Behaviorally correct identification

Nina Gierasimczuk Dick de Jongh

March 8th, 2013

In *behaviorally correct identification, BC-identification* of languages (previously called extensional identification) the success criterion is not converging to a unique index for the target language but stabilizing to an arbitrary series of indices for the target language.

Example 1 *$\{K \cup D \mid D$ finite$\}$ is effectively BC-identifiable.*

Proof $\{K \cup D \mid D \text{ finite}\}$ is BC-identified by M defined as follows:
$M(\sigma) = U(k, content(\sigma))$ where k is a code for K. □

Of course, $\{K \cup D \mid D \text{ finite}\}$ is not effectively identifiable, since $\{K \cup \{x\} \mid x \in \mathbb{N}\}$ isn't.

Theorem 2 (Baliga, Case, Smith) *A uniformly recursive $\mathcal{L}$ is effectively BC-identifiable iff $\mathcal{L}$ has telltale sets.*

Proof From left to right is obvious, since effectively BC-identifiable implies identifiable, and by the non-effective Angluin theorem we then have telltale sets.

Suppose that r.e. indexed and uniformly recursive $\mathcal{L}$ has telltale sets D_i for each L_i. To prove is that there exists a function M that BC-identifies $\mathcal{L}$.

We have to define M on an arbitrary σ, but since M needs to work properly only in the limit and only on texts for some L_i we assume a lot about σ, and we don't care what happens before that assumption is justified.

Consider a σ that is part of some text for L_i such that L_j unequals L_i for each $j < i$. We can assume that D_i is part of σ, and write n for $max(lh(\sigma), content(\sigma))$.

We look at L_j with $j \leq n$ restricted to $\{0, \ldots, n\}$, notation $L_j^{(n)}$. Consider the $j < i$ such that content(σ) is in L_j, i.e., in $L_j^{(n)}$. This then means that L_j is not a proper subset of L_i: either L_j properly contains L_i, or neither is contained in the other. We assume in addition that if L_i contains an element not in L_j, then there is already one in σ. Hence the only L_j with $j < i$ that contain σ are such that $L_j^{(n)}$ contains (properly or improperly) $L_i^{(n)}$.

Of course, if we consider σ we do not know the i, but after we have restricted our attention to those L_i such that $L_i^{(n)}$ contains σ, we do know that we can exclude any i such that for some $j < i$, $L_j^{(n)}$ contains σ, but $L_i^{(n)}$ contains an element not in $L_j^{(n)}$.

This means that, for any σ, we can restrict our attention to $P_n = \{i \leq n \mid L_i^{(n)}$ contains σ and for all $j < i$ such that $L_j^{(n)}$ contains σ, $L_j^{(n)}$ is a superset of $L_i^{(n)}\}$. Clearly, this set is linearly ordered by the converse inclusion relation. The proper i is somewhere in there, but we do not know where. We have not restricted the behavior of the $j > i$ sufficiently.

To determine the value of $M(\sigma)$ we fix the n and enumerate a language in stages s with at stage 0 the situation we just described. **This enumeration determines the index = the value M assigns to σ.** Of course the index will have to be an index for the right language L_i.

The last one of the series of languages in P_n, $L_{j_{max}}$ is the smallest one and we cannot go wrong by starting the enumeration in stage 0 with $L_{j_{max}}^{(n)}$. In the stages $s > 0$ we still investigate only the languages $L_1, \ldots, L_n$, but now restricted to $n + s$ instead of n. We set

$P_{n,0} = P_n$ and

$P_{n,s} = \{i \leq n \mid L_i^{(n+s)}$ contains σ and for all $j < i$ such that $L_j^{(n+s)}$ contains σ, $L_j^{(n+s)}$ is a superset of $L_i^{(n+s)}\}$.

Again this set is linearly ordered. Each time we take the last element j_{max}^s. The language $L_{j_{max}^s}$ is the smallest language still in the running at stage s. Of course, it contains (or =) the last language of the previous stage. We add in the enumeration the elements in $L_{j_{max}^s}$ that have not been enumerated before.

To see that the right language is enumerated we just have to see that from a certain point on $j^{s_{max}}$ will be equal to i (or at least will designate the same language). If any L_j with $j > i$ is a language different from L_i that contains D_i, then it will contain an element not in L_i. Ultimately we will discover this element in $L_j^{(n+s)}$ and j will be excluded from $P_{n,s}$. So, we will end up with L_i.

Note that the procedure is only BC, because the enumeration is dependent on the starting point n, which comes from σ. Note also that it is essential that the language is enumerated as an r.e. set. It is not possible to give an index of the characteristic function of the recursive set. It seems that effective BC-identification collapses to standard effective identification if one requires indices of the characteristic function of the recursive set. $\square$