Validation of datalog reasoning results

Johannes Tantow

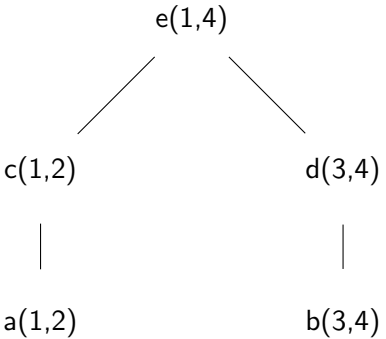
February 5, 2024

Datalog


```
a(1,2).
b(3,4).

c(?x,?y) :- a(?x,?y).

d(?x,?y) :- b(?x,?y).

e(?x,?y) :- c(?x,?z), d(?a, ?y).
```

```
a(1,2).
b(3,4).

c(1,2) :- a(1,2).

d(3,4) :- b(3,4).

e(1,4) :- c(1,2), d(3,
      4).

----
c(1,3) :- a(1,3).
```

Proof trees




```
def isValid(P: program τ) (d:
    database τ) (t: proofTree τ):
    Prop :=
  match t with
  | proofTree.node a l =>
  ( ∃(r: rule τ) (g:grounding τ),
    r ∈ P ∧ ruleGrounding r g =
    groundRuleFromAtoms a (List.map
     root l)
    ∧ l.attach.All₂ (fun ⟨x, _h⟩ =>
     isValid P d x))
  ∨ (l = [] ∧ d.contains a)
```

Datalog modelling

Syntax:

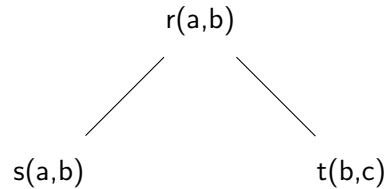
```
structure rule where
(head: atom τ)
(body: List (atom τ))

def grounding:=
    τ.vars → τ.constants
```

Semantics:

```
def proofTheoreticSemantics (P:
    program τ) (d: db):=
{a:| ∃ (t: proofTree τ), root t = a ∧
     isValid P d t}

theorem SemanticsEquivalence
 (P: program τ) (d: db τ):
    proofTheoreticSemantics P d =
    modelTheoreticSemantics P d :=
```

Tree validation

```
P:

r(x,y) :- s(y,x).

r(x,y) :- s(x,z),t(z,y).
```



Formally implemented and proved unification:
Try:
r(x,y) :- s(x,z),t(z,y).
with
r(a,b) :- s(a,b), t(b,c)

Completeness

Is our solution all we can derive ?

i elements in valid trees

$i \subseteq \texttt{proofTheoreticSemantics P d} = \texttt{modelTheoreticSemantics P d} \subseteq i$