Complexity Theory

Alternating Complexity Classes

Daniel Borchmann, Markus Krötzsch

Computational Logic

2016-01-06

Review

Alternating vs. Deterministic Time and Space

From Alternating Time to Deterministic Space

Theorem 15.1

For $f(n) \geq n$*, we have* $\textsc{ATime}(f) \subseteq \textsc{DSpace}(f^2)$*.*

From Alternating Time to Deterministic Space

Theorem 15.1

For $f(n) \geq n$, we have $\text{ATIME}(f) \subseteq \text{DSPACE}(f^2)$.

Proof.

We simulate an ATM $\mathcal{M}$ using a TM $\mathcal{S}$:

- $\mathcal{S}$ performs a depth-first search of the configuration tree of $\mathcal{M}$
- The acceptance status of each node is computed recursively (similar to typical PSPACE algorithms we have seen before)
- $\mathcal{M}$ accepts exactly if the root of the configuration tree is accepting

The maximum recursion depth is $f(n)$. The maximum size of a configuration is $O(f(n))$. Hence the claim follows. □

Note: the result can be strengthened to $\text{ATIME}(f) \subseteq \text{DSPACE}(f)$ by not storing the whole configuration. See [Sipser, Lemma 10.22].

From Nondeterministic Space to Alternating Time

Theorem 15.2

For $f(n) \geq n$*, we have* $\mathrm{NSPACE}(f) \subseteq \mathrm{ATIME}(f^2)$*.*

From Nondeterministic Space to Alternating Time

Theorem 15.2

For $f(n) \geq n$, we have $\textsc{NSpace}(f) \subseteq \textsc{ATime}(f^2)$.

Proof.

We simulate an NTM $\mathcal{M}$ using an ATM $\mathcal{S}$.
Challenge: the computing paths of $\mathcal{M}$ might be up to $2^{df(n)}$ in length.

From Nondeterministic Space to Alternating Time

Theorem 15.2

For $f(n) \geq n$, we have $\mathrm{NSPACE}(f) \subseteq \mathrm{ATIME}(f^2)$.

Proof.

We simulate an NTM $\mathcal{M}$ using an ATM $\mathcal{S}$.
Challenge: the computing paths of $\mathcal{M}$ might be up to $2^{df(n)}$ in length.
Solution: recursively solve YIELDABILITY problems, as in Savitch's Theorem:

- We want to check if $\mathcal{M}$ can go from configuration C_1 to C_2 in at most k steps
- To do this, existentially guess an intermediate configuration C'.
- Universally check if $\mathcal{M}$ can go from C_1 to C' in $k/2$ steps, and from C' to C_2 in $k/2$ steps.

Storing one intermediate configuration C' takes space $O(f(n))$. Maximal recursion depth is $O(f(n))$. Hence the result follows. □

Harvest: Alternating Time = Deterministic Space

For $f(n) \geq n$, we have shown
$\textsc{ATime}(f) \subseteq \textsc{DSpace}(f^2)$ and $\textsc{DSpace}(f) \subseteq \textsc{NSpace}(f) \subseteq \textsc{ATime}(f^2)$.

The quadratic increase is swallowed by (super)polynomial bounds:

Corollary 15.3 ("Alternating Time = Deterministic Space")
$\textsc{APTime} = \textsc{PSpace}$ *and* $\textsc{AExpTime} = \textsc{ExpSpace}$.

Proof.

- $\textsc{ATime}(n^d) \subseteq \textsc{DSpace}(n^{2d}) \subseteq \textsc{PSpace}$
  $\textsc{DSpace}(n^d) \subseteq \textsc{NSpace}(n^d) \subseteq \textsc{ATime}(n^{2d}) \subseteq \textsc{APTime}$
- Second claim is left as an exercise

□

One can also read this as "Parallel Time = Sequential Space."

From Alternating Space to Deterministic Time

In this direction, the increase is exponential:

Theorem 15.4

For $f(n) \geq \log n$, we have $\textsc{ASpace}(f) \subseteq \textsc{DTime}(2^{O(f)})$.

From Alternating Space to Deterministic Time

In this direction, the increase is exponential:

Theorem 15.4

For $f(n) \geq \log n$, we have $\textsc{ASpace}(f) \subseteq \textsc{DTime}(2^{O(f)})$.

Proof.

The proof is similar to the exponential deterministic simulation of space-bounded NTMs in Lecture 13 (Theorem 10.7):

- Construct configuration graph of ATM
- Iteratively compute acceptance status of each configuration
- Check if starting configuration is accepting

Each step can be done in exponential time (in particular, computing the acceptance condition in each step is no more difficult than for plain NTMs). □

From Deterministic Time To Alternating Space

The exponential blow-up can be reversed when going back to ATMs:

Theorem 15.5

For $f(n) \geq \log n$ space-constructible, we have
$\textsc{DTime}(2^{O(f)}) \subseteq \textsc{ASpace}(f)$.

From Deterministic Time To Alternating Space

The exponential blow-up can be reversed when going back to ATMs:

Theorem 15.5

For $f(n) \geq \log n$ *space-constructible, we have* $\textsc{DTime}(2^{O(f)}) \subseteq \textsc{ASpace}(f)$*.*

Proof.

We show: for any $g(n) \geq n$, we have $\textsc{DTime}(g) \subseteq \textsc{ASpace}(\log g)$.

We simulate an TM $\mathcal{M}$ using an ATM $\mathcal{S}$. This is not so easy

From Deterministic Time To Alternating Space

The exponential blow-up can be reversed when going back to ATMs:

Theorem 15.5

For $f(n) \geq \log n$ space-constructible, we have $\text{DTIME}(2^{O(f)}) \subseteq \text{ASPACE}(f)$.

Proof.

We show: for any $g(n) \geq n$, we have $\text{DTIME}(g) \subseteq \text{ASPACE}(\log g)$.

We simulate an TM $\mathcal{M}$ using an ATM $\mathcal{S}$. This is not so easy:

- A computation of $\mathcal{M}$ is exponentially longer than the space available to $\mathcal{S}$ $\rightsquigarrow$ we solved this before with YIELDABILITY

From Deterministic Time To Alternating Space

The exponential blow-up can be reversed when going back to ATMs:

Theorem 15.5

For $f(n) \geq \log n$ space-constructible, we have $\mathrm{DTIME}(2^{O(f)}) \subseteq \mathrm{ASPACE}(f)$.

Proof.

We show: for any $g(n) \geq n$, we have $\mathrm{DTIME}(g) \subseteq \mathrm{ASPACE}(\log g)$.

We simulate an TM $\mathcal{M}$ using an ATM $\mathcal{S}$. This is not so easy:

- A computation of $\mathcal{M}$ is exponentially longer than the space available to $\mathcal{S}$ $\leadsto$ we solved this before with YIELDABILITY
- A configuration of $\mathcal{M}$ is exponentially longer than the space available to $\mathcal{S}$ $\leadsto$ this is more tricky ...

There is a coarse proof sketch in [Sipser, Lemma 10.25]. We follow a more detailed proof from the lecture notes of Erich Grädel [Complexity Theory, WS 2009/10] (link).

From Deterministic Time To Alternating Space (2)

Notation: The proof is easier if we write a configuration $\sigma_1 \cdots \sigma_{i-1} q \sigma_i \sigma_{i+1} \cdots \sigma_m$ as a sequence

$$* \; \sigma_1 \; \cdots \; \sigma_{i-1} \; \langle q, \sigma_i \rangle \; \sigma_{i+1} \; \cdots \; \sigma_m \; *$$

of symbols from the set $\Omega = \{*\} \cup \Gamma \cup (Q \times \Gamma)$.

From Deterministic Time To Alternating Space (2)

Notation: The proof is easier if we write a configuration $\sigma_1 \cdots \sigma_{i-1} q \sigma_i \sigma_{i+1} \cdots \sigma_m$ as a sequence

$$* \; \sigma_1 \; \cdots \; \sigma_{i-1} \; \langle q, \sigma_i \rangle \; \sigma_{i+1} \; \cdots \; \sigma_m \; *$$

of symbols from the set $\Omega = \{*\} \cup \Gamma \cup (Q \times \Gamma)$.

Then the Ω-symbol (state and tape) at position i follows deterministically from the Ω-symbols at positions $i-1$, i, and $i+1$ in the previous step. We write $\mathcal{M}(\omega_{i-1}, \omega_i, \omega_{i+1})$ for this symbol.

From Deterministic Time To Alternating Space (2)

Notation: The proof is easier if we write a configuration $\sigma_1 \cdots \sigma_{i-1} q \sigma_i \sigma_{i+1} \cdots \sigma_m$ as a sequence

$$* \, \sigma_1 \ \cdots \ \sigma_{i-1} \ \langle q, \sigma_i \rangle \ \sigma_{i+1} \ \cdots \ \sigma_m \, *$$

of symbols from the set $\Omega = \{*\} \cup \Gamma \cup (Q \times \Gamma)$.

Then the Ω-symbol (state and tape) at position i follows deterministically from the Ω-symbols at positions $i-1$, i, and $i+1$ in the previous step. We write $\mathcal{M}(\omega_{i-1}, \omega_i, \omega_{i+1})$ for this symbol.

Proof idea:

- Only store a pointers to one cell in one configuration of $\mathcal{M}$
- Verify the contents of current cell i in step j by guessing the previous cell contents $\omega_{i-1}, \omega_i, \omega_{i+1}$ in step j.
- Check iteratively that the guessed symbols are correct

From Deterministic Time To Alternating Space (3)

Let $h : \mathbb{N} \to \mathbb{R}$ be a space-constructible function in $O(g)$ that defines the exact time bound for $\mathcal{M}$ (no O-notation).

```
ATMSIMULATETM(TM M, input word w, time bound h) :
  existentially guess s ≤ h(|w|)      // halting step
  existentially guess i ∈ {0,...,s}   // halting position
  existentially guess ω ∈ Q × Σ       // halting cell + state
  if ω does not have a halting state :
    return FALSE
  for j = s,...,1 do :
    existentially guess ⟨ω_{-1}, ω_0, ω_1⟩ ∈ Ω^3
    if M(ω_{-1}, ω_0, ω_{+1}) ≠ ω : return FALSE
    universally choose ℓ ∈ {-1, 0, 1}
    ω := ω_ℓ
    i := i + ℓ
  // after tracing back s steps, check input configuration:
  return "input configuration of M on w has ω at position i"
```

A Remark About Space-Constructibility

Our algorithm needs space-constructibility of h to implement the line

```
02   existentially guess s ≤ h(|w|)    // halting step
```

However, we could also avoid this:

- The algorithm from line 03 on checks if the TM halts after s steps
- We can make a similar algorithm that checks if the TM does not halt after s steps
- We can then use an overall algorithm that increments s one by one (starting from 1):
  - For each value of s, guess if the TM halts after this time or not
  - Check the guess using the above procedures
  - Stop when the halting configuration has been found
- Because of the time bound on the simulated TM, s will not become larger than $2^{O(f)}$ here, so we can always store it in space f.

Harvest

For $f(n) \geq \log n$, we have shown $\mathrm{ASPACE}(f) = \mathrm{DTIME}(2^{O(f)})$.

Corollary 15.6 ("Alternating Space = Exponential Deterministic Time")
$\mathrm{AL} = \mathrm{P}$ *and* $\mathrm{APSPACE} = \mathrm{EXPTIME}$.

We can sum up our findings as follows:

$$\begin{array}{ccccccccc} \mathrm{L} & \subseteq & \mathrm{PTIME} & \subseteq & \mathrm{PSPACE} & \subseteq & \mathrm{EXPTIME} & \subseteq & \mathrm{EXPSPACE} \\ & & \parallel & & \parallel & & \parallel & & \parallel \\ & & \mathrm{ALOGSPACE} & \subseteq & \mathrm{APTIME} & \subseteq & \mathrm{APSPACE} & \subseteq & \mathrm{AEXPTIME} \end{array}$$