Durand-Pasari
5	42	46	124	120	135	140	119
7	115	391	330	321	372	420	310
9	1250	1699	1995	1464	2542	2115	1330
11	42343	2108	5431	5440	6423	7345	5219
13	1976312	4324	20470	19989	22170	23575	18322

- Execution time

Vertices	McGregor	Larrosa	Akkoyunlu	Bron-Kerbosch	Balas-Yu	Koch	Durand-Pasari
5	14	1	9	7	23	10	9
10	82	11	22	34	42	36	30
15	5021	25	50	56	89	82	51
20	143023	36	56971	56392	54112	88523	52332
25	X	53	179434	181302	178342	221390	171000

Conclusions and Future Work

- Conclusions
  - An overview of the research conducted in graph matching
  - We have implemented and adapted several graph matching algorithms for the context matching problem
  - Matching context graphs can be a good answer to the problems that appears when dealing with context-awarness
- Future work
  - Improving efficiency in real time problems
  - Deployment on real devices with capabilities of interaction between different machines
  - Dealing with uncertain knowledge

Thank you

Any questions?