A Cute Trick for Calculating Saturated Sets

Asta Halkjær From

Copenhagen Logic Gathering, 2023-03-17

The Scheme of Things

I have formalized completeness for five logics in Isabelle/HOL

- Propositional sequent calculus and tableau
- First-order and hybrid logic natural deduction
- Modal logic System K (axiomatic system)

The recipe:

- Build MCS with my generic, transfinite formalization of Lindenbaum’s lemma
- **Isabelle/HOL calculates the saturated set conditions for the logic**
- Prove that the MCSs fulfil the conditions
- Profit

Saturated Sets

Saturated sets are saturated in both directions via *conditions* like:

$$p \longrightarrow q \in H \qquad \longleftrightarrow \qquad p \in H \text{ IMPLIES } q \in H \qquad (*)$$

Membership equals satisfiability, so we can prove completeness:

- Build Maximal Consistent Sets (MCSs) using Lindenbaum's lemma
- Prove that any MCS is saturated
- Any non-derivable formula is then falsifiable (its negation is consistent)

But how exactly did we arrive at condition (*)

Semantics

Take propositional logic as a running example.

Syntax: falsity, propositional symbols, implication.

Semantic brackets lift the interpretation I to formulas p, q:

$$
\begin{aligned}
[\![\_]\!]\ \bot \quad &\longleftrightarrow \quad \mathtt{False} \\
[\![I]\!]\ (\ddagger P) \quad &\longleftrightarrow \quad I\ P \\
[\![I]\!]\ (p \longrightarrow q) \quad &\longleftrightarrow \quad [\![I]\!]\ p \longrightarrow [\![I]\!]\ q
\end{aligned}
$$

(code from the Isabelle/HOL formalization)

Semics

Punch a hole in the sem[ant]ics, replacing the recursive call with `rel`:

```
semics _ _ ⊥             ↔     False
semics I _ (‡P)          ↔     I P
semics I rel (p ⟶ q)     ↔     rel I p ⟶ rel I q
```

Now we can express other properties based on subformulas.

Saturated Sets Redux

Saturated sets are saturated in both directions, e.g.:

$$p \longrightarrow q \in H \qquad \longleftrightarrow \qquad p \in H \text{ IMPLIES } q \in H$$

The connection between object-logical $\longrightarrow$ and meta-logical IMPLIES?

It is exactly the *semics*:

```
semics (hmodel H) (rel H) p ↔ rel H (hmodel H) p
```

Under the model induced by `H`, namely `hmodel H`, the relation `rel H` holds for the subformulas of `p` exactly when it holds for `p`.

Example

Take the usual term model:

$$\mathtt{hmodel}\ H \equiv \lambda P.\ \ddagger P \in H \qquad \text{and} \qquad \mathtt{rel}\ H\ \_\ p = (p \in H)$$

and the equation from before:

$$\mathtt{semics}\ (\mathtt{hmodel}\ H)\ (\mathtt{rel}\ H)\ p \leftrightarrow \mathtt{rel}\ H\ (\mathtt{hmodel}\ H)\ p$$

For each syntactic constructor, it reduces to a saturated set condition:

$$\begin{array}{lcl} \mathtt{False} & \leftrightarrow & \bot \in H \\ \ddagger P \in H & \leftrightarrow & \ddagger P \in H \\ (p \in H \to q \in H) & \leftrightarrow & (p \to q \in H) \end{array}$$

First-Order Logic

Evaluate universal quantifiers by extending (⨾) the variable denotation E

```
semics (E, F, G) rel (∀p)   ↔   ∀x. rel (x ⨾ E, F, G) p
```

The term model again:

```
hmodel H ≡ (#, †, λP ts. ‡P ts ∈ H)
```

We need to apply E as a substitution to account for p's context:

```
rel H (E, _, _) p = (sub-fm E p ∈ H)
```

The resulting saturated set condition:

```
(∀x. ⟨x⟩p ∈ H)   ↔      ∀p ∈ H
```

Hybrid Logic I

Abridged semics:

```
semics (_, g, w) _ (·i)       ⟷  w = g i
semics (M, g, w) rel (◇p)     ⟷  ∃v ∈ R M w. rel (M, g, v) p
semics (M, g, _) rel (@i p)   ⟷  rel (M, g, g i) p
```

We account for the context by labeling the formula `p` with the world `i`:

```
rel H (_, _, i) p = ((i, p) ∈ H)
```

Thus we calculate saturated sets of labeled formulas

Hybrid Logic II

The model is based on equivalence classes `[i]` of nominals where two nominals `i` and `k` are equivalent (wrt. `H`) when `(i, ·k) ∈ H`

The saturated set conditions become (for all `i`):

```
[i] = [k]                          ↔    ([i], ·k) ∈ H
([k], p) ∈ H                       ↔    ([i], @k p) ∈ H
(∃v∈reach H [i]. (v, p) ∈ H)       ↔    ([i], ◇p) ∈ H
```

where `reach H i ≡ {[k] | (i, ◇k) ∈ H}`

Fin

Future work: *Downwards* saturated sets (Hintikka)?

Need to consider each syntactic constructor *and their negation:*

$$p \longrightarrow q \in H \quad \longrightarrow \quad \neg\, p \in H \text{ OR } q \in H$$
$$\neg\,(p \longrightarrow q) \in H \quad \longrightarrow \quad p \in H \text{ AND } \neg\, q \in H$$

References:

Formalization in the Archive of Formal Proofs:
https://devel.isa-afp.org/entries/Synthetic_Completeness.html

My PhD thesis:
https://people.compute.dtu.dk/ahfrom/ahfrom-thesis.pdf