Karlsruhe Institute of Technology

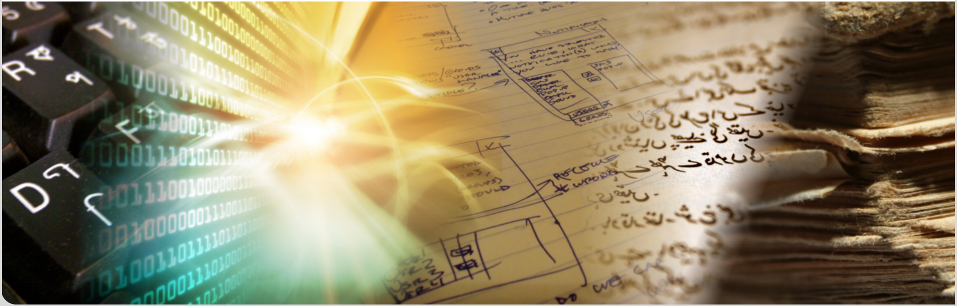
Fakultät für Informatik

Bachelor’s Thesis
Flexible User-Friendly Trip Planning Queries

Adviser: Saeed Taghizadeh

Violina Zhekova | May 9, 2019

INSTITUTE FOR PROGRAM STRUCTURES AND DATA ORGANIZATION (IPD), CHAIR PROF. BÖHM



Outline

1. Motivation
2. Related Work
3. Example
4. Problem Definition
5. Introducing the Operators
   - Equality operator
   - Inequality
   - Or
   - Order
6. Evaluating the operators
   - Equality
   - Inequality
   - Or
   - Order
7. Conclusion
8. Discussion

Motivation

- *Sequenced Route Queries* (SRQ) - finding routes passing through multiple *Points of Interest* (PoIs)
- Advances in *Location Based Services* (LBS) and *Geographic Information System* (GIS) applications (e.g. logistics and supply chain management)
- Aim: Designing a language to enable the user to express his query requirements in a flexible manner

Motivation

- *Sequenced Route Queries* (SRQ) - finding routes passing through multiple *Points of Interest* (PoIs)
- Advances in *Location Based Services* (LBS) and *Geographic Information System* (GIS) applications (e.g. logistics and supply chain management)
- Aim: Designing a language to enable the user to express his query requirements in a flexible manner

Motivation

- *Sequenced Route Queries* (SRQ) - finding routes passing through multiple *Points of Interest* (PoIs)
- Advances in *Location Based Services* (LBS) and *Geographic Information System* (GIS) applications (e.g. logistics and supply chain management)
- Aim: Designing a language to enable the user to express his query requirements in a flexible manner

Related work

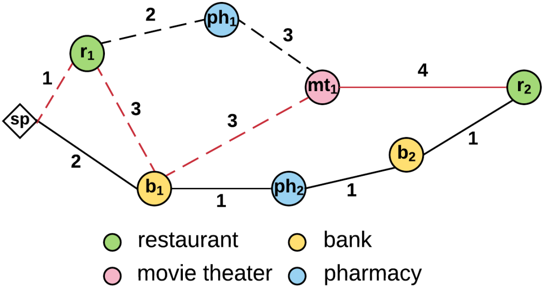
- Vector vs. metric spaces
- *Trip Planning Queries* (TPQ)
- The *Optimal Sequenced Route* (OSR) Query
- The Skylyne concept applied to SRQ
- Considering multiple factors of a route – rating of PoIs, distance and category weights, dynamic factors (e.g. traffic information)
- SRQ issued by users moving along a route
- *Multi-rule Partial Sequenced Route* (MRPSR) Query

Example

- **Category sequence**: (restaurant, bank, movie theater, restaurant)
  Condition: equal restaurants
- *Optimal Sequenced Route* (OSR): (r_1, b_1, mt_1, r_2), length: 11 (shown with red lines)
- Optimal route with equal restaurants: (r_1, b_1, mt_1, r_1), length: 12 (shown with dashed lines)

Example

- **Category sequence**: (restaurant, bank, movie theater, restaurant)
  Condition: equal restaurants
- *Optimal Sequenced Route* (OSR): (r_1, b_1, mt_1, r_2), length: 11 (shown with red lines)
- Optimal route with equal restaurants: (r_1, b_1, mt_1, r_1), length: 12 (shown with dashed lines)



Problem definition

Karlsruhe Institute of Technology

- **Problem**: Need for flexibility in route finding queries
- **Solution**: Developing a query language operators to fulfill the essential user's requirements:
  - Relationships among the PoIs
  - Order and priority of the PoIs
  - Expressing multiple travel variations
- Proposing four essential operators: "equality" operator, "inequality" operator, "or" operator, "order" operator
- Making use of existing approaches (PNE (*Progressive Neighbor Exploration*)) to transform the complex user query

Problem definition

Karlsruhe Institute of Technology

- **Problem**: Need for flexibility in route finding queries
- **Solution**: Developing a query language operators to fulfill the essential user's requirements:
  - Relationships among the PoIs
  - Order and priority of the PoIs
  - Expressing multiple travel variations

- Proposing four essential operators: ”equality” operator, ”inequality” operator, ”or” operator, ”order” operator
- Making use of existing approaches (PNE (*Progressive Neighbor Exploration*)) to transform the complex user query

PNE Algorithm

Algorithm 1: PNE

fetch a *PSR* from the heap;
switch $s = size(PSR)$ **do**
    case $s == l$ **do**
        PSR is the optimal route;
        return *PSR*;
    case $s \neq l$ **do**
        a)
            $\mathtt{NN}(r_{|PSR|}, C_{M_{|PSR|+1}})$;
            update *PSR* and perform trimming in case it is a candidate SR ;
            put *PSR* back on the *heap* ;
        b)
            $\mathtt{kNN}(r_{|PSR|-1}, C_{M_{|PSR|}})$;
            generate a new *PSR* and place it on the *heap*;

Equality operator

- *Input*: A category sequence $M = (c_1, c_2, ..., c_l)$, a starting point *sp* in $\mathbb{R}^2$ and indices of the equal PoIs i and j, where $c_i = c_j$
- *Output*: Optimal route $R = (r_1, r_2, ..., r_l)$, where $r_i = r_j$

- **Proposed approach**: uses the Progressive Neighbor Explorator (PNE) as its base to upgrade on and extends it with a heuristic approach to shrink the search space
- **Baseline/trivial approach**: extends the PNE with forcing the PoIs r_i and r_j to be equal; does not use optimization techniques

Heuristic

Given a sequence of categories $M = (c_1, c_2, ..., c_l)$ and a PSR $R' = (r_1, r_2, ..., r_k)$ the ***heuristic*** for this route is defined as:

$$h(R') = \max_{i \in [k+1, l]} nearestNeighbor(r_k, C_{M_i}) \tag{1}$$

- Informal: The heuristic of a certain PSR is the maximum distance out of the distances to the nearest PoIs from the set of categories that are yet to be expanded.

The ***lower bound*** of a PSR R' represents the sum of its length and its heuristic:

$$LB(R') = length(R') + h(R') \tag{2}$$

- The proposed algorithm uses a heap, sorted by the lower bound of the routes

Heuristic

Given a sequence of categories $M = (c_1, c_2, ..., c_l)$ and a PSR $R' = (r_1, r_2, ..., r_k)$ the ***heuristic*** for this route is defined as:

$$h(R') = \max_{i \in [k+1, l]} nearestNeighbor(r_k, C_{M_i}) \quad (1)$$

- Informal: The heuristic of a certain PSR is the maximum distance out of the distances to the nearest PoIs from the set of categories that are yet to be expanded.

The ***lower bound*** of a PSR R' represents the sum of its length and its heuristic:

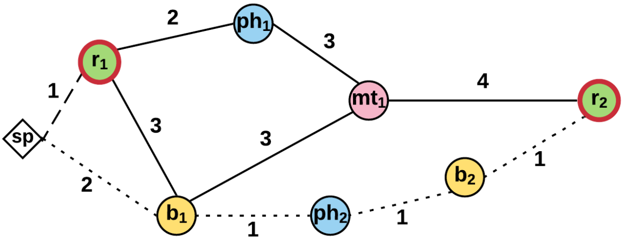
$$LB(R') = length(R') + h(R') \quad (2)$$

- The proposed algorithm uses a heap, sorted by the lower bound of the routes

Example: Step 1

$M = (r, b, mt, r), EQUAL(0, 3)$

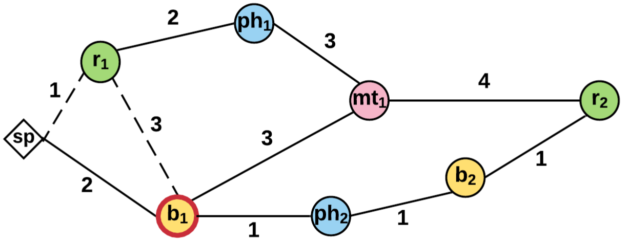
- Optimal route found with PNE: (r_1, b_1, mt_1, r_2)
- Dummy SR: (r_1, b_1, mt_1, r_1); Upper Bound $UB = length(dummySR)$



Step	Heap contents (PSR R : $length(R), heuristic(R)$)
1	$(r_1 : 1, 5), (r_2 : 5, 4)$

Example: Step 2

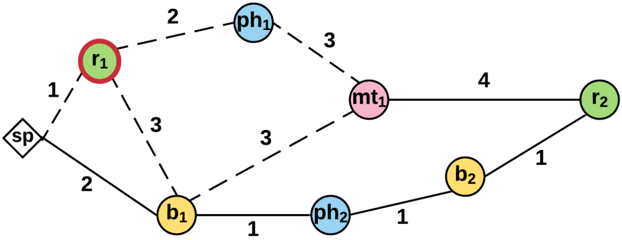
$M = (r, b, mt, r)$, $EQUAL(0, 3)$



Step	Heap contents (PSR R : $length(R)$, $heuristic(R)$)
1	$(r_1 : 1, 5)$, $(r_2 : 5, 4)$
2	$(r_1, b_1 : 4, 3)$, $(r_2 : 5, 4)$

Example: Step 8

$M = (r, b, mt, r)$, $EQUAL(0, 3)$

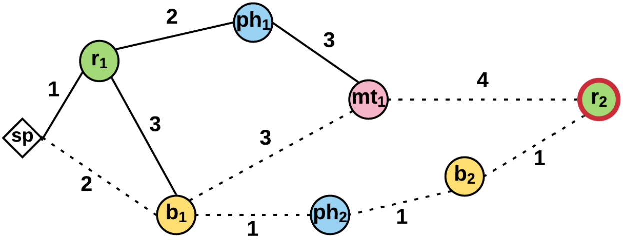


Candidate SR: $(r_1, b_1, mt_1, r_1 : 12, 0)$

Step	Heap contents (PSR R : $length(R)$, $heuristic(R)$)
7	$(r_1, b_1, mt_1 : 7, 5), (r_2, b_1, mt_1 : 11, 4), (r_2, b_2, mt_1 : 11, 4), (r_1, b_2, mt_1 : 11, 5)$
8	$(r_2, b_1, mt_1 : 11, 4), (r_2, b_2, mt_1 : 11, 4), (r_1, b_2, mt_1 : 11, 5)$

Example: Step 9

$M = (r, b, mt, r), EQUAL(0, 3)$



Candidate SR: $(r_2, b_1, mt_1, r_2 : 15, 0)$

Step	Heap contents (PSR R : $length(R)$, $heuristic(R)$)
8	$(r_2, b_1, mt_1 : 11, 4), (r_2, b_2, mt_1 : 11, 4), (r_1, b_2, mt_1 : 11, 5)$
9	$(r_2, b_2, mt_1 : 11, 4), (r_1, b_2, mt_1 : 11, 5)$

Inequality

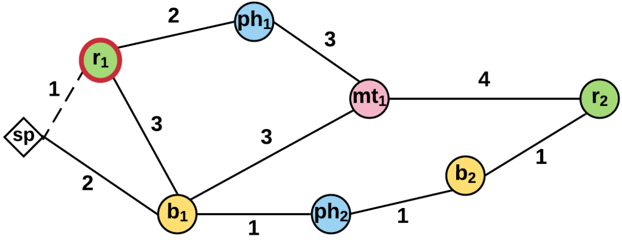
- *Input*: A category sequence $M = (c_1, c_2, ..., c_l)$, a starting point *sp* in $\mathbb{R}^2$ and indices of the unequal PoIs i and j, where $c_i = c_j$
- *Output*: Optimal route $R = (r_1, r_2, ..., r_l)$, where $r_i \neq r_j$

- **Proposed approach**: based on the Progressive Neighbor Explorator (PNE)

Example: Step 1

$M = (r, ph, r), UNEQUAL(0, 2)$

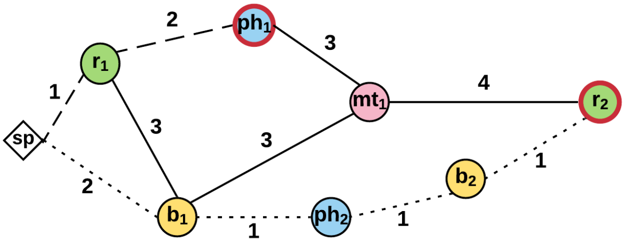
- Optimal route found with PNE: (r_1, ph_1, r_1)



Step	Heap contents (PSR R : $length(R)$)
1	$(r_1 : 1)$

Example: Step 2

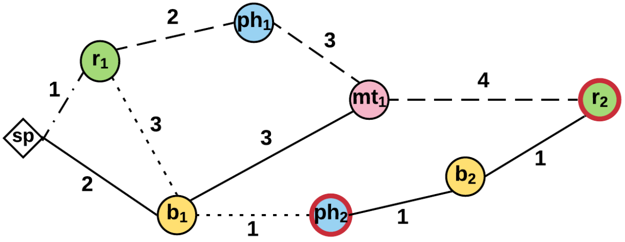
$M = (r, ph, r), UNEQUAL(0, 2)$



Step	Heap contents (PSR R : $length(R)$)
1	$(r_1 : 1)$
2	$(r_1, ph_1 : 3), (r_2 : 5)$

Example: Step 3

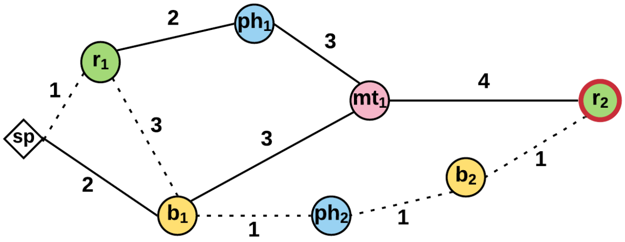
$M = (r, ph, r), UNEQUAL(0, 2)$



Step	Heap contents (PSR R : $length(R)$)
2	$(r_1, ph_1 : 3), (r_2 : 5)$
3	$(r_2 : 5), (r_1, ph_2 : 5), (r_1, ph_1, r_2 : 10)$

Example: Step 5

$M = (r, ph, r)$, $UNEQUAL(0, 2)$



Step	Heap contents (PSR R : $length(R)$)
4	$(r_1, ph_2 : 5), (r_2, ph_2 : 7), (r_1, ph_1, r_2 : 10)$
5	$(r_1, ph_2, r_2 : 7), (r_2, ph_2 : 7), \cancel{(r_1, ph_1, r_2 : 10)}$

Or

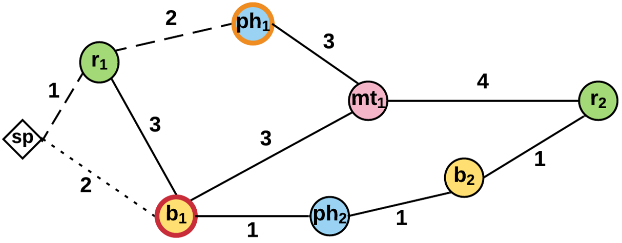
- **OR sequence:** An OR sequence $OR = (M_1, M_2, ..., M_m)$ represents the disjunction of category sequences, such as $M_1 = (c_1, c_2, ..., c_l)$.

- *Input*: A sequence of OR sequences $S = (OR_1, OR_2, ..., OR_n)$ and a starting point *sp* in $\mathbb{R}^2$
- *Output*: Optimal route $R = (r_1, r_2, ..., r_l)$

- **Proposed approach**: progressively inspects each option M_i from the OR sequences OR_i in $S = (OR_1, OR_2, ..., OR_n)$, compares them and continues with the best one, based on length, until it reaches a full sequenced route
- **Proposed approach**: runs the PNE algorithm on all possible combinations of the query to find the shortest route out of them

Example: Step 1

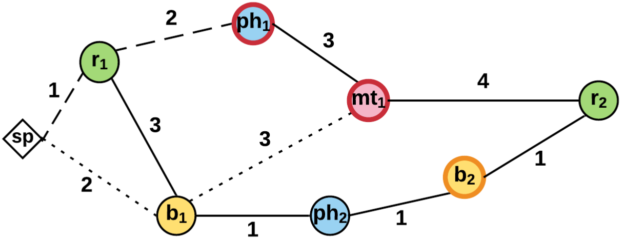
$S = (OR_1, OR_2, OR_3)$, $OR_1 = ((b), (ph))$, $OR_2 = ((mt))$, $OR_2 = ((r))$



Step	Heap contents (PSR R : $length(r)$, $index(R)$)
1	$(b_1 : 2, 1)$

Example: Step 2

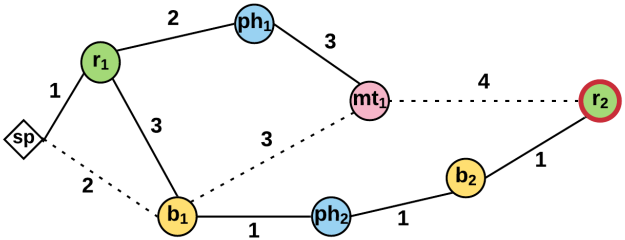
$S = (OR_1, OR_2, OR_3)$, $OR_1 = ((b), (ph))$, $OR_2 = ((mt))$, $OR_2 = ((r))$



Step	Heap contents (PSR R : $length(r)$, $index(R)$)
1	$(b_1 : 2, 1)$
2	$(ph_1 : 3, 1)$, $(b_1, mt_1 : 5, 2)$

Example: Step 6

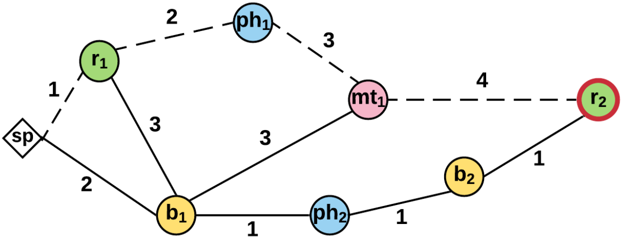
$S = (OR_1, OR_2, OR_3)$, $OR_1 = ((b), (ph))$, $OR_2 = ((mt))$, $OR_2 = ((r))$



Step	Heap contents (PSR R : $length(r), index(R)$)
5	$(b_1, mt_1 : 5, 2), (ph_1, mt_1 : 6, 2), (ph_2, mt_1 : 7, 2), (b_2, mt_1 : 9, 2)$
6	$(ph_1, mt_1 : 6, 2), (ph_2, mt_1 : 7, 2), (b_1, mt_1, r_2 : 9, 3), (b_2, mt_1 : 9, 2)$

Example: Step 7

$S = (OR_1, OR_2, OR_3)$, $OR_1 = ((b), (ph))$, $OR_2 = ((mt))$, $OR_2 = ((r))$



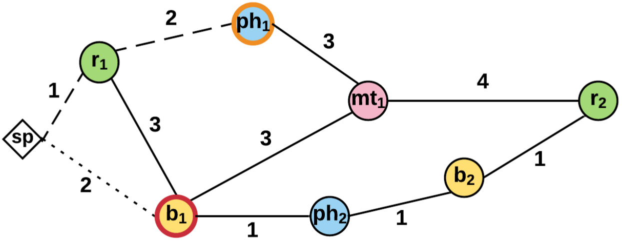
Step	Heap contents (PSR R : $length(r), index(R)$)
6	$(ph_1, mt_1 : 6, 2), (ph_2, mt_1 : 7, 2), (b_1, mt_1, r_2 : 9, 3), (b_2, mt_1 : 9, 2)$
7	$(ph_2, mt_1 : 7, 2), (b_1, mt_1, r_2 : 9, 3), (b_2, mt_1 : 9, 2)$, $\cancel{(ph_1, mt_1, r_2 : 10, 3)}$
8	$(b_1, mt_1, r_2 : 9, 3), (b_2, mt_1 : 9, 2)$, $\cancel{(ph_2, mt_1, r_2 : 11, 3)}$

Order

- **ORDER sequence:** An order sequence $ORDER = (i_1, i_2, ..., i_k)$, is a sequence of indices in a category sequence $M_1 = (c_1, c_2, ..., c_l)$, which indicate that the categories at the given indices should remain in the fixed positions in this category sequence.
- $NOTORDERED = \overline{ORDER}$
- *Input*: A sequence of categories $M = (c_1, c_2, ..., c_l)$, a starting point sp in $\mathbb{R}^2$ and an ORDER sequence $ORDER = (i_1, i_2, ..., i_k)$
- *Output*: Optimal route $R = (r_1, r_2, ..., r_l)$
- **Proposed approach**: inspects progressively each category option for the indices out of the NOTORDERED sequence, compares them and continues with the best one, based on length, until it reaches a full sequenced route.
- **Proposed approach**: runs the PNE algorithm on all possible permutations of the query to find the shortest route out of them

Example: Step 1

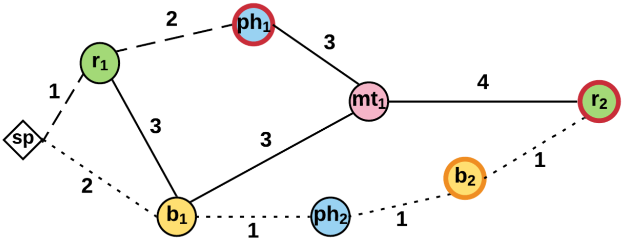
$M = (b, r, ph)$, $ORDER = (1)$



Step	Heap contents (PSR R : $length(r), r.notordered$)
1	$(b_1 : 2, [ph])$

Example: Step 1

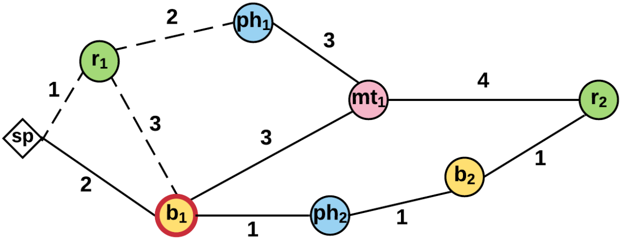
$M = (b, r, ph)$, $ORDER = (1)$



Step	Heap contents (PSR R : $length(r), r.notordered$)
1	$(b_1 : 2, [ph])$
2	$(ph_1 : 3, [b]), (b_1, r_2 : 5, [ph])$

Example: Step 6

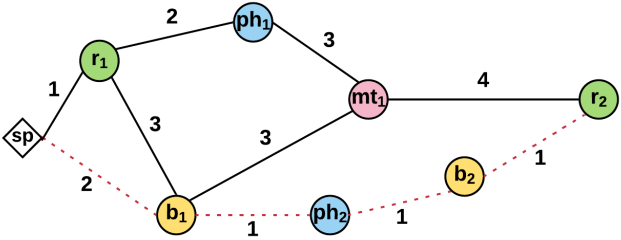
$M = (b, r, ph)$, $ORDER = (1)$



Step	Heap contents (PSR R : $length(r), r.notordered$)
5	$(ph_1, r_1 : 5, [b]), (b_2, r_2 : 5, [ph]), (ph_2, r_2 : 5, [b]), (b_1, r_2 : 5, [ph])$
6	$(b_2, r_2 : 5, [ph]), (ph_2, r_2 : 5, [b]), (b_1, r_2 : 5, [ph])$, $(ph_1, r_1, b_1 : 8, [])$

Example: Step 12

$M = (b, r, ph)$, $ORDER = (1)$



Step	Heap contents (PSR R : $length(r)$, $r.notordered$)
10	$(b_2, r_2 : 5, [ph]), (ph_2, r_2, b_2 : 6, []), \cancel{(b_1, r_1, ph_1 : 7, [])}, (ph_2, r_1 : 7, [b]), (b_2, r_1 : 9, [ph]), (ph_1, r_2 : 10, [b])$
11	$(ph_2, r_2, b_2 : 6, []), (ph_2, r_1 : 7, [b]), (b_2, r_1 : 9, [ph]), (ph_1, r_2 : 10, [b]), \cancel{(b_2, r_2, ph_1 : 12, [])}$

Example

Step	Heap contents (PSR R : $length(r), r.notordered$)
1	$(b_1 : 2, [ph])$
2	$(ph_1 : 3, [b]), (b_1, r_2 : 5, [ph])$
3	$(ph_2 : 3, [b]), (ph_1, r_1 : 5, [b]), (b_1, r_2 : 5, [ph])$
4	$(b_2 : 4, [ph]), (ph_2, r_2 : 5, [b]), (ph_1, r_1 : 5, [b]), (b_1, r_2 : 5, [ph])$
5	$(ph_1, r_1 : 5, [b]), (b_2, r_2 : 5, [ph]), (ph_2, r_2 : 5, [b]), (b_1, r_2 : 5, [ph])$
6	$(b_2, r_2 : 5, [ph]), (ph_2, r_2 : 5, [b]), (b_1, r_2 : 5, [ph]), (ph_1, r_1, b_1 : 8, [])$
7	$(b_1, r_2 : 5, [ph]), (b_2, r_2, ph : 5, [ph]), (ph_2, r_2 : 5, [b]), (b_2, r_2, ph_2 : 7, []), \cancel{(ph_1, r_1, b_1 : 8, [])}, (b_2, r_1 : 9, [ph]), (ph_1, r_2 : 10, [b])$
8	$(b_2, r_2 : 5, [ph]), (ph_2, r_2 : 5, [b]), (b_1, r_1 : 5, [ph]), (b_2, r_2, ph_2 : 7, []), \cancel{(b_1, r_2, ph_2 : 7, [])}, (b_2, r_1 : 9, [ph]), (ph_1, r_2 : 10, [b])$
9	$(b_1, r_1 : 5, [ph]), (b_2, r_2 : 5, [ph]), (ph_2, r_2, b_2 : 6, []),$ [illegible]

Equality

Inequality

Or

Order

Conclusion

Discussion and Future Work

Literatur I

- Didier Guzzni, Adam Cheyer, Charles Baur. *Active, a platform for building intelligent software*. IEEE Web Intelligence and Intelligent Agent Technology Workshops, 2006.
- Alexander Wachtel, Jonas Klamroth, Walter F. Tichy. *A Natural Dialog System Based on Active Ontologies*. Annalen der Physik, 322(10):891–921, 1905.
- Steven Levy. *An exclusive inside look at how artificial intelligence and machine learning work at Apple*. https://www.wired.com/2016/08/an-exclusive-look-at-how-ai
- Informationen zum Sicherheitsinhalt von iOS 7.1.2. https://support.apple.com/de-de/HT203014
- Bernadette Johnson. *How Siri Works*. http://electronics.howstuffworks.com/gadgets/high-tech-ga

Literatur II

- Stasys Bielinis. *How Siri on iPhone 4S works and why it's a big deal. Apple's AI tech details in 230 pages of patent app*. http://www.unwiredview.com/2011/10/12/how-siri-on-iphone-
- Johan Schalkwyk, Doug Beeferman, Francoise Beaufays, Bill Byrne, Ciprian Chelba, Mike Cohen, Maryam Garret, Brian Strope. *Google Search by Voice: A case study*.
- Ciprian Chelba, Dan Bikel, Maria Shugrina, Patrick Nguyen, Shankar Kumar. *Large Scale Language Modeling in Automatic Speech Recognition*.
- Voice Search https://www.xovi.de/wiki/Voice_Search

Literatur III

- Matthew Lynley. *Google unveils Google Assistant, a virtual assistant that's a big upgrade to Google Now*. https://techcrunch.com/2016/05/18/google-unveils-google-a
- Get notifications from Google Now in Chrome https://chrome.googleblog.com/2014/02/get-notifications-f
- Use Google Now on tap https://support.google.com/websearch/answer/6304517?hl=de
- Mariella Moon. *Google Voice Search can now handle multiple languages with ease*. https://www.engadget.com/2014/08/15/google-voice-search-m
- Eric Ravenscraft. *Use Some Google Now Voice Commands Without an Internet Connection*. http://lifehacker.com/use-some-google-now-voice-commands-

Literatur IV

- Meet your Google Assistant
https://assistant.google.com/
- Joe Belfiore. *Windows Phone 8.1 Update brings Cortana to new markets + new features*.
https://blogs.windows.com/windowsexperience/2014/07/30/wi
- Mary Jo Foley. *Microsoft gears up for Cortana on Android preview*.
http://www.zdnet.com/article/microsoft-gears-up-for-corta
- Brad Sams. *Windows 10: Cortana now syncs reminders*.
https://www.neowin.net/news/windows-10-cortana-now-syncs-
- Virginia Backaitis. *Why Microsoft's Cortana is 14 for 14 Calling World Cup Matches*.
http://www.cmswire.com/cms/big-data/why-microsofts-cortan

Literatur V

- Tina Sieber. *Here's how to make the most of Cortana, the Windows 10 digital assistant*.
  https://www.digitaltrends.com/computing/get-know-cortana-
- Brad Sams. *Cortana Can Now Sync Notifications From Your Android Phone To Your Windows 10 PC*.
  https://www.thurrott.com/windows/windows-10/67251/cortana
- Daniel Rubino. *Hands on with 'Hey Cortana' and the Lumia 930 Denim update*
  https://www.windowscentral.com/hands-on-hey-cortana-video
- Cortana's regions and languages
  https://support.microsoft.com/en-gb/instantanswers/557b5e

Literatur VI

- R. Sarikaya, P.A. Crook, A. Martin, M. Jeong, J.P. Robichaud, A. Celikyilmaz, Y.B. Kim, A. Rochette, O.Z. Khan, X. Liu, D. Boies, T. Anastasakos, Y. Feizollahi, N.Ramesh, H. Suzuki, R. Holenstein, E. Krawczyk, V. Radostev. *An overview of end-to-end language understanding and dialog management for personal digital assistants*.
- Chris Hoffman. *15 Things You Can Do With Cortana on Windows 10*. `https://www.howtogeek.com/225458/15-things-you-can-do-wit`
- Tom Warren. *1The story of Cortana, Micrsoft's Siri killer*. `https://www.theverge.com/2014/4/2/5570866/cortana-windows`
- Five Things You Didn't Know About Cortana, Microsoft's Virtual Assistant `https://www.recode.net/2014/9/22/11631126/five-things-you`

Literatur VII

- Melissa Riofrio. *Cortana's UI now expresses 18 different emotions. Siri remains detached and aloof*
  `http://www.pcworld.com/article/2881902/cortanas-ui-now-ex`
- Microsoft takes aim at Alexa with Cortana Skills Kit
  `https://www.engadget.com/2017/05/10/stub-microsoft-takes-`
- Steve Dent. *Die Digitalen Assistenten im Vergleich – Teil 2 unserer Miniserie*.
  `http://www.paksafe.de/blog/die-digitalen-assistenten-im-v`
- Adrianne Jeffries *Google's featured snippets are worse than fake news*.
  `https://theoutline.com/post/1192/google-s-featured-snippe`
- Peter Stelzel-Morawietz. *Was taugen Siri, Cortana und Google Now?*
  `https://www.pcwelt.de/ratgeber/Spracherkennung-Siri-Corta`

Literatur VIII

- Eric Enge *The Great Knowledge Box Showdown: Google Now vs. Siri vs. Cortana*. https://www.stonetemple.com/great-knowledge-box-showdown/
- Jason Parker. *Google Search vs. Siri: Voice search speed test*. https://www.cnet.com/news/google-search-vs-siri-voice-sea
- Mehdi Assefi, Guangchi Liu, Mike P. Wittie, Clemente Izurieta. Department of Computer Science, Montana State University. *An Experimental Evaluation of Apple Siri and Google Speech Recognition*.
- Britta O'Boyle, Chris Hall. *Siri vs Cortana: Which is the funniest assistant?*. http://www.pocket-lint.com/news/136238-siri-vs-cortana-wh