Relational Clausal Logic - *Syntax*: clauses

statements concern relations among objects from a universe of discourse

add constants, variables and predicates to propositional logic

```
 constant : single word starting with lower case
 variable : single word starting with upper case
     term : constant | variable
predicate : single word starting with lower case
     atom : predicate[(term[,term]*])]
   clause : head [:- body]
     head : [atom[;atom]*]
     body : atom[,atom]*
```

"peter likes anybody who is his student. maria is a student of peter"

```
likes(peter,S) :- student_of(S,peter).
student_of(maria,peter).
```

Relational Clausal Logic - *Semantics*: Herbrand universe, base, interpretation

Herbrand universe of a program P

```
{ peter, maria }
```

set of all terms that are ground in P

term without variables

Herbrand base B_P of a program P

```
{ likes(peter,peter),likes(peter,maria),
  likes(maria,peter),likes(maria,maria),
  student_of(peter,peter), student_of(peter,maria),
  student_of(maria,peter), student_of(maria,maria) }
```

set of all ground atoms that can be constructed using predicates in P and arguments in the Herbrand universe of P

Herbrand interpretation I of P

```
{ likes(peter,maria), student_of(maria,peter) }
```

subset of B_P consisting of ground atoms that are true

is this a model?
need to consider
variable substitutions

Relational Clausal Logic - *Semantics*: substitutions and ground clause instances

A substitution is a mapping $\sigma : \mathrm{Var} \rightarrow \mathrm{Trm}$.
For a clause C, the result of σ on C, denoted $C\sigma$
is obtained by replacing all occurrences of $X \in \mathrm{Var}$ in C by $\sigma(X)$.
$C\sigma$ is an instance of C.

```
if σ={S/maria} then
(likes(peter,S):-student_of(S,peter))σ
=likes(peter,maria):-student_of(maria,peter)
```

Relational Clausal Logic - *Semantics*: models

ground instances of relational clauses are like propositional clauses

interpretation I is a model of a clause C
$\iff$ I is a model of every ground instance of C.

interpretation I is a model of a program P
$\iff$ I is a model of each clause C $\in$ P.

P
```
likes(peter,S) :- student_of(S,peter).
student_of(maria,peter).
```

I
```
{ likes(peter,maria), student_of(maria,peter) }
```

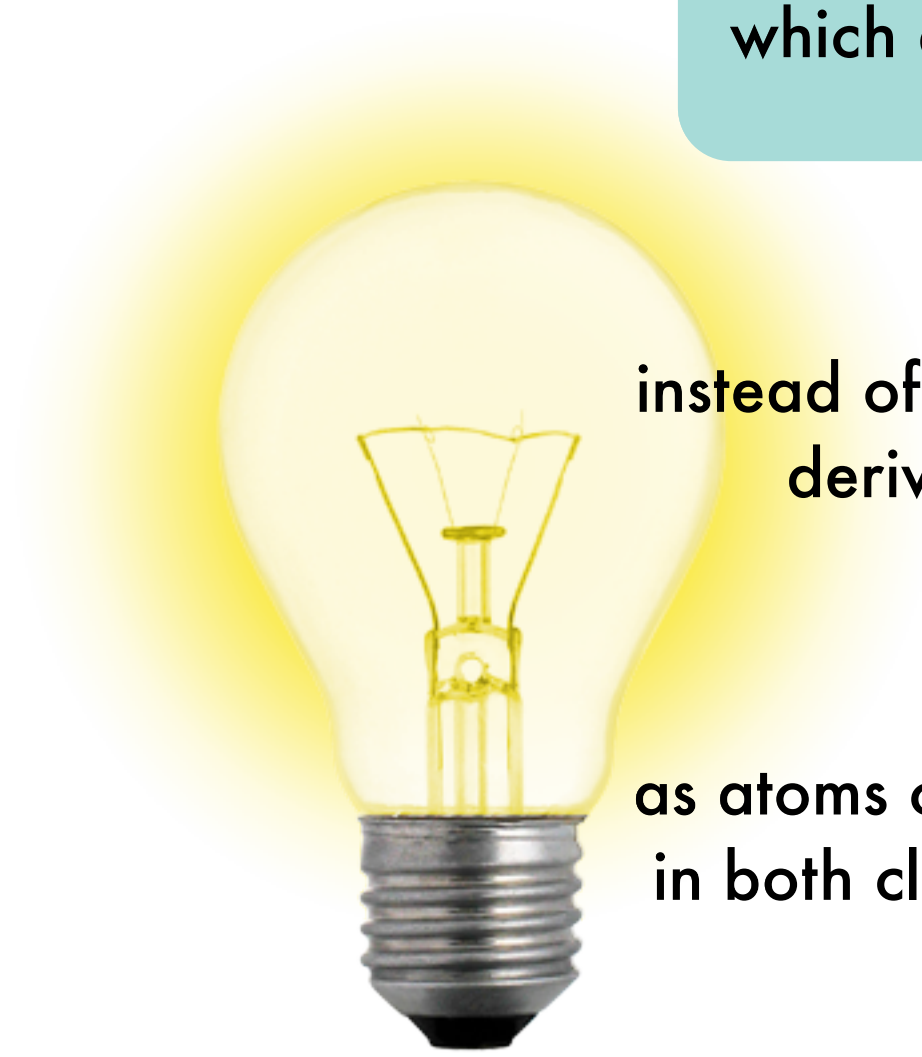
I is a model for P
because it is a model of all ground instances of clauses in P:

```
likes(peter,peter) :- student_of(peter,peter).
likes(peter,maria) :- student_of(maria,peter).
student_of(maria,peter).
```

Relational Clausal Logic - *Proof Theory*: naive version

naive because there are many grounding substitutions, most of which do not lead to a proof

derive the empty clause through propositional resolution from all ground instances of all clauses in P



instead of trying arbitrary substitutions before trying to apply resolution, derive the required substitutions from the literal resolved upon (positive in one clause and negative in the other)

as atoms can contain variables, do not require exactly the same atom in both clauses ... rather a complementary pair of atoms that can be made equal by substituting terms for variables

Relational Clausal Logic - *Proof Theory*: unifier

A substitution σ is a **unifier** of two atoms a_1 and a_2
$\iff a_1\sigma = a_2\sigma$. If such a σ exists, a_1 and a_2 are called unifiable.

A substitution σ_1 is **more general** than σ_2 if $\sigma_2 = \sigma_1\theta$ for some substitution θ.

A unifier θ of a_1 and a_2 is a **most general unifier** of a_1 and a_2
$\iff$ it is more general than any other unifier of a_1 and a_2.

If two atoms are unifiable then they their mgu is **unique** up to renaming.

Relational Clausal Logic - *Proof Theory*: unifier examples

```
p(X, b) and p(a, Y) are unifiable
with most general unifier {X/a,Y/b}
```

```
q(a) and q(b) are not unifiable
```

```
q(X) and q(Y) are unifiable:
```

```
{X/Y} (or{Y/X}) is the most general unifier
```

```
{X/a, Y/a} is a less general unifier
```

[http://users.informatik.uni-halle.de/~brass/lp03/c3_purep.pdf]

Relational Clausal Logic - *Proof Theory*: resolution using most general unifier

apply resolution on many clause-instances at once

$$\begin{aligned}
\textbf{if} \quad C_1 &= L_1^1 \vee \dots L_{n_1}^1 \\
C_2 &= L_1^2 \vee \dots L_{n_2}^2 \\
L_i^1\theta &= \neg L_j^2\theta \quad \text{for some } 1 \leq i \leq n_1,\ 1 \leq j \leq n_2
\end{aligned}$$

$$\text{where } \theta = \mathbf{mgu}(L_i^1, L_j^2)$$

$$\textbf{then} \quad L_1^1\theta \vee \dots \vee L_{i-1}^1\theta \vee L_{i+1}^1\theta \vee \dots \vee L_{n_1}^1\theta$$

$$\vee L_1^2\theta \vee \dots \vee L_{j-1}^2\theta \vee L_{j+1}^2\theta \vee \dots \vee L_{n_2}^2\theta$$

Relational Clausal Logic - *Proof Theory*:

example of proof by refutation using resolution with mgu

P
```
likes(peter,S) :- student_of(S,peter).
student_of(S,T) :- follows(S,C), teaches(T,C).
teaches(peter,decprog).
follows(maria,decprog).
```

"is there anyone whom peter likes"? ➠ add "peter likes nobody" to P

```
:-likes(peter,N)                          likes(peter,S):-student_of(S,peter).
                                          {S/N}
:-student_of(N,peter)
                                          student_of(S,T) :- follows(S,C), teaches(T,C).
                                          {S/N, T/peter}
:-follows(N,C),teaches(peter,C)           follows(maria,decprog).
                                          {N/maria, C/decprog}
:-teaches(peter,decprog)
                                          teaches(peter,decprog).
□
```

`:- likes(peter,N)){N/maria}` ∪ P ⊢ □ hence P ⊨ `likes(peter,maria)`

Relational Clausal Logic - *Meta-theory*: soundness and completeness

sound

relational clausal logic is sound

$P \vdash C \Rightarrow P \models C$

complete

relational clausal logic is refutation-complete

$P \cup \{C\}$ inconsistent $\Rightarrow P \cup \{C\} \vdash \square$

new formulation because

:- p(X). $\equiv \forall X \cdot \neg p(X)$

while $\neg$(p(X).) $\equiv \neg(\forall X \cdot p(X)) \equiv \exists X \cdot \neg p(X)$

Relational Clausal Logic - *Meta-theory*: decidability

The question "P⊨C?" is decidable for relational clausal logic.

also for propositional clausal logic

Herbrand universe and base are finite

therefore also interpretations and models

could in principle enumerate all models of P and check whether they are also a model of C