Optimized, Bottom-Up Semantic Web Reasoning based on OWL2 RL in Resource-Constrained Settings

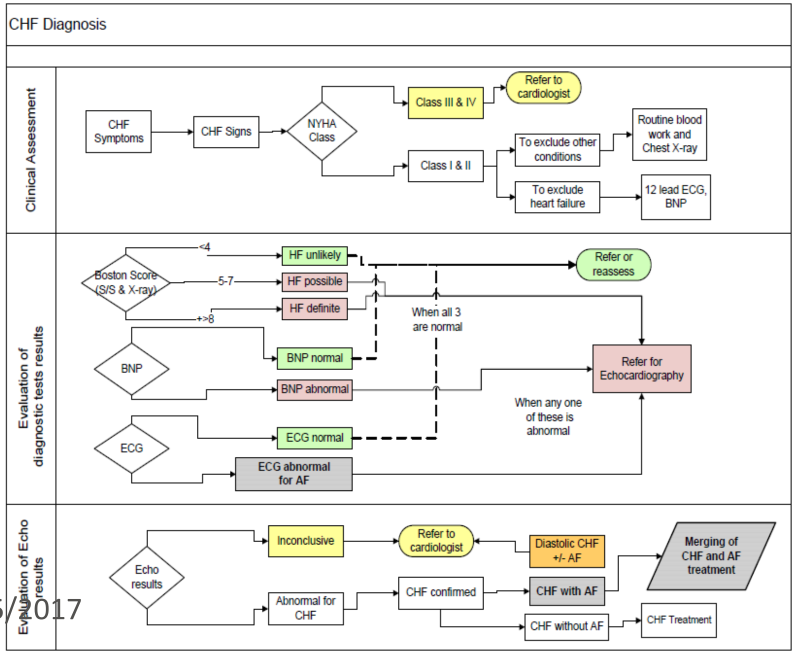
William Van Woensel

DALHOUSIE UNIVERSITY

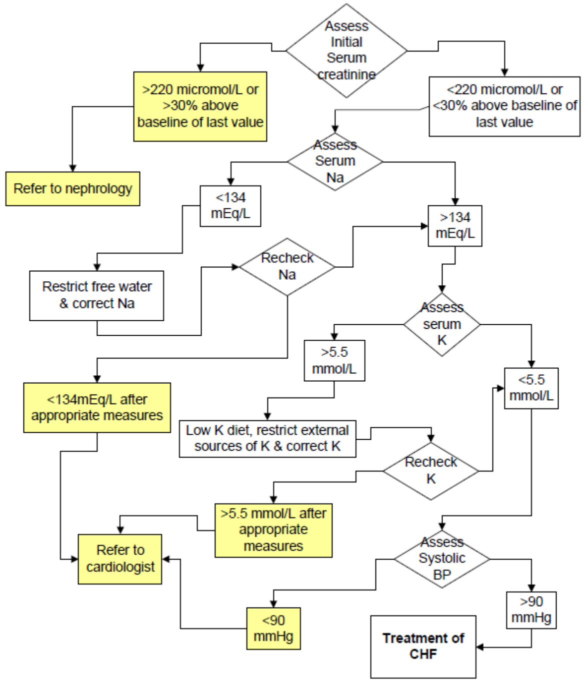
Context

- Clinical Practice Guidelines (CPG)
  - Disease-specific, evidence-based recommendations
  - Standard for decision making on diagnosis, prognosis and treatment
  - a) Context-sensitive care recommendations
  - b) Clinical workflow of relevant clinical activities

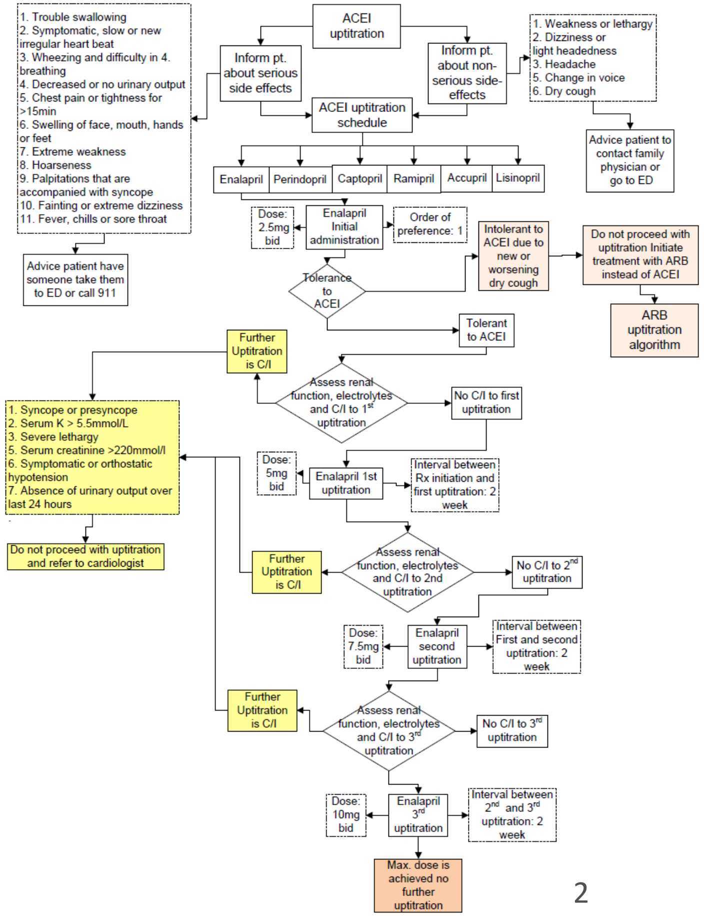
Algorithm for diagnosis of heart failure [1]



Pre-treatment assessment and correction of electrolytes [1]



ACEI upitration [1]



Context (2)

1) Clinical Decision Support Systems (CDSS)
   - Automated systems that incorporate computerized CPG
   - Pro-actively guide physician through decision processes

➢ *Decision Logic (OWL2 DL), IF-THEN (SWRL) rules, ..*

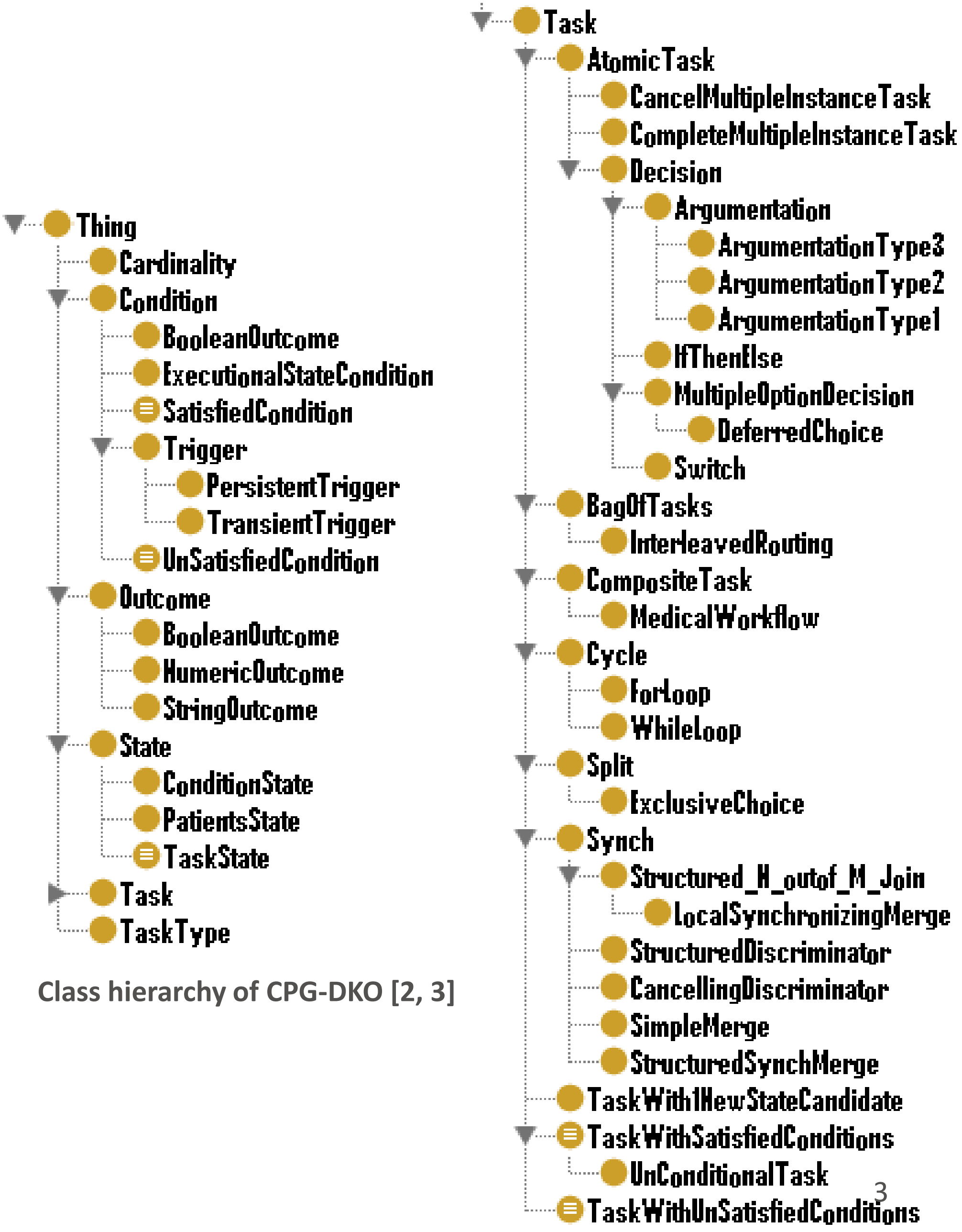
Switch [2, 3]:

$$SatisfiedCondition \sqcap \exists conditionOf.ActiveTask \sqcap \exists leadsTo.InactiveTask \sqcap \forall lessPriorityThan.UnsatisfiedCondition \sqsubset ChosenCondition$$

$$UnsatisfiedCondition \sqcap \exists conditionOf.ActiveTask \sqcap \exists leadsTo.InactiveTask \sqsubset DiscardedCondition$$

$$SatisfiedCondition \sqcap \exists conditionOf.ActiveTask \sqcap \exists leadsTo.InactiveTask \sqcap lessPriorityThan.SatisfiedCondition \sqsubset DiscardedCondition$$

…



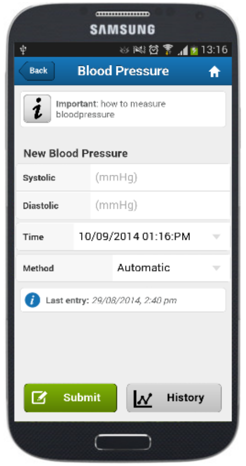
Class hierarchy of CPG-DKO [2, 3]

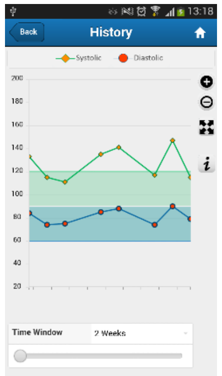
Context (2)

1) Clinical Decision Support Systems (CDSS)
   - Automated systems that incorporate computerized CPG
   - Pro-actively guide physician through decision processes

2) Involve patients in their own long-term care
   - Canadian Community Health Survey (2014):
     - Chronic illnesses affect ca. 40% of Canadians
     - With multi-morbidity of ca. 15%
   - Increase self-sufficiency and quality of life
   - Reduce healthcare costs

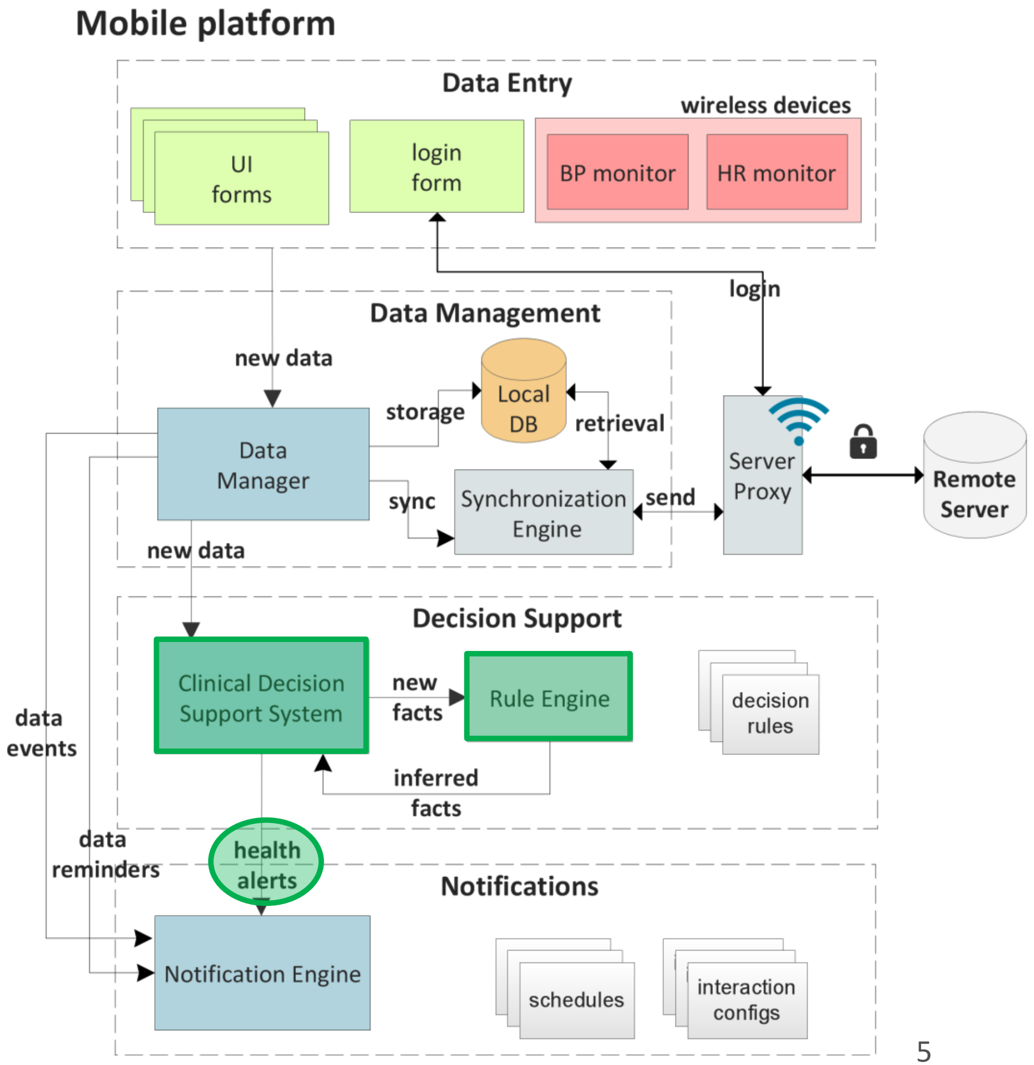
- Mobile patient diaries
  - IMPACT-AF project
  - Self-collect health data at any time and place
    - Using Bluetooth measurement devices (e.g., IBGStar, OneTouch, Withings, iHealth)
  - ✓ Increase mobility of chronic patients
  - ✓ Up-to-date health profile
  - ✓ No delays in supplying health-critical info





Context (3)

- Requirements:
  - Connectivity
    - Cope with short/long-term disconnections (lack of WiFi, 3G)
    - Should not limit mobile patient diary usage
  - Response latency
    - Slow / lacking connectivity may occur frequently
    - Server = *single point of failure*
- Solutions:
  - Offline data entry (*BP, HR, ..*)
    - Synchronize with online EMR when connectivity is restored
  - **Local Clinical Decision Support System**
    - ✓ Independent of connectivity
    - ✓ Enables timely health alerts
  - **Distributed setup**
    - **Local:** lightweight, time-sensitive reasoning is deployed locally
    - **Remote**: heavyweight processes are delegated to the server



Context (4)

- Ontology-based (OWL) reasoning
  - **OWL2 DL**: too resource-intensive on mobile systems
  - Recent empirical work by Bobed et al. [4]:
    - PC outperforms Android by **1,5 – 150**
    - Larger number of out-of-memory errors
  - Most mobile approaches are rule-based
    - E.g., OWL2 RL or custom entailment
- **OWL2 RL**
  - Suitable W3C OWL2 profile
    - Allows scalable reasoning without sacrificing too much expressivity
  - Adjust reasoning complexity to suit scenario & resources
    - Choose rule subsets based on task & overhead
  - Enhance any rule-based task with semantic features
    - I.e., include OWL2 RL (subset) into ruleset
    - Such as **computerized, rule-based CPG** in CDSS

1) Optimizing the OWL2 RL ruleset

Multi-stage OWL2 RL ruleset selection

- Stable vs. volatile ontology
- Conformant

1) Equivalent OWL2 RL ruleset
   a) Removing logically equivalent rules
   b) Replace 2+ specific rules with more general rules & axioms
   c) Removing “stand-alone” schema inference rules
2) Purpose- and reference-based subsets
   a) *Purpose*: inferencing vs. validation
   b) *Reference*: instances vs. schema
3) Remove inefficient rules
   – Leave out rules with large performance impact
      • E.g., *#eq-ref* infers each resource is equivalent to itself
4) Domain-based ruleset selection
   – I.e., leave out rules not needed by ontology & dataset
   – Forward-chaining algorithm (Tai et al. [8])

Non-conformant

Stable

1) Optimizing the OWL2 RL ruleset: Evaluation

OWL2 RL*						
AndroJena	original	2819 (88 | 2731)				
		volatile ontology	stable ontology			
	conformant	*full*	*inf-schema*	*inf-inst*	*consist*	
		2639 (90 | 2549) + <u>entailed</u>	1001 (69 | 932)	1245 (187 | 1058) + <u>entailed</u>, <u>domain-based</u>	418 (195 | 223)	
	non-conformant	*full*	*inf-schema*	*inf-inst*		
		1547 (93 | 1455) + <u>entailed</u>, <u>ineff</u>	919 (65 | 854) <u>inst-ent</u>	272 (165 | 106) + <u>entailed</u>, <u>domain-based</u>, <u>ineff</u>, <u>inst-ent</u>		

* : [*total-time*] ([*load-time*] | [*reason-time*] ; applied selections are shown, if any.
**: *total-time*

OWL2 DL**	
Hermit	21111
Pellet	6978
JFact	7034

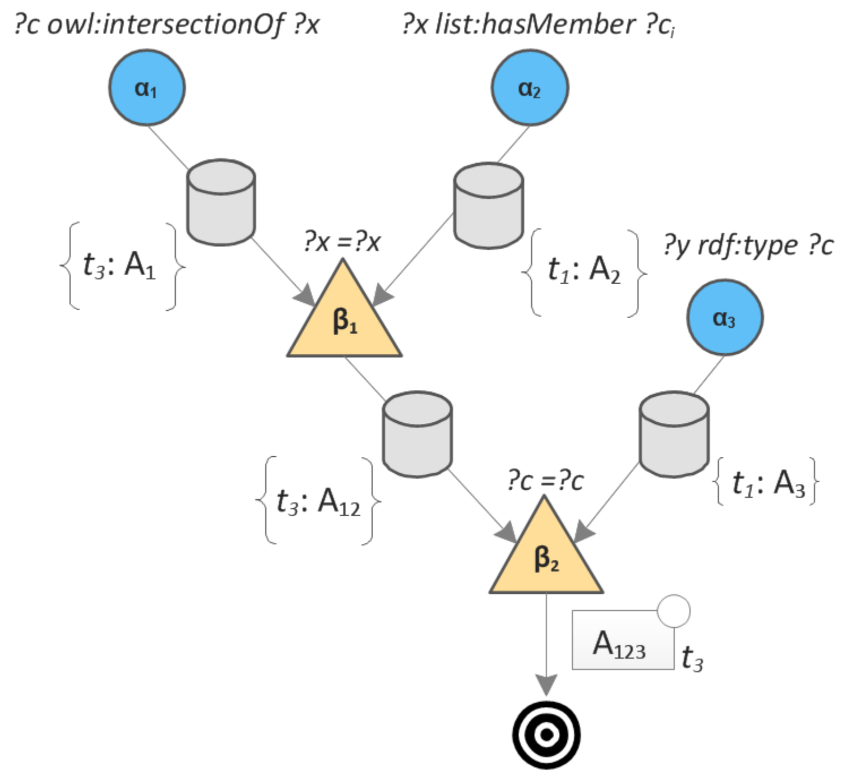
1) Optimizing the OWL2 RL ruleset: Future work

- Rule instantiation [7, 9, 10]
  1) Materialize schema inferences in ontology
  2) Instantiate each *instance rule* with schema terms
     - Increase rule selectivity
     - Reduce # of joins
  - Requires a “stable” ontology

- Domain-specific rulesets
  - Large impact on performance
  - Currently, does not support “volatile” ontologies
    - Ruleset needs to be re-calculated on ontology changes
    - Avg. ca. 291ms (PC), 4183ms (mobile)
  - Deploy on mobile device, integrate with reasoner?

2) RETE Strategies for Resource-Constrained Settings

- RETE Algorithm
  - Well-known solution to implement production rule systems
  - **Rule premise** = *alpha node*
    - *Alpha memory*: keeps matched facts
  - **Join** = *beta node*
    - *Beta memory*: keeps join results
  - Useful in dynamic environments, due to its incremental nature
- Known for trading *memory* for *performance*
  1) *Alpha memories* will overlap depending on premise selectivity
  2) Many SW applications already involve an RDF store for query access
     - Collection of alpha memories ***duplicate*** RDF store
- Many rules will not be needed for domain
  - But, still consume computing & memory resources in RETE
  - Tailor RETE networks during execution
    - In light of dynamic & incremental situations



2) RETE Strategies for Resource-Constrained Settings (2)

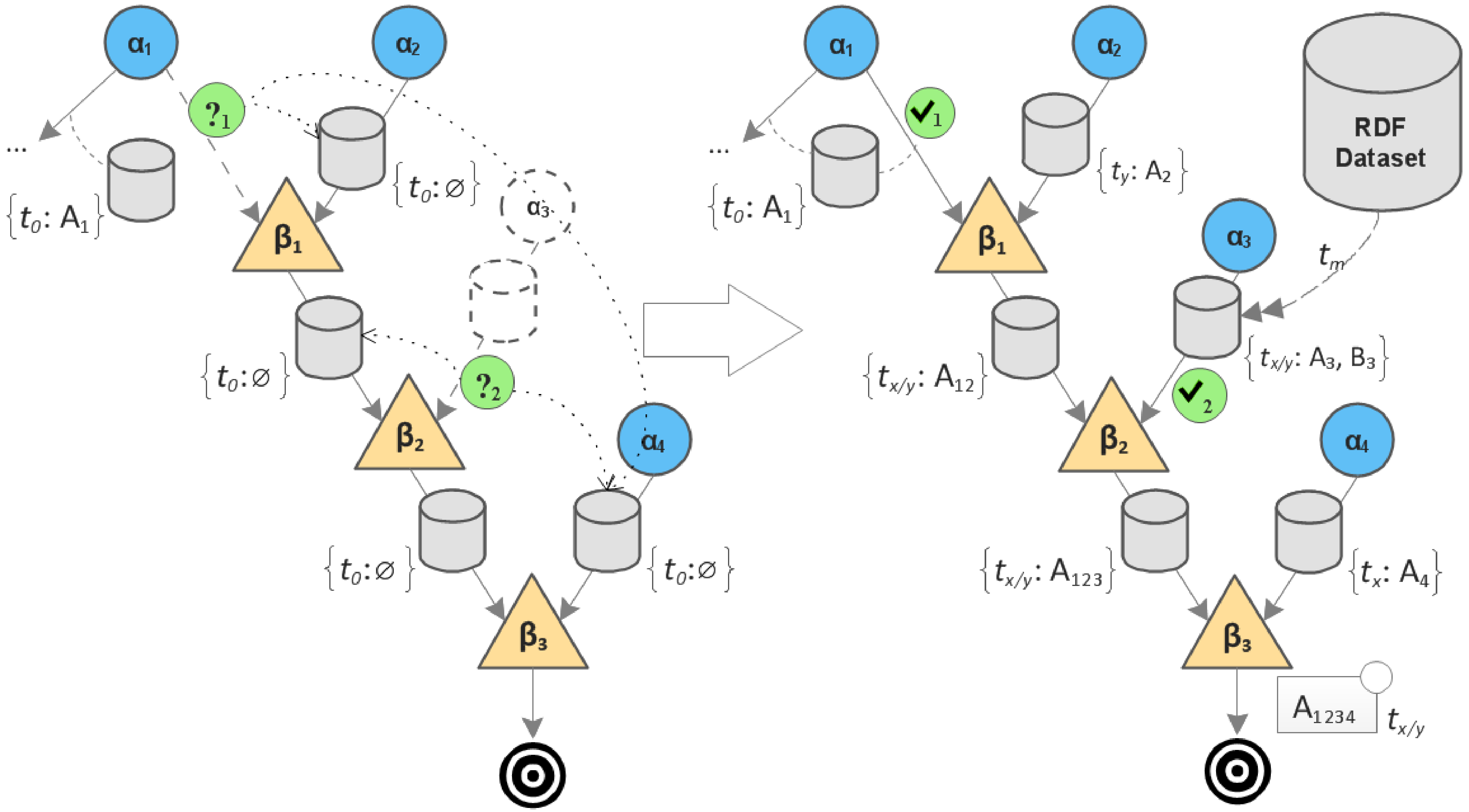
- *Dataset-mask* memory strategy
  - ➢ Keep alpha memories as *masks* on the RDF store
  - ➢ Query RDF store using *joining token* & *rule premise* as constraint
  - – **Hybrid version**: *dataset-mask* vs. regular memory, based on premise selectivity

- Dynamic tailoring of RETE networks
  1) *Avoid redundant join attempts* [11]
     - Unlink alpha memory from its beta node in case join attempts are useless
  2) *Avoid redundant token matches*
     - Pause alpha nodes in case they are unlinked from each rule
     - Requires separate RDF store for synchronizing alpha memory upon *resume*

  – **Join-utility heuristics**
     - Determine utility of join attempts
  1) *Empty sibling memory*
     - In case alpha (i <= 2) or beta (i > 2) memory is empty, no joins are possible [10]
  2) *Lower failed alpha nodes*
     - Pointless to attempt joins in case a failed alpha node occurs lower down

Dynamic tailoring of RETE networks

Responsiveness to Incremental Reasoning Scenarios



Dataset-mask: Evaluation (1)

strategy*	**memory usage**		**reasoning performance (ms)†** *(P.3)*	
	*# α memories*** (M.1)*	*α contents†* *(M.2)*	*PC*	*mobile*
regular-memory	**r**: 46, **d**: 0	95342 (2900 - 747114)	15705 (18 - 322352)	24968 (1051 - 199974)
dataset-mask	**r**: 0, **d**: 46	46 (46 – 46)	51905 (51 - 1187570)	69573 (2903-542670)
hybrid-0.1,0.25	**r**: 42, **d**: 4	6224 (279 - 50827)	16303 (27 - 340194)	29287 (1526 - 212573)
hybrid-0.5	**r**: 43, **d**: 3	18885 (1115 - 188153)	16475 (23 - 338109)	26444 (1145 - 202646)
hybrid-0.75,1	**r**: 44, **d**: 2	43090 (1248 - 335444)	17715 (25 - 365843)	25203 (1115 - 198404)

Table 1. Benchmark results for RETE alpha memory strategies

*: hybrid-[**x**]: **x** represents the utilized threshold (Section 4.2)

: **r = regular memories, **d** = dataset-masks

†: showing averages, with min-max in parenthesis

Dataset-mask: Evaluation (2)

What if SW scenario does not include an RDF store?

- Introduce RDF store as *shared alpha memory pool*

- Updated memory reductions:
  - *Dataset-mask*: avg. ca. -55%
  - *Hybrid-0.1,0.25*: avg. ca. -27%
  - *Hybrid-0.5:* avg. ca. -9%
  - *Hybrid-0.75,1*: avg. ca. +1%
- RDF store update operations:
  - *PC*: avg. ca. +0,67s
  - *Mobile*: avg. ca. +1s

Dynamic RETE tailoring: Evaluation

tailoring			reverting		
queue-unlink		*node-pause*	*queue-relink*		*node-resume*
#	*heuristic*		#	*heuristic*	
2625	(1)	44	14	(1)	2
24	(1), (2)				

Table 4. Dynamic tailoring statistics (T.2) (total number of tailoring operations) (PC)

config	RETE operation statistics *(T.1)*		reasoning times (ms)*			complete *(T.4)*
	# token matches	*# join attempts*	*preproc (T.3)*	*initial dataset (P.3)*	*incremental (P.4)*	
default	218038	2062420	n/a	**pc**: 825 **mo**: 2070	**pc**: 14065 **mo**: 23573	✓
dynamic tailoring	181687	657075			**pc**: 13989 **mo**: 20682	✓
a priori tailoring	132831	279997	**pc**: 291 **mo**: 4183	**pc**: 808 **mo**: 1529	**pc**: 13893 **mo**: 19356	X (- 11448)

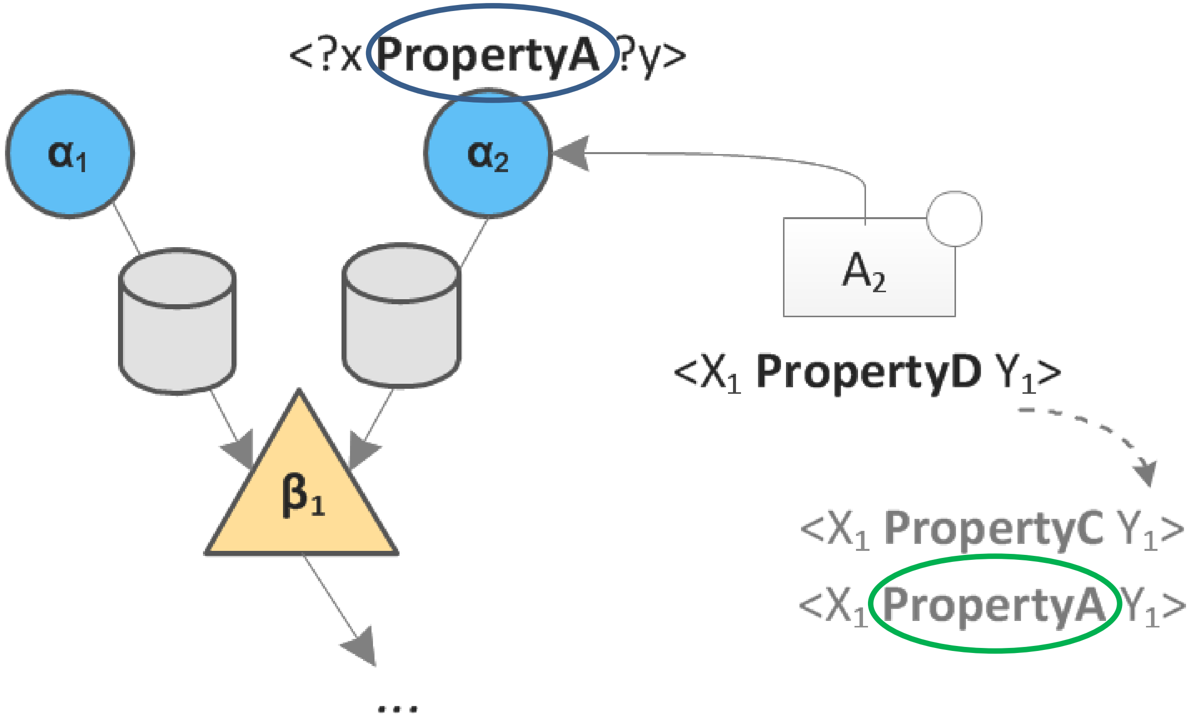
Table 5. Comparison of three configurations for incremental reasoning.
*: **pc** = PC performance, **mo** = mobile performance

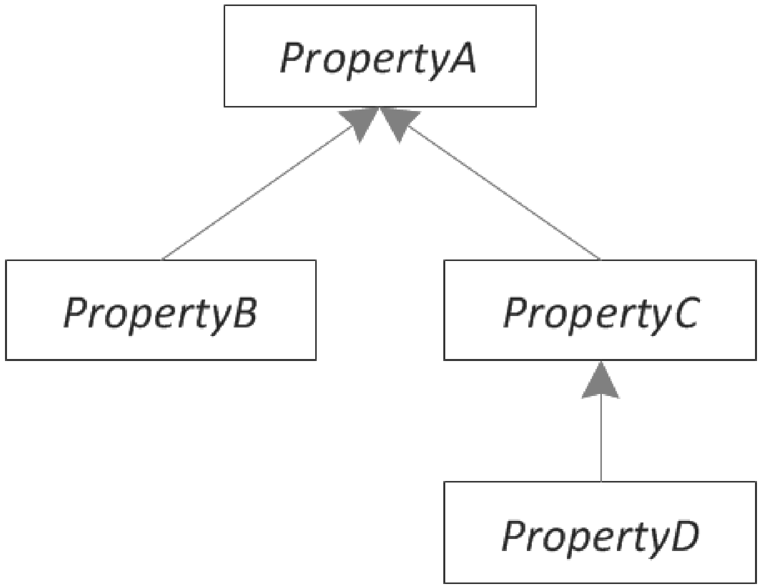
2) RETE Strategies for Resource-Constrained Settings: Future work (in progress)

- Currently: mostly based on OWL2 RL ruleset in clinical decision support
  - Also, benchmarks done using OWL2 RL ruleset
  - Additional benchmarks needed for other rulesets
- More advanced heuristics to determine join utility
  - Eager vs. lazy algorithm
- More fine-grained memory strategy
  - Alpha memories will often subsume other memories
  - E.g., subsumed (virtual) alpha memories access their subsuming, concrete alpha memory behind-the-scenes (comparable to *dataset-mask* but with a smaller query access overhead)
- Dynamic *hybrid* memory strategies
  - Switch between regular and *dataset-mask* memories based on evolving selectivities

2) RETE Strategies for Resource-Constrained Settings:
Future work (in progress) (2)

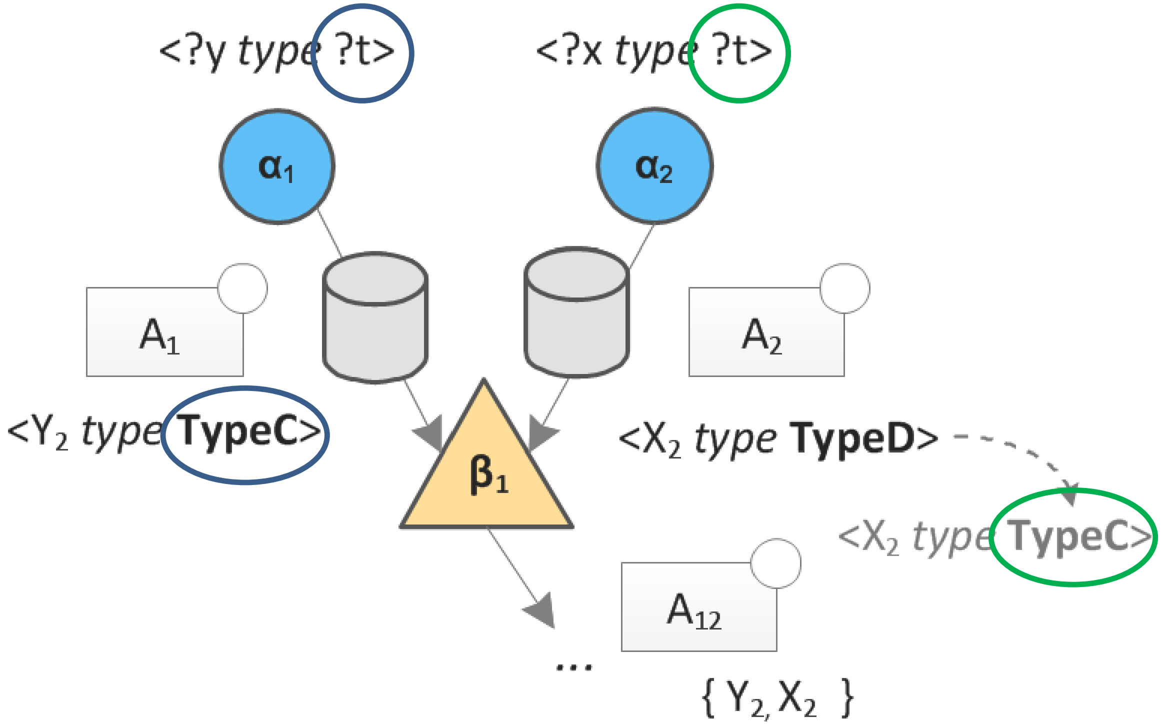
- Virtual materialization of OWL2 semantics in match & join operations
  - Consider OWL2 semantics when matching & joining tokens
  - Avoid explicit materialization, which takes up memory

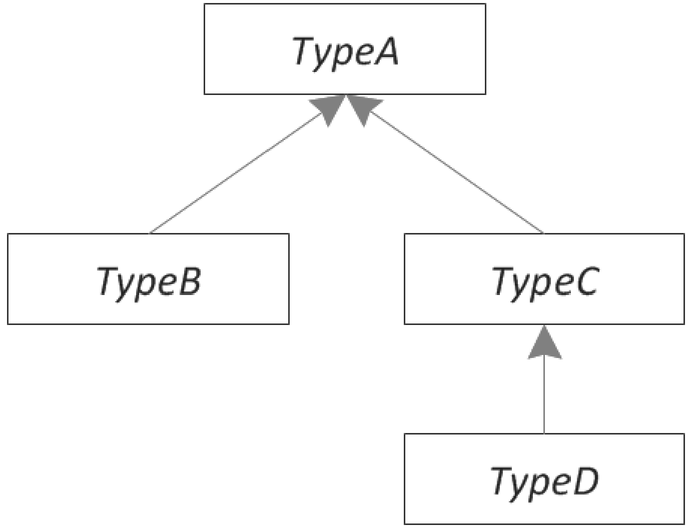




2) RETE Strategies for Resource-Constrained Settings:
Future work (in progress) (2)

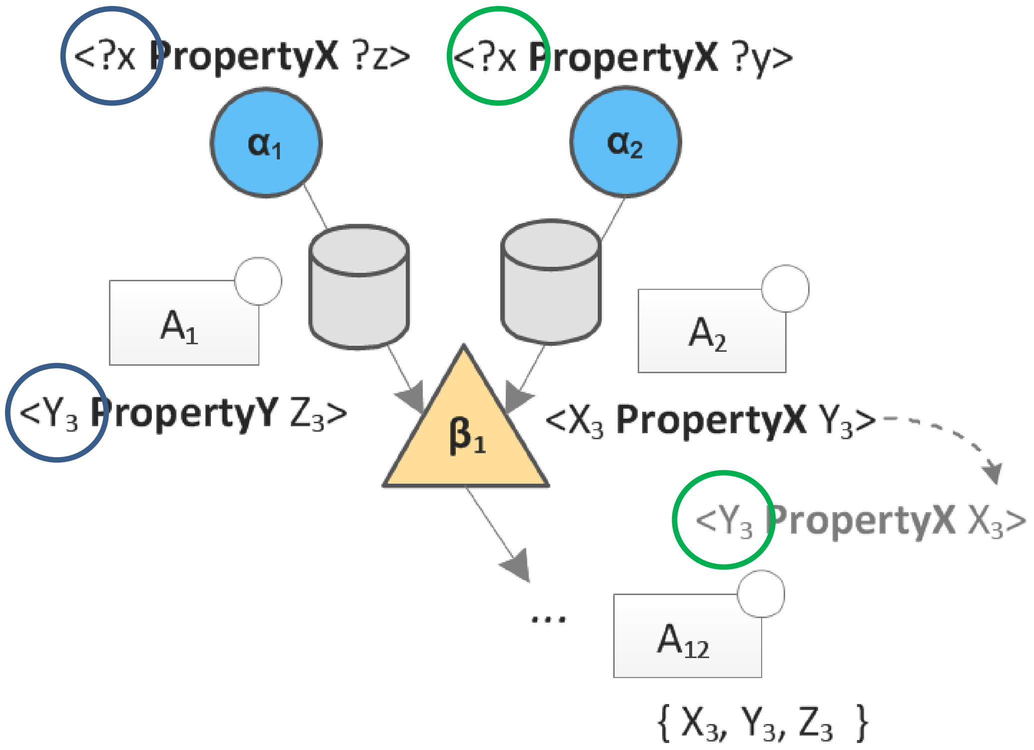
- Virtual materialization of OWL2 semantics in match & join operations
  - Consider OWL2 semantics when matching & joining tokens
  - Avoid explicit materialization, which takes up memory





2) RETE Strategies for Resource-Constrained Settings:
Future work (in progress) (2)

- Virtual materialization of OWL2 semantics in match & join operations
  - Consider OWL2 semantics when matching & joining tokens
  - Avoid explicit materialization, which takes up memory



PropertyX *type* SymmetricProperty

Questions?

- **E-mail**: william.van.woensel@gmail.com

References

[1] *S. Abidi*. PhD Thesis, 2010.

[2] *B. Jafarpour*. PhD Thesis, 2010.

[3] *B. Jafarpour, S. S. R. Abidi, S. R. Abidi*. **Exploiting Semantic Web Technologies to Develop OWL-Based Clinical Practice Guideline Execution Engines**. *IEEE J. Biomed. Heal. Informatics*, 2014.

[4] *C. Bobed, R. Yus, F. Bobillo, E. Mena*. **Semantic reasoning on mobile devices: Do androids dream of efficient reasoners?** Web Semantics: Science, Services and Agents on the World Wide Web, 35:167–183, December 2015.

[5] *S. Ali and S. Kiefer*. **“microOR --- A Micro OWL DL Reasoner for Ambient Intelligent Devices.** In *Proceedings of the 4th International Conference on Advances in Grid and Pervasive Computing*, 2009, pp. 305–316.

[6] *T. Kim, I. Park, S. J. Hyun, D. Lee.* **MiRE4OWL: Mobile Rule Engine for OWL.** In *Proceedings of the 2010 IEEE 34th Annual Computer Software and Applications Conference Workshops*, 2010, pp. 317–322.

[7] *B. Motik, I. Horrocks, S. M. Kim.* **Delta-reasoner: A Semantic Web Reasoner for an Intelligent Mobile Platform.** In *Proceedings of the 21st International Conference Companion on World Wide Web*, 2012, pp. 63–72.

[8] *W. Tai, J. Keeney, D. O’Sullivan*. **Resource-constrained reasoning using a reasoner composition approach.** *Semant. Web*, vol. 6 (1), pp. 35–59, 2015

[9] *J. Bak, M. Nowak, C. Jedrzejek*. **RuQAR: Reasoning Framework for OWL 2 RL Ontologies**. In *The Semantic Web: ESWC 2014 Satellite Events, Anissaras, Crete, Greece, May 25-29, 2014,* 2014, vol. 8798, pp. 195–198.

[10] *G. Meditskos N. Bassiliades*. **DLEJena: A Practical Forward-chaining OWL 2 RL Reasoner Combining Jena and Pellet**. *Web Semant.*, vol. 8, no. 1, pp. 89–94, Mar. 2010.

[11] *R.B. Doorenbos.* **Combining Left and Right Unlinking for Matching a Large Number of Learned Rules**. In Hayes-Roth, B. and Korf, R.E. (eds.) Proceedings of the 12th National Conference on Artificial Intelligence, Seattle, WA, USA, July 31 - August 4, 1994, Volume 1. pp. 451–458.