

Algebra	Datalog	SQL
Conjunctions	rule bodies	simple SQL + nested positive
SPJR union	positive nonrecursive Datalog	union
+ Difference	negative in the bodies + stratification, still nonrecursive	not negative subqueries MINUS, INTERSECT
+ TC (transitive closure)	recursive Datalog Stratified Negation	CONNECT BY
possible to add [group by, agg exprs] (R)	group-by in Datalog / Prolog	GROUP BY (HAVING)

X recursive Datalog  
 X stratified negation  
 X group-by in Datalog / Prolog  
 X nested subqueries

what is there additionally (classical) queries ✓  
 would it will not be our central example in the rest of the lecture

Jun 17-10:07

What is there ~~below~~ <sup>more than</sup> beyond queries (in IT)

- Algorithms
  - procedural programming (js, Java, Pascal, C, ... C++, ... PL/SQL)
  - Turing-complete, solve decidable problems
- Full Prolog; Prolog Lists → trees "Logic Programming"
- Functional Programming (LISP, Haskell, XQuery)

Here: extend Datalog, stay on the declarative side → restricted expressiveness

What do we have?

- universal Model Semantics
- $T_P$  operator
- stratification:  $T_P$  + strategy
- more involved strategies

Jun 17-10:53

What do we have

$T_P, T_P^w$ , minimal model

→ based on sets of ground facts, <sup>Herbrand interpretation</sup>  
ground rule applications

→ only positive facts

disadvantage: how to express negative knowledge

→ we cannot express it, we have only Negation as Failure

advantage: closed-world-reasoning

→ adding any fact leads again to a new stratified model (nonmonotonicity)

↳ leads to WFS a little bit later

Jun 17-11:02

Rules

$h \leftarrow \text{body}$  (up to now stratified)  
 conjunctive,  $\exists$ , negation

↑ possible atoms (incl. variables)  
 but in  $T_P$  ground-instantiated  
 + efficiency  
 - restrictive

not possible

Ex: airport :- country(N,C,...)  
 this country → goto OWL/semWeb

white(x) v black(x) :- position(x). (goals)

(possibilities) :- (procedures) (planning problems)

→ conclusion/head would be a formula  
 $a \vee b \rightarrow$  not Herbrand  
 insolve

→ consider different Herbrand models  
 advantage: problems with multiple solutions!  
 ↳ stable models

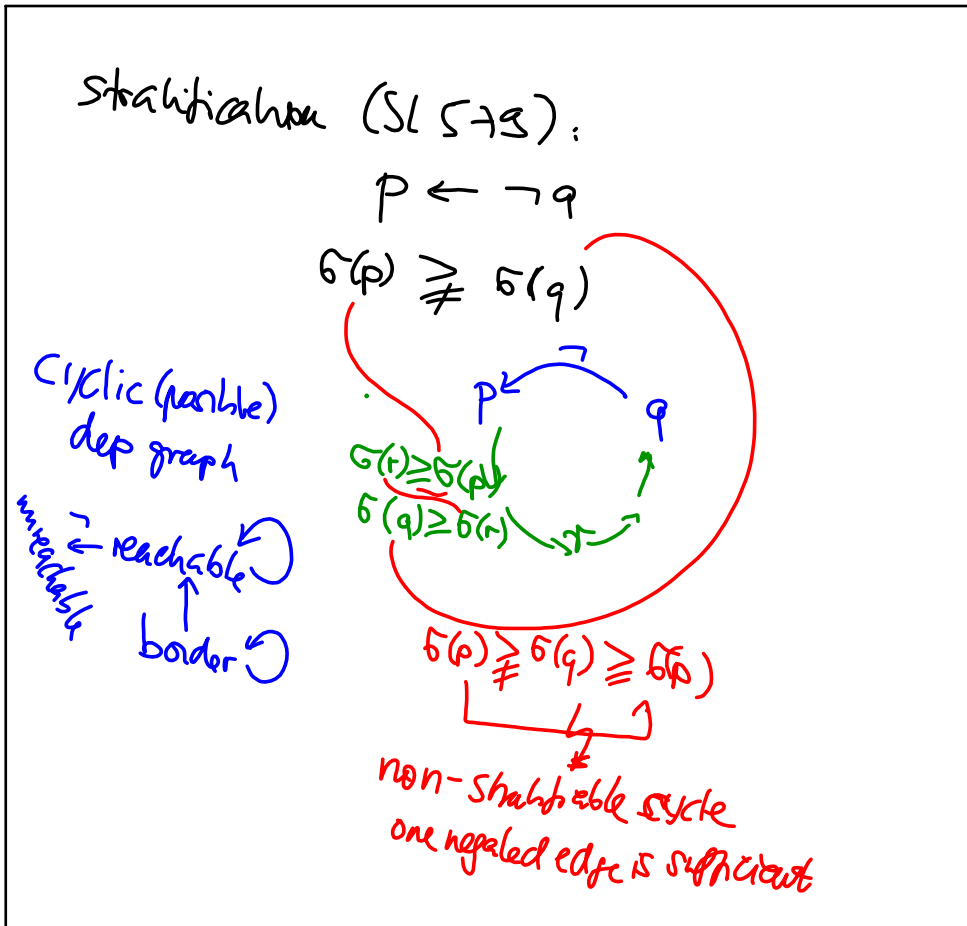
Jun 17-11:14

Is stratification a <sup>serious</sup> restriction?

- do we need anything non-stratified?
- Queries are always stratified.

What {problems, programs} are not stratified?

Jun 17-11:20



Jun 17-11:26

Example

$$p(b) \leftarrow \neg p(a).$$

$$p(c) \leftarrow \neg p(b).$$

$$p(d) \leftarrow \neg p(c).$$

equiv. boolean / "founded" prog

$$p_b \leftarrow \neg p_a$$

$$p_c \leftarrow \neg p_b$$

$$p_d \leftarrow \neg p_c$$

Models: study with  $\phi$

$\top p \rightsquigarrow$	$\{p(b)\}$	two-valued knowledge (and not $p(c)$ )
$\top p \rightsquigarrow$	$\{p(b)\}$	
$\top p \rightsquigarrow$	$\{p(b), p(d)\}$	

"well-founded argumentation"

kind of stratification on-demand

"local stratification"

based on instances

not on the symbols

$\sim$  large space

Jun 17-11:32

Other case:

$$p(a) :- \neg p(b).$$

$$p(b) :- \neg p(a).$$

Locally equiv.

$$p(a) \vee p(b)$$

$$p(b) \vee p(a)$$

well-founded argumentation: no conclusion

plaus problem: both  $\{p(a)\}$  and  $\{p(b)\}$  are good solutions

Stable

So far the idea ....

$\searrow$  Friday continue with slide 595.

Jun 17-11:41



Jun 17-11:48