September 4, 2024

Week 2 Notes

# Vocabularies and Metavocabularies

<u>Outline:</u>

I. The 'Vocabulary' Vocabulary
II. Freedom, Language, and Politics
III. Relations Between Vocabularies.
Pragmatic metavocabularies:
a Theory (decomposition), and
a Model for a Syntactic Concept of Vocabularies (*BSD*1).
a Surprising Result distinguishing pragmatic from semantic MVs in terms of relative expressive power.

## <u>Part I: The 'vocabulary' vocabulary</u>

1. Languages/Theories → vocabularies. Quine. Instituting and Applying Concepts (Quine:Carnap) vs. single process (Hegel:Kant, pragmatists).
RR: use "vocabulary"

a) Distinction between *artificial* languages (calculi modeled on mathematical proof, paradigmatically, Hilbert-style axiomatic systems) and *natural* languages.
b) Carnap treats natural languages as though they were artificial ones. He distinguishes two phases of linguistic activity:
    i) picking or settling or specifying *meanings* and
    ii) against the background of those meanings, pick, settle, or specify a *theory*, consisting of the sentences (with those meanings) one takes to be *true*.

    Viewed pragmatically, in terms of what one is *doing,* phase (i) is *instituting* the meanings or contents that determine conceptual norms for the proper *use* of expressions with those meanings, and phase (ii) is *applying* those meanings, to say how the world is.
    In this regard, Carnap follows *Kant*. Kant thought of concepts as determinate rules in virtue of, to begin with, *transcendental* (noumenal) activity, which precedes the *empirical* employment of those concepts (which might yield new "concepts of reflection").
c) Quine objects to Carnap (as Hegel had in effect objected to Kant), that if we look at the *use* of sentences (concepts…), there is only one thing we do with them: make claims (and, as we'll see, challenge and defend them). Somehow that must not only count as

*applying* concepts or meanings, but is all there is to *institute* those meanings. We must be able to think of *use* as having *both* these aspects.

d) More specifically, Quine argues:
   i. Meaning must at least determine inferential role: what counts as reasons for and against claims with those meanings.
   ii. What counts as a reason for or against a claimable depends on (differs with differences in) the context of collateral commitments that provide further premises or auxiliary hypotheses.
   iii. So, he says, meaning depends on belief. The converse obviously holds, too. So we should not theoretically distinguish between meaning and belief, language and theory.
   iv. Further, when we alter our commitments, there is nothing in the practice that corresponds to doing so because we changed our collateral beliefs and doing so because we changed our meanings.

This line of thought leads to two further ones (Sellars on inferential roles, Alternative of rejecting semantics in favor of pragmatics):

e) Inferential roles:
   Sellars on labeling vs. describing, as requiring "situation in a space of implications", and the further claim that those implications must be subjunctively robust, so come with a *range of subjunctive robustness*. [This *might* be done in connection with Quine's claim that meaning must at least determine inferential role, and that what follows from, implies, or is incompatible with what depends on the *context of collateral commitments*, i.e. what *theory* one endorses. ]

f) Semantic nihilism and instrumentalism:
   LW has argued, in parallel to Quine, that discursive practice requires not just agreement in meanings, but also (as a consequence), and he thinks, surprisingly, agreement in belief. (His example of arbitraging with folks who sell firewood by the area of ground the pile covers, rather than its volume.)
   Dummett on LW as semantic nihilist: don't postulate meanings, because "philosophy is not one of the natural sciences." LW agrees with Sellars on picking out postulation of unobservables as characteristic of scientific inquiry, and means to be ruling that out. Dummett himself endorses that conclusion, requiring that semantic (meta)concepts be cashable out into proprieties of *use*.
   In effect: *semantic* MVs must be *translatable* into *pragmatic* MVs.
   But, Sellars would object, this is *instrumentalism* about theoretical entities.

g) Q: So what is the lesson we should draw about the proper relations between pragmatic and semantic MVs?

**<u>Part II: Freedom, Language, and Politics</u>**

**Current plan is only to *mention* Rorty on the power of redescription,**
**Rush through rest:**
**Essentially self-conscious (self-descriptive, vocabulary-using) creatures.**
**[Skip Kant.]**
Mention: **a) Chomsky point**
**As providing an argument for:**
**b) inapplicability of instrumental 'tool' form of intelligibility**
For that model requires at least i) antecedent specifiability of goal or end, ii) in-principle availability of alternate means to that end.
Dewey on language as the "tool of tools": a tool for making tools.
But, really, it is a tool for getting new ends, goals, or desires.
And *that* feature of it is not to be understood instrumentally.
**Conclude: Give up negative freedom for bonanza of positive expressive freedom.**
Hegel: This feature makes *language* the paradigm of a social institution for which the *political* question "Why is this constraint on my freedom legitimate?" does not arise, or is quickly dealt with.

2. Rorty on vocabularies and the distinctive power of redescription.

a) Rorty introduces the 'vocabulary' vocabulary, in responses to Quine's pragmatist objection to the language/theory, meaning/belief distinctions.
b) **<u>Redescription</u>**: (This story from *Contingency, Irony, and Solidarity* [1989].)

- In many ways this course wants to use the tools of the book *RLLR* to investigate the power of *redescription*: using a different *vocabulary* to state facts.
- We would like a formally tractable theory of what we are doing when we redescribe things, restate the facts in some region.
- To do that, we must think about what it is to state facts, using OED vocabulary, on the way to seeing what difference it makes—what distinctive kind of *understanding* we achieve—if we *change* vocabularies in addressing some problem or topic.
- One of the things we must understand in order to understand the kind understanding characteristic of redescription is what a descriptive vocabulary is: what it is to use vocabulary in a descriptive way. Holding, as it were, the *world* fixed, what difference does it make what *vocabulary* we use to *describe* it? Notice that this question is prior to, or at least equiprimordial with, questions about the *truth* of descriptions. For we could restrict our attention to *true* descriptions, and still hope to get some insight from thinking about them in terms of *multiple* true descriptions, *re*descriptions. This is a way into thinking about vocabularies, descriptions, and truth.

- In fact we will address this issue to begin with at a *meta*level, by thinking about different specifications or descriptions of *vocabularies*: constellations of expressions that can be used to *say* something, to say *that* things are thus-and-so.
- Doing this will, I claim, provide tools useful for thinking about lots of philosophical issues.
- Further, it will teach us something important about what we are doing when we address an issue philosophically. For it is useful to think of at least many central philosophical tasks as redescriptive tasks, as a matter of bringing to bear a new vocabulary on an area of concern.

3. Here is a way of thinking about the transformative potential of redescription.
**Essentially self-conscious creatures**: what they are for themselves (description in a vocabulary) is part of what they are in themselves. So they are subject to a distinctive kind of self-transformative possibility: changing what they are *in* themselves by changing what they are *for* themselves, i.e. the vocabulary they use to describe the world and themselves. These are *geistig* beings.

4. Background: Kantian tradition (out of Rousseau) on freedom. “Obedience to a law one has set down for oneself is freedom.” ( *Social Contract*)

Two features of language:
A)
5. Chomskyan Observation: Nearly every sentence uttered by an adult native speaker is radically novel, not only has the speaker/audience never used that exact sentence, neither has anyone else, *ever*.
So:
6. Not a tool:
It is not *for* anything. More specifically, it is not to be accounted for on the instrumental model of a *tool* for doing something. For that model requires an independently specifiable end or goal, which could in principle (as far as its definition is concerned) be achieved by different means. These are not satisfied in the case of language: it is not to satisfy desires, but to formulate new ones.
This is not to say that an evolutionary account of the *advent* (better: persistence, flourishing, adaptational significance) of language might not be possible, in terms of how discursive practices allowed new kinds of cooperation that were useful nutritionally and in other biological ways.

B)
7. Conclusion: Can use *language* as a paradigm of a social institution that asks us to surrender some negative freedom—freedom from constraint—by subjecting ourselves to

assessment according norms, but that repays that sacrifice a hundred times over in the positive expressive power that it grants in return.
This is discursive practice as a *political* paradigm.

A punchline here is that we want our formal theories of language to address this crucial emancipatory-developmental dimension of language use. That is, I want an account of pragmatics/semantics that will explain the role of language as the medium and form of *freedom* (and self-constitution), *because* it is the medium of *reasons*.
This is a serious criterion of adequacy to put on a formal theory—who else's semantics addresses this *desideratum*?
This criterion of adequacy we'll address with theory of recollective rationality in terms of revising reason relations to permit explicitation paths of a certain kind.

## <u>**Part III: Relations between vocabularies. *BSD* on pragmatic MVs**</u>

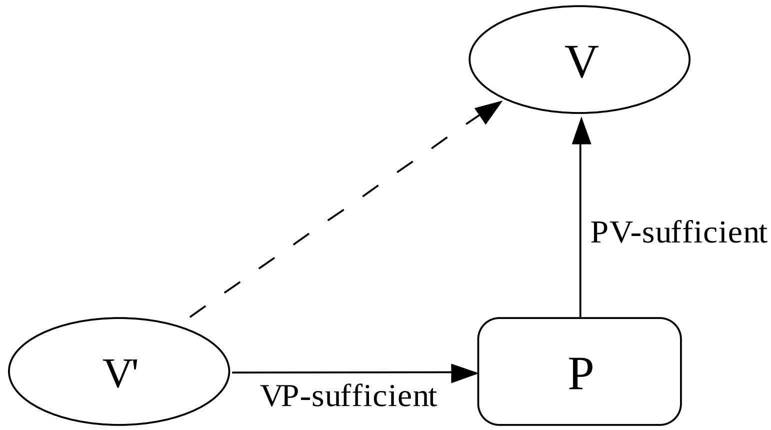
8. Let us now use the ‘vocabulary’ vocabulary to recharacterize the analytic-logistical tradition in TwenCen philosophy of language. (BSD story.)
Reconstructing the project of analytic philosophy in terms of the relations between two vocabularies: base and target.
The special role of logic, as “coming for free” in this enterprise.
The idea is that logic adds no *content*, merely making explicit *form*. (MacFarlane dissertation.)

We can now see Quine’s (and LW’s) points as directly criticizing the presuppositions of the analytic program. ***Meaning* only matters if and insofar as it contributes to codifying/explaining/explicating proprieties of *use***.

Wilfrid Sellars (one of my particular heroes) criticizes the empiricist core program of the classical project of analysis on the basis of what one must *do* in order to *use* various vocabularies, and so to count as *saying* or *thinking* various things. He argues **that *none* of the various candidates for empiricist base vocabularies are practically *autonomous*, that is, could be deployed in a language game one played though one played no other**. For instance, no discursive practice can consist entirely of making non-inferential observation reports. For such reliably differentially elicited responses qualify as *conceptually* contentful or *cognitively* significant only insofar as they can serve as *premises* from which it is appropriate to draw *conclusions*, that is, as *reasons* for other judgments. Drawing such conclusions is applying concepts *inferentially*—that is, precisely *not* making non-inferential *observational* use of them

9. Further relations among vocabularies:
The development of the concept of <u>metavocabulary</u>:
Carnap, Tarski, young Sellars.
Carnap in *The Logical Syntax of Language* (1934, in English in 1937) used *syntactic* metavocabulary.
Tarski introduced *semantic* MVs in his mid-30s work on *truth*.
The young Sellars (1947-1955) introduces the notion of *pragmatic metalanguage*, and with it the idea that all of philosophy is a form of *a priori pragmatic theory*.

10. A *theory* of pragmatic metavocabularies:
We can **decompose** the notion of a paragmatic metavocabulary into two components:
The MV is *VP-sufficient* to *specify*
Discursive practices that are *PV-sufficient* to *deploy* a target vocabulary.




The conventions of this diagram are:

- Vocabularies are shown as ovals, practices-or-abilities as (rounded) rectangles.
- Basic meaning-use relations are indicated by solid arrows, numbered and labeled as to kind of relation.
- Resultant meaning-use relations are indicated by dotted arrows, numbered, and labeled as to kind and the basic MURs from which they result.

The idea is that a resultant MUR is the relation that obtains when all of the basic MURs listed on its label obtain.

Putting these relations together:
This holds when
the statements made in the pragmatic MV V' are *true of* (if they succeed in describing) practices-or-abilities P
[A word about "practices-or-abilities", as agnostic between the two: social vs. individual emphasis. For the purposes of the present project, I will maintain a studied neutrality between these options. The apparatus I am introducing can be noncommittal as to whether we understand content-conferring *uses* of expressions in terms of social practices or individual abilities.]
such that anyone who engages in those practices or exercises those abilities will be deploying the vocabulary V, using those words to mean what they mean in V.

Note that this is a *meaning-use* diagram (MUD), expressing a *meaning-use* analysis (MUA) of a *pragmatically mediated semantic relationship among vocabularies*. This meta-metavocabulary of MUAs and MUDs can characterize *many* more kinds of pragmatically mediate semantic relations among vocabularies. This capacity is exploited in the rest of *Between Saying and Doing*—but I'm not going to rehearse any of that here.

**My basic suggestion for extending the classical project of *analysis* so as to incorporate as essential positive elements the insights that animate the *pragmatist* critique of that project is**

**that, alongside the classical semantic relations between vocabularies that project has traditionally appealed to, we consider also *pragmatically mediated* ones—of which the relation of being a pragmatic metavocabulary is a paradigm.**

11. I claim that this simple theoretical decomposition of the notion of a <u>pragmatic MV</u>, and the genus of *pragmatically mediated (semantic) relations among vocabularies* that it introduces and epitomizes, is illuminating. (Next week, in Week 3, we present our more specific suggestions for a pragmatic MV, which lets us introduce a notion of <u>reason relations</u> that is suitable for formal work.) One way to begin to benefit from that illumination is to look at a simplified model.
We can start by applying this theory to the pragmatics of vocabularies thought of independently of their semantics, solely in terms of their syntax. (In this we resonate with Carnap's 1934 *LSL*, which remains resolutely at the level of syntax.)

We need some elementary computational linguistics:
Specifically, we can look at strings of symbols generated by various kinds of **formal grammars**, and the **finite-state automata** that can "use" such strings in the sense of **'reading' and 'writing'** them.

a) We start with an *alphabet*: a finite set of symbols.

From that one forms the *universe* of strings freely generated by that alphabet: all the finite strings one gets by concatenating elements of the alphabet.
And we then define, in this automaton-theoretic setting, a purely syntactic notion of a *vocabulary*: Vocabularies are just subsets of the universe.

b) "Use" of the vocabulary is reading or writing it:

- The task of *reading* the vocabulary is the task of offering a verdict on any sample string: is it in the distinguished proper subset of the universe that is the vocabulary? Yes or no?
- The task of *writing* the vocabulary is the task of producing, for any input or for none, *only* strings that are in the vocabulary, in a way that *can* produce *any* of them.

This can be done by *finite-state automata*.

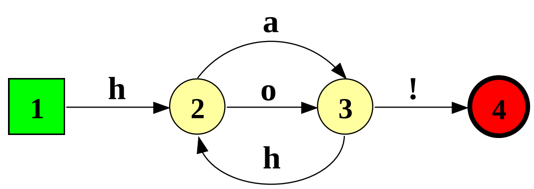
c) <u>Example of a pragmatic MV for a very simplified case</u>:
A vocabulary in a purely *syntactic* sense is a proper subset of the *universe* of strings generated by concatenating elements of some finite *alphabet*.
The ability to 'read' that vocabulary is the ability to tell, of any given string, whether or not it is in the privileged vocabulry.
The ability to 'write' that vocabulary is the ability to produce only (and each of) the licit strings of the vocabulary.
The theory of *finite-state automata* (FSA) offers pragmatic MVs for specifying automata that can read and write vocabularies (in this syntactic sense) of various kinds.

Such automata are presumed to have the primitive ability to read and write (when prompted) arbitrary letters from the alphabet, and the primitive rule-following abilities needed to implement conditional branched-schedule algorithms: to exercise its abilities differently depending on what state it is in, and to move between states as the algorithm specifies.

d) The laughing Santa vocabulary includes not only his trademark "Hohoho!", but also all other finite strings consisting of a sequence of 'ha's as well as 'ho's, finished with an exclamation mark: "Hahahoho!" and "Hohahoha!" and so on.

e) Here are specifications of the Laughing Santa automaton in three different pragmatic MVs:

**The Laughing Santa Automaton**




| | **State 1** | **State 2** | **State 3** |
|---|---|---|---|
| **a** | Halt | 3 | Halt |
| **h** | 2 | Halt | 2 |
| **o** | Halt | 3 | Halt |
| **!** | Halt | Halt | 4 |

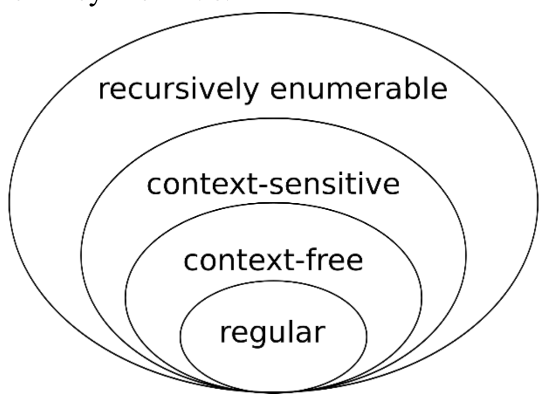
aHalt3Halth2Halt2oHalt3Halt!HaltHalt4.

12. This model gives us concrete examples of both
Practices PV-sufficient to *deploy* a (syntactic) vocabulary: i.e. to read and write it, and
A vocabulary VP-sufficient to *specify* those practices.
These latter can be the state-*diagram* formulation, the state-*table* formulation, or in the form of a *string* that encodes the state-table.
This latter is itself just a syntactic vocabulary in the universe generated by some alphabet.
It is a *pragmatic meta*vocabulary in this purely syntactic setting.

13. We can now investigate the practices of *using* that pragmatic syntactic metavocabulary: that is we can look at the finite automata that can read and write that pragmatic metavocabulary.

When we do, we make an astonishing discovery: The structure of the relations between the expressive power of *pragmatic* MVs and their base Vs is *very* different from the structure of the relations between the expressive power of *semantic* MVs relative to their base Vs.
A new phenomenon come into view: *pragmatic expressive bootstrapping*.
What we'll look for, when we introduce a proper pragmatic MV for semantically significant strings (those that stand in reason relations), is an analogue of this result, but for semantically significant sentences. We'll see that in Week 4 (sketched in ATBUYO).

14. Pragmatic Expressive Bootstrapping.
Compared to *semantic* MVs, where Tarski showed that under minimal conditions (concerning what the underlying V can express), semantic MVs must be asymmetrically expressively *stronger* than the Vs for which they are MVs.




The Chomsky Hierarchy

| Vocabulary | Grammar | Automaton |
|---|---|---|
| Regular | A→aB<br>A→a | Finite State Automaton |
| Context-Free | A→<anything> | Push-Down Automaton |
| Context-Sensitive | $c_1Ac_2$→c1<anything>$c_2$ | Linear Bounded Automaton |
| Recursively Enumerable | No Restrictions on Rules | Turing Machine (= 2 Stack PDA) |

Wikipedia:
A linear bounded automaton is a Turing machine that satisfies the following three conditions:
- Its input alphabet includes two special symbols, serving as left and right endmarkers.
- Its transitions may not print other symbols over the endmarkers.
- Its transitions may neither move to the left of the left endmarker nor to the right of the right endmarker.[1]:225

In other words: instead of having potentially infinite tape on which to compute, computation is restricted to the portion of the tape containing the input plus the two tape squares holding the endmarkers.

An alternative, less restrictive definition is as follows:

- Like a Turing machine, an LBA possesses a tape made up of cells that can contain symbols from a finite alphabet, a head that can read from or write to one cell on the tape at a time and can be moved, and a finite number of states.
- An LBA differs from a Turing machine in that while the tape is initially considered to have unbounded length, only a finite contiguous portion of the tape, whose length is a linear function of the length of the initial input, can be accessed by the read/write head; hence the name *linear bounded automaton*.[1]:225

This limitation makes an LBA a somewhat more accurate model of a real-world computer than a Turing machine, whose definition assumes unlimited tape.

Linear bounded automata are acceptors for the class of context-sensitive languages.[1]:225–226 The only restriction placed on grammars for such languages is that no production maps a string to a shorter string. Thus no derivation of a string in a context-sensitive language can contain a sentential form longer than the string itself. Since there is a one-to-one correspondence between linear-bounded automata and such grammars, no more tape than that occupied by the original string is necessary for the string to be recognized by the automaton

Pragmatic expressive bootstrapping shows that context-free and context-sensitive pragmatic MVs are in principle adequate to specify Turing Machines (which can read and write all computable (=recursively enumerable) syntactic vocabularies, not just regular, context-free, and context-sensitive ones.

(Note that context-sensitive grammars, which have context both before and after the insertion, are in a way *bilateral* in their notion of context: it comes in two parts. Context-free only allow one sort (side) of context.

15. Punchline:

We have now seen a concrete (relatively) model of VP-suff/PV-suff theory of pragmatic MVs. State-diagrams and state-tables are expressively sufficient (in both stages) pragmatic MVs for the syntactic vocabularies that finite-state automata (including single and double pushdown automata = Turing Machines) can read and write (the notion of "use" of expressions relevant in this setting).

Here we have a sufficiently clear and precise notion of pragmatic MVs that we can prove a theorem about them: that pragmatic expressive bootstrapping is not just intelligible, but real.

We aspire to being able to work with this sort of precision on vocabularies that involve semantic articulation of the expressions of the lexicon: vocabularies as sets of sentences plus reason relations among those sentences. (We will use sets of pairs of sets of sentences to represent the reason relations of implication and incompatibility.)

In Week 3 we will elaborate a pragmatics and a pragmatic MV that specifies what one must *do*, what structure the practices must have, in order properly to be understood as governed by implicit norms in the form of reason relations.

In Week 4 we will see how to compare, contrast, and connect this formal pragmatic MV with Fine's truthmaker representational semantic MV. Once again, we come across a surprising theorem about the relation between (the right kind of) pragmatic and semantic MVs—and the reason relations they express and explain.

This should be understood as the semantic cash for the syntactic promissory note of the pragmatic expressive bootstrapping result. Both present *formal* results relating pragmatic to semantic MVs. In the *BSD* case I use to introduce the idea, the model is purely syntactic. But there we can see already that approaching things from the pragmatic side, that is, using *pragmatic* MVs, has expressive advantages (pragmatic MVs can be expressively weaker than their base Vs, where semantic MVs must always be stronger). In the full-blown semantic case we care about, the Hlobil isomorphism is the first real result of its kind relating pragmatic and semantic MVs. One reason for that is that people haven't developed their pragmatic MVs in an appropriately mathematically tractable form. For us, the pragmatic MV came from using Restall/Ripley versions of sequent calculi in the context of *MIE* double-barreled deontic pragmatic MV. (Ryan Simonelli was a very active participant in bringing these two together—and he has lots of further, different ideas about how to do it all better.)
This movement completes the arc that *BSD* was meant to take. There, the semantics was the (unsatisfactory) incompatibility semantics, and the pragmatics/semantics MVs connection was deontic/alethic. That last feature is the basis on which Ulf worked out his isomorphism to implement. So we are filling in a different path from (a version of) the beginning of *BSD* to (a version of) the end.

16. Meaning and Use.

The thought underlying the pragmatist line of thought is that what makes some bit of vocabulary mean what it does is how it is used. What we could call *semantic* pragmatism is the view that the only explanation there could be for how a given *meaning* gets associated with a vocabulary is to be found in the *use* of that vocabulary: the practices by which that meaning is conferred or the abilities whose exercise constitutes deploying a vocabulary with that meaning. To broaden the classical project of analysis in the light of the pragmatists' insistence on the centrality of pragmatics, we can focus on this fundamental relation between use and meaning, between practices or practical abilities and vocabularies. **We must look at what it is to use locutions *as* expressing meanings—that is, at what one must *do* in order to count as *saying* what the vocabulary lets practitioners express. I am going to call this kind of relation "practice-vocabulary sufficiency"—or usually, "PV-sufficiency,"** for short. It obtains when engaging in a specified set of practices or exercising a specified set of abilities[1] is sufficient for someone to count as *deploying* a specified vocabulary.

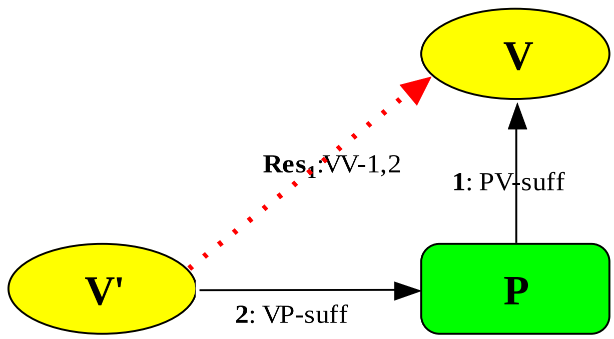
Of course it matters a lot how we think about these content-conferring, vocabulary-deploying practices or abilities. The semantic pragmatist's claim that use confers meaning (so talk of practices or the exercise of abilities as deploying vocabularies) reverts to triviality if we are allowed to talk about "using the tilde to express negation," "the ability to mean red by the word 'red'," or "the capacity to refer to electrons by the word 'electron'," (or, I think, even *intentions* so to refer). And that is to say that the interest of the PV-sufficiency of some set of practices or abilities for the deploying of a vocabulary is quite sensitive to the *vocabulary* in which we *specify* those practices-or-abilities. **Talk of practices-or-abilities has a definite sense only insofar as it is relativized to the vocabulary in which those practices-or-abilities are specified.** And that means that **besides PV-sufficiency, we should consider a second basic meaning-use relation: "vocabulary-practice sufficiency," or just "VP-sufficiency," is the relation that holds between a vocabulary and a set of practices-or-abilities when that vocabulary is sufficient to *specify* those practices-or-abilities**.[2] VP-sufficient vocabularies that *specify* PV-sufficient practices let one *say* what it is one must *do* to count as engaging in those practices or exercising those abilities, and so to deploy a vocabulary to *say* something.

PV-sufficiency and VP-sufficiency are two basic *meaning-use* relations (MURs). In terms of those basic relations, we can define a more complex relation: the relation that holds between vocabulary V' and vocabulary V when V' is VP-sufficient to specify practices-or-abilities P that are PV-sufficient to deploy vocabulary V. This VV-relation is the *composition* of the two basic MURs. When it obtains I will say that V' is a *pragmatic metavocabulary* for V. It allows one to *say* what one must *do* in order to count as *saying* the things expressed by vocabulary V. We can present this relation graphically in a *meaning-use diagram* (MUD):

1 For the purposes of the present project, I will maintain a studied neutrality between these options. The apparatus I am introducing can be noncommittal as to whether we understand content-conferring *uses* of expressions in terms of social practices or individual abilities.

2 Somewhat more precisely: some theory (a set of sentences), formulable in the vocabulary in question, is such that if all those sentences are true of some interlocutor, then it thereby counts as exercising the relevant ability, or engaging in the relevant practices.

**Meaning-Use Diagram #1:**
**Pragmatic**
**Metavocabulary**

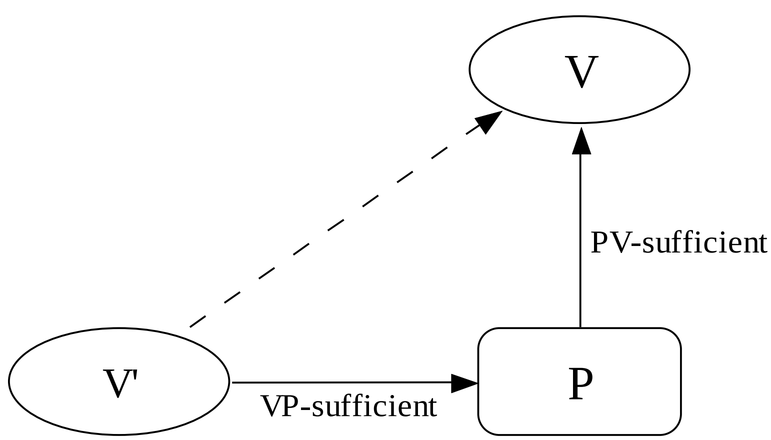



The conventions of this diagram are:

- Vocabularies are shown as ovals, practices-or-abilities as (rounded) rectangles.
- Basic meaning-use relations are indicated by solid arrows, numbered and labeled as to kind of relation.
- Resultant meaning-use relations are indicated by dotted arrows, numbered, and labeled as to kind and the basic MURs from which they result.

The idea is that a resultant MUR is the relation that obtains when all of the basic MURs listed on its label obtain.

17. Reconstructing the concept of pragmatic MV.




18.

19. Being a pragmatic metavocabulary is the simplest species of the genus I want to introduce here. It is a *pragmatically mediated semantic relation* between vocabularies. It is pragmatically mediated by the practices-or-abilities that are *specified* by one of the

vocabularies (which *say* what counts as *doing* that) and that *deploy* or are the *use* of the other vocabulary (what one says *by* doing that). The semantic relation that is established thereby between the two vocabularies is of a distinctive sort, quite different from, for instance, definability, translatability, reducibility, and supervenience. **My basic suggestion for extending the classical project of *analysis* so as to incorporate as essential positive elements the insights that animate the *pragmatist* critique of that project is that, alongside the classical semantic relations between vocabularies that project has traditionally appealed to, we consider also *pragmatically mediated* ones—of which the relation of being a pragmatic metavocabulary is a paradigm.**

20.

21. Under what circumstances would this simplest pragmatically mediated semantic relation be philosophically interesting, when considered in connection with the sorts of vocabularies that have traditionally been of most interest to classical analysis? At least one sort of result that could be of considerable potential significance, I think, is if it turned out that in some cases pragmatic metavocabularies exist that differ significantly in their expressive power from the vocabularies for the deployment of which they specify sufficient practices-or-abilities. I will call that phenomenon **"*pragmatic expressive bootstrapping*."**

The point of looking at **pragmatic expressive bootstrapping** is that this shows the *value* of defining pragmatic MVs.

In working out the example, we get precise, formally tractable versions of VP-sufficiency (state description of automaton suffices for practical ability) and PV-sufficiency (automaton can read/write vocabulary: distinguish strings of the vocabulary in those two senses).

And we can then use the automaton-theoretic formalism to show a surprising and potentially significant result: pragmatic expressive bootstrapping.

**So in outline, it is these three things:**

i. **MUD representation of pragmatic MVs**
ii. **Finite automata model of VP-suff and PV-suff,**
iii. **Expressive bootstrapping by pragmatic MV is *provable*.**

So we *can* hope for both formally tractable pragmatic MV, to lay alongside traditional formal semantic MVs, *and* for the *relations* between *pragmatic* and *semantic* MVs to be formally expressible, as well.

We will see how to do that with Week 3's carefully articulated pragmatics, culminating in the definition of the two kinds of reason relation, to get the formally tractable pragmatic MV, and then Week 4 on the relations between such pragmatic MVs and truthmaker representational formal semantic MVs.

22. Now this is all extremely abstract. To make it more definite, we need to fill in (at least) the notions of vocabulary, practice-or-ability, PV-sufficiency, and VP-sufficiency, which are the fundamental elements that articulate what I am calling the "meaning-use analysis" of resultant meaning-use relations—in particular, the pragmatically mediated semantic relations between vocabularies that I am claiming we must acknowledge in order to pursue the classical project of philosophical analysis in the light of what is right about the pragmatist critique of it. We can begin to do that by looking at a special case in which it is possible to be unusually clear and precise about the things and relations that play these metatheoretic roles. This is **the case where 'vocabulary' takes a purely *syntactic* sense**. Of course, the cases we eventually care about involve vocabularies understood in a sense that includes their *semantic* significance. But besides the advantages of clarity and simplicity, we will find that some important lessons carry over from the syntactic to the semantic case.

This is the motivation for bringing in the *model* of finite automata:
It gives us ***precise* analogues of VP sufficiency and PV-sufficiency**.
They are precise enough for us to show an expressive result, to prove a theorem about the relative expressive powers of some vocabularies.

Restricting ourselves to a purely syntactic notion of a vocabulary yields a clear sense of 'pragmatic metavocabulary': **both the digraph and the state-table vocabularies are VP-sufficient to specify practical abilities articulated as a finite-state automaton that is PV-sufficient to deploy—in the sense of recognizing and producing—the laughing Santa vocabulary**, as well as many others. (Of course, it does that only against the background of a set of abilities PV-sufficient to deploy *those* vocabularies.) Perhaps surprisingly, it also offers a prime example of *strict pragmatic expressive bootstrapping*. For in this setting we can *prove* that one vocabulary that is expressively weaker than another can nonetheless serve as an adequate *pragmatic* metavocabulary for that stronger vocabulary. That is, even though one can*not say* in the weaker vocabulary everything that can be *said* in the stronger one, one can still *say* in the weaker one everything that one needs to be able to *do* in order to deploy the stronger one.

23.
If we look at *pragmatically mediated* relations between these syntactically characterized vocabularies, we find that they make possible a kind of *strict expressive bootstrapping* that permits us in a certain sense to evade the restrictions on expressive power enforced for purely syntactic relations between vocabularies. The hierarchy dictates that only the abilities codified in Turing Machines—two-stack push-down automata—are *PV-sufficient* to *deploy* recursively

enumerable vocabularies in general. But now we can ask: what class of languages is *VP-sufficient* to *specify* Turing Machines, and hence to serve as sufficient *pragmatic* metavocabularies for recursively enumerable vocabularies in general? The surprising fact is that **the abilities codified in Turing Machines—the abilities to recognize and produce arbitrary recursively enumerable vocabularies—can quite generally be specified in *context-free* vocabularies**. It is demonstrable that context-free vocabularies are strictly weaker in syntactic expressive resources than recursively enumerable vocabularies. The push-down automata that can read and write only context-free vocabularies cannot read and write recursively enumerable vocabularies in general. But it is possible to *say* in a context-free vocabulary what one needs to be able to *do* in order to deploy recursively enumerable vocabularies in general.

24.

## IV. Automata: Syntactic PV-sufficiency and VP-sufficiency

Now this is all extremely abstract. To make it more definite, we need to fill in (at least) the notions of <u>vocabulary</u>, <u>practice-or-ability</u>, <u>PV-sufficiency</u>, and <u>VP-sufficiency</u>, which are the fundamental elements that articulate what I am calling the "meaning-use analysis" of resultant meaning-use relations—in particular, the pragmatically mediated semantic relations between vocabularies that I am claiming we must acknowledge in order to pursue the classical project of philosophical analysis in the light of what is right about the pragmatist critique of it. We can begin to do that by looking at a special case in which it is possible to be unusually clear and precise about the things and relations that play these metatheoretic roles. This is the case where 'vocabulary' takes a purely *syntactic* sense. Of course, the cases we eventually care about involve vocabularies understood in a sense that includes their *semantic* significance. But besides the advantages of clarity and simplicity, we will find that some important lessons carry over from the syntactic to the semantic case.

The restriction to vocabularies understood in a spare syntactic sense leads to correspondingly restricted notions of what it is to *deploy* such a vocabulary, and what it is to *specify* practices-or-abilities sufficient to deploy one. Suppose we are given an *alphabet*, which is a finite set of primitive sign types—for instance, the letters of the English alphabet. The *universe* generated by that alphabet then consists of all the finite strings that can be formed by concatenating elements drawn from the alphabet. A *vocabulary* over such an alphabet—in the syntactic sense I am now after—is then any subset of the universe of strings that alphabet generates. If the generating alphabet is the English alphabet, then the vocabulary might consist of all English sentences, all possible English texts, or all and only the sentences of *Making It Explicit*.[3]

What can we say about the abilities that count as *deploying* a vocabulary in this spare syntactic sense?[4] The abilities in question are the capacity to *read* and *write* the vocabulary. In this purely syntactic sense, 'reading' it means being able practically to *distinguish* within the universe generated by the alphabet, strings that do, from those that do not, belong to the specified vocabulary. And 'writing' it means practically being able to *produce* all and only the strings in the alphabetic universe that do belong to the vocabulary.

We assume as primitive abilities the capacities to read and write, in this sense, the alphabet from whose universe the vocabulary is drawn—that is, the capacity to respond differentially to alphabetic tokens according to their type, and to produce tokens of antecedently specified alphabetic types. Then the abilities that are PV-sufficient to deploy some vocabularies can be specified in a particularly simple form. They are *finite-state automata* (FSAs). As an example, suppose we begin with the alphabet {a, h, o, !}. Then we can consider the *laughing Santa vocabulary*, which consists of strings such as 'hahaha!', 'hohoho!', 'hahahoho!' 'hohoha!', and so on.[5] Here is a graphical representation of a *laughing Santa finite-state automaton*, which can read and write the laughing Santa vocabulary:

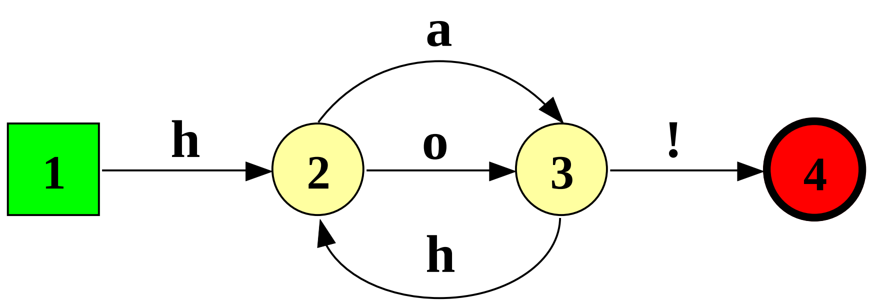
---

3 Computational linguists, who worry about vocabularies in this sense, have developed metalanguages for specifying important classes of such vocabularies: the syntactic analogues of semantic metalanguages in the cases we will eventually address. So, for instance, for the alphabet {a,b}, '$a^n b^n$' characterizes the vocabulary that comprises all strings of some finite number of 'a's followed by the *same* number of 'b's. '$a(ba)^* b$' characterizes the vocabulary that comprises all strings beginning with an 'a', ending with a 'b', and having *any* number of repetitions of the sub-string 'ba' in between.

4 Here we can safely just talk about *abilities*, without danger of restricting the generality of the analysis.

5 In the syntactic metalanguage for specifying vocabularies that I mentioned in the note above, this is the vocabulary (ha|ho)*!

**The Laughing Santa**
**Automaton**




The numbered nodes represent the *states* of the automaton, and the alphabetically labeled arcs represent *state-transitions*. By convention, the starting state is represented by a square (State 1), and the final state by a circle with a thick border (State 4).

As a *reader* of the laughing Santa vocabulary, the task of this automaton is to process a finite string, and determine whether or not it is a licit string of the vocabulary. It processes the string one alphabetic character at a time, beginning in State 1. It recognizes the string if and only if (when and only when) it arrives at its final state, State 4. If the first character of the string is not an 'h', it remains stuck in State 1, and rejects the string. If the first character is an 'h', it moves to State 2, and processes the next character. If that character is not an 'a' or an 'o', it remains stuck in State 2, and rejects the string. If the character is an 'a' or an 'o', it moves to State 3. If the next character is an exclamation point, it moves to State 4, and recognizes the string 'ha!' or 'ho!'—the shortest ones in the laughing Santa vocabulary. If instead the next character is an 'h', it goes back to State 2, and repeats itself in loops of 'ha's and 'ho's any number of times until an exclamation point is finally reached, or it is fed a discordant character.

As a *writer* of the laughing Santa vocabulary, the task of the automaton is to produce only licit strings of that vocabulary, by a process that can produce any and all such strings. It begins in its initial state, State 1, and emits an 'h' (its only available move), changing to State 2. In this state, it can produce either an 'a' or an 'o'—it selects one at random[6]—and goes into State 3. In

6 As a matter of fact, it can be shown that every vocabulary readable/writeable by a non-deterministic finite-state automaton—such as the laughing Santa automaton—is also readable/writeable by a deterministic one.

this state, it can either tack on an exclamation point, and move into its final state, State 4, finishing the process, or emit another 'h' and return to State 2 to repeat the process. In any case, whenever it reaches State 4 and halts, the string it has constructed will be a member of the laughing Santa vocabulary.

I hope this brief rehearsal makes it clear how the constellation of nodes and arrows that makes up this directed graph represents the abilities to read and write (recognize and produce arbitrary strings of) the laughing Santa vocabulary. What it represents is abilities that are *PV-sufficient* to *deploy* that vocabulary—that is, read and write it, in the attenuated sense appropriate to this purely syntactic case. And the digraph representation is itself a *vocabulary* that is *VP-sufficient* to *specify* those vocabulary-deploying abilities. That is, the digraph representation of this finite-state automaton is a *pragmatic metavocabulary* for the laughing Santa vocabulary. The relation between the digraph vocabulary and the laughing Santa vocabulary is, then, a *pragmatically mediated*—not now *semantic*, but *syntactic*—relation between vocabularies.

It may seem that I am stretching things by calling the digraph form of representation a 'vocabulary'. It will be useful, as a way of introducing my final point in the vicinity, to consider a different form of pragmatic metavocabulary for the laughing Santa vocabulary. Besides the digraph representation of a finite-state automaton, we can also use a *state-table* representation. For the laughing Santa automaton this is:

| | **State 1** | **State 2** | **State 3** |
|---|---|---|---|
| **a** | Halt | 3 | Halt |
| **h** | 2 | Halt | 2 |
| **o** | Halt | 3 | Halt |
| **!** | Halt | Halt | 4 |

In read mode, the automaton starts in State 1. To see what it will do if fed a particular character, we look at the row labeled with that character. The LSA will Halt if the input string starts with anything other than an 'h', in which

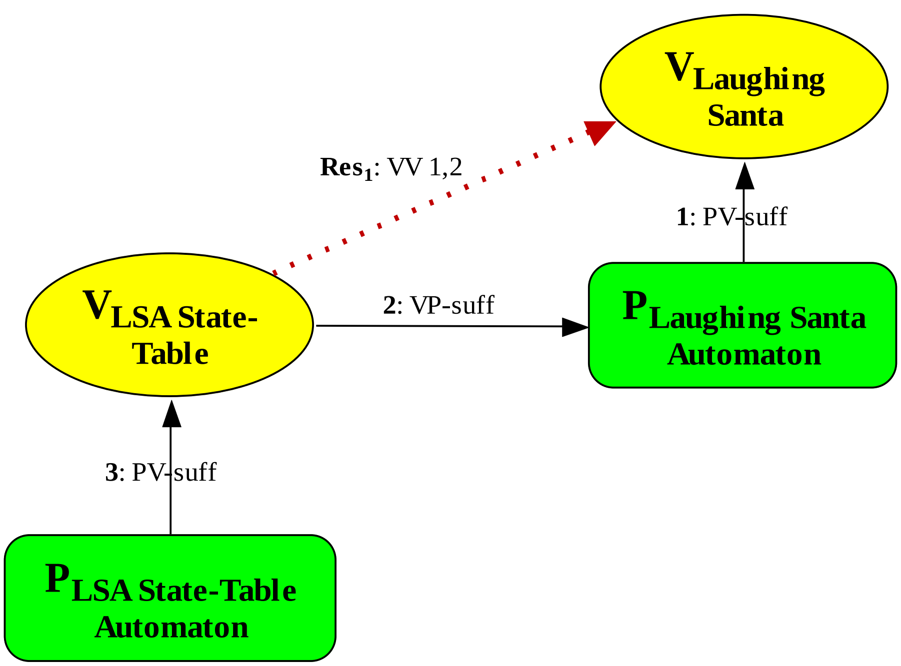
M. O. Rabin and D. Scott, "Finite Automata and their Decision Problems," *IBM Journal of Research and Development* 3, no. 2 (1959), pp. 115-25.

case it will change to State 2. In that state, the automaton specified by the table will halt unless the next character is an 'a' or an 'o', in which case it changes to State 3, and so on. (There is no column for State 4, since it is the final state, and accepts/produces no further characters.) Clearly there is a tabular representation corresponding to any digraph representation of an FSA, and vice versa. Notice further that we need not use a two-dimensional table to convey this information. We could put the rows one after another, in the form:

aHalt3Halth2Halt2oHalt3Halt!HaltHalt4.

This is just a string, drawn from a universe generated by the alphabet of the LSA, together with 'Halt' and the designations of the states of that automaton. The strings that specify finite-state automata that deploy vocabularies defined over the same basic alphabet as the LSA then form a vocabulary in the technical syntactic sense we have been considering. And that means we can ask about the automata that can read and write *those* state-table encoding vocabularies. The meaning-use diagram for this situation is then:

**Meaning-Use Diagram #6: Specifying the Automaton that Deploys the Laughing Santa Vocabulary**



## V. The Chomsky Hierarchy:

## A Syntactic Example of Pragmatic Expressive Bootstrapping

Restricting ourselves to a purely syntactic notion of a vocabulary yields a clear sense of 'pragmatic metavocabulary': both the digraph and the state-table vocabularies are VP-sufficient to specify practical abilities articulated as a finite-state automaton that is PV-sufficient to deploy—in the sense of recognizing and producing—the laughing Santa vocabulary, as well as many others. (Of course, it does that only against the background of a set of abilities PV-sufficient to deploy *those* vocabularies.) Perhaps surprisingly, it also offers a prime example of *strict pragmatic expressive bootstrapping*. For in this setting we can *prove* that one vocabulary that is expressively weaker than another can nonetheless serve as an adequate *pragmatic* metavocabulary for that stronger vocabulary. That is, even though one can*not say* in the weaker vocabulary everything that can be *said* in the stronger one, one can still *say* in the weaker one everything that one needs to be able to *do* in order to deploy the stronger one.

Here the relevant notion of the relative expressive power of vocabularies is also a purely syntactic one. Already in the 1950's, Chomsky offered mathematical characterizations of the different sets of strings of characters that could be generated by different classes of grammars (that is, in my terms, characterized by different kinds of syntactic metavocabularies) and computed by different kinds of automata. The kinds of vocabulary, grammar, and automata lined up with one another, and could be arranged in a strict expressive hierarchy: the Chomsky hierarchy. It is summarized in the following table:

| **Vocabulary** | **Grammar** | **Automaton** |
|---|---|---|
| Regular | A→aB<br>A→a | Finite State<br>Automaton |
| Context-Free | A→<anything> | Push-Down<br>Automaton |
| Context-Sensitive | $c_1Ac_2$→c1<anything>$c_2$ | Linear Bounded<br>Automaton |
| Recursively Enumerable | No Restrictions on Rules | Turing Machine<br>(= 2 Stack PDA) |

The point I want to make fortunately does not require us to delve very deeply into the information summarized in this table. A few basic points will suffice. The first thing to realize is that not all vocabularies in the syntactic sense we have been pursuing can be read and written by finite-state automata. For instance, it can be shown that no finite-state automaton is PV-sufficient to deploy the vocabulary $a^nb^n$, defined over the alphabet {a,b}, which consists of all strings of any arbitrary number of 'a's followed by the same number of 'b's. The idea behind the proof is that in order to tell whether the right number of 'b's follow the 'a's (when reading) or to produce the right number of 'b's (when writing), the automaton must somehow keep track of how many 'a's have been processed (read or written). The only way an FSA can store information is by being in one state rather than another. So, it could be in one state—or in one of a class of states—if one 'a' has been processed, another if two have, and so on. But by definition, a finite-state automaton only has a finite number of states, *and that number is fixed in advance* of receiving its input or producing its output. Whatever that number of states is, and whatever system it uses to code numbers into states (it need not be one-to-one—it could use a decimal coding, for instance), there will be some number of 'a's that is so large that the automaton runs out of states before it finishes counting. But the vocabulary in question consists of arbitrarily long strings of 'a's and 'b's. In fact, it is possible to say exactly which vocabularies finite-state automata (specifiable by digraphs and state-tables of the sort illustrated above) *are* capable of deploying. These are called the 'regular' vocabularies (or languages).

The next point is that slightly more complex automata *are* capable of deploying vocabularies, such as $a^n b^n$, that are not regular, and hence cannot be read or written by finite-state automata. As our brief discussion indicated, intuitively the problem FSAs have with languages like $a^n b^n$ is that they lack *memory*. If we give them a memory, we get a new class of machines: (non-deterministic[7]) *push-down automata* (PDAs). In addition to being able to respond differentially to and produce tokenings of the alphabetic types, and being able to change state, PDAs can *push* alphabetic values to the top of a *memory-stack*, and *pull* such values from the top of that stack. PDAs can do everything that finite-state automata can do, but they can also read and write many vocabularies, such as $a^n b^n$, that are *not* regular, and so cannot be read and written by FSAs. The vocabularies they can deploy are called "context free." All regular vocabularies are context-free, but not *vice versa*. This proper containment of classes of vocabularies provides a clear sense, suitable to this purely syntactic setting, in which one vocabulary can be thought of as "expressively more powerful" than another: the different kinds of grammar can specify, and the different kinds of automata can compute, ever larger classes of vocabularies. Context-free vocabularies that are not regular require more powerful grammars to specify them, as well as more powerful automata to deploy them. FSAs are special kinds of PDAs, and all the automata are special kinds of Turing Machines. Recursively enumerable vocabularies are not in general syntactically reducible to context-sensitive, context-free, or regular ones. And the less capable automata cannot read and write all the vocabularies that can be read and written by Turing Machines.

Nonetheless, if we look at *pragmatically mediated* relations between these syntactically characterized vocabularies, we find that they make possible a kind of *strict expressive bootstrapping* that permits us in a certain sense to evade the restrictions on expressive power enforced for purely syntactic relations between vocabularies. The hierarchy dictates that only the abilities codified in Turing Machines—two-stack push-down automata—are *PV-sufficient* to *deploy* recursively enumerable vocabularies in general. But now we can ask: what class of languages is *VP-sufficient* to *specify* Turing Machines, and hence to serve as sufficient *pragmatic*

7 By contrast to FSA's, there need not in general be, for every vocabulary computable by a non-deterministic PDA, some deterministic PDA that reads and writes the same vocabulary.

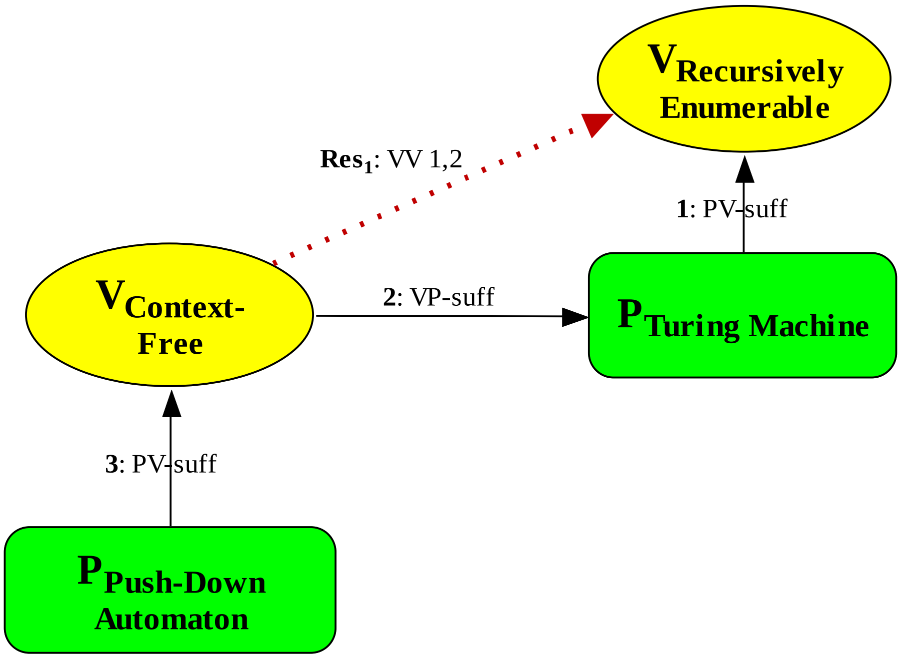
metavocabularies for recursively enumerable vocabularies in general? The surprising fact is that **the abilities codified in Turing Machines—the abilities to recognize and produce arbitrary recursively enumerable vocabularies—can quite generally be specified in *context-free* vocabularies**. It is demonstrable that context-free vocabularies are strictly weaker in syntactic expressive resources than recursively enumerable vocabularies. The push-down automata that can read and write only context-free vocabularies cannot read and write recursively enumerable vocabularies in general. But it is possible to *say* in a context-free vocabulary what one needs to be able to *do* in order to deploy recursively enumerable vocabularies in general.

The proof of this claim is tedious, but not difficult, and the claim itself is not at all controversial—though computational linguists make nothing of it, having theoretical concerns very different from those that lead me to underline this fact. (My introductory textbook leaves the proof as an exercise to the reader.[8]) General-purpose computer languages such as Pascal and $C^{++}$ can specify the algorithms a Turing Machine, or any other universal computer, uses to compute any recursively enumerable function, hence to recognize or produce any recursively enumerable vocabulary. And they are invariably context-free languages[9]—in no small part just because the simplicity of this type of grammar makes it easy to write parsers for them. Yet they suffice to specify the state-table, contents of the tape (or of the dual stacks), and primitive operations of any and every Turing Machine. Here is the MUD characterizing this pragmatically mediated relation between syntactically characterized vocabularies:

8 Thomas Sudkamp, *Languages and Machines,* 2nd ed. (Reading, MA: Addison-Wesley, 1997), Chapter 10.
9 In principle. There are subtleties that arise when we look at the details of actual implementations of particular computer languages, which can remove them from qualifying as strictly context-free.

**Meaning-Use Diagram #7:**

**Syntactic Pragmatic**
**Expressive Bootstrapping**

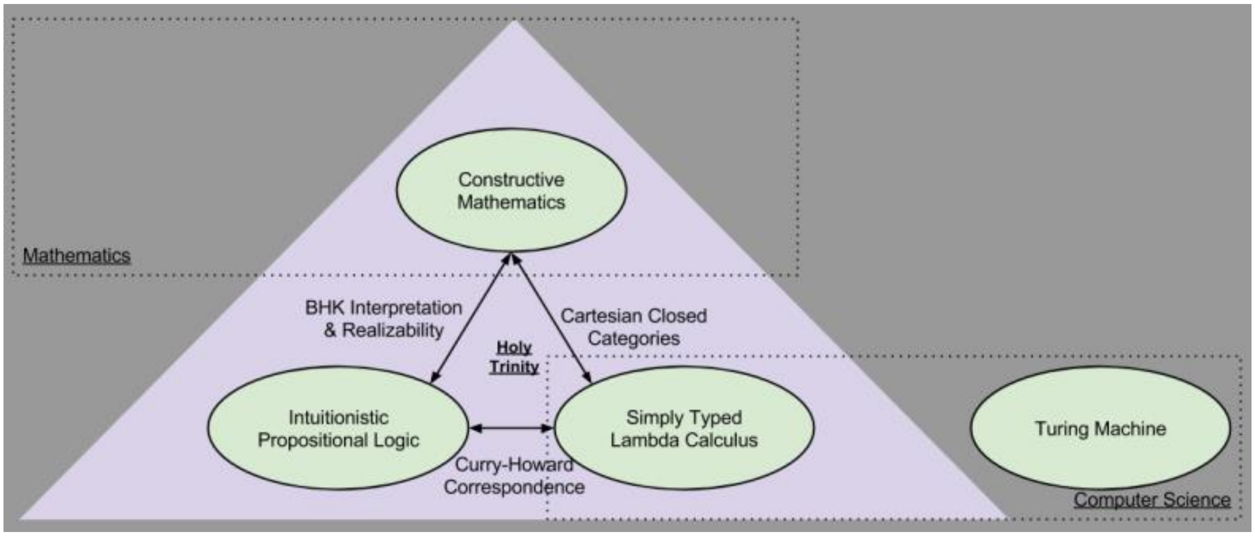



I called the fact that context-free vocabularies can be adequate pragmatic metavocabularies for recursively enumerable vocabularies in general ‘surprising’, because of the provable syntactic irreducibility of the one class of vocabularies to the other. But if we step back from the context provided by the Chomsky hierarchy, we can see why the possibility of such pragmatic expressive bootstrapping should not, in the end, be surprising. For all the result really means is that context-free vocabularies let one *say* what it is one must *do* in order to say things they cannot themselves say, because the ability to deploy those context-free vocabularies does not include the abilities those vocabularies let one specify. Thus, for instance, there is no reason that an FSA could not read and write a vocabulary that included commands such as “Push an ‘a’ onto the stack,”—and thus specify the program of a PDA—even though it itself has no stack, and could not *do* what the vocabulary it is deploying specifies. A coach might be able to tell an athlete exactly what to do, and even how to do it, even though the coach cannot himself do what he is telling the athlete to do, does not have the abilities he is specifying. We ought not to boggle at the possibility of an expressively weaker pragmatic metavocabulary having the capacity to say what one must do in order to deploy an expressively stronger one. We should just look to see where this seems in fact to be possible for vocabularies we care about, and what we can learn from such relations when they do obtain.

Point of this argument is to show that we can think formally about pragmatics (here, with automata models) *and* about the *relations* between pragmatics and semantics. Ulf's isomorphism is *another* example of formally characterizing those relations, appealing to the regimented pragmatic MV that is out topic next week.
Weeks 3 and 4 show another trajectory we can take from *BSD*1 to *BSD*6 (the latter in the form of the deontic/alethic isomorphism at the level of reason relations).

<u>Can end with</u>:

1. "*Computational trinitarianism*" is the discovery that functions, programs, and (logical—usually intuitionistic) *proofs* all provide perspectives on a common topic, and are related by a translation-supporting isomorphism.




https://kevinbinz.com/2015/06/26/computational-trinitarianism/

Robert Harper coined the term **computational trinitarianism** to denote this inter-referential characteristic of type theory, proof theory, and category theory.

("BHK" is Brouwer-Heyting-Kolmogorov).

Curry-Howard-Lambek isomorphism.

2. Technical Concept of Vocabulary, as lexicon plus set of pairs of sets of lexical items.

This is as precise a notion of 'vocabulary' as the syntactic one: subset (target) of universe generated from a finite alphabet (base).

Our claim, which must be defended, is that this is a (indeed, in some sense, *the*) *semantic* notion of 'vocabulary'. Or, if one wants to restrict 'semantics' not just to doing *model theory,* as opposed to *proof* theory, but also to *representational* model-theoretic semantic theories with substantial *metaphysical* commitments, paradigmatically PW or truthmaker semantics, then what we get is an account of *conceptual roles* (rational forms).