Relational Logic

Syntax

- **Vocabulary**: a vocabulary is defined by a set of *object constants*, a set of *relation constants* with the corresponding *arity mapping*.
- **Term**: a term is a *variable* or an *object constant*.
- **Sentence**: a sentence can be of three forms:
  - **Relational sentence**: sentence formed of a *n-ary* relation constant and n terms.
  - **Logical sentence**: negation, conjunction, disjunction, implication, and equivalence of *relational sentences*.
  - **Quantified sentences**: sentence formed of *quantifiers*.
- **Ground**: an expression is ground if and only if it does not contain variables.
- **Free**: a variable is free relatively to a quantifier if it is not in the scope of the quantifier.
- **Bound**: a variable is bounded relatively to quantifier if it is in the scope of the quantifier.
- **Open sentence**: a sentence is open if it has free variables.
- **Closed sentence**: A sentence is closed it doesn't contain free variables.

Semantic

- **Herbrand base**: set of all ground sentences that can be formed from the constants of the language, i.e all sentences of the form $r(t_1, \ldots, t_n)$ where r is a $n-ary$ relation constant and $t_1, \ldots, t_n$ are object constants.
- **Truth assignement**: mapping of each ground relational sentence in the Herbrand base to a truth value.
- **Instance**: an instance of an expression is an expression where all free variables have been consistently replaced by ground terms.

Satisfaction

- A $\forall$ quantified sentence is true if and only if every instances of the scope of the quantified is true for that assignment.
- A $\exists$ quantified sentence is true if and only if at least one instance of the scope of the quantified is true for that assignment.
- A truth assignment satisfies a sentence with free variables if and only if it satisfies every instance of that sentence.
- A truth assignment satisfies a set of sentences if and only if it satisfies every sentences.

Evaluation

- Treat $\forall$ as conjunction of all instances
- Treat $\exists$ as a disjunction of all instances

Logical properties

A sentence can be:

- **valid**: if and only if it is satisfied by all truth assignment
- **unsatisfiable**: if and only if it isn't satisfied by any truth assignment
- **contingent**: if and only if there are truth assignment that satisfied it and other that falsifies it

A *ground sentence* in relational logic is valid/unsatisfiable/contingent if and only if the corresponding proposition in propositional logic is valid/unsatisfiable/contingent.

Relational Logic and Propositional Logic

Relational Logic (RL) is *expressively equivalent* to Propositional Logic (PL). Therefore for any set of arbitrary sentences in our RL language, there is a corresponding set of sentences in the language of PL such that any RL truth assignment that satisfies our RL sentences agrees with the corresponding PL truth assignment applied to the PL sentences.

To convert a RL sentence to a PL proposition we first convert our sentence to ***prenex form***, then we ground the results and rewrite it in PL.

Prenex Form

To convert a RL sentence to prenex we do as follow:

1. Rename variables in different quantified sentences to eliminate any duplicates
2. Apply quantifier distribution rules in reverse to move quantifiers outside of logical operators
3. Universally quantify any free variables in our sentences

Grounding

To ground a set of sentences Δ in prenex form into the set Γ of propositions we process each sentence thanks to the following rules:

- If the processed sentence φ is ground, we remove φ from Δ and add it to Γ

$$\begin{aligned} \Delta_{i+1} &= \Delta_i - \{\varphi\} \\ \Gamma_{i+1} &= \Gamma_i \cup \{\varphi\} \end{aligned}$$

- If the processed sentence is of the form $\forall\nu.\varphi[\nu]$, we remove it from Δ and replace it with copies of the scope, one copy for each object constant τ in our language

$$\begin{aligned} \Delta_{i+1} &= \Delta_i - \{\forall\nu.\varphi[\nu]\} \cup \{\varphi[\tau] |\ \tau \textit{ an object constant}\} \\ \Gamma_{i+1} &= \Gamma_i \end{aligned}$$

- If the processed sentence is of the form $\exists\nu.\varphi[\nu]$, we remove it from Δ and replace it with a disjunction, where each disjunct is a copy of the scope in which the quantified variable is replaced by an object constant in our language

$$\begin{aligned} \Delta_{i+1} &= \Delta_i - \{\forall\nu.\varphi[\nu]\} \cup \{\varphi[\tau_1] \vee \ldots \vee \varphi[\tau_n]\} \\ \Gamma_{i+1} &= \Gamma_i \end{aligned}$$