Abstract Consistency Properties for Hybrid Logic

Smullyan's Unifying Principle in Isabelle/HOL

Logic: from Arthur Prior to Computation and Cognition

Workshop in Honour of Professor Torben Braüner - 2025-05-15

Asta Halkjær From, University of Copenhagen

Agenda

Concrete Consistency 3
Abstract Consistency — Smullyan 7
Abstracter Consistency — Fitting 13
Abstract Abstract Consistency — From 15
Application: “Bounded” First-Order Logic 19
Application: Second-Order Logic 22
Application: Prior’s Ideal Language 24
Conclusion 30

Concrete Consistency

Concrete Consistency

- We have a concrete language

$$\varphi := p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \to \varphi \mid \forall x.\, \varphi(x) \mid \exists x.\, \varphi(x) \mid \ldots$$

- We have a concrete calculus with concrete proof rules:

$$\frac{\varphi \in \Gamma}{\Gamma \vdash \varphi}\,\text{Assm} \qquad \frac{\Gamma \vdash \varphi \quad \Gamma \vdash \psi}{\Gamma \vdash \varphi \wedge \psi}\,\wedge\text{I}$$

$$\frac{\Gamma \vdash \varphi \to \psi \quad \Gamma \vdash \varphi}{\Gamma \vdash \psi}\,\to\text{E} \qquad \frac{\Gamma \vdash \forall x.\ \varphi(x)}{\Gamma \vdash \varphi(t)}\,\forall\text{E}$$

- A set Γ is consistent wrt. $\vdash$ (concretely) when we cannot derive a contradiction from it (e.g. $\Gamma \vdash \varphi$ and $\Gamma \vdash \neg\varphi$).

Concrete Maximal Consistency

- A consistent set Γ is a maximally consistent set (MCS) when it contains every formula consistent with it:

$$\text{if } \Gamma \subseteq \Delta \text{ and } \Delta \text{ consistent, then } \Gamma = \Delta$$

- We can build an MCS by trying to add every formula and taking the union $\Delta = \bigcup_i \Delta_i$ (Lindenbaum-Tarski):

$$\begin{aligned}
\Delta_0 &= \Gamma \\
\Delta_{i+1} &= \{\varphi_i, \psi(a)\} \cup \Delta_i \text{ if consistent and } \varphi_i = \exists x.\, \psi(x) \\
\Delta_{i+1} &= \{\varphi_i\} \cup \Delta_i && \text{otherwise if consistent} \\
\Delta_{i+1} &= \Delta_i && \text{otherwise}
\end{aligned}$$

Concrete MCS Properties

- Our MCS Γ has properties corresponding to our calculus:

$$\text{if } \varphi \in \Gamma \text{ and } \psi \in \Gamma \text{ then } \varphi \wedge \psi \in \Gamma$$

$$\text{if } \varphi \to \psi \in \Gamma \text{ and } \varphi \in \Gamma \text{ then } \psi \in \Gamma$$

$$\text{if } \forall x.\, \varphi(x) \in \Gamma \text{ then } \varphi(t) \in \Gamma \text{ for all } t$$

- Think: if we can derive φ from formulas in Γ, then $\varphi \in \Gamma$
- We can prove completeness by building a model over Γ.
- Note: There are some nuances here with respect to the *cut* rule and *upwards* and *downwards saturation*.

Abstract Consistency – Smullyan

Abstract Consistency

- *What if we start from the properties instead of the calculus?*
- We still have a concrete language (of quantification theory)

$$\varphi := p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \to \varphi \mid \forall x.\, \varphi(x) \mid \exists x.\, \varphi(x) \mid \dots$$

- Characterize it with *Smullyan's uniform notation*:
  - *Conjunctive*, *disjunctive*, *universal* and *existential* kinds.

$\boldsymbol{\alpha}$ $\varphi \wedge \psi : \alpha_1 = \varphi, \alpha_2 = \psi \quad \neg(\varphi \to \psi) : \alpha_1 = \varphi, \alpha_2 = \neg\psi$

$\boldsymbol{\beta}$ $\varphi \to \psi : \beta_1 = \neg\varphi, \beta_2 = \psi \quad \neg(\varphi \wedge \psi) : \beta_1 = \neg\varphi, \beta_2 \neg\psi$

$\boldsymbol{\gamma}$ $\forall x.\, \varphi(x) : \gamma(t) = \varphi(t) \quad \neg(\exists x.\, \varphi(x)) : \gamma(t) = \neg\varphi(t)$

$\boldsymbol{\delta}$ $\exists x.\, \varphi(x) : \delta(t) = \varphi(t) \quad \neg(\forall x.\, \varphi(x)) : \delta(t) = \neg\varphi(t)$

Consistency Property

- A family of sets C is a (first-order) *consistency property* when all sets $S \in C$ obey:

conflict for all p, not both $p \in S$ and $\neg p \in S$

banned $\bot \notin S$ (and $\neg\top \notin S$)

double neg. if $\neg\neg\varphi \in S$ then $\{\varphi\} \cup S \in C$

alpha if $\alpha \in S$ then $\{\alpha_1, \alpha_2\} \cup S \in C$

beta if $\beta \in S$ then $\{\beta_1\} \cup S \in C$ or $\{\beta_2\} \cup S \in C$

gamma if $\gamma \in S$ then $\{\gamma(t)\} \cup S \in C$
for every (closed term) t (…)

delta$_\mathrm{E}$ if $\delta \in S$ then $\{\delta(a)\} \cup S \in C$ for *some* a (…)

- Think: we have *evidence* for every member formula.

Maximal Element?

- Set theory: under the axiom of choice, *finite character* of a family of sets C guarantees a maximal member wrt. $\subseteq$:

```
‹finite_char C ≡
  ∀S. S ∈ C ⟷ (∀S' ⊆ S. finite S' → S' ∈ C)›
```

- Problem: imposing finite character might break **delta$_\mathbf{E}$**.
  - Exercise for the reader.

- Solution: interpret it universally rather than existentially.

delta$_\mathbf{A}$ if $\delta \in S$ then $\{\delta(a)\} \cup S \in C$ for every *new* a (...)

- How do we recover **delta$_\mathbf{E}$**? Manually!
  - As earlier in the Lindenbaum-Tarski construction.

Maximal Element!

1. Take our consistency property C
2. Impose finite character to obtain $C^* \supseteq C$
3. Any consistent set $\Gamma \in C$, has an MCS $\Delta \supseteq \Gamma \in C^*$
4. (Add witnesses manually when constructing Δ)

- Our MCS has *Hintikka* properties by construction
  - Each condition on the family of sets gives the maximal element a corresponding property.
  - E.g. for **alpha**: if $\alpha \in \Delta$, then $\alpha_1 \in \Delta$ and $\alpha_2 \in \Delta$
- So we can prove *model existence* in the abstract:
  - For any consistency property C and set $S \in C$, we can build a model that satisfies all formulas $\varphi \in S$.

But Why?

Gödel's completeness theorem concrete consistency constitutes a consistency property.

Gentzen's Hauptsatz our consistency property has the sub-formula property, so we get cut-elimination for free

Compactness theorem the sets where every finite subset is satisfiable constitute a consistency property

Downward Löwenheim-Skolem the satisfiable sets constitute a consistency property, and the domain of our model is countable

Craig's interpolation theorem the sets with an interpolant-free partition constitute a consistency pro...

Abstracter Consistency — Fitting

The Core of the Argument

- *What if we take a different language?*

- Fitting applied consistency properties to both term-modal logic and intuitionistic logic:

modal if $\langle t \rangle \varphi \in S$ then $\{\varphi\} \cup S^{[t]} \in C$

- What matters is that the conditions on our consistency property respect *Fitting's three-step recipe*:

  1. Subset close the family
  2. Satisfy $\mathbf{delta_A}$ via parameter substitutions
  3. Impose finite character

- Guarantees a maximal element for model existence.

Abstract Abstract Consistency — From

Abstract Abstract Consistency

- *What if we take any **consistency kind**.*
- There are two types:
  - Wits W:
    - W: witnesses a (δ-)formula using a given parameter.
  - Cond $P\ H$:
    - P: relates formulas (e.g. α) with their consistency conditions (e.g. $\lambda C\ S.\ \{\alpha_1, \alpha_2\} \cup S \in C$).
    - H: gives the corresponding Hintikka property.
- We carve out those that respect *Fitting's three-step recipe.*
- We get the ingredients for model existence without fixing a calculus, a consistency property *or even a language.*

In Isabelle/HOL

The Params locale defines parameter substitutions.

```
locale Consistency_Kind = Params map_fm params_fm
 for map_fm :: ‹('x ⇒ 'x) ⇒ 'fm ⇒ 'fm›
 and params_fm :: ‹'fm ⇒ 'x set› +
 fixes K :: ‹('x, 'fm) kind›
 assumes respects_close:
  ‹⋀C. sat_E K C ⟹ sat_E K (close C)›
 and respects_alt:
  ‹⋀C. sat_E K C ⟹ subset_closed C ⟹ sat_A K
(mk_alt_consistency C)›
 and respects_fin:
  ‹⋀C. subset_closed C ⟹ sat_A K C ⟹ sat_A K
(mk_finite_char C)›
 and hintikka:
  ‹⋀C S. sat_E K C ⟹ S ∈ C ⟹ maximal C S ⟹ sat_H K S›
```

Pre-Defined Kinds

- For a user-given predicate $\leadsto$ we can define the following:
  - (Under some natural conditions on each $\leadsto$.)

Confl ‹ps $\leadsto_\chi$ qs ⟹ cond ps
(λ_ S. set qs ∩ S = {})›

Alpha ‹ps $\leadsto_\alpha$ qs ⟹ cond ps
(λC S. set qs ∪ S ∈ C)›

Beta ‹ps $\leadsto_\beta$ qs ⟹ cond ps
(λC S. ∃q ∈ set qs. {q} ∪ S ∈ C)›

Gamma ‹ps $\leadsto_\gamma$ (F, qs) ⟹ cond ps
(λC S. ∀t ∈ F S. set (qs t) ∪ S ∈ C)›

Modal ‹ps $\leadsto_\square$ (F, qs) ⟹ cond ps
(λC S. set qs ∪ F S ∈ C)›

Application:
“Bounded” First-Order Logic

Restricted Instantiation

- Consider first-order logic with the following sort of rule:

$$\frac{\Gamma \vdash \forall x.\ \varphi(x) \qquad t \text{ is a sub-term of } \Gamma, \varphi}{\Gamma \vdash \varphi(t)}\ \forall\text{E}$$

- Can we give an easy semantic proof?
- Make use of the ability to bound our **gamma** kind:
  - `‹[ ⊥ ] ↝χ [ ⊥ ]›`
  - `‹[ ¬ (⋅P ts) ] ↝χ [ ⋅P ts ]›`
  - `‹[ ¬ (p ⟶ q) ] ↝α [ p, ¬ q ]›`
  - `‹[ p ⟶ q ] ↝β [ ¬ p, q ]›`
  - `‹[ ∀p ] ↝γ (terms, λt. [ ⟨t⟩p ])›`
  - `‹δ (¬ ∀p) x = [ ¬ ⟨⋆x⟩p ]›`

Completeness

- The framework says we can extend a consistent set to an MCS with Hintikka properties corresponding to our consistency property.
- Build a Herbrand model with a domain of sub-terms.
- Prove a truth lemma by induction on the size of the member formula
  - The Hintikka properties discharge each case.
- Result: A proof of completeness in Isabelle/HOL for a natural deduction system with the restricted proof rule.
  - (and also a mechanized proof of compactness.)

Application:
Second-Order Logic

Scaling Up

- Joint work with Anders Schlichtkrull, Aalborg University Copenhagen.
- Quantify over functions and predicates besides terms.
- **gamma**s for different quantifiers at different types:
  - `‹[ ∀p ] ↝γ (λt. [ ⟨t/0⟩p ])›`
  - `‹[ ∀P p ] ↝γP (λs. [ ⟨s/0⟩P p ])›`
  - `‹[ ∀F p ] ↝γF (λs. [ ⟨s/0⟩F p ])›`
- Each **gamma** can only instantiate with one type of term
  - compose our consistency property of multiple **gamma**s.
- Mechanized completeness as before.

Application:
Prior’s Ideal Language

Very Strong Hybrid Logic

- Based on work by Blackburn, Braüner and Kofod.

```
p := •i | ·P | ¬ p | p ∧ p | □p
     | @i p | A p | ↓p | ∀p
```

- Easy to express semantics in Isabelle/HOL:
  - ‹(M, _) ⊨ A p ⟷ (∀v ∈ 𝔚 M. (M, v) ⊨ p)›
  - ‹(M, w) ⊨ ↓ p ⟷ (M⦇𝔑 := (w ⋗ 𝔑 M)⦈, w) ⊨ p›
  - …

- Propositional quantification over *admissible* propositions:
  - ‹(M, w) ⊨ ∀ p ⟷
    (∀P ∈ Π M. (M⦇𝔙 := (P ⋗ 𝔙 M)⦈, w) ⊨ p)›

- This quantification gives some second-order expressivity.

Conflicts and Disjunctives

- *Can we fit this very strong hybrid logic into our setup?*
- *What properties do we need in order to build a model?*
- We label formulas by pairing them with a nominal: (i, φ)
  - φ holds at the world denoted by i.
  - (I need to see what happens if I just write $@_i\varphi$ instead.)
- Conflicts are natural:
  - `‹[ (i, ¬ ▪P) ] ⇝✗ [(i, ▪P)]›`
  - `‹[ (i, ¬ •k) ] ⇝✗ [(i, •k)]›`
- We only have one disjunctive condition:
  - `‹[ (i, ¬ (p ∧ q)) ] ⇝β [(i, ¬ p), (i, ¬ q)]›`

Conjunctives

- We have many conjunctive conditions.
  - $\langle[\ (i,\ \neg\ \neg\ p)\ ]\ \rightsquigarrow_{\alpha}\ [(i,\ p)]\rangle$
  - $\langle[\ (i,\ p \wedge q)\ ]\ \rightsquigarrow_{\alpha}\ [(i,\ p),\ (i,\ q)]\rangle$
- We can express nominal symmetry and transitivity.
  - $\langle[\ (i,\ \bullet k)\ ]\ \rightsquigarrow_{\alpha}\ [(k,\ \bullet i)]\rangle$
  - $\langle[\ (i,\ \bullet k),\ (i,\ p)\ ]\ \rightsquigarrow_{\alpha}\ [(k,\ p)]\rangle$
- Satisfaction operators and downarrow binders fit right in:
  - $\langle[\ (i,\ @k\ p)\ ]\ \rightsquigarrow_{\alpha}\ [(k,\ p)]\rangle$
  - $\langle[\ (i,\ \neg\ @k\ p)\ ]\ \rightsquigarrow_{\alpha}\ [(k,\ \neg\ p)]\rangle$
  - $\langle[\ (i,\ \downarrow\ p)\ ]\ \rightsquigarrow_{\alpha}\ [\ (i,\ \langle i\rangle_{i}\ p)\ ]\rangle$
  - $\langle[\ (i,\ \neg\ \downarrow\ p)\ ]\ \rightsquigarrow_{\alpha}\ [\ (i,\ \neg\ \langle i\rangle_{i}\ p)\ ]\rangle$

Conjunctives and Universals

- The box modality is truly *local*:
  - `‹[ (i, □p), (i, ◇(•k)) ] ⇝α [(k, p)]›`
- The global modality is truly *universal*:
  - `‹[ (i, A p) ] ⇝γi (λk. [ (k, p) ])›`
- We want nominal reflexivity *unconditionally*:
  - `‹[] ⇝γi (λi. [ (i, •i) ])›`
- Limit propositional quantification to *soft-qdf* formulas:
  - `‹[ (i, ∀ p) ] ⇝γp (λ_. {q. softqdf q}, λq. [ (i, ⟨q⟩p p) ])›`
  - These correspond to *admissible* propositions.
  - The Isabelle/HOL mechanization helped clarify this.

Existentials and Completeness

- Finally, we have witnesses for existential formulas:
  - `‹δ (i, ¬ □p) k =`
    `[(○k, ¬ p), (i, ◇ (• (○ k)))]›`
  - `‹δ (i, ¬ A p) k = [ (○k, ¬ p) ]›`
  - `‹δ (i, ¬ ∀ p) P = [ (i, ¬ ⟨⋅(○P)⟩_p p) ]›`
- From this consistency property, we get MCSs with corresponding Hintikka properties.
- Completeness a la Blackburn, Braüner and Kofod
  - But for a natural deduction system.
  - With equivalence classes of nominals as worlds.
  - Without tweaking the Lindenbaum-Tarski construction.

Conclusion

So What?

- Consistency properties are *expressive*.
- They provide a declarative interface for building MCSs.
  - No need to tweak the Lindenbaum-Tarski enumeration of consistent sets for each and every new logic.
  - Simply write down a consistency property.
- How lucky did we get?
  - Can we handle higher-order logic?
  - Can we handle more exotic non-classical logics?
  - We derived the pre-defined kinds.
    - We did not bake them in from the beginning.
- (How much could we do in a constructive meta-logic?)