Session Notes: Topics for Exploration
KDT MindMeld, University of California, Santa Barbara

March 7, 2012

**Betweenness Centrality.**
How can we implement B.C. (and other global structure algorithms) in SPARQL/uRiKA?

**Multithreaded SPGEMM \(\approx\) SPARQL Join?**
Join optimization \(\approx\) heuristics for graph subisomorphism?

**Integrating SPARQL and CombBLAS/KDT**
Is there a way to map SPARQL onto CombBLAS? Can we avoid having to implement B.C./Clustering/etc... in RDF/SPARQL and just use what we already implemented in KDT? Converting between different native formats shouldn’t be too hard. However, if CombBLAS were to “sit under” SPARQL, the infrastructure might become a problem – different edge types, etc...

**Subgraph Isomorphism in KDT.**
Do we need more primitives in CombBLAS to implement subgraph isomorphism? Is it worthwhile to have KDT do the common routines that are already implemented in RDF?

**Subgraphs/Patterns in RDF/SPARQL.**
Instead of generating three different lists, then doing a join, could we do a breadth-first search from different starting vertices to find the pattern/subgraph that we are looking for?

**Streaming data and updates.**
How much work do we do in real time? Do we need constant updates? Can we have a “knowledge discovery” period and an “update” period? What granularity is important?
Figure 1: SPARQL is a graph generator – can we feed its output graphs into KDT?

Figure 2: Could combBLAS sit under SPARQL to implement global structure algorithms?