Fully abstract valuations for subgames

Wilfrid Hodges
Queen Mary, University of London
November 2005

`www.maths.qmul.ac.uk/~wilfrid/social.pdf`

$$
\begin{tikzcd}
\text{Sentences} \arrow[r] \arrow[d, dash] & \text{Sentence meanings} \\
\text{Phrases} &
\end{tikzcd}
$$

The extension problem for semantics is:

Given an assignment of meanings to sentences,
to extend this assignment to all phrases,
so that the meaning of each phrase exactly captures its contribution to the meanings of sentences containing it.
(The meaning assignment is 'fully abstract'
with respect to sentence meanings.)

Some writers refer to this (I think misleadingly) as the *compositionality problem*.

$$
\begin{tikzcd}
\text{Sentences} \arrow[r] \arrow[d, dash] & \text{Sentence meanings} \arrow[d, dash, dashed] \\
\text{Phrases} \arrow[r, dashed] & \text{Phrase meanings}
\end{tikzcd}
$$

Classical example: the Tarski truth definition, starting from an intuitive notion of when a formal sentence is true.

In the mid 1990s the same problem was solved for some languages (due to Henkin and Hintikka) where the notion of truth for sentences is given in terms of games.
The solution is the *trump semantics*.

$$
\begin{tikzcd}
\text{Sentences} \arrow[r] \arrow[d, dash] & \text{Games} \arrow[r] & \text{Sentence meanings} \arrow[d, dash, dashed] \\
\text{Formulas} \arrow[rr, dashed] & & \text{Formula meanings}
\end{tikzcd}
$$

Unlike Tarski's notion of satisfaction, trumps are not a notion that we already had before solving the problem.
This created a new problem: Give an intuitive justification for the trump semantics.

Two possible approaches

These approaches both put the blame on having to translate between formulas and games.

I. (This talk) Formulate the extension problem directly for games, and solve it there.

II. (Following Väänänen) Remove the games altogether, and redefine the sentence semantics using an intuitive 'dependence' predicate.

Approach I: Removing the syntax

$$
\begin{tikzcd}
\text{Sentences} \arrow[r] \arrow[d, dash] & \text{Games} \arrow[r] \arrow[d, dash, dashed] & \text{Sentence meanings} \arrow[d, dash, dashed] \\
\text{Formulas} \arrow[r, dashed] & \text{Subgames} \arrow[r, dashed] & \text{Formula meanings}
\end{tikzcd}
$$

Our analysis of games is strongly influenced by

> Rohit Parikh, 'Propositional logics of programs: new directions', in *Foundations of Computation Theory*, LNCS 158 (1983).

In the logical problem the games were for two players.
This is inessential for our purposes.
We assume given a set $\mathcal{P}$ of *players*.

We assume given a set $\mathcal{Q}$ of *questions*,
each of which has a set of answers.

A *(data) state* is a function defined on a set of questions,
taking each of these questions to one of its answers.

We describe the set of *subgames*.
Each subgame G will have a *matter* $m(G)$,
which is a set of questions.
We say how G is played at each state with domain $\supseteq m(G)$.

A *game* is a subgame G with $m(G) = \emptyset$.

NB Our notion of subgame is not that of Selten (1975).
A Selten subgame is essentially a pair (G, s) where
G is a subgame that can be played at state s.

An *atomic subgame* G has no moves,
and yields an immediate payoff $G[s]$ at each state s
whose domain includes $m(G)$;
the payoff depends only on $s \restriction m(G)$.

For any subgame G, player P and question q,
the subgame PqG has matter $m(G) \setminus \{q\}$.

At any state s with domain $\supseteq m(PqG)$,
player P moves by choosing an answer to q
and adding it to s (if necessary replacing
a previous answer to q), yielding a state s'.

The subgame then proceeds as G at s'.

For some sets $\{G_i : i \in I\}$ of subgames and some players P,
the game $P(G_i)_{i\in I}$ has matter $\bigcup_{i\in I} m(G_i)$.
For each state s whose domain includes this matter,
the game $P(G_i)_{i\in I}$ is played at s as follows:

Player P chooses $i \in I$, and the game proceeds as G_i at s.

The set of subgames is the closure of
the set of atomic subgames under Pq, $P_{i\in I}$.

If H is a subgame occurring in game G
(clear what this means!), and $m(H) = m(H')$,
then we can describe a game $G(H'/H)$ got by
replacing the occurrence of H by H'.
(The notation is loose; it should indicate the occurrence.)
We assume the set of games is closed under these
substitutions.

Suppose for example that each payoff is
a win for just one player.

For simplicity we localise the values of games.

For each player P, define the *P-value* of a game G to be

> T if P has a winning strategy for G,
> F otherwise.

This is independent of the state.

For each subgame G, define the ***P-value***$^\star_P$ of G, $v^\star_P(G)$, to be the set of states at which P has a winning strategy for G.

Theorem If $m(H) = m(H')$ and $v^\star_P(H) = v^\star_P(H')$,
then for every game G in which H occurs,
G has the same P-value as $G(H'/H)$.

Hence $(m, v^\star_P)_{P\in\mathcal{P}}$ provides sufficient information for a 'semantics' on subgames which is fully abstract wrt games.
One can write down conditions on the class of games,
under which this information is also necessary.

Theorem If $m(H) = m(H')$ and $v^\star_P(H) = v^\star_P(H')$,
then for every game G in which H occurs,
G has the same P-value as $G(H'/H)$.

Proof sketch By induction on the complexity of the subgame G,
we prove that P has a winning strategy
at the same states in G and in $G(H'/H)$.

The key steps are at $P'qG$ where $P' \neq P$.

If P has a winning strategy in $P'qG$ at s,
then for every answer a to q,
P has a winning strategy at $s\hat{\ }a$ in G.

So by induction hypothesis,
P has a winning strategy σ_a at $s\hat{\ }a$ in $G(H'/H)$,
for every answer a to q.

Gluing the σ_a together gives
a winning strategy for P at s in $P'qG(H'/H)$. $\square$

We extend easily, e.g. to zero-sum 2-player games with real payoffs.

For each real λ,
a λ-*win* for $\exists$ is a play where $\exists$ receives payoff $\geqslant \lambda$,
and a λ-*win* for $\forall$ is a play where $\exists$ receives payoff $< \lambda$.

By above, each game G is determined for λ-wins.
Define the P-*value* of G to be the sup of the λ such that
$\exists$ has a winning strategy for λ-wins.

For each subgame H, define $v^\star(H)$ to be
the function taking each state s with domain $m(H)$
to

$$\sup\{\lambda : \exists \text{ has a strategy guaranteeing payoff at least } \lambda \text{ when } H \text{ is played at } s\}.$$

Again $(m, v^\star_P)_{P \in \mathcal{P}}$ is adequate for a fully abstract semantics
on subgames wrt games.

Now return to the simple win / lose case,
but impose imperfect information.

Instead of PqG we consider the game $(Pq/\pi)G$,
where π is a partition of the set of states with domain
$m(G) \setminus \{q\}$.
Then π induces a partition of all states with domain
including $m(G) \setminus \{q\}$.

In a strategy for P, her choice when playing $(Pq/\pi)G$ at s
must be the same as that at t whenever $s \sim_\pi t$.
We call this the *uniformity condition* on strategies.

The previous definition of $v^\star_P(H)$ fails.
In the proof of the theorem, there is no guarantee that the
strategies σ_a glue together into a strategy σ.
The uniformity condition might fail.

Instead we define the value $v^\star_P(H)$ to be the set of sets X of
states with domain $m(H)$ such that there is strategy for P
in H which is winning at s for all $s \in X$.

We repeat the Theorem, but under imperfect information
and with the new definition of $v^\star_P$.

Theorem If $m(H) = m(H')$ and $v^\star_P(H) = v^\star_P(H')$,
then for every game G in which H occurs,
G has the same value as $G(H'/H)$.

Proof sketch Again the key step is at $P'qG$ where $P' \neq P$.

Let X be a set of states with domain $m(P'qG)$.

Suppose P has a strategy τ for $P'qG$ which wins at each $s \in X$.

Write $X \times q$ for the set of states $s\hat{}a$ where $s \in X$ and a is an answer to q.
Moving one step forwards, P has a strategy τ' for G which wins at each $s' \in X \times q$.

By induction hypothesis, P has a strategy σ' which wins in $G(H'/H)$ at every $s' \in X \times q$.

Since P doesn't move at $P'q$,
σ' is already a strategy for P at $P'qG(H'/H)$ that wins at all $s \in X$. $\square$

Arguably this proof is simpler than the previous one; the gluing step is left out.

The value $v_P^\star(H)$ is a *set of sets of states.*
This corresponds exactly to the trump semantics:
in place of *satisfying assignments* we have
uniformly satisfying sets of assignments.

Approach II: Removing the games

Following Jouko Väänänen, we introduce a relation

‘x_n depends only on $x_1, \ldots, x_{n-1}$’.

Then for example

$$\forall x \exists y \forall z (\exists w/xy) \phi(x, y, z, w)$$

is equivalent to

$$\forall x \exists y \forall z \exists w (\phi(x, y, z, w) \wedge \text{‘}w \text{ depends only on } z\text{’}).$$

This relation appears in English and seems easy to understand:

Serge Lang, *Algebraic Number Theory* (2nd edition) p. 335:

“Let $0 < a \leq 1$, and m an integer with $|m| \geq 2$. Let $s = \sigma + iT_m$ with $-a \leq \sigma \leq 1 + a$ and T_m as above. Then

$$|\xi'/\xi(s)| \leq b(\log |m|)^2,$$

where b is a number depending on a but not on m and σ.”

Now the problem is to give a semantics for this phrase *in ordinary English*.

We need a class of phrases that require similar treatment. For example

> Different people have different religions.

Clearly we can't interpret this using the class

$$\{x : x \text{ is a different religion}\}.$$

The key is to follow

> Barbara Partee, 'Noun phrase interpretation and type-shifting principles', *Formal Semantics* ed. Portner and Partee, Blackwell 2002, pp. 357–381.

With Partee we assume that English semantics allows some kinds of phrase to *raise types* systematically.

Technically we allow raising from e to $\langle \ldots \langle e, e, \rangle, \ldots , e\rangle$.

Specifically, given any index set I, we can replace unindexed semantic values by families of values indexed by I.

For example 'ginger cat' has semantics

$$\{x : x \text{ is ginger}\} \cap \{x : x \text{ is a cat}\}.$$

Type raising by fibring over I gives

$$\{(f : I \to X) : \forall i, f(i) \text{ is ginger}\} \cap$$
$$\{(f : I \to X) : \forall i, f(i) \text{ is a cat}\}.$$

In a loose notation, 'ginger$\uparrow^I$ cat$\uparrow^I$'.

We allow a quantifier to introduce an indexing set I and then type-raise over I throughout its scope.

For example

> Most ginger cats are male.
>
> There is I such that I is a majority of ginger cats and male$\uparrow^I$ (id_I).

Then we can interpret ‘different’ as acting directly on indexed families, without raising:

different($f : I \to X$) $\Leftrightarrow$ f is injective.

For example ‘Different people have different religions’ comes out as

For any (typical?) set I of people, if different(id_I) then different(g).

where g is the function ($i \mapsto i$’s religion).

Another:

Countries at different latitudes have different vegetation.

If I is a set of countries and different(f) then different(g).

where f is the function ($i \mapsto i$’s latitude)
and g is the function ($i \mapsto i$’s vegetation).

Now we can interpret ‘f_2 depends only on f_1’:

For all $i, j \in I$, if $f_1(i) = f_1(j)$
then $f_2(i) = f_2(j)$

This exactly agrees with Väänänen’s semantics for the dependency relation.

Note that

y is a function of x

is ambiguous between

- keeping x fixed keeps y fixed, and
- changing x changes y.

(For the second, ‘The quality of your essay is a function of the time you put into it’.)

The fact that even linguists miss this ambiguity is a warning that this is slippery terrain.