4 Code for-Itself

To Arendt, whatever can be known or experienced can only make sense in relation to speech and action, and to the human capacity for thought and thoughtlessness. In making the distinction between contemplation (*vita contemplativa*) and action (*vita activa*), she concludes that it is not possible to go through life without acting in it, whereas contemplation is optional. Her proposal is simple, "it is nothing more than to think what we are doing."[1] She distrusts science as a context in which "speech has lost its power," where a language of mathematical symbols has replicated spoken statements but cannot be translated back to speech.[2] These are the conditions in which thoughtlessness seems to have gained ground, and technology is rendering us ever more thoughtless.

Berardi's understanding of the problem is rather similar to Arendt's in his observation that we have been acquiring language from machines, not from other humans (namely, from our mothers), in situations where the learning of language and affectivity have been largely separated.[3] The problem of language acquisition is extended to intellectual and social behavior, and he calls this a catastrophe of modern humanism, in which we no longer have sufficient attention spans for love, tenderness, and compassion. We are more and more estranged from affect as speech becomes commodified, and is increasingly rendered using devices locked into neoliberal markets that have no interest in the voice as a sign of human solidarity.

Things are rather differently described in early coding cultures. In Hayles's *My Mother Was a Computer*, she charts how in the 1930s and 1940s, people, mostly women, were employed to do calculations and were referred to *as* computers (Hayles appropriates her title from a chapter in *Technologies of the Gendered Body* by Anne Balsamo, whose mother was one of these computers).[4] Kittler's *Discourse Networks* also refers to how the mother's voice haunts reading, making reference to the period around 1800 when children were taught using phonetics, to sound out and voice words. By 1900, he observes that distinctions were drawn between noise and signal.[5] This further detail helps to explain Berardi's lament on the conditions for the teaching of speech and the disconnection from the body, and importantly serves to stress how the voice of

the computer connects the construction of particular kinds of subjectivity and learning practices that use informational systems. It might be said that what is lacking is closer attention to nurture and the articulation of human feeling that cannot be expressed by words alone.

If the voice also haunts computing in this way, it interpellates us in new ways that affect our thinking processes, intellectual capacities, and abilities to express empathy with others. It is these kinds of interactions of code and language that also interest Hayles, as artificial languages proliferate and in recognition of the acknowledged cultural influence of human languages on constructions of subjectivity. Her concern is that programming languages are too easily dismissed as artificial and of lesser consequence.[6] In technologically advanced cultures, language and program code interact all the time in complex ways, and even mothers and computers can become confused in their assigned normative functions. Central to this approach is that the understanding of code (as of speech and writing) is constituted ideologically (through what Hayles calls a "worldview"). Like speech, program code is active in the world and has a lived body, indeed is intimately connected to a social body. The issue that has occupied the book thus far is about control of that body, its expurgation, and related processes of autonomy activated through coding practices.

Execution

The human capacity to speak and act is enduring, as Arendt has stated, even when language itself is used as an apparatus of power against it. This is partly because language is paradoxical, and holds an innate ability to transform itself as well as those who are constituted through it. The relation between speech and action has been explored in previous chapters: to speak is to actualize something, announcing the intention to produce an action in the moment of doing so. It is also clear that language has effects, as it acts for us and against us, and this is what Butler has referred to as "linguistic vulnerability," confirming the unerring power of language to affect us from our beginnings, as with the case of the Althusserian call to order as a form of violence on the human subject.[7] Butler's point is that violence is embodied in language, not simply in the way it might be used to incite a violent action or in the ways that language reflects social domination more generally, but in the way that it produces meaning. This relates to Hegel's observation that there is something inherently violent in the capacity of language to represent a thing, what he calls "its essencing ability," which is equivalent to its symbolic death. As it stands in for something, "it dismembers the thing, destroying its organic unity," and forces the thing into a field of meaning that is outside of itself.[8]

This also happens with source code and perhaps in a more extreme manner, as code exceeds natural language through its protocological address to humans and machines.

It says something and does something at the same time—it symbolizes and enacts the violence on the thing: moreover, it *executes* it. In addition to these symbolic and physical forms of violence, there is also something that comes close to "pure violence" (in the sense of Benjamin's essay "Critique of Violence" of 1921), action that is directed not to other human beings but to the symbolic powers and operating systems that reign over them.[9] Pure violence appears to come from nowhere, as "an expression of pure drive, of the undeadness, the excess of life, which strikes the 'bare life' regulated by law."[10] One might speculate that code might similarly express pure means through collective actions like DDoS attacks, SQL code injection techniques,[11] the spread of viruses, directed at the sovereign technical infrastructures that already exert forms of violence on clients through the enforcement of restrictive terms of service and the like.[12] The hacktivist tactics of Anonymous, mentioned in the previous chapter, or of LulzSec, the splinter group who have been "Laughing at your security since 2011!," exemplify such ways of thinking.[13]

When Virno confirms how language radicalizes "aggression beyond measure,"[14] he is drawing on Aristotle's description of contingency at the heart of our use of language. Besides the enduring capacity to speak and act, his interest is in the ability of the human species to execute "innovative actions" that he likens to recursive plays of language.[15] Underpinning political possibilities, for Virno, is the simple fact that the human animal is capable of modifying its forms of life, of innovating new forms.[16] This is what enables and produces innovative action, in the sense that newly invented forms might diverge from established rules and perceived or consolidated norms (based on Chomsky's idea of innate creativity previously discussed, although Virno prefers not to use the term creativity, as it has become so instrumentalized through the creative economy agenda). He cites the example of jokes, as demonstrations of the ways that "linguistic animals give evidence of an unexpected derivation from their normal praxis."[17] Rather than taking jokes to be Freudian clues to the workings of the unconscious, he regards them as examples of sociolinguistic games that demonstrate innovative techniques and possibilities for transformation. He explains that this happens in two main ways: first, by demonstrating how divergences in following rules often result in changing the rule itself; and second, through the incorrect use of semantic ambiguity.[18] Of course his point is not the content of the jokes, which might of course be political (and yet perhaps counterproductive in serving to obscure or normalize the issue), but the linguistic apparatus or the "*logicolinguistic resources* that jokes utilize."[19] He characterizes this sense of linguistic innovation as "how to do new things with words" (after Austin once more), in which doing something relies on public action and also "presupposes and revives a public space," thus reiterating earlier references to linguistic-communicative performances that necessitate a public space.[20] So, to Virno, innovative utterances are similar to collective speech acts where speech constitutes action in and of itself, as potential speech in-itself.

The argument relies on the recognition that innovative action uses linguistic and performative resources in similar ways to jokes, and is thus able to intervene in the workings of contemporary capitalism because language has been absorbed at a structural level into the political economy. Behind the possibility of innovative action is the enduring ability of language to create unexpected relations between multiple speakers, as speech is necessarily shared and collective. Language constitutes what Virno calls a "pure institution" (before and beyond the law), as it underwrites all other institutions, and emphasizes that the human animal is ready-made for language but only enters into language through socialization.[21] To Virno, this confirms the biopolitical dimension of the human animal in the world, and the ability to act in unexpected and innovative ways to challenge institutional norms and presuppositions of normative logic. Moreover, if the subject is to some extent constituted in language, then to think that someone saying and doing something is a straightforward demonstration of agency misses the point that actions are always already encoded, like the innate ability to produce sounds from the body, however abstract they may seem. The historical subject is always ready to speak and act, whether conscious or not yet conscious of the need to do this. It is in recognition of this fact that the public is always able to act for-itself as a body politic, however constrained the conditions may appear to be.

Negation

When Virno describes "not-yet public forms of government,"[22] he is pointing to the ways in which the public is always ready for collective action in this sense. They are able to construct innovative forms of self-organization by negating received organizational forms, similar to the way a recursive public is able to modify the means through which it is constituted as a public (discussed in the previous chapter). To Virno, this possibility of reinvention is explained through the interplay of innovation and negation inherent to language. In contrast to representational democracy, for instance, he refers to *non*representational forms inspired by a nondialectical understanding of negation. Like negative feedback, a determinate negation becomes a constructive influence on the system as a whole, allowing it to self-regulate.[23]

To clarify the concept of negation a little more, the distinction needs to be made between mere difference (something is not something else) and the more fundamental claim that something is not something else but depends on it to exist. This is the basis of dialectical logic, where the role of negation, and its further negation (negation of negation), become important for understanding some of the ways in which dominant ideas attempt to reproduce themselves, even when an oppositional stance is taken. This is why it has to "die twice," as Žižek puts it (in tarrying with the negative), in order to reject its symbolic confines.[24] But this is not quite what is

meant by Virno, who recognizes that neoliberalism is a negative condition that requires further negation, not through negative dialectics but on the basis that negation is embedded in the paradoxes of language. For him, it is the determining role of language that needs negating to protect the possibility of a reciprocal *non*recognition: "the implicit presupposition of rhetorical persuasion and, in general, of the permanence of a public sphere."[25] His reading of the system of language reveals its inherent paradox in the way it represents, as it both does negation (by identifying what something is not) and is negation (inasmuch as it can only signify something): "The negation, or something that language *does*, is understood, above all, as something that language *is*."[26]

The concept of negation appears to operate like a speech act too, countering the authoritative call to order and opening up alternative possibilities outside the "sovereign autonomy of speech."[27] Moreover, Virno speculates on forms of politics that recognize the ways that sovereign forces try to restrain these inherent capacities for innovation or social transformation, as with the illusion of free speech as a mechanism of power. This leads him to conclude that self-government needs to adapt itself directly to the "linguistic aspect of the human species."[28] Therefore it becomes necessary to produce paradoxes between the determinism of grammatical rules and the ways in which speech is always to some extent out of control (something Butler previously identified). This is in recognition that the linguistic-communicative aspects of capitalism have become central to its structural logic, as well as its potential reinvention. If both negation and innovation are constituted in language, like human subjectivity, then any sense of agency afforded to it is also constrained and activated in reciprocal relation.

The contingency at the heart of this is explained neatly by Žižek, who describes the illusion of free choice through the notion of interpellation and how it chooses its subjects. He refers to the "vulgar liberal notion" of freedom of choice as a "fundamental choice by means of which I 'choose myself'."[29] *Forced choice*, on the other hand, is explained as the subject freely choosing the inevitable, such as the historical subject's recognition of class consciousness (class in-itself). The subject recognizes itself as encoded, "always-ready" for action, and only in this way can begin to act freely. The subject is thereby called by history to act in the way it should act and take the right course of action. In Žižek's explanation and in parallel to the perceived determining role of language, this is not ideological manipulation at all as it is already programmed in advance and there is simply no choice to be made. This holds for speech, as it preexists itself; it is speech before speech, or speech in-itself. Things are decided before they are enacted and they act on us, not the other way around. But if this is the case, Žižek asks, are we simply turned into computers or thinking machines,[30] or input-output machines as Laporte suggests? Is it that human subjects are preprogrammed and merely execute their preprogrammed instructions and scripts?

If human action is largely rendered ineffective under current conditions, perhaps this is because language has become instrumentalized by machine logic, and thereby disconnected from human feelings. Virno concludes: "For political anticapitalist and antistate action there is no positive presupposition to be vindicated. Its eminent duty is to experiment with new and more effective ways of negating negation, of placing 'not' in front of 'not human'."[31] Should we not do the same with the figure of the programmer and with programming in general, remove it from the determining conditions of the market that strip it of human fallibility and the possibility of innovative action? Like freedom of choice, is it only possible to begin to think of free software as a result of the self-consciousness of conditions in recognition of it being programmed in the first place? If so, then there really is no real choice at all and all software can be considered to be free at source. Code is always ready to execute, in the move from in-itself to for-itself, and it seems verified that information wants to be free.[32] Code can express freedom only inasmuch as it is able to execute the right course of action.

Coda

By indicating that language is taught by computers and not by another caring human, Berardi points to the way that human expression and social attention have become overtly economicized (as part of the so-called "attention economy").[33] Like fast speech, he thinks interpretation has become schizophrenic, and that the relations between metaphors and things, representation and life, have become thoroughly confused, leading to the conclusion that "a hyper-stimulation of attention reduces the ability to critically and sequentially interpret the speech of the other who tries and yet fails to be understood."[34] The inability to produce collective speech acts leads to tragic consequences in terms of the human psyche, according to Berardi, as of course language acts on the construction of subjectivity itself. This separation of language from affect leads to a situation in which this excess of signs, obsessive accumulation, and accelerated communications leave little time for love, tenderness, and compassion, or even for contemplation, as Arendt lamented.

If we no longer have sufficient attention spans for free thinking or love, then perhaps we simply haven't recognized that both have been there all along. Again an example from Žižek, quoting Bertrand Russell, helps to clarify the point: "I did not know I loved you till I heard myself telling you so. . . ."[35] In other words, love preexists the knowledge of it. A similar paradox underlay Berardi's reading of the source code of the "I Love You" virus (which spread through communities of the Internet in 2000).[36] Although seemingly declaring love, the message "love letter for you" if opened erased documents from your hard drive and then propagated itself by sending new

copies of itself through the address book of your mail program.[37] Berardi effectively negates the negation by turning it into spoken poetry.

In another example from the history of computing, the generative "love-letters" that first appeared on the notice board of Manchester University's Computer Department in 1953 are similarly revealing. Predating the chatterbot *Eliza*, these computer-generated declarations were produced by a program written by the programmer Christopher Strachey, using the built-in random generator of the Manchester University Computer (the Ferranti Mark I), the earliest programmable computer (first functioning as a prototype in 1948). Artist David Link reconstructed a functional replica of the hardware and the original program, following meticulous research on the functional aspects but also speculating on why the programmer may have decided to generate love letters at all. The main program is relatively simple, using loops and a random variable to follow the sentence structure: "You are my — Adjective — Substantive," and "My — [Adjective] — Substantive — [Adverb] — Verb — Your — [Adjective] — Substantive."[38]

Some words are fixed and some optional, indicated by the square brackets; the program selects from the list of options—adjectives, adverbs, and verbs—and loops are configured to avoid repetition. The software could generate over 318 billion variations. In terms of effect, the dialogic structure is important too in setting up an exchange between "Me" (the program writer) and "You" (the human reader), such that you feel addressed directly—it *interpellates* you.

The resultant declarations suggest a surprising tenderness of expression that runs contrary to what we consider the standard functional outcomes of computational procedures (for commerce or war). Here is an example:

```
DEAR DARLING
    YOU ARE MY BEAUTIFUL RAPTURE. MY INFATUATION BEAUTIFULLY CLINGS TO
YOUR ADORABLE LUST. MY INFATUATION LUSTS FOR YOUR WISH. MY AMBITION
CURIOUSLY LIKES YOUR LOVE. YOU ARE MY DEAR EAGERNESS.
                                                    YOURS WISTFULLY
                                                        M. U. C.[39]
```

If computers could speak freely, is this what they would say? Is love reducible to a "recombinatory procedure"?[40] Surely not. On the one hand, as Link points out, it seems to portray a reductionist view of love, but on the other, love is also characterized by projection, and in this way fires the imagination. He explains this by quoting Goethe, who "once made the cynical suggestion that love-letters should be formulated in a completely cryptic way, so that the recipient could project whatever she liked into the text."[41] Moreover, cryptology aside, is the love letter reducible to the problem of memory like the Universal Machine, to be written, read, stored, and deleted like any other data? Once more, it is worth being reminded that people are not simply

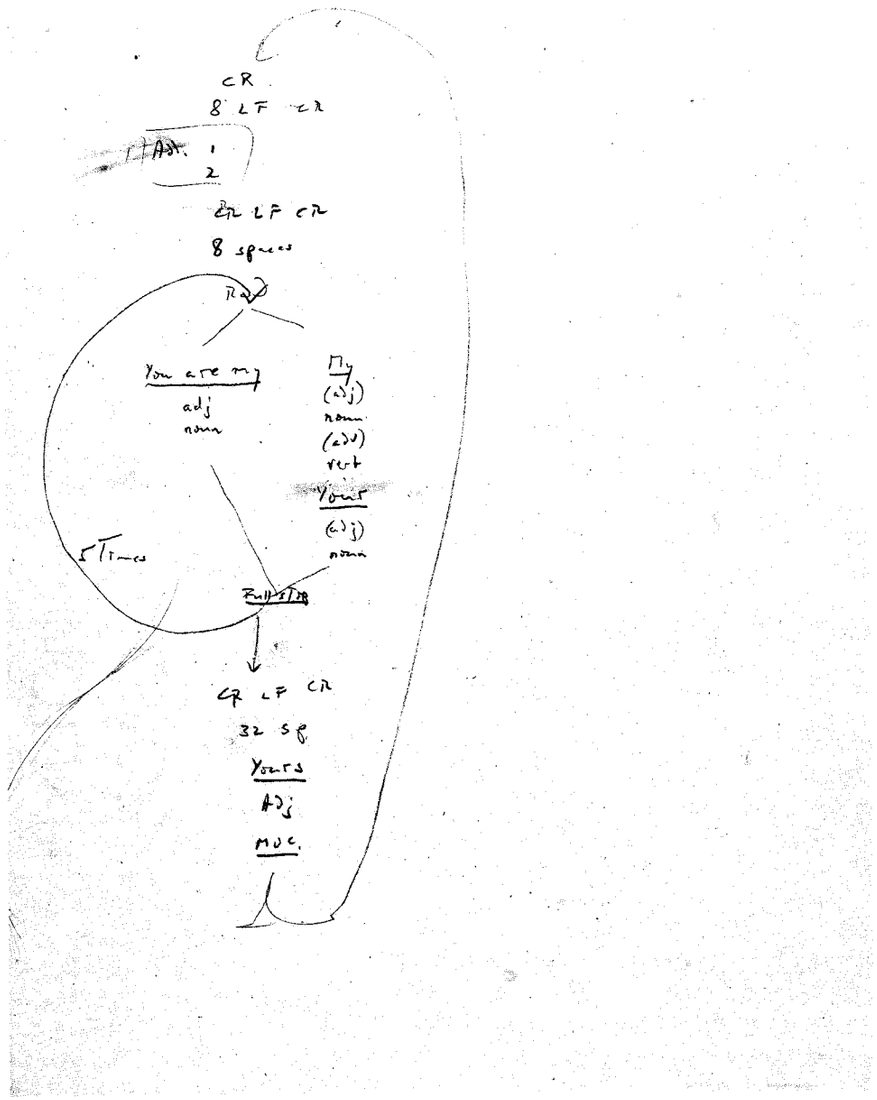

Figure 4.1
Schematic of Christopher Strachey's love letters program (1953). © The Bodleian Libraries, University of Oxford, CSAC 71.1.80/C.34.

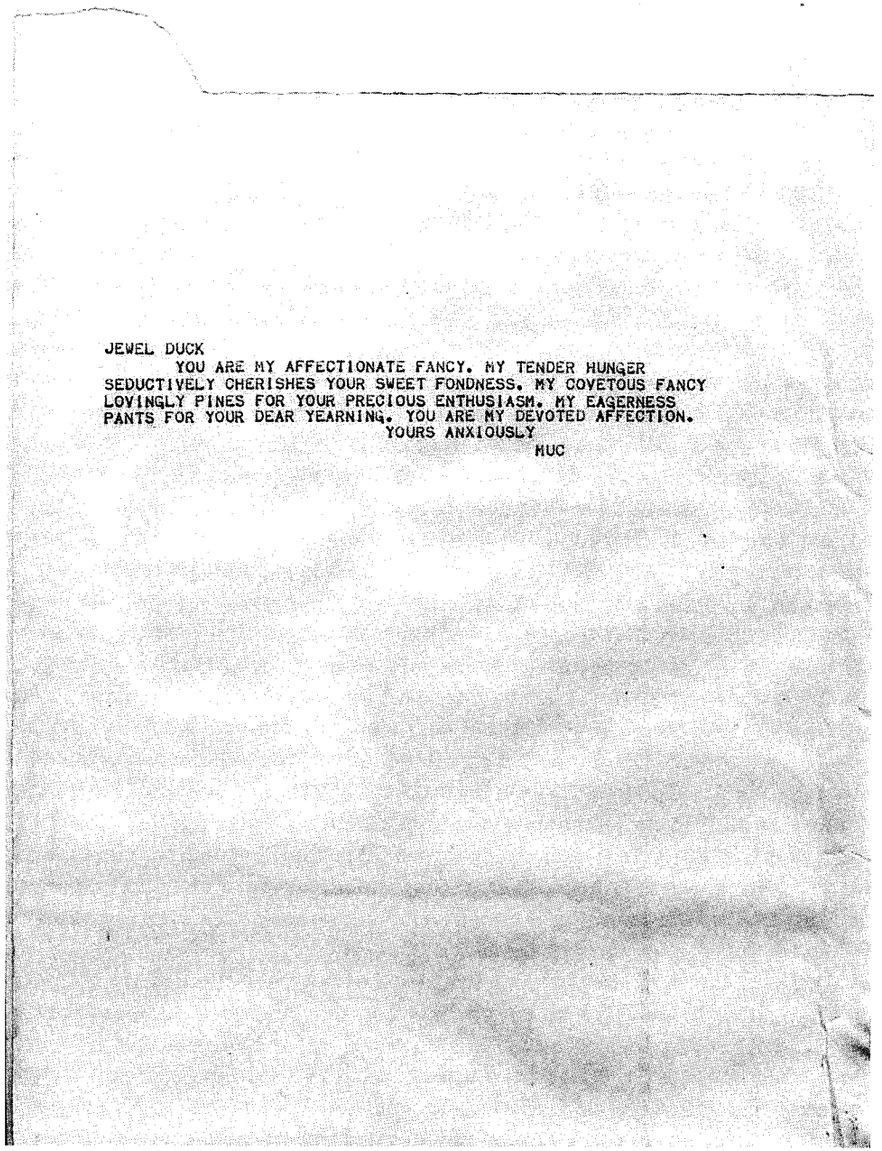
JEWEL DUCK
YOU ARE MY AFFECTIONATE FANCY. MY TENDER HUNGER
SEDUCTIVELY CHERISHES YOUR SWEET FONDNESS. MY COVETOUS FANCY
LOVINGLY PINES FOR YOUR PRECIOUS ENTHUSIASM. MY EAGERNESS
PANTS FOR YOUR DEAR YEARNING. YOU ARE MY DEVOTED AFFECTION.
YOURS ANXIOUSLY
MUC

Figure 4.2
Image from David Link's *Loveletters_1.0* (2009). Image courtesy of David Link.

determined by such histories but are also involved in its very construction: both programmed and able to program.

If love and tenderness are indeed neglected, as Berardi indicates, then we haven't recognized that love is preprogrammed, and that attempts to pervert this "truth" ultimately fail to interpellate. Indeed if running algorithms is a kind of "truth telling," as Link argues elsewhere,[42] then recognition holds a far more positive message than the hallucinations imposed by neoliberal markets, and the increased prominence of privatized technologies that deny access to source code. If human experience is ever more prescribed through scripts and programs like this, and which we have less and less access and attachment to, then the challenge for those making programs is to open up aesthetic and political possibilities of recombination and to liberate the imagination and desire from the market. Thankfully humans are not reducible to computational logic, but perhaps more importantly love was there all along, even with computer programs. But with little time for this realization, humans have become more and more distanced from the ability to communicate intimately with others.

Speech appears to have lost its power, along with the efficacy of human solidarity. The biopolitical dimension of this is key, as it addresses the ability of subjects to have a voice, one that connects to the expression of opinions in public and that is tied to subjective expression. Together these conditions operate as guiding concepts for contemporary politics and for the ability to think and act in the world with any degree of thoughtfulness or effectiveness. Dolar explains how in Aristotle's *Politics*, the political is defined in terms of the distinction between mere voice (*phoné*) and speech (*logos*), the former common to all animals including the human animal, but the latter distinguishing humans from other animals in their ability to articulate judgments in association with others. This distinction (as Dolar further explains) is what Agamben is also referring to with his opposition between "bare life" (*zoe*), life in common with animals, and life in the community (*bios*), the commons and political life.[43] Each of these elements, however, is reciprocally embedded in the other and not simply external to it. Referring to Schmitt's view of sovereignty and the rule of exception, Agamben explains this as "the condition of being excluded through an exclusion, of being in relation to something from which one is excluded."[44]

For Dolar, this is an invitation to extend the analogy and think of the inclusion and exclusion of the voice in speech. Like bare life, the voice is both included and excluded in the political realm. This is a voice that is connected to politics and the essence of life itself, both within and beyond politics. It also connects usefully to the biopolitical dimension of technology in the development of speaking and thinking machines, or of telecommunications platforms where the voice is paradoxically included and excluded at the same time. Dolar neatly takes this to represent the Hegelian move from "in-itself," in the case of the speaking machine, to "for-itself" in the thinking machine.[45] The voice stands as a figure for something that does

not entirely compute and resists its capture, while retaining the possibility of free thinking.

What seem to be required are dynamic recombinations of speaking, thinking, and coding. To enable this, forms of totality have to be rejected, whether related to a view of the historical subject or to a view of program code that is deterministic or totalitarian, thus rupturing the relations between being programmed and programming. If the autonomy of intellectual labor from economic rule can alone save us from semiocapital, as Berardi states, then it needs recombining with the voice *and* code in recognition that both are always ready for action and at the same time ready to run out of control (like a live-coding performance). For it is this sense of incompleteness that drives transformational agency, and the ways in which human subjects retain the ability to modify their lived circumstances knowing their experiences to be incomplete. The recognition of the choice of action, already programmed but perhaps not knowingly so, confirms that both subjectivity and code recursively write their own instrumentation. Yet the subject is not simply preprogrammed like a machine but more like code in actively combining internal and external factors, standing between what is possible and what actually exists. Extending the move from in-itself to for-itself further, collective and networked intelligence open up the conditions of possibility for reinvention by embracing broader contingencies, to challenge overpowering forces that wish to close them down, encapsulate and subsume them.

If lived experience is ever more prescribed through scores, scripts, and programs, then the challenge for those making program scripts that underscore these procedures is to open up aesthetic and political possibilities of recombination and free the imagination to further use. Thus the performativity of code, in live coding or code acts, demonstrates the potential for collective intelligence and effective action. It proposes coding practices that have not only a body but also a body politic.

```
#!/usr/bin/python

# (C) alex@siab.org
# Please edit and share.
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.

import time, os, MultipartPostHandler, urllib2, cookielib
import libtorrent as lt

fs = lt.file_storage()
lt.add_files(fs, __file__)
t = lt.create_torrent(fs, 65536)
t.add_tracker("udp://tracker.openbittorrent.com:80", 0)
```

```
lt.set_piece_hashes(t, '.')
tf = lt.bencode(t.generate())

fh = open('me.torrent', 'wb')
fh.write(tf)
fh.close()

info = lt.torrent_info(lt.bdecode(lt.bencode(t.generate())))

opener = urllib2.build_opener(
    urllib2.HTTPCookieProcessor(cookielib.CookieJar()),
    MultipartPostHandler.MultipartPostHandler
    )

opener.open("http://runme.org/submit/",
            {'act': 'login',
             'email': 'torrentpy@mail.slab.org',
             'password': 'hjrvrs'
             }
            )

opener.open("http://runme.org/submit/",
            {'explicit_project_id': '974',
             'act': 'submit_software',
             'file_software': open('me.torrent')
             }
            )
ses = lt.session()
h = ses.add_torrent(info, "./")

state_str = ['queued', 'checking', 'downloading metadata',
             'downloading', 'finished', 'seeding', 'allocating']

while (1):
        s = h.status()
        print '%.2f%% complete (down: %.1f kb/s up: %.1f kB/s peers:
%d) %s' % (s.progress * 100, s.download_rate / 1000, s.upload_rate /
1000,
            s.num_peers, state_str[s.state])
        time.sleep(5)
```

Notes

0 Double Coding

1. For more on the Befunge programming language, see http://en.wikipedia.org/wiki/Befunge.
2. *The Hello World Collection*, compiled by Wolfram Roesler (with help from many people around the world), includes 421 "Hello world" programs in many more or less well-known programming languages, plus 63 human languages (available at http://roesler-ac.de/wolfram/hello.htm).
3. Referring to John 1:1: "In the beginning was the Word, and the Word was with God, and the Word was God."
4. Jonathan Rée, *I See a Voice: Language, Deafness and the Senses—A Philosophical History* (London: HarperCollins, 1999), 75–76. He is citing a passage from Genesis 11:1–9. Also of relevance here is the constructed language Volapük, created in 1879–1880 by Johann Martin Schleyer, who claimed that God had told him in a dream to create an international language. See Leo Findeisen, "Some Code to Die For: On the Birth of the Free Software Movement in 1887" (2003; available at http://www.monochrom.at/codetodiefor/).
5. Although the relationship between the study of machines and religious thinking is not the concern of this book, it is worth registering that Norbert Wiener identified commonalities in that machines must learn and reproduce in accordance with what he called the rules of the game, a game increasingly set by the dark forces of informational capitalism and the industrial-military complex. See Norbert Wiener, *God and Golem, Inc.* (Cambridge, MA: MIT Press, 1964).
6. "Twitspeak" is the vernacular form of language used with Twitter, such as the use of hashtags (see http://twitter.com/).
7. The Bodyfuck demo shows the physical exertion required to produce a short sequence of symbols (available at http://forum.openframeworks.cc/index.php?topic=2772.0).
8. Friedrich Kittler, "There Is No Software," in Timothy Druckrey, ed., *Electronic Culture* (New York: Aperture, 1996), 331–337 (available at http://www.ctheory.net/articles.aspx?id=74). Matthew Fuller's "It Looks Like You're Writing a Letter," in *Behind the Blip: Essays on the Culture of Software* (New York: Autonomedia, 2003), makes a similar reading through a close analysis of the word-processing software Microsoft Word, with a further link to the politics of work through the Office suite of programs.
9. Louis Althusser, "Ideology and Ideological State Apparatuses: Notes Toward an Investigation" (1969), in Slavoj Žižek, ed., *Mapping Ideology* (London: Verso, 1997), 131. To explain, "ideological