IAS/PCMI Summer Session 2000
Clay Mathematics Undergraduate Program
Basic Course on Computational Complexity

Lecture 5: The Landscape of Complexity Classes

David Mix Barrington and Alexis Maciel
July 21, 2000

1. Our Current Knowledge

In this first week of lectures, we have defined a number of complexity classes. In this lecture we try to relate them to each other.

In the world of machines, we have that

$$\mathrm{L} \subseteq \mathrm{NL} \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE} = \mathrm{NPSPACE} \subseteq \mathrm{EXP}.$$

The inclusions $\mathrm{L} \subseteq \mathrm{NL}$ and $\mathrm{P} \subseteq \mathrm{NP}$ are trivial since deterministic machines can be viewed as special kinds of nondeterministic machines. The equality $\mathrm{PSPACE} = \mathrm{NPSPACE}$ is a consequence of Savitch's Theorem and the inclusion $\mathrm{NL} \subseteq \mathrm{P}$ can also be obtained by that same idea of reducing to graph reachability. The fact that EXP contains PSPACE, and therefore NPSPACE, was pointed out just before the proof of Savitch's Theorem. In the lecture on nondeterminism, we showed that $\mathrm{NP} \subseteq \mathrm{EXP}$. The inclusion $\mathrm{NP} \subseteq \mathrm{PSPACE}$ can be established by examining the space requirements of that simulation.

In the world of circuits, we have

$$\mathrm{NC}^0 \subseteq \mathrm{AC}^0 \subseteq \mathrm{NC}^1 \subseteq \mathrm{AC}^1 \subseteq \mathrm{PSIZE}.$$

The bounded-depth classes can be generalized as follows. Define AC^i to be the class of languages decidable by circuit families of depth $\log^i n$ and polynomial size, with no restriction on the fan-in. NC^i is defined as the fan-in two version of AC^i. In addition, let $\mathrm{AC} = \bigcup_{i \geq 0} \mathrm{AC}^i$ and $\mathrm{NC} = \bigcup_{i \geq 0} \mathrm{NC}^i$. It should be clear that AC and NC are actually equal so we get the following:

$$\mathrm{NC}^0 \subseteq \mathrm{AC}^0 \subseteq \mathrm{NC}^1 \subseteq \mathrm{AC}^1 \subseteq \cdots \subseteq \mathrm{NC} = \mathrm{AC} \subseteq \mathrm{PSIZE}.$$

What we want to do next is relate to each other the machine and the circuit classes. One key point will be that of the *uniformity* of the circuits. Without any uniformity condition, that is, by simply defining a circuit family without providing any means for actually constructing the circuits, it is possible to define simple circuit families that decide languages that are not decidable by any machine whatsoever. To avoid this situation, we will require that circuits be uniform, in the sense that they can be somehow constructed.

We define uniformity more precisely by saying that a circuit family is uniform if there are machines that can answer the following questions:

1. Given an input length n and a gate number i, what is the function computed by gate i?

2. Given an input length n, a gate number i and a number k, what is the number of the kth input of gate i?

The uniformity of the circuit family can be further qualified by the complexity of these questions. For example, we talk of *polynomial-time uniformity* if these questions can be answered in polynomial time. Note that time here is measured with respect to the input length n of the circuit, not with respect to the length of the input of the questions themselves. This detail can be enforced by requiring that the input to the questions be followed by a string of n 1's.

The advanced course will deal with the very restrictive notion of *logarithmic-time uniformity*. In this lecture and the rest of the basic course, the more generous notion of *logarithmic-space uniformity* will be perfectly adequate.

As an example of a uniform circuit, consider the AC^1 circuit for NL that was presented in the previous lecture. The circuit is in two parts or stages. The first stage computes the configuration graph of the machine for the given input while the second stage computes the appropriate power of the adjacency matrix of that graph. It should be clear that the second stage is uniform. The key to seeing the details of this clearly is to carefully number the gates of the circuit so that their names make it easy to identify their role in the circuit. The first stage of the circuit is also uniform but now the machines answering the above questions will need to examine the input and consider the program of the machine we are simulating. This is all fine since knowledge of that program can be included in the program of the "constructor" machines.

2. Simulation of Circuits by Machines

In this section, we show that small-depth circuits can be simulated by space-efficient machines and that small-size circuits can be simulated by time-efficient machines.

Theorem 1 *Logspace-uniform* NC^1 *is contained in* L.

Proof The idea is to simply evaluate the circuit using a *depth-first search*, which can be defined inductively as follows: visit the output gate of the circuit, visit the gates of the left subcircuit, visit the gates of the right subcircuit. At any moment, we store the number of the current gate, the path that led us there (as a sequence of left's and right's) and any partial values that were already computed. For example, we could have L, R (0), R (1), L, L, 347. This would mean that we are visiting gate 347 and we got there by going left, right, right, left and left. The 0 after the first R indicates that the left input of the second gate on the path evaluated to 0.

When we are done visiting a gate (and computed its value), we return to the previous gate on the path. Note that we can recompute the number of that gate by following the path from the beginning. The space requirements of this algorithm are therefore determined by the maximum length of the path, which is equal to the depth of the circuit. The uniformity of the circuit is used when traveling through the circuit and evaluating its gates. □

Theorem 2 P*-uniform* PSIZE *is contained in* P.

Proof Here we simply use the polynomial-time circuit evaluation algorithm presented in the second lecture. First compute a representation of the circuit, using the fact that it is uniform, and then evaluate it on the given input. □

Now that we have these two simulations, we can combine our machine and circuit classes as follows. Here all circuit classes below L must be logspace uniform while those between NL and P can either be all logspace uniform or all P uniform.

$$\mathrm{AC}^0 \subseteq \mathrm{NC}^1 \subseteq \mathrm{L} \subseteq \mathrm{NL} \subseteq \mathrm{AC}^1 \subseteq \cdots \subseteq \mathrm{NC} \subseteq \mathrm{PSIZE} \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE} \subseteq \mathrm{EXP}.$$

3. Circuit Definition of P

In this section, we establish the converse of the previous simulation: uniform polynomial-size circuits can simulate polynomial-time machines. In fact, we will

obtain logspace-uniform circuits. This implies that the class P can be equivalently defined in terms of polynomial-time machines or uniform polynomial-size circuits.

Theorem 3 P = P-*uniform* PSIZE = *logspace-uniform* PSIZE.

Proof We have already shown that P-uniform PSIZE circuits can be evaluated in P. Logspace-uniform PSIZE is contained in P-uniform PSIZE because L $\subseteq$ P. Therefore, we only need to show that P is contained in logspace-uniform PSIZE.

The computation of a polynomial-time machine M on an input x of length n can be represented by a table. Each row of this table corresponds to a step of the computation and describes the configuration of M after that step. The first row is the initial configuration of M on x. Each row contains information about the current state of M, the location of its heads and the contents of its memory.

To each piece of information in each row, we associate a circuit. That circuit is responsible for computing that particular piece of information. The input to these circuits is the information from the previous row. For example, the circuit responsible for computing the contents of a particular memory location will examine the previous row for the precise conditions needed for that memory location to take on the various possible values. In a sense, these circuits encode the behavior of M.

The output of the circuit is simply whether the last row of the table contains the accept state. The size of each row is linear in the running time of M since M cannot use more memory than that. The total size of the table and of the circuit is therefore polynomial in the running time of M. Is not difficult to see that the circuit is uniform. Its overall structure is simple and its gate connections are easy to determine given the program of M. Note that the only subcircuits that need to access the input are those associated with the first row. □

Given this characterization of P in terms of uniform polynomial-size circuits, we can simplify our landscape of classes as follows. Once again, all the circuit classes below L must be logspace uniform while those between NL and P can either be all logspace uniform or all P uniform.

$$\mathrm{AC}^0 \subseteq \mathrm{NC}^1 \subseteq \mathrm{L} \subseteq \mathrm{NL} \subseteq \mathrm{AC}^1 \subseteq \cdots \subseteq \mathrm{NC} \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE} \subseteq \mathrm{EXP}.$$

Note that only one of these containments is known to be strict: $\mathrm{AC}^0 \subset \mathrm{NC}^1$. All the others have been conjectured to be strict but so far no proof has been found. However, it is known that NL $\subset$ PSPACE and that P $\subset$ EXP. This implies that some of these other containments must be strict.

4. Summary of Circuit-Machine Simulations

In this and the previous lectures, we have established several circuit-machine simulations that show how some resource in one model can be simulated by some other resource in the other model. Some of these simulations were done for specific classes but it is not hard to see that the techniques we used apply to a more general setting. In this section, we summarize the simulations that were obtained and state their generalizations.

First, we consider the simulation of circuit depth by machine space. Using the idea of a depth-first search, in this lecture we showed that logspace-uniform $\mathrm{NC}^1 \subseteq \mathrm{L}$. This can be generalized as follows:

Theorem 4 *Every language decided by* $\mathrm{DSPACE}(s)$*-uniform circuits of fan-in* 2 *and depth* s *can be decided in* $\mathrm{DSPACE}(s)$*.*

Second, we simulated circuit size by machine time when we showed that $\mathrm{PSIZE} \subseteq \mathrm{P}$ using our simple circuit evaluation algorithm. This can be generalized to

Theorem 5 *Every language decided by* $\mathrm{DTIME}(t)$*-uniform circuits of size* t *can be decided in* $\mathrm{DTIME}(t^{O(1)})$*.*

Third, in the previous lecture, we established a general simulation of machine space by circuit depth when we proved that every language in $\mathrm{NSPACE}(s)$ can be decided by a family of unbounded fan-in circuits of depth $O(s)$ and size $2^{O(s)}$, provided $s(n) \geq \log n$. As a consequence, $\mathrm{NL} \subseteq \mathrm{AC}^1$. The more general result can be strengthened by examining the uniformity of the resulting circuit in the proof. A first result is that the circuits are $\mathrm{DSPACE}(s^2)$ uniform. More uniformity is possible but under the condition that the function s be *space constructible*. That is, given 1^n, the binary representation of $s(n)$ can be computed in space $O(s(n))$.

Theorem 6 *Provided* $s(n) \geq \log n$*, every language in* $\mathrm{NSPACE}(s)$ *can be decided by the following:*

1. *a family of* $\mathrm{DSPACE}(s^2)$*-uniform unbounded fan-in circuits of depth* $O(s)$ *and size* $2^{O(s)}$*, and*
2. *a family of* $\mathrm{DSPACE}(s)$*-uniform unbounded fan-in circuits of depth* $O(s)$ *and size* $2^{O(s)}$*, if* s *is space constructible.*

Fourth and last, in this lecture we showed that P machines can be simulated by polynomial-size circuits. We did this by constructing a circuit that computes the successive configurations of the machine. This simulation of machine time by circuit size can be generalized as follows:

Theorem 7 *Every language in* DTIME(t) *can be decided by a family of* DSPACE($\log t$)*-uniform circuits of size* $t^{O(1)}$*, provided* $t(n) \geq n$ *and* $\log t$ *is space constructible.*

5. Exercises

1. Show that the NC^1 parity circuit is uniform.

2. Show that L-uniform AC^1 is contained in DSPACE($\log^2 n$). (Note that this provides an alternate proof that NL $\subseteq$ DSPACE($\log^2 n$).)

3. Provide further detail on the circuit in the proof of the polynomial-size simulation of P. In particular, justify the claim that the circuit is logspace uniform.

4. A circuit is *levelled* if its gates can be divided into sets $l_1, \ldots, l_d$ such that each gate at level i has children that are either gates at level $i-1$ or input gates. The *width* of such a circuit is the size of the largest level. Show that a language can be decided by logspace-uniform circuits of $O(\log n)$ width and polynomial size if and only if it is in L.

5. Use the general simulation results to obtain a different proof of Savitch's Theorem that NSPACE(s) is contained in DSPACE(s^2), provided $s(n) \geq \log n$.

6. Establish the more uniform version of the general space by depth simulation.

7. Show that NSPACE(s) $\subseteq$ DTIME($2^{O(s)}$). Do you need any hypotheses on the function s?