CSE 414: Section 4
Datalog

July 12th, 2018
Datalog Terminology

Head - Body - Atom/Subgoal/Relational predicate
Base Relations (EDB) vs Derived Relations (IDB)
Wildcard -> ignore

Helper(a,b):-Base1(a,b,_) 
NonAns(j):-Base2(j,k),!Base3(k) 
Ans(x):-Helper(x,y),!NonAns(y)
Query Safety

Need a positive relational atom of every variable

What’s wrong with this query?

Find all of Alice’s children without children:
\[ U(x) :\text{ ParentChild(“Alice”,x), !ParentChild(x,y)} \]

A datalog rule is **safe** if every variable appears in some positive relational atom.
Query Safety

U(x) :- ParentChild("Alice",x), !ParentChild(x,y)
It is domain dependent! Unsafe!

Double negation to the rescue. Why does this work?
NonAns(x) :- ParentChild("Alice",x), ParentChild(x,y)
# All of Alice’s children with children
U(x) :- ParentChild("Alice",x), !NonAns(x)
# All of Alice’s children without children (safe!)
Query Safety

But we can do better...

`hasChild(x) :- ParentChild(x,_)`

# People with children

`U(x) :- ParentChild("Alice",x), !hasChild(x)`

# All of Alice’s children without children (safe!)
Datalog with Recursion

Able to write complicated queries in a few lines

Graph analysis

Done with query once output does not change.
Recursion might not work well with negation

E.g.
\[ A(x) :\neg \text{Table}(x), \neg B(x) \]
\[ B(x) :\neg \text{Table}(x), \neg A(x) \]

Solution: Don’t negate or aggregate on an IDB predicate until it is defined

Stratified Datalog Query
Only IDB predicates defined in strata 1, 2, ..., n may appear under ! or agg in stratum n+1
Souffle (HW4)

Install from source:

https://github.com/souffle-lang/souffle/wiki/build