Description Logic as a fragment of first-order logic

- **concept** $C$ as a first-order formula $\tau_x(C)$ with free variable $x$
  \[
  \tau_x(\text{ant} \sqcap \text{big}) = \text{ant}(x) \land \text{big}(x)
  \]

- A **is-a** B as
  
  instance-of $B(A)$ (ABox)

  or

  inclusion $A \subseteq B$ (TBox)

- other binary relations as **roles** that restrict/guard $\exists, \forall$
  
  \[
  \tau_x((\exists \text{ daughter}) \text{ student}) = (\exists y)(\text{daughter}(x, y) \land \text{student}(y))
  \]

  \[
  \tau_x((\forall \text{ friend}) \text{ kind}) = (\forall y)(\text{friend}(x, y) \rightarrow \text{kind}(y))
  \]

- roles can be complex
  
  \[
  \tau_{x,y}(\text{daughter}; \text{friend}) = (\exists z)(\text{daughter}(x, z) \land \text{friend}(z, y))
  \]

A DL with structural subsumption

*Brachman and Levesque*

\[
\langle \text{concept} \rangle ::= \langle \text{atomic-concept} \rangle | [\text{ALL} \langle \text{role} \rangle \langle \text{concept} \rangle] | [\text{EXISTS} \ n \langle \text{role} \rangle] | [\text{FILLS} \langle \text{role} \rangle \langle \text{constant} \rangle] | [\text{AND} \langle \text{concept} \rangle \cdots \langle \text{concept} \rangle]
\]

Subsumption, $D \subseteq E$, can be computed **structurally**

1. **normalize** to a conjunction
   
   \[
   D \leadsto [\text{AND } d_1 \cdots d_n]
   \]

2. **match** every $E$-requirement by a $D$-requirement
   
   \[
   [\text{AND } D] \subseteq [\text{AND } E] \iff (\forall e \text{ in } E)(\exists d \text{ in } D) d \subseteq e
   \]

No Unique Names Assumption: need $c_1 \neq c_2$ for

\[
[\text{FILLS } r \ c_1], [\text{FILLS } r \ c_2] \subseteq [\text{EXISTS } 2 \ r]
\]

No disjunction or negation:

\[
e \sqcup d, \neg d \subseteq e
\]
**ALC (attributive language with complement)**

Boolean operations reduce subsumption to satisfiability

\[ D \sqsubseteq E \iff \neg D \sqcup E = \top \]
\[ \iff D \sqcap \neg E = \bot \]
\[ \text{i.e. } D \sqcap \neg E \text{ is unsatisfiable} \]

**ALC**: Boolean operations \((-, \sqcap, \sqcup)\) + roles via \(\forall\) and \(\exists\)

\[
\langle \text{concept} \rangle ::= \langle \text{atomic-concept} \rangle | \neg \langle \text{concept} \rangle |
\langle \text{concept} \rangle \sqcap \langle \text{concept} \rangle | \langle \text{concept} \rangle \sqcup \langle \text{concept} \rangle |
(\forall \langle \text{role} \rangle) \langle \text{concept} \rangle | (\exists \langle \text{role} \rangle) \langle \text{concept} \rangle
\]

---

**Tableaux and disjunctive normal form**

Tableau rules expand ABox \(A\) with its subformula consequences

\(\text{e.g. } A := A \cup \{C(t), C'(t)\} \text{ if } (C \sqcap C')(t) \in A\)

branching in case \((C \sqcup C')(t) \in A \wedge C(t), C'(t)\)

\[
A_1 := A \cup \{C(t)\}
\]
\[
A_2 := A \cup \{C'(t)\}
\]

so that

\[
\{A_1, \ldots, A_n\} \approx \bigvee_{i=1}^{n} \bigwedge_{\varphi \in A_i} \varphi
\]

i.e., a Disjunctive Normal Form — beyond \([\text{AND } d_1 \cdots d_n]\) in structural subsumption.

A fully expanded \(A\) with no contradictory pairs specifies a model.
### Tableau rules for $\mathcal{ALC}$

**(∩-rule)** If $\mathcal{A}$ contains $(C \cap C')(x)$ but not both $C(x)$ and $C'(x)$,

\[
\mathcal{A} := \mathcal{A} \cup \{C(x), C'(x)\}.
\]

**(⊔-rule)** If $\mathcal{A}$ contains $(C \sqcup C')(x)$ but neither $C(x)$ nor $C'(x)$,

\[
\mathcal{A}_1 := \mathcal{A} \cup \{C(x)\} \\
\mathcal{A}_2 := \mathcal{A} \cup \{C'(x)\}.
\]

**(∃-rule)** If $\mathcal{A}$ contains $(\exists R.C)(x)$ but no $z$ s.t. $R(x, z)$ and $C(z)$,

\[
\mathcal{A} := \mathcal{A} \cup \{R(x, y), C(y)\} \quad \text{where } y \text{ is fresh}.
\]

**(∀-rule)** If $\mathcal{A}$ contains $(\forall R.C)(x)$ and $R(x, t)$ but not $C(t)$,

\[
\mathcal{A} := \mathcal{A} \cup \{C(t)\}.
\]

---

### Some fine points about tableaux

Negation (for post-processing and pre-processing)

- No tableau rule for negation — except to close off a branch (contradictory pair $p, \neg p$)
- Push negations inside before applying tableau rules.

For non-empty T-Box $\mathcal{T}$, add the rule that

for every $C \sqsubseteq D$ in $\mathcal{T}$ and $t \in \mathcal{T}_m$,

if $\mathcal{A} \models C(t)$ and not $D(t) \in \mathcal{A}$,

\[
\mathcal{A} := \mathcal{A} \cup \{D(t)\}
\]

N.B. Keep $\models$ decidable (and $\mathcal{T}$ and $\mathcal{T}_m$ finite)