This is a greatly expanded version of an APMP presentation, 10 December 2010.
Pragmatism and logic?

- Pragmatism is a perspective on the nature of belief and meaning—in the latter case, promoting a method for defining words and concepts in terms of practices and their effects.

- The following material is an overview of a project aimed at showing how a pragmatist method of definition reaches to the foundations of logic and mathematics and thus grounds them in practices outside of the use of words and sentences.

This material covers two main topics:

I. A brief discussion of pragmatism, particularly as a perspective on meaning.

II. A lengthy illustration / a running example
   - to show how the method can be formalized, and
   - to show how it explains analytic and material truth.
I. Pragmatism:

- The following material is an overview of (part of) the second of three monographs dealing respectively with:
  1. some history of classical pragmatism
  2. pragmatist semantics
  3. pragmatist pragmatics

- The first musters materials from the writings of Peirce and James to show that pragmatism originally was a conception of belief and a related conception of meaning.

- The pragmatic maxim is central:
  
  “Consider what effects, which might conceivably have practical bearings, we conceive the object of our conception to have. Then, our conception of these effects is the whole of our conception of the object.” [Peirce 1878 (“How to Make Our Ideas Clear”) EP1:132]

  Peirce insists that this is a maxim about meaning—a “maxim of logic.” It promotes a method, not a doctrine.

- But the maxim as stated is ambiguous, to say the least; and deep differences and disagreements between Peirce and James did not help matters.
In the first monograph I argue that there are two valid readings of the maxim:

- an inferentialist reading
- an operationalist reading

For instance:

- Brandom (following James) has developed an inferentialist semantics.
  - E.g., what’s the meaning of ‘∈’?
  - Answer: It’s a matter of how the symbol is used as part of a vocabulary.

- The present work (following Peirce) aims to develop an operationalist semantics.
  - E.g., what’s the meaning of ‘∈’?
  - Answer: It’s a matter of how we identify, compare, and manipulate collections of things—in herds, in boxes, in baskets, in sacks, in buckets, in pockets, in bank accounts, in portfolios, . . . —

Each approach captures an essential aspect of pragmatist semantics.

But the focus here is on operationalist semantics. . . . How does it work?
II. An Example

We will focus on a particular interpreted first-order language, $L_{lpl}$, to show how to operationally “clarify” the meanings of its predicate symbols.

This method of clarification will be explained by chasing around through a network of languages which “correspond” one to another in one way or another, semantically speaking. One of these ($L_{abd}$) is explicitly operational in character.

In short, we will show how to use an operation-based language $L_{abd}$ to interpret the meanings of the predicate symbols of $L_{lpl}$.
Why proceed this way?

– We will cover some familiar territory making it easier to see how operationalist semantics and extensionalist semantics work together.
– This tack will also help to clarify how an operationalist semantics can explain things like analytic consequence and material inference.

A note about notation:

– The use of different fonts and the superscripting and subscripting scheme may seem to be excessively complicated, but something like this is needed to make distinctions and track correspondences among a large number of different formal languages.
The example: $L_{lpl}$ (a blocks language).

- $L_{lpl}$ is a language used in Barwise and Etchemendy, *Language, Proof, and Logic*.
- $L_{lpl}$ has nineteen predicate symbols and six names (all of which will be regarded as “predicate symbols” reflecting properties of and relations among blocks).
- We want to show in steps how to operationalize (clarify) these nineteen (+ six = twenty-five) predicate symbols.
**$L_{lpl}$ : A first-order blocks language**

<table>
<thead>
<tr>
<th><strong>Atomic Sentence-Type</strong></th>
<th><strong>Intended Interpretation</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Tet($a$)</td>
<td>$a$ is a tetrahedron</td>
</tr>
<tr>
<td>(2) Cube($a$)</td>
<td>$a$ is a cube</td>
</tr>
<tr>
<td>(3) Dodec($a$)</td>
<td>$a$ is a dodecahedron</td>
</tr>
<tr>
<td>(4) Small($a$)</td>
<td>$a$ is small</td>
</tr>
<tr>
<td>(5) Medium($a$)</td>
<td>$a$ is medium</td>
</tr>
<tr>
<td>(6) Large($a$)</td>
<td>$a$ is large</td>
</tr>
<tr>
<td>(7) Smaller($a, b$)</td>
<td>$a$ is smaller than $b$</td>
</tr>
<tr>
<td>(8) Larger($a, b$)</td>
<td>$a$ is larger than $b$</td>
</tr>
<tr>
<td>(9) LeftOf($a, b$)</td>
<td>$a$ is located nearer to the left edge of the grid than $b$</td>
</tr>
<tr>
<td>(10) RightOf($a, b$)</td>
<td>$a$ is located nearer to the right edge of the grid than $b$</td>
</tr>
<tr>
<td>(11) BackOf($a, b$)</td>
<td>$a$ is located nearer to the back of the grid than $b$</td>
</tr>
<tr>
<td>(12) FrontOf($a, b$)</td>
<td>$a$ is located nearer to the front of the grid than $b$</td>
</tr>
<tr>
<td>(13) SameSize($a, b$)</td>
<td>$a$ is the same size as $b$</td>
</tr>
<tr>
<td>(14) SameShape($a, b$)</td>
<td>$a$ has the same shape as $b$</td>
</tr>
<tr>
<td>(15) SameRow($a, b$)</td>
<td>$a$ is in the same row as $b$</td>
</tr>
<tr>
<td>(16) SameCol($a, b$)</td>
<td>$a$ is in the same column as $b$</td>
</tr>
<tr>
<td>(17) Adjoins($a, b$)</td>
<td>$a$ and $b$ are located on non-diagonally adjacent squares</td>
</tr>
<tr>
<td>(18) $a = b$</td>
<td>$a$ is one and the same thing as $b$</td>
</tr>
<tr>
<td>(19) Between($a, b, c$)</td>
<td>$a, b,$ and $c$ are in the same row, column, or diagonal, and $a$ is between $b$ and $c$</td>
</tr>
</tbody>
</table>
**$L_{lpl}$**: A simpler blocks-language schema

(henceforth referred to as “the blocks language”):

<table>
<thead>
<tr>
<th>Atomic Sentence-Type</th>
<th>Intended Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(1 \leq l \leq 6)$</td>
<td>$P_1^l(a)$</td>
</tr>
<tr>
<td>$(7 \leq l \leq 18)$</td>
<td>$P_2^l(a, b)$</td>
</tr>
<tr>
<td>$(l = 19)$</td>
<td>$P_{19}^l(a, b, c)$</td>
</tr>
</tbody>
</table>

Let ‘$\|\#\|_w$’ denote “the referent of $\#$ in world $w$.” For a name or variable, this will be a single *block* in $w$. For a predicate symbol, if we are not nominalists, it will be a *property of* or *relation among* blocks in $w$. 
We need to distinguish

atomic formulas: \( P_i^s(\vec{t}) \) (\( \vec{t} \) is an \( s \)-tuple of terms, including both variables \( x \) and names \( c \) as possible components.)

atomic sentence-types: \( P_i^s(\vec{c}) \) (\( \vec{c} \) is an \( s \)-tuple of names)

atomic sentences: \( S_w \models P_i^s(\vec{c}) \) (with reference to a structure \( S_w \), \( V_w \models P_i^s(\vec{t}) \) \( S_w \) and referent-assignment \( V_w \) specific to a world \( w \) where \( \|\vec{c}\|_w,\|\vec{t}\|_w \) and \( \|P_i^s\|_w \) are determined)

Sentence-types are expressions that may be used with reference to some given world where they will be true or false. Sentences (as such) are true or false.

Sentences are *essentially indexical* given this requirement of reference to a given world so as to anchor the referents of its constituents.
An $L_{lpl}$-model $\mathcal{M}_w$ is an extensional interpretation for the first-order language $L_{lpl}$—essentially a translation of $L_{lpl}$ into a first-order language of set theory.

We can use a language of set theory, $L_{setu}$, where blocks in a given blocks world serve as urelements. (If we are not nominalists, we might expand the class of urelements to include certain properties of and relations among blocks in $w$.)

$L_{setu}$: A first-order language of set theory with blocks as urelements
A fixed blocks world $w$ determines a unique (maximal) model $\mathcal{M}_w$ for the first-order blocks language $L_{lpl}$.

An $L_{lpl}$-model $\mathcal{M}_w$ consists of a pair $\langle U_w, S_w \rangle$, specific to a given blocks world $w$, where

- $U_w$ is the set of all individual blocks in $w$.
- $S_w$ is a respective extensional structure for $L_{lpl}$:
  - $S_w(c) = \{ \|c\|_w \} \subseteq U_w$ (so that $\|c\|_w \in U_w$)
  - $S_w(P^s_l) \subseteq U^s_w = \{ \langle u_1, \ldots, u_s \rangle \mid u_i \in U_w \}$

where in the present case, $1 \leq s \leq 3$ (and names $c$, like unary predicate symbols $P^1_l$, have singleton extensions).
A referent-assignment $V_w$ is an assignment of a block in $w$ to each variable in $L_{lpl}$ (if not a specific property or relation to each $L_{lpl}$-predicate symbol as well). An assignment of referents to variables (and predicate symbols) extends to all $L_{lpl}$-terms, i.e., $V_w(c) = \|c\|_w \in \{\|c\|_w\} = S_w(c)$.

So, a given blocks world $w$ determines an extensional structure $S_w$ for $L_{lpl}$ where, e.g., ‘Cube(b)’ is true in that world iff $S_w(b) = \{V_w(b)\}$ is a subset of the $S_w$-extension of ‘Cube’ in the respective $L_{setu}$-universe of sets for $w$. 
<table>
<thead>
<tr>
<th>Atomic Sentence-Type</th>
<th>Intended Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(1 \leq l \leq 6)$</td>
<td>$V_w(a) \in S_w(P^1_l)$</td>
</tr>
<tr>
<td></td>
<td>$S_w(a) \subseteq S_w(P^1_l)$</td>
</tr>
<tr>
<td>$(7 \leq l \leq 18)$</td>
<td>$\langle V_w(a), V_w(b) \rangle \in S_w(P^2_l)$</td>
</tr>
<tr>
<td></td>
<td>$S_w(\langle a, b \rangle) \subseteq S_w(P^2_l)$</td>
</tr>
<tr>
<td>$(l = 19)$</td>
<td>$\langle V_w(a), V_w(b), V_w(c) \rangle \in S_w(P^3_l)$</td>
</tr>
<tr>
<td></td>
<td>$S_w(\langle a, b, c \rangle) \subseteq S_w(P^3_l)$</td>
</tr>
<tr>
<td></td>
<td>$\langle V_w(a), V_w(b), V_w(c) \rangle \in S_w(P^3_l)$</td>
</tr>
<tr>
<td></td>
<td>$S_w(\langle a, b, c \rangle) \subseteq S_w(P^3_l)$</td>
</tr>
</tbody>
</table>

A standard recursive definition of satisfaction has as its basis a one-to-one $\mathcal{M}_w$-translation between $L_{lpl}$-atomic formulas and $L_{setu}$-atomic formulas:

$$
\mathcal{M}_w \models P^s_i(\vec{c}) \iff V_w(\vec{c}) = \|\vec{c}\|_w \in S_w(\vec{c}) \subseteq S_w(P^s_i)
$$

$$
\mathcal{M}_w, V_w \models P^s_i(\vec{t}) \iff V_w(\vec{t}) \in S_w(P^s_i)
$$

where a term $t$ is either a variable $x$ or a name $c$ and $\vec{t}$ is an $s$-tuple of terms.

Here, *referents of terms* are single blocks (urelements) having properties or being related in various ways.
The translation between $L_{lpl}$ and $L_0^*$ is designed to accommodate correspondence theory (van Benthem 1984) where any modal sentential language has a characteristic “first-order correspondence language” (treating the necessity operator as a universal quantifier ranging over worlds, etc.)

$L_{lpl}$ does not have an appropriate grammar to be such a first-order correspondence language; but $L_0^*$ does.
The first step in this translation is to treat *block-tuples* as individuals. Each \( L^*_0 \)-predicate symbol is monadic but can have term-tuples of *any* arity as a single argument.

Thus \( L^*_0 \)-predicate symbols will be of the form \( P^s_l|\bar{a} \) for each \( L_{lpl} \)-predicate symbol \( P^s_l \) and for some appropriate \( s \)-tuple \( \bar{a} \).

- \( s \) is the arity (sort) of predicate symbol \( P^s_l|\bar{a} \).
- \( \bar{a} \) is an \( s \)-tuple of integers indicating which tuple of components of a block-tuple \( \bar{u} \) (of arbitrary arity) is to serve as the argument for \( P^s_l|\bar{a} \).

Each \( L_{lpl} \)-predicate symbol \( P^s_l \) corresponds to a potential infinity of \( L^*_0 \)-predicate symbols \( P^s_l|\bar{a} \).
extensional interpretations:

languages to be interpreted:

ability-based interpretations:

extensional interpretations:

$L^*_{setu}$: A language of set theory with ordered block-tuples as urelements

- For $L^*_{setu}$, urelements are ordered tuples of blocks in a given blocks world $w$.
- Though their interpretations are systematically related, predicate symbols are interpreted differently in the two languages:

  A tuple $\vec{u}$ (as an individual block-tuple, of any arity) is in the extension of $L^*_0$-predicate symbol $P_s^i|\vec{a}$ just in case $p_\vec{a}(\vec{u})$ (as a 1-, 2-, or 3-tuple of individuals) is in the extension of the $L_{lpl}$-predicate symbol $P_s^i$. 

Burke (Philosophy USC, USA)  Pragmatism and Dynamic Logic  2013/4/24  17 of 73
This yields the following correspondences among $L^*_0$, $L^*_{setu}$, and $L_{setu}$:

<table>
<thead>
<tr>
<th>$L^*_0$-Atomic Sentence-Types</th>
<th>$L^*_{setu}$-Atomic Sentence-Types</th>
<th>$L^*_{setu}$-Atomic Sentence-Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(1 \leq l \leq 6)$ $P^1_l</td>
<td>_i(\vec{c})$</td>
<td>$S^<em>_w(\vec{c}) \subseteq S^</em>_w(P^1_l</td>
</tr>
<tr>
<td>$(7 \leq l \leq 18)$ $P^2_l</td>
<td>_{i,j}(\vec{c})$</td>
<td>$S^<em>_w(\vec{c}) \subseteq S^</em>_w(P^2_l</td>
</tr>
<tr>
<td>$(l = 19)$ $P^3_l</td>
<td>_{i,j,k}(\vec{c})$</td>
<td>$S^<em>_w(\vec{c}) \subseteq S^</em>_w(P^3_l</td>
</tr>
</tbody>
</table>

Note: $p_{\vec{a}}$ is a “projection” function such that $p_{\vec{a}}(\vec{c})$ is the projection of $\vec{c}$ onto its $\vec{a}$-th components. With a slight but harmless abuse of notation, ‘$p_i(\vec{c})$’ denotes “the $i$-th component of $\vec{c}$”; ‘$p_{i,j}(\vec{c})$’ denotes “the $\langle i, j \rangle$-th pair of components of $\vec{c}$”; etc.
Some details:

A fixed blocks world $w$ determines an $L^*_0$-model $\mathcal{M}^*_w$ consisting of a pair $\langle U^*_w, S^*_w \rangle$ where $U^*_w$ is the countably infinite domain of finite $n$-tuples $\vec{u}$ of blocks in $w$ and $S^*_w$ is a unique structure for the first-order block-tuples language $L^*_0$:

- $S^*_w(\mathcal{P}^*_i|\vec{a}) = \{ \vec{u} \mid \mathcal{P}_a(\vec{u}) \in S_w(\mathcal{P}^*_i) \} \subseteq U^*_w$.
- $S^*_w(\langle \mathcal{C} \rangle) = \{ \| \langle \mathcal{C} \rangle \|_w^* \} = \{ \| \mathcal{C} \|_w \} \subseteq U_w \subseteq U^*_w$. 

-, -
A referent-assignment $V^*_w$ assigns an $n$-tuple $\vec{u}$ in $\mathcal{U}^*_w$ to each $n$-tuple term $\vec{t}$ in $L^*_0$ where

1. $V^*_w(\langle c \rangle) = S^*_w(\langle c \rangle) = S_w(c)$.
2. $V^*_w(\langle x \rangle) \in \mathcal{U}_w \subseteq \mathcal{U}^*_w$.
3. $V^*_w(\vec{t}) = \vec{u} \in \mathcal{U}^*_w$ such that $p_i(\vec{u}) = V^*_w(\langle p_i(\vec{t}) \rangle)$.

Here, *referents of terms* are block-tuples (urelements) having properties, period. All $L^*_0$-predicate symbols are unary.
A recursive definition of satisfaction has as its basis a one-to-one $S^*_w$-translation between $L^*_0$-atomic sentence-types and $L^*_{setu}$-atomic sentence-types:

$$M^*_w \models P^s_{P_1}(\vec{a}(\vec{c})) \iff S^*_w(\vec{c}) \subseteq S^*_w(P^s_{P_1}(\vec{a}))$$

$$M^*_w, V^*_w \models P^s_{P_1}(\vec{a}(\vec{t})) \iff V^*_w(\vec{t}) \in S^*_w(P^s_{P_1}(\vec{a}))$$
extensional interpretations: 

languages to be interpreted: 

ability-based interpretations: 

extensional interpretations: 

\[
\langle V_w(t'_1), \ldots, V_w(t'_s) \rangle \in S_w(P^s_i) \iff V^*_w(\langle t_1, \ldots, t_n \rangle) \in S^*_w(P^s_i|\vec{a})
\]

\[
\mathcal{M}_w, V_w \models P^s_i(t'_1, \ldots, t'_s) \iff \mathcal{M}^*_w, V^*_w \models P^s_i|\vec{a}(\langle t_1, \ldots, t_n \rangle)
\]

Let \( p_{\vec{a}}(\vec{t}) = p_{\vec{a}}(\langle t_1, \ldots, t_n \rangle) = \langle t'_1, \ldots, t'_s \rangle = \vec{t}' \). Namely, “\( p_{\vec{a}}(\vec{u}) \)” denotes the projection of \( \vec{u} \) onto components designated by the 1-, 2-, or 3-tuple \( \vec{a} \in \mathbb{N} \cup \mathbb{N}^2 \cup \mathbb{N}^3 \).

Then, \( V^*_w(\langle t_1, \ldots, t_n \rangle) = V^*_w(\vec{t}) = \vec{u} \) implies that \( \langle V_w(t'_1), \ldots, V_w(t'_s) \rangle = V_w(\vec{t}') = p_{\vec{a}}(\vec{u}) \).

Thus, more succinctly, …
extensional interpretations:

$$L_{setu} \overset{1'}{\longleftrightarrow} L_{setu}^* \overset{2'}{\longleftrightarrow} L_{setu}^\circ$$

languages to be interpreted:

$$L_{lpl} \overset{1}{\longleftrightarrow} L_0^* \overset{2}{\longleftrightarrow} L_{setu} \overset{3}{\longleftrightarrow} L_0$$

ability-based interpretations:

$$L_{abd} \overset{3'}{\longleftrightarrow} L_{abf}$$

extensional interpretations:

$$L_{lpl}^* \overset{3''}{\longleftrightarrow} L_{setu}^* \overset{1'}{\longleftrightarrow} L_{setu} \overset{3}{\longleftrightarrow} L_{lpl}$$

$$p_{\vec{a}}(\vec{u}) \in S_w(P_l^s) \overset{\mathfrak{m}_w}{\longleftrightarrow} \vec{u} \in S_w^*(P_l^s|_{\vec{a}})$$

$$\mathcal{M}_w, V_w^{\vec{t}/\vec{u}} \models P_l^s(p_{\vec{a}}(\vec{t})) \overset{\mathfrak{m}_w^*}{\longleftrightarrow} \mathcal{M}_w^*, V_w^*^{\vec{t}/\vec{u}} \models P_l^s|_{\vec{a}}(\vec{t})$$
extensional interpretations: $L_{setu} \Leftrightarrow L^*_{setu} \Leftrightarrow L^\circ_{setu}$

languages to be interpreted: $L_{dpl} \Leftrightarrow L^*_0 \Leftrightarrow L^\circ_0 \Leftrightarrow L_0$

ability-based interpretations: $L_{abd} \Leftrightarrow L^\circ_{abd} \Leftrightarrow L^\circ_{abf}$

extensional interpretations: $L^\circ_{setu} \Leftrightarrow L^\circ_{setu}$

$L^\circ_0$ : A modal sentential reverse-correspondence language for the monadic first-order language $L^*_0$

$L^*_0$ is supposed to be the first-order correspondence language for the modal sentential language $L^\circ_0$ —
  - the first-order universal quantifier in $L^*_0$
    (ranging over a domain of “block-tuples”) is translated as
  - a necessity operator in $L^\circ_0$
    (the same domain regarded as a domain of “worlds”).
This reverses the usual translation \textit{from} a given modal sentential language \textit{to} its first-order correspondence language. The backwards move here consist in

1. treating \( n \)-tuples of blocks (for \( n \in \mathbb{N} \)) as \textit{worlds}—worlds of “block-features”—e.g., worlds in which certain entities like faces, vertices, etc., have properties and/or are related in various ways;

2. translating the first-order \( L_0^* \)-universal quantifier as a modal sentential \( L_0^* \)-necessity operator;

3. designating a unique atomic \( L_0^* \)-sentence-type \( p_i^s|\vec{a} \) for each \( L_0^* \)-predicate symbol \( P_i^s|\vec{a} \) (and thus a unique atomic \( L_0^* \)-formula-type \( p_i^s \) for each \( L_0^{lp} \)-predicate symbol \( P_i^s \)) such that

\[
M_w^*, V_w^* \iota/\vec{u} \models P_i^s|\vec{a}(\vec{t}) \text{ iff } M_w^*, \vec{u} \models p_i^s|\vec{a}
\]

That is, . . .
... first, the following should be distinguished:

formula-types \( p^s_l \) for fixed \( s, l; \ 1 \leq s \leq 3 \) and \( 1 \leq l \leq 25 \)

formulas \( p^s_l |_{i} \) where \( i \) is an \( s \)-tuple of index-variables \( i_1, i_2, \ldots \)

sentence-types \( p^s_i |_{a} \) where \( a \) is an \( s \)-tuple of positive integers

sentences \( M^w, \vec{u} | = p^s_i |_{a} \) for an \( M^w \)-world \( \vec{u} \) where \( p^a(\vec{u}) \) is determined

Worlds \( \vec{u} \) are ordered \( n \)-tuples of blocks in some given blocks world \( w \).
extensional interpretations: $L_{setu} \iff L^*_{setu} \iff L^\circ_{setu}$

languages to be interpreted: $L_{lpl} \iff L^*_0 \iff L^\circ_0 \iff L_0$

ability-based interpretations: $L_{abd} \iff L_{abf}$

extensional interpretations: $L^\circ_{setu} \iff L^*_\circ_{setu}$

$L^\circ_0$-sentence-types must be indexical in that they refer to a world ($here_{p\vec{a}(\vec{u})}$, $this_{p\vec{a}(\vec{u})}$, $that_{p\vec{a}(\vec{u})}$, . . .) relative to which their truth-value is evaluated.

The truth-value of an $L^\circ_0$-formula $p^\vec{x}|_{\vec{a}}$ can be evaluated only with reference to some $\mathcal{M}_w^\circ$-world $\vec{u}$—just as the corresponding $L^*_0$-formula $P(\vec{x})$ has no truth-value except with respect to an $L^*_0$-structure $S_w^*$ and a respective $S_w^*$-valuation $V_w^*$ assigning some object to $\vec{x}$. 
The correspondence “$\leftrightarrow$”:

A translation between the first-order language $L_0^*$ and the modal sentential language $L_0^\circ$ may be materially substantive or not. Standard correspondence theory does not decide this option one way or the other.

$$L_0^* \quad \leftrightarrow \quad L_0^\circ$$

Trivial: $\text{Cube}(\langle x \rangle) \mapsto \text{“This}_x \text{ is a cube.”}$

Substantive: $\text{Cube}(\langle x \rangle) \mapsto \text{“Each face here}_x \text{ has four sides.”}$

$\text{“Each face here}_x \text{ has four vertices.”}$

$\text{“Each face here}_x \text{ is a square.”}$

$\text{“There are exactly six faces here}_x \text{.”}$

$\text{“There are exactly eight vertices here}_x \text{.”}$

etc.
An important point:

- The substantive correspondence is easily “operationalized” using any one or more abilities like the following:
  - ABILITY: count the number of edges of any one face; OUTCOMES: three, four, or five.
  - ABILITY: count the number of vertices of any one face; OUTCOMES: three, four, or five.
  - ABILITY: note the shape of any one face; OUTCOMES: triangle, square, or pentagon.
  - ABILITY: count the number of faces; OUTCOMES: four, six, or more than six.
  - ABILITY: count the number of vertices; OUTCOMES: four, eight, or more than eight.
Another important point:

- The trivial correspondence is also easily operationalized.
  - ABILITY: note the shape of the block;
  - OUTCOMES: tetrahedron, cube, or dodecahedron.

(If one can note shapes of faces, why not shapes of whole blocks?)

- But the materially substantive correspondence includes descriptive if not explanatory details that the trivial correspondence lacks.
An $L^*_0$-model $\mathcal{M}_w$ is a triple including

- a set $\mathcal{W}_w^\circ$ of block-tuple worlds,
- a universal accessibility relation $\mathcal{R}_w^\circ$, and
- a valuation function $\mathcal{V}_w^\circ$ assigning some subset of $\mathcal{W}_w^\circ$ to each atomic sentence-type.

This is expressible in a language of set theory, $L^\circ_{setu}$, with elements of $\mathcal{W}_w^\circ$ as urelements.
extensional interpretations:

languages to be interpreted:

ability-based interpretations:

extensional interpretations:

Such an $L_0^0$-model corresponds exactly to an $L_0^*$-structure (with a set of block-tuples as a domain of discourse, and an extensional interpretation for each $L_0^*$-predicate symbol):

- $\vec{u} \in S_w^*(P_i|\vec{a})$ if and only if $\vec{u} \in V_w^0(p_i|\vec{a})$,
- $\mathcal{M}_w^*, V_w^*\vec{t}/\vec{u} \models P_i|\vec{a}(\vec{t})$ if and only if $\mathcal{M}_w^0, \vec{u} \models p_i|\vec{a}$.
In general, the set of total truth-assignments for an uninterpreted sentential language \( L \) yields a frame for a respective modal sentential language \( L^\circ \) by treating some nonempty subset of these truth-assignments as a set of worlds, etc. Partial truth-assignments determine sets of such worlds.

In the present case, similarly, partial truth-assignments for \( L_0 \) (an interpreted sentential language) correspond to sets of block-tuple worlds in \( \mathcal{W}_w^\circ \).
For example, an atomic sentence-type $p_{2|1}$ might state that “all of the six faces here $p_{1}(\theta)$ have four sides of equal lengths that meet only at right angles”—more than enough (relative to any given blocks world $w$) to establish that “this $p_{1}(\theta)$ world” is or is not a cube world (i.e., a cubical block).

A coherent truth assignment $\mathcal{T}_{w}$ that assigns TRUE to this sentence-type corresponds to a class of worlds $\vec{u}$ whose first components $p_{1}(\vec{u})$ are all cubical in shape (regardless of any differences in size, location, etc.).
Another atomic sentence-type $p_{1|1}$ might state that “all of the four faces here $p_{1}(\theta)$ have three sides of equal lengths that meet only at sixty-degree angles.”

A truth assignment that assigns TRUE to both $p_{1|1}$ and $p_{2|1}$ is analytically incoherent and thus determines the empty set of worlds. (A one-tuple world cannot be both tetrahedral and cubical. There are no square triangles or triangular squares. Three is not equal to four. Etc.) Such a truth assignment fails to respect the incompatibility of the characters of the two sentence-types.
Likewise, a third atomic sentence-type $p_{3|1}$ might state that “all of the twelve faces here $p_1(\theta)$ have five sides of equal lengths that meet only at seventy-two-degree angles.”

A truth assignment that assigns FALSE to $p_{1|1}$, $p_{2|1}$, and $p_{3|1}$ is materially incoherent and thus determines the empty set of worlds. (A one-tuple world must be tetrahedral, cubical, or dodecahedral, given the specific nature common to all blocks worlds $w$.) Such a truth assignment conflicts with contingent but universal features of blocks worlds in general.
Thus, a truth-assignment to the atomic sentence-types in a given $L_0$-sentence-type will determine the empty set of possible block-tuple worlds for either of two reasons:

- because of incompatible characters of the various atomic sentence-types (analytic incoherence), or
- because of conflicts with contingent features of blocks worlds in general (material incoherence).

We can on such grounds distinguish truth-functional, analytic, and material truths among $L_0$-sentence-types.
An $L_0$-sentence-type is

- a **truth-functional truth** if it is true in all truth-assignments to its constituent atomic sentence-types (e.g., $p_1 \mid_1 \lor \neg p_1 \mid_1$);

- an **analytical truth** if it is true in all such truth-assignments excluding any that determine the empty set of worlds due to incompatible “characters” of its constituent sentence-types (e.g., $p_1 \mid_1 \rightarrow \neg p_2 \mid_1$); and

- a **material truth** if it is true in all of the latter truth-assignments excluding further any that are empty due to conflicts with “natural” features of blocks worlds in general (e.g., $p_1 \mid_1 \lor p_2 \mid_1 \lor p_3 \mid_1$).
Let $\mathcal{X}_w^\circ(\vec{u})$ be the set of all atomic $L_0^\circ$-sentence-types that are $\mathcal{M}_w^\circ$-true in $\vec{u}$. This set corresponds to a truth-assignment $\mathcal{T}_w^\circ\vec{u}$ where

\[
\mathcal{T}_w^\circ\vec{u}(p^s_{l}|\vec{a}) = \text{TRUE} \iff p^s_{l}|\vec{a} \in \mathcal{X}_w^\circ(\vec{u}) \iff \vec{u} \in \mathcal{V}_w^\circ(p^s_{l}|\vec{a}).
\]

Then, we may define

\[
\mathcal{V}_w^\circ(p^s_{l}|\vec{a}) = \{ \vec{u} \mid \mathcal{T}_w^\circ\vec{u}(p^s_{l}|\vec{a}) = \text{TRUE} \}
\]

as the $\mathcal{M}_w^\circ$-character of $p^s_{l}|\vec{a}$.

\[
\mathcal{X}_w^\circ(\vec{u}) = \{ p^s_{l}|\vec{a} \mid \mathcal{T}_w^\circ\vec{u}(p^s_{l}|\vec{a}) = \text{TRUE} \}
\]

as the $\mathcal{M}_w^\circ$-characterization of $\vec{u}$. 
extensional interpretations:

languages to be interpreted:

ability-based interpretations:

extensional interpretations:

$L_{abf}$: An ability-based first-order language

- This, finally, is where the pragmatic maxim comes into play.
- Claim: All $L_0$-formulas can be operationalized in terms of abilities and “sensible effects” (e.g., measuring abilities and possible outcomes that may result from using such abilities).
- $L_{abf}$ is a sorted first-order language with symbols for basic abilities and their possible outcomes as primitive elements (rather than predicate symbols and names as in a standard first-order language).
- $L_{abf}$ is designed specifically to express the operational characters of atomic $L_0$-formulas.
For instance, a single ability consisting of applying the operation $f$ (e.g., counting the sides of a face) in a world “$p_i(\theta)$” with three possible outcomes (3sides, 4sides, or 5sides) is enough to operationalize all three atomic shape formulas:

<table>
<thead>
<tr>
<th>$L_0^*$</th>
<th>$L_0^*$, $L_0$</th>
<th>$L(abl)$, $L(abd)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{Tet}_1</td>
<td>_{i}(\vec{x})$</td>
<td>“Each face here $p_i(\theta)$ has three sides.”</td>
</tr>
<tr>
<td>$\text{Cube}_2</td>
<td>_{i}(\vec{x})$</td>
<td>“Each face here $p_i(\theta)$ has four sides.”</td>
</tr>
<tr>
<td>$\text{Dodec}_3</td>
<td>_{i}(\vec{x})$</td>
<td>“Each face here $p_i(\theta)$ has five sides.”</td>
</tr>
</tbody>
</table>
The expression ‘$p_i(\theta)$’ denotes “the $i$-th component here.”

In $L_0$ and $L_0$, such formulas will be used in a given world—the standing referent of ‘$\theta$’—relative to a given model for the language.

In $L_{abl}$ and $L_{abd}$, ‘$f(p_i(\theta))$’ denotes an ability consisting of ‘$\theta$’ (referencing “this world”), ‘$p_i(\theta)$’ (locating the $i$-th component of “this world”), and applying operation $f$ to that component (counting the sides of its faces).
<table>
<thead>
<tr>
<th>$L^*$</th>
<th>$L^+_0$, $L_0$</th>
<th>$L_{abl}$, $L_{abd}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tet$^1_i(x)$</td>
<td>“Each face here$_{p_i(\theta)}$ has three sides.”</td>
<td>$\text{f}(p_i(\theta)) = 3\text{sides}$</td>
</tr>
<tr>
<td>Cube$^2_i(x)$</td>
<td>“Each face here$_{p_i(\theta)}$ has four sides.”</td>
<td>$\text{f}(p_i(\theta)) = 4\text{sides}$</td>
</tr>
<tr>
<td>Dodec$^3_i(x)$</td>
<td>“Each face here$_{p_i(\theta)}$ has five sides.”</td>
<td>$\text{f}(p_i(\theta)) = 5\text{sides}$</td>
</tr>
</tbody>
</table>

Similar operations $\text{f}$ can be uniquely specified for each of the nineteen predicate symbols and six names in $L^*_0$ (see below).

The notation ‘$v$’ denotes a variable associated with ability $v$, ranging over the set of possible outcomes resulting from the use of that ability.

The notation ‘$v|_u$’ denotes a variable with the constraint that the value of constituent ability $u$ is $u$. 
The language $L_{abf}$ will include means for showing how complex abilities and complex outcomes may be constructed.

*Variables* will be defined recursively in terms of such abilities, as just noted.

A sorted identity symbol ‘$\equiv$’ may be the sole *predicate symbol*—applicable to $L_{abf}$-*terms* (variables, outcome-types, etc.).

First-order *formulas* are defined recursively in a standard way.
Outcomes $o_i^s$ are like measurement results—e.g., numerical values with units ("the result of measuring the width here $p_i(\theta)$ is four meters")—though outcomes need not be numerical.

There are many ways to operationalize the intended meanings of the atomic sentence-types in $L_0$, and thus the predicate symbols of $L_{lpl}$. The following is just one example.
Here, $f_1^1 = f_2^1 = f_3^1$. This is an operation of counting edges of faces, with outcome set $\{o_1^1, o_2^1, o_3^1\} = \{3\text{ sides}, 4\text{ sides}, 5\text{ sides}\}$. This is enough to distinguish the three possible shapes.

Also, $f_4^1 = f_5^1 = f_6^1$. This is an operation of gauging heights of blocks, with outcome set $\{o_4^1, o_5^1, o_6^1\} = \{<1\text{ sq, } =1\text{ sq, } >1\text{ sq}\}$—enough to distinguish the three possible sizes.

And $f_7^2 = f_8^2 = f_{13}^2$. This is a composite operation of, first, gauging heights of each of two components, again with respective outcome sets $\{<1\text{ sq, } =1\text{ sq, } >1\text{ sq}\}$, and second, identifying which (if either) is least, with outcome set $\{\text{ht1, ht2, } -\}$—indicating that the first is shorter than the second, that the second is shorter than the first, or that the two are the same height. This is enough to determine whether one component is larger, smaller, or the same size as the other.
<table>
<thead>
<tr>
<th>(L_0)-Atomic Formulas</th>
<th>(L_{abf})-Atomic Formulas</th>
</tr>
</thead>
<tbody>
<tr>
<td>((1 \leq l \leq 6))</td>
<td>(p^1_{l}</td>
</tr>
<tr>
<td>((7 \leq l \leq 18))</td>
<td>(p^2_{l}</td>
</tr>
<tr>
<td>((l = 19))</td>
<td>(p^3_{l}</td>
</tr>
</tbody>
</table>

The operation \(f^2_{9} = f^2_{10} = f^2_{16}\) is a composite operation of, first, determing the row and column numbers of each of two components, with respective intermediate outcome sets \(\langle c_1, r_1 \rangle\) and \(\langle c_2, r_2 \rangle\), and second, identifying the least (if either) of the two column numbers, with outcome set \(\{1st\text{col}, 2nd\text{col}, =\text{col}\}\)—a set sufficient to tell whether or not one of the components is left of, right of, or in the same column as the other.

The operation \(f^2_{11} = f^2_{12} = f^2_{15}\) is a composite operation of, first, determing the row and column numbers of each of two components, with respective intermediate outcome sets \(\langle c_1, r_1 \rangle\) and \(\langle c_2, r_2 \rangle\), and second, identifying the least (if either) of the two row numbers, with outcome set \(\{1st\text{row}, 2nd\text{row}, =\text{row}\}\)—a set sufficient to tell whether or not one of the components is in back of, in front of, or in the same row as the other.
\begin{align*}
\begin{array}{ccc}
\text{\textbf{\begin{tabular}{c}
\text{}\textbf{L}_0-\text{Atomic} \\
\text{Formulas}
\end{tabular}}} & \text{\textbf{\begin{tabular}{c}
\text{}\textbf{L}_{abf}-\text{Atomic} \\
\text{Formulas}
\end{tabular}}} \\
(1 \leq l \leq 6) & p^1_{i} & \mathcal{f}_{i}^1(p_i(\theta)) = o^1_l \\
(7 \leq l \leq 18) & p^2_{i,j} & \mathcal{f}_{i,j}^2(p_i,j(\theta)) = o^2_l \\
(l = 19) & p^3_{i,j,k} & \mathcal{f}_{i,j,k}^3(p_i,j,k(\theta)) = o^3_l
\end{array}
\end{align*}

Operation $\mathcal{f}_{18}^2$ like those above, is a composite operation of, first, determining the row and column numbers of each of two components, with respective intermediate outcome sets \{\langle c_1, r_1 \rangle \} and \{\langle c_2, r_2 \rangle \}, and second, comparing these to determine whether the two results are equal or not, with outcome set \{\text{eq-loc, neq-loc}\}—serving to distinguish whether the two components are identical (having the same location on the grid) or not.

Operation $\mathcal{f}_{14}^2$ is a composite operation of, first, counting the number of edges of the faces of each of two components, with respective outcome sets \{o^1_1, o^1_2, o^1_3 \} = \{\text{3sides, 4sides, 5sides}\}, and, second, gauging whether the two results are equal or not, with outcome set \{\text{eq-sides, neq-sides}\}—serving as evidence that the first and second have the same shape (the same number of sides per face) or different shapes.
The operation $f^2_{17}$ is a simple operation of determining the distance between (the centers of) two components, with outcome set \{<1sq, =1sq, >1sq\}, such that the two components adjoin one another only in the second of these three cases.

Operation $f^1_{20} = f^1_{21} = f^1_{22} = f^1_{23} = f^1_{24} = f^1_{25}$ is a simple operation of reading off the name on a component’s name tag if the component has a name tag. The outcome set is \{a, b, c, d, e, f\}. 

<table>
<thead>
<tr>
<th>$L_0$-Atomic Formulas</th>
<th>$L_{abf}$-Atomic Formulas</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(1 \leq l \leq 6)$</td>
<td>$p^1_l</td>
</tr>
<tr>
<td>$(7 \leq l \leq 18)$</td>
<td>$p^2_l</td>
</tr>
<tr>
<td>$(l = 19)$</td>
<td>$p^3_l</td>
</tr>
</tbody>
</table>
Finally, operation $\mathbf{f}_{19}^3$ is a composite operation of, first, determining the row and column numbers of each of three components, with respective outcome sets $\{\langle c_1, r_1 \rangle, \langle c_2, r_2 \rangle, \langle c_3, r_3 \rangle\}$. Second, respective row- and column-triples of these results are compared such that the process halts with no outcome unless (a) all three row numbers are equal and no two column numbers are equal, or (b) all three column numbers are equal and no two row numbers are equal, or (c) no two column numbers are equal, no two row numbers are equal, the difference between the first and second row numbers equals the difference between the first and second column numbers, and the difference between the first and third row numbers equals the difference between the first and third column numbers. Third, if one of (a)–(c) holds, the distances between the three pairs of components are gauged and the greatest of the three is identified, with outcome set $\{1\text{st}2\text{nd}, 1\text{st}3\text{rd}, 2\text{nd}3\text{rd}\}$.
<table>
<thead>
<tr>
<th>$L_0$-Atomic Formulas</th>
<th>$L_{abf}$-Atomic Formulas</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(1 \leq l \leq 6)$</td>
<td>$p^1_l</td>
</tr>
<tr>
<td>$(7 \leq l \leq 18)$</td>
<td>$p^2_l</td>
</tr>
<tr>
<td>$(l = 19)$</td>
<td>$p^3_l</td>
</tr>
</tbody>
</table>

$L^*_0 : p^1_{11} (\bar{x}) [\bar{x} \leftarrow \theta]$  
This $p_1(\theta)$ is a cube.

$L_0 : p^2_{11}$  
Each face here $p_1(\theta)$ has four sides.

$L_{abf} : \mathbf{f}^1_{11} (p^1) = 4\text{ sides}$  
The result of counting the sides of faces here $p_1(\theta)$ is four.

$L^*_0 : p^2_{11,2} (\bar{x}) [\bar{x} \leftarrow \theta]$  
This $p_1(\theta)$ is in back of that $p_2(\theta)$.

$L_0 : p^2_{11} | 1,2$  
The row number of this $p_1(\theta)$ is greater than the row number of that $p_2(\theta)$.

$L_{abf} : \mathbf{f}^2_{11} (p^1,2) = 2\text{nd row}$  
The result of ordering the rows of this $p_1(\theta)$ and that $p_2(\theta)$ in non-decreasing order is that the latter is less than the former.

$L^*_0 : p^3_{11,2,3} (\bar{x}) [\bar{x} \leftarrow \theta]$  
This $p_1(\theta)$ is between that $p_2(\theta)$ and the other $p_3(\theta)$.

$L_0 : p^3_{11} | 1,2,3$  
This $p_1(\theta)$, that $p_2(\theta)$, and the other $p_3(\theta)$ are appropriately collinear and this $p_1(\theta)$ is closer to that $p_2(\theta)$ and the other $p_3(\theta)$ than they are to each other.

$L_{abf} : \mathbf{f}^3_{19} (p^1,2,3(\theta)) = 2\text{nd 3rd}$  
The result of gauging distances among this $p_1(\theta)$, that $p_2(\theta)$, and the other $p_3(\theta)$ (having determined first that the three are appropriately collinear) is that that $p_2(\theta)$ and the other $p_3(\theta)$ are the greatest distance apart.
$L_{setu}^*$ : A language of set theory with possible outcomes for given abilities as urelements

$L_{abf}$-models $\mathcal{M}_w^* = \langle \mathcal{U}_w^*, \mathcal{S}_w^* \rangle$ are definable in a language $L_{setu}^*$. In steps:

- A basis for an $L_{abf}$-grammar includes, first, a set of basic operations $\mathbf{f}$, illustrated earlier.

Abilities $\mathbf{v}$ with respective outcome sets $Outc(\mathbf{v})$ are constructible, as illustrated earlier.

Atomic $L_{abf}$-formulas are of the form `$\mathbf{v} = \mathbf{o}$' where $\mathbf{o} \in Outc(\mathbf{v})$. 
A valuation $V^*_w$ is a partial function defined on the set of abilities such that $V^*_w(v) \in Outc(v)$. Valuations extend as usual to variables and terms in $L_{abf}$. In particular, $V^*_w(\$v) = V^*_w(v)$; and $V^*_w(o) = o$.

A structure $S^*_w$ assigns outcomes (as possibilities) to themselves, and interprets the identity-predicate `\(\equiv\)` as expected, namely,

$$S^*_w(\equiv) = \{\langle o, o \rangle \mid o \in \bigcup Outc(v)\}.$$

Then,

$$S^*_w, V^*_w \models t_1 \equiv t_2 \text{ iff } \langle V^*_w(t_1), V^*_w(t_2) \rangle \in S^*_w(\equiv) \text{ iff } V^*_w(t_1) = V^*_w(t_2).$$

$$S^*_w, V^*_w \models \beta_1 \rightarrow \beta_2 \text{ iff } S^*_w, V^*_w \not\models \beta_1 \text{ or } S^*_w, V^*_w \models \beta_2$$

$$S^*_w, V^*_w \models \beta \rightarrow \bot \text{ iff } S^*_w, V^*_w \not\models \beta.$$
extensional interpretations:

languages to be interpreted:

ability-based interpretations:

extensional interpretations:

$L_{abf}$-models $\mathcal{M}_w^* = \langle \mathcal{U}_w^*, S_w^* \rangle$ are defined in a language $L_{setu}^*$, as already indicated.

- Let $\mathcal{U}_w^*$ be $\bigcup_v \text{Outc}(v)$.
- Let $S_w^*$ be the simple structure defined above.

Also, let $V_w^*$ be a valuation as defined above (such that $V_w^*(\$v) = V_w^*(v) \in \text{Outc}(v)$). Then,

- $\mathcal{M}_w^*, V_w^* \models \alpha$ iff $S_w^*, V_w^* \models \alpha$
- $\mathcal{M}_w^*, V_w^* \models \forall \$v \delta$ iff $\mathcal{M}_w^*, V_w^*\$v/o \models \delta$
  for any $o \in \text{Outc}(v)$
- Etc.
Note the following:

- There is a distinct atomic $L_{abf}$-sentence-type $\bar{\alpha} = \$ f_l^\delta (p_\bar{a}(\theta)) = o_l^\delta$ for each atomic $L_0^\circ$-sentence-type $\alpha = p_l^\delta | \bar{a} \in X^\circ_w(\vec{u})$.
- An $L_{abf}$-valuation $V_w^*$ determines a truth-assignment $\mathcal{T}_w^*$ for atomic $L_{abf}$-formulas where $\mathcal{T}_w^*(\bar{\alpha}) = \text{TRUE}$ iff $S_w^*, V_w^* \models \bar{\alpha}$.
- Let $V_{w \vec{u}}^*$ be the valuation whose respective truth-assignment $\mathcal{T}_{w \vec{u}}^*$ assigns TRUE to each of the atomic $L_{abf}$-sentence-types $\bar{\alpha}$ corresponding to respective $L_0^\circ$-sentence-types $\alpha \in X^\circ_w(\vec{u})$. 
Then,

- \( \mathcal{T}_w^{\ast \bar{u}} (\bar{\alpha}) = \text{TRUE} \) iff \( \mathcal{T}_w^{\circ \bar{u}} (\alpha) = \text{TRUE} \).

So let \( \mathcal{X}_w^{\ast}(V_w^{\ast \bar{u}}) = \{ \bar{\alpha} \mid \mathcal{T}_w^{\ast \bar{u}} (\bar{\alpha}) = \text{TRUE} \} \)
= the \( \mathcal{M}_w^{\ast} \)-characterization of \( \bar{u} \).

And let \( \mathcal{Y}_w^{\ast}(\bar{\alpha}) = \{ V_w^{\ast \bar{u}} \mid \mathcal{T}_w^{\ast \bar{u}} (\bar{\alpha}) = \text{TRUE} \} \)
= the \( \mathcal{M}_w^{\ast} \)-character of \( \bar{\alpha} \).

Then,

- \( \bar{\alpha} \in \mathcal{X}_w^{\ast}(V_w^{\ast \bar{u}}) \) iff \( \alpha \in \mathcal{X}_w^{\circ}(\bar{u}) \)
- \( V_w^{\ast \bar{u}} \in \mathcal{Y}_w^{\ast}(\bar{\alpha}) \) iff \( \bar{u} \in \mathcal{Y}_w^{\circ}(\alpha) \)
From another direction,

- The set $\mathcal{T}_0^\delta$ of partial materially-coherent truth-assignments $\mathcal{T}_0^\delta$ restricted to the constituent atomic sentence-types $\alpha_0^\delta$ in the $L_0^\delta$-sentence-type $\delta$, such that $\mathcal{T}_0^\delta(\delta) = \text{TRUE}$, will determine a set $\Upsilon^w(\delta)$ of $L_{abf}$-valuations.

- On the other hand, $\mathcal{T}_0^\delta$ determines a set of block-tuple worlds $\vec{u}$ such that $\delta$ accurately describes features of $\vec{u}$—namely, all $\vec{u}$ such that $\mathcal{M}_0^\delta, \vec{u} \models \delta$. This description is operationalized in the form of the corresponding set of $L_{abf}$-valuations $V_{w}^{*\vec{u}} \in \Upsilon^w(\delta)$. 
For example:

- Let $\delta_1$ be the $L_0^\cdot$-sentence-type
  \[ p_1^1 |_1 \rightarrow \neg(p_2^1 |_1 \vee p_3^1 |_1). \]
  This is a material necessity—true in any block-tuple world $\vec{u}$ since
  the first component of any such world must have exactly one of the three possible shapes.
  Thus $\mathcal{V}_0^\cdot(\delta_1) = \mathcal{V}_0^\cdot(\top)$. That is, $\mathcal{X}_0^\delta_1$ determines the
  whole set $\mathcal{W}_0^\cdot$ of $\mathcal{M}_0^\cdot$-worlds $\vec{u}$—which corresponds
  one-to-one to the whole set of $L_{abf}$-valuations
  $\gamma_{\ast}^w = \gamma_{\ast}^w(\top)$.

- Or let $\delta_2$ be the $L_0^\cdot$-sentence-type
  \[ p_2^1 |_1 \wedge p_2^5 |_1. \]
  This will be true in any world $\vec{u}$ whose first
  component is a medium cube. This may be the empty
  set in $\mathcal{W}_w^\cdot$ or not (depending on $w$), but it corresponds
  in either case to a set $\mathcal{V}_w^\cdot(\delta_2) \subseteq \mathcal{W}_w^\cdot$—which
  corresponds then to a respective set $\gamma_{\ast}^w(\delta_2) \subseteq \gamma_{\ast}^w$. 
In particular,

- If the $L^\circ_0$-sentence-type $\delta$ is itself atomic and thus corresponds to a single $L^*_0$-predicate symbol $P$, then the set of materially-coherent truth assignments $T^\delta_0$ that assign TRUE to $\delta$ serves as a substantive $L^\circ_0$-interpretation of any block-tuple in the intended extension (in $w$) of the $L^*_0$-predicate symbol $P$.

- In that case, the set $\Upsilon^*_w(\bar{\delta})$ of $L_{abf}$-valuations corresponding to the various truth-assignments $T^\delta_0 \in \Xi^\delta_0$ determines a $w$-specific $L_{abf}$-model $M^*_w\bar{\delta}$ where $\Upsilon^*_w = \Upsilon^*_w(\bar{\delta})$. This model serves as a $w$-specific operational $L_{abf}$-interpretation of the $L^*_0$-predicate symbol $P$. 
extensional interpretations:

languages to be interpreted:

ability-based interpretations:

extensional interpretations:
For a given atomic $L_{abf}$-sentence-type $\alpha$ and corresponding $L_0^*$-predicate symbol $P_\alpha$, the collection of all $w$-specific $L_{abf}$-models $\mathcal{M}_{w}^{*\alpha}$ provides a general (though possibly not exhaustive) operational $L_{abf}$-interpretation for $P_\alpha$. Generally speaking, the intended meaning of the predicate symbol $P_\alpha$ could be specified by a class (if not this full class) of such $L_{abf}$-models.

Echoing Peirce, such an $L_{abf}$-model is not so much a “definition” as it is a “precept” concerning which outcomes should result from the exercise of certain abilities (1903, EP2:286). This is our main target— to show how to interpret first-order predicate symbols in line with an operationalist reading of the pragmatic maxim.

But we are not done . . .
**extensional interpretations:**

- $L_{setu}$
- $L_{setu}^{\top}$
- $L_{setu}^{\bot}$

**languages to be interpreted:**

- $L_{lpl}$
- $L_0$
- $L_0^{\top}$

**ability-based interpretations:**

- $L_{abd}$
- $L_{setu}$
- $L_{setu}^{\top}$

---

$L_{abd}$: A language for a multi-modal ability-based first-order dynamic logic.

- $L_{abf}$ is a first-order fragment of a first-order dynamic language $L_{abd}$.
- $L_{abd}$-programs $\pi$ and $L_{abd}$-formulas $\varphi$ are mutually-recursively defined in much the same way as is done for a standard language of first-order dynamic logic.
- *But* ... atomic programs will include *productions* of outcomes using abilities associated with respective variables ($v \Rightarrow o$)—as well as *assignments* of outcomes to ability-based variables ($v \Leftarrow o$).
Atomic programs may be compounded in various ways: by sequential composition, choice, finite-iteration, test, array-formation, recursion, concurrence, etc., to generate a full set of standard programming capabilities.

Test programs are defined for some set of formulas (perhaps only Boolean formulas).

Conversely, a modal operator is associated with each program, yielding modal formulas in a language for a multi-modal logic.
extensional interpretations:

languages to be interpreted:

ability-based interpretations:

extensional interpretations:

$L_{setu}^\bullet$ : A language of set theory again with possible outcomes of given abilities as urelements

$L_{abd}$-models $\mathcal{M}_w^\bullet$ are four-tuples $⟨\mathcal{W}_w^\bullet, [\cdot]_w^\bullet, S_w^\bullet, V_w^\bullet⟩$. For instance,

- $\mathcal{W}_w^\bullet$ is a set of “states.”

Here, let $\mathcal{W}_w^\bullet$ be a set of $L_{abf}$-valuations $V_w^{\star\vec{u}}$. We have seen that there is a one-to-one correspondence between such valuations and block-tuple worlds $\vec{u}$ in models for $L_0^\circ$.
An $L_{abd}$-model includes a structure $S_w^\bullet$ for $L_{abd}$. Here, let $S_w^\bullet$ be the structure $S_w^*$ outlined earlier.

An $L_{abd}$-model also includes a valuation function $\mathcal{V}_w^\bullet$ assigning to each state $V_w^{*\vec{u}} \in \mathcal{W}_w^\bullet$ a valuation $\mathcal{V}_w^\bullet(V_w^{*\vec{u}})$. Here, let $\mathcal{V}_w^\bullet(V_w^{*\vec{u}}) = V_w^{*\vec{u}}$. That is, $\mathcal{V}_w^\bullet$ is the identity function on $\mathcal{W}_w^\bullet$. Given the correspondence between valuations $V_w^{*\vec{u}}$ in $L_{abs}$-models and block-tuple worlds $\vec{u}$ in $L_0^\circ$-models, we may safely abbreviate $\mathcal{V}_w^\bullet(V_w^{*\vec{u}})$ as $V_w^{\bullet\vec{u}}$. In that case, we have $V_w^{\bullet\vec{u}} = V_w^{*\vec{u}}$.

(Namely, $V_w^{\bullet\vec{u}}(v) = \mathcal{V}_w^\bullet(V_w^{*\vec{u}})(v) = V_w^{*\vec{u}}(v)$.)
extensional interpretations:

languages to be interpreted:

ability-based interpretations:

extensional interpretations:

- $L_{abd}$-models include a binary accessibility (state-change) relation $[\pi]^w \subseteq \mathcal{W}_w \times \mathcal{W}_w$ for each program $\pi$, with various features reflecting the “semantics” of programming:
  - $V_1 [[v \leftarrow o]] V_2$ iff $V_2 = V_1^{v/o}$
  - $V_1 [[v \Rightarrow o]] V_2$ iff $V_2 = V_1^{v/o}$
  - $[[\pi_1 ; \pi_2]] = [[\pi_1]] \circ [[\pi_2]]$
  - $[[\pi_1 \cup \pi_2]] = [[\pi_1]] \cup [[\pi_2]]$
  - $[[\pi^*]] = [[\pi]^*] = \text{the ancestral of } [[\pi]]$
  - $[[\beta]] = \{ \langle V, V \rangle \mid S_w^*, V \models \beta \}$
  - etc.
In saying that an $L_{lp}$-predicate symbol may be interpreted as a class of models, we mean not classes of $L_{lp}$ models but classes of $L_{abd}$ models of the sort just introduced.

For example, let $\vec{u}$ be the triple of dodecahedra in the world $w$ above, taken in order from left to right. Then, clearly, $\vec{u}$ is in the extension of the $L_0^*$-predicate symbol $P^2_{9} \langle 2,3 \rangle$ (i.e., “LeftOf $\langle 2,3 \rangle$”).

Recall that $f^2_6$ is the composite operation of, first, determining the column and row numbers of each of two components, with respective outcome sets $\{ \langle c_1, r_1 \rangle \}$ and $\{ \langle c_2, r_2 \rangle \}$ (in this case, $\langle 7_1, 8_1 \rangle$ and $\langle 8_2, 8_2 \rangle$), and second, identifying the least (if either) of the two column numbers, with outcome set $\{ 1stcol, 2ndcol, =col \}$. The outcome $1stcol$ is conclusive evidence that the first of the two components is to the left of the second.
In general, there is a unique $M_w$-valuation $V_w^{\vec{u}}$ corresponding to a given $n$-tuple $\vec{u}$ (for $n \geq 3$) such that

$$V_w^{\vec{u}}(f^s_i(p_{\vec{a}}(\theta))) = 0^s_i \iff \langle f^s_i(p_{\vec{a}}(\theta))|_{\vec{u}}, 0^s_i \rangle \in S_w^{\bullet}(\equiv)$$

$$\iff \vec{u} \in S_w^*(\vec{P}_s|_{\vec{a}}).$$

That applies in the present case, where $f^s_i(p_{\vec{a}}(\theta))) = f^2_9(p_{(2,3)}(\theta)))$, $0^s_i = 0^2_9 = 1stcol$, and $\vec{P}_s|_{\vec{a}} = P^2_9|_{\vec{a}} = \text{LeftOf }_{(2,3)}$. 
extensional interpretations:

languages to be interpreted:

ability-based interpretations:

extensional interpretations:
extensional interpretations:  

languages to be interpreted:  

ability-based interpretations:  

extensional interpretations:  

$L_{abd}$ is defined in terms of the kinds of abilities and outcomes illustrated earlier (counting edges of faces, etc.) in the discussion of $L_{abf}$.

$M^*_w$-worlds in $L_{abd}$-models are $L_{abf}$-valuations, which correspond to block-tuple worlds in $L_0^p$-models.

Moving through this network of languages, we can see that we have an operational semantics for the original blocks language $L_{lpl}$.

This is the case, moreover, so as to respect the logic not just of an uninterpreted first-order language but for the logic of analytic and material inference for the interpreted language $L_{lpl}$. 
extensional interpretations:

languages to be interpreted:

ability-based interpretations:

extensional interpretations:
extensional interpretations: 

languages to be interpreted:

ability-based interpretations:

extensional interpretations:
References

