Κωστόπουλος Δημήτριος
Μ.Π.Λ.Α.

TAPE COMPRESSION (θεώρημα 2.3 – Παπαδημητρίου)

Εισαγωγή.

Αυτό το φυλλάδιο έχει στόχο να δώσει ένα ανάλογο αποτέλεσμα με αυτό του linear speedup θεωρήματος, εάν έχουμε μία μηχανή Turing M που χρησιμοποιεί f(n) χώρο για να λειτουργήσει, τότε μπορούμε να κατασκευάσουμε μία άλλη μηχανή Turing M', η οποία προσομοιώνει τη M και χρησιμοποιεί χώρο εf(n) + 2, όπου n είναι το μέγεθος της εισόδου. Το θεώρημα και η απόδειξη μοιάζουν πάρα πολύ με αυτή του linear speedup θεωρήματος. Παρόλα αυτά, πρώτα είναι σκόπιμο να κάνουμε μια συζήτηση.

k – string Turing μηχανές με είσοδο και έξοδο.

Όταν υπολογίζουμε το χώρο που καταλαμβάνει μια μηχανή Turing, την κρίνουμε ποιοτικά ως προς το πόσο είναι ικανή στο να σπαταλά όσο το δυνατό λιγότερο γίνεται χώρο. Αν λοιπόν στο χώρο αυτό υπολογίσουμε το χώρο που καταλαμβάνει η ταινία εισόδου, τότε είμαστε άδικοι με τη μηχανή αυτή, γιατί καμιά άλλη μηχανή δε μπορεί να σπαταλήσει λιγότερο χώρο, όσο καλά σχεδιασμένη και να είναι. Το ίδιο συμβαίνει και με την έξοδο. Η έξοδος, ότι και να κάνει η μηχανή, δε μπορεί να αλλάξει, θα καταλαμβάνει τον ίδιο χώρο. Το πιο λογικό θα έπρεπε να είναι να υπολογίζουμε μόνο τον επιπρόσθετο χώρο που καταλαμβάνει η μηχανή μας. Αυτό και κάνουμε.

Ορισμός.

Έστω ακέραιος k > 2. Μία **k – string μηχανή Turing με είσοδο και έξοδο** είναι μια κανονική k – string μηχανή Turing τέτοια ώστε για κάθε μετάβασή της δ, $\delta(q, \sigma_1, ..., \sigma_k) = (p, \rho_1, D_1, \dots, \rho_k, D_k)$:

α) $\rho_1 = \sigma_1$

β) $D_k \neq$ <-

γ) εάν $\sigma_1 = |\_|$, τότε D_1 = <-.

Λεπτομέρειες για τον ορισμό μπορείτε να βρείτε στο βιβλίο. Αυτό που μας ενδιαφέρει εμάς είναι ότι μία «k – string μηχανή Turing με είσοδο και έξοδο» είναι μια μηχανή με ταινία εισόδου μόνο για ανάγνωση και με μια ταινία εξόδου στην οποία γράφουμε μόνο το αποτέλεσμα, τις άλλες ταινίες θα τις καλούμε εσωτερικές ταινίες. Έτσι, ο χώρος που θα μετράμε τώρα θα είναι στις εσωτερικές ταινίες. Εύκολα (βλ. Βιβλίο) μπορεί κανείς να δείξει ότι οποιαδήποτε μηχανή Turing με k ταινίες μπορεί να μετατραπεί σε μία k+2 – string με είσοδο και έξοδο. Άρα μπορούμε από εδώ και πέρα να μιλάμε για k – string μηχανές Turing με είσοδο και έξοδο και να εννοούμε οποιαδήποτε μηχανή Turing (όσο αφορά το χώρο βέβαια).

Θεώρημα:

Έστω γλώσσα L ε SPACE(f(n)). Τότε, για κάθε ε > 0, η γλώσσα L ανήκει στο SPACE(2 + εf(n)).

Απόδειξη:

Έστω M = (K, Σ, δ, s) μία k – string Turing μηχανή με είσοδο και έξοδο η οποία αποφασίζει την L και χρησιμοποιεί χώρο f(n). Θα κατασκευάσουμε μία k – string Turing μηχανή με είσοδο και έξοδο M' = (K', Σ', δ', s') και η οποία θα χρησιμοποιεί χώρο εf(n) + 2, για ε > 0, με σκοπό να προσομοιώσει την M. Πώς το επιτυγχάνουμε αυτό; Κωδικοποιούμε πολλά σύμβολα σε ένα. Με τη συμπίεση των συμβόλων η μηχανή σπαταλά λιγότερο χώρο. Προσοχή, δε συμπιέζουμε την ταινία εισόδου και την ταινία εξόδου.

Το αλφάβητο της M' θα μεγαλώσει και θα περιέχει και τα νέα
σύμβολα – κωδικούς. Εάν για την κωδικοποίηση συμπιέζουμε m σύμβολα για να πάρουμε ένα νέο σύμβολο, τότε θα έχουμε $\Sigma' = \Sigma \cup \Sigma^m$.

Επειδή δε μας νοιάζει ο χρόνος με τον οποίο θα τρέχει η μηχανή, παρά μόνο ο χώρος, μας φτάνει μόνον το να χρησιμοποιούμε κωδικοποιημένα σύμβολα. Έτσι, θα μετατρέψουμε όλες τις μεταβάσεις της μηχανής M σε μεταβάσεις που να έχουν στην πρώτη και την τελευταία ταινία «παλαιά σύμβολα» και στην άλλες ταινίες νέα, κωδικοποιημένα σύμβολα.

Για κάθε παλιά μετάβαση θα πρέπει να φτιάξουμε μια νέα που να χρησιμοποιεί συμπιεσμένα σύμβολα στις εσωτερικές ταινίες. Πρέπει, επομένως, να ξέρουμε που στο συμπιεσμένο σύμβολο της κεφαλής μιας εσωτερικής ταινίας της M', βρίσκεται η κεφαλή της M. Για αυτό εμπλουτίζουμε λίγο τις καταστάσεις.

Για κάθε κατάσταση της M φτιάχνουμε νέες καταστάσεις $K \times \{1, 2, \ldots, m\}^{k-2}$.
Το K αντιστοιχεί στις παλιές καταστάσεις της M.
Το $\{1, 2, \ldots, m\}^{k-2}$ είναι μία k-2 – άδα που αντιστοιχεί στη θέση του κέρσορα της M στο σύμβολο που διαβάζει ο κέρσορας της M'. Για παράδειγμα εάν για την i – οστή εσωτερική ταινία ο κέρσορας της M βρίσκεται l θέσεις μακριά από το |>, στην i – οστή θέση της k-2 – άδας θα υπάρχει ο αριθμός l mod m.

Οπότε, εάν έχουμε αριστερή ή δεξιά κίνηση στην ταινία θα αλλάζει κατάλληλα και η k – άδα.

Προφανώς, θα χρησιμοποιούμε στις εσωτερικές ταινίες χώρο $\left\lceil \frac{f(|x|)}{m} \right\rceil$. Αν πάρουμε $m = \left\lceil \frac{1}{\varepsilon} \right\rceil$, έχουμε SPACE(f(|x|).-

Παρατηρήσεις.

Ανάλογα το μοντέλο που έχει υλοποιηθεί η μηχανή Turing και την ακρίβεια που μετράμε και εκτελούμε τις πράξεις, οι σταθεροί παράγοντες ποικίλλουν. Αλλά σε όλα τα μοντέλα, επιτυγχάνουμε επιτάχυνση στο χρόνο κατά ένα γραμμικό παράγοντα. Για παράδειγμα εγώ έβγαλα άλλους σταθερούς παράγοντες από αυτούς του Παπαδημητρίου, αλλά αυτό δε μας πειράζει φαντάζομαι.

Τι μας λέει αυτό το θεώρημα; Για κάθε μηχανή Turing η οποία απαιτεί χώρο f(n), τότε μπορούμε να φτιάξουμε άλλη μία τέτοια που να απαιτεί χώρο εf(n) + 2. Δηλαδή εάν η μηχανή Turing θέλει $10n^2$ χώρο, όπου n είναι το μήκος της εισόδου, τότε μπορούμε να κατασκευάσουμε άλλη μια Turing μηχανή τέτοια που να καταλαμβάνει χώρο $\varepsilon 10n^2 + 2$. Για ε = 100 > 0, παίρνουμε ότι η μηχανή μας θα χρειαστεί χώρο $\frac{n^2}{10} + 2$.

Παράδειγμα για το TAPE COMPRESSION theorem (δικό μου, μπορεί να έχει λάθη).

Έστω μια μηχανή Turing με είσοδο και έξοδο που υπολογίζει πόσο είναι η απόλυτη διαφορά μεταξύ των 0 και 1 σε ένα μη κενό δυαδικό αριθμό. Έχουμε
Σ = {0, 1, |>, |_|}
K = {s, q, p, r, t, h}
k = 3, (1 ταινία εισόδου, 3 η ταινία εξόδου, 2 η εσωτερική ταινία)

m = 2
x = 0110111010111101011110110

M

<table>
<tr><td></td><td>p ε K</td><td>σ_1 ε Σ</td><td>σ_2 ε Σ</td><td>σ_3 ε Σ</td><td>Δ(p, σ_1, D_1, σ_2, D_2, σ_3, D_3)</td></tr>
<tr><td>1</td><td>S</td><td>|></td><td>|></td><td>|></td><td>(s, |>, ->, |>, ->, |>, ->)</td></tr>
<tr><td>2</td><td>S</td><td>0</td><td>|_|</td><td>|_|</td><td>(p, 0, ->, 0, ->, |_|, -)</td></tr>
<tr><td>3</td><td>S</td><td>1</td><td>|_|</td><td>|_|</td><td>(q, 1, ->, 1, ->, |_|, -)</td></tr>
<tr><td>4</td><td>P</td><td>0</td><td>|_|</td><td>|_|</td><td>(p, 0, ->, 0, ->, |_|, -)</td></tr>
<tr><td>5</td><td>Q</td><td>1</td><td>|_|</td><td>|_|</td><td>(q, 1, ->, 1, ->, |_|, -)</td></tr>
<tr><td>6</td><td>Q</td><td>0</td><td>|_|</td><td>|_|</td><td>(r, 0, ->, |_|, <-, |_|, -)</td></tr>
<tr><td>7</td><td>P</td><td>1</td><td>|_|</td><td>|_|</td><td>(r, 1, ->, |_|, <-, |_|, -)</td></tr>
<tr><td>8</td><td>R</td><td>0 ή 1 ή |_|</td><td>|></td><td>|_|</td><td>(s, 0 ή 1 ή |_|, -, |>, ->, |_|, -)</td></tr>
<tr><td>9</td><td>R</td><td>0 ή 1 ή |_|</td><td>0</td><td>|_|</td><td>(p, 0 ή 1 ή |_|, -, 0, ->, |_|, -)</td></tr>
<tr><td>10</td><td>R</td><td>0 ή 1 ή |_|</td><td>1</td><td>|_|</td><td>(q, 0 ή 1 ή |_|, -, 1, ->, |_|, -)</td></tr>
<tr><td>11</td><td>s ή p ή q</td><td>|_|</td><td>|_|</td><td>|_|</td><td>(t, |_|, -, |_|, <-, |_|, -)</td></tr>
<tr><td>12</td><td>T</td><td>|_|</td><td>0</td><td>|_|</td><td>(t, |_|, -, 0, <-, 0, ->)</td></tr>
<tr><td>13</td><td>T</td><td>|_|</td><td>1</td><td>|_|</td><td>(t, |_|, -, 1, <-, 1, ->)</td></tr>
<tr><td>14</td><td>T</td><td>|_|</td><td>|></td><td>|_|</td><td>(h, |_|, -, |>, ->, |_|, -)</td></tr>
</table>

s είναι η κατάσταση «ισοπαλία».
p είναι η κατάσταση «τα μηδέν είναι περισσότερα».
q είναι η κατάσταση «οι άσσοι είναι περισσότεροι».
r είναι η κατάσταση «πρέπει να βρω τι είναι πιο πολύ».
t είναι η κατάσταση «γυρίζουμε την ταινία 2 πίσω».
u είναι η κατάσταση «αντίγραψε την ταινία 2 στην ταινία3».
Με λίγα λόγια αυτό που κάνει το πρόγραμμα είναι κάθε φορά που διαβάζει ένα σύμβολο να πηγαίνει είτε να το γράφει στην ταινία 2, είτε να σβήνει ένα σύμβολο από την ταινία 2, ανάλογα με το ποιο σύμβολο κερδίζει εκείνη τη στιγμή. Οπότε, στο τέλος στην ταινία 2 υπάρχει η απόλυτη διαφορά των δύο συμβόλων.

Ο νέος πίνακας μετάβασης θα γίνει.

M’

<table>
<tr><td></td><td>p ε K’</td><td>σ_1 ε Σ’</td><td>σ_2 ε Σ’</td><td>σ_3 ε Σ’</td><td>Δ’(p, σ_1, D_1, σ_2, D_2, σ_3, D_3)</td></tr>
<tr><td>1</td><td>(s, 0)</td><td>|></td><td>|></td><td>|></td><td>((s, 0), |>, ->, |>, ->, |>, ->)</td></tr>
<tr><td>2</td><td>(s, 0)</td><td>0</td><td>|_|</td><td>|_|</td><td>((p, 1), 0, ->, $\sigma_{0|\_|}$, -, |_|, -)</td></tr>
<tr><td>3</td><td>(s, 0)</td><td>1</td><td>|_|</td><td>|_|</td><td>((q, 1), 1, ->, $\sigma_{1|\_|}$, -, |_|, -)</td></tr>
<tr><td>4</td><td>(p, 0)</td><td>0</td><td>|_|</td><td>|_|</td><td>((p, 1), 0, ->, $\sigma_{0|\_|}$, -, |_|, -)</td></tr>
<tr><td></td><td>(p, 1)</td><td>0</td><td>$\sigma_{0|\_|}$</td><td>|_|</td><td>((p, 0), 0, ->, σ_{00}, ->, |_|, -)</td></tr>
<tr><td>5</td><td>(q, 0)</td><td>1</td><td>|_|</td><td>|_|</td><td>((q, 1), 1, ->, $\sigma_{1|\_|}$, -, |_|, -)</td></tr>
<tr><td></td><td>(q, 1)</td><td>1</td><td>$\sigma_{1|\_|}$</td><td>|_|</td><td>((q, 0), 1, ->, σ_{11}, ->, |_|, -)</td></tr>
<tr><td>6</td><td>(q, 0)</td><td>0</td><td>|_|</td><td>|_|</td><td>((r, 1), 0, ->, |_|, <-, |_|, -)</td></tr>
<tr><td></td><td>(q, 1)</td><td>0</td><td>$\sigma_{1|\_|}$</td><td>|_|</td><td>((r, 0), 0, ->, |_|, -, |_|, -)</td></tr>
<tr><td>7</td><td>(p, 0)</td><td>1</td><td>|_|</td><td>|_|</td><td>((r, 1), 1, ->, |_|, <-, |_|, -)</td></tr>
<tr><td></td><td>(p, 1)</td><td>1</td><td>$\sigma_{0|\_|}$</td><td>|_|</td><td>((r, 0), 1, ->, |_|, -, |_|, -)</td></tr>
<tr><td>8</td><td>(r, 0)</td><td>0 ή 1 ή |_|</td><td>|></td><td>|_|</td><td>((s, 0), 0 ή 1 ή |_|, -, |>, ->, |_|, -)</td></tr>
<tr><td>9</td><td>(r, 0)</td><td>0 ή 1 ή |_|</td><td>$\sigma_{0|\_|}$</td><td>|_|</td><td>((p, 1), 0 ή 1 ή |_|, -, $\sigma_{0|\_|}$, -, |_|, -)</td></tr>
<tr><td></td><td>(r, 1)</td><td>0 ή 1 ή |_|</td><td>$\sigma_{00|}$</td><td>|_|</td><td>((p, 0), 0 ή 1 ή |_|, -, $\sigma_{00|}$, ->, |_|, -)</td></tr>
<tr><td>10</td><td>(r, 0)</td><td>0 ή 1 ή |_|</td><td>$\sigma_{1|\_|}$</td><td>|_|</td><td>((q, 1), 0 ή 1 ή |_|, -, $\sigma_{1|\_|}$, -, |_|, -)</td></tr>
<tr><td></td><td>(r, 1)</td><td>0 ή 1 ή |_|</td><td>$\sigma_{11|}$</td><td>|_|</td><td>((q, 0), 0 ή 1 ή |_|, -, $\sigma_{11|}$, ->, |_|, -)</td></tr>
<tr><td>11</td><td>(s, 0) ή (p, 0) ή (q, 0)</td><td>|_|</td><td>|_|</td><td>|_|</td><td>((t, 1), |_|, -, |_|, <-, |_|, -)</td></tr>
<tr><td></td><td>(s, 1) ή (p, 1) ή (q, 1)</td><td>|_|</td><td>$\sigma_{1|\_|}$ ή $\sigma_{0|\_|}$</td><td>|_|</td><td>((t, 0), |_|, -, $\sigma_{1|\_|}$ ή $\sigma_{0|\_|}$, <-, |_|, -)</td></tr>
<tr><td>12</td><td>(t, 0)</td><td>|_|</td><td>$\sigma_{0|\_|}$</td><td>|_|</td><td>((t, 1), |_|, -, $\sigma_{0|\_|}$, <-, 0, ->)</td></tr>
<tr><td></td><td>(t, 1)</td><td>|_|</td><td>σ_{00}</td><td>|_|</td><td>((t, 0), |_|, -, σ_{00}, -, 0, ->)</td></tr>
<tr><td></td><td>(t, 0)</td><td>|_|</td><td>σ_{00}</td><td>|_|</td><td>((t, 1), |_|, -, σ_{00}, <-, 0, ->)</td></tr>
<tr><td>13</td><td>(t, 0)</td><td>|_|</td><td>$\sigma_{1|\_|}$</td><td>|_|</td><td>((t, 1), |_|, -, $\sigma_{1|\_|}$, <-, 1, ->)</td></tr>
<tr><td></td><td>(t, 1)</td><td>|_|</td><td>σ_{11}</td><td>|_|</td><td>((t, 0), |_|, -, σ_{11}, -, 1, ->)</td></tr>
<tr><td></td><td>(t, 0)</td><td>|_|</td><td>σ_{11}</td><td>|_|</td><td>((t, 1), |_|, -, σ_{11}, <-, 1, ->)</td></tr>
<tr><td>14</td><td>(t, 0)</td><td>|_|</td><td>|></td><td>|_|</td><td>(h, |_|, -, |>, ->, |_|, -)</td></tr>
</table>

¡Στους δύο πίνακες έχω βάλει μόνο τις πιθανές μεταβάσεις που μπορούμε να συναντήσουμε για κάθε είσοδο!

Παρακάτω δίνεται σχηματικά η μηχανή Turing M’ με είσοδο και έξοδο που θα προσομοιώνει τη M. Αφήνω σαν άσκηση αφού έχετε και το πρόγραμμα της M’ να τρέξετε τη μηχανή για την είσοδο x.

M’

|>	0	1	1	0	1	1	1	0	1	0	1	1	1	1	0	1	0	1	1	1	0	1	1	0	

|>																									

|>																									

FINE