On the Concept of a Token Generator

Takashi IIDA[*]

Abstract

There is a widely shared account of the distinction between types and tokens, which might be termed the standard account. However, it has some surprising consequences that are not always realized. According to the standard account, a type is a contingent abstract object that can be created by us, but it does not allow any change and can never be destroyed once it is created, because it is an abstract object. I would like to present an alternative account of types and tokens, according to which types are concrete objects that are located in space and time. This new account is based on a concept that I call a "token generator", which is something that specifies in detail how to produce the tokens.

Key words: type and token, abstract and concrete, token generator

0. Outline

This paper is concerned with the distinction between types and tokens. There is a widely shared account of this distinction, which I shall call "the standard account", according to which a type is an abstract object in the sense that it is not in space or time, while its tokens are concrete objects or events which exist in a particular place and time. The standard account, however, has some surprising — we can even say "startling" — consequences that are not always realized.

(1) A type is a contingent abstract object.
(2) A type does not allow any change.
(3) We can create a type, which is an abstract object.

[*] Department of Philosophy, College of Humanities and Sciences, Nihon University. E-mail: iida.takashi19@nihon-u.ac.jp
This paper is based on my talk given at Tokyo Forum for Analytic Philosophy (TFAP), Nov.29, 2012. I profited very much from many comments and questions on the occasion. I wish to thank the audience and Richard Dietz, who invited me to give a talk at TFAP. I am indebted to Kengo Miyazono for making me realize that I ought to say much more about how a type should be distinguished from a universal than I had done. I wish to thank Lajos Brons for having done a through job in checking the manuscript, which greatly improved the present version.

(4) Once created, a type can never be destroyed.

I once argued that despite their bizarreness it was not unreasonable to accept all of these consequences ([Iida 2011]), but in this paper I would like to present an alternative account of types and tokens and discuss its consequences. This new account is based on a concept that I call a "token generator"[1]. A token generator of a type is something that specifies in detail how to produce its tokens, like a detailed specification of a commercial product or a score of a musical composition. Additionally, it may be some person or thing that embodies such a specification like the speaker of a particular language.

1. Types and tokens: the standard account

Here is an example that illustrates the distinction between types and tokens.

無
無

Looking at this, one speaker says

(A) There are two characters here.

Another speaker says

(B) There is only one character here.

Although what these speakers say seems to be incompatible with each other, both of them may be right. The appearance of incompatibility arises only from the ambiguity of the word "character"; the first speaker uses it in the sense of tokens, while the second speaker uses it in the sense of a type.

A token of a character type is a concrete entity located at a particular time and place, while a type is not bound to any particular time or place. We can use the same character type repeatedly at different times and places, similar to a tool.

It should be noted that a token is necessarily a token of some type or other. The concept of a token is essentially relational in that it is derived from a relation

x is a token of y.

Although types constitute an ontological category of its own (or, so I am going to argue), tokens do not. Of course, they are concrete entities, but tokens can be either physical objects or events, which belong to different ontological categories.

[1] This does not mean that the concept of a token generator does not have a place in the standard account. On the contrary, I believe it is indispensable for the standard account as well if it is to do full justice to our type-token talk.

The type-token distinction is applicable to a variety of things that can be classified in at least three main kinds.

1. Mass-produced commodities such as computers and cars. Tokens of this kind are produced in quantities according to a certain specification. Various breeds of domestic animals and varieties of plants might be included in this class as well.
2. The elements that consititute a language or a tonal system like words and tones. Types and tokens of this kind are related to each other in a systematic way and are of different complexities.
3. Poems, novels, songs, and symphonies. The existence of types and tokens of this kind depends on the entities of the second kind.

You may notice that the things I have just listed are all artifacts that are invented by us, but the type-token distinction has been applied to other kinds of things as well[2]. It has been claimed frequently that a species and an individual of that species are related as a type and its token. It has been common, for example, to hear a philosopher of mind discuss whether the correspondence between a mental event like a pain and a physical event like a change in the brain as a relation between types or between tokens. Alhough I am not yet decided whether we should say that a species is a type whose tokens are individuals that belong to that species, I am pretty convinced that the use of type-token terminology in the contemporary philosophy of mind is loose or metaphorical, and strictly speaking, we should not talk about a mental event type or a mental event token.

As shown by the fact that there are twenty-six letters in the Latin alphabet and each of them has a name, we can count type entities and name them. There is a definite number of Chinese characters first graders in Japan should learn in a year. These Chinese characters can be named in various ways; for example, you can get a Chinese character on your computer screen by typing its character code, which might be regarded as its name. Furthermore, types can be in the extension of a common noun like “letter of the Latin alphabet” and “Chinese character”. Thus, types can be counted, have their own names, and be in the extension of a common noun.

However, these characteristics might not be enough to distinguish a type from a universal like whiteness. You may argue that the colors can be counted, have their own names like “whiteness”, and are in the extension of a common noun, namely, “color”.

A reply to this would be that a particular type is introduced into a discourse only

[2] In her book [Wetzel 2009] Linda Wetzel enumerates the following examples of kinds of things to which the type-token distinction applies: linguistic entities like a phoneme, word and sentence, mental events/states/processes, works of art, actions, biological species, genes, artifacts like computers, chess moves, physical fields and particles ([Wetzel 2009], Chapter 1).

by means of a singular term, while a universal is typically introduced into a discourse through a predicate or a general term like a common noun. If you would wish to talk about a particular word or sentence, you would use an expression like "the word *snow*" or "the sentence *The snow is white*", which are singular terms. Similarly, we would use such singular terms such as "West End Blues" and "War and Peace" if we would wish to talk about a particular piece of music or a particular novel. If we follow Frege and hold that an object is what a singular term stands for, then types like words, sentences, musical tunes and novels are all objects (Cf. [Dummett 1978, 42]). In this respect, types differ from universals like whiteness which can be introduced into a discourse by a predicate like "is white".

Although this works well for art works belonging to the types in (3) above and the complex types in (2) like sentences, it does not help us to differentiate other kinds of types from universals in general. For example, consider the following sentence.

I drive a Toyota Carolla Axio.

Here the name of a car model, which is supposed to be a type, appears as a part of a common noun whose extension consists of the tokens of that model. And a teacher who is teaching children how to write of the Latin alphabet might point to two scribbles on the blackboard and say

This A is much better than that A.

Here "A" should be construed as a common noun[3].

I am afraid that we should conclude that linguistic clues alone cannot determine the boundary between types and universals, and that we need to bring in some other kinds of considerations to do so. Still, I believe that the kinds of things listed above constitute the core cases of types; they are artifacts with some common pattern which can be produced and reproduced at different times and places[4]. If we lift the restriction to artifacts, then a species would count as a type and its tokens would be the individuals belonging to that species, because the individual of the same species are reproduced again and again.

How far should we go? Could we talk about a mental event type *pain* or an action type *promise*? I am reluctant to count them as the genuine cases of the type-token distinction. Philosophers of mind talk about pain as a type and pain as a token, and

[3] It might be argued that what the teacher said was an abbreviation of "This token of the letter A is much better than that token of the same letter", and that the unabbreviated sentence contains only a singular term that refers to the type. But I suppose that such a response might seem to beg the question.

[4] As I read him, Richard Wollheim's usage of "type" seems to be very close to mine. According to him, we postulate a type when "we can correlate a class of individuals with a piece of human invention" ([Wollheim 1980, 78]).

moral philosophers may talk about promise as a type and promise as a token, but is there any good reason to suppose that there is some object which is called "pain as a type" or "promise as a type"? I don't think anything is lost if the type-token distinction is replaced by the good old universal-instance distinction. What is meant by "token" identity between a mental event and a physical event is just the identity between some individual mental event and some physical event, while "type" identity between them is meant to describe the circumstance that for any individual mental event that is an instance of a certain mental kind (universal) such as *pain*, there is an individual physical event that is an instance of a certain physical kind (universal) such as *C-fiber firings*, and that is identical with the mental event and vice versa. It cannot be denied that this is much more cumbersome than the usual talk of token identity and type identity, and therefore, it is understandable that such talk is very popular. If you do not wish to commit yourself, however, to the existence of pain or promises as self-subsistent objects, you should have in mind that such a talk is just an abbreviation of a more cumbersome way of talking involving only universals and their instances.

2. A type is a contingent abstract object

Whatever their exact characterization might turn out to be, type entities are objects. Moreover, as they are supposed to be not in space or in time, they are abstract objects.

It is now universally accepted that, in general, types cannot be reduced to tokens. Take one of the letters of the Latin alphabet, "A" for example, and consider the immense variety of different shapes, sizes and styles of the tokens of "A". We can find no characteristic common to all these different tokens that does not mention the very type itself. Rather, we should say that the only property those entities have in common is that they are all tokens of the type "A" ([Wetzel 2006] 4.2.3).

Although types cannot be reduced to tokens, the existence of a type depends on the existence of its tokens. In the second kind of type entities, there are complex types such as very long sentences which have no tokens, but if such sentences would be regarded as existent, they would consist of the existing words, and those words should have tokens if they exist. Hence, in general, the existence of a type depends on the existence of some tokens. If those tokens do not exist, then the type does not exist.

For each indvidual token, it is a contingent matter whether it exists or not. Therefore, it is also a contingent matter whether a type whose existence depends on that of tokens exists or not. This means that type entities are contingent abstract objects. When we talk about abstract objects, we usually talk about numbers and pure sets, which are considered to be necessary beings, but there are also contingent beings among abstract objects. This is the first surprising consequence of the standard account of the type-token distinction. Type entities are beyond space and time, but they are not beyond the world in which they exist.

3. A type does not admit any change

There are some other surprising consequences of the standard account of type entities as abstract objects whose existence depends on the existence of tokens. The second is that type entities do not allow any change, because a change is only possible for something that exists in time, but a type does not exist in time.

This seems to raise serious difficulties in the case of languages, because we talk of a language as if it is constantly changing, but a language is a system consisting of abstract objects like phonemes, words, and grammatical rules.

There are many examples of language change; for example, a word may become pronounced or spelled differently, or it may acquire a new meaning. If a word is an abstract object that does not exist in time, it cannot undergo such changes. For that matter, nothing can happen to an abstract object except some extrinsic changes such as becoming somebody's favorite word .

One way of coping with this difficulty is to claim that so-called language change is not a change in a language but a change in people's linguistic behavior; people have started to use a slightly new language instead of the old language. This means that a language change is really a replacement of one language by another[5] .

4. Creating an abstract object

Let us move to another counter-intuitive consequence, namely, that we can create new type entities, which are abstract objects. Why do we find this claim incredible? It might be because we subscribe to a principle like the following.

(C) If X is created at time t, then X does not exist before t.

However, a type entity is an abstract object, and an abstract object always exists. Therefore, given the principle (C), a type entity cannot be created, as there is no time in which it does not exist.

I am inclined to say this is nothing but a muddle; a confusion of the existence beyond time with the existence throughout time. That an abstract object does not exist in time does not mean that it always exists. So, the principle (C) is false for an abstract X.

Or we should say that the principle (C) does not apply to an abstract X because

[5] This is exactly what I argued for in [Iida 2009].

a tensed existential predicate in (C) does not apply to an abstract object. When we talk about existence, we usually use tensed existential predicates like "existed", "exists" and "will exist". But they are not appropriate for abstract objects, which do not exist in time. Instead, we may use "real" and "unreal" or "real" and "imaginary". These predicates apply to both temporal existents and non-temporal existents.

Among type entities, some are real because their tokens exist, and some are unreal because they have no tokens. Just like some types would not be real if somebody had not created them by producing their tokens, some types would be real if somebody had produced their tokens. These are possible but unreal type entities that failed to be created.

5. The impossibility of destroying an abstract object

A type is real insofar as there are some past, present, or future tokens. Hence, even if one of our words now in use will disappear completely in the future, and thus some future generation will have no clue as to its existence, the word will not become an unreal word. This is an example of the last of the consequences of the standard account mentioned above, namely,

(4) Once created, a type can never be destroyed[6].

A principle similar to (C), namely,

(D) If X is destroyed at time t, then X does not exist after t,

does not apply to an abstract X. Therefore, it is a mistake to argue that (4) is true because there is no time when X does not exist because X is an abstract object. But then the situation seems to be quite symmetrical to the creation of a type, which raises the question: Why is creation possible while destruction is impossible in case of an abstract object?

In case of concrete things, the difference between creation and destruction might be expressed by the difference between the following two counterfactuals.

(C1) Were it not for my action, X would not exist now.
(D1) Were it not for my action, X would exist now.

However, they do not yet capture the concepts of creation and destruction. There are two different cases for each counterfacutual. In (C1), X exists now, but X either did not exist before my action or did exist before my action. Similarly, in (D1), X does not exist now, but either X did not exist before my action or did exist before my action.

[6] This is explicitly endorsed in [Shapiro 2000, 262].

(C1-a) If X did not exist before my action, my action created X.
(C1-b) If X did exist before my action, my action prevented the destruction of X.
(D1-a) If X did not exist before my action, my action prevented the creation of X.
(D1-b) If X did exist before my action, my action destroyed X.

If we would use such a contrast for abstract objects, we should use "real" instead "exist".

(C2) Were it not for my action, X would not be real.
(D2) Were it not for my action, X would be real.

Are there also two possibilities for each of these? Let us consider (C2) first. It might seem that there are two possible situations before my action. Either (a) a token of X did not exist before my action, or (b) it did exist before my action. In the case (a), I create a type X by producing its token by my action, but in case (b), X is already real, and my action does not make X real; my action cannot make X unreal, either, because if something is real it cannnot be made unreal. Hence, case (b) is impossible. In other words, my action can create a type, but once created, my action cannot have any influence on its being real.

How about (D2)? There are two cases to be considered. These are case (a) in which a token of X did not exist before my action, and case (b) in which a token of X existed before my action. But, if a token of X existed before my action, then X is already real and none of my action can have anything to do with its being real. Therefore, the only possibility is (b), that is, a token X did not exist before my action. In this case, my action prevents the creation of X by preventing the creation of any token of X.

In other words, we can either create an abstract object or prevent its creation, but we cannot make an unreal object real nor a real object unreal.

6. An alternative account of type existence

We have seen that the standard account of type-token distinction has the following consequences.

(1) A type is a contingent abstract object.
(2) A type does not allow any change.
(3) We can create a type, which is an abstract object.
(4) Once created, a type can never be destroyed.

Some of these consequences may be considered too bizarre, suggesing a search for an alternative to the standard account. Particularly, I suspect that (2) and (4) are most objectionable for many people, because they seem to be contrary to our usual way of talking. In fact, we frequently talk about a type entity as if it has a temporal existence; for example, we talk about various kinds of type entities which used to exist but no longer exist now. They might be particular types (or models) of a car which are no longer in production, certain words which are no longer used, or musical compositions which were lost forever long time ago.

Hence, an obvious idea is to think of a type entity as a temporal being. If a type entity would be a temporal being, then it would be contingent, allow change, and could be created and destroyed. A thought that immediately comes to mind would be something like this:

(TE) A type begins to exist when its first token is produced, and ceases to exist when its last token ceases to exist.

If we adopt this, then we will be able to avoid the consequence of the standard account that a type never ceases to exist once it is created. However, this idea, though obvious, faces some serious difficulties.

Suppose a token of a type α is produced at time t_1 and ceases to exist at time t_2. Further suppose that we are now at time t_3 which is later than t_2, and between t_2 and t_3 no tokens of α were produced. Does the type α still exist now? The answer depends on what is going to take place in the future; if there will be a time t_4 later than t_3 when a further token of α will be produced, then α exists now, while it does not exist now if there will be no further token of it in the future.

Looking at such a situation, we can see there are two sorts of problems.

First, the possibility of such a situation reveals that there would be some propositions about the present state of the world that become true in the future. In other words, what will happen in the future may decide what is the case now. And this means that what happens now may decide what was the case in the past. Suppose that in the above situation I produced a token of α at time t_4. Then, this action of mine makes true the proposition that the type α existed at time t_3, whose truth value was not determinate at t_3. This means that my present action may influence the past. It has also the consequence that what exists at any time is not determinate unless it is presupposed that determinism is true, because whether the type α exists at t_3 depends on whether another token of α is going to exist after t_3. In general, what type entities exist now is not determinate, unless determinism is true.

Although this is rather strange, it may not be an incoherent position. There is, however, another problem, which is epistemological. Suppose that after a token of α is produced at time t_3, we have not encountered any further token of α. Then can we conclude that the type α ceases to exist? Obviously, the answer is no. We cannot be sure that there won't be another token of α in the future. We can never know that the last token we have encountered is in fact the last token of the given type. This means that, once a type is created, we cannot know whether it still exists or not. Hence, as far as the consequence (4) of the standard account is concerned, epistemologically there is no real choice between the present account and the standard account.

It should be concluded that this alternative account of type existence is not satisfactory, or at least not in the present form. The next question is whether it can be improved such that it conceives of types temporal entities without any untoward consequences like the above.

7. The concept of a token generator and the causal continuity of a type entity

In the above discussion, we imagined a situation in which a token of a given type appears after a certain period of time since a prior token appeared. There is something suspicious about imagining such a situation. Suppose the period between a new token and a prior token is very long; what makes them tokens of the same type?

Is it a similarity between them? The following example shows that even exact similarity is not enough.

> Suppose some poet today composes a haiku. But it turns out that the single sentence that constitutes her haiku is exactly the same as the sentence that constitutes a haiku composed by another poet who lived 200 years ago. Suppose further that the modern day poet did not know this fact and has never encountered the older haiku in any form before. Is the haiku composed by her the same as the older one?

We should say that the modern day poet uttered or wrote the same sentence as the poet of 200 years ago had done, but they composed different haikus; their products are of the same type considered as a sentence, but they are different haikus.

Why is this? They are different haikus, because they have different and causally unrelated origins. This suggests that the causal origin is essential for the identity of a type entity.

If we look at some examples more closely, we will find that there is always some sort of causal connection between the different tokens of the same type[7]. Let me give two examples.

(a) A token of a word in Japanese should be causally related to some word

[7] It should be noted that a causal connection is only a necessary condition for two things to be tokens of the same type. If we wish to know the sufficient condition, a separate discussion is needed for each kind of type, which belongs to the different fields of philosophy such as philosophy of economics, philosophy of linguistics and aesthetics.

tokens that were uttered by speakers of Japanese.

(b) What makes a particular car a token of a particular type of car is that it owes its existence to the causes it has in common with the other cars of the same type. Among such common causes are manufacturers and designers.

It becomes obvious that some causal connections are necessary for two objects to be the tokens of the same type, if you compare the two cases in the following example.

(Case 1) A new plant is to be set up to start a production line for a certain car model. The new production line is built by the company that has produced such models in another plant, and designed after the production line of that plant.

(Case 2) A newly formed company has set up a plant and started a production line for a certain car model. This model is designed by the company's engineers from scratch. However, by pure coincidence, the car that comes out of the production line happens to be exactly the same in both appearance and internal mechanism to the car that has been produced by another company whose existence is unknown to the new company.

In (Case 1), the cars which come out from the new production line will be of the same type as the cars which has been produced in another plant, but in (Case 2) the relation between the cars produced by the new company and those from another company is the same as that between two haikus in the example above. Even though you cannot tell them apart just by looking, they are of different types because their causal origins are completely independent from each other.

The causal connections among the tokens of the same type are frequently mediated by something that is not one of these tokens. Imagine a variant of (Case 1):

(Case 1′) A new plant is to be set up to start a production line for a certain car model, whose production was discontinued many years ago. Moreover, no car of that model still exists now. In spite of this, it is possible to restart the production, because the company still owns all the documents about that model, including its specification and detailed plans.

Like the original (Case 1) and unlike (Case 2), the cars that come out of this production line are of the same type as the cars that were produced many years ago. It is because there clearly is a causal connection between the cars produced many years ago and the cars produced now through those documents that have made it possible to resume production in spite of the absence of a real specimen of the same model.

There are many other cases in which a causal connection between the different tokens of the same type is established by some sort of mediator. The relation between the performances of a musical composition and its score gives us a particularly clear example. If we think that only actual performances of the piece constitute the tokens of a musical composition, then its score is not one of its tokens but something else; it is something that gives us detailed instructions for producing the tokens, namely, the actual performances of the composition.

Let us call this sort of thing which specifies in (sufficient) detail how to produce tokens of a given type "token generators". A detailed plan for making a certain model of car is a token generator of that model and a musical score of a symphony is a token generator of that symphony. A token generator of a given type is something that makes it possible to produce its type's tokens if you follow its instructions. A token generator of a commercial product should be something that contains the complete and detailed specification of that product, while a token generator of a musical composition or a play should be something that determines its performance in sufficient detail. A mere sketch for a musical composition or a rough description of an idea for a product cannot be a token generator[8].

We might extend the concept of a token generator to include that what is not itself a specification for a token but that embodies such specifications. There are many tales and songs which came down through oral tradition; they were never recorded in any form and preserved only in the memory of those who could tell or sing them. Such people may be regarded as token generators of a particular tale or song; it is not by having some conscious set of instructions but by mastering a skill that they can tell a tale or sing a song.

Various kinds of recordings are other examples of token generators. The recorded performance of a piece of music gives rise to a token of that piece of music when it is played back. Therefore, such a recording can be regarded as a token generator of that piece of music[9].

The most interesting instance of a token generator in this extended sense is a speaker of a particular language. Of course, it needs a detailed argument to claim that the speaker of a language is a token generator. As I am not sufficiently prepared to give such an argument right now, all I can do now is to suggest a possible way to do it.

First, although a language consists of type objects like phonemes, words, and sentences, it is not itself something which has a token; we cannot imagine what a token of Japanese could be. So, strictly speaking, a speaker of a language could be a token generator of words and sentences of the language, but not of the language itself. Secondly, if a person is a token generator of a word or a sentence of a certain

[8] A token generator of a type α might be itself a token of another type β. This fact is worth investigating, but I am not going into it now.

[9] There is an interesting discussion about the relation between a performance and its recording in [Kurata 2012].

language, she should also be a token generator of other words and sentences of that language. As there are supposed to be infinitely many phrases and sentences in a language, she should be a token generator of infinitely many phrases and sentences. How could such an infinite capacity be embodied in a single person?

According to one familiar line of thought, a complete description of a grammar of a language can be used as a finite device to generate an infinity of phrases and sentences of the language, and we might regard a speaker of the language as an embodiment of such a grammar. It goes without saying, however, that it has been hotly debated whether such a view has any validity. Still it is a promising idea worth exploring, I believe.

8. Type existence redefined

From the above consideration, we may conclude that the tokens of the same type must be causally related to each other. In some cases, they are directly related; in other cases, they are indirectly related through the token generators of the type in question.

With the concept of a token generator, what could we say about the temporal existence of a type?

First, it is almost obvious that a token generator of a type sometimes precedes any of the tokens of the type. Think of any composition of Western classical music; although a composer might be hearing her music in her mind while she writes down its score, there is nothing but a token generator of her piece at that time. Its tokens in the form of actual performances come later. Should we say that her piece begins to exist only when the first performance is given? I suppose the answer is no; we would say that her piece already exists before the performance.

It might not be generally true, however, that a type comes into existence with its token generator even if a token generator precedes acutal tokens . For example, we might not say that a certain car model exists now because we have its very detailed plan; it seems that the model begins to exist only after the first actual car is made. Hence, it depends on the kind of type entity whether the existence of a token generator prior to any tokens is sufficient for the existence of the relevant type.

Secondly, a type exists even in the period in which none of its tokens exist, if a token generator of the type exists in that period. Again the musical example gives the best illustration. No one thinks that a piece of music ceases to exist in between one of its performances and the next. (Case 1′) above suggests that this is true in the case of a car model as well.

Finally, I would like to claim that the type ceases to exist when no tokens exist anymore and there are no token generators either. As an example, consider some commercial product, such as a brand of bottled soft drink that was once popular a long time ago. Suppose that not one bottle of this brand is left now and nothing is known about its ingredients or its recipe. In these circumstances, many would say that this brand of soft drink once existed but no longer exists now. It is not so easy, however, to argue that a type ceases to exist once all of its tokens and token generators are gone in general. To be able to do that, we need to establish that the temporal existence of a type must be continuous.

Here is a very rough sketch of an argument for this thesis. As we saw with some examples, if two tokens of the same type exist at different times, there must be some causal connections between them. We may suppose that such causal connections could be broken down ultimately to simple steps, which are causal interactions between either (a) two tokens of the type in question, (b) a token and a token generator of the type, or (c) two token generators of the type. For a causal interaction to occur between two items, there should be no instant at which both items do not exist. Hence, if a token of a certain type exists at t_1 and another token of the same type exists at t_2 ($t_1 < t_2$), then throughout the period between t_1 and t_2 there must exist at least one token or one token generator of the same type.

Of course, this is very rough and I know there remain many unclarities and some insufficiently supported assumptions, but if I am allowed to suppose that th line of thought in the above argument is not far from the truth, then we may conclude that the existence of a type requires the existence of its token or its token generator as a necessary condition. Thus, we can revise our former claim about type existence in the following way[10].

(TE2) A type exists only when its token or token generator exists.

Let us see whether this new definition is free from the difficulties the original definition had.

Now that a type exists only if its token or token generator exists, we know that a type no longer exists if we have made sure that none of its tokens or token generators exist at present. Although there is a general problem of how we can be sure that certain things do not exist, at least we need not wait to see what will happen in the future in order to decide whether a type exists now or does not. And, I believe that the problem of the knowledge of non-existence does not arise in this case, because we need not search the entire universe but only some limited region in which causal interactions among those tokens and token generators may take place.

[10] As was noted above, it depends on the kind of type entity whether the existence of a token generator is sufficient for that of a type itself. Thus, a stronger claim than (TE2), namely,

(TE2′) A type exists when and only when its token or token generator exists.

is valid for some kinds of type entities, but not valid for others.

Consequently, according to our new definition, there is no difficulty in principle to infer the non-existence of a certain type from the evidence that its tokens or token generators are no longer found now.

Moreover, this accords well with our way of talking; we say that a certain composition of a famous composer is lost now because the original and all the copies of its score were destroyed, or that some earlier car model no longer exists and will not exist because none of its specimen or blueprints exists now and the engineers who made it died long ago.

Talk of a dead language might be different from these cases. Why do we say that a language is dead if there are no speakers left? In some cases, even though nobody speaks the language, we still have the records of the language in the form of written documents or recordings. In such cases, the language is not completely destroyed. It seems that we say a language is dead for at least two reasons; a language is dead because it is no longer used in actual communication between people, and it is dead because there is no possibility of change. Hence, a dead language is not a lost language. A lost language in the strict sense would be a language about which we know almost nothing. There must be many such lost languages that were destroyed in the course of our history.

Although we do not usually talk about destroying a type, it is obvious we can do that by destroying all its tokens and token generators. Consequently, unlike the standard account of types and tokens, the present account makes it possible for us to destroy a type as well as create it.

9. Type existence as spatial and temporal

We have made a type a temporal being that begins its existence at some time, continues to exist for a while, and then ceases to exist at last. We meet a type at one time and meet the same type again at another time, just as we meet a person at one time and meet the same person again at another time. In the same way that we know that the very person we see now exists at another time, we know that the very type we now meet in the form of its token or token generator might have existed at some time in the past and might exist at another time in the future. It is a part of our understanding of the concept of a type that it is a recurring entity. We meet the same type again and again whenever we come across with its various tokens or token generators. In this respect, a type is not like an event, but like a thing as a physical object or a person.

A type is not only a temporal entity; it is a spatial entity as well. If you see the same word ocurring twice on a page of your book, then that word exists at two different locations in space.

Hence, a type has spatial location as well as temporal location.

(TE3) A type α exists at time t at location l only if a token of α or a token generator of α exists at time t at location l.

There seems to be an obvious objection to making a type a spatial entity. It is based on the following principle. ("DD" stands for "Disjoint locations imply Difference".)

(DD) For any time t and any object x and y, if the location of x at time t and that of y at the same time t are disjoint from each other, then x and y are different objects

Or, if we designate the location of x at time t by "$l(x,t)$", then this principle can be expressed more concisely.

(DD) For any time t and any object x and y, if $l(x,t)$ is disjoint from $l(y,t)$, then $x \neq y$.

This principle is used when, for example, you judge that, however similar two coins in front of you are, they are different coins because they have different locations at the same time. In such a situation, we cannot do without this principle, but think of the same word appearing twice on a page of your book. How can they be the same object —I suppose a word as a type is an object— and occupy two disjoint locations at the same time? Hence, to conceive a type as a spatial entity obviously contradicts to the principle (DD).

To this objection, I reply that the principle (DD) is not valid as it stands. Although it is true for physical objects, it is not true for concrete objects in general. Here I take a concrete object to be simply an object that exists in space and time. It is part of our understanding of the concept of a type that a type can occur at different places even at the same time. When we see a word token and recognize it as a token of a particular word type, we know that the same word may occur at that instant in various places where its tokens exist.

In other words, there may be no single continuous location a type α occupies at t, and therefore, the $l(x,t)$ in (DD) may be undefined for a type α. This is not any different from the possibility there is not any single continuous period $p(x,l)$ for a location l and a physical object x. Although a physical object may be found at the same place at different times, there may not be any single continuous period in which it continues to be located there.

The principle (DD) presupposes another principle ("SL" stands for "single location")

(SL) For any time t and any object x, there is a single continuous location $l(x,t)$ where x occupies at t.

We saw that this prinicple is not true with a type object. We also saw that a similar principle (SP), which stands for "single period", is not true for objects in general.

(SP) For any location l and any object x, there is a single continuous period $p(x,l)$ during which x is located at l.

Hence, we might ask ourselves which fact needs explanation, the fact that a type object does not satisfy (DD), or the fact that a physical object does satisfy (DD). It might well turn out that what needs explaining is the latter.

At any rate, it is one of the major characteristics distinguishing type entities from physical objects that the principle (DD) does not apply to type entities. According to our ordinary way of speaking, which is sometimes called "three-dimensionalism", a physical object like my computer exists "as a whole" at every instant of the period of its existence. That is, at each instant what exists of my computer is not its instantaenous part, but my computer as a whole with all other temoral parts. With a type entity like a word, whenever we come across its token, we also encounter the type "as a whole", that is, with all its tokens existing at different times and different places.

A natural reaction to such a conception of a type entity would be to ask why a type cannot be identified with someting like a mereological sum of all its tokens and token generators. However, I will show that a simple identification of a type as a mereological sum of all its tokens and token generators does not succeed.

The reason why it does not work is that the same object or event can be a token of different types at the same time. Consider this.

無

This object (or event?) can be a token of at least three different types.

1. a Chinese character.
2. a Japanese word.
3. a line of a poem.

As it is almost certain that, if we list all the tokens of each of these three types, these lists differ from each other, we can distinguish them by taking the mereological sums of all the tokens of each type (*i.e.* items on the list), but obviously it is possible for two different types to have exactly the same objects or events as their tokens or token generators.

Suppose a poet writes a poem that consists of a single sentence of Japanese. Further suppose that this sentence is perfectly all right grammatically, but it happens that its only occurrence is as the single line of the poem. In this case, all the tokens of the poem and those of the Japanese sentence that constitutes the poem are the same, and hence, there is no difference between the mereological sums of all the tokens of each type.

Of course, the Japanese sentence which is the single line of the poem might have been uttered or written not as a line of a poem, and the mereological sums of their tokens might have been different. This thought, however, can not be of much help here.

If we wish to take up this idea, a natural way to do so is to identify a type with something like a function mapping each possible world to the mereological sum of the tokens and token generators of a given type that exist in that world. Obviously, such a function is not a concrete object; it is a typical abstract object. Hence, it cannot be created nor destroyed.

Of course, there might be some other way to use the modal intuition as above without making a type an abstract object. However, until such a way is found, we should reject the idea that a type is the same as the mereological sum of all its tokens.

10. Concluding remarks

In conclusion, let me sketch the general picture that has emerged from the alternative account of types and tokens that I have been trying to develop here.

According to this account, a type is not an abstract object, but a concrete object that exists in space and time, though a type is different in some important respects from other kinds of concrete objects such as physical objects and persons. One of the differences is that a type can exist at different places at the same time. Moreover, it is not the case that different parts of the type exist at different places, but that the type itself "as a whole" exists at each of the different locations. We might talk about spatial parts of a given type, but this is similar to talk about "temporal parts" of a physical object or a person.

A type exists continuously in time. At any given time it might exist in discontinuous locations, but as a type owes its identity to the causal connectons between its tokens and token generators, each location the given type occupies must be linked by various continuous paths that connect its tokens and token generators. Hence, a certain continuous region can be assigned to a type as the part of space that it occupies during its whole history, in the same way a certain period of time can be assigned to it.

I would like to emphasize that what I have been proposing here is to develop an account of types and tokens that is an alternative to the standard account, and try to see what it involves. Although I pointed out that the standard account has some striking consequences that might be contrary to our intuition, it is a well-known fact that our intuition is not always reliable. In order to make the final judgment, we need to know much more about what each account involves. I have just indicated a general outline of an alternative account, and there remain many problems that need to be confronted before it can be judged. In particular, the following seems important to me.

(1) Once a type becomes a temporal entity, there is no longer any reason to think a type does not admit any change. What will be an account of the change a type may undergo?

(2) There are still many unclarities about what a token generator is. In particular, in what sense a speaker of a language is a token generator? I believe this question has a vital importance to the philosophy of linguistics.

(3) As we remarked above, a language is not a type; only its elements are types. If we wish to construe a language change as a genuine change, not as a replacement of one language with another in people's linguistic practice, then we should be clear what a language change consists in, in particular, what the exact relation between a language and its elements is.

References

[Dummett 1978] Michael Dummett. *Truth and Other Enigmas*. Duckworth, 1978.

[Iida 2009] Takashi Iida. "How are language changes possible?" in Mitsuhiro Okada (ed.), *Ontology and Phenomenology: Franco-Japanese Collaborative Lectures*. Keio University, Open Research Centre for Logic and Formal Ontology, 2009. pp.75–96.

[Iida 2011] Takashi Iida. "On the contingency of some abstract objects" a paper read at *Inter-ontology 2011* held at Keio University on Feb. 24th, 2011.

[Kurata 2012] Tsuyoshi Kurata. "Geijutsu sakuhin no sonzai-ron: bunseki-teki keijijyo-gaku no tachiba kara (Ontology of art - from the standpoint of analytic metaphysics)" (in Japanese). in Nishi-nihon-tetsugaku-kai (ed.), *Tetsugaku no Chousen*. Shumpusha Publishing, 2012, pp.167–209.

[Shapiro 2000] Stewart Shapiro. *Thinking about Mathematics*. Oxford University Press, 2000.

[Wetzel 2006] Linda Wetzel. "Types and tokens". *Stanford Encyclopedia of Philosophy*. http://plato.stanford.edu/entries/types-tokens/ (Last accessed Dec. 20, 2012).

[Wetzel 2009] Linda Wetzel. *Types and Tokens*. The MIT Press, 2009.

[Wollheim 1980] Richard Wollheim. *Art and Its Objects*. Second Edition. Cambridge University Press, 1980.

(Received 2013.1.14; Accepted 2013.2.4)